Tools for Creating OER

Tools for Creating OER

Selecting appropriate Technologies

ISAAC MULOLANI

UNIVERSITY OF REGINA
REGINA

Contents

Part I. Main Body

Acknowledgements

ISAAC MULOLANI

This guide is the result of a final capstone project for the SPARC Open Education Leadership Program for the Class of 2022. Participation in this program was funded by the University of Regina through the office of the Associate Vice-President Academic and the Center for Teaching and Learning.

Special Thanks

- To Dr. Nilgun Onder for providing oversight and support for the OER Publishing Program
- To Dr. Alec Couros for providing the opportunity to integrate work within the Center for Teaching and Learning
- To the SPARC Open Education Leadership Program, for their continued support and connections.
- To Brad Ost for being my SPARC mentor and taking the time to review this guide.

ISAAC MULOLANI

The use of open education is growing and has become a global movement. Across much of North America, most post-secondary institutions are in the process of integrating the use of open education resources (OER) into their teaching and learning activities. The number of OER repositories from which instructors can draw resources continues to grow each year.

In addition, the number of different technology tools used to develop these resources also continues to increase. These digital technologies include both commercial and open-source options. Since the open-source movement preceded the open education movement, there are significant numbers of open-source tools available for OER creation.

This resource is intended to provide the OER community with a summary of some of the currently available digital technologies for creating open content. The decision on which technology to use is one that every OER creator will eventually need to make. This guide is designed to provide a starting point particularly for instructors and faculty at post-secondary institutions. The guide should help prospective creators of OER pick both the most appropriate tool for their specific context as well as their level of technical expertise.

Chapters 1 – 4 and **8** of this resource are accessible to all audiences. **Chapters 5 – 7** contain more technical information related to open-source tools and may not be as accessible for people new to Open Education. It may also be inaccessible for those with low to medium technical skill levels. This is normal since open-source tools require a high degree of technical expertise. For anyone new to the field, these can be tackled last.

Chapter 1 starts with very basic information on the definition and description of what constitutes OER. There is a description of open licensing and specifically, the Creative Commons licenses. This includes a description of the 5 R Framework provided by David Wiley. The ALMS Framework is described along with how it allows creators to create and remix OER. The importance of this framework when using Creative Commons licenses is highlighted.

Chapter 2 introduces the United Nations Sustainable Development Goals by providing a brief listing of each goal. The discussion progresses to discussing how digital technology can be leverage to achieve these goals particularly those in Education. A discussion on technology frameworks and the affordances digital technologies can potentially provide is crucial to this conversation on selecting appropriate tools for OER creation.

Chapter 3 focuses on commercial word processing tools. It starts by briefly discussing Microsoft Word, a tool that most people around the globe are familiar with. The specific features of MS Word useful for the creation of OER are described. In addition, the Softmaker Office tool is described along with features helpful for OER creation. Another useful tool described in this chapter is WPS Office. A description of its useful features is provided including its built-in PDF creator and convertor. Comparison of features available for each of these tools is provided at the end of the chapter.

Chapter 4 describes a number of open-source word processing and additional tools. The word processing tools include LibreOffice and Google Documents. Two additional technologies that can be used in combination with word processors are Gnu Image Manipulation Program (Gimp) and Inkscape. A simple description of the features useful for OER creation is provided with each of these.

Chapter 5 introduces the basic open-source TeX-based systems that arose out of the open-source software movement. These include MiKTeX, TeXlive and MacTeX on Windows, Unix/Linux and Mac systems respectively. Some of the relevant tools that can be used to help create OER are briefly described. An introduction to editors, both commercial and open-source, that can be used with TeX systems to automate the workflow is provided.

Chapter 6 delves further into TeX-based open-source tools by highlighting some packages useful for content creation. There is a discussion on including graphical elements and mathematical content into a document. Some presentation tools of relevance to instructors creating content for the classroom are also listed for reference.

Chapter 7 continues discussing TeX-based tools further by describing some other helpful tools such as LyX and TeXmacs that have a user interface designed to simplify the editing process. Two tools that allow collaborative editing are presented: Overleaf and Papeeria. R, a more hybrid open-source development tool useful for instructors who work with statistical content is introduced. The RStudio editor with its ability to use RMarkdown is discussed briefly. RStudio can be used to create multiple formats from a single file which is extremely helpful especially for academic use.

Chapter 8 provides a description of some of emerging OER tools such as Pressbooks, EdTech Books and LibreTexts. Pressbooks networks are becoming more widely used across North American colleges and universities. LibreTexts is an OER repository with increasing numbers of resources and course ware, particularly in STEM areas. More recently, a homework system and H5P repository have also been added to LibreTexts' offerings. To round out this chapter, the Ximera TeX-based system is described specifically for instructors who are open-source users. Another tool of interest for open-source users is the PreTeXt system. Both Ximera and PreTeXt allow a user to create multiple formats from a single source file thereby increasing the accessibility of a resource. These tools also provide interactive online documents in addition to static PDF files increasing both the engagement and accessibility of resources.

Accessibility Statement

ISAAC MULOLANI

The University of Regina OER Program believes that education must be available to everyone which means supporting the creation of free, open, and accessible educational resources. We are actively committed to increasing the accessibility and usability of the textbooks we produce.

Accessibility features of the web version of this resource

The web version of the *Tools for Creating OER* has been designed with accessibility in mind by incorporating the following features:

- It has been optimized for people who use screen-reader technology.
    - all content can be navigated using a keyboard
    - links, headings, and tables are formatted to work with screen readers
    - images have alt tags
- Information is not conveyed by colour alone.
- There is an option to increase font size (see tab on top right of screen).

Other file formats available

In addition to the web version, this book is available in a number of file formats including PDF, EPUB (for eReaders), MOBI (for Kindles), and various editable files. Here is a link to where you can download this book in another file format. Look for the "Download this book" drop-down menu to select the file type you want.

This book links to a number of external websites. For those using a print copy of this resource, the link text is underlined, and you can find the web addresses for all links in the back matter of the book.

Known accessibility issues and areas for improvement

While we strive to ensure that this resource is as accessible and usable as possible, we might not always get it right. Any issues we identify will be listed below. There are currently no known issues.

List of Known Accessibility Issues

Location of issue	Need for improvement	Timeline	Work around

Accessibility standards

The web version of this resource has been designed to meet Web Content Accessibility Guidelines 2.0, level AA. In addition, it follows all guidelines in Appendix: Checklist for Accessibility. The development of this toolkit involved working with students with various print disabilities who provided their personal perspectives and helped test the content.

Let us know if you are having problems accessing this toolkit

We are always looking for ways to make our resources more accessible. If you have problems accessing this resource, please contact us to let us know so we can fix the issue.

Please include the following information:

- The location of the problem by providing a web address or page description
- A description of the problem
- The computer, software, browser, and any assistive technology you are using that can help us diagnose and solve your issue
  - e.g., Windows 10, Google Chrome (Version 65.0.3325.181), NVDA screen reader

You can contact through the OER Program manager by e-mail.

This statement was last updated on December 15 , 2021.

Attribution

ISAAC MULOLANI

Suggested attribution:

Tools for Creating OER, by Isaac Mulolani, University of Regina, available under a Creative Commons Attribution 4.0 International License.

Tools for Creating OER was adapted for the University of Regina from:

- *The OER Starter Kit*, by Abbey Elder, Iowa State University, available under a Creative Commons Attribution 4.0 International License.
- *The OER Starter Kit Workbook* by Abbey Elder and Stacy Katz, available under a Creative Commons Attribution 4.0 International License.
- *The OER and Alternative Textbook Handbook* by Ariana Santiago, University of Houston, available under a Creative Commons Attribution 4.0 International License.
- *Rethinking Assessment Strategies for Online Learning* by Seneca College; Durham College; Algonquin College; and University of Ottawa, available under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, except where otherwise noted.
- *Introduction to Sustainable Development Goals (SDGs)* by Jocelyn Baker. This work is licensed under a Creative Commons Attribution 4.0 International License, except where otherwise noted.
- *Teaching in a Digital Age – Second Edition* by A. W. (Tony) Bates. This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License, except where otherwise noted.
- *Rethinking Pedagogy: Exploring the Potential of Digital Technology in Achieving Quality Education*. Aleman de la Garza, L., Antal, P. t., Beaune, A. I., Bruillard, E. R., Burke, D., … Tsinakos, A. This work is under an Attribution-ShareAlike 3.0 license.
- Cover image attribution: Open Educational Resources – Open 24/7 by Eugene Open Educational Resources. This work is marked with a CC BY-NC-SA 2.0 license.

Additional adapted resources

The OER Starter Kit by Abbey Elder was adapted from the following resources:

Introduction to Open Educational Resources, “Benefits for Instructors” was adapted from the SUNY OER Community Course, licensed CC BY 4.0.

Copyright & Open Licensing, “Licensing” and “Public Domain” were adapted in part from UH OER Training by Billy Meinke, University of Hawai’i Outreach College, and collaborators, licensed CC BY 4.0.

Creative Commons Licenses was adapted from The ABOER Starter Kit by Technologies in Education at the Faculty of Education, the University of Alberta, CC BY 4.0.

Evaluating OER was adapted from the Affordable Learning Georgia Selecting Textbooks webpage.

Diversity & Inclusion was adapted from Including all students by SUNY OER Services, licensed CC BY 4.0.

Planning & Completing Your OER Project was adapted from UH OER Training by Billy Meinke and University of Hawai’i Outreach College, licensed CC BY 4.0.

Tools & Techniques for Creating OER was adapted from the SPARC Open Education Primer by the SPARC Open Education Leadership Program, licensed CC BY 4.0.

Accessibility & Usability was adapted from The ABOER Starter Kit by Technologies in Education at the Faculty of Education, the University of Alberta, CC BY 4.0., the Affordable Learning Georgia Accessibility webpage and UH OER Training by Billy Meinke and University of Hawai’i Outreach College, licensed CC BY 4.0.

Versioning History

ISAAC MULOLANI

This page provides a record of edits and changes made to this text since its initial publication. Whenever edits or updates are made in the text, we provide a record and description of those changes here.

If you have a correction or recommendation you would like to suggest, please contact the author at Open.Textbooks@uregina.ca

Version	Date	Type	Description	Page
1.00	12 May 2022	original text		

1. Open Educational Resources Defined

ISAAC MULOLANI

Learning Outcomes

By the end of this chapter, the reader will be able to:

1. Define open educational resources
2. List the different Creative Commons licenses
3. Describe the 5R framework

Open Educational Resources

In the 1980s Richard Stallman initiated the Gnu Project to write a complete operating system free from constraints on the use of its source code.[1] This led to the Free Software Definition by the Free Software Foundation in 1986:

> The word "free" in our name does not refer to price; it refers to freedom. First, the freedom to copy a program and redistribute it to your neighbors, so that they can use it as well as you. Second, the freedom to change a program, so that you can control it instead of it controlling you; for this, the source code must be made available to you.

The definition was further refined to its current modern form to include the following four freedoms (See Wikipedia):

1. The freedom to run the program as you wish, for any purpose (freedom 0).
2. The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
3. The freedom to redistribute copies so you can help your neighbor (freedom 2).
4. The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can

---

[1] See Wikipedia.

give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

The key issue to note is that Freedom 1 and 3 require source code to be available because studying and modifying software without its source code is highly impractical. This is a key element of the open-source movement – providing source code for each project for others to be able to modify as needed. In 1989, the Gnu General Public License was published. Although there is no commonly agreed upon definition of Free and Open Source Software (FOSS), various groups maintain approved lists of licenses (see Wikipedia). From the list provided on Wikipedia, there are over 50 different **open source** licenses currently available.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=5#h5p-6

This glance at the **open-source** movement is deliberate because it was a pre-cursor to open educational resources. Under the open-source movement, there were people who were creating educational materials for their courses putting them under an open license (GPL) and sharing them with others in the community.

Open-source doesn't just mean access to the source code. The distribution terms of **open source** software must comply with the following criteria:

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=5#h5p-7

The **open-source** software tools commonly used were TeX-based tools in the majority of the cases. TeX user groups (TUG mirrors) sprang up across the globe where people would share **code snippets** with one another and sometimes links to **source code** that was downloadable for users. The global sharing that has been occuring in the open-source community since the 1990s forked into the open educational resource movement in the 2000s.

Open Licensing

One of the key issues borrowed by the open education movement from the open-source movement is the idea of **open licensing**. What sets copyrighted resources apart from open resources is the license placed on open resources. These licenses allow users to remix, rework, adapt, retain redistribute copies of the resource. In short, the open license provides the user of a resource with a set of permissions allowing some or all of these activities. Before describing licensing, a description of the allowable permissions is provided.

The 5R Framework

David Wiley provided this 5 R framework that is key to defining **open educational resources.**[2] Open Educational Resources refers to any copyrighted work that is either

1. in the public domain or
2. licensed in a manner that provides everyone with free and perpetual permission to engage in the 5R activities.

These 5 R activities/permissions are:

1. **Retain** – make, own, and control a copy of the resource (e.g., download and keep your own copy)
2. **Revise** – edit, adapt, and modify your copy of the resource (e.g., translate into another language)
3. **Remix** – combine your original or revised copy of the resource with other existing material to create something new (e.g., make a **mashup**)
4. **Reuse** – use your original, revised, or remixed copy of the resource publicly (e.g., on a website, in a presentation, in a class)
5. **Redistribute** – share copies of your original, revised, or remixed copy of the resource with others (e.g., post a copy online or give one to a friend)

It is important to note that not all the open licenses provide all five of these permissions. Some of them carry restrictions that the user needs to be aware of by noting license requirements. For example, some licenses restrict commercial use while others require the adapted resource be released under the same license as the original.

The ALMS framework

As David Wiley explains on the opencontent.org definition page, **Creative Commons licenses** give us permissions to exercise the 5 R's (reuse, revise, remix, redistribute, retain), but poor technical choices can make open content less open (and thus, harder to work with).

The **ALMS Framework** provides a way of thinking about those technical choices and understanding the degree to which they enable or impede a user's ability to engage in the 5R activities permitted by open licenses. The framework includes four buckets (or whichever your preferred container is) that questions about the technical openness of an OER likely fit into. Here are the descriptions of each bucket.

(This material is based on original writing by David Wiley, which was published freely under a Creative Commons Attribution 4.0 license at http://opencontent.org/definition/)

Access to Editing Tools

Is the open content published in a format that can only be revised or remixed using tools that are extremely expensive (e.g., 3DS MAX)? Is the open content published in an exotic format that can only be revised or remixed

---

[2] See Open Content Definition.

using tools that run on an obscure or discontinued platform (e.g., OS/2)? Is the open content published in a format that can be revised or remixed using tools that are freely available and run on all major platforms (e.g., OpenOffice)?

Takeaway: Can you edit the OER without the need for specialized or expensive tools?

Level of Expertise Required

Is the open content published in a format that requires a significant amount technical expertise to revise or remix (e.g., Blender)? Is the open content published in a format that requires a minimum level of technical expertise to revise or remix (e.g., Word)?

Takeaway: Would most faculty be able to edit the OER at their current skill level?

Meaningfully Editable

Is the open content published in a manner that makes its content essentially impossible to revise or remix (e.g., a scanned image of a handwritten document)? Is the open content published in a manner making its content easy to revise or remix (e.g., a text file)?

Takeaway: Can all parts of the OER be edited?

A word about the **ALMS framework** is in order with respect to this particular resource. In the remaining chapters of this guide, tools that require different levels of expertise will be described and discussed. Some of the tools that are word processors tend to require lower to medium technical expertise. In the later chapters, the open-source tools tend to require higher technical expertise.

Self-Sourced

Is the format preferred for consuming the open content the same format preferred for revising or remixing the open content (e.g., HTML)? Is the format preferred for consuming the open content different from the format preferred for revising or remixing the open content (e.g. Flash FLA vs SWF)?

Takeaway: Can you edit the OER directly or is a separate editable file needed?

Creative Commons Licenses

For OER, the most widely used open licenses are the Creative Commons (CC) licenses, which make it possible for educators to freely and legally share their work. **Creative Commons licenses** work with copyright to automatically give users a set of usage rights pertaining to that work. When something is licensed with a Creative Commons license, users know how they are allowed to use it. Since the copyright holder retains copyright, the user may still seek the creator's permission when they want to reuse the work in a way not permitted by the license.

License Terms

Creators or copyright holders who wish to apply a Creative Commons license to their work can choose the conditions of reuse and modification by selecting one or more of the restrictions listed below. Every Creative Commons license except the Public Domain designation requires users to give attribution to the creator of the work. Other restrictions are optional and may prevent reuse in unintended ways, so care is suggested in selecting a license.

Icon	Right	Description
🅯	**Attribution (BY)**	Licensees may copy, distribute, display, perform and make derivative works and remixes based on it only if they give the author or licensor the credits (attribution) in the manner specified by these. Since version 2.0, all Creative Commons licenses require attribution to the creator and include the BY element.
🄏	**Non-Commercial (NC)**	Licensees may copy, distribute, display, perform the work and make derivative works and remixes based on it only for non-commercial purposes.
🄎	**Share Alike (SA)**	Licensees may distribute derivative works only under a license identical to ("not more restrictive than") the license that governs the original work. (See also copyleft.) Without share-alike, derivative works might be sublicensed with compatible but more restrictive license clauses, e.g. CC BY to CC BY-NC.)
⊜	**No Derivative Works(ND)**	Licensees may copy, distribute, display and perform only verbatim copies of the work, not derivative works and remixes based on it. Since version 4.0, derivative works are allowed but must not be shared. *Note: works licensed with the ND restriction are not considered OER.*

License Types

There are six possible licenses that can be derived from combining the license terms described above and assigned to materials by the original creator or author. To learn more about the license designs, rationale, and structure of **Creative Commons licenses**, please read About the licenses by Creative Commons.

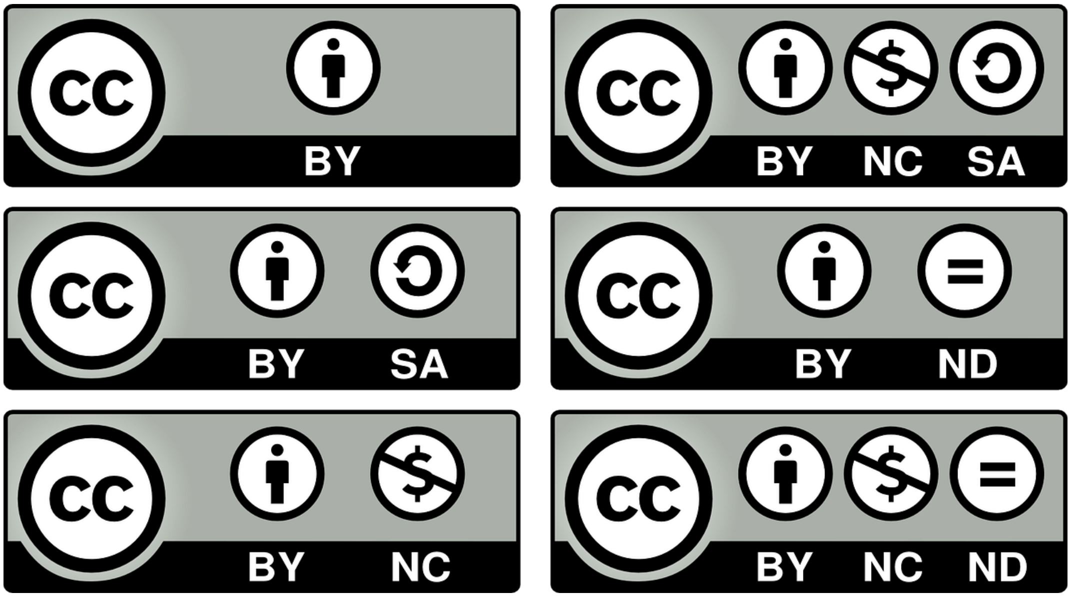


Figure 1.1 The six Creative Commons Licenses with their attributes.

It is helpful to point out that the two **Creative Commons licenses** at the bottom of the second stack in the image (CC-BY-ND and CC-BY-NC-ND) are not open educational resources. For open educational resources in education, the power comes from being able to revise, remix and redistribute. When the resources can only be used as they are then we are almost back to using copyrighted content. The following graphic depicts this fact.

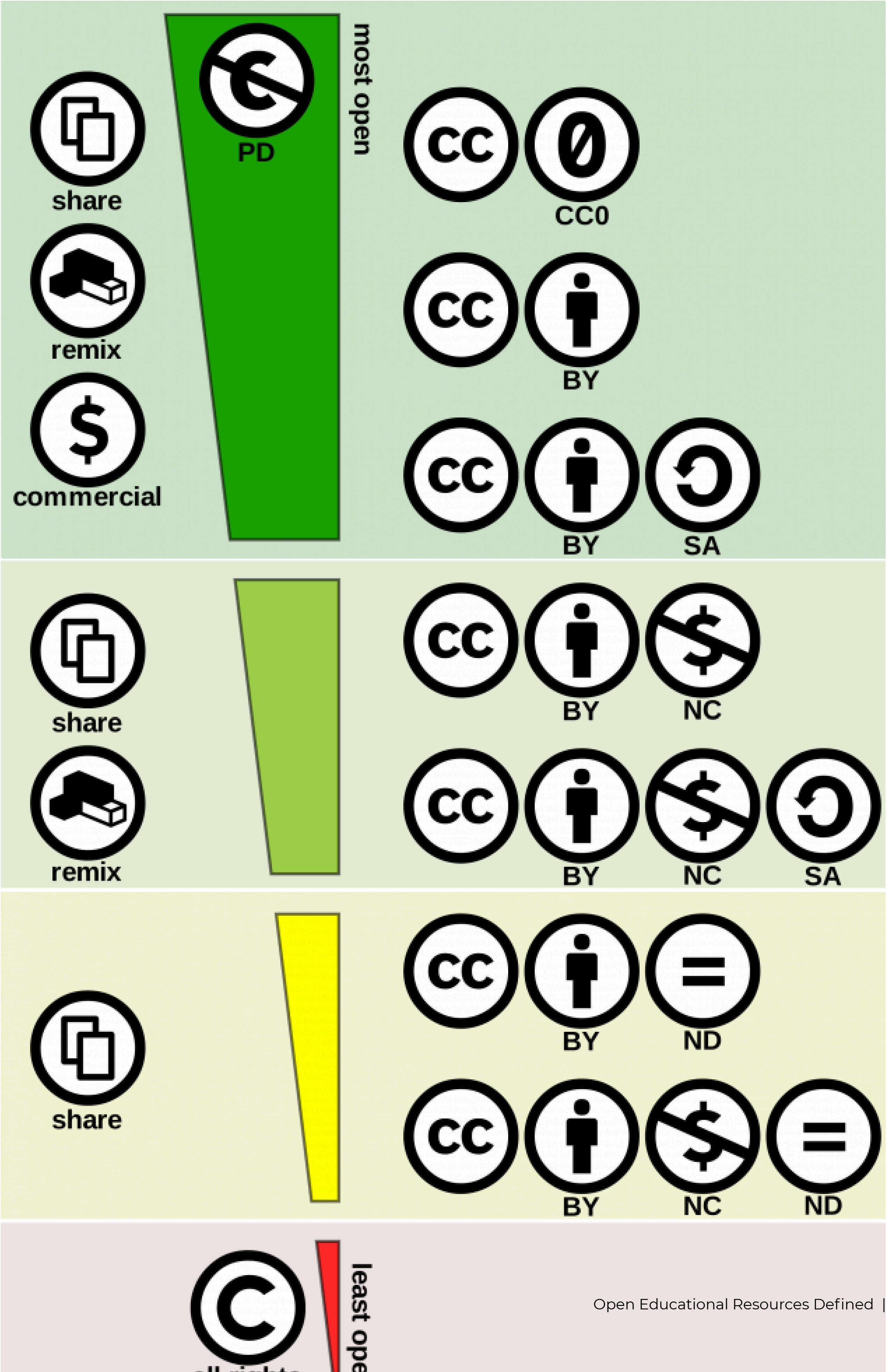

most open
PD
share
remix
commercial
CC0
BY
BY SA
share
remix
BY NC
BY NC SA
share
BY ND
BY NC ND
least ope

Fig 1.2 Creative Commons licenses from most to least open

Technical Expertise

Before moving on to describing the technologies and platforms that can be used to create open resources, it will be useful to provide categories of technical expertise needed for prospective tools. This will be useful in helping OER authors determine which tools would be most appropriate for their level of technology expertise. The following are the three categories of skills required:

1. **Low-tech Skills**: The simplest way to create educational resources is by using familiar word processing tools such as Microsoft Word, Google Docs, or Libre Office. This software includes most of the features needed for standard content, and the file can be easily exported as a PDF or printed.
2. **Medium-tech Skills**: Another common way to create or edit educational resources is to create a website or hosted resource. This could be in the form of a blog, a static website, or a wiki. WordPress can be a great tool for these sorts of medium-tech projects.
3. **High-tech Skills**: There are a number of platforms that provide professional tools for authoring content, and some are very easy to use. A common tool used by OER projects is Pressbooks (in which this text is published), a publishing software that makes it easy to produce interactive e-books and other text-based content

These levels will be used to categorise all the tools that will be discussed in the remaining chapters of this guide. Table G 1.1 in [Appendix G](#) summarises the technology skills required for each of the tools discussed in this guide.

To summarise this chapter, it seems fitting to end with the following OER Concept Map.

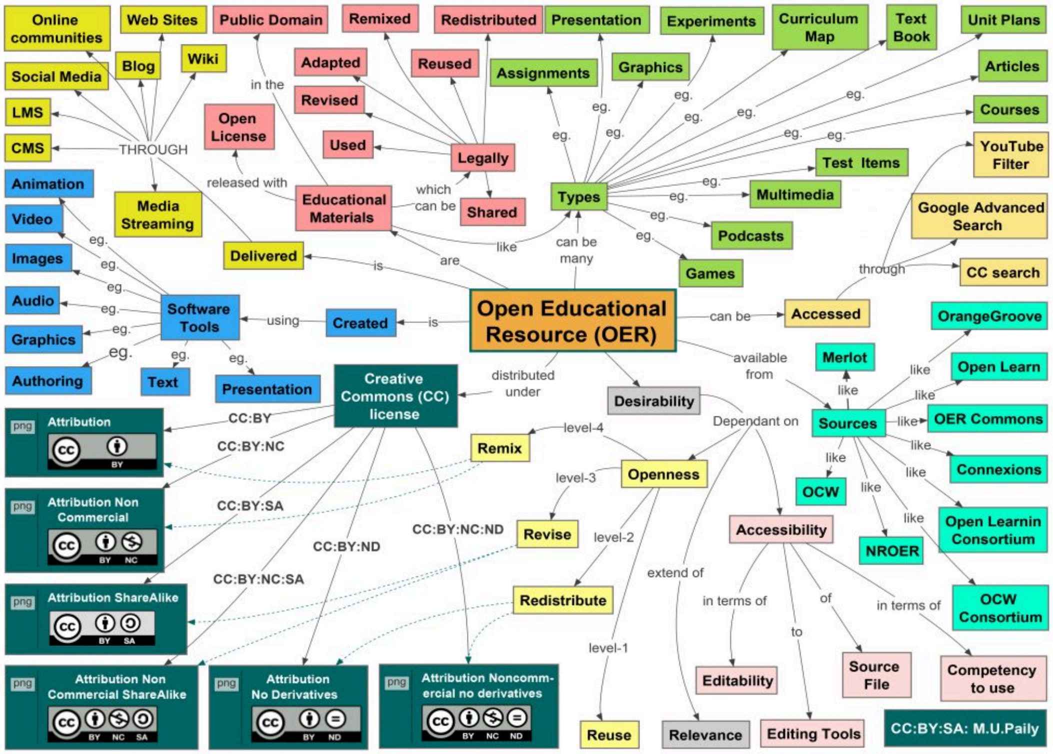


Figure 1.3 OER Concept Map. Source: [OER Concept Map](#) by M.U. Pailey. This file is licensed under the [Creative Commons Attribution-Share Alike 4.0 International](#) license.

For the remainder of this resource, the focus will be on technology tools that can be used to create open educational resources. Both commercial and open platforms will be briefly described along with key features. A useful crowd-sourced tool was created by Abbey Elder in 2018 under a Creative Commons Attribution 4.0 International license. This tool has a list of additional platforms particularly with a focus on those useful for assessment purposes.

References

1. The Four Freedoms. 23 January 2014. "I [Matt Mullenweg] originally thought Stallman started counting with zero instead of one because he's a geek. He is, but that wasn't the reason. Freedoms one, two and three came first, but later he wanted to add something to supersede all of them. So: Freedom zero. The geekness is a happy accident."
2. Free Software Foundation (2018-07-21). "What is free software? Gnu Project – Free Software Foundation (Footnote)." "The reason they are numbered 0, 1, 2 and 3 is historical. Around 1990 there were three freedoms, numbered 1, 2, and 3. Then we realized that the freedom to run the program needed to be mentioned explicitly. It was clearly more basic than the other three, so it properly should precede them. Rather than renumber the others, we made it freedom 0."
3. Hilton, John L. III; Johnson, Aaron; Stein, Jared; and Wiley, David. (2010). The Four R's of Openness and ALMS Analysis: Frameworks for Open Educational Resources. Faculty Publications. 822. https://scholarsarchive.byu.edu/facpub/822/.
4. Menke, William (2018). UH OER Training, https://pressbooks.oer.hawaii.edu/oertraining2018/. Licensed under Creative Commons Attribution 4.0 International License, except where otherwise noted.
5. OpenLearn Create, About Creative Commons Licenses.
6. Open Source Initiative, The Open Source Definition, https://opensource.org/osd. Licensed under a Creative Commons Attribution 4.0 International License.
7. Stallman, Richard. "FLOSS and FOSS." www.gnu.org. Archived from the original on 2018-09-16. Retrieved 2022-01-07.
8. Stallman, Richard. "The Free Software Definition." Free Software Foundation. Retrieved 2022-01-07.
9. Stallman, Richard (February 1986). "Gnu's Bulletin, Volume 1 Number 1." Gnu.org P.8. Retrieved on 2022-01-07.
10. Vetter, G. (2009). "Commercial Free and Open Source Software: Knowledge Production,Hybrid Appropriability, and Patents." *Fordham Law Review*. **77**(5):2087-2141.
11. Wheeler, David A. (May 8, 2014). "Why Open Source Software/Free Software (OSS/FS, or FOSS)? Look at the numbers!" dwheeler.com. Retrieved 2022-01-07.

2. Criteria for the Selection of Digital Technology for use in Education

ISAAC MULOLANI

Learning Objectives

By the end of this chapter, the reader will be able to:

1. Describe the UN Sustainable Development Goals related to Education
2. List some frameworks for digital technology use.
3. Describe potential educational benefits of digital technologies

What is Sustainable Development?

We learned that sustainability is the process of living within the limits of available physical, natural, and social resources in ways that allow all living things, not only humans to thrive well into the future.

Sustainable development is a process that creates growth and progress through the addition of physical, economic, environmental, and social components to improve quality of life without damaging the resources of the environment. Simply put, sustainable development is a way for people to use resources without the resources running out[3].

The concept of sustainable development arrived in 1987 by the Brundtland Commission "Our Common Future", the document that defined sustainable development as an approach designed to meet the needs of the present [generation] without compromising the ability of future generations to meet their own needs. This definition incorporated the understanding that economic growth is required to provide societies with the necessities of life such as clean water and food, while acknowledging the dilemma of environmental degradation that often coincides with economic development.

Recognizing some of the key challenges with the implementation of sustainable development and the quest for achieving a balance between the environment and economies, the role of people and societies were formally added into the equation for sustainable development in 2005 at the UN World Summit on Social Development. The three pillars of sustainability became widely known and currently used today:

(Click on the "?" icons below for more information):

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=922#h5p-26

This updated model for sustainable development recognizes that in order to meet the needs of current and future generations you have to consider the three pillars or the 3P's (people, planet, prosperity), and they **all** need to be working together at the same time. The key being all at the same time, or simultaneously.

Integrating the short-term and long-term needs with a focus on future generations, will require social development, environmental protection, and economic prosperity working in unison. Being able to incorporate sustainability into your day to day activities, this is what will create change.

The United Nations and the Path to the Sustainable Development Goals (SDGs)

Pathway to the Sustainable Development Goals (SDGs)

In 2015, the 2030 Agenda for Sustainable Development was adopted by 193 United Nations (UN) Member States. The 2030 Agenda is centered on the 17 SDGs which are underpinned by the Millennium Development Goals (MDGs). The MDGs were developed in 2000 to end poverty and hunger, fight inequality and injustice, advance climate change action, create sustainable consumption and production, and promote peace and prosperity for all. One major change between the MDGs versus the SDGs is that for the SDGs, all countries are now involved. The MDGs only applied to developing countries. Another difference is that each country has set their own goals and priorities for achieving the SDGs. International collaboration to advance the SDG Agenda remains a critical component. The 17 SD goals, with their 169 targets, and over 230 indicators work together at the local and international level to help promote a shared global framework to achieve a fair, equitable, and sustainable future for all. Currently, all countries and international organizations are working on the achievement of the UN 2030 Agenda serving as the basis for better economic development that is environmentally low impact, socially just, and economically efficient and fair.

In the previous paragraphs we learned about sustainability, sustainable development, and the sustainable development goals and how currently, all countries are working on the achievement of the UN 2030 Agenda, serving as the basis for better economic development that is environmentally low impact, socially just, and economically efficient and fair.

In this 11-minute video made available from the SDG Academy, Jeffrey Sachs (Director of the Sustainable Development Solutions Network), provides an overview of sustainable development.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=922#oembed-1

Video length: 12:24 minutes

As you will recall in 2015, the 2030 Agenda for Sustainable Development was adopted by 193 United Nations (UN) Member States. The 2030 Agenda is centred on the 17 SDGs which are underpinned by the Millennium Development Goals (MDGs). You will recall the 17 SD goals, with their 169 targets, and over 230 indicators work together at the local and international level to help promote a shared global framework to achieve a fair, equitable, and sustainable future for all. We also learned that each country has set their own goals and priorities for achieving the SDGs, with international collaboration to advance the SDGs as a critical component.

In this 11-minute video made available from the SDG Academy, Jeffrey Sachs provides an overview and history of the SDGs.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=922#oembed-2

Video length: 11:00 minutes

Adoption of Sustainable Development Goals (SDGs)

Currently, 193 countries (known as Member States) are signatories to the United Nations. This means almost every country on the planet has adopted the Sustainable Development Goals (SDGs) as the organizing framework for global cooperation on sustainable development. It also means that 193 countries have agreed to work together for the period 2015 (when the SDGs were adopted) until at least to 2030. This level of global cooperation is unprecedented. Think about your own experiences and how hard it can be to get people to agree on something. Now imagine a whole country, and multiple that by 193 countries.

So why have some many countries committed to the SDGs and the 2030 Agenda?

Collectively, it is understood that we have significant environmental threats, such as global warming and the loss of biodiversity. We have widening inequalities between the rich and the poor. Therefore, the UN Member States adopted the Sustainable Development Goals precisely to help reset the direction of the world economy, from one of widening inequalities and social exclusion and great environmental threats to a trajectory of sustainable development. Meaning a path for the world in which prosperity is shared, in which societies are inclusive, and in which the environment is kept safe because we have changed the ways that our industries and technologies are impinging on the physical earth processes.[1]

The Sustainable Development Goals that were adopted on September 25th, 2015, span a remarkable range of aspirations. View the slide show below for more information.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=922#h5p-27

As you recall, the Sustainable Development Goals are part of an overall agenda, a universal agenda called Transforming the World: the 2030 Agenda for Sustainable Development. As mentioned, it has a 15 year forward framework with the following statement of purpose.

---

[1] SDG Academy. (2021). How to achieve the SDGs course. Adapted from module one, chapter one.

> "This agenda is a plan of action for people, planet and prosperity. People, planet, and prosperity, social inclusion planet meaning environmental sustainability and prosperity meaning a shared, economic benefit across the world. Agenda 2030 also seeks to strengthen universal peace in larger freedom. Eradicating poverty in all its forms and dimensions, including extreme poverty, is the greatest global challenge and an indispensable requirement for sustainable development. All countries and all stakeholders acting in collaborative partnership will implement this plan."[2]

Bold and transformative steps are urgently needed to shift the world onto a sustainable and resilient path. The pledge is that no one will be left behind. Bold statements, bold ambitions. And as the title of Agenda 2030 says, requiring transforming the world. The agenda does not just call for change, it calls for deep and radical change all over the world.

The remainder of this course will focus on the components of the 17 specific goals, including the 169 specific targets each goal has. And because you cannot effectively manage what you do not measure, there are also 230 indicators for the 17 goals.

The SDGS are complex. That is why it involves all parts of government, business, and civil society around the world. And it involves all of us, because successfully implementing the SDGs will have a profound positive effect for all human and non-human well-being.

At the core of the 2030 agenda is improving the quality of life and well-being for today's generation and for all the generations to come.

In this 11-minute video made available from the SDG Academy, Jeffrey Sachs helps us get to know the SDGs.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=922#oembed-3

Video length: 11:53 minutes

These 17 SDGs provide a background for all our efforts in education as well as other areas. In open education, the SDG #4 and SDG #12 are critical for inclusive and equitable education for all.

Digital Technology Frameworks

Exploiting the affordances of a medium

It was noted in the previous section that video technology can be used as a straight replacement for a face-to-face lecture by merely substituting the face-to-face delivery with online delivery. The mode of delivery has changed but not the pedagogy. The full affordances of the medium of video have not been exploited.

---

2. United Nations. (2021). Transforming our world: the 2030 Agenda for Sustainable Development.

On the other hand, using video to show a documentary can bring powerful examples of situations to which can be applied the ideas and concepts covered in an academic course. A documentary thus has the potential to make better use of the affordances of video than recording a lecture because the learning experience from watching a documentary is different from watching a lecture; at the same time, using a documentary video will require a different approach to teaching than using a lecture and will probably have different outcomes. With the video lecture students will focus on comprehension and understanding; with the documentary the students' focus will be on analysing and critiquing the material.

The SAMR model

A good way to assess whether a particular application of media or technology is making full use of the affordances of a medium is to apply the SAMR model developed by Dr. Ruben Puentedura, a technology consultant based in the USA.

Puentedura suggests four 'levels' of technology application in education:

- **substitution**: *a direct tool substitute, with no functional change*, for example, a video recording of a classroom lecture on water quality, made available for downloading by students; students are assessed on the content of the lecture by written exams at the end of the course.
- **augmentation**: *a direct tool substitute, with functional improvement*, for example, the video lecture is embedded in an LMS, and edited into four sections, with online multiple-choice questions at the end of each section for students to answer.
- **modification**: *significant task redesign*, for example, the instructor provides video recordings of water being tested, and asks students to analyse each of the recordings in terms of the principles taught in the course in the form of essay-type questions that are assessed.
- **redefinition**: *creation of new tasks*, *inconceivable without the use of technology,* for example, the instructor provides readings and online guidance through the LMS, and students are asked to record with their mobile phones how they selected samples of water for testing quality, and integrate their findings and analysis in the form of an e-portfolio of their work.

In the first two levels, substitution and augmentation, video is used to enhance the method of teaching but it is only where video is used in the final two stages, modification and redefinition, that teaching is actually transformed. Significantly, Puentedura links the modification and transformation levels to the development of Bloom's higher order '21st century' skills such as analysis, evaluation and creativity (Puentedura, 2014). For a more detailed description of the model and how it works, see the video: Introduction to the SAMR model.

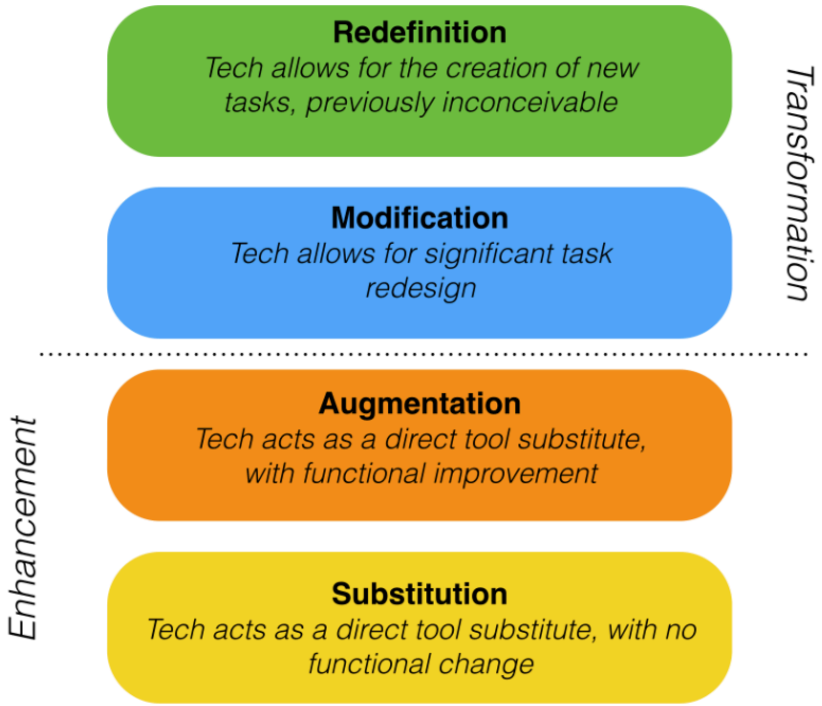

Redefinition
Tech allows for the creation of new tasks, previously inconceivable
Modification
Tech allows for significant task redesign
Augmentation
Tech acts as a direct tool substitute, with functional improvement
Substitution
Tech acts as a direct tool substitute, with no functional change
Transformation
Enhancement

Figure 2.1 The SAMR model Image: Ruben Puentedura

Strengths and limitations of the model

First, there is an absence of research that validates this model. It has a powerful feel of common sense behind it, but it would be good to see it more empirically validated, although there are many examples of its actual use, particularly in teacher education in the k-12 sector (one can find some examples collected by Kelly Walsh here. For a more critical response to the SAMR model, see Linderoth, 2013).

Second, while the model is a useful means of evaluating whether a use of technology merely enhances or radically changes teaching, it doesn't help much with the hard part, and that is imagining the transformative ways in which a technology could be used in the first place. Nevertheless it is a good heuristic device to get one to think about the best way to use technology in teaching.

Third, there will be situations where substitution and augmentation will still be a perfectly justifiable use of technology, for instance for students with disabilities, or to increase accessibility to learning materials.

On balance, it is a very useful model by which an instructor can evaluate a potential or actual use of technology. In particular it focuses on the way students will need to interact with the technology and the ways technology can be used to assist the development of 21st century skills. At the same time, we still need to understand how and why media and technology could be used to transform teaching in the first place. The first step then is to understand better the unique properties of different technologies, which is the subject of the next section.

SECTIONS Model

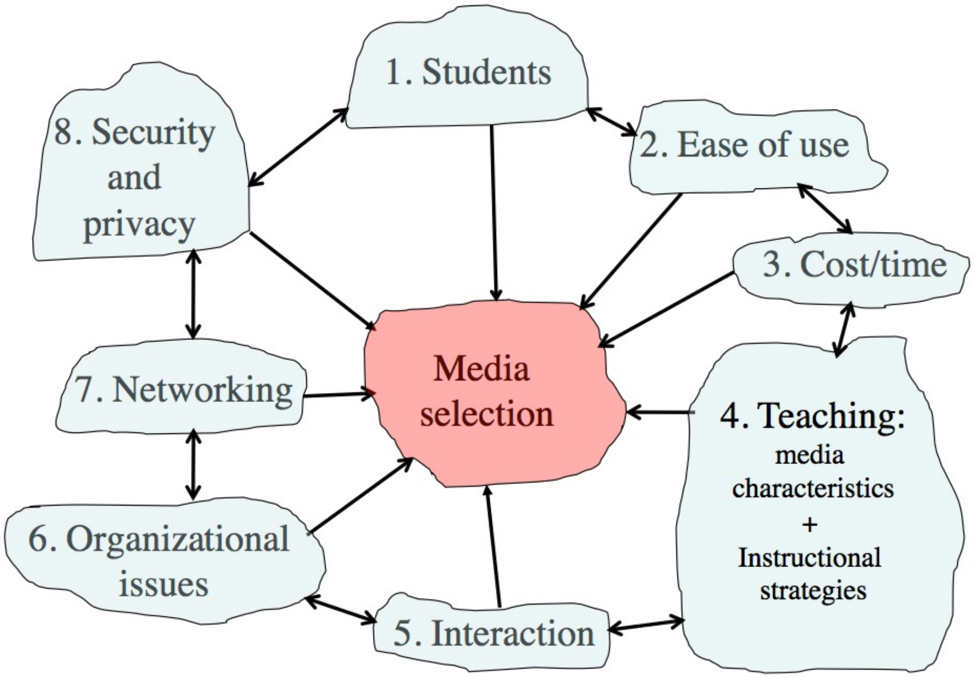


Figure 2.2 The SECTIONS model

What the literature tells us

Given the importance of the topic, there is relatively little literature on how to choose appropriate media or technologies for teaching. There was a flurry of not very helpful publications on this topic in the 1970s and 1980s, but relatively little since (Baytak, undated). Indeed, Koumi (1994) stated that:

> there does not exist a sufficiently practicable theory for selecting media appropriate to given topics, learning tasks and target populations . . . the most common practice is not to use a model at all. In which case, it is no wonder that allocation of media has been controlled more by practical economic and human/political factors than by pedagogic considerations (p. 56).

Mackenzie (2002) comments in a similar vein:

> When I am discussing the current state of technology with teachers around the country, it becomes clear that they feel bound by their access to technology, regardless of their situation. If a teacher has a television-computer setup, then that is what he or she will use in the classroom. On the other hand, if there is an LCD projector hooked up to a teacher demonstration station in a fully equipped lab, he or she will be more apt to use that set up. Teachers have always made the best of whatever they've got at hand, but it's what we have to work with. Teachers make due.

Mackenzie (2002) has suggested building technology selection around Howard Gardner's multiple intelligences theory (Gardner, 1983, 2006), following the following sequence of decisions:

learner → teaching objective → intelligences → media choice.

Mackenzie then allocates different media to support the development of each of Gardner's intelligences. Gardner's theory of multiple intelligences has been widely tested and adopted, and Mackenzie's allocations of media to intelligences make sense intuitively, but of course it is dependent on teachers and instructors applying Gardner's theory to their teaching.

A review of more recent publications on media selection suggests that despite the rapid developments in media and technology over the last 20 years, my ACTIONS model (Bates, 1995) is one of the major models still being applied, although with further amendments and additions (see for instance, Baytak, undated; Lambert and Williams, 1999; Koumi, 2006). Indeed, I myself modified the ACTIONS model, which was developed for distance education, to the SECTIONS model to cover the use of media in campus-based as well as distance education (Bates and Poole, 2003).

Patsula (2002) developed a model called CASCOIME which includes some of the criteria in the Bates models, but also adds additional and valuable criteria such as socio-political suitability, cultural friendliness, and openness/flexibility, to take into account international perspectives. Zaied (2007) conducted an empirical study to test what criteria for media selection were considered important by faculty, IT specialists and students, and identified seven criteria. Four of these matched or were similar to Bates' criteria. The other three were student satisfaction, student self-motivation and professional development, which are more like conditions for success and are not really easy to identify before making a decision.

Koumi (2006) and Mayer (2009) have come closest to to developing models of media selection. Mayer has developed twelve principles of multimedia design based on extensive research, resulting in what Mayer calls a cognitive theory of multimedia learning. (For an excellent application of Mayer's theory, see UBC Wikis.) Koumi (2015) more recently has developed a model for deciding on the best mix and use of video and print to guide the design of xMOOCs.

It is not surprising that there are not many models for media selection. The models developed in the 1970s and 1980s took a very reductionist, behaviourist approach to media selection, often resulting in several pages of decision-trees, which are completely impractical to apply, given the realities of teaching, and yet these models still included no recognition of the unique affordances of different media. More importantly, technology is subject to rapid change, there are competing views on appropriate pedagogical approaches to teaching, and the context of learning varies so much. Finding a practical, manageable model founded on research and experience that can be widely applied has proved to be challenging.

Why we need a model

At the same time, every teacher, instructor, and increasingly learner, needs to make decisions in this area, often on a daily basis. A model for technology selection and application is needed therefore that has the following characteristics:

- it will work in a wide variety of learning contexts;
- it allows decisions to be taken at both a strategic, institution-wide level, and at a tactical, instructional, level;
- it gives equal attention to educational and operational issues;
- it will identify critical differences between different media and technologies, thus enabling an appropriate mix to be chosen for any given context;
- it is easily understood, pragmatic and cost-effective;
- it will accommodate new developments in technology.

For these reasons, we will use the Bates' SECTIONS model, with some modifications to take account of recent developments in technology, research and theory. The SECTIONS model is based on research, has stood the test of time, and has been found to be practical. SECTIONS stands for:

- **S Students:** what is known about the students-or potential students and the appropriateness of the technology for this particular group or range of students?
- **E Ease of use and reliability:** how easy is it for both teachers and students to use? How reliable and well tested is the technology?
- **C Costs:** what is the cost structure of each technology? What is the unit cost per learner?
- **T Teaching and learning:** what kinds of learning are needed? What instructional approaches will best meet these needs? What are the best technologies for supporting this teaching and learning?
- **I Interactivity:** what kind of interaction does this technology enable?
- **O Organizational issues:** What are the organizational requirements and the barriers to be removed before this technology can be used successfully? What changes in organization need to be made?
- **N Novelty:** how new is this technology?
- **S Speed:** how quickly can courses be mounted with this technology? How quickly can materials be changed?

The following video provides a description of the model. One issue to note is the different modalities are useful for incorporating learner preferences that are increasingly found in post-secondary education.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=922#oembed-4

Video length: 7:59 minutes

The interested reader can access the Educause article, A Rubric for Evaluating E-Learning Tools in Higher Education by Lauren Anstey and Gavan Watson. The corresponding Rubric for E-Learning Tool Evaluation can be used to help with technology selection. Educause also has another article, A Rubric for Selecting Active Learning Technologies by Katie Bush, Monica Cormier and Graham Anthony. They have developed a Rubric for Active Learning Technology Evaluation that anyone may make a copy of and use.

Mayer’s 12 Principles of Multimedia Learning

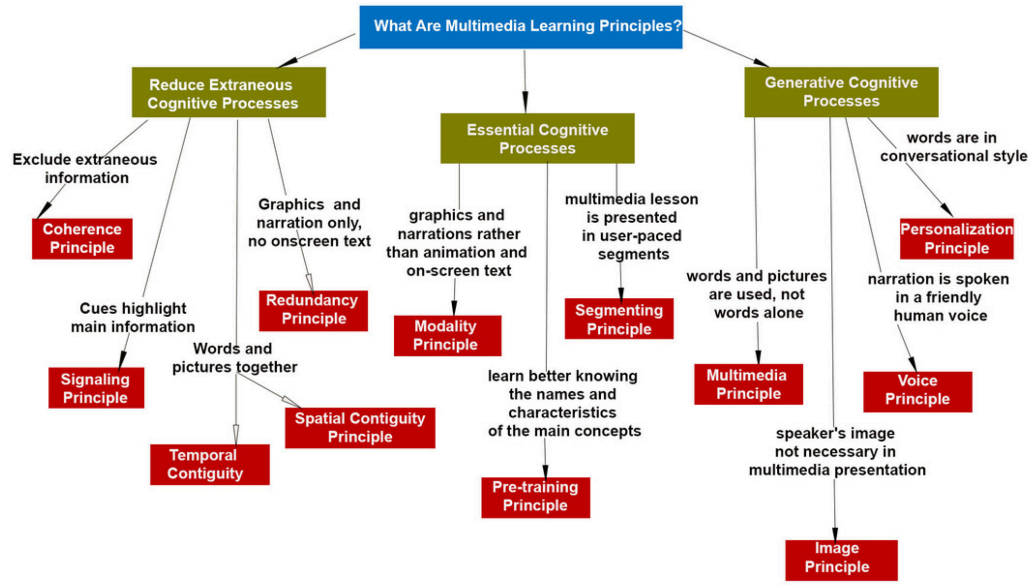
Identifying appropriate uses of media is both an increasingly important requirement of teachers and instructors in a digital age, and a very complex challenge. This is one reason for working closely with instructional designers and media professionals whenever possible. Teachers working with instructional designers will need to decide which media they intend to use on pedagogical as well as operational grounds.

However, once the choice of media has been made, by focusing on design issues we can provide further guidelines for making appropriate use of media. In particular, having gone through the process of identifying possible teaching roles or functions for different media, we can then draw on the work of Mayer (2020) and Koumi (2006, 2015) to ensure that whatever choice or mix of media we have decided on, the design leads to effective teaching.

Mayer’s research focused heavily on cognitive overload in rich, multimedia teaching. From all his research over many years, Mayer identified 12 principles of multimedia design, based on how learners cognitively process multimedia:

1. **Coherence** – *People learn better when extraneous words, pictures and sounds are excluded rather than included*. Basically, keep it simple in media terms.
2. **Signalling** – *People learn better when cues that highlight the organization of the essential material are added*. This replicates earlier findings by Bates and Gallagher (1977). Students need to know what to look for in multimedia materials.
3. **[Avoid] Redundancy** – *People learn better from graphics + narration, than from graphics, narration and on-screen text.*
4. **Spatial contiguity** – *People learn better when corresponding words and pictures are presented near rather than far from each other on the page or screen*
5. **Temporal contiguity** – *People learn better when corresponding words and pictures are presented simultaneously rather than successively.*
6. **Segmenting** – *People learn better when a multimedia lesson is presented in user-paced segments rather than as a continuous lesson*. Thus several ‘YouTube’ length videos are more likely to work better than a 50 minute video.
7. **Pre-training** – *People learn better from a multimedia lesson when they know the names and the characteristics of the main concepts*. This suggests a design feature for flipped classrooms, for instance. It may be better to use a lecture or readings that provide a summary of key concepts and principles before showing more detailed examples or applications of such principles in a video.
8. **Modality** – *People learn better from graphics and narration than from animation and on-screen text*. This reflects the importance of learners being able to combine both hearing and viewing at the same time to reinforce each other in specific ways.
9. **Multimedia** – *People learn better from words and pictures than from words alone*. This also reinforces what I wrote in 1995: *Make all four media available to teachers and learners* (Bates, 1995, p.13).
10. **Personalization** – *People learn better from multimedia lessons when words are in conversational style rather than formal style*. I would go even further than Mayer here. Multimedia can enable learners (particularly distance learners) to relate to the instructor, as suggested by Durbridge’s research (1983, 1984) on audio combined with text. Providing a ‘human voice and face’ to the teaching helps motivate learners, and makes multimedia teaching feel that it is directed solely at the individual learner, if a conversational style is adopted.
11. **Voice** – *People learn better when the narration in multimedia lessons is spoken in a friendly human voice rather than a machine voice.*
12. **[No] image** – *People do not necessarily learn better from a multimedia lesson when the speaker’s image*

is added to the screen.



What Are Multimedia Learning Principles?
Reduce Extraneous Cognitive Processes
Essential Cognitive Processes
Generative Cognitive Processes
Exclude extraneous information
Coherence Principle
Graphics and narration only, no onscreen text
Redundancy Principle
Cues highlight main information
Signaling Principle
Words and pictures together
Temporal Contiguity
Spatial Contiguity Principle
graphics and narrations rather than animation and on-screen text
Modality Principle
multimedia lesson is presented in user-paced segments
Segmenting Principle
learn better knowing the names and characteristics of the main concepts
Pre-training Principle
words and pictures are used, not words alone
Multimedia Principle
words are in conversational style
Personalization Principle
narration is spoken in a friendly human voice
Voice Principle
speaker's image not necessary in multimedia presentation
Image Principle

Figure 2.3 Mayer's 12 Principles of Multimedia Learning Source: Principles of Multimedia Learning by Tuba Öney 2017

In re-reading Mayer's work, one is struck by the similarities in findings, using different research methods, different multimedia technologies, and different contexts, to the research from the Audio-Visual Media Research Group at the British Open University in the 1970s and 1980s (Bates, 1984).

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=922#oembed-5

Video length: 12:53 minutes

Usually the technology frameworks are used when discussing the most appropriate use of technology in education. They help answer questions on how best to integrate digital technology into teaching and learning. This specific application involves the selection of technologies for creating OER. An increasingly common practice is for creators of OER to develop more engaging digital content using appropriate technology tools. Newer OER content increasingly includes images, animations, videos, and interactive H5P elements.

Mayer's 12 principles of multimedia learning provide a more deliberate way in which to design learning that leverages the use of multimedia elements. Many of the OER are increasingly digital first and also designed to be content delivery mechanisms. Creators of OER should consider these principles when designing open content. Consequently, many OER development teams include instructional designers and graphic artists to help with these design requirements.

The 8A Framework Model for Digital Technologies

A final point to discuss is the potential affordances digital technologies offer when used appropriately. The following information is a summary of content from chapter 4 of the resource Rethinking Pedagogy: Exploring the Potential of Digital Technology in Achieving Quality Education by Lorenzo et al. and licensed under a Creative Commons ShareAlike IGO 3.0 license. Figure 2.4 provides a visual representation of each of these affordances or attributes.

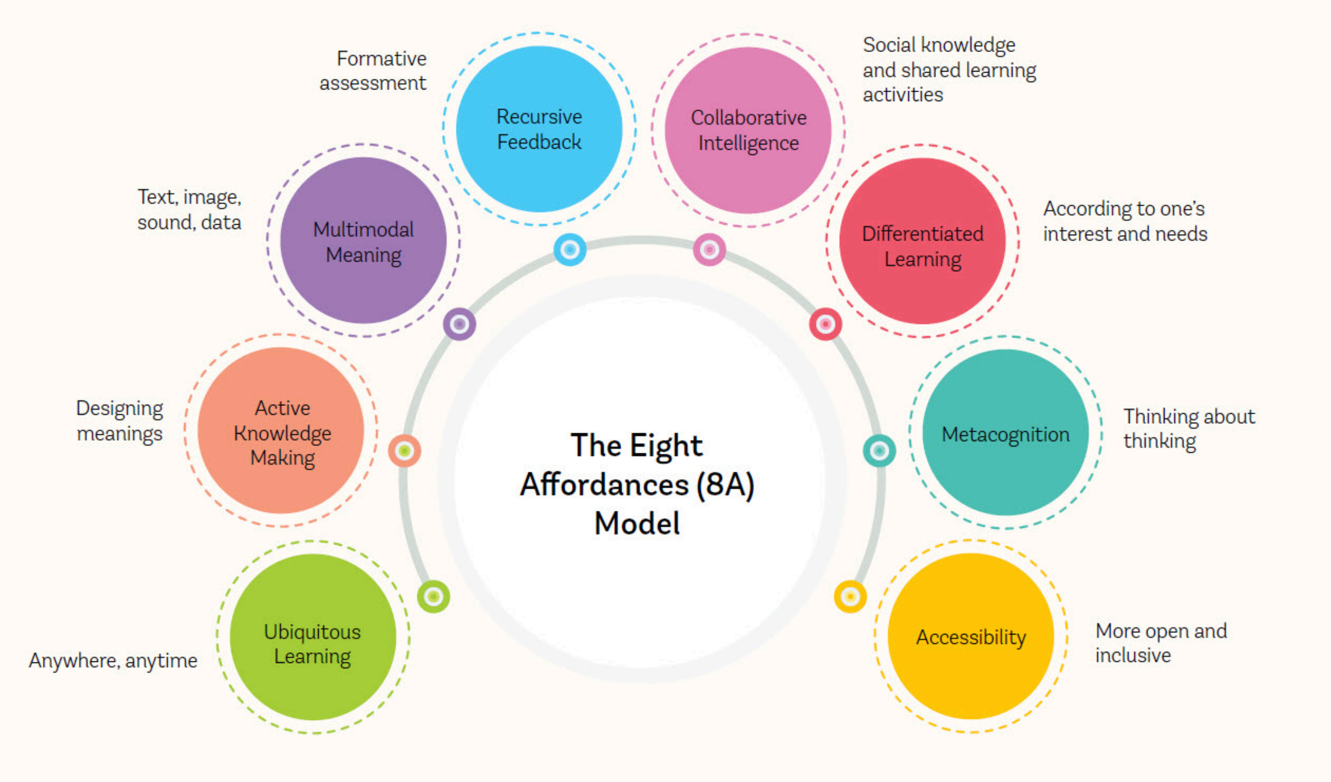


Figure 2.4 Eight Affordances Model graphical depiction

Each of these eight attributes will be briefly described to provide insight on how they relate to our current context.

Ubiquitous Learning

The idea of ubiquitous learning is an extension of an everyday reality that is today called ubiquitous computing or the ready availability of interconnected computing devices, many of which are portable or, when fixed, offer cloud access to shared spaces. This is critical today because it means learners can access learning spaces and content anytime, anywhere. This affordance allows formal learning to break out of the spatial confinement of classrooms and the temporal confinement of timetables. Significantly, ubiquitous computing offers new opportunities for ubiquitous learning. It extends the action space for learning, blurring the traditional boundaries of space and time.

Ubiquitous digital tools also defy geographical constraints and offer opportunities to collaborate and create partnerships with learners and educators all over the world. All these possibilities can be achieved with much lower costs than the regular channels of physical communications. Specifically, this affordance also offers opportunities for individualized learning where curious learners can pursue their various learning pathways at their convenience.

Active Knowledge Making

One of the key challenges for an education that fosters skills to contribute to a more sustainable world is for students not only to acquire subject-oriented knowledge, but also to recognize problems and challenges that surround us and explore ways to solve them. Active knowledge making can support exactly these approaches: students are encouraged to discover things, understand challenges and find solutions by actively making things and constructing knowledge.

Interactive digital technologies provide tools that are *objects-to-think-with* (Papert 1980, p.11). New possibilities emerged in experiential learning where learners engage in meaning making and knowledge construction with the real or virtual space of "microworlds" (Papert 1980, p.117). "Microworlds" and tools are all-inclusive environments that enable learners to be totally immersed in experimenting within the same closed setting. They can be characterised by an action space with a set of controls and constraints to enable learners to get actively engaged with the agents and environments. This possibility is remarkably different to the transmissive pedagogies of rote learning and memorization or even merely theoretical information gathering as learners are engaged in a discovery-oriented and adventurous learning process.

Multimodal Meaning

Much of our everyday representational experience is fundamentally multimodal. However, traditional methods and teaching-learning toolbox consisting of books, pens and vocal chords, are limited by what and how much they can represent. By comparison, digital technologies possess multiple and powerful ways of representation with a combination of text, image, audio, video, simulation, interactive, immersive environment, virtual and augmented reality and so forth. Specifically, concepts and information content can be represented or perhaps enacted multifariously in different modes and forms and thereby allowing learners to understand things in many different modes – textual, oral, visual, spatial and embodied (Kalantzis 2016).

Research suggests that multimodal resources can cater more effectively to diverse learners (Cope and Kalantzis 2017). With multimodal representation learners get empowered to not only choose preferred media, have concepts reinforced but also make multiple meanings including meanings unthinkable in traditional forms of representation. Importantly, the multimodal meaning making affordance of digital learning resources contributes to SDG 4. Together with the Accessibility affordance, the usage of different environments and media types can support a wide range of learners.

Recursive Feedback

While there may be varied positions on the most desirable type of and time for feedback, there is fair level of consensus among educators that feedback is essential for learning. Widely prevalent regimes of summative assessment offer retrospective and judgemental perspective of learning, which mostly do not contribute to learning in prospective and constructive ways. However, technology-enabled learning environments and digital educational resources offer various opportunities for both feedback and feed-forward, supporting learners during their learning journeys.

Recursive feedback also contributes to the notion that every individual's learning can be supported by an individual learning path. With recursive feedback, instructors can help students to find their own way. When it comes to peer reviewing and peer-to-peer learning students can experience collaborative learning while they help each other. In this way, they learn to communicate in a cooperative way and so acquire knowledge through a process of participation in a knowledge community.

Collaborative Intelligence

As reflected in the metaphor of learning as knowledge acquisition, education has tended to focus on learning as individual memory. Not to discount memory, knowledge and learning are social, as captured in participation and knowledge creation metaphors. Knowledge has social provenance which requires acknowledgement, and social learning can be a powerful, indeed for today's society, an essential supplement to individual memory work – working with peers, offering and receiving feedback, and undertaking collaborative learning activities.

Opportunities to collaboratively construct unique pathways to accomplish learning goals are important for both learners and instructors. The rise of web-based social networking has enabled learners and instructors to organize learning collaborations at all levels, purposes, and group sizes. Learners are no longer restricted to forming collaborations with just their peers in a course. They can organize inter-institutional collaborations, discover content, and participate in other learning communities to augment their learning.

Differentiated Learning

Differentiated learning refers to tailoring a teaching resource to a learner's needs and interests, with recognition that not everyone learns in the same way. Differentiated learning pays attention to diversity in the approaches and levels of students' needs. It offers every student the opportunity to find his or her own way into better understanding the world around. Sometimes, differentiated learning comes in the form of 'personalized learning,' however at times this can mean learning where individuals are isolated from each other. Another more collaborative version of differentiated learning we would call 'productive diversity,' leveraging the different interests and perspectives of learners as they work with each other – in 'jigsaw' learning, peer review, and discussion boards, for instance, where differentiation opens expression of learner diversity and deploys this as a resource for learning.

Digital learning platforms make differentiated instruction more feasible. Adaptive and personalized learning enables students to work at their own pace. While self-organization helps students to organize themselves also in terms of getting into action and organize not only their own learning but also learn to organize projects.

Metacognition

Metacognition or thinking about one's processes of thinking is a means to think more deeply, and at a higher level of abstraction (Livingston 2003). It produces efficiencies in thinking and learning, as conceptualization broadens the scope of ideas in application, transfer and understanding. Thinking about thinking is a valuable activity for online learners which leads to active learning (Huffaker and Calvert 2003). Thinking is also more efficient and effective when accompanied by the process of metacognition or monitoring and reflecting upon one's own thinking. An integral part of this process is weaving between the new knowledge and self-reflection about one's own knowledge background and thinking processes (Brown 1987). This requires interpretation of the social and cultural context of an expression of meaning or a piece of knowledge.

Metacognition in all its different shapes is core to understanding the complex issues of sustainable development and developing ideas and actions to meet those challenges. So, metacognition means to learn critical thinking and understanding complex systems. This is necessary in order to find solutions to complex problems. Understanding complex systems has to be the first step in order to make change in the material and social conditions of life.

Accessibility

Accessibility refers to the availability of digital educational resources to all, irrespective of geographic location, language(s), disability and other demographic and socio-economic variables. Educational technology as a cognitively dis-embodied thing posits both limitations and opportunities. On the one hand it may extend action space for learning, while on the other hand it may impose new barriers. These traits are fundamentally shaped by the design of technology. Therefore, it is imperative to ensure that digital educational resources are designed to extend the action space for learning – the key to doing so is make them accessible. In terms of SDG 4 to ensure "inclusive and equitable quality education and promote lifelong opportunities for all", the affordance of accessibility is a matter of promoting openness and inclusion.

In this context, accessibility of digital educational resources has three primary dimension. The first dimension is to promote the availability of digital educational resources. This is achieved in two ways. The first is for education systems to ensure that there are no barriers to access based on the cost of published materials and digital resources. The second is free and open source digital resources. However, the challenge here is to offer proper remuneration for creators, traditionally taking the forms of author royalties and employment in the educational publishing industry.

The second dimension is interoperability, which requires that the system is open to different kinds of expression, and integration with other (external) tools and systems. Complying to interoperability standards allows aggregation of efforts, integration of systems, and opening of learning opportunities.

Third, it is important to follow the Universal Design for Learning (UDL) principles to ensure that learning resources are accessible to all, including people with disabilities. It is also important to ensure accessibility of resources across a range of devices, in online as well as offline mode. Provision for translations and internationalization of interfaces is another key consideration.

For an extensive discussion of this issue, the interested reader should reference the resource, Rethinking Pedagogy: Exploring the Potential of Digital Technology in Achieving Quality Education, developed by the Mahatma Ghandhi Institute of Education for Peace and Sustainable Development[3]. Appendix I provides indicators for each of these Eight Affordances.

References

Anstey, L. and Watson, G. (2018). A Rubric for Evaluating E-Learning Tools in Higher Education. Educause Review.

Baytak, A.(undated) Media selection and design: a case in distance education *Academia.edu*

Bates, A. (1984) *Broadcasting in Education: An Evaluation* London: Constables

Bates, A. (1995) *Teaching, Open Learning and Distance Education* London/New York: Routledge

Bates, A. and Gallagher, M. (1977) *Improving the Effectiveness of Open University Television Case-Studies and*

---

[3] Rethinking Pedagogy: Exploring the Potential of Digital Technology in Achieving Quality Education. Aleman de la Garza, L., Antal, P. t., Beaune, A. I., Bruillard, E. R., Burke, D., ... Tsinakos, A. This work is under an Attribution-ShareAlike 3.0 IGO license.

Documentaries Milton Keynes: The Open University, I.E.T. Papers on Broadcasting, No. 77 (out of print – copies available from tony.bates@ubc.ca)

Bates, A. and Poole, G. (2003) *Effective Teaching with Technology in Higher Education* San Francisco: Jossey-Bass/John Wiley and Son

Bush, K. Cormier, M. and Anthony. G. (2022). A Rubric for Selecting Active Learning Technologies. Educause Review.

Cope, B. and Kalantzis, M. (2016). Big data comes to school: implications for learning, assessment and research. *AERA Open*, 2(2), pp. 1-9.

Durbridge, N. (1983) *Design implications of audio and video cassettes* Milton Keynes: Open University Institute of Educational Technology (out of print)

Durbridge, N. (1984) Audio-cassettes, in Bates, A. (ed.) *The Role of Technology in Distance Education* London/New York: Croom Hill/St Martin's Press

Gardner, H. (1983) *Frames of Mind: The Theory of Multiple Intelligences* New York: Basic Books

Gardner, H. (2006) *Multiple Intelligences: New Horizons and Theory in Practice* New York: Basic Books

Huffaker, D. A. and Calvert, S. L. (2003). The new science of learning: active learning, metacognition, and transfer of knowledge in e-learning applications. *Journal of Educational Computing Research*, 29(3), pp. 325-334.

Kalantzis, M. Cope, B., Chan, E. and Dalley-Trim, L. (2016). *Literacies*, 2nd Edition, Cambridge UK: Cambridge University Press.

Koumi, J. (1994). Media comparisons and deployment: A practitioner's view *British Journal of Educational Technology*, Vol. 25, No. 1

Koumi, J. (2006). *Designing video and multimedia for open and flexible learning* London: Routledge

Koumi, J. (2015) Learning outcomes afforded by self-assessed, segmented video-print combinations *Cogent Education*, Vol. 2, No.1

Lambert, S. and Williams R. (1999) *A model for selecting educational technologies to improve student learning* Melbourne, Australia: HERDSA Annual International Conference, July

Linderoth, J. (2013) Open letter to Dr. Ruben Puentedura *Spelvetenskapliga betraktelser*, 17 October

Livingston, J.A. (2003). Metacognition: an overview. *Psychology*. 13, pp. 259-266.

Mackenzie, W. (2002) *Multiple Intelligences and Instructional Technology: A Manual for Every Mind* Eugene, Oregon: ISTE

Mayer, R. E. (2009). *Multimedia learning* (2nd ed). New York: Cambridge University Press

Papert, S. (1980). Mindstorms: children, computers and powerful ideas. New York: Basic Books.

Patsula, P. (2002) Practical guidelines for selecting media: An international perspective *The Useableword Monitor*, February 1

Puentedura, R. (2014) SAMR and Bloom's Taxonomy: Assembling the Puzzle *common sense education*, September 24

UBC Wikis (2014)*Documentation: Design Principles for Multimedia* Vancouver BC: University of British Columbia

Zaied, A. (2007) A Framework for Evaluating and Selecting Learning Technologies *The International Arab Journal of Information Technology*, Vol. 4, No. 2

3. Commercial Word Processing Tools

ISAAC MULOLANI

Learning Outcomes

By the end of this chapter, the reader will be able to:

1. Describe the features of word processing tools useful for creating OER
2. Identify various commercial word processing tools useful for creating OER
3. Compare different tools to choose the ones most suitable for given projects

Brief History of Word Processing Software

It would be helpful to trace the emergence of word processing tools just to get a sense of where they came from. Wikipedia[1] provides a detailed history of Microsoft Word.

Word processors did not develop *out* of computer technology. Rather, they evolved from mechanical machines and only later did they merge with the computer field.[5] The history of word processing is the story of the gradual automation of the physical aspects of writing and editing, and then to the refinement of the technology to make it available to corporations and Individuals.

The term *word processing* appeared in American offices in early 1970s centered on the idea of streamlining the work to typists, but the meaning soon shifted toward the automation of the whole editing cycle. At first, the designers of word processing systems combined existing technologies with emerging ones to develop stand-alone equipment, creating a new business distinct from the emerging world of the personal computer. The concept of word processing arose from the more general *data* processing, which since the 1950s had been the application of computers to business administration.[6]

Through history, there have been three types of word processors: mechanical, electronic and software.

The final step in word processing came with the advent of the personal computer in the late 1970s and 1980s and with the subsequent creation of word processing software. Word processing software that would create much more complex and capable output was developed and prices began to fall, making them more accessible to the public. By the late 1970s, computerized word processors were still primarily used by employees composing documents for large and mid-sized businesses (e.g., law firms and newspapers). Within a few years, the falling prices of PCs made word processing available for the first time to all writers in the convenience of their homes.

---

1. See Wikipedia

The first version of Microsoft Word was developed by Charles Simonyi and Richard Brodie, former Xerox programmers hired by Bill Gates and Paul Allen in 1981. Both programmers worked on Xerox Bravo, the first WYSIWYG (What You See Is What You Get) word processor. The first Word version, Word 1.0, was released in October 1983 for Xenix and MS-DOS. This was followed by the release of four very similar versions that were not successful.

The first Windows version was released in 1989, with a slightly improved interface. When Windows 3.0 was released in 1990, Word became a huge commercial success. Word for Windows 1.0 was followed by Word 2.0 in 1991 and Word 6.0 in 1993. Then it was renamed to Word 95 and Word 97, Word 2000 and Word for Office XP (to follow Windows commercial names). With the release of Word 2003, the numbering was again year-based. Since then, Windows versions include Word 2007, Word 2010, Word 2013, Word 2016, and most recently, Word for Office 365.

What is the difference between Microsoft 365 and Office 2021?

Microsoft 365 is a subscription service that many people have been using in recent years. With this service comes a continually updated version of the software. Plans exist for personal, home as well as for small and mid-sized businesses, large enterprises, schools, and non-profits. The home and personal plans include the familiar desktop apps like Word, Excel, PowerPoint in addition to extra cloud storage that enables online collaboration. The subscription ensures that the user has all the latest features, fixes, security updates along with perennial tech support. The Microsoft 365 Family plan also lets you share your subscription with your family for up to six people, and use your apps on multiple PCs, Macs, tablets, and phones[2].

On the other hand, Office 2021 is a one time purchase of the office apps for a single computer. Unfortunately, there are no upgrades options available for this purchase. This means that to upgrade to the next major release, you will have to buy it at the full price.

---

2. See Microsoft

	One-time purchase (Example: Office Home & Student 2021)	**Microsoft 365 subscription (Example: Microsoft 365 Family)**
Cost	Pay a single, one-time cost.	Pay a small monthly fee, or save by paying for a full year.
Office applications	Get Office apps like Excel, Word, and PowerPoint.	Get the latest version of Office apps like Excel, Word, PowerPoint and Outlook. You'll always get the latest features, new tools, security updates, and bug fixes. PC users also get Access and Publisher.
Feature updates	Security updates are included, but you won't get any new features. Upgrades to major releases aren't included.	Your version of Office will always be improving. You'll get all the latest features and updates, as well as security updates and bug fixes.
Install Office on more than one computer (Mac or PC)	One-time purchases can be installed once on either a PC or Mac.	With Microsoft 365 Family, you can install Microsoft 365 on all your devices and sign in to five at the same time—that means you can use Office no matter where you are or what device you're on. This includes PCs, Macs, tablets, and phones. You can also share you subscription with up to five other people.
Advanced features on tablets and phones	Install the mobile apps for free and get basic editing features on tablets or phones under 10.1 inches.	Install the mobile apps for free and get extra features when you sign in to Office apps on your device.
Extra online storage	Not included.	Securely store your files in the cloud and access them from anywhere. Get 1 TB of OneDrive cloud storage per user, for up to 6 users including yourself. (Microsoft 365 Family).
Technical support is included	Initial technical support is included for installing only.	Contact us throughout your subscription at no extra cost for help with technical issues, or for subscription and billing support.

Table 3.1 Table comparing Microsoft 365 subscription with Office 2021 (one-time purchase)[3]

Features of word Processing Tools

At the middle of the last century, tools such as typewriters were in common use in many companies. People in the workplace relied on the typing pool to create the necessary communications with others within and outside the office. These administrative tasks were performed by individuals working in this dedicated secretarial pool. One of the obvious consequences of this was the time and effort it took to communicate through this secretarial position. One of the additional tasks required by this workflow is correcting errors made in typing these communications[4].

With the advent of computers and the development of the word processor, professionals were then able to create their own documents in the workplace. Not only did the word processor eliminate the need for the secretarial pool, it also allowed for standardization of the suite of tools facilitating sharing of media much easier[5].

Microsoft developed the suite of tools we today know to be Microsoft Office. There are other word processors out there, but Microsoft Office is one of the most commonly used word processing tools[6]. The main tools that make up Microsoft Office are[7]:

---

3. See Microsoft
4. See Cook, Simon (2021). Advantages and Disadvantages of Microsoft Word. TurboFuture.
5. See Cook, Simon (2021). Advantages and Disadvantages of Microsoft Word. TurboFuture
6. See Maxwell, C. (2021). Ascendency of Word Processing and Microsoft Word. Towering Skills.
7. See Cook, Simon (2021). Advantages and Disadvantages of Microsoft Word. TurboFuture.

- **Excel**: a powerful spreadsheet tool.
- **PowerPoint**: a tool for creating presentations.
- **OneNote**: a program allowing one to gather notes and information in one place.
- **Outlook**: a personal information manager that allows one to manage e-mail accounts, calendars and address books.
- **Publisher**: for designing and publishing document such as brochures and newsletters.
- **Access**: a database management system.
- **Word**: for creating documents with embedded graphics and media for business and educational communication.

While Microsoft Office is composed of these tools, the focus of this guide will be Microsoft Word and tools that can be considered alternatives. There are many users of Microsoft Office (Office 365) across the globe[8]. Go Skills is an online learning platform that helps people learn business skills that help fulfill their personal and professional goals[9]. Go skills estimates that that 1.2 billion people across 140 countries and 107 languages that use Microsoft Office.

Many of us encounter the use of Microsoft Office at home, at school and at work. Across most of North America, the use of Microsoft Office is common place at all levels of education. A Statistica survey indicates that there are over a million companies worldwide that use Office 365[10]. Of this number, a significant majority (over 879,851 companies) are in the United States, 211, 770 in the United Kingdom and 97,139 are in Canada.

With this in mind, the focus will be on the features in Word useful for the creation of OER. These will be contrasted/compared with the features available in the alternatives examined.

Microsoft Word

In this section, the key features of interest for those wishing to create open content with Word are briefly listed. The following is a list of key menu items in Microsoft Word.

1. **Home** – This contains a number of basic document elements. Elements such as headings, finding, replacing and selecting text, office dictation and checking for spelling, writing and grammar suggestions. Additional features include options like font colour, font size, font style, alignment, bullets, line spacing, etc.
2. **Insert** – Some elements useful in documents include tables, shapes, images, charts, graphs, header, footer, page number, etc. Also from this menu one can take a screenshot, find and insert online videos, find and quote information from Wikipedia, create links, insert a bookmark or a cross reference and more.
3. **Design** – Under this menu item are themes for designing documents, templates for titles, colors, fonts, paragraph spacing, watermarks, page colors and page borders.
4. **Page Layout** – Under this are page margins, paper size and orientation, columns, page breaks, hyphenation, indentation and paragraph spacing.
5. **References** – This contains options such as table of contents, footnotes, endnotes, search tabs for quotes, citations, citations, bibliography, bibliography style selector, source manager, insert captions, table of figures, cross references, insert index, mark citations and insert a table of authorities.
6. **Review** – This tab contains options like spell checker, thesaurus, word count, read aloud (speech), accessibility check, language translator, set language, add document comments, track changes, options to

---

[8] See the Go Skills website
[9] See the Go Skills about us page
[10] See Statistica

show and hide document markup, options to accept or reject document changes, compare documents, option to restrict editing, hide all ink in the document and a resume assistant.

Many of these features are familiar to most users of word processing programs. Clearly, these features would be extremely useful to those creating open educational resources such as manuals, course notes and textbooks.

One of the major challenges for many users is the cost for using Office 365 particularly in developing parts of the world. On top of this, to add images to a Word document requires external drawing programs. Word comes with a limited number of graphical objects that can be used in documents. There are several options: commercial drawing programs and open-source programs. Similarly, there are free image repositories and many commercial repositories. The point here is that in order to create a quality open resource in Word with images and photos generally requires more than one tool.

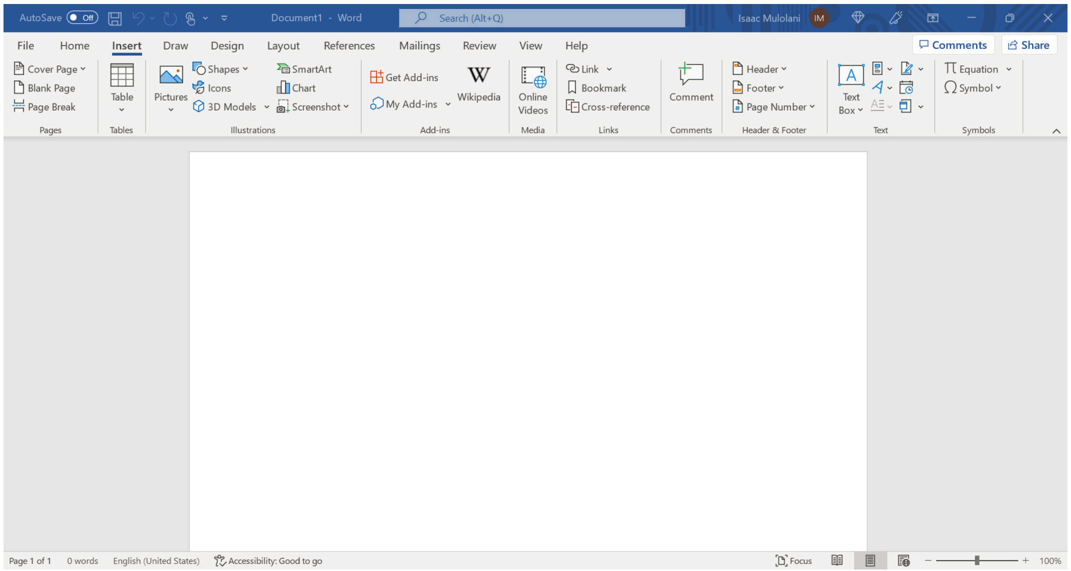


Figure 3.1 a Word open in the Insert tab

From Figure 3.1 a, the different options under the insert tab can be clearly seen. The many options available under this tab can be used to create open resources.

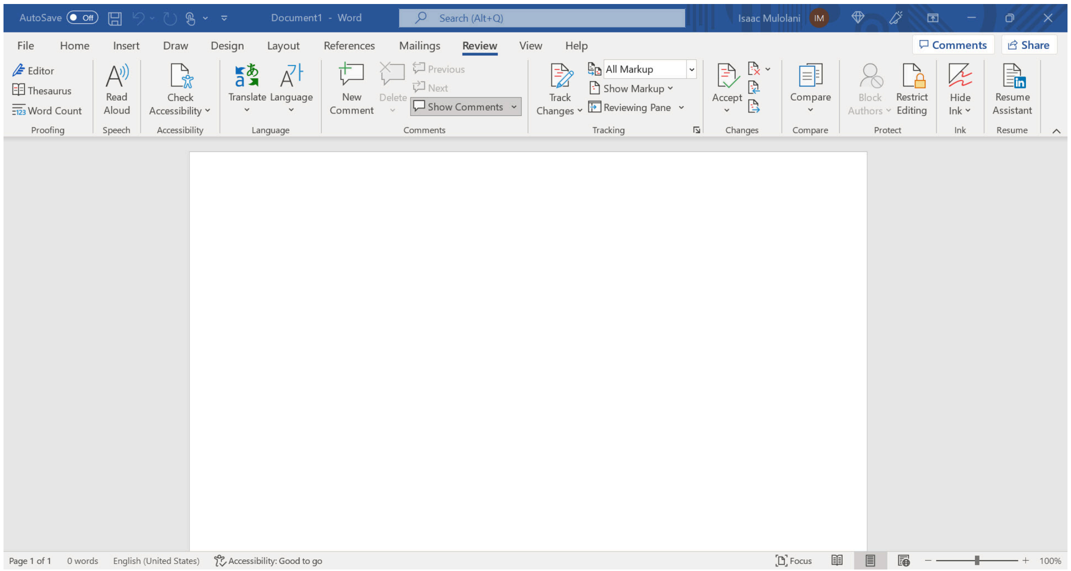


Figure 3.1 b Word open in the Review tab

From Figure 3.1 b, the options available under the Review tab include the common ways to track changes in documents, check accessibility and make comments. These allow collaboration among teams.

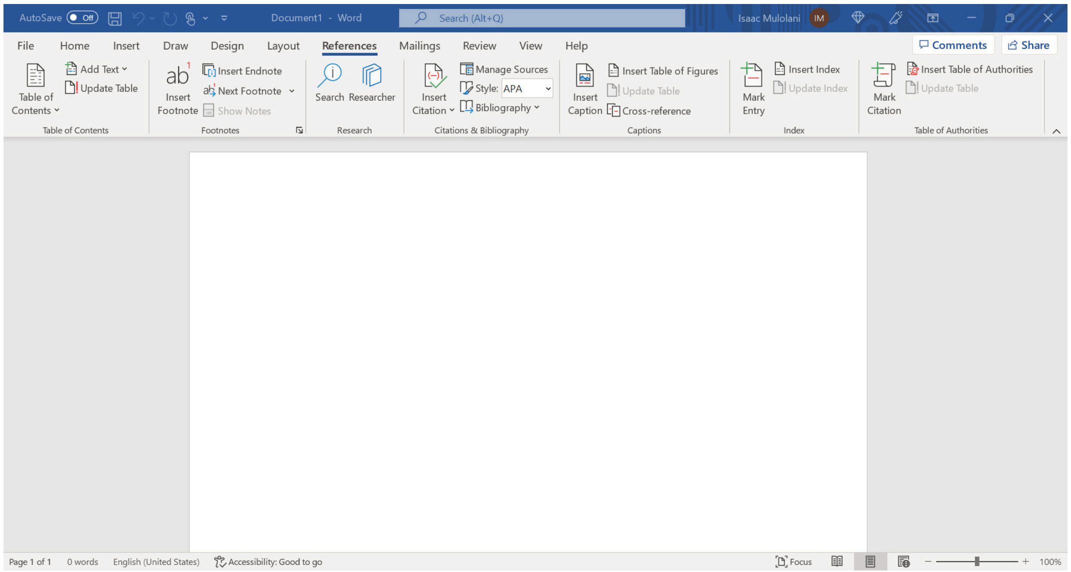


Figure 3.1 c Word open in the References tab

In Figure 3.1 c, we can see the other useful document elements e.g., table of contents, footnotes, endnotes, managing sources, inserting a table of figures, inserting an index etc. These are book features familiar to all in post secondary education.

SoftMaker Office

SoftMaker was founded in 1987 and has been developing office software in the form of:

1. TextMaker – word processing
2. PlanMaker – spreadsheet
3. SoftMaker Presentations – presentation graphics
4. DataMaker – database software

SoftMaker's flagship product, SoftMaker Office, is available for Windows, Mac, Linux and Google Android. There are a number of features that put SoftMaker Office in a class of its own. These include its intuitive user interface, ease of use, seamless compatibility with Microsoft Office file formats and the sheer speed of the applications. On top of this, the pricing of the software make it a very attractive alternative.

SoftMaker has a second productivity tool, FlexiPDF, which is an editor that makes it easy to modify PDF files. FlexiPDF provides the full range of PDF editing options without breaking the bank. This makes it a much cheaper alternative to the pricey Adobe Acrobat. In addition, SoftMaker Office provides high-quality computer fonts MegaFont Now and infiniType. This allows both home users and professional designers access to affordable font libraries.

Features of SoftMaker Office

In this section, we provide the features for each of the productivity tools TextMaker, PlanMaker and Presentations. These features are for the most recent version 2021. PDF files can be created out of every application which is helpful. There are also 32-bit and 64-bit versions of the same package. The software can also be installed to a USB drive for use without installing on a PC. The new license for SoftMaker Office 2021 allows the use on up to five computers running any combination of Windows, MacOS or Linux. With SoftMaker office, you can work on documents in multiple languages.

TextMaker

TextMaker 2021 provides the following features:

- DOCX is the default file format which allows TextMaker to open documents created in Microsoft Word and vice-versa without conversion.
- edit DOC and RTF files in addition to producing PDF files. SoftMaker Office Professional and NX Universal allow one to create e-books in EPUB format.
- has a comprehensive tool set for academic work – footnotes, endnotes, bibliographies, cross references, image captions, indexes, table of contents and tables of figures.
- a built-in spell checker.
- a real-time word counter.
- integrated ten-language thesaurus.
- track all changes made to a document.
- insert comments in the right-hand margin.
- add pictures, drawing, text frames, crop images and add different effects.
- use master pages to watermark pages

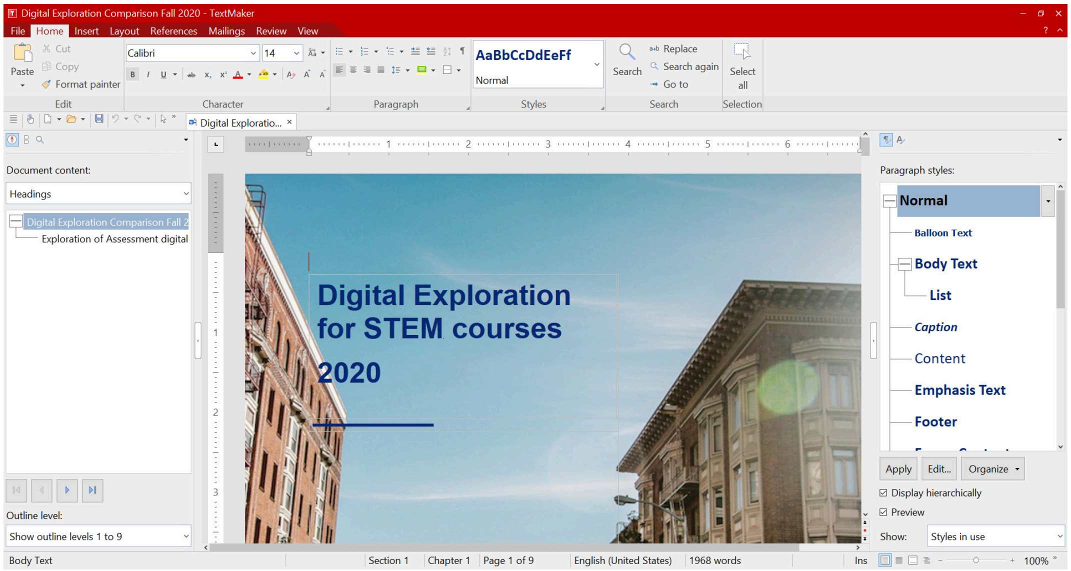


Figure 3.2 A document open in the TextMaker Home menu tab

From Figure 3.2, the TextMaker Home menu can be seen to be similar to that of Word.

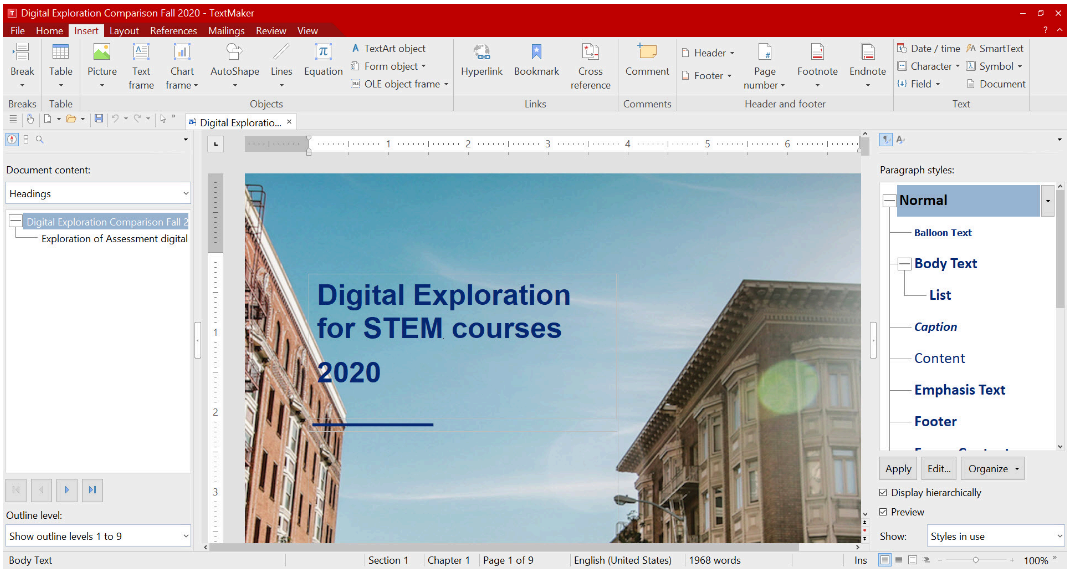


Figure 3.3 The TextMaker Insert menu tab option

PlanMaker

Features of PlanMaker 2021 include:

- default file format is XLSX which allows you to skip import/export. PlanMaker workbooks can be opened in Microsoft Excel and vice versa without conversion.
- edit XLS files and create PDF files from workbooks.
- compatible with Excel.
- syntax highlighting and formula auditing functions.

- excel-style conditional formatting.
- 80 different chart types in 2D and 3D with an array of effects.

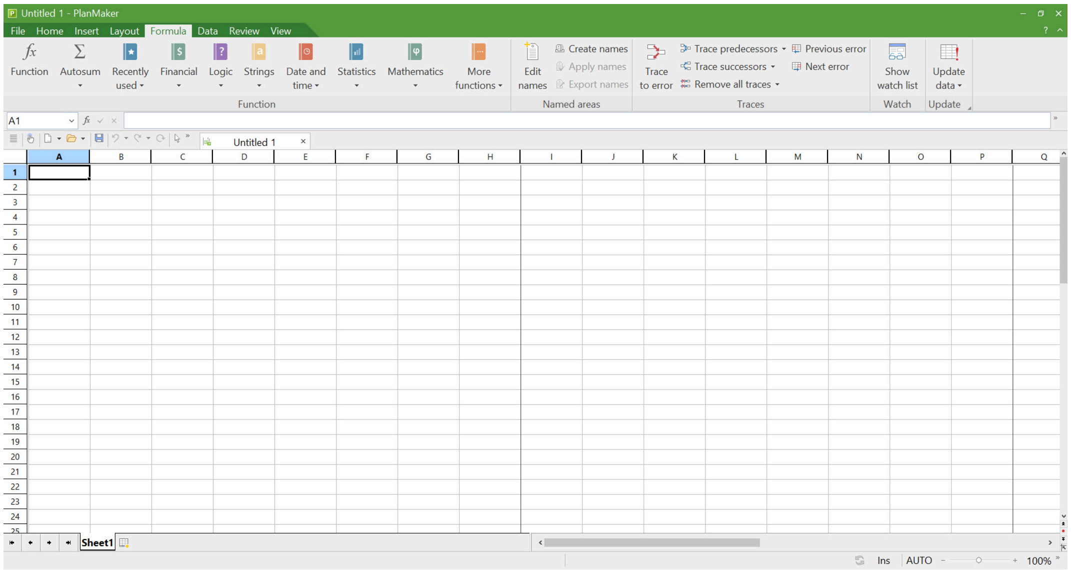


Figure 3.4 PlanMaker open in the Formula tab

As can be seen from Figure 3.4, many of the familiar features in Excel are available in PlanMaker as well. Even the look and feel of the editing menu is similar.

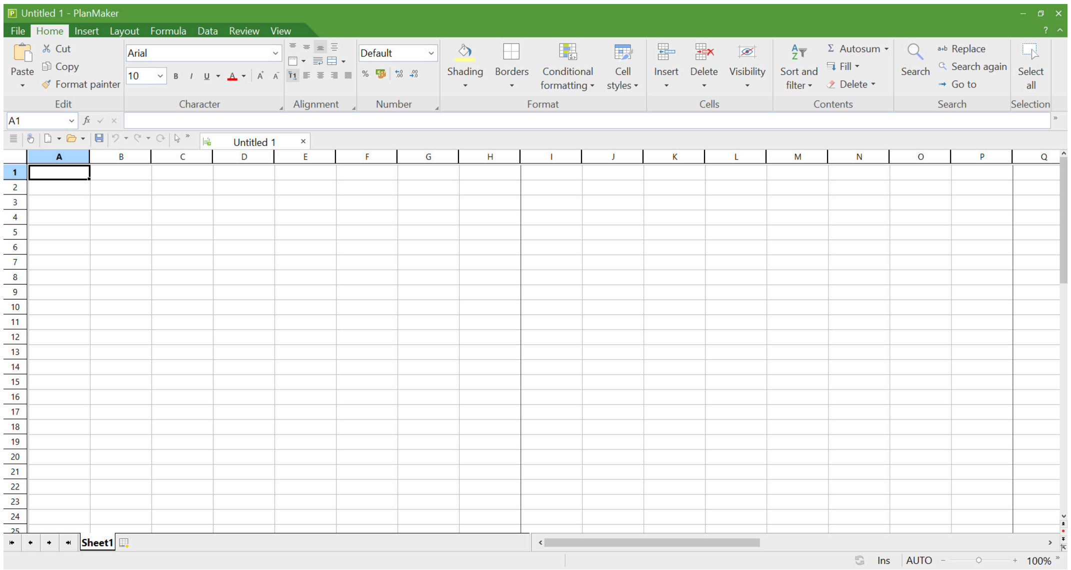


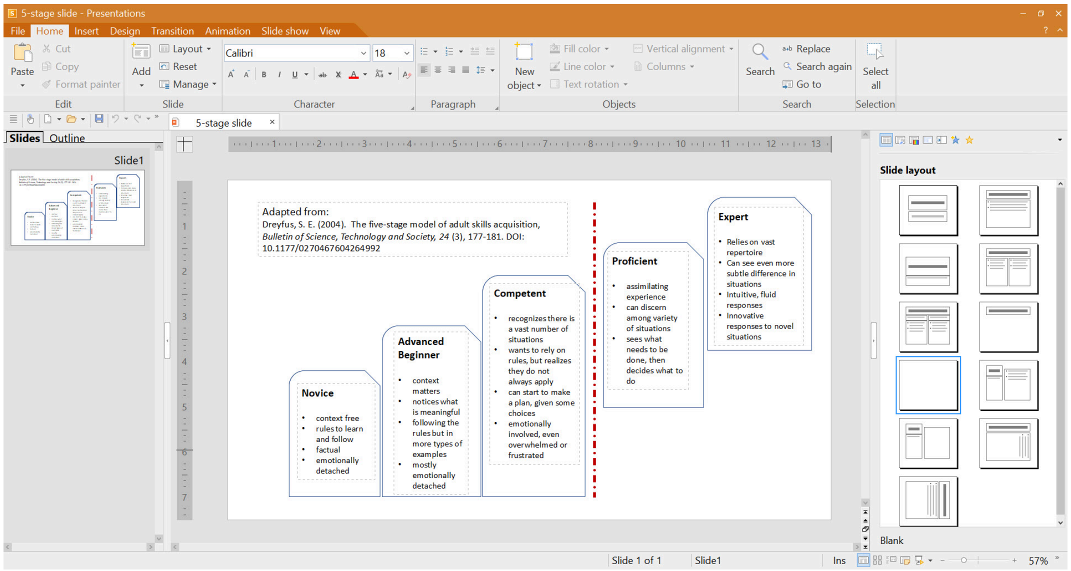
Figure 3.5 PlanMaker open in the Home tab

From this look of the Home menu, the look and feel of PlanMaker is similar to that of Excel. A user choosing to switch between the tools should have no problem adjusting to PlanMaker. It is not clear whether the advanced usage of Excel is comparable to PlanMaker.

Presentations

Features of [Presentations 2021](#) include:

- Default file format is PPTX allowing one to skip import export. Files created by Presentations can be opened in Microsoft PowerPoint and vice-versa.
- edit PPT and export slides as HTML and PDF files.
- master-page concepts supports creation of comprehensive presentations.
- impressive animations and slide transitions using OpenGL graphics acceleration.
- add graphics, tables and charts as well as multimedia.
- add TextArt to create jazzy slogans.



Adapted from:
Dreyfus, S. E. (2004). The five-stage model of adult skills acquisition, *Bulletin of Science, Technology and Society, 24* (3), 177-181. DOI: 10.1177/0270467604264992

Novice
- context free
- rules to learn and follow
- factual
- emotionally detached

Advanced Beginner
- context matters
- notices what is meaningful
- following the rules but in more types of examples
- mostly emotionally detached

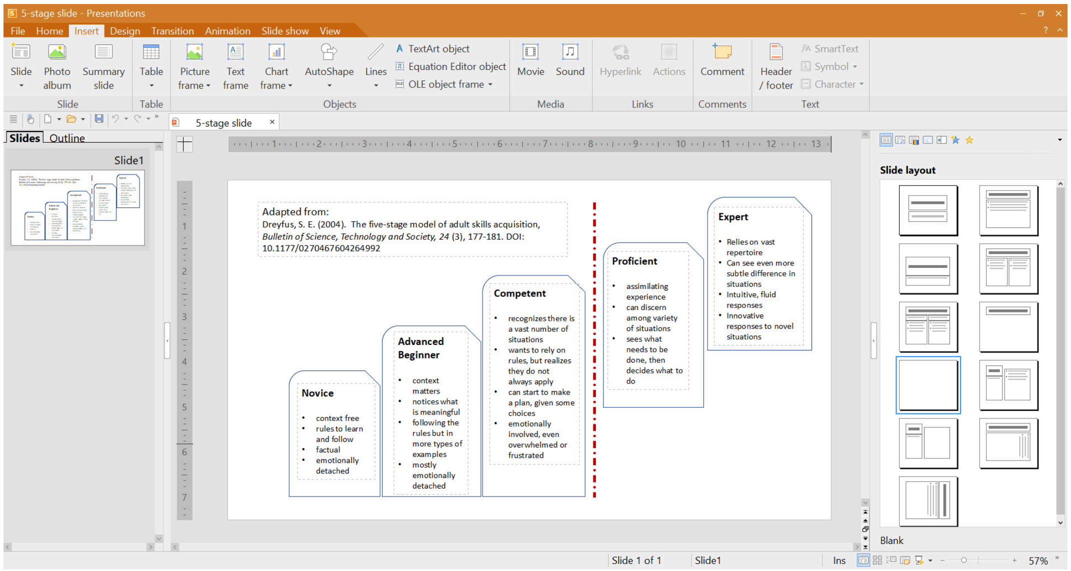
Competent
- recognizes there is a vast number of situations
- wants to rely on rules, but realizes they do not always apply
- can start to make a plan, given some choices
- emotionally involved, even overwhelmed or frustrated

Proficient
- assimilating experience
- can discern among variety of situations
- sees what needs to be done, then decides what to do

Expert
- Relies on vast repertoire
- Can see even more subtle difference in situations
- Intuitive, fluid responses
- Innovative responses to novel situations

Figure 3.6
Presentations open in the Home tab

Again the look and feel of Presentation is similar to PowerPoint which would help those choosing to switch.



Adapted from:
Dreyfus, S. E. (2004). The five-stage model of adult skills acquisition, *Bulletin of Science, Technology and Society, 24* (3), 177-181. DOI: 10.1177/0270467604264992

Novice
- context free
- rules to learn and follow
- factual
- emotionally detached

Advanced Beginner
- context matters
- notices what is meaningful
- following the rules but in more types of examples
- mostly emotionally detached

Competent
- recognizes there is a vast number of situations
- wants to rely on rules, but realizes they do not always apply
- can start to make a plan, given some choices
- emotionally involved, even overwhelmed or frustrated

Proficient
- assimilating experience
- can discern among variety of situations
- sees what needs to be done, then decides what to do

Expert
- Relies on vast repertoire
- Can see even more subtle difference in situations
- Intuitive, fluid responses
- Innovative responses to novel situations

Figure 3.7
Presentations open in the Insert tab

Figure 3.7 shows some of the additional features available in Presentations that are similar to those in PowerPoint.

WPS Office

WPS Office is an all-in-one office suite that allows a person to edit and manage writer, presentation , spreadsheet and PDF collaboratively. It supports 47 files formats, runs on all major operating systems and is available in 46 languages. The supported formats for WPS Office component are:

1. **Writer** – .wps, .Wpt, .doc, .docx, .docm, .dot, .dotx, .dotm, .mht, .mhtml, .htm, .html, .rtf, .txt, .xml
2. **Spreadsheet** – .et, .ett, .xls, .xlt, .xlsx, .xlsm, .dbf, .csv, .prn, .dif, .xltx, .xltm, .xla, .xlam
3. **Presentation** – .dps, .dpt, .ppt, .pptx, .pptm, .pot, .potx, .potm, .pps, .ppsx, .ppsm, .jpg, .png, .tif, .bmp, .ts
4. **PDF** – .pdf

WPS Office has three main tools: Writer, Spreadsheet, Presentation and PDF Editor[11]. In addition, there are PDF tools integrated with the software. WPS Office is meant to function in the cloud which enables collaboration much like Google Docs. There are three plans available: Standard, Premium and Business. The Premium version provides all the functionality of the PDF tools as shown below.

		Standard	**Premium**
PDF Annotation	PDF Fill-in, Annotation, Text Comment, Export Annotation	Yes	Yes
PDF Editing	Edit & Extract PDF Text, Image	No	Yes
Page Management	Manage & Crop & Replace PDF Page	No	Yes
PDF Signature & Encryption	PDF Encrypt, Watermark, E-Signature, Recovery	No	Yes
Insert Object in PDF	Insert PDF image & Background & Attachment	No	Yes
File Format Conversion	Convert PDF & Image to different file formats. Convert different file formats to PDF	No	Yes

Table 3.2 Feature comparison of PDF Benefits in plans

One of the key distinctions that WPS Office has is that its PDF tools are integrated into the suite. Within Writer, there are options for exporting to PDF, picture to PDF and PDF to Word under the Tools tab. This would be similar to the menu items added to Word when Adobe Acrobat Professional is installed. Within Writer, one interesting feature is the ability to record the screen from within the program.

---

[11] See WPS Office

Comparison of Features

Many personal computer users gravitate towards the use of Office 365 due to its popular use at home, school and business. Those advocating for the use of cheaper alternatives have the onus of providing a reasonable rationale for the use of those alternatives. In this section, a brief description of how the two alternatives presented compare with the features available in Word is presented. The following table lists the key features of Word and whether they exist in TextMaker of WPS Office.

Feature	Word	TextMaker	WPS Office
Fonts, Font Size	Yes	Yes	Yes
List Environments	Yes	Yes	Yes
Line & Paragraph spacing	Yes	Yes	Yes
Styles	Yes	Yes	Yes
Editing - Find, Replace, Select	Yes	Yes	Yes
Office Dictation	Yes	No	No
Editor - Spell Check, Grammar, Writing suggestions	Yes	No	Yes
Pages - Cover Page, Blank Page, Page Break	Yes	Yes	Yes
Tables	Yes	Yes	Yes
Illustrations - Pictures, Shapes, Icons, 3 D Models, Smart Art, Chart	Yes	Yes	Yes
Add-ins - Get Add-ins, Wikipedia	Yes	No	No
Media - Online Videos	Yes	No	No
Links - Link, Bookmark, Cross-reference	Yes	Yes	Yes
Comments	Yes	Yes	Yes
Header & Footer	Yes	Yes	Yes
Text - Text Box, Quick Parts, Date and time	Yes	No	Yes
Symbols - Equation, Symbols	Yes	Yes	Yes
Document formatting - Colors, Fonts, Paragraph spacing, Effects	Yes	No	No
Page Background - Watermark, Page Color, Page borders	Yes	No	No (only watermark)
Page Setup - Margins, Orientation, Size, Columns	Yes	Yes	Yes
Paragraph - Indent, Spacing	Yes	No	Yes
Arrange -Position, Wrap Text, Bring Forward, Send Backward, Select Pane, Align, Group, Rotate	Yes	Yes	Yes
Table of Contents	Yes	Yes	Yes
Footnotes - Footnotes, Endnotes	Yes	Yes	Yes
Research - Search, Researcher	Yes	No	No
Citations & Bibliography - Citation, Manage Sources, Style, Bibliography	Yes	Yes	No
Captions - Insert Caption, Insert Table of Figures, Cross-reference	Yes	Yes	Yes
Index	Yes	Yes	Yes
Table of Authorities	Yes	No	No
Proofing - Editor, Thesaurus, Word count	Yes	Yes	Yes
Speech	Yes	No	No
Accessibility check	Yes	No	No
Language - Translate, Language select	Yes	No	No
Comments	Yes	Yes	Yes
Tracking - Track changes, Show Markup, Reviewing Pane, All Markup	Yes	Yes	Yes
Changes	Yes	Yes	Yes
Compare	Yes	No	Yes
Protect - Restrict Editing, Block Authors	Yes	No	Yes
Ink	Yes	No	No

Feature	Word	TextMaker	WPS Office
Resume - Resume Assistant	Yes	No	No

Table 3.3 Comparison of Features available in Word, TextMaker and WPS Office

As Table 3.2 shows, not all Word features are available in the two listed alternatives. One would expect this since Word has been under development for a longer period of time. However, most of the features are available in both options. Added to this is the fact that documents created in Word can be used within both these alternatives without the need to import a document.

There are some features that WPS Office comes with that are not present in either Word of TextMaker. The Tools tab contains the following additional options:

- Export to PDF – export a document to PDF
- Export to picture – save a document as a picture
- Picture to Text – convert a picture to text
- Picture to PDF – convert a picture to a PDF
- PDF to Word – convert a PDF document to Word
- Split or Merge – this allows one to either split or merge files, up to 50 documents can be merged
- Auto Backup – allows a timed back of files
- Files Repair – allows the repair of damaged files
- Screen Recorder – allows one to record and create a video (options to record screen, window, audio and webcam)
- Save to Cloud Docs – saves files to the cloud
- File Collect – allows the collection of files
- Scan to Mobile – scan QR with mobile to receive files, sent file to mobile, attach file to e-mail, provide a link to the file
- Design Library – providing templates for document design

WPS Office includes WPS PDF as part of the suite of tools. A PDF file can be opened directly within this tool and annotated in a manner similar to the features available in Adobe Acrobat Professional.

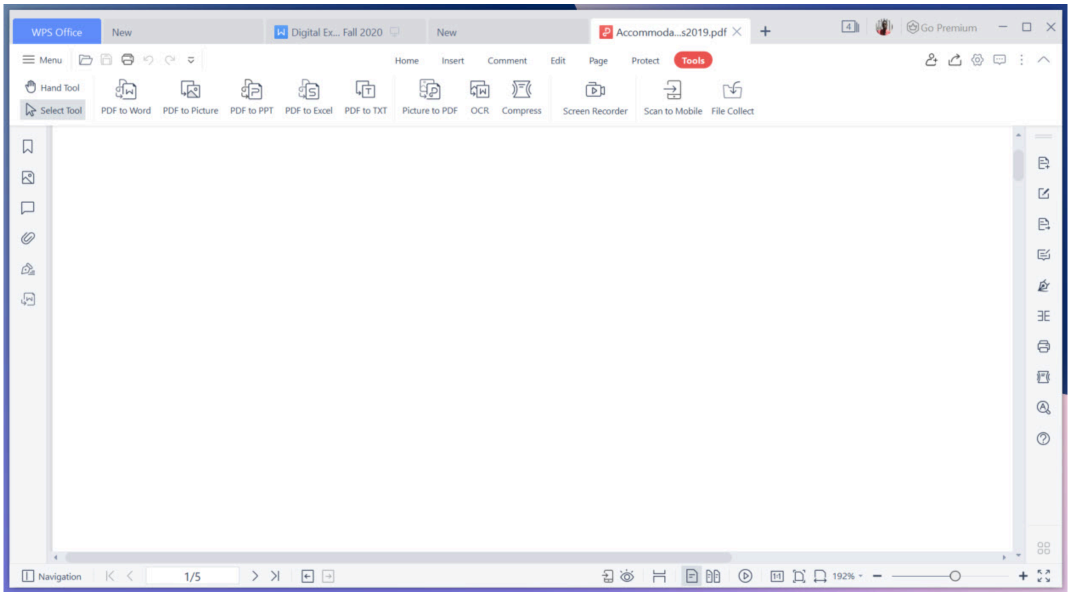


Figure 3.8 A document open in WPS PDF with the tools tab options visible

One of the observations from opening the Tools tab when editing a PDF file is that there are some additional options available:

- PDF to Word – convert the document to Word
- PDF to picture – convert the document to a picture
- PDF to PPT – convert the document to PowerPoint
- PDF to Excel – convert the document to Excel
- PDF to TXT – convert the document to TXT files
- Picture to PDF – convert a picture to PDF
- OCR – convert picture to text format
- Compress – reduce the size of PDF files
- Screen Recorder – record the screen and create a video
- Scan to mobile – scan QR with mobile to receive files, attach file to e-mail, provide a link to the file
- File Collect – allows the collection of files

This has interesting applications for those who receive a Creative Commons licensed resource as a PDF with no editable source. WPS PDF essentially will allow you to use the PDF to Word convertor to obtain editable source for the document.

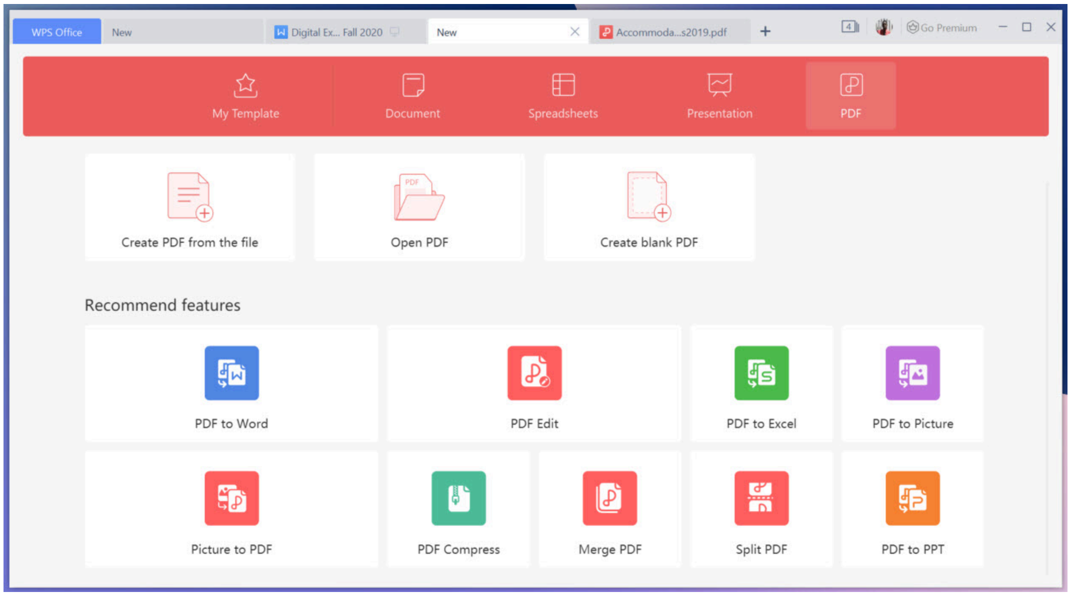


Figure 3.9 The WPS PDF menu for editing PDF documents

Conclusion

While Office 365 continues to be the most popular word processing tool on personal computers, there are some emerging tools with comparable features at a fraction of the cost. Two such tools have been examined in this chapter, SoftMaker Office and WPS PDF. Both tools can edit documents in Office 365 formats thus forgoing the need to import document.

SoftMaker Office provides three tools – TextMaker, PlanMaker and Presentations. These are similar to Word, Excel and PowerPoint respectively. Document exchange between Office 365 and the three SoftMaker Office products are seamless. The price for SoftMaker Office is relatively cheaper than Office 365 through a one-time purchase option.

There is also a subscriptions option available. In addition, a user can purchase FlexiPDF for either an additional onetime fee or by subscription. FlexiPDF provides the usual features associated with editing PDF files (similar to Adobe Acrobat Professional options). It allows the editing of PDF files as easily as with a word processor which is useful for those working with PDF files.

WPS Office provides the ability to edit and manage files with Writer, Presentation, Spreadsheet and PDF at the same time. It is fully supported on Windows, MacOS, Linux, Android and iOS and also supports 47 file formats and 46 languages. A key feature of this tool is that it operates through the cloud in a similar way to Google Docs. It can be used for free with limited features or with one of two paid subscriptions: WPS Premium and WPS Business. Both subscriptions allow use on multiple devices simultaneously. They also come with a significant amount of cloud storage. WPS Office has the additional feature of providing a Screen Recorder to create video. In addition, the WPS PDF provides a number of useful convertors that would make a good addition to any document creation workflow.

4. Open-source Word Processing Tools

ISAAC MULOLANI

Learning Outcomes

By the end of this chapter, the reader will be able to:

1. Identify the various open-source word processing tools for creating OER
2. Identify word-processing features appropriate for the creation of OER
3. Describe additional tools that can be used in conjunction with open-source word processing programs

Open-source Word Processing Options

Following on from the previous chapter, we will now look at open-source options. In this chapter, there are a couple of tools that complement the word processing tools that will be discussed. The first tool will be LibreOffice which started as a fork of the OpenOffice office suite. This free suite is used by millions of people around the globe.

The next tool to discuss will be the familiar Google Docs which is a cloud-based free word processing option. These tools allows online collaboration and sharing of documents. The software suite now has the ability to extend the functionality of the software using a number of add-ons.

When creating documents in word processors, there are times when pictures, graphics and photos are needed. As was mentioned in the previous chapter, any graphics to be used in a document need to be either created or brought in from image repositories. This is usually done with external tools. In this chapter we will look at GIMP (Gnu Image Manipulation Program) , a free tool that can be used in support of document creation. Another free tool is called Inkscape which is a drawing program. Inkscape has been considered by some to be an open-source version of Adobe Illustrator.

LibreOffice

LibreOffice is compatible with a wide range of document formats such as Microsoft® Word (.doc, .docx), Excel (.xls, .xlsx), PowerPoint (.ppt, .pptx) and Publisher. But LibreOffice goes much further with its native support for a modern and open standard, the Open Document Format (ODF). With LibreOffice, you have maximum control over your data and content – and you can export your work in many different formats including PDF.

The LibreOffice website describes the software to be a successor to OpenOffice that has a world-wide user base. It contains several applications:

1. Writer – for word processing
2. Calc – for spreadsheets
3. Impress – for presentations
4. Draw – for vector graphics and flowcharts
5. Base – for database
6. Math – for editing formulae

For the sake of this writing, our interest is to see how many features LibreOffice's Writer provides in comparison to Word. The following table provides a comparison of the two word processors.

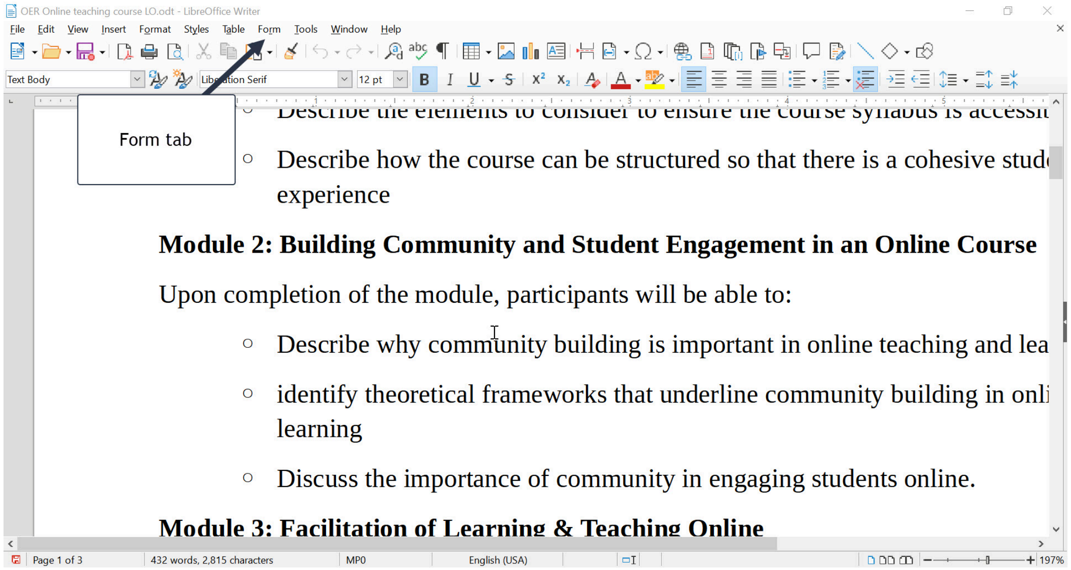
Feature	**Word**	**LibreOffice**
Fonts, Font Size	Yes	Yes
List Environments	Yes	Yes
Line & Paragraph spacing	Yes	Yes
Styles	Yes	Yes
Editing - Find, Replace, Select	Yes	Yes
Office Dictation	Yes	No
Editor - Spell Check, Grammar, Writing suggestions	Yes	Yes
Pages - Cover Page, Blank Page, Page Break	Yes	Yes
Tables	Yes	Yes
Illustrations - Pictures, Shapes, Icons, 3 D Models, Smart Art, Chart	Yes	Yes
Add-ins - Get Add-ins, Wikipedia	Yes	No
Media - Online Videos	Yes	No
Links - Link, Bookmark, Cross-reference	Yes	Yes
Comments	Yes	Yes
Header & Footer	Yes	Yes
Text - Text Box, Quick Parts, Date and time	Yes	Yes
Symbols - Equation, Symbols	Yes	Yes
Document formatting - Colors, Fonts, Paragraph spacing, Effects	Yes	Yes
Page Background - Watermark, Page Color, Page borders	Yes	No (only watermark)
Page Setup - Margins, Orientation, Size, Columns	Yes	Yes
Paragraph - Indent, Spacing	Yes	Yes
Arrange -Position, Wrap Text, Bring Forward, Send Backward, Select Pane, Align, Group, Rotate	Yes	No
Table of Contents	Yes	Yes
Footnotes - Footnotes, Endnotes	Yes	Yes
Research - Search, Researcher	Yes	No
Citations & Bibliography - Citation, Manage Sources, Style, Bibliography	Yes	No (only Bibliography)
Captions - Insert Caption, Insert Table of Figures, Cross-reference	Yes	No (only cross-reference)
Index	Yes	Yes
Table of Authorities	Yes	No
Proofing - Editor, Thesaurus, Word count	Yes	Yes
Speech	Yes	No
Accessibility check	Yes	Yes
Language - Translate, Language select	Yes	Yes
Comments	Yes	Yes
Tracking - Track changes, Show Markup, Reviewing Pane, All Markup	Yes	Yes
Changes	Yes	Yes
Compare	Yes	No
Protect - Restrict Editing, Block Authors	Yes	Yes
Ink	Yes	No

Feature	Word	LibreOffice
Resume - Resume Assistant	Yes	No

Table 4.1 Comparison of features in Word vs LibreOffice

A more detailed comparison is provided on the [Document Foundation Wiki](). This contains a comparison between LibreOffice 7.3.0 vs MS Office 2021/365.

One of the key differences from examining the Writer menu is that LibreOffice has a Form tab with options for creating forms.



- Describe the elements to consider to ensure the course syllabus is accessi
- Describe how the course can be structured so that there is a cohesive stud experience

Module 2: Building Community and Student Engagement in an Online Course

Upon completion of the module, participants will be able to:

- Describe why community building is important in online teaching and lea
- identify theoretical frameworks that underline community building in onli learning
- Discuss the importance of community in engaging students online.

Module 3: Facilitation of Learning & Teaching Online

Figure 4.1 Writer editor with the menu including the Form tab

LibreOffice Extensions

A very interesting feature of LibreOffice is the ability to install extensions through the Extension Manager under the Tools tab. The [LibreOffice Extensions]() page has a list of extensions and the ability to search for extensions as well. Clearly, this adds many more features and/or templates that a person can install to use for document creation. One example is that for those needing it, there is an [APA extension]() that can be installed.

Additional LibreOffice Tools

LibreOffice comes with the Draw program which enables you to draw some limited picture elements. This is helpful for creating simple graphics for inclusion in documents. Another included program is the Math program that may be useful for some people. The Calc program is the option for working with spreadsheets similar Excel.

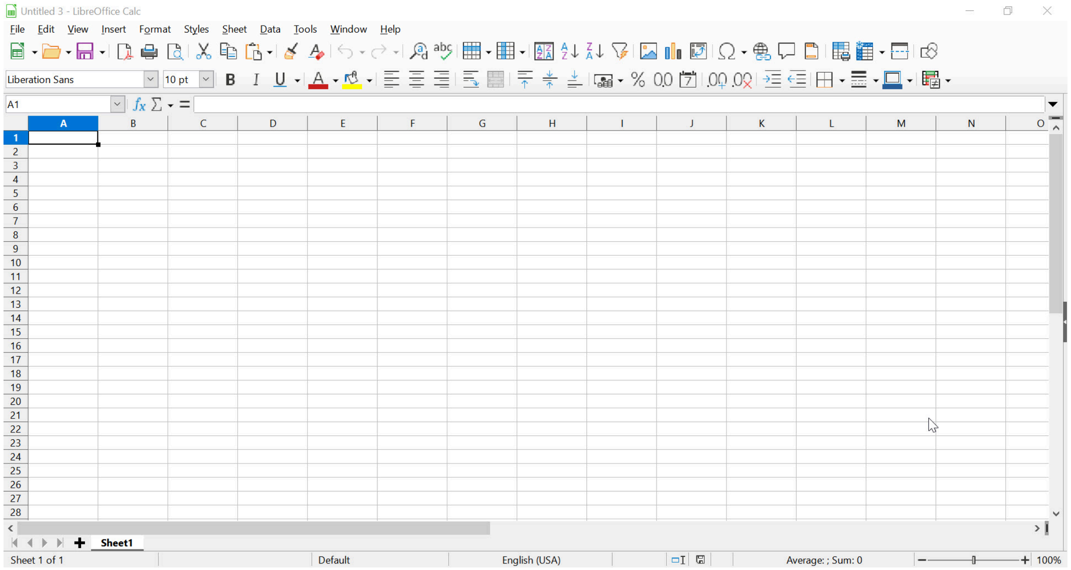

Untitled 3 - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Liberation Sans
10 pt
A1
Sheet1
Sheet 1 of 1
Default
English (USA)
Average: ; Sum: 0
100%

Figure 4.2 Calc program editing menu

Impress provides the ability to create presentations in a similar manner to PowerPoint.

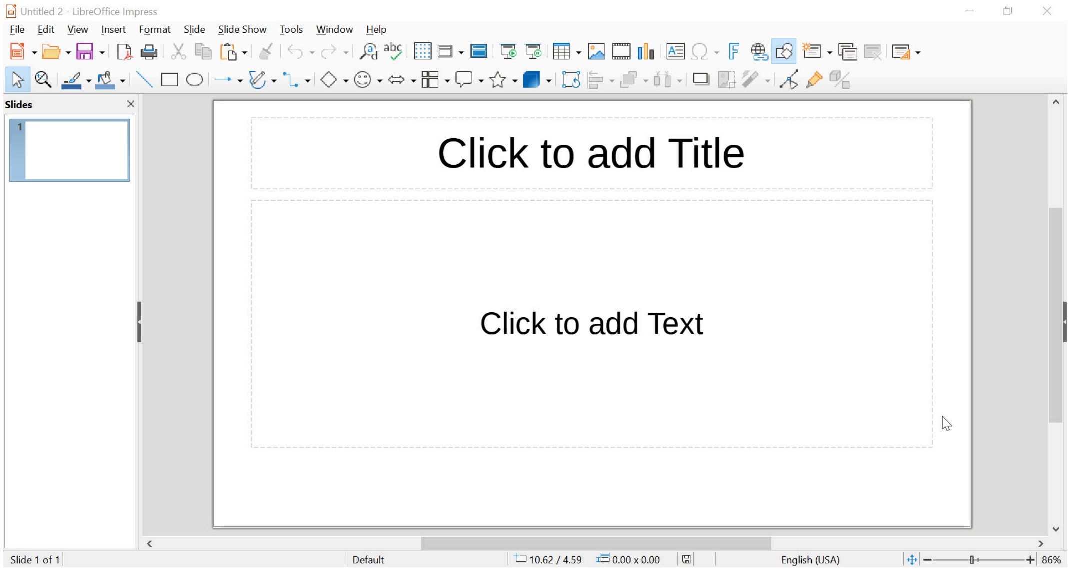


Figure 4.3 Impress editing menu

Presentation created in PowerPoint can be edited within Impress since LibreOffice supports the file formats from Office 365. Some of the [features of Impress](https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=26) are described in the H5P below.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=26#h5p-1

Reasons for lack of Popularity of LibreOffice

The Linux Pro Magazine in their Winter 2020 issue 40, featured LibreOffice in this issue. One question addressed is, if LibreOffice really is totally free and compatible with Microsoft Office, then why hasn't the whole world moved over to it? Why is MS Office still popular around the world? Here are some key reasons.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=26#h5p-4

What is the OpenDocument Format?

LibreOffice uses the OpenDocument Format (ODF), a fully open and ISO standardised file format that guarantees access to your data forever. (You can, of course, encrypt your documents with a password.) Because ODF is standardised, other office software can implement support for it as well – and many programs have done so. By using ODF, you ensure that your data can be transferred between different computers and operating systems, without having to worry about vendor lock-in or license fees.

ODF extensions

Typical extensions for ODF files include the following:

- **.odt** – a text document
- **.ods** – a spreadsheet file
- **.odp** – a presentation file
- **.odg** – an illustration or graphic

What to do if you're sent an ODF file

If you are sent a file with one of the above extensions, but your software or operating system can't identify it, then simply download LibreOffice – it's free and open source software, originally based on OpenOffice.org, and handles all of the above extensions.

Google Documents

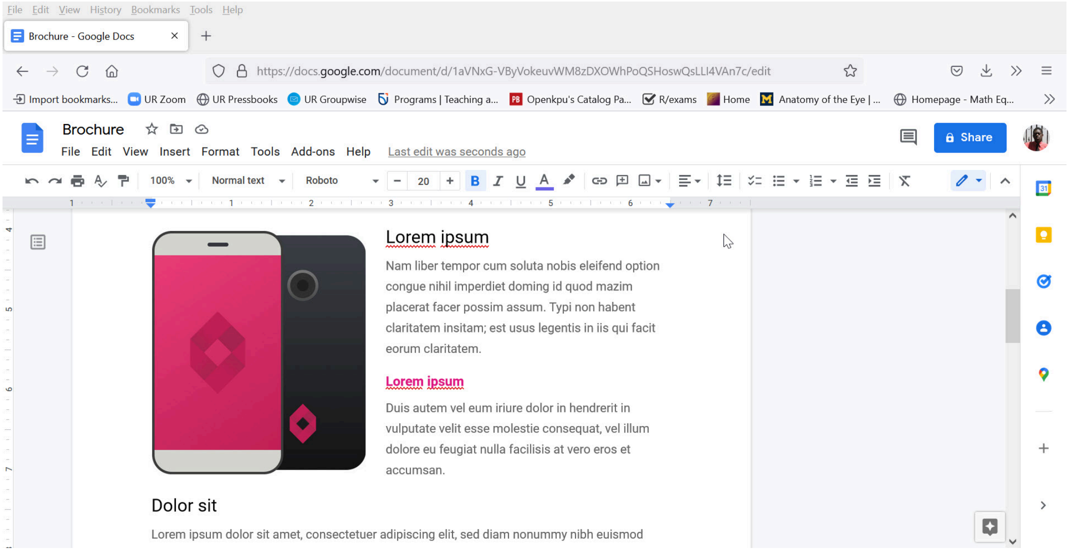
Google Docs is an online office suite that allows the creation and collaboration on online documents in real-time and from any device. The following are the features of Google docs.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=26#h5p-5

There are two different options for using [Google Docs:](#) free for personal use or Business standard for $12 USD.

	For Personal (Free)	**Business Standard** **$12 USD/user/month**
Docs, Sheets, Slides, Forms content creation	Yes	Yes
Drive Secure cloud storage	15 GB per user	2 TB per user
Shared drives for your team	no	Yes
Gmail Secure e-mail	Yes	Yes
Custom business email	No	Yes
Meet Video and voice conferencing	100 participants	150 participants
Meeting recordings save to Drive	No	Yes
Admin Centralized administration	No	Yes
Group-based security policy controls	No	Yes
Customer support	Self-service online and community forums	24/7 online support and community forums

Figure 4.4 shows a screen capture of Google Docs editing menu.


File Edit View History Bookmarks Tools Help
Brochure - Google Docs
https://docs.google.com/document/d/1aVNxG-VByVokeuvWM8zDXOWhPoQSHoswQsLLl4VAn7c/edit
Import bookmarks... UR Zoom UR Pressbooks UR Groupwise Programs | Teaching a... Openkpu's Catalog Pa... R/exams Home Anatomy of the Eye | ... Homepage - Math Eq...
Brochure
File Edit View Insert Format Tools Add-ons Help
Last edit was seconds ago
Share
100% Normal text Roboto 20

Lorem ipsum

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem.

Lorem ipsum

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan.

Dolor sit

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod

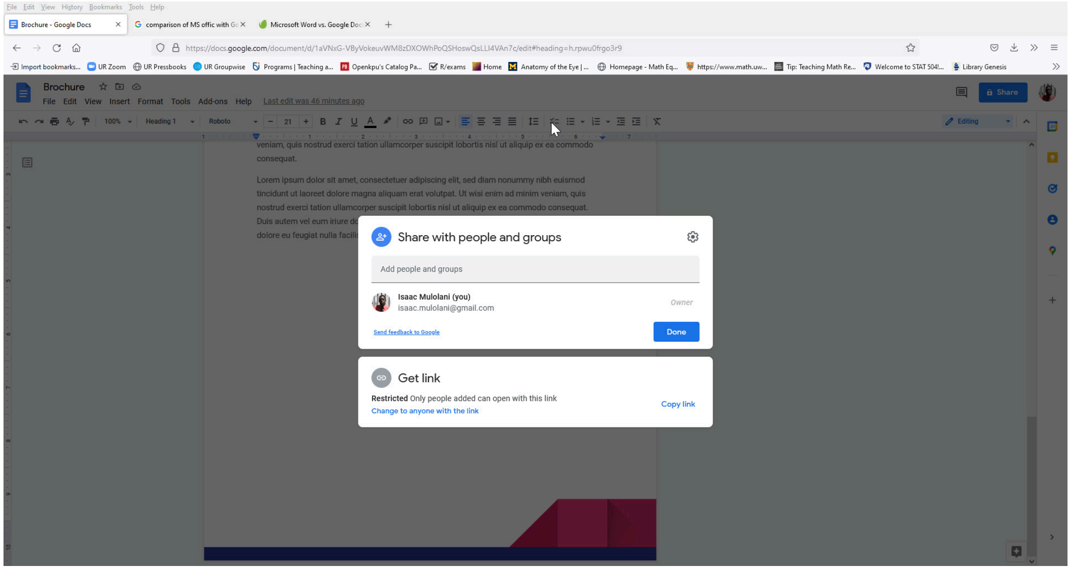
Figure 4.4 Google Docs editor with menu options

There are some similar editing menu tab items. The one that is different is the Add-ons tab option that allows one to extend the features of Google Docs. On the right side of Figure 4.4, there are the following options:

- Calendar – add items to the calendar
- Keep – Take notes for the document

- Tasks – add some tasks
- Contacts – access the contacts in Google
- Maps – access Google Maps
- Get Add-ons – one can add additional add-ons

At the top right of this editing screen is an Editing tab which allows collaborative editing. Another tab adds comments and shows the comment history. The final item is the Share button.


Brochure - Google Docs
https://docs.google.com/document/d/1aVNxG-VByVokeuvWM8zDXOWhPoQSHoswQsLLI4VAn7c/edit#heading=h.rpwu0frgo3r9
Brochure
File Edit View Insert Format Tools Add-ons Help
Last edit was 46 minutes ago
100%
Heading 1
Roboto
21
Editing
Share

veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure do… dolore eu feugiat nulla facili…

Share with people and groups

Add people and groups

Isaac Mulolani (you)
isaac.mulolani@gmail.com
Owner

Send feedback to Google
Done

Get link

Restricted Only people added can open with this link
Change to anyone with the link
Copy link

Figure 4.5 Google Docs Sharing activated

This provides the option to share a document with other people, groups or provide a link that provides everyone with the link access to the document.

	Google Docs	Microsoft Word
Overall Findings	• Free for all features. • Easy to use from anywhere. • All the basic editing tools most people need. • Useful and simple sharing options.	• Robust desktop program. • Great for offline work. • Free during the trial period.
Cost	Its all free	Free only during a limited-time trial
Features	• Lots of basic features. • Useful for light writing requirements. • Can open DOCX files from Word.	• Extensive menus full of options. • Ideal for research and writing. • Accepts several document file formats.
Mobility	• Website access from anywhere. • Mobile app for Android and iOS. • Consistency no matter where it's used.	• Runs on Windows and Mac. • Mobile app for Android and iOS.
Sharing	• Ultra-fast collaboration. • Built-in email form.	• Delayed updates when working with shared files. • Requires a desktop email client.
Offline Usage	• Relies on an active internet connection. • Supports offline access but it's off by default.	• Runs entirely offline. • Easy to save documents to attached hard drives.

Table 4.1 Comparison of features between Google Docs and Microsoft Word.[1]

When looking at this feature comparison, it is clear that both tools are useful for different purposes. Google Docs is perfect for those who don't want to pay for a word processor but still need a functioning method for viewing and editing documents. Google Docs still works with Word files and has all the features most people need. It is also easier to understand and works great for sharing and backing up documents. On the other hand Word has more features and has long been the business standard. To get an updated version of Word installed, you will have to pay for all those benefits.

Open-source Tools for use with Word Processors

In this section, we describe some additional tools that can be used in combination with the LibreOffice suite of tools. GIMP is an open source cross-platform image editor available for GNU/Linux, macOS, Windows and other operating systems. Inkscape is Free and Open Source Software (FOSS) licensed under Gnu General Public License. Inkscape is free! By this, we mean it is free of cost, free to use and distribute, and open to peek into the source code if you wish to do so.

---

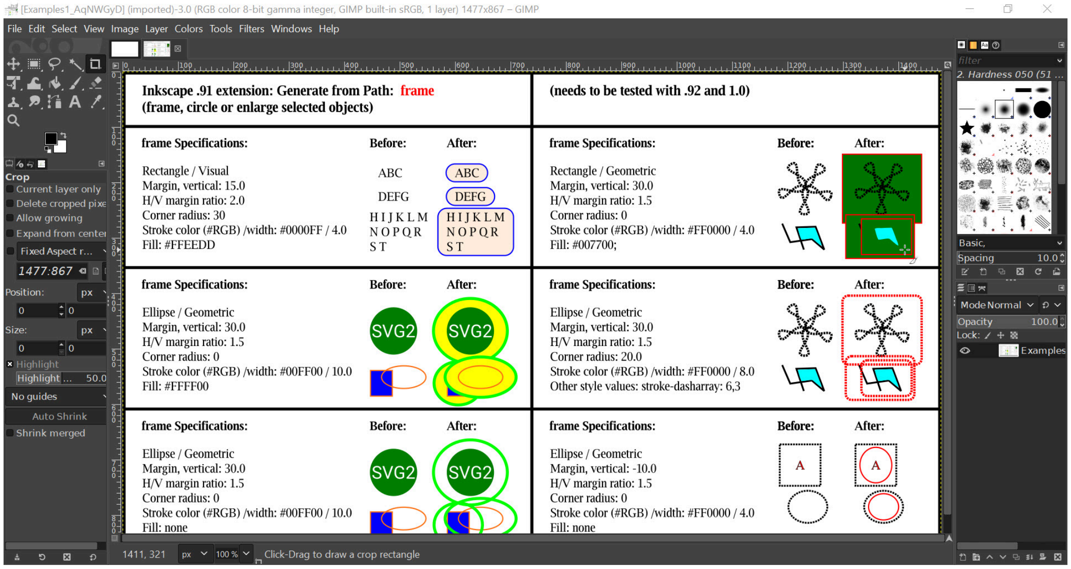
[1] See Lifewire Tech for Humans.

Gnu Image Manipulation Program (GIMP)

GIMP is an acronym for GNU Image Manipulation Program. It is a freely distributed program for such tasks as photo retouching, image composition and image authoring. The terms of usage and rules about copying are clearly listed in the GNU General Public License. There is a nice Frequently Asked Questions (FAQ) page.

The GIMP program is a cross platform image editor available for multiple operating systems. Since it is free software, you can change its source code and distribute your changes. Whether you are a graphic designer, photographer, illustrator, or scientist, GIMP provides you with sophisticated tools to get your job done. You can further enhance your productivity with GIMP thanks to many customization options and 3rd party plugins.

GIMP provides top-notch color management features to ensure high-fidelity color reproduction across digital and printed media. From the program's website we are told that GIMP is best used in workflows involving other free software such as Scribus, Inkscape, and SwatchBooker. These are other open-source tools that are under a GNU General Public License.



Inkscape .91 extension: Generate from Path: frame
(frame, circle or enlarge selected objects)

(needs to be tested with .92 and 1.0)

frame Specifications: **Before:** **After:**

Rectangle / Visual
Margin, vertical: 15.0
H/V margin ratio: 2.0
Corner radius: 30
Stroke color (#RGB) /width: #0000FF / 4.0
Fill: #FFEEDD

ABC
DEFG
H I J K L M
N O P Q R
S T

frame Specifications: **Before:** **After:**

Rectangle / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #FF0000 / 4.0
Fill: #007700;

frame Specifications: **Before:** **After:**

Ellipse / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #00FF00 / 10.0
Fill: #FFFF00

SVG2

frame Specifications: **Before:** **After:**

Ellipse / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 20.0
Stroke color (#RGB) /width: #FF0000 / 8.0
Other style values: stroke-dasharray: 6,3

frame Specifications: **Before:** **After:**

Ellipse / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #00FF00 / 10.0
Fill: none

SVG2

frame Specifications: **Before:** **After:**

Ellipse / Geometric
Margin, vertical: -10.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #FF0000 / 4.0
Fill: none

A

Figure 4.6 An image open in the GIMP editor

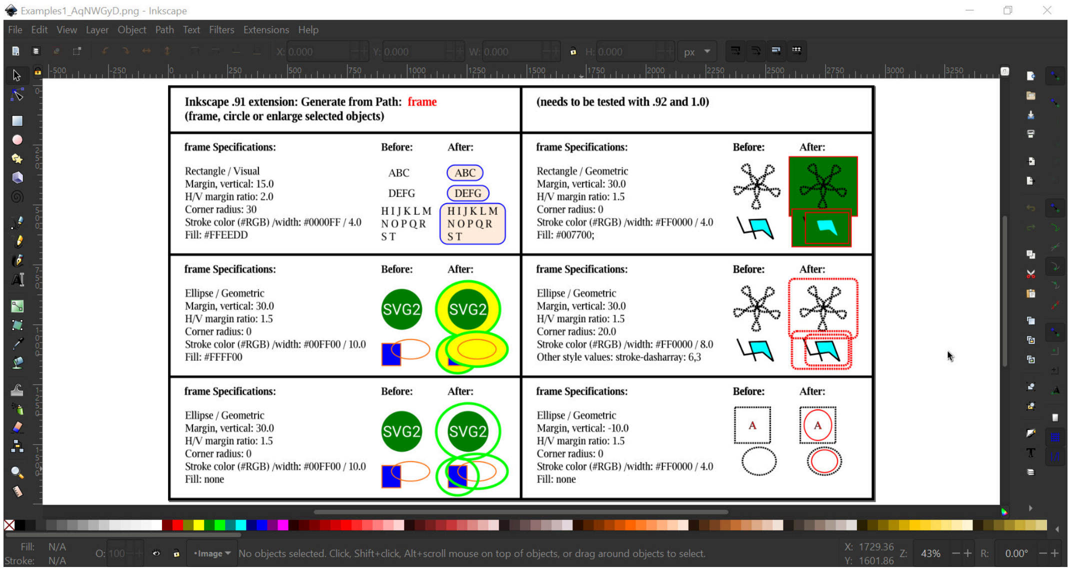
Features of GIMP

GIMP is a versatile graphics manipulation package. The following H5P provides a summary of some of the features of the tool.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=26#h5p-3

Inkscape

Inkscape has been developed for designers of all kinds. The tool allows you to take a project from a doodle on a napkin to a final professional-grade design format which is ready for publication on the web or in physical form. If you are new to the process of creating vector graphics it may feel different, but you will quickly be pleased by the flexibility, and power Inkscape offers. Vector design is often the preferred method of image creation for logos, illustrations and art which require high scalability. The Inkscape application is used across a wide variety of industries (marketing/branding, engineering/CAD, web graphics, cartooning) and individual uses.


Examples1_AqNWGyD.png - Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
X: 0.000 Y: 0.000 W: 0.000 H: 0.000 px
Before:
After:
ABC
DEFG
H I J K L M
N O P Q R
S T
SVG2
A
Fill: N/A
Stroke: N/A
O: 100
•Image
X: 1729.36
Y: 1601.86
Z: 43%
R: 0.00°

Inkscape .91 extension: Generate from Path: frame
(frame, circle or enlarge selected objects)

(needs to be tested with .92 and 1.0)

frame Specifications:

Rectangle / Visual
Margin, vertical: 15.0
H/V margin ratio: 2.0
Corner radius: 30
Stroke color (#RGB) /width: #0000FF / 4.0
Fill: #FFEEDD

frame Specifications:

Rectangle / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #FF0000 / 4.0
Fill: #007700;

frame Specifications:

Ellipse / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #00FF00 / 10.0
Fill: #FFFF00

frame Specifications:

Ellipse / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 20.0
Stroke color (#RGB) /width: #FF0000 / 8.0
Other style values: stroke-dasharray: 6,3

frame Specifications:

Ellipse / Geometric
Margin, vertical: 30.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #00FF00 / 10.0
Fill: none

frame Specifications:

Ellipse / Geometric
Margin, vertical: -10.0
H/V margin ratio: 1.5
Corner radius: 0
Stroke color (#RGB) /width: #FF0000 / 4.0
Fill: none

No objects selected. Click, Shift+click, Alt+scroll mouse on top of objects, or drag around objects to select.

Figure 4.7 The Inkscape editor with an open image

Features of Inkscape

In this section, the [features of Inkscape](https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=26#h5p-2) are briefly described. The H5P summarises these features.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=26#h5p-2

These features appear similar to those found in the commercial tool Adobe Illustrator. For this reason, some people consider Inkscape to be an open-source alternative to Adobe Illustrator.

5. Open-source Document Creation Tools I

ISAAC MULOLANI

Learning Outcomes

By the end of this chapter, the reader will be able to:

1. Describe what the LaTeX document typesetting system is and how it can be used
2. Identify some relevant packages/elements of LaTeX that can be used to create OER
3. Describe the LaTeX installation process on a Windows systems
4. Identify an appropriate LaTeX editor from a list of available options
5. Describe the basic layout required to create a LaTeX document

The LaTeX Document Preparation System

Donald Knuth of Stanford University began working on a text processing system known as TeX and METAFONT. Knuth wrote in the forward to his book, *The TEXbook*.[1]

> TEX [is] a new typesetting system intended for the creation of beautiful books – and especially for books that contain a lot of mathematics. By preparing a manuscript in TEX format, you will be telling a computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the world's finest printers.

While Knuth was developing TeX in the early 1980s, Leslie Lamport began work on a document preparation system. This new system, called LaTeX, made use of TeX's typesetting engine and macro system. In this system, high-level LaTeX commands allowed a user to easily compose a range of documents without worrying about typographical issues.

---

[1] See Donald Knuth. The TEXbook, volume A of Computers and Typesetting. Addison-Wesley, Reading, MA, USA, 1986.

The first release of this system appeared in 1985 and was numbered 2.09. The release was accompanied by the second edition of LaTeX: A Document Preparation System, which led to a rapid spread of TeX-based document processing beyond the community of mathematicians. In the following decade, its use spread very rapidly becoming extremely popular in the scientific and academic communities and extensively used in industry.

Mirror Sites for Access to LaTeX

Since LaTeX has a current global following, it is helpful to describe the mirrors that exist across the globe. There are several different flavours of LaTeX that can be downloaded for installation on different platforms. These will be described in this section. There are three main variants of LaTeX available: MiKTeX, TeXLive and MacTeX. These systems can be used on different operating systems.

TeXLive

TeXLive is designed to be a straight forward way to get up and running with the TeX document preparation system. It provides a comprehensive TeX system with binaries for most flavors of Unix, including GNU/Linux, macOS, and also Windows. It includes all the major TeX-related programs, macro packages, and fonts that are free software, including support for many languages around the world. Many operating systems provide it via their own distributions. Instructions are provided on how to install these on different systems. This flavour can be installed on various Linux distributions and a specific version (MacTeX) has been designed for installation on MacOS personal PC and Intel platforms.

MiKTeX

From the TeX User Group mirrors, one can obtain proTeXt which is a MiKTeX distribution designed for installation on a Windows system. ProTeXt guides[2] the MiKTeX installation via a short pdf document (available in German, English, French and Italian), which provides clickable links to install the various components, along with explanations. For ease of use, ProTeXt also adds TeXstudio, and Sumatra PDF.

MacTeX

Again, from the TeX User Group mirrors, one can obtain MacTeX which is an easy-to-install TeX distribution for MacOS. The distribution contains all of TeXLive along with GUI applications and the most recent version of Ghostscript. One of the editors that can be used with MacTeX is TeXShop. As always, there are other options for editors as well.

---

[2] See ProTeXt.

TeX Collection DVD

All members of TUG can get a copy of the *TeX Collection*. This is the name given to the complete collection of software distributed by TUG each year. Non-members can also purchase the TeX Collection DVD from the TUG Store. The components that are included in the TeX Collection DVD are:

- **TeXLive** – a comprehensive cross-platform system. It includes support for Unix-like systems, MacOS and Windows
- **MacTeX** – an easy to install TeX system for MacOS based on TeXLive. Included are a native Mac installer and the TeXShop editor in addition to some Mac specific tools.
- **proTeXt** – an easy to install TeX system for Windows that is based on MiKTeX. It comes with a detailed installation guide and some Windows-specific tools including the TeXworks editor.
- **CTAN** – the DVD contains a snapshot of the backbone CTAN server. It is a subset of what is available on the public servers.

The creation of the TeX Collection involves hundreds of people from all over the TeX world. There are contributers uploading packages, maintainers providing a repository, individuals building binaries for a variety of platforms, people testing results, developers supporting the software, user group members keeping the infrastructure TUG provides, etc. Every one of these individuals volunteers their time for this project.

Advantages and Disadvantages in using LaTeX

This section will discuss the pros and cons of using LaTeX to create documents. This way, potential users will have an opportunity to determine whether an investment of time in learning the system is worth it. To start, the following are some common arguments against using LaTeX:[3]

- **LaTeX is a difficult program to learn**. There is a learning curve to get up to speed with being able to use LaTeX. However, in the long run it will save you time through code re-use.
- **LaTeX is not a What You can See is What You Get (WYSIWYG) word processor**. There are many editors that can be used with LaTeX plugins.
- **There is little support for physical markup**. For most areas across subjects, there are packages that can be used without needing to tweak the look and feel.
- **Using non-standard fonts is difficult**. The fontspec package and XeLaTeX make this easy.
- **It takes practice to let text flow around pictures**. Text flow can be improved by rearranging words but it does take practice.
- **LaTeX doesn't provide spell check**. Most editors have spell checkers. There are also commandline tools for this.
- **There are too many packages**. While that may be true, a document only requires a few packages.
- **LaTeX encourages structured writing and the separation of style from content which is not how everybody works**. There are more communities starting to use LaTeX which would indicate that this is not as big a problem anymore.

---

[3] See LaTeX and Friends by M.R.C. Van Dongen, Springer 2012.

The advantages of using LaTeX are listed in the following.[4]

- LaTeX provides state-of-the art typesetting.
- Many conferences and publishers accept documents in LaTeX. The majority of them provide classes and packages to ensure that documents meet submission formatting requirements.
- LaTeX is a Turing-complete programming language.**Turing-complete programming language** This provides almost complete control over your document.[5]
- LaTeX can be used to prepare several documents from the same source. Some recent document creation platforms such as Pressbooks have this feature.
- LaTeX is highly configurable. The appearance of a document is determined by choosing the appropriate document class, class options, packages and package options. In addition, the proper use of command (from packages and user defined) gives control over the appearance of your document.
- LaTeX can be translated to multiple formats including html/ps/pdf/DocBook, etc.
- LaTeX automates chapter, section, equation, figure numbering among many others. Support for cross-reference is also provided.
- LaTeX provides excellent support for bibliography. The organisation of the bibliography and the style of citations is also configurable.
- Support for WYSIWYG document preparation exists: LyX, TeXmacs, etc. There are many editing environments that provide support for LaTeX, e.g., vim, emacs, eclipse, Sublime Text, etc.
- LaTeX is very stable and freely available on most operating systems.
- As mentioned earlier, there is a worldwide active TeX/LaTeX user-base.
- LaTeX has comments.
- LaTeX is fun to work with.

LaTeX Installation on Windows

In this section, the installation of MiKTeX on windows is described. The first step to this installation start with visiting the TUG site. The steps to installing are enumerated below. One should set aside at least one hour for installation.

1. Download ProTeXt-3.2-031721.zip from www.tug.org/protext/ directly from CTAN: https://ctan.org/pkg/protext
2. Unzip ProTeXt-3.2-031721.zip into a temporary folder
3. Navigate into this temporary folder and double-click on the setup.exe
    - Install MiKTeX from the ProTeXt Panel
    - Choose *Complete MiKTeX*
    - Choose *Anyone who uses this computer* (all users)
    - Install MiKTeX to C:/Program Files/MikTeX 2.9 (default)
    - Change *Preferred paper* to Letter
    - Set *Install missing packages on the fly* to YES
    - Start the installation
    - Click *Finish* when done

---

[4] See LaTeX and Friends by M.R.C. Van Dongen, Springer 2012.

[5] See Dev Community.

4. Install the TeXstudio
    - In the window that opened after running *setup.exe* click the second install box.
    - Use all Defaults

Choice of Editors to use with LaTeX/TeXLive

As has been described, in order to use TeX, LaTeX, TeXLive etc., one must install an appropriate editor. There are many options available: free, freeware or commercial. Then there are some editors that need to be installed while others can be used online. The best editor will be the one that meets the typesetting requirements for the current project. Many people pick an editor or two and use those for the all their projects. We will start by looking at the free editors that are available. Our focus will be on using LaTeX on Windows.

Free Editors

This list will describe the free editors that are available to use. For each of these, we will examine the features in the editor that would be useful for editing projects.

1. **TeXWorks** – TeXworks is free and open-source application software available for Windows, Linux and macOS. The software is licensed under the GPL. This is designed to be a simple editor available for all major operating systems (MS Windows, Gnu/Linux distros and MacOS). The following are some of the features available:
    1. Scripting
        - QtScript, Lua and Python languages are supported
        - use scripts to add new features or extend existing functionality
        - assign custom shortcuts for quick access
        - customizable access to other files and programs on your computer
        - use some of the bundled scripts to e.g., set the spellchecker language based on babel options or tune the lengthy console output into a concise list of TeX errors and warnings
    2. New features
        - use Follow focus to keep the source and preview in sync all the time
        - window positions and sizes are saved for recent documents
        - a command line parser allows for greater interoperability with other programs
        - resources (templates, scripts, …) are updated automatically when upgrading to a post-0.4 version
        - use one of numerous file encodings and line ending conventions to increase interoperability between different systems
        - save all open documents using this single menu item
        - auto-completion for the beamer class
    3. Major improvements & bug fixes

    - implement fine-grained synchronization using text searching to assist SyncTeX
    - support background color/font flags in syntax highlighting
    - show the spellchecker languages in human-readable form; no more ISO language codes, no more entries for the same language on unix platforms
    - preserve document view when reloading after external changes
    - switched to pdfLaTeX as the default engine on fresh installation
    - allow smart quotes to be applied to a selection
    - allow Esc and Return in the tags and search result windows
    - fix handling of external file links in PDFs
    - Fix handling of *All Files* in the *Save As* dialog on MS Windows

2. **TeXstudio** – This is an open-source option that is available for all major operating systems. This started as a fork of TeXmaker in 2009. It was originally called TeXmakerX with extensions for TeXmaker. Since them, significant changes to the features and code base have result in a fully independent program. TeXstudio has the following features:
    1. Comfortable editing
        - Multi-Cursors
        - Auto-Completion
        - More than 1000 mathematical symbols
        - Bookmarks
        - Link Overlay
        - Assistants for tables, images, formula, …
        - Drag & Drop support for images
        - Table formatting
    2. Keep the overview
        - structure view
        - Code folding
        - Advanced syntax highlighting
        - Interactive spellchecker
        - Interactive grammar checker
        - Interactive reference checker
        - Clear display of LaTeX errors and warnings (in the editor and as a list)
    3. Advanced Build System
        - Built-in support for various LaTeX compilers, index, bibliography and glossary tools, Latexmk, and many more
        - Automatic detection of the need for multiple LaTeX runs
        - Run any program you like
        - Completely customizable for the creation of the complete document
    4. View the Result
        - Integrated PDF viewer with (almost) word-level syncing
        - Live-updating inline preview for formulas and code segments

  - Tooltip preview for included images

5. Easy Setup

  - Installers or packages available for Windows, Linux, and Mac OS X
  - Portable USB version available
  - Automatic detection of MikTeX, TeX Live, Ghostscript and Standardlatex
  - SVN support

3. **TeXmaker** – This is a free cross-platform (Windows, MacOSX, Linux) LaTeX editor that started in 2003. It is easy to use and configure and released under a GNU Public License (GPL). It has the following features:

    - Unicode Editor – fully unicode and supports a large variety of encodings
    - Spellchecker – includes spell checking while typing
    - Code folding – all part, chapter, section, etc., can be collapsed
    - Code completion – main latex commands can be quickly inserted while typing
    - Fast navigation – TeXmaker includes a *structure view* which is automatically updated while typing.
    - Master mode – TeXmaker allows you to work easily onto documents separated in several files with the *master mode*.
    - Integrated PDF viewer – TeXmaker includes a built-in pdf viewer with continuous scrolling and synctex support.
    - Easy compilation – One-click compilation with the predefined *Quick Build* commands.
    - Mathematical symbols – 370 mathematical symbols can be inserted in just one click.
    - Wizards – TeXmaker includes wizards to generate the most standard LaTeX code (*Quick Document*, *Quick Beamer Presentation*, *Quick letter*, *tabular*, *tabbing and array environments*)
    - Latex documentation – an extensive LaTeX documentation is furnished with TeXmaker.
    - Error handling – TeXmaker automatically locates errors and warnings detected in the log file after a compilation and you can reach the corresponding lines in the document in one-click.
    - Rectangular block selection – easy rectangular selection with mouse +Alt key. Users can easily cut/copy/paste columns of a table.
    - Find in folders – you can search for text in all the latex documents included in a folder (and subfolders) . If you click on a line, TeXmaker will open the corresponding document at the right line.
    - Rotation mode for pdf viewer – the integrated pdf viewer supports rotation mode.
    - Regular expressions support – you can use regular expressions for search and replace.
    - Full asymptote support – syntax highlighting (commands) compilation in one click.
    - Unlimited number of snippets – users can define an unlimited number of snippets with keyboard triggers.

4. **TeXnicCenter** – This editor is licensed under GPL. The following are the features available in the editor.

        - User Interface
            - LaTeX code snippets – Simple insertion of LaTeX constructs via toolbars and

    menus
  - Tabbed MDI window – Open as many files as needed. Each open file is represented by a tab.
  - Customizable – All toolbars, menus and short cuts are customizable.
  - Customizable window layout – All panes can be docked to any window edge or undocked if required.
  - Integratable tools – Integrate any third party tools into TeXnicCenter's tools menu
  - Multi-language – Out of the box support for English and German
- Text Editor
  - Syntax highlighting – Highlighting of LaTeX constructs with customizable colors.
  - Auto completion – Simply hit [Ctrl]+[Space] to complete the command you've just started to type.
  - Paranthesis matching – Highlighting of missing and matching parenthesis.
  - Dynamic word wrapping – Wrap long lines at editor's edge or defined text column while typing.
  - Spell checker – Integrated spell checker including check while you type.
  - Forward and inverse search – Support for forward and inverse search in conjunction with any viewer supporting these features
- File Handling
  - Template based file creation – Comfortable feature for creating new projects or single files based on your own templates.
  - Plug-In interface for third party wizards – The Microsoft COM-interface provides the possibility to implement document and project wizards.
  - Document projects – Document projects keep all files of a document together and ease your daily work.
- Navigator View
  - Structure view – Browse the complete document structure grouped by the logical header levels.
  - Object view – Browse the list of all your document's headers, figures, tables, equations and other environments.
  - File view – Browse all your project's files grouped by type (LaTeX, Bibtex, Graphic files).
  - Reference view – Browse all your Project's BibTeX references.
- Building & Viewing Output
  - Concept of output profiles – Output profiles define the steps to be executed to compile your LaTeX files to the resulting output file (e.g., DVI, PDF) and how to view it.
  - Unlimited number of output profiles – Define as many output profiles as needed. Simply select the one to be used from the toolbar.
  - Flexible post processor mechanism – Define additional commands to be executed during the build process per output profile (e.g., dvips).
  - Flexible viewer integration – Define for each output profile how to invoke the viewer either by command line or DDE commands. This supports the integration of nearly every viewer including Adobe Reader and Ghostview.
  - Just one key press – Building or viewing your document's output requires just

    one key press or menu selection.
    - Error Detection – Errors, warnings and box errors in LaTeX's log output are highlighted and you can simply jump to the relevant text location.
    - Import/Export of output profiles – Share your output profiles with other TeXnicCenter users.
- Setup
    - Easy Installation – TeXnicCenter comes with a standard Windows installer providing useful default. Installation is finished within a few minutes.
    - Easy Configuration – The configuration wizard just queries for the path of your LaTeX installation and then you are done.

Freeware Editor

The following is a **shareware** LaTeX editor for use with MiKTeX installed on Windows 10.

WinEdt

WinEdt is a front-end Integrated Development Environment (IDE) for compilers and typesetting systems such as TeX, HTML of NSIS. It's highlighting schemes can be customised for different modes and its spell checking functionality supports multilingual setups with dictionaries for many languages available on WinEdt's Community Site. WinEdt may be used for an evaluation period of 31 days. Any further use requires a license, obtained through a Registration Procedure. The editor comes with a QuickGuide available from the WinEdt website. A few of its many features are:

- Tree view can be customised
- Automatically display your current location in TOC.
- Colour coding that aids usability.
- Intelligent defaults,
- Options to customise almost anything.
- Easy configuration interface with MiKTeX.
- One click build process for LaTeX documents.
- Intuitive default shortcut keys and intuitive alt menu letters.

The following video describes the workflow that can be used to create documents with WinEdt.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=28

Download text transcript: WinEdt-share-video2-transcript Video length: 5:23 minutes

Commercial Editors

The following is a list of commercial editors and their respective features.

1. **Overleaf** – Overleaf is an easy to use cloud-based LaTeX editor. It allows you to view document history, insert images, equations, bibliographies and more. It's feature list includes:
    - Overleaf offers a wide range of templates.
    - It makes writing, editing and publishing documents quicker.
    - This application offers a real-time preview of code.
    - Share withe other people effortlessly.
    - You can switch to LaTeX and Rich Text mode.
    - Find LaTeX errors in less time.
    - You can track changes and comments in real-time.
2. **Authorea** – Authorea is another online editor which allows you to write and publish easily. It supports commenting to discuss changes in real-time. It has the following features:
    - It helps you manage versions of documents without any hassle.
    - Authorea helps you start developing using templates.
    - This app provides 24×7 hour support.
    - You can convert documents from PDF to Word.
    - The tool can add rich media to documents.
    - It offers collaboration in real time.
    - The supported platform is the web.
3. **Archimedes** – Archimedes is a LaTeX and Markdown editor built from the ground up or the Mac and designed to make mathematical writing fun and easy. It has the following features
    - Markdown Editing – It is a fully-featured plain text and Markdown editor. It includes a fast syntax highlighter and provides convenient keyboard shortcuts for common actions, such as inserting images and links.
    - Math Mode + Autocomplete – Archimedes lest you write mathematics in a subset of LaTeX. Just enter $$ to get started. The editor autocompletes commands and intelligently matches closing braces, brackets and parentheses as you type.
    - Live Preview – As you work, a beautifully-typeset live preview of your document is always visible. You can even switch between horizontal or vertical orientation and select a custom theme.
    - Math Library – Browse the math library to see all available LaTeX commands and their previews in one place.
    - Macros – You can define macros for expressions you work with the most. The macros show up in autocompletion results alongside built-in completions.

Basic Document Layout in LaTeX

Leslie Lamport wrote LaTeX as an extension of Donald Knuth's TeX program.[6] LaTeX consists of a Turing-complete procedural markup language and a typesetting processor. This combination allows one to control both the visual presentation and the content of a document. The following steps outline the use of LaTeX.

---

[6] See LaTeX and Friends by M.R.C. Van Dongen, Springer 2012.

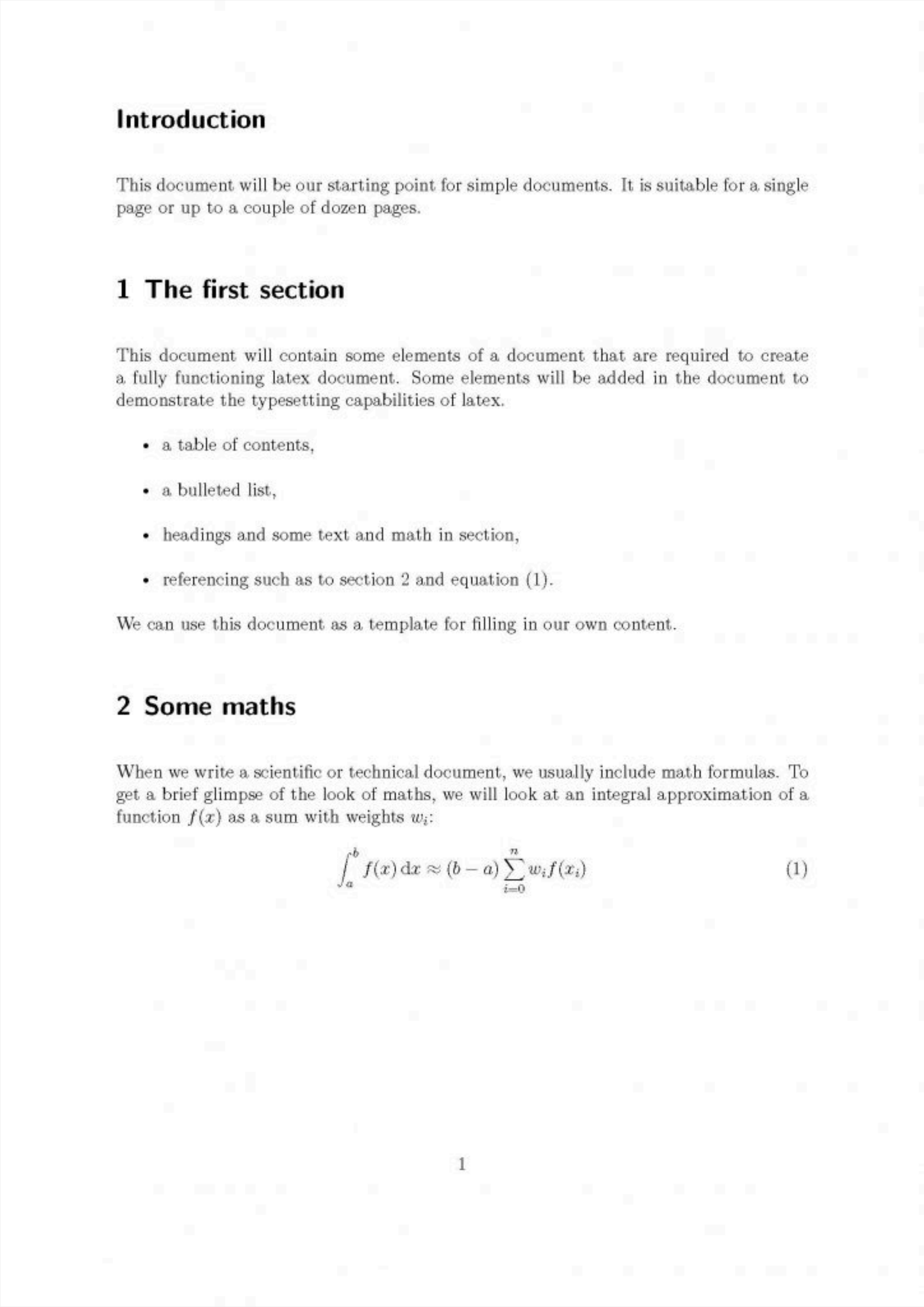
1. A document is written in a LaTeX (.tex) input (source) file. This a human readable file containing a combination of commands, text and special characters.
2. A typesetting processor (latex, pdflatex, xetex, xelatex etc.,) is run on the input file. This converts the input file into a device independent (.dvi) file or a portable document format (.pdf) file. if the processors flags errors in the input file, they should be fixed before moving on to step 3.
3. The dvi of pdf file can then be viewed on the computer. For printing, the dvi file must be converted to either postscript or pdf format.

In general, working with editors makes these steps seamless and easier to manage. These editors also make finding and fixing errors much easier in many instance. You are also able to preview the processed document within most editors. Some even have a *live preview* which automatically generates the preview once the processing is complete.

The following steps lead to the creation of a basic LaTeX document.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=28#h5p-8

This document can be run using any LaTeX editor to show the typeset result.



Introduction

This document will be our starting point for simple documents. It is suitable for a single page or up to a couple of dozen pages.

1 The first section

This document will contain some elements of a document that are required to create a fully functioning latex document. Some elements will be added in the document to demonstrate the typesetting capabilities of latex.

- a table of contents,
- a bulleted list,
- headings and some text and math in section,
- referencing such as to section 2 and equation (1).

We can use this document as a template for filling in our own content.

2 Some maths

When we write a scientific or technical document, we usually include math formulas. To get a brief glimpse of the look of maths, we will look at an integral approximation of a function $f(x)$ as a sum with weights w_i:

$$\int_a^b f(x)\,\mathrm{d}x \approx (b-a)\sum_{i=0}^{n} w_i f(x_i) \qquad (1)$$

1

Figure 4. Visual result of typesetting the LaTeX document

A manner of documents can be created using LaTeX and it associated editors. The TUG website maintains a catalogue of all packages currently available for use in LaTeX documents. Every aspect of a document can be automated using these packages:

- creating a book cover
- cover page for reports, articles
- classes covering a range of different document types including those for masters and doctoral theses
- many different classes for creating books and manuals
- commands to create tables of contents, lists of figures, list of tables
- packages that cover a range of different mathematical needs with special symbols and formulae
- mechanism for creating reference, bibliographies, and index, glossary, list of acronyms, etc.
- there are several packages for creating graphics within LaTeX documents, i.e., PGF/Tikz, PSTricks, Asymptote, etc. These allow the creation of graphics using macros.
- convertors to allow a LaTeX document to be provided in multiple formats, i.e., postscript, PDF, HTML, ePUB, etc.

As can be seen from this, LaTeX is a self-contained publishing environment that is used to create complete documents without the need for any external graphics programs.

Summary

In this chapter, the open-source document creation platform LaTeX has been briefly described. Its connection to Donald Knuth's TeX was mentioned and the different variants of the software for different platforms described. LaTeX's power is the ability to select as many or as few packages as needed to create a given document.

To help work with documents, there are a number of editors with built-in tools that make working with LaTeX documents straight-forward. Tools such as conversion to PDF and viewing the result is provided by most editors, for example. These editors also provide helps in the form of provide command completion, menus with mathematical symbols, buttons for index creation, etc.

One of the key aspects of using LaTeX is the ability to tap into a global group of users in document creation. There are many different TeX User Groups across the globe that one can turn to for help. There are also a number of community forums that can also be leveraged (LaTeX Community, TeXample.net, LaTeX Project, LaTeX Stack Exchange, etc.).

6. Open-source Document Creation Tools II

ISAAC MULOLANI

Learning Outcomes

By the end of this chapter, the reader will be able to:

1. Identify latex packages useful for the creation of OER across subject areas.
2. Describe the inclusion of graphical elements into a latex document
3. Describe how to create content rich with mathematical and algorithmic elements
4. Construct a basic presentation using latex package elements

LaTeX Packages useful for the Creation of OER

In the previous chapter, the LaTeX typesetting platform was discussed along with some editors. As was mentioned in the chapter, most of these editors simplify the workflow required to create documents. Our interest in this section is to identify packages across subject areas that are useful for the creation of open resources.

One of the most basic type of packages would be those for creating books or manuals. Along with this are those for creating content in specific areas such as books containing significant amounts of equations and mathematics.

Class name	TUG location	Description
report	https://ctan.org/pkg/report	The class provides typesetting rather similar to that of the book class, omitting some features relevant only when professional book-publishing is to be used.
article	https://ctan.org/pkg/article	A ‘basic’ class for articles in LaTeX; it is part of the LaTeX distribution.
scrartcl – Koma-Script article class	https://ctan.org/pkg/scrartcl	The class provides the “article”-like element of the koma-script collection. The document layout of the class is less ‘strident’ than that of article, and it offers much more flexibility than article via other elements of the koma-script collection.
scrbook – Komas-Script book class	https://ctan.org/pkg/scrbook	The class provides the “book”-like element of the koma-script collection. The document layout of the class is more ‘calm’ than that of book, and it offers much more flexibility than book via other elements of the koma-script collection.
octavo	https://ctan.org/pkg/octavo	The octavo class is a modification of the standard LaTeX book class. Its purpose is to typeset books following classical design and layout principles, with the express intention of encouraging the making of beautiful books by anyone with access to a good printer and with an inclination towards venerable crafts, e.g., bookbinding.
memoir	https://ctan.org/pkg/memoir	The memoir class is for typesetting poetry, fiction, non-fiction, and mathematical works. Permissible document ‘base’ font sizes range from 9 to 60pt. There is a range of page-styles and well over a dozen chapter-styles to choose from, as well as methods for specifying your own layouts and designs. The class also provides the functionality of over thirty of the more popular packages, thus simplifying document sources.
ycbook	https://ctan.org/pkg/ycbook	This class is intended to be an interpretation of the mwbk class which is a part of the mwcls package. The mwcls classes are simple, yet powerful and customizable classes that allow the end-user to customize the layout of headers, headings, etc. They also have the benefit of being more economic in space than the most common LaTeX classes, while keeping a clear appearance and a smooth flow.
willowtreebook	https://ctan.org/pkg/willowtreebook	The willowtreebook class is a simple book class, which author Benjamin McKay uses for his lecture notes. It actually just selects options for the more sophisticated memoir class.
ElegantBook	https://ctan.org/pkg/elegantbook and https://elegantlatex.org/en/	ElegantBook is designed for writing Books. This template is based on the standard LaTeX book class. The goal of this template is to make the writing process more elegant. Just enjoy it!
Tufte-latex	https://ctan.org/pkg/tufte-latex	Provided are two classes inspired, respectively, by handouts and books created by Edward Tufte.
ElegantNote	https://ctan.org/pkg/elegantnote	ElegantNote is designed for writing working papers, especially for economics students. This template is based on the standard LaTeX article class. The goal of this template is to make the writing process easier and more comfortable.
amsbook	https://ctan.org/pkg/amsbook	A LaTeX document class for books that is tailored to the design of American Mathematical Society publications. If you have a standard installation of LaTeX, you should find that amsbook is already present, since it is part of the (“required”) AMS-LaTeX distribution. For publication-specific AMS document classes (surv-l.cls, memo-l.cls, mmono-l.cls, gsm-l.cls, stml-l.cls, etc.), see the AMS author information.
Memdesign class	https://ctan.org/pkg/memdesign	“A Few Notes on Book Design” provides an introduction to the business of book design. It is an extended version of what used to be the first part of the memoir users’ manual.

Table 6.1 Table showing some classes useful for creating books/manuals

To use these classes/packages in LaTeX, one must first check to make sure they are installed. The TUG repository is updated on a regular basis and any local installation of MiKTeX/MacTeX/TeXLive needs to be regularly updated as well. For each class and package documentation is provided detailing usage. In some cases, there are examples of usage provided as part of the documentation.

Besides these classes, there are author defined classes and packages useful for specific purposes. This of course requires the author to have an understanding of how to define a class/package. As open-source users usually do, existing class or package files can be used to customise ones own class or package. The advantage of this is that a user can add personal macros for each application.

Next is a set of packages that can be used to create presentations. Note this list is not exhaustive as there are many users who create custom packages for their own use. Many of these on further refinement are added to the TUG package repository.

Package name	**TUG Location**	**Description**
Prosper	https://ctan.org/pkg/prosper	Prosper is a LaTeX class for writing transparencies. It is written as an extension of the seminar class by Timothy Van Zandt. Prosper offers a friendly environment for creating slides for both presentations with an overhead projector and a video projector. Slides prepared for a presentation with a computer and a video projector may integrate animation effects, incremental display, and so on. Various visual styles are supported (including some that mimic PowerPoint) and others are being contributed.
ffslides	https://ctan.org/pkg/ffslides	The ffslides ("freeform slides") class is intended to make it easier to place various types of content freely on the page, and therefore easier to design documents with a strong visual component: presentations, posters, research or lecture notes, and so on.
gridslides	https://ctan.org/pkg/gridslides	This package allows creating free form slides with blocks placed on a grid. The blocks can be filled with text, equations, figures etc. The resulting slides are similar to the ones produced with LaTeX beamer, but more flexible.
fancyslides	https://ctan.org/pkg/fancyslides	This class is prepared for short presentations with a modern look & feel. It offers the following features: • custom background for each slide, • predefined types of slides, • simplified commands (e.g. for starting and ending slide).
beamer	https://ctan.org/pkg/beamer	The beamer LaTeX class can be used for producing slides. The class works in both PostScript and direct PDF output modes, using the pgf graphics system for visual effects. Content is created in the frame environment, and each frame can be made up of a number of slides using a simple notation for specifying material to appear on each slide within a frame. Short versions of title, authors, institute can also be specified as optional parameters. Whole frame graphics are supported by plain frames. The class supports figure and table environments, transparency effects, varying slide transitions and animations. Beamer also provides compatibility with other packages like prosper.
beamerthemetamu	https://ctan.org/pkg/beamerthemetamu	This is a beamer theme designed for use in Texas A&M University (TAMU). It can be used to create slides with TAMU brand colors.
Beamertheme-detlevcm class	https://ctan.org/pkg/beamertheme-detlevcm	A beamer theme designed for use in the University of Leeds. The bundle provides a simple theme that has been used in the author’s department.
Beamer-verona	https://ctan.org/pkg/beamer-verona	This package provides the ‘Verona’ theme for the beamer class by Till Tantau.
Beamertheme-metropolis	https://ctan.org/pkg/beamertheme-metropolis	The package provides a simple, modern Beamer theme for anyone to use. It tries to minimize noise and maximize space for content.
Beamer2thesis	https://ctan.org/pkg/beamer2thesis	The package specifies a beamer theme for presenting a thesis.
Beamertheme-trigon	https://ctan.org/pkg/beamertheme-trigon	This package provides a modern, elegant and versatile theme for Beamer, with a high degree of customization. Trigon found its origin and inspiration in the graphical guidelines resulting from the visual identity overhaul of the University of Liège. Although directly inspired from these guidelines, the theme was stripped out of any mention or specificities related to the University and its faculties. This makes the Trigon theme perfectly suitable for many different contexts.

Table 6.2 Table showing packages useful for creating presentations in LaTeX

The following table provides a list of packages that can be used to create exercise sheets across different subject areas. These can be combined with any of the previous packages and classes.

Packages	TUG Location	Description
exercises	https://ctan.org/pkg/exercises	This package defines the environments exercise and solution. The layout of these environments can be customized. The — optional — points in the exercises can be added automatically. The package also permits to hide the solutions.
Brandeis-problemset	https://ctan.org/pkg/brandeis-problemset	Brandeis University's computer science ("COSI") courses often assign "problem sets" which require fairly rigorous formatting. This document class, which extends article, provides a simple way to typeset these problem sets in LaTeX.
pbsheet	https://ctan.org/pkg/pbsheet	This class is designed to simplify the typesetting of problem sheets with Mathematics and Computer Science content. It is currently customised towards teaching in French (and the examples are in French).
answers	https://ctan.org/pkg/answers	The package allows a lot of flexibility in constructing question and answer sheets.
worksheet	https://ctan.org/pkg/worksheet	This package provides macros and an environment for easy worksheet creation: Use the exercise environment for formating exercises in a simple, efficient design; typeset customized and automatically numbered worksheet titles in the same way as standard LaTeX titles (using \maketitle); provide course and author information with a scrlayer-scrpage based automated header; conforming to different babel languages. (Currently English, French, and German are supported.)
exerquiz	https://ctan.org/pkg/exerquiz	This package defines three new environments for defining exercises and quizzes as PDF files. The solutions to the exercises are hyperlinked to the questions. The quizzes are graded and may optionally be corrected by JavaScript. The package is part of the AcroTeX bundle.
acrotex	https://ctan.org/pkg/acrotex	The bundle contains: • the web package to redefine page layout to web-friendly dimensions; • the exerquiz package for defining on-line exercises and quizzes of various sorts; • the eForms package for support of PDF forms; • the insdljs package for inserting document-level JavaScript in LaTeX documents; • the dljslib library of JavaScript functions for use with exerquiz; and • the eq2db package for converting an exerquiz quiz for processing by a ASP server-side script.
eqexam	https://ctan.org/pkg/eqexam	eqExam is a LaTeX package for writing exams, tests, quizzes, homework assignments, etc. It is a stand alone package, yet is tightly integrated with the AcroTeX eDucation Bundle (AeB).
exam	https://ctan.org/pkg/exam	Provides a class exam.cls, which eases production of exams, even by a LaTeX novice. Simple commands are provided to: • create questions, parts of questions, subparts of parts, and subsubparts of subparts, all with optional point values; • create a grading table, indexed either by question number (listing each question and the total possible points for that question) or by page number (listing each page with points and the total possible points for that page); • create headers and footers that are each specified in three parts: one part to be left justified, one part to be centered, and one part to be right justified, in the manner of fancyhdr.
Exam-n	https://ctan.org/pkg/exam-n	The class design offers: • Direct support for collaborative development of an exam, using a model in which a departmental 'exams convener' or 'exam chair' coordinates multiple authors writing individual questions (the class file and associated process is in regular use within a physics and astronomy department). • All of the 'traditional' exam paper features such as sectioning, per-part running marks, 'Question n continued' catchwords, and so on. • Readily configured local adaptation.

Packages	TUG Location	Description
exsheets	https://ctan.org/pkg/exsheets	The package provides the means to create exercises or questions and their corresponding solutions. The questions may be divided into classes and/or topics and may be printed selectively. Meta-data to questions can be added and recovered. The solutions may be printed where they are, or collected and printed at a later point in the document all together, section-wise or selectively by ID. The package provides the means to selectively include questions from an external file, and to control the style of headings of both questions and solutions.
xsim	https://ctan.org/pkg/xsim	This package helps in creating exercises and the corresponding solutions. It is the official successor of the exsheets package and fixes/improves various long-standing issues.
exesheet	https://ctan.org/pkg/exesheet	This package is used for typesetting exercise or exam sheets. In addition, the exesheet class loads the schooldocs package. The package provides: • macros to mark out exercises and subparts, • specific settings for enumeration lists, • environments for questions and answers, with conditional display, • macros for marking schemes with detailed comments.
exframe	https://ctan.org/pkg/exframe	This LaTeX2ε package provides a general purpose framework to describe and typeset exercises and exam questions along with their solutions. The package features mechanisms to hide or postpone solutions, to assign and handle points, to collect problems on exercise sheets, to store and use metadata, and to implement a consistent numbering. It also provides a very flexible interface for configuring and customising the formatting, layout, and representation of the exercise content.
exercisebank	https://ctan.org/pkg/exercisebank	This package makes it easier to maintain and edit your exercise sets. Exercises are saved as separate files containing part problems. These files can be used to make sets, and you can cherry-pick or exclude certain part problems as you see fit.
schooldocs	https://ctan.org/pkg/schooldocs	The purpose of this package is to provide several layout styles for school documents. It is useful for exercise sheets, exams, course materials. The package sets the page geometry (dimensions of text and margins) and the title typesetting; the various styles define the header, footer and title formatting. Many features are freely configurable.
probsoln	https://ctan.org/pkg/probsoln	The package is designed for lecturers who have to generate new problem sheets for their students on a regular basis (e.g. yearly) by randomly selecting a specified number of problems defined in another file. The package allows you easily to generate a new problem sheet that is different from the previous year, thus alleviating the temptation of students to seek out the previous year's students and checking out their answers. The solutions to the problems can be defined along with the problem, making it easy to generate the solution sheet from the same source code; problems may be reused within a document, so that solutions may appear in a different section of the same document as the problems they cover.

Table 6.3 Table showing packages useful for creating exercises/problem sheets

Graphical Elements in LaTeX Documents

One of the most important aspects of LaTeX documents is the ability to include graphical elements. There are several packages that allow for the inclusion of these graphical elements directly in a LaTeX document. When the LaTeX processor is run, it renders the graphical code element as the required image. Open-source users love this feature of TeX-based open projects because they can *describe* a graphical element use source code which is rendered as an image by the processor. This bypasses the need for external drawing programs and image repositories. The following table lists some of those packages.

Package name	TUG Location	Description
PSTricks	https://tug.org/PSTricks/main.cgi/	PSTricks offers an extensive collection of macros for generating PostScript that is usable with most TeX macro formats, including Plain TeX, LaTeX, AMS-TeX, and AMS-LaTeX. Included are macros for colour, graphics, pie charts, rotation, trees and overlays. It has many special features, including a wide variety of graphics (picture drawing) macros, with a flexible interface and with colour support. There are macros for colouring or shading the cells of tables.
pgf	https://www.ctan.org/pkg/pgf	PGF is a macro package for creating graphics. It is platform- and format-independent and works together with the most important TeX backend drivers, including pdfTeX and dvips. It comes with a user-friendly syntax layer called TikZ. Its usage is similar to pstricks and the standard picture environment. PGF works with plain (pdf-)TeX, (pdf-)LaTeX, and ConTeXt. Unlike pstricks, it can produce either PostScript or PDF output.
tikz-3dplot	https://ctan.org/pkg/tikz-3dplot	The package provides straightforward ways to define three-dimensional coordinate frames through which to plot in TikZ. The user can specify the orientation of the main coordinate frame, and use standard TikZ commands and coordinates to render their tikzfigure. A secondary coordinate frame is provided to allow rotations and translations with respect to the main coordinate frame. In addition, the package can also handle plotting user-specified functions in spherical polar coordinates, where both the radius and fill color can be expressed as parametric functions of polar angles.
Tikz-optics	https://ctan.org/pkg/tikz-optics	This package provides a new TikZ library designed to easily draw optical setups with TikZ. It provides shapes for lens, mirror, etc. The geometrically (in)correct computation of light rays through the setup is left to the user.
Tikz-planets	https://ctan.org/pkg/tikz-planets	This TikZ-package makes it easy to illustrate celestial mechanics and the solar system. You can use it to draw sketches of the eclipses, the phases of the Moon, etc.
worldflags	https://ctan.org/pkg/worldflags	This is a package for drawing flags using TikZ. Currently the national flags of all independent nations are included, additionally some other flags of various organizations. A flag can be drawn … • as a single TikZ-picture within ordinary text, • as a picture element within a TikZ-picture. The appearance of a flag (size, frame etc.) can be adapted using optional parameters.
smartdiagram	https://ctan.org/pkg/smartdiagram	The package will create 'smart' diagrams from lists of items, for simple documents and for presentations.
circuitikz	https://ctan.org/pkg/circuitikz	The package provides a set of macros for naturally typesetting electrical and (somewhat less naturally, perhaps) electronic networks. It is designed as a tool that is easy to use, with a lean syntax, native to LaTeX, and directly supporting PDF output format. It has therefore been based on the very impressive PGF/TikZ package.
Tikz-network	https://ctan.org/pkg/tikz-network	This package allows the creation of images of complex networks that are seamlessly integrated into the underlying LaTeX files. The package requires datatool, etex, graphicx, tikz, trimspaces, xifthen, and xkeyval.
quantikz	https://ctan.org/pkg/quantikz	The purpose of this package is to extend TikZ with the functionality for drawing quantum circuit diagrams.
ticollege	https://ctan.org/pkg/ticollege	This package provides commands to draw scientific calculator keys with the help of TikZ. It also provides commands to draw the content of screens and of menu items.
logicpuzzle	https://ctan.org/pkg/logicpuzzle	The package allows the user to typeset various logic puzzles. At the moment the package supports the creation of puzzles for 2D-Sudoku, Battleship and 26 other puzzles.
tikzpeople	https://ctan.org/pkg/tikzpeople	This package provides people-shaped nodes in the style of Microsoft Visio clip art, to be used with TikZ. The available, highly customizable, node shapes are: alice, bob, bride, builder, businessman, charlie, chef, conductor, cowboy, criminal, dave, devil, duck, graduate, groom, guard, jester, judge, maininblack, mexican, nun, nurse, physician, pilot, police, priest, sailor, santa, surgeon.
Tikz-euclide	https://ctan.org/pkg/tkz-euclide	The tkz-euclide package is a set of files designed to give math teachers and students easy access to the programming of Euclidean geometry with TikZ.

Package name	TUG Location	Description
Tikz-fct	https://ctan.org/pkg/tkz-fct	The tkz-fct package is designed to give math teachers (and students) easy access to programming graphs of functions with TikZ and gnuplot.
Tikz-graph	https://ctan.org/pkg/tkz-graph	The package is designed to create graph diagrams as simply as possible, using TikZ.
tzplot	https://ctan.org/pkg/tzplot	This is a LaTeX package that provides TikZ-based macros to make it easy to draw graphs. The macros provided in this package are just abbreviations for TikZ codes, which can be complicated; but using the package will hopefully make drawing easier, especially when drawing repeatedly. The macros were chosen and developed with an emphasis on drawing graphs in economics.
Circuit-macros	https://ctan.org/pkg/circuit-macros	A set of m4 macros for drawing high-quality electric circuits containing fundamental elements, amplifiers, transistors, and basic logic gates to include in TeX, LaTeX, or similar documents. Some tools and examples for other types of diagrams are also included. The macros can be evaluated to drawing commands in the pic language, which is very easy to understand and which has a good power/complexity ratio.
chemfig	https://ctan.org/pkg/chemfig	The package provides the command chemfig{}, which draws molecules using the TikZ package. The argument provides instructions for the drawing operation. While the diagrams produced are essentially 2-dimensional, the package supports many of the conventional notations for illustrating the 3-dimensional layout of a molecule.
euclidean geometry	https://ctan.org/pkg/euclideangeometry	This package provides tools to draw most of the geometrical constructions that a high school instructor or bachelor degree professor might need to teach geometry. The connection to Euclid depends on the fact that in his times calculations were made with ruler, compass and also with ellipsograph.
Asymptote	https://ctan.org/pkg/asymptote	Asymptote is a powerful descriptive vector graphics language for technical drawing, inspired by METAPOST but with an improved C++-like syntax. Asymptote provides for figures the same high-quality level of typesetting that LaTeX does for scientific text.
ePiX	https://ctan.org/pkg/epix	ePiX is a collection of batch-oriented utilities for Unix-like systems. It creates mathematically accurate line figures, plots, and movies using easy-to-learn syntax. LaTeX and dvips comprise the typographical rendering engine, while ImageMagick is used to create bitmapped images and animations.
Eukleides	https://ctan.org/pkg/eukleides	Eukleides is a geometry drawing language. The Eukleides distribution provides two interpreters that produce Encapsulated PostScript and TeXable PSTricks macros. The distribution also provides a package allowing to use Eukleides code within a LaTeX source file. Eukleides may be compiled under MinGW, and a win32 binary is also available.
metapost	https://ctan.org/pkg/metapost	METAPOST uses a language based on that of METAFONT to produce precise technical illustrations. Its output is scalable PostScript or SVG, rather than the bitmaps METAFONT creates.
Pgf-pie	https://ctan.org/pkg/pgf-pie	The package provides the means to draw pie (and variant) charts, using PGF/TikZ. The license of the package is MIT as mentioned here: https://code.google.com/archive/p/pgf-pie/.
pgfplots	https://ctan.org/pkg/pgfplots	PGFPlots draws high-quality function plots in normal or logarithmic scaling with a user-friendly interface directly in TeX. The user supplies axis labels, legend entries and the plot coordinates for one or more plots and PGFPlots applies axis scaling, computes any logarithms and axis ticks and draws the plots, supporting line plots, scatter plots, piecewise constant plots, bar plots, area plots, mesh-- and surface plots and some more. Pgfplots is based on PGF/TikZ (PGF); it runs equally for LaTeX/TeX/ConTeXt.

Table 6.4 Table of packages for creating graphical elements

Typesetting Graphical Elements in LaTeX Documents

Given all the packages available for creating graphics in LaTeX, a logical questions is how these packages are used in a LaTeX document. The best way to describe this is to provide a demonstration of the process in action. In LaTeX documents, there are several ways to include graphical elements:

1. use a graphical package to create the figures within the document
2. include the figures from a separate LaTeX file containing the commands to create the graphical elements
3. run LaTeX files with only graphics commands to create image file formats (.PDF, .JPG, .PNG, etc.) and then include these in the main file

These are the most common ways most LaTeX users work with graphics. They are by no means the only ways to include graphics in files. TeX-based documents can include graphical elements of great complexity as can be seen by looking at examples on the PSTricks and TeXample.net pages.

The following H5P shows images created using LaTeX source code. The two specific graphics packages demonstrated are TikZ and PStricks.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=30#h5p-25

Typesetting Equations, Mathematical and Algorithmic Elements

For many areas in Science Technology Engineering and Mathematics (STEM), there is a need for extensive equations that are mathematical, physical and chemical. For areas such as Computer Science, there is a need to include algorithms within documents. In other areas such as Economics and Finance combinations of these are needed. LaTeX and TeX-based tools are able to incorporate these into documents very easily through the use of packages.

Mathematical Equations

In this section, a few samples of the type of equations that can be included in documents are shown. Within Pressbooks, to allow the use of LaTeX, one must enable the WP QuickLaTeX plugin. Then, at the start of each file where LaTeX constructs are to be used, one must use the command .

$$P(X = x) = \binom{n}{x} \cdot p^x \cdot (1 - p)^{n-x} \tag{1}$$

And another Statistics equation can be seen below.

$$
\begin{aligned}
(x+y)^0 &= 1\\
(x+y)^1 &= x+y\\
(x+y)^2 &= x^2+2xy+y^2\\
(x+y)^3 &= x^3+3x^2y+3xy^2+y^3\\
(x+y)^4 &= x^4+4x^3y+6x^2y^2+4xy^3+y^4\\
(x+y)^5 &= x^5+5x^4y+10x^3y^2+10x^2y^3+5xy^4+y^5\\
(x+y)^6 &= x^6+6x^5y+15x^4y^2+20x^3y^3+15x^2y^4+6xy^5+y^6
\end{aligned}
$$

This is an expansion of the Binomial Theorem. The following is another equation. For n a positive integer, we have:

$$
\begin{aligned}
(x+y)^n &= \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k\\
&= x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \binom{n}{3} x^{n-3} y^3 + \ldots\\
&+ \binom{n}{n-1} x y^{n-1} + y^n
\end{aligned}
$$

where the general term is $\binom{n}{r} x^{n-r} y^r$ and

$$\binom{n}{r} = \frac{n!}{(n-r)!r!} = \frac{n(n-1)(n-2)\ldots(n-r+1)}{r!}.$$

One of the challenges of using LaTeX in Pressbooks is that not all elements of TeX are supported. Not all packages are supported and elements such as macros cannot be included. However, when using the workflow mentioned in the previous chapters (i.e., a LaTeX installation with an editor) all the features are available.

Creating Presentations Using LaTeX Tools

For the purpose of this guide, the focus will be on the use of one particular presentation package for the sake of brevity. There are several packages that can be used to create presentations in LaTeX. The table below lists some of the packages available for creating presentations.

Package name	**Description**	**Location**
Beamer	The beamer LaTeX class can be used for producing slides. The class works in both PostScript and direct PDF output modes, using the pgf graphics system for visual effects.	https://ctan.org/pkg/beamer
powerdot	Powerdot is a presentation class for LaTeX that allows for the quick and easy development of professional presentations. It comes with many tools that enhance presentations and aid the presenter. Examples are automatic overlays, personal notes and a handout mode. To view a presentation, DVI, PS or PDF output can be used. A powerful template system is available to easily develop new styles. A LyX layout file is provided.	https://ctan.org/pkg/powerdot
HA-prosper	HA-prosper is a patch for prosper that adds new functionality to prosper based presentations. Among the new features you will find automatic generation of a table of contents on each slide, support for notes and portrait slides. The available styles demonstrate how to expand the functionality of prosper even further.	https://ctan.org/pkg/ha-prosper
Powerdot-fuberlin	The bundle provides a powerdot-derived class and a package for use with powerdot to provide the corporate design of the Free University in Berlin.	https://ctan.org/pkg/powerdot-fuberlin
Sslides	The class provides a variant of the LaTeX standard slides class, in which the user may add headers and footers to the slide.	https://ctan.org/pkg/sslides
uwmslide	A slide format which produces slides with a simple Power Point like appearance. Several useful features include: use of standard titlepage to produce title slide; several slide environments including plain (page with a title), double slide (two column page with slide title), item slide (item list with title), left item slide, and right item slide. Logos are placed in the upper left corner of each slide if the logo file logo.eps is present. Preconfigured in landscape mode by default and uses Times Roman by default (originally, it was claimed, for simple conversion to PDF format).	https://ctan.org/pkg/uwmslide
pdfslide	This is a package for use with pdfTeX, to make nice presentation slides. Its aims are: to devise a method for easier technical presentation; to help the mix of mathematical formulae with text and graphics which other present day document processing tools fail to accomplish; to exploit the platform independence of TeX so that presentation documents become portable; and to offer the freedom and possibilities of using various backgrounds and other embellishments that a user can imagine to have in as presentation.	https://ctan.org/pkg/pdfslide
ffslides	The ffslides ("freeform slides") class is intended to make it easier to place various types of content freely on the page, and therefore easier to design documents with a strong visual component: presentations, posters, research or lecture notes, and so on.	https://ctan.org/pkg/ffslides
Texpower	TeXPower is a bundle of packages intended to provide an all-inclusive environment for designing pdf screen presentations to be viewed in full-screen mode, especially for projecting `online' with a video beamer. For some of its core functions, it uses code derived from ppower4 packages.	https://ctan.org/pkg/texpower
gridslides	This package allows creating free form slides with blocks placed on a grid. The blocks can be filled with text, equations, figures etc. The resulting slides are similar to the ones produced with LaTeX beamer, but more flexible.	https://ctan.org/pkg/gridslides

fancyslides	This class is prepared for short presentations with a modern look & feel. It offers the following features: - custom background for each slide, - predefined types of slides, - simplified commands (e.g. for starting and ending slide). The class is built upon LaTeX beamer, so all beamer commands should work.	https://ctan.org/pkg/fancyslides
soton	The bundle contains two packages: soton-palette which defines colour-ways, and soton-beamer, which uses the colours to produce compliant presentations.	https://ctan.org/pkg/soton

Table 6.5 LaTeX packages for creating Presentations

The Beamer Presentation Package

The Beamer presentation package is installed as one of the default packages in LaTeX. Table 5.1 shows the location of the package including its documentation. To use any package in LaTeX, one must be willing to reference the appropriate documentation. From the documentation, one can find the list of options, features and examples of how to use the package to create specific document elements.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=30

Video length: 4:30 minutes

Frequently Asked Questions about TeX/LaTeX[1]

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=30#h5p-13

For further information, Appendix B contains a list of a few open resources created using TeX-based tools. In addition, Appendix C contains a list of LaTeX references the interested reader can consult.

---

[1] See UKTUG FAQ.

7. Open-source Document Creation Tools III

ISAAC MULOLANI

Learning Outcomes

By the end of this chapter, the reader will be able to:

1. Identify other open-source latex-based tools useful for creating OER
2. Describe the process of creating OER independent of the technology tool
3. Explain the process of making use of online Latex editors to create documents
4. Describe the more hybrid development tools available for creation of OER

Additional TeX-based Open-source tools

In this section, there will be an examination of some additional TeX-based tools that relate to the creation of open resources. These are in addition to those presented in the previous chapters. Hopefully, these will be a useful addition to the tools already described. The tools to be described are listed below:

- LyX
- Gnu TeXmacs
- Papeeria
- Overleaf
- R Markdown

It is useful to note that two of these tools do not require any installation since they are online tools: Overleaf and Papeeria. The focus of both of these is collaborative editing of resources online. The advantage of these online tools is that one avoids the need to install the required LaTeX distribution (MiKTeX, TeXLive or MacTeX). In addition, these online tools come with cloud storage for all project files. Another feature of the online editors is the ability to access many different document templates as starting points for documents. This greatly simplifies the use of different classes to something similar to word processing templates.

TeX-based Tools requiring Installation

In a previous chapter, the basic requirements of a TeX-based system have been discussed. For the most popular TeX distributions (MiKTeX, TeXLive, MacTeX), the basic requirement is the installation of the distribution and then an editor on top of that. Editors allow automation of many packages and functions. Unlike, word processors, these TeX-based tools are not WYSIWYG (What You See Is What You Get). Instead, LaTeX provides packages and commands to control the structure and layout of a document. Just about every LaTeX editor available provides a way for the user to preview a processed document – usually an iterative process in the workflow.

In this section, two specific TeX-based tools will be examined that are almost a bridge between traditional LaTeX typesetting and WYSIWYG. The first tool is LyX and the second is TeXmacs. Both tools are open-source options freely downloadable from their respective websites. Both programs run on all Linux/Unix systems, Windows, and Mac OS.

LyX Document Processor

The LyX document processor is released under a GNU General Public License, version 2 or later. LyX is a document processor that facilitates a writing approach that combines structure considerations with the appearance. It is more of a What You See Is What You Mean (WYSIWYM) approach. Combined in this tool is TeX/LaTeX's power and flexibility with the ease of a graphical user interface. The process supports the creation of mathematical content through its integrated equation editor along with other types of structured documents.

One point that must be made is that LyX requires a working LaTeX distribution of MiKTeX. For the Windows installers, a check is made during installation on whether a working TeX distribution exists. If not, the installer will automatically initiate a MiKTeX install first before installing LyX.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#oembed-1

Video length: 19:08 minutes

Features of LyX

Here are the features of LyX.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#h5p-11

GNU TeXmacs

GNU TeXmacs is provided under a GNU General Public License. This is another option that is similar to TeX-based systems described so far. TeXmacs has evolved from a scientific text editor into a scientific office suite, with an integrated presentation mode, technical drawing editor, versioning tools, bibliography tool, etc. The typesetting quality has continued to improve with a better support of microtypography and a large variety of fonts. The converters for LaTeX and HTML have also been further perfected and TeXmacs now comes with a native support for PDF.

It should be said that TeXmacs provides much more functionality than most TeX-based systems. The following are some of the options available

- GNU TeXmacs is a free scientific editing platform designed to create beautiful technical documents using a WYSIWYG interface.
- It provides a unified and user friendly framework for editing structured documents with different types of content: text, mathematics, graphics, interactive content, slides, etc.
- TeXmacs can be used as a graphical front-end for many systems in computer algebra (Sage, Maxima, Maple, Mathematica), numerical analysis, statistics, etc.
- Documents can be saved in TeXmacs, XML or Scheme format and printed as PDF or Postscript files. Converters exist for TeX/LaTeX and HTML/Mathml. Notice that TeXmacs is *not* based on TeX/LaTeX.
- Its rendering engine uses high-quality typesetting algorithms so as to produce professionally looking documents, which can either be printed out or presented from a laptop.
- New styles can be written by the user and new features can be added to the editor using the Scheme extension language.
- Runs on all major Unix platforms, MacOS, and Windows.

It is important to note that TeXmacs is not based on TeX/LaTeX but can use the functionality within documents. As can be seen, LaTeX documents can be imported and exported from the basic Scheme or XML format. The familiar PDF and postscript files can also be generated from a TeXmacs document.

The following video quickly demonstrates some of the most important features of TeXmacs: starting a scientific article, typing mathematical formulas, inserting and annotating pictures, citations, references, drawing pictures, document styles and fonts, HTML export, laptop presentations, and animations.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#oembed-2

Video length: 3:41 minutes

Online TeX-based Tools

Papeeria

The Papeeria online editor has two different plans for potential users to choose from.

Epsilon plan (free forever)

Unlimited public projects and public templates

Unlimited collaborators per project

One active private project you can archive unused projects.

Full-text search restricted to the current project files

Git sync full read-write access to public repositories

File edit history, covers last 24 hours

Simple plot builder

Delta Plan (starts $5 per month)

Adds to Epsilon, up to 10 active private projects You can buy more quota if needed

Auto compiling

Priority compiling

Private templates

Full-text search across all projects which you own or joined

Git sync access private repositories too

File edit history, covers last 30 days

Synchronization with Dropbox and Google Drive

Backup snapshots

Integration with gnuplot

Integration with Mendeley

Table 7.1 The two Papeeria plans available to users

The Epsilon plan is free for all time should one choose this option. Clearly, there are not as many features available and there is a limit on the number of active private projects. The website indicates that this option is best suited for students and those learning LaTeX. The Delta plan starts at $5 per month ($60 a year) and adds more features on top of the basic Epsilon plan.[1] It has the feature of being able to synchronize with Dropbox and Google Drive which some users may find useful.

The integration with Gnuplot, an open-source plotting program, requires additional installation information which the website does not provide. The integration of this tool (and Mendeley) would require sending a request to Papeeria support. Mendeley is a reference manager which is used to manage and share research papers and generate bibliographies for scholarly articles.[2] It is not clear whether the free Mendeley account would work with Papeeria.

A templates page provides downloadable files for the interested user. This is especially useful for the new user or student looking for specific document formats. Another interesting feature of the Papeeria editor is auto-compilation of a document to keep it up to date. Within Papeeria, it is possible to use LaTeX commands or mix it with Markdown (a text-to-HTML conversion tool for web writers).[3] As was mentioned before, the user does not need to update the underlying TeX distribution – this is taken care of by Papeeria support persons. There is also a mobile version of the tool for those interested in this option.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32

Video length: 5:52 minutes

Features of Papeeria

The following are the features of Papeeria.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#h5p-9

---

1. See the Papeeria pricing plans page
2. See the Mendeley site for information
3. See Markdown in Wikipedia.

Overleaf

Like the previous tool, Overleaf is an online collaborative LaTeX editing tool. It shares some of the features of any online editing tool. The following are the plans available for purchase with Overleaf.

Personal	**Collaborator**	**Professional**
Ideal when working alone	Great for shared projects	For working with multiple collaborators
$11 per month	$17 per month	$34 per month
Only one collaborator	10 collaborators per project	Unlimited collaborators
Sync with Dropbox and GitHub, full document history and more	Sync with Dropbox and GitHub, full document history, track changes and more	Sync with Dropbox and GitHub, full document history, track changes and more

Table 7.2 Overleaf pricing plans to choose from

There is a free plan available as well. However, the Overleaf website does not provide information on what features are available for this plan.

Features of Overleaf

Here are the features of Overleaf.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#h5p-10

Support pages to help users Learn

Overleaf provides a number of pages to help their users with different tasks. These are itemized as follows.

1. LaTeX in 30 Minutes – this is a guide for those who are new to LaTeX. No prior knowledge of LaTeX is assumed.
2. Templates – this is where the thousands of templates can be searched for and selected. The code for each template opens by default in the editor when selected.
3. Webinars – Enjoy free webinars and learn how you can make the most of your Overleaf account. There are webinars for all types of Overleaf users.
4. Tutorials – From this Overleaf learn wiki, you can access a wide range of help and information on Overleaf and LaTeX.
5. Inserting Images – This page steps through the process of adding images to documents as well as manipulating them (shrinking, enlarging, rotating).
6. How to create tables – This page describes the details of adding tables to LaTeX documents. **Code snippets** are provided as part of the explanation.

Overleaf provides an extensive list on their website of all the post-secondary institutions that currently have an institutional subscription. The list includes universities across North America, Europe, the Middle East, Asia and Australia.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#oembed-3

Video length: 2:07 minutes

R Markdown and Document Generation

The next tool to be examined in this chapter is R Markdown. The first observation is that there are two parts to this document generation process:

1. installation of R and the R Markdown package
2. installation of an appropriate editor, for example, RStudio or Pandoc

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

In a nutshell, R Markdown stands on the shoulders of knitr and Pandoc. The former executes the computer code embedded in Markdown, and converts R Markdown to Markdown. The latter renders Markdown to the output format you want (such as PDF, HTML, Word, and so on).[4]

To make the generation of documents easier, the use of the RStudio integrated development environment (IDE) is recommended. There are four different options available for use:

1. **RStudio Desktop** under an open source license which can be used for free
2. **RStudio Desktop Pro** under a commercial license at a cost of $995 per year
3. **RStudio Server** under an open source license which can be used for free
4. **RStudio Workbench** under a commercial license at a cost of $4,975 per year with five named users

As with other IDEs, RStudio includes a set of integrated tools designed to help you be more productive with R and Python applications. It includes a console, syntax-highlighting, an editor supporting direct code execution, tools for plotting, viewing history, debugging and workspace management.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#oembed-4

---

[4] Xie, Y., Allaire, J.J. and Grolemund, G. (2019). *R Markdown: The Definitive Guide*. CRC Press.

Video length: 1:12 minutes

Yihue Xie et. al., state that:[5]

> R Markdown was designed for easier reproducibility, since both the computing code, and narratives are in the same document, and results are automatically generated from the source code. R Markdown supports dozens of static and dynamic/interactive output formats.

Here is the workflow for creating documents from R Markdown using the RStudio IDE.

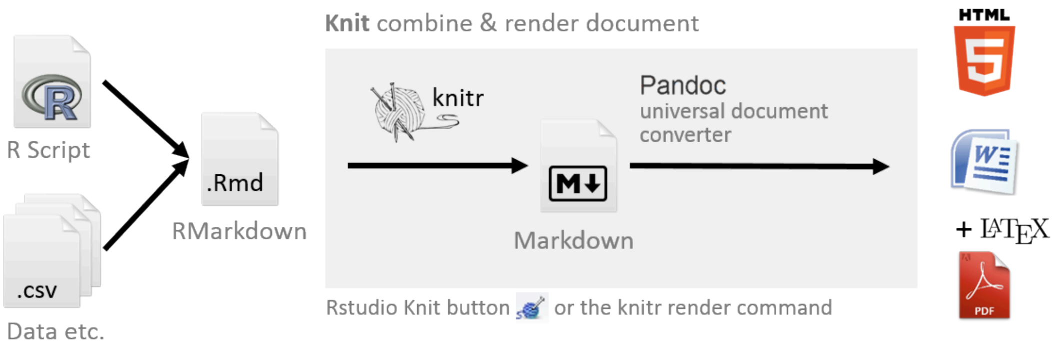

Knit combine & render document
R Script
.csv
Data etc.
.Rmd
RMarkdown
knitr
Pandoc
universal document converter
Markdown
Rstudio Knit button or the knitr render command
HTML
+ LaTeX
PDF

Figure 7. 1 The process for creating multiple document formats from an R Markdown script

Another feature of interest to educators is RStudio's free publishing platform RPubs which has a significant number of student homework assignments posted on it. This indicates the ease of use and convenience of R Markdown for students completing homework assignments.

The following two figures display an R Markdown cheatsheet that shows the simplicity of creating documents with it.

---

[5] Xie, Y., Allaire, J.J. and Grolemund, G. (2019). *R Markdown: The Definitive Guide*. CRC Press.

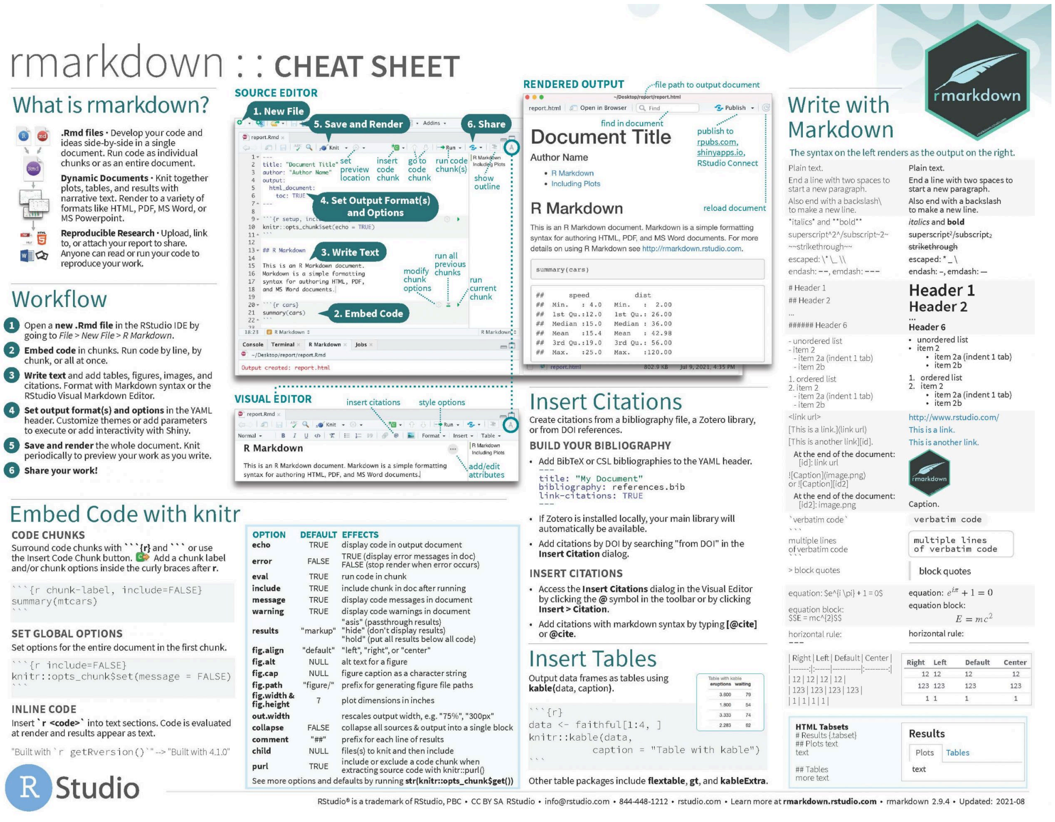


Figure 7.2 (a) First page of R Markdown cheatsheet

SOURCE EDITOR
1. New File
5. Save and Render
6. Share
4. Set Output Format(s) and Options
3. Write Text
2. Embed Code
set preview location
insert code chunk
go to code chunk
run code chunk(s)
show outline
modify chunk options
run all previous chunks
run current chunk
report.Rmd
Knit
Run
Addins
R Markdown
Including Plots
Console
Terminal
R Markdown
Jobs
VISUAL EDITOR
insert citations
style options
add/edit attributes
Normal
Format
Insert
Table
RENDERED OUTPUT
file path to output document
~/Desktop/report/report.html
report.html
Open in Browser
Find
Publish
find in document
publish to rpubs.com, shinyapps.io, RStudio Connect
reload document
Table with kable
Results
Plots
Tables
text

rmarkdown : : CHEAT SHEET

What is rmarkdown?

.Rmd files · Develop your code and ideas side-by-side in a single document. Run code as individual chunks or as an entire document.

Dynamic Documents · Knit together plots, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS Powerpoint.

Reproducible Research · Upload, link to, or attach your report to share. Anyone can read or run your code to reproduce your work.

SOURCE EDITOR

~~~~
---
title: "Document Title"
author: "Author Name"
output:
  html_document:
    toc: TRUE
---

```{r setup, incl
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document.
Markdown is a simple formatting
syntax for authoring HTML, PDF,
and MS Word documents.

```{r cars}
summary(cars)
```
~~~~

```
~/Desktop/report/report.Rmd
Output created: report.html
```

Workflow

1. Open a **new .Rmd file** in the RStudio IDE by going to *File > New File > R Markdown*.
2. **Embed code** in chunks. Run code by line, by chunk, or all at once.
3. **Write text** and add tables, figures, images, and citations. Format with Markdown syntax or the RStudio Visual Markdown Editor.
4. **Set output format(s) and options** in the YAML header. Customize themes or add parameters to execute or add interactivity with Shiny.
5. **Save and render** the whole document. Knit periodically to preview your work as you write.
6. **Share your work!**

VISUAL EDITOR

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

RENDERED OUTPUT

Document Title

Author Name

- R Markdown
- Including Plots

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

report.html 802.9 KB Jul 9, 2021, 4:35 PM

Embed Code with knitr

CODE CHUNKS

Surround code chunks with ```{r} and ``` or use the Insert Code Chunk button. Add a chunk label and/or chunk options inside the curly braces after **r**.

~~~~
```{r chunk-label, include=FALSE}
summary(mtcars)
```
~~~~

SET GLOBAL OPTIONS

Set options for the entire document in the first chunk.

~~~~
```{r include=FALSE}
knitr::opts_chunk$set(message = FALSE)
```
~~~~

INLINE CODE

Insert **`r <code>`** into text sections. Code is evaluated at render and results appear as text.

"Built with `r getRversion()`" --> "Built with 4.1.0"

OPTION	DEFAULT	EFFECTS
echo	TRUE	display code in output document
error	FALSE	TRUE (display error messages in doc) FALSE (stop render when error occurs)
eval	TRUE	run code in chunk
include	TRUE	include chunk in doc after running
message	TRUE	display code messages in document
warning	TRUE	display code warnings in document
results	"markup"	"asis" (passthrough results) "hide" (don't display results) "hold" (put all results below all code)
fig.align	"default"	"left", "right", or "center"
fig.alt	NULL	alt text for a figure
fig.cap	NULL	figure caption as a character string
fig.path	"figure/"	prefix for generating figure file paths
fig.width & fig.height	7	plot dimensions in inches
out.width		rescales output width, e.g. "75%", "300px"
collapse	FALSE	collapse all sources & output into a single block
comment	"##"	prefix for each line of results
child	NULL	files(s) to knit and then include
purl	TRUE	include or exclude a code chunk when extracting source code with knitr::purl()

See more options and defaults by running **str(knitr::opts_chunk$get())**

Insert Citations

Create citations from a bibliography file, a Zotero library, or from DOI references.

BUILD YOUR BIBLIOGRAPHY

- Add BibTeX or CSL bibliographies to the YAML header.

```
---
title: "My Document"
bibliography: references.bib
link-citations: TRUE
---
```

- If Zotero is installed locally, your main library will automatically be available.
- Add citations by DOI by searching "from DOI" in the **Insert Citation** dialog.

INSERT CITATIONS

- Access the **Insert Citations** dialog in the Visual Editor by clicking the **@** symbol in the toolbar or by clicking **Insert > Citation.**
- Add citations with markdown syntax by typing **[@cite]** or **@cite**.

Insert Tables

Output data frames as tables using **kable**(data, caption).

~~~~
```{r}
data <- faithful[1:4, ]
knitr::kable(data,
        caption = "Table with kable")
```
~~~~

Table with kable

eruptions	waiting
3.600	79
1.800	54
3.333	74
2.283	62

Other table packages include **flextable**, **gt**, and **kableExtra**.

Write with Markdown

The syntax on the left renders as the output on the right.

Plain text. End a line with two spaces to start a new paragraph. Also end with a backslash\ to make a new line.	Plain text. End a line with two spaces to start a new paragraph. Also end with a backslash to make a new line.
italics and **bold**	*italics* and **bold**
superscript^2^/subscript~2~	superscript^2/subscript$_2$
~~strikethrough~~	~~strikethrough~~
escaped: * _ \\	escaped: * _ \
endash: --, emdash: ---	endash: –, emdash: —
# Header 1 ## Header 2 ... ###### Header 6	**Header 1** **Header 2** ... **Header 6**
- unordered list - item 2 - item 2a (indent 1 tab) - item 2b	• unordered list • item 2 • item 2a (indent 1 tab) • item 2b
1. ordered list 2. item 2 - item 2a (indent 1 tab) - item 2b	1. ordered list 2. item 2 • item 2a (indent 1 tab) • item 2b
<link url>	http://www.rstudio.com/
[This is a link.](link url)	This is a link.
[This is another link][id]. At the end of the document: [id]: link url	This is another link.
![Caption](image.png) or ![Caption][id2] At the end of the document: [id2]: image.png	Caption.
`verbatim code`	verbatim code

```
multiple lines
of verbatim code
```
	multiple lines
	of verbatim code

> block quotes	block quotes
equation: $e^{i \pi} + 1 = 0$	equation: $e^{i\pi}+1=0$
equation block: $$E = mc^{2}$$	equation block: $E = mc^2$
horizontal rule: ---	horizontal rule:

```
| Right | Left | Default | Center |
|-------:|:------|-----------|:---------:|
| 12 | 12 | 12 | 12 |
| 123 | 123 | 123 | 123 |
| 1 | 1 | 1 | 1 |
```

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

HTML Tabsets

```
# Results {.tabset}
## Plots text
text

## Tables
more text
```

Results

Plots | Tables

text

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **rmarkdown.rstudio.com** • rmarkdown 2.9.4 • Updated: 2021-08

From Figure 7.2 (a) the basic elements of Markdown can be seen in the various code snippets shown.

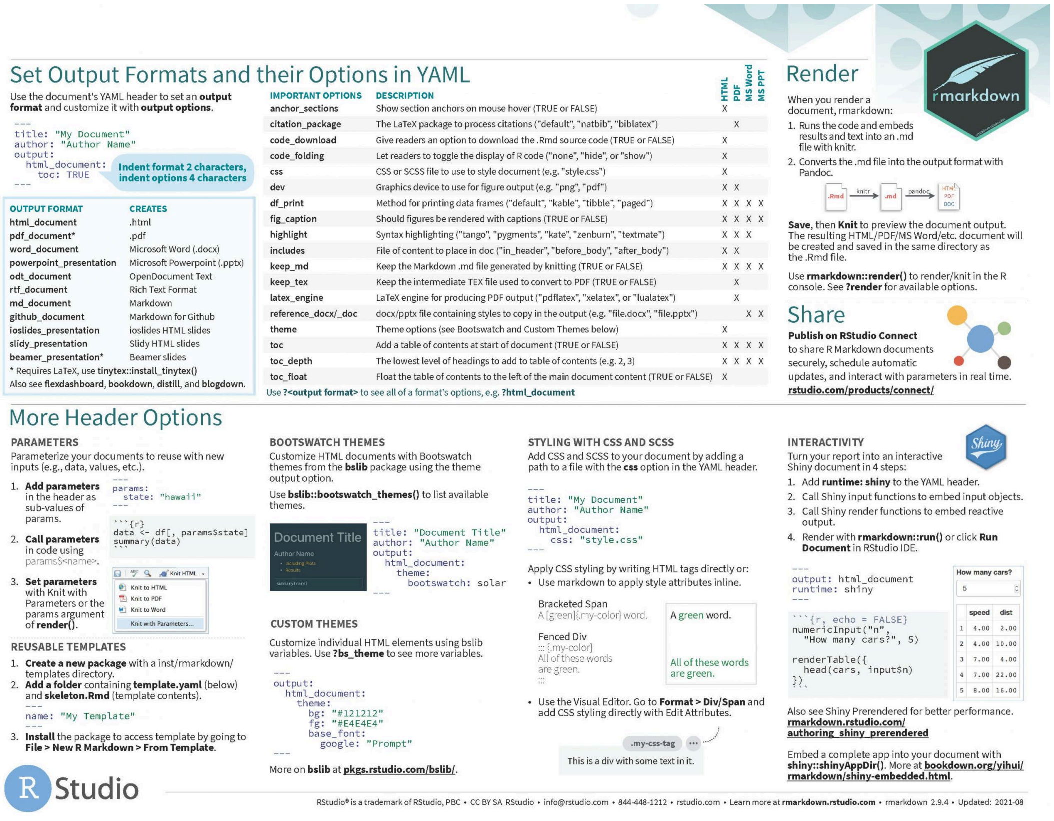


Figure 7.2 (b) Second page of R Markdown cheatsheet

.Rmd
knitr
.md
pandoc
HTML
PDF
DOC
Knit HTML
Knit to HTML
Knit to PDF
Knit to Word
Knit with Parameters...
Document Title
Author Name
Including Plots
Results
summary(cars)
.my-css-tag
How many cars?

Set Output Formats and their Options in YAML

Use the document's YAML header to set an **output format** and customize it with **output options**.

```
---
title: "My Document"
author: "Author Name"
output:
  html_document:
    toc: TRUE
---
```

Indent format 2 characters, indent options 4 characters

OUTPUT FORMAT	CREATES
html_document	.html
pdf_document*	.pdf
word_document	Microsoft Word (.docx)
powerpoint_presentation	Microsoft Powerpoint (.pptx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	Markdown for Github
ioslides_presentation	ioslides HTML slides
slidy_presentation	Slidy HTML slides
beamer_presentation*	Beamer slides

* Requires LaTeX, use **tinytex::install_tinytex()**

Also see **flexdashboard**, **bookdown**, **distill**, and **blogdown**.

IMPORTANT OPTIONS	DESCRIPTION	HTML	PDF	MS Word	MS PPT
anchor_sections	Show section anchors on mouse hover (TRUE or FALSE)	X			
citation_package	The LaTeX package to process citations ("default", "natbib", "biblatex")		X		
code_download	Give readers an option to download the .Rmd source code (TRUE or FALSE)	X			
code_folding	Let readers to toggle the display of R code ("none", "hide", or "show")	X			
css	CSS or SCSS file to use to style document (e.g. "style.css")	X			
dev	Graphics device to use for figure output (e.g. "png", "pdf")	X	X		
df_print	Method for printing data frames ("default", "kable", "tibble", "paged")	X	X	X	X
fig_caption	Should figures be rendered with captions (TRUE or FALSE)	X	X	X	X
highlight	Syntax highlighting ("tango", "pygments", "kate", "zenburn", "textmate")	X	X	X	
includes	File of content to place in doc ("in_header", "before_body", "after_body")	X	X		
keep_md	Keep the Markdown .md file generated by knitting (TRUE or FALSE)	X	X	X	X
keep_tex	Keep the intermediate TEX file used to convert to PDF (TRUE or FALSE)		X		
latex_engine	LaTeX engine for producing PDF output ("pdflatex", "xelatex", or "lualatex")		X		
reference_docx/_doc	docx/pptx file containing styles to copy in the output (e.g. "file.docx", "file.pptx")			X	X
theme	Theme options (see Bootswatch and Custom Themes below)	X			
toc	Add a table of contents at start of document (TRUE or FALSE)	X	X	X	X
toc_depth	The lowest level of headings to add to table of contents (e.g. 2, 3)	X	X	X	X
toc_float	Float the table of contents to the left of the main document content (TRUE or FALSE)	X			

Use **?<output format>** to see all of a format's options, e.g. **?html_document**

Render

When you render a document, rmarkdown:

1. Runs the code and embeds results and text into an .md file with knitr.
2. Converts the .md file into the output format with Pandoc.

Save, then **Knit** to preview the document output. The resulting HTML/PDF/MS Word/etc. document will be created and saved in the same directory as the .Rmd file.

Use **rmarkdown::render()** to render/knit in the R console. See **?render** for available options.

Share

Publish on RStudio Connect to share R Markdown documents securely, schedule automatic updates, and interact with parameters in real time. **rstudio.com/products/connect/**

More Header Options

PARAMETERS

Parameterize your documents to reuse with new inputs (e.g., data, values, etc.).

1. **Add parameters** in the header as sub-values of params.

```
---
params:
  state: "hawaii"
---
```

2. **Call parameters** in code using params$<name>.

~~~~
```{r}
data <- df[, params$state]
summary(data)
```
~~~~

3. **Set parameters** with Knit with Parameters or the params argument of **render()**.

REUSABLE TEMPLATES

1. **Create a new package** with a inst/rmarkdown/ templates directory.
2. **Add a folder** containing **template.yaml** (below) and **skeleton.Rmd** (template contents).

```
---
name: "My Template"
---
```

3. **Install** the package to access template by going to **File > New R Markdown > From Template**.

BOOTSWATCH THEMES

Customize HTML documents with Bootswatch themes from the **bslib** package using the theme output option.

Use **bslib::bootswatch_themes()** to list available themes.

```
---
title: "Document Title"
author: "Author Name"
output:
  html_document:
    theme:
      bootswatch: solar
---
```

CUSTOM THEMES

Customize individual HTML elements using bslib variables. Use **?bs_theme** to see more variables.

```
---
output:
  html_document:
    theme:
      bg: "#121212"
      fg: "#E4E4E4"
      base_font:
        google: "Prompt"
---
```

More on **bslib** at **pkgs.rstudio.com/bslib/**.

STYLING WITH CSS AND SCSS

Add CSS and SCSS to your document by adding a path to a file with the **css** option in the YAML header.

```
---
title: "My Document"
author: "Author Name"
output:
  html_document:
    css: "style.css"
---
```

Apply CSS styling by writing HTML tags directly or:

- Use markdown to apply style attributes inline.

Bracketed Span A [green]{.my-color} word.	A green word.
Fenced Div ::: {.my-color} All of these words are green. :::	All of these words are green.

- Use the Visual Editor. Go to **Format > Div/Span** and add CSS styling directly with Edit Attributes.

This is a div with some text in it.

INTERACTIVITY

Turn your report into an interactive Shiny document in 4 steps:

1. Add **runtime: shiny** to the YAML header.
2. Call Shiny input functions to embed input objects.
3. Call Shiny render functions to embed reactive output.
4. Render with **rmarkdown::run()** or click **Run Document** in RStudio IDE.

```
---
output: html_document
runtime: shiny
---
```

~~~~
```{r, echo = FALSE}
numericInput("n",
  "How many cars?", 5)

renderTable({
  head(cars, input$n)
})
```
~~~~

How many cars? 5

	speed	dist
1	4.00	2.00
2	4.00	10.00
3	7.00	4.00
4	7.00	22.00
5	8.00	16.00

Also see Shiny Prerendered for better performance. **rmarkdown.rstudio.com/authoring_shiny_prerendered**

Embed a complete app into your document with **shiny::shinyAppDir()**. More at **bookdown.org/yihui/rmarkdown/shiny-embedded.html**.

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **rmarkdown.rstudio.com** • rmarkdown 2.9.4 • Updated: 2021-08

From Figure 7.2 (b), more code snippets can be seen. The number of output formats for R Markdown are also shown. One observation is that R Markdown makes use of LaTeX to produce files in PDF format. Within the RStudio editor is the option to use an installed LaTeX distribution (MiKTeX, MacTeX or TeXLive) to create files in PDF.

Case Study of using R Markdown to Publish a book

The following video describes the process for using RStudio and R Markdown to create the book, *Hands-on Data Visualization* by Jack Dougherty and Ilya Ilyankou.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=32#oembed-5

Video length: 3:20 minutes

The workflow for this book (and the above video) is shown below. Notice how the single source from RStudio produces multiple output formats.

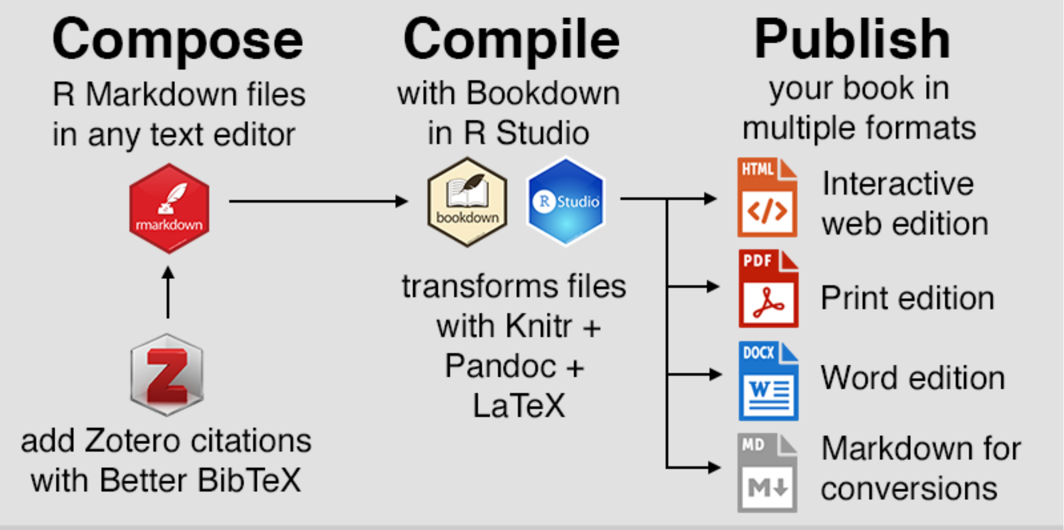

Compose
R Markdown files in any text editor
rmarkdown
add Zotero citations with Better BibTeX
Compile
with Bookdown in R Studio
bookdown
R Studio
transforms files with Knitr + Pandoc + LaTeX
Publish
your book in multiple formats
HTML
Interactive web edition
PDF
Print edition
DOCX
Word edition
MD
Markdown for conversions

Figure 7.3 : Simplified workflow to compose, compile, and publish in multiple formats with Bookdown. Images from [Daniel Hendricks](), [RStudio](), and [Zotero]().

It is worth mentioning that there are other editors that can be used in combination with R besides RStudio. One of these is the WinEdt freeware editor mentioned in a previous chapter as a TeX-based editor.

The **bookdown** package is an [open-source R package]() that facilitates writing books and long-form articles/reports with R Markdown. Features include:[6]

- Generate printer-ready books and ebooks from R Markdown documents.
- A markup language easier to learn than LaTeX, and to write elements such as section headers, lists, quotes,

---

6. See [Bookdown]() webpage.

figures, tables, and citations.
- Multiple choices of output formats: PDF, LaTeX, HTML, EPUB, and Word.
- Possibility of including dynamic graphics and interactive applications (HTML widgets and Shiny apps).
- Support a wide range of languages: R, C/C++, Python, Fortran, Julia, Shell scripts, and SQL, etc.
- LaTeX equations, theorems, and proofs work for all output formats.
- Can be published to GitHub, bookdown.org, and any web servers.
- Integrated with the RStudio IDE.
- One-click publishing to Bookdown.

A list of books, mostly published through CRC Press and O'Reilly, can be seen on the Bookdown webpage. A number of these books are openly licensed and can be accessed online. Clearly, the benefit of this is any potential author can leverage the online content as a starting point for an adapted or new resource.

For more detailed information about the use of R Markdown, Bookdown and RStudio to create open resources, the interested reader should consult the references list in Appendix A: Books created using R Markdown and Bookdown.

8. Emerging Publishing Tools

ISAAC MULOLANI

Learning Outcomes

By the end of this chapter, the reader should be able to

1. Identify alternative publishing tools for OER
2. Describe the features of each tools useful for publishing
3. Compare the list of features in each of these tools

Up to this point, we have discussed a number of tools ranging from word processors (commercial and open-source) and several open-source tools that have been around for a while. In this chapter, we will describe a select group of emerging tools that are growing in use. In particular, we will look at Pressbooks, EdTech Books, LibreTexts, Ximera and PreTeXt.

All of these tools are open-source options although PressbooksEDU is provided as a paid subscription to many institutions across the globe. Institutions with the means to do so can install and maintain an open-source instance of Pressbooks that can be connected to other paid subscription networks as well. BCcampus and Concordia University are examples of open-source Pressbooks installations.

Pressbooks

Pressbooks is an open educational resources development platform based on WordPress. The platform is available for a number of different audiences.

Higher Education Institutions	Faculty Authors	Self-Publishers
Build a publishing program on a branded institutional Pressbooks network or get a central network for a consortium of schools.	Take advantage of our suite of EDU features to create books on a shared network. This option is based on a per book fee.	Publish your own novel, monograph, or other text in multiple online and offline formats.

Table 8.1 Audiences using Pressbooks

From their website, Pressbooks states that they have three products designed by and for educators

1. Pressbooks Directory,
2. Authoring & Editing Platform, and
3. Results for LMS.

Together, these elements form the infrastructure for open publishing.

An interactive H5P element has been excluded from this version of the text. You can view it online here:
https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#h5p-12

Key Pressbooks Open Education Features

For each of these, there are specific features of interest to open education users. A document on the 2020-PressbooksEDU-Overview can be obtained from the Pressbooks website.

Pressbooks Directory

One of the most useful features of Pressbooks is making all public books created in the platform available and discoverable in the Pressbooks Directory. This means that both Pressbooks users and non-users can access material, read it on the web and download it in multiple formats. However, only PressbooksEDU users can clone and adapt OER in accordance with its license. The features include:

- Discover New Content – Any open education practitioner can search for open textbooks, course outlines, lab manuals and more. Currently there are over 3,314 texts created using Pressbooks.
- Filter by License – The Directory is a search facility that has several filters. One of these is the ability to search for content by its licence.
- Find Interactive OER – Another filter that the Directory uses for searching is by the number of H5P interactive elements.
- Import Content – For users with access to a PressbooksEDU instance, openly licensed content can be imported from the Pressbooks Directory. It can also be imported from other sources including MS Word documents and Open Document Format. This will then allow the user to adapt the OER as per license permissions.

Pressbooks Authoring & Editing Platform

Adaptation of existing OER for classrooms can be achieved through Pressbooks Authoring & Editing Platform.

Users can create content from scratch and enrich OER with accessible math notation, videos, audio, web annotation and interactive elements. Specific features include:

- Write New content or Import & Edit – A user can write original OER content or adapt an existing OER. Options exist to import content from the Pressbooks Directory and other sources to adapt to a specific local need.
- Add Accessible Math Notation – A user can ensure that all STEM content uses accessible math notation. Pressbooks renders LaTeX, AsciiMath and MathML with MathJax. Content can be exported in a variety of downloadable formats including PDF, and EPUB.
- Include Audio and Video Content – OER can be enhanced by including audio/visual content. It is possible to include music, recorded lectures, lab demonstrations etc., into OER.
- Add Assessment with H5P – One can include assessments throughout an OER using H5P. The performance on these assessments can be reviewed in the LMS gradebook using Pressbooks Results for LMS.
- Create an Online Social Annotation Space with Hypothesis – An instructor can assign students in-text activities incorporating discussions. These discussions can be enriched with audio/visual content.
- Publish with beautiful book templates – Pressbooks current has a wide range of book templates presenting content in accessible and professional form.
- Conform with the Highest Accessibility Standards – Pressbooks is built to conform with WCAG 2.0 A & AA guidelines. It also encourages the creation of accessible OER content that satisfies the UN SDG 4.
- Delivery of OER in multiple formats – A user can export and deliver content in 10+ formats that include web, PDF, EPUB and Thin Common Cartridge. This conforms to accessibility requirements to provide multiple means of access to content.

Pressbook Results for LMS

The Pressbooks development platform supports the creation of open content and the integration of course material into a learning management system (LMS). Pressbooks has achieved IMS Global certification for the LTI 1.3 and LTI Assignment & Grade Service (See IMS Global certification). The following are features available:

- Secure LTI 1.3 Advantage connection to LMS – Educational content created using Pressbooks can be securely and seamlessly integrated into online classes delivered in an institution's learning management system.
- Interactions and display – Students and instructors can better understand how students are learning using Pressbooks. Analytics from H5P quizzes and other interactive elements can be passed to the LMS's gradebook.
- Analytics – The analytics for how an institution is using Pressbooks can be accessed through Network Analytics. This allows network managers to track information about users, books, network visits and downloads.

There are many different groups that use Pressbooks: educational institutions, academic presses, small publishers and individual authors. For any group interested in using it, there are the following options:

- PressbooksEDU plans: Pressbooks hosts and maintains a Pressbooks network for you. The plans include priority setup, tech support and training
- Host and maintain your own Pressbooks network: Pressbooks is open source software and requires staff

for development and maintenance.
- Use Pressbooks.com: This option is best for individuals or as a testing option for an institution.

For those interested in hosting and maintaining their own Pressbooks instance, the platform is free software released under the GNU Public Licence. There are a number of institutions such as BCcampus and Concordia University who maintain their own networks. The inhouse expertise for system administration and maintenance is a prerequisite to leveraging this option.

For those accessing the hosted PressbooksEDU plan (an increasingly popular option), access is provided to premium support. There is also documentation available on the Pressbooks Support page. Users can also access a Pressbooks channel in YouTube to watch product demos, update webinars etc. The following is a recent webinar from the Pressbooks YouTube channel.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-1

Video length: 1:03:32

The Appendix contains some useful Pressbooks guides and resources. The interested user is asked to consult these as a starting point.

Collaborative Pressbooks Repositories

Across North America, an increasing number of higher education institutions are leveraging Pressbooks as one of their primary open publishing platforms. In Canada, provincially supported Pressbooks repositories have been implemented to help facilitate collaboration across higher education institutions. Current examples of these include:

1. BCcampus – supporting the adaptation and evolution of teaching and learning practices in post-secondary institutions across British Columbia through collaboration, communication, and innovation.
2. Open Education Alberta – Open Education Alberta is a collaborative, no-fee publishing service for open textbooks and other open educational resources. This service is available to all Alberta post-secondary institutions.
3. Open Manitoba – Campus Manitoba and the Manitoba Open Education Initiative have established a PressbooksEDU network for Campus Manitoba's partner institutions, as a pilot project for the 2021-2022 academic year.
4. eCampusOntario – Pressbooks accounts at eCampusOntario are available at no cost to educators affiliated with one of Ontario's public post-secondary institutions.
5. AtlanticOER – Founded by CAUL-CBUA, AtlanticOER is a service that supports the hosting and creation of open educational resources by educators and students in the Atlantic provinces of Newfoundland and Labrador, New Brunswick, Nova Scotia, and Prince Edward Island.

One thing to note is that even with a provincial Pressbooks instance, some colleges and universities within the province do maintain their own institutional instance. For example in the Atlantic Canadian provinces, the University of Prince Edward Island, Dalhousie University, and Nova Scotia Community College all have their own Pressbooks instances.

LibreTexts

LibreTexts is a tool that was developed with the idea of, *freeing the textbook from the limitations and costs of traditional textbooks…*[1] LibreTexts provides open and freely accessible resources that give students a more engaging learning experience without financial burden. The University of California initiated the LibreText Project to create a leading non-commercial open textbook technology.

LibreTexts' goal is to work with faculty, students and outside experts to build open educational resources. It is a simple to use online platform for creating, adapting and distributing OER. The LibreTexts team can be viewed on their site and includes administrative and development teams along with general and industrial advisory boards.

LibreText Libraries

The LibreTexts online platform is organised according to libraries: there are 14 different libraries:

Biology	Spanish	Mathematics	Statistics
Chemistry	Geosciences	Medicine	Workforce
Business	Humanities	Physics	
Engineering	K12 Education	Social Sciences	

Table 8.2 LibreText available Libraries

Within each of these libraries are:[2]

- **Campus Bookshelves** – this area holds campus-specific and faculty-specific course shells. These shells can be exported into an LMS as PDF or physical texts.
- **Bookshelves** – the LibreTexts team curates these texts for direct use or in creating remixes for use in localized resources. Two categories of resources exist: **Textbooks** and **Textmaps**.
- **Learning Objects** – these are content items, practice items and assessment items. They are all self-contained digital and non-digital resources useful for learning.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-2

Video length: 49:01

One of the key features of LibreTexts is the OER Remixer. The following video explains the purpose and use of the OER Remixer.

---

1. See the LibreTexts web page.
2. See LibreTexts Libraries.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-3

Video length: 28:42

This OER Remixer greatly simplifies the task of adapting an OER from several sources. This is one of the major features of interest to instructors who do not have time to create a new resource.

Another feature recently added is ADAPT, the LibreTexts homework system. A somewhat longer video of this system is shown below.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-4

Video length: 1:19:35

Additional videos demonstrating many of the other features available in LibreTexts can be accessed through their LibreTexts YouTube channel. The channel contains playlists from LibreFest2020, LibreFest2021, tutorials and sessions from OpenEdWeek 2022.

EdTech Books

From the EdTech Books website, comes the following statement: "Our goal is nothing less than providing the best open textbook publishing platform and author experience on the web!"

The values behind the EdTech Books initiative are worth noting here.

1. **Freedom** — All of our content is freely available, and most of it is free to remix, reuse, and redistribute without seeking permission.
2. **Accessibility** — We design all of our content with a mobile-first mindset that focuses on making content fast and accessible to all.
3. **Usability** — All of our content undergoes ongoing usability testing to improve our users' experiences.
4. **Quality** — Our content is created by leaders in the field, and much of it undergoes similar peer review processes used by journals or editorial review processes used by book publishers.

The key point to note is that EdTech Books main focus is to create mobile-first resources. In the current day where smart phones and mobile devices are key tools for many in education, this is a critical strategy. Many students who do not have Internet access at home will use their mobile phone to access educational content. Any tools for creating open content should at the very least have an option to provide a mobile viewing option.

A look at the EdTech Books website indicates that the platform is being used by authors to create books and also journals. This is similar to how Pressbooks is being used in some institutions. One key difference on this site is the availability of book reviews with each resource. This feature is built into the creation of each chapter of a resource: authors are provided with a mechanism to collect feedback on each chapter from the reader.

Another feature available is the use of badges to signal quality.[3] Clearly, this helps interested instructors make informed decisions on adopting a given resource. This option deals with the quality questions some instructors still may have.

Another resource for the interested user is the EdTech Books User Guide. This provides a user with instructions on how to use the features of EdTech Books effectively. Some of the basic features available for use are:

- Book covers – provides instructions on creating a book cover
- Callouts – different types of callouts for emphasizing content
- Equations with LaTeX – similar to Pressbooks, EdTech Books provides support for using LaTeX within a resource.
- Footnotes – EdTech Books does not automate the creation of footnotes like other platorms. Instructions are provided on how to create footnotes in the absence of this automation.
- Glossaries – instructions on how to access the glossary feature
- Images – instructions on how to add images to the media library
- Videos – this provides instructions on how to add YouTube and Vimeo videos
- Microsoft Word Conversion – by default, all chapters can be downloaded as Microsoft Word documents
- PDF conversion – by default, EdTech Books provides downloadable PDF chapter files for mobile and for print
- Public submissions – You can turn on public submissions for a book.
- Styles – there are a few styles that can be used in a resource.
- WYSIWYG Editor – a visual editor provided for creating a resource.
- Advanced Features – some of these include H5P, Code Snippets, CSS Customization, Google Analytics, Languages, Practice Quizzes, and Text-to-Speech.

One key difference between Pressbooks and EdTech Book is the fact that this platform is only accessible online from the EdTech Books website. For interested authors, a free account can be created using either Google or ORCID.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-5

Video length: 3:36

TeX-Based Publishing Extensions

Ximera

There are many instructors who use LaTeX workflows for their publication and teaching workflows. The output from a LaTeX workflow is a static document either in PDF or HTML format. Two faculty from Ohio State

---

[3] See quality indications in Edtech Books

University, Dr. Bart Snapp and Dr. Jim Fowler led a project to take a LaTeX source file and add interactive elements that allows students to access an online document that they can interact with. This would be in the form of a webpage. Watch the video Great Lakes Science Boot Camp for Librarians 2021 talk Ximera 2.0: Compiling in the Browser.

For students at Ohio State University, Ximera provides free interactive textbooks in a three sequence Calculus text. Clicking on each course, reveals a page organised by chapters and sections for a student to navigate. For example, the Calculus 1 text has 36 chapters organised by section for students to work through. In each section, there are a combination of text, videos, graphics, examples and interactive questions for students to answer. The purpose is to provide a more engaging student learning experience where they can immediately test their understanding of concepts just presented.

Ximera also comes with the ability to provide students with hints to the interactive questions, a math editor for entering responses, and feedback to the submitted responses. All these activities arise from a LaTeX text for Calculus with interactive elements added to it and provided as online interactive HTML content. The idea, of course, is this can be replicated in any TeX-based resource by following the same process.

The interested instructor can access the LaTeX source code from GitHub and provide students with either a static PDF document or learn how to create and deploy the interactive version of the text online through their own institution. Ximera Calculus is an open-source project licensed under a Creative Commons license.

The appeal of this is that an instructor can use single source to provide content in multiple formats. For many instructors who have created OER using LaTeX and/or TeX-based tools, this ability to add interactivity for students is very appealing. In this day when students want more interactive and engaging learning experiences, this is one option instructors can pursue. The advantage of Ximera is that the these interactive texts can also be integrated into a number of learning management systems.

The two issues connected to using Ximera that an instructor must consider are:

1. Ximera requires a text written using LaTeX source
2. Ximera use requires one to have the technical ability to install the backend server
3. an LMS to display the interactive content for students

With these required features, Ximera can be considered to be a solution that requires high technology skills to implement. This is probably the reason why only instructors who use LaTeX and have the technical skills to install this system are among its main users.

PreTeXt

Similar to Ximera, PreTeXt seeks to create interactive textbooks that allow a student to interact with the content. The key difference here is that the interactive content is provided to students via its own webpage. Another difference is that PreTeXt uses XML to create the interactive web content. A PDF version of the text can also be made available for those who prefer that option. Here again we see the use of a single input source file to produce multiple viewable output formats.

Once again, the use of this tool requires a high level of technical ability from the user. To use PreTeXt, one needs to convert a LaTeX document using the conversion process described on the PreTeXt page. A service for initial conversion is provided by David Farmer from the American Institute of Mathematics using file sharing through GitHub.

PreTeXt Documentation

There is a PreTeXt documentation page that includes an online and downloadable guide. The guide provides a summary of the technical details that are required to create a text in PreTeXt format. Some of the elements that can be added in a document include mathematical symbols using LaTeX, WeBWorK exercises, graphics and videos.

The source for the documentation is also available to aid understanding of the process. The usual elements needed in a book can also be included in the online book. It must be mentioned that to include WeBWorK exercises, one must have access to a local repository of questions. Other question repositories that can also be included include Numbas and Stack questions. There is a page provided that demonstrates the use of WeBWorK exercises in a PreTeXt resource.

Installing PreTeXt on Windows

The process of installing PreTeXt on Windows contains a number of steps. These are outlined in the following video.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-6

Video length: 12:38

Once the PreTeXt installation process is installed as in the previous video, then one can begin the process of creating an online text. The following video describes the process.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-7

Video length: 3:01

The following video shows the PreTeXt version of an APEX Calculus adaptation by Dr. Sean Fitzpatrick from the University of Lethbridge in Alberta Canada. APEX Calculus is an openly licensed resource originally created by faculty at Virginia Military Institute. The source files for the original version of this open resource are available from GitHub.

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=105#oembed-8

Video length: 10:49

For those interested, Dr. Fitzpatrick has additional OER available on his website. These include adaptations of APEX Calculus mentioned in the video as well as other texts.

Gallery of Examples

The PreTeXt page has a gallery of examples demonstrating how texts in different areas can be provided to students. A selection of some texts created using PreTeXt are described below.

1. Discrete Mathematics: An Open Introduction, 3rd Edition by Oscar Levin, 2019 – This CC licensed resource is endorsed by the American Institute of Mathematics' Open Textbook Initiative and is well reviewed on the Open Textbook Library[4]. The online homework sets are available through Edfinity or as WeBWorK sets from the author. A PDF version is provided for offline use and both PreTeXt and LaTeX source are available from GitHub. The text contains 473 exercises with 275 solutions and 109 hints. Discrete Mathematics: An Open Introduction by Oscar Levin is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
2. Active Calculus, 2019 Edition by Matthew Bolkins – this openly licensed resource contains 150 interactive exercises, interactive Desmos activities, interactive applets and some interactive javascript applets. A PDF version is available for offline use and the source code can be downloaded from GitHub. The text is licensed under a Creative Commons Share Alike 4.0 International License.
3. Abstract Algebra: Theory and Applications, 2021 Annual Edition by Tom Judson – this text is licensed using a GNU Free Documentation License (GFDL). There are English and Spanish versions available for download along with PDF versions and source code for the text from GitHub. The text contains additional material created using the Sage open source computer algebra system.
4. Understanding Linear Algebra by David Austin – this contains Sage evaluations throughout the text for some of the computations. The text is licensed under a Creative Commons Attribution 4.0 International License.
5. Music Theory for the 21st-Century Classroom by Robert Hutchinson – this resource is licensed under a GNU Free Documentation License (GFDL). It contains audio and video elements to help the student identify specific musical elements.
6. Engineering Statics: Open and Interactive by Danial W. Baker and William Haynes – this book can be obtained as a downloadable PDF document in addition to online access. The source files are available through GitHub. It contains interactive GeoGebra diagrams. The text is licensed under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 International License.

---

[4] See Discrete Mathematics 3rd Edition by Oscar Levin

Glossary

ISAAC MULOLANI

Accessibility

Accessibility can be viewed as the "ability to access" something. The concept of accessible design and practice of accessible development ensures both "direct access" (i.e. unassisted) and "indirect access" meaning compatibility with a person's assistive technology (for example, computer screen readers). (Source: Wikipedia.org)

ALMS framework

The ALMS Framework provides a way of thinking about those technical choices and understanding the degree to which they enable or impede a user's ability to engage in the 5R activities permitted by open licenses.

Alt text

A word or phrase that can be inserted as an attribute in an HTML (Hypertext Markup Language) document to tell website visitors the nature or contents of an image. (Source: WhatIs.com)

Attribution

The process by which a content user gives proper credit to the original creator of a work when a portion of that work is reused or adopted outside of its original context. Attribution typically includes a link to the original work and information about the author and license.

Backward design

A model for designing instructional materials where the instructor or designer begins the design process with a focus on the desired results (i.e., the outcome) of instruction. (Source: Learning-Theories.com)

code snippets

A code Snippet is a programming term that refers to a small portion of re-usable source code, machine code, or text. Snippets help programmers reduce the time it takes to type in repetitive information while coding. Code Snippets are a feature on most text editors, code editors, and IDEs.

Copyright

A set of intellectual property laws that give the rightsholder of a work (usually the author) exclusive rights over the reproduction, reuse, remixing, display, performance, and redistribution of their work.

Copyright license

A license permits users to certain rights over a copyrighted work. These can be exclusive (allowed for individual groups) or nonexclusive (allowed for all users). Licenses can be restricted by certain factors such as purpose, territory, duration, and media (Source: Findlaw.com).

Course Learning Outcomes

The final outcomes that an instructor expects their students to gain by the time the students complete a course.

Creative Commons

A set of open licenses that allow creators to clearly mark how others can reuse their work through a set of four badge-like components: Attribution, Share-Alike, Non-Commercial, and No Derivatives.

Creative Commons licenses

A Creative Commons (CC) license is one of several public copyright licenses that enable the free distribution of an otherwise copyrighted "work". A CC license is used when an author wants to give other people the right to share, use, and build upon a work that the author has created. CC provides an author flexibility (for example, they might choose to allow only non-commercial uses of a given work) and protects the people who use or redistribute an author's work from concerns of copyright infringement as long as they abide by the conditions that are specified in the license by which the author distributes the work.

Derivative works

A work based on or derived from one or more already existing works. Common derivative works include translations, musical arrangements, art reproductions, and abridgments. (Source: USLegal.com)

Fair Use

A legal doctrine that promotes freedom of expression by permitting the unlicensed use of copyright protected works in certain circumstances. In Canada, this is known as Fair Dealing.

Inclusivity

The practice or policy of including people who might otherwise be excluded or marginalized, such as those who have physical or mental disabilities and members of minority groups. (Source: Oxford living dictionary)

Learning Management System (LMS)

A piece of software that manages, analyses, and runs educational courses. Canvas and Blackboard are two popular examples.

Licensing

The process by which a rightsholder (usually the creator of a work) dictates that others can reuse their work in specific ways.

mashup

a mixture or fusion of disparate elements.
"the movie becomes a weird mash-up of 1950s western and 1970s TV cop show"
a recording created by digitally combining and synchronizing instrumental tracks with vocal tracks from two or more different songs.
"a classic dancefloor mash-up"
Computing

a web page or application created by combining data or functionality from different sources.
"a mash-up that mixes CNN news with links to Wikipedia articles"

Open access

A model by which content creators make their scholarly outputs free to access without cost to users. This can be done either by publishing content with an OA publisher or by sharing a copy of the content on an open repository.

Open educational practices

Practices which encourage the development of openness, community engagement, transparency, responsibility, sharing, and accountability in education. (Source: Open Education Practices [Wikibooks])

Open educational resources

Free educational materials that are openly licensed to enable reuse and redistribution by users.

Open license

A copyright license which grants permission for all users to access, reuse, and redistribute a work with few or no restrictions.

open licensing

A licence is a document that specifies what can and cannot be done with a work (whether sound, text, image or multimedia). It grants permissions and states restrictions. Broadly speaking, an open licence is one which grants permission to access, re-use and redistribute a work with few or no restrictions. (A full set of conditions which must be met in order for a licence to be open is available in the Open Knowledge Definition 1.0.). For example, a piece of writing on a website made available under an open licence would be free for anyone to: print out and share, publish on another website or in print, make alterations or additions, incorporate, in part or in whole, into another piece of writing, use as the basis for a work in another medium – such as an audio recording or a film, and do many other things …

Openly licensed works are hence free to be shared, improved and built upon! The exact permissions granted depend on the full text of the open license that is applied.

Open pedagogy

A set of pedagogical practices that include engaging students in content creation and making learning accessible to all.

Open science

An umbrella term for a movement comprised of a variety of practices aiming to remove barriers for sharing any kind of research output, including resources, methods, or tools created at any stage of the research process. (Source: FosterOpenScience.eu)

open source

denoting software for which the original source code is made freely available and may be redistributed and modified.

Open source software

Software with source code that anyone can inspect, modify, and enhance. (Source: OpenSource.com)

Open textbook

An openly licensed and free to access textbook; an OER meant to be used as a textbook for a course.

Public Domain

A work which is not covered under copyright law, whose copyright has expired, or which has been dedicated to the public domain by its rightsholder is said to be in the public domain.

shareware

Shareware is a type of proprietary software which is initially shared by the owner for trial use at little or no cost with usually limited functionality or incomplete documentation but which can be upgraded upon payment. Shareware is often offered as a download from a website or on a compact disc included with a magazine. Shareware differs from freeware, which is fully-featured software distributed at no cost to the user but without source code being made available; and free and open-source software, in which the source code is freely available for anyone to inspect and alter.

source code

Source code is generally understood to mean programming statements that are created by a programmer with a text editor or a visual programming tool and then saved in a file.

Student Learning Outcomes

The outcomes that an instructor expects their students to display at the end of a learning experience (an activity, process, or course). (Source: Elhabashy, 2017).

Textbooks

Textbooks are the central spot for integrated content into our library and are identified by "Book:" in their titles.

Textmaps

Textmaps are specialized remixes that are constructed to follow the organization of existing commercial textbooks. Textmaps facilitate adoption by faculty that are unable to switch from a commercial textbook to an OER alternative; these texts are identified by "Map:" in their titles.

Turing-complete programming language

Turing completeness is a concept from theoretical computer science. It tells you how powerful a programming language is. Not in terms of performance or maintainability or how rich its ecosystem is. A programming language is Turing complete if you can implement any possible algorithm with it.

Universal Design

A process intended to design products that are usable by all people,with or without disabilities, to the greatest extent possible (Edyburn, 2015).

Universal Design for Learning

A framework to improve and optimize teaching and learning for all people based on the concept that, by providing multiple ways of engaging with content, the diverse educational needs of learners can be met.

Appendix A: Books created using R Markdown and Bookdown

ISAAC MULOLANI

Here is a list of some of the books that have been created with R Markdown and Bookdown. This list should be added into the guide to provide clear examples of what has already been created so far.

1. Bookdown Tutorial. No mention of a licence.
2. QCBS R Workshop Series: A Template Guide for Written Material. Licenced as CC-BY-NC-SA.
3. R Markdown for scientists. Licenced as CC-BY-NC.
4. R for Data Science. Licenced as CC-BY.
5. Open Tools for writing open interactive textbooks. Licenced as CC-BY-SA.
6. Hands on Data Visualization. Licenced as CC-BY-NC-ND.
7. An Introduction to R. Licenced as CC-BY-NC.
8. Epidemiologist R Handbook. Licenced as CC-BY-NC-SA.
9. Answering Questions with Data. Licenced as CC-BY-SA.
10. Introduction to Statistical Modeling. Course text notes with no licence indicated.
11. Advanced Statistics using R. Copyrighted text.
12. An Introduction to Statistical Programming Methods with R. Licenced as CC-BY-NC-SA.
13. R Markdown: The Definitive Guide. Licenced as CC-BY-NC-SA.
14. Knitr with R Markdown blog page.
15. R Markdown Basics seminar. No licence mentioned.
16. Getting Started with R, R Studio, and R Markdown personal website.
17. R Markdown Cookbook. Licenced as CC-BY-NC-SA.
18. R for Reproducible Analysis and Research Transparency workshop. Copyrighted.
19. Documents with R Markdown. Licenced as CC-BY-NC-SA.
20. Geocomputation with R, first edition. Licenced as CC-BY-NC-ND
21. Bookdown: Authoring Books and Technical Documents with R Markdown. Licenced as CC-BY-NC-SA.
22. Welcome to Text Mining with R. Licenced as CC-BY-NC-SA.
23. Introduction to Data Science. Licenced as CC-BY-NC-SA.
24. Beyond Multiple Linear Regression: Applied Generalized Linear Models and Multilevel Models in R. Licenced as CC-BY-NC-SA.
25. Forecasting: Principles and Practice, Second Edition. Free and online as well as accessible. No licence mentioned.
26. Data Science Live Book. Licenced as CC-BY-NC-SA.

Appendix B: Books Created using LaTeX-based Tools

ISAAC MULOLANI

In this appendix, some books created with LaTeX-based tools are listed for the interested reader. Note that all of these resources are openly licensed.

1. Abstract Algebra: Theory and Applications by Tom Judson 2021.
2. Active Calculus series by Matthew Bolkins 2019.
3. A First Course in Linear Algebra by K. Kuttler 2021.

4. A gentle introduction to the art of mathematics by Joe Fields 2015.
5. APEX Calculus by Gregory Hartman 2018.
6. Applied Combinatorics by M.T. Keller and W.T. Trotter 2016.
7. Applied Discrete Structures by Al Doerr and Ken Levasseur 2020.
8. Basic Analysis: Introduction to Real Analysis by Jiri Lebl 2021.
9. Calculus Early Transcendentals by D. Guichard 2021.
10. CLP Calculus Textbooks by Joel Feldman, Andrew Rechnitzer and Elyse Yeager 2021.
11. Combinatorics Through Guided Discovery by Kenneth P. Bogart 2017.
12. Discrete Mathematics An Open Introduction, 3rd Edition by Oscar Levin 2013.
13. Fundamentals of Linear Algebra, 3rd Edition by Gregory Hartman 2011.
14. Linear Algebra by David Cherney, Tom Denton, Rohit Thomas and Andrew Waldron 2019.
15. Linear Algebra with Applications by W. Keith Nicholson 2021.
16. Notes on Diffy Qs: Differential Equations for Engineers by Jiri Lebl 2021.
17. Number Theory: In Context and Interactive by Karl-Dieter Crisman 2021.
18. Open Resources for Community College Algebra (ORCCA) by Portland Community College faculty 2019.
19. Precalculus, 3rd Corrected Edition by Carl Stitz and Jeff Zeager 2013.
20. Understanding Linear Algebra by David Austin 2020.

Appendix C: List of Latex References

ISAAC MULOLANI

The following are a list of books on learning how to use Latex.

1. Mittelbach, F., Goossens, M., with Braams, J., Carlisle, D. and Rowley, C. and contributions by Detig, C. and Schrod, J. (2004). The LaTeX Companion, 2nd Edition. Addison-Wesley, Boston, USA 2004.
2. van Dongen, M. R. C. (2012). LaTeX and Friends, Springer-Verlag, New York, USA 2012.
3. Griffiths, D. F. and Higham, D. J. (2016). Learning LaTeX, Second Edition. SIAM, Philadelphia, PA, USA.
4. Kopka, H. and Daly, P. W. (2003). Guide to LaTeX, 4th Edition. Addison-Wesley Professional, Boston USA 2003.
5. Gratzer, G. (2014). Practical LaTeX. Springer-Verlag, New York USA.
6. Goossens, M., Rahtz, S., Gurari, E.M., Moore, R. and Sutor, R.S. (1999). *The LaTeX Web Companion: Integrating TeX, HTML, and XML*. Addison-Wesley Professional, Boston USA.
7. Gratzer, G. (2016). More Math into LaTeX, 5th Edition. Springer-Verlag, New York USA.
8. Kottwitz, S. (2011). LaTeX Beginner's Guide. Packt Publishing, Birmingham-Mumbai.
9. Kottwitz, S. (2015). The Latex Cookbook. Packt Publishing, Birmingham-Mumbai.
10. Lamport, L. (1994). The LaTeX Document Preparation System, 2nd Edition. Addison-Wesley
11. Talbot, N. L. C. (2013). LaTeX for Complete Novices. Dickimaw Books.
12. Parthasarathy, S. (2014). Let's Learn LaTeX.
13. Goossens, M., Mittelbach, F., Rahtz, S., Roegel, D., and Voss, H, (2008). The LaTeX Graphics Companoion, 2nd Edition. Addison-Wesley Professional, Boston USA.
14. Voss, H. (2011). PSTricks: Graphics and Postscript for TEX and LaTeX. UIT Cambridge, England.
15. Voss, H. (2011). Typesetting mathematics with LaTeX. UIT Cambridge, England.
16. Voss, H. (2011). Typesetting tables with LaTeX. UIT Cambridge, England.
17. Voss, H. (2011). LaTeX Quick Reference. UIT Cambridge, England.
18. Voss, H. (2012). Presentations with LaTeX. Lehmanns Media, Berlin Germany.

Appendix D: Useful Pressbooks Guides and Resources

ISAAC MULOLANI

Key Pressbooks Documentation

There are a number of resources that any potential Pressbooks user many find extremely helpful. These have to do with technical details on the use of the tool, open resource creation, specific accessibility particulars etc. An increasing number of Pressbooks networks provide institutional users with guides on the use of Pressbooks along with webinars and/or workshops. The following list contains some useful Pressbooks resources for the first-time user.

1. Pressbooks User Guide – The Pressbooks guide provided by Pressbooks.com. Licensed under a CC-BY.
2. Pressbooks Network Manager's Guide – Documentation for those responsible for administer a network instance. Licensed under a CC-BY.
3. General Pressbooks FAQ – A Pressbooks Frequently Asked Questions.
4. Publishing with Pressbooks: A Visual Guide – This BCcampus resource provides a step-by-step visual instructions on creating your text using Pressbooks. Licensed under a CC-BY-SA.
5. Using LaTeX in Pressbooks – This BCcampus resource focusses on the use of LaTeX in Pressbooks including an introduction to the program and worked tutorials on creating equations, plots, and multi-lingual documents. Licensed under a CC-BY-SA.
6. Pressbooks Guidelines for the University of Minnesota – This guide is created by the University of Minnesota Libraries. Licensed under a CC-BY-NC.
7. Ryerson Open Textbook Authoring Guide – This book is a practical guide to adapting or creating open textbooks using the Pressbooks platform. Licensed under a CC-BY.
8. Pressbooks Tutorials – This Pressbooks Video Tutorial Series was created to assist faculty and staff to use, create, and adapt open textbooks with Pressbooks. This series was created by BCcampus.
9. Accessibility Toolkit, 2nd Edition – A general accessibility toolkit for users creating open educational resources. This is from BCcampus. Licensed under a CC-BY.
10. Adaptation Guide – BCcampus's reference to adapting or revising an open textbook. Licensed under a CC-BY.
11. Adoptions Guide, 2nd Edition – BCcampus's reference for instructors, institutions and students on adopting open textbooks. Licensed under a CC-BY.
12. The OER Starter Kit – This general guide is from Iowa State University. This starter kit has been created to provide instructors with an introduction to the use and creation of open educational resources (OER). Licensed under a CC-BY.
13. OER Starter Kit Workbook – This was created to work with the previous resource. It is designed for use by instructors, librarians, instructional designers, administrators, and anyone else interested in OER to explore. Licensed under a CC-BY.
14. UH OER Training – This University of Hawaii resources is a three-part training guide for bringing higher education instructors up to speed with Open Educational Resources (OER). This book was developed to serve as a standalone guide for independent creators and to support OER training through face-to-face, online, and hybrid delivery modes. Licensed under a CC-BY.
15. OER and Alternative Textbook Handbook – This handbook from the University of Houston provides

instructors with an introduction to the use and creation of open educational resources (OER). Licensed under CC-BY.

Appendix E: Checklist for Accessibility

Organizing content

- ☐ Content is organized under headings and subheadings.
- ☐ Headings and subheadings are used sequentially (e.g., Heading 1, Heading 2).

Images

- ☐ Images that convey information include alternative text (alt text) descriptions of the image's content or function.
- ☐ Graphs, charts, and maps also include contextual or supporting details in the text surrounding the image.
- ☐ Images do not rely on colour to convey information.
- ☐ Images that are purely decorative do not have alt-tag descriptions. (Descriptive text is unnecessary if the image doesn't convey contextual content information).

Links

- ☐ The link is meaningful in context and does not use generic text such as "click here" or "read more."
- ☐ Links do not open in new windows or tabs.
- ☐ If a link must open in a new window or tab, a textual reference is included in the link information (e.g., [NewTab]).

Tables

- ☐ Tables include row and column headers.
- ☐ Row and column headers have the correct scope assigned.
- ☐ Tables include a caption.
- ☐ Tables avoid merged or split cells.
- ☐ Tables have adequate cell padding.

Multimedia

- ☐ A transcript is available for each multimedia resource including relevant non-speech content.
    - Transcript includes:

    - speaker’s name
    - all speech content
    - relevant descriptions of speech
    - descriptions of relevant non-speech audio
    - headings and subheadings
- ☐ Captions of all speech content and relevant non-speech content are included in the multimedia resource; this includes the audio synchronized with a video presentation.
- ☐ Audio descriptions of contextual visuals (e.g., graphs, charts) are included in the multimedia resource.

Formulas

- ☐ Formulas have been created using MathML.
- ☐ Formulas are images with alternative text descriptions if MathML is not an option.

Font size

- ☐ Font size is 12 point or higher for body text.
- ☐ Font size is 9 point for footnotes or endnotes.
- ☐ Font size can be zoomed to 200%.

Appendix F: Accessibility and Usability

ABBEY ELDER

Learning Objectives

By the end of this chapter, you should be able to:

- Provide three examples of ways an OER can be checked for accessibility.
- Explain how Universal Design for Learning is a good practice for both pedagogy and accessibility.

Accessibility is one of the things that will determine the usability of an OER for learners with diverse needs. Exemplary OER borrow many best practices from web design, ensuring that content is readable and works as intended for all users.[1]

W3 Schools defines web accessibility as:

> "Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging."[2]

One or more interactive elements has been excluded from this version of the text. You can view them online here: https://opentextbooks.uregina.ca/toolsforcreatingoer/?p=477#oembed-1

Attribution: "Open Dialogues: Open education and accessibility" by CTLT, University of British Columbia [Youtube] is licensed CC BY 4.0.

Universal Design for Learning

Apart from more traditional aspects of accessibility, you can also make your course more accessible through the

---

1. **Attribution:** This chapter was adapted in part from The ABOER Starter Kit, by Technologies in Education at the Faculty of Education, the University of Alberta, licensed CC BY 4.0.
2. W3 Schools. "Web Accessibility." Accessed May 15, 2019. https://www.w3.org/WAI/bcase/soc.html#of

way(s) in which you present that content. One method is **Universal Design for Learning** (UDL), “a framework to improve and optimize teaching and learning for all people based on scientific insights into how humans learn.”[3] UDL claims that you can improve education for all learners by providing multiple ways of engaging with your course’s content. There are various ways to implement UDL in your teaching:

1. Represent ideas from different angles and in different media types to accommodate the diverse needs of learners.
2. Provide support for students to express their understanding of concepts in different ways.
3. Allow students to engage in different ways by providing a variety of assignment types.[4]

As the BC Campus OER Accessibility Toolkit (2015) argues, UDL principles can be applied to accessibility as well as learning. The toolkit provides the following examples for instructors:[5]

- Design resources that can be accessed by learners in a variety of ways. If there is a text component, provide the ability to enlarge the font size or change the text color.
- Provide multiple ways for learners to engage with information and demonstrate their knowledge.
- Identify activities that require specific sensory or physical capability and for which it might be difficult to accommodate the accessibility needs of learners (e.g., color matching activities).

Universal Design for Learning and **Universal Design** are two separate but interrelated concepts. While UDL is intended to improve both the accessibility and pedagogy of a learning environment, UD is primarily intended as an approach to making content accessible to as many people as possible.[6] We focus on UDL in this chapter because the design of open educational resources is inextricably connected to how they will be implemented as learning objects. As an educator, thinking about UDL as a process tied to the creation, sharing, and use of course content is essential.

Software & File Format Choice

The usability of an OER is heavily impacted by how easily users can access it.[7] Two aspects of content design that are inherently tied to accessibility are a resource’s file format and the software used to access it.

Choose open file formats

If someone wants to read your work, they need to be able to open the file on their computer; however, some

---

[3] CAST. "About Universal Design for Learning." Accessed July 15, 2019. http://www.cast.org/our-work/about-udl.html
[4] CAST. "Universal Design for Learning Guidelines Version 2.2," 2018. Accessed July 1, 2019. http://udlguidelines.cast.org/
[5] Coolidge, Amanda, et al. *Accessibility Toolkit – 2nd Edition*. Victoria, BC: BCcampus, 2015. https://opentextbc.ca/accessibilitytoolkit
[6] Edyburn, Dave L. *Accessible Instructional Design*. Bingley: Emerald Group Publishing Limited, 2015.
[7] **Attribution:** "Software & File Format Choice" was adapted from "Accessibility webpage" by Affordable Learning Georgia and UH OER Training by Billy Meinke, licensed CC BY 4.0.

file formats require specific proprietary software to open. Saving your work in open file formats can give your students more options for accessing their course content on whatever platform best meets their needs.

Examples

- Open formats: HTML, ePub, RTF, Mobi, PNG, XML, PDF, Markdown
- Proprietary formats: MS Word, Pages, PowerPoint, Keynote

- Markdown converter tool: The University of Oklahoma Libraries' Markdown Converter can be used to easily convert your Markdown text into most other formats.

Use accessible software

Some software used to create or display content disables accessibility features built into your computer's operating system, such as zoom, text-to-speech, and speech-to-text. It is important to check whether the software students will use to view your course content disables the accessibility features of their computer's operating system. This can be an issue both for OER and for traditional, publisher-provided course content.

Considerations

- Is the software used to view the OER compatible with most assistive devices?
- Does the software require point-and-click interaction to work properly?
- Can the software menus be "seen" and properly interpreted by screen readers?

How to check software accessibility

- Check common assistive keyboard shortcuts while using the software.
- Ensure that users can navigate content using only the keyboard if necessary.
- Enable OS accessibility features and check their effectiveness with the required software.

How to access common accessibility features

- Windows accessibility features
- Mac OS X accessibility features

Image & Text Readability

Whenever you are presenting content to students, it's important to check whether the text in your course content is recognizable to a computer **as text**. For PDFs, accurate optical character recognition (OCR) is often required to make the text understandable. Screen-readers require this information to accurately relay text back to students. Other best practices for making course materials readable are listed in the sections below.

Use heading levels (h1, h2, h3)

Text-based OER should always have a clear and logical structure. Using headings and other structural elements to organize your resource can make it easier for all learners to access and understand the material. Many editing tools support table of contents (TOC) generation based on where these section markers are placed. This can help students navigate to a specific chapter or section of a text, especially if the digital version of the resource has its TOC hyperlinked to each section within the text.

Individuals using screen readers can also more easily navigate the sections of your content when headings levels have been applied consistently.

Use true lists

While they may "look" similar to bulleted lists, using asterisks or icons to create a visual list of items can confuse a screen reader that is expecting to encounter structured content. Whenever listing items, use the true list features of your content editor, such as bullet points or numbered lists.

Examples

Good example:

- First list item
- Second one

Bad example:

* First list item

* Second item

Provide alt text & captions

No matter the subject of an image used in your content, you need to offer descriptive text. A screen reader will look for a contextual description of an image to share with readers, which should live in the text surrounding the image (title or caption) or as **alternative ("alt") text**. This is one of the most commonly overlooked aspects of accessibility for instructional content, but most text editors include tools for adding alt text to images.

When adding alt text to an image, be sure to clearly and succinctly describe the most important elements for the student to know. Do not include extraneous detail. In some cases, you do not need to add alt text at all, as in the case of purely decorative images.

Examples

- Necessary descriptive alt text: "Part a of the figure shows a container which has a gas of volume V subscript 1 on the left side and nothing on the right side. Part b shows a container which is completely filled with a gas of volume V subscript 2."
- Too much descriptive alt text: "There is a figure with a white background and two squares labeled a and b. Part a has a rectangle (representing a container) with a shaded grey section on the left half of the container with dots representing a gas. The gas is labeled V subscript 1. Part b…"
- Unnecessary descriptive alt text: "An icon of a person smiling – I put this here as a cute picture to liven up the page!"

Video and audio content needs descriptive text as well, but these usually take the form of captions or, in the case of podcast recordings, transcripts. You can easily add captions to videos using Canvas' Arc tool or by using YouTube's built-in editor tools. For more help with this process, read through the excellent Captioning Videos guide from the University of Washington or reach out to an instructional designer near you.

Use descriptive link text

Ensure that all web pages and links have titles that describe a topic or purpose. The purpose of the link can be determined by the text alone. That is, you don't need to include additional information justifying the use of the link. You want the link to be meaningful in context. For example, do not use generic text such as "click here" or "read more" unless the purpose of the link can be determined by meaning in the surrounding content.

Examples

Digital OER should have descriptive links that explain to where the hyperlink is going to navigate the reader.

- Good example: Information on the BC Open Textbook Project is available online.
- Bad example: Click here for information on the BC Open Textbook Project.

If the OER design does not permit the inclusion of explicit links in the text, implicit links can be used, and a more detailed list of sources should be provided at the end of the resource or in a separate document. Footnotes are a great way of providing more explicit links for content without cluttering the text on a page.

Use Accessible Fonts & Colors

OER should be readable for those with disabilities related to color as well. Some best practices for ensuring that fonts and colors are accessible are described below:

- Use dyslexic-friendly fonts, such as Arial, Century Gothic, Open Sans, and Verdana. Your institution might recommend certain fonts for digital and print materials. These recommended fonts are usually chosen for ease of use and accessibility and may be a good fit for your needs as well.
- Make sure there is a clear contrast between colors (e.g. between the background and font color, or between separate colors on a graph). There are many free online tools available for checking color contrast, but we recommend WebAim's Color Contrast Checker and ContrastChecker.com.
- Do not use color to communicate meaning without other markers of that meaning present. If you have color-dependent information in images or within the text of your resource, be sure that either alternative methods of recognition (such as differing patterns) are present, or that the contrast can be adjusted by users.

Online Accessibility Tools

A great deal of OER content is displayed on websites, where we can use accessibility-checking tools to identify areas that can make it difficult for assistive technology tools to work properly. The online WAVE tool does just that: identifying errors and possible issues with the accessibility of websites.

The Flexible Learning for Open Education (floe) website provides access to a suite of tools intended to "supports learners, educators and curriculum producers in achieving one-size-fits-one learning design for the full diversity of learners."[8]

---

8. For more information, see floe's Inclusive Learning Design Handbook online at https://handbook.floeproject.org/ or visit their source code on GitHub: https://github.com/fluid-project/

Appendix G: Summary of Technical Skills for OER Tools

ISAAC MULOLANI

At the start of this guide, the technical skills required for different kind of tools were stated as:

1. **Low-tech Skills**: The simplest way to create educational resources is by using familiar word processing tools such as Microsoft Word, Google Docs, or Libre Office. This software includes most of the features needed for standard content, and the file can be easily exported as a PDF or printed.
2. **Medium-tech Skills**: Another common way to create or edit educational resources is to create a website or hosted resource. This could be in the form of a blog, a static website, or a wiki. WordPress can be a great tool for these sorts of medium-tech projects.
3. **High-tech Skills**: There are a number of platforms that provide professional tools for authoring content, and some are very easy to use. This category requires the highest level of skills.

Using these categories, all the tools discussed in this guide will be ranked according to the skills they require. This should help faculty select tools most appropriate for their current technology skill level.

Technology Tool	Low-Tech Skills	Medium-Tech Skills	High-Tech Skills
Microsoft Word	Yes		
LibreOffice	Yes		
WPS Office	Yes		
Softmaker Office	Yes		
Google Docs	Yes		
Inkscape	Yes		
GIMP		Yes	
Pressbooks		Yes	
LibreTexts			Yes
EdTech Books		Yes	
RMarkdown			Yes
LaTeX			Yes
TeXLive			Yes
Gnu TeXmacs			Yes
LyX			Yes
Overleaf			Yes
Papeeria			Yes
Ximera			Yes
PreTeXt			Yes

Table G 1.1 Technology skills required for OER Tools

One general observation is that open source tools tend to require medium- to high-tech skills to use. One of the reasons for this is that these are technology tools designed by a global community. Most open source tools use other open source technologies which requires additional skills. For example, GIMP and Inkscape are open-source tools that can be used with any other open-source tools such as the TeX-based platform discussed.

To use the LaTeX based tools, there are proprietary and free editors provided. These options are designed to simplify the OER creation process. However, a user still must learn how to use these additional helper tools. For example, in conjunction with a MiKTeX installation on Windows, one can use either TeXnicCenter, TeXstudio or Kile which are open-source editors. Though they simplify the editing process, one must still learn to use these tools.

Appendix H: List of Open source Licenses

ISAAC MULOLANI

There are a number of different types of open licenses that can be used. The first set are the open source licenses that arose out of the open source movement in the 1990s. These are sometimes called Free and Open Source Software licenses[1]. The second set are most commonly used in the open education community – Creative Commons licenses. These were described in Chapter 1. In this appendix, a listing of some of the open source licenses in use is provided.

License	Author	Latest version	Publication date
Academic Free License	Lawrence E. Rosen	3.0	2002
Apache License	Apache Software Foundation	2.0	2004
CeCILL	CEA / CNRS / INRIA	2.1	June 21, 2013
Eclipse Public License	Eclipse Foundation	2.0	August 24, 2017
European Union Public License	European Commission	1.2	May 2017
FreeBSD	The FreeBSD Project		April 1999
GNU General Public License	Free Software Foundation	3.0	June 2007
GNU Lesser General Public License	Free Software Foundation	3.0	June 2007
IBM Public License	IBM	1.0	1999
Microsoft Public License	Microsoft	N/A	
MIT License/X11 License	MIT	N/A	1988
Open Software License	Lawrence Rosen	3.0	2005
OpenSSL License	OpenSSL Project	N/A	
PHP License	PHP Group	3.01	2019
Python Software Foundation License	Python Software Foundation	3.9.1	2020-10-05
Unlicense	unlicense.org	1	December 2010
W3C Software Notice and License	W3C	20021231	December 31, 2002
BSD License	Regents of the University of California	3.0	
Common Public License	IBM	1.0	May 2001
Educational Community License	Indiana University	1.0	2007
LaTeX Project Public License	LaTeX Project	1.3c	January 3, 2012
Mozilla Public License	Mozilla Foundation	2.0	
Sun Public License	Sun Microsystems		

Table H 1.1 List of some open source licenses

---

[1] See Wikipedia Comparison of free and open source software licenses

Appendix I: Indicator Traits for the Eight Affordances Framework of Digital Educational Resources

The following table provides indicators for the Eight Affordances Framework for exploring pedagogical possibilities of digital educational resources described in the second chapter of this resource.

Affordance	**Indicators**
1. Ubiquitous Learning	1. Anytime, anywhere availability to broaden educational access 2. Blurring the traditional boundaries of space and time: extending the scope of learning beyond the walls of the classroom and the cells of the institution's timetable 3. Curriculum-community connections
2. Active Knowledge Making	1. Learners as designers of knowledge and meaning 2. Demonstrated capacity to collect information, conceptualize its meaning, think critically and apply in real contexts 3. Making knowledge artefacts: projects, objects, social interventions 4. Learners have autonomy, control and agency as knowledge creators 5. Discovery and exploration 6. Opportunities for innovation and creativity
3. Multimodal Meaning	1. using a variety of modes of meaning (text, image, space, body, audio, simulations, virtual and augmented reality) 2. Making available a wide range of digital media resources 3. Supporting learners to make knowledge resources in a wide ranges of digital and non-digital media
4. Recursive Feedback	1. Appropriate feedback during learning and feedback-on-feedback 2. Assessments for learning that promote learning from mistakes and foster deeper meaning 3. Digital learning analytics 4. Peer review 5. Dashboard visualizations that make progress explicit to learners and instructors
5. Collaborative Intelligence	1. Peer-to-Peer learning 2. Group activities and social networking 3. Distributed cognition: learning by thinking, aware of the social nature of knowledge 4. Acknowledging the community and intellectual provenance of information and concepts 5. Networks of knowledge and learning
6. Differentiated Learning	1. Variable learning paths 2. Adaptive and personalized learning 3. Self-regulation and self-management of learning 4. Recognizing learner diversity and harnessing diversity as a productive learning resource 5. Supporting students to express their own identities, develop personal pathways 6. Trust and open-ness: nurturing digital citizenship

7. Metacognition	1. Cognition = the empirical, the topic, the theme — always linked to metacognition, hence multilevel thinking 2. Metacognition = the disciplinary framework, thinking conceptually/theoretically, regulating one's own thinking processes 3. Linking concrete and particular to the abstract, general and conceptual 4. Complex problem solving, addressing challenges with holistic, multiperspectival thinking 5. Authentic learning, linking disciplinary practice to local and personal circumstances
8. Accessibility	1. Affordability (with Open Access as one option) 2. Ownership: credit to creators, whether resources are free or at a price 3. Interoperability, removing digital systems silos in a way that a system can freely communicate and operate with other external systems and thereby open to them 4. Hybrid deployment across multiple platforms, browsers, operating systems and devices in a way that an application or resource is accessible over more than one platform like Windows, Mac. Android, Unix and Ubuntu 5. Universal Design for Learning (UDL) requirements for disability accessibility 6. Internationalization of functionalities in all resources and their interfaces, facilitating ease of translation