

Handbook of Big Data Applications and Analytics

Theory and Practice

Gregor von Laszewski

laszewski@gmail.com

Copyright © 2017 Gregor von Laszewski

laszewski@gmail.com

[HTTPS://GITHUB.COM/CLOUDMESH/CLASSES](https://github.com/couldmesh/classes)

First printing, October 2017



Contents

I

Preface

1	Introduction	9
1.1	About	9
1.2	Contributors	9

II

Documenting Scientific Research

2	Documenting Scientific Research	13
2.1	Writing a Scientific Article or Conference Paper	13
2.1.1	Professional Paper Format	14
2.1.2	Submission Requirements	14
2.1.3	Microsoft Word vs. \LaTeX	14
2.1.4	Working in a Team	15
2.1.5	Timemanagement	15
2.1.6	Paper Checklist	16
2.1.7	Example Paper	17
2.1.8	Creating the PDF from \LaTeX on your Computer	17
2.1.9	Class Specific README.md	17
2.1.10	Exercise	18

3	Introduction to \LaTeX	19
----------	---	-----------

3.1	Installation	19
3.1.1	Local Install	19
3.1.2	Online Services	20

3.2 Basic LaTeX Elements	21
3.2.1 Characters	21
3.2.2 Highlighting Text	21
3.2.3 Sections	22
3.2.4 Empty Lines	22
3.2.5 Itemize	22
3.2.6 Enumerate	22
3.2.7 Descriptions	23
3.2.8 Images	23
3.2.9 Tables	23
3.2.10 Labels	23
3.2.11 Mathematics	24
3.3 Advanced topics	25
3.3.1 ACM and IEEE Proceedings Format	25
3.3.2 Generating and Managing Images	25
3.3.3 Slides	25
3.3.4 Useful Online Information about \LaTeX	26
3.3.5 LaTeX vs. X	26
3.4 Editing	28
3.4.1 Emacs	28
3.4.2 Vi/Vim	28
3.4.3 TeXshop	28
3.4.4 LyX	28
3.4.5 WYSIWYG locally	29
3.4.6 Markdown and \LaTeX	29
3.4.7 pyCharm	30
3.4.8 MSWord	30
3.5 The LaTeX Cycle	30
3.6 Tips	31
4 Managing Bibliographies	33
4.0.1 Integrating Bibliographies	33
4.1 Entry types	34
4.1.1 Source code References	35
4.1.2 Researching proper bibtex entries	36
4.1.3 Article in a conference proceedings	38
4.1.4 What are the differnt entry types and fields	42
4.1.5 InProceedings	42
4.1.6 TechReport	43
4.1.7 Article	43
4.1.8 Proceedings	44
4.1.9 Wikipedia Entry	45
4.1.10 Blogs	45
4.1.11 Web Page	46
4.1.12 Book	46
4.2 Integrating Bibtex entries into Other Systems	49
4.2.1 Bibtex import to MSWord	49
4.3 Other Reference Managers	49
4.3.1 Endnote	49

4.3.2	Mendeley	49
4.3.3	Zotero	50
5	Editors	51
5.1	Basic Emacs	51
6	Other Formats	55
6.1	reStructuredText	55
6.1.1	Links	55
6.1.2	Source	55
6.1.3	Sections	56
6.1.4	Listtable	56
6.1.5	Exceltable	56
6.1.6	Boxes	56
6.1.7	Sidebar directive	57
6.1.8	Sphinx Prompt	57
6.1.9	Programm examples	57
6.1.10	Hyperlinks	58
6.1.11	Todo	58
6.2	Markdown	58
6.3	Communicating Research in Other Ways	59
6.3.1	Blogs	59
6.3.2	Sphinx	59
6.3.3	Notebooks	60



Preface

1	Introduction	9
1.1	About	
1.2	Contributors	



1. Introduction

F chapter/preface/about.tex

1.1 About

The document is based on selected material published at the following Web page

- <https://cloudmesh.github.io/classes/>

It is part of a class taught at Indiana University. The class communication takes place at:

- <https://piazza.com/class/ix39m27czn5uw>

The PDF version will be made in future available at

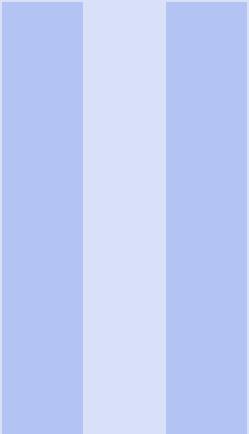
- <https://cloudmesh.github.io/classes/i524-notes.pdf>

This PDF document will be updated based on feedback from the students and once we have now material available. For a more complete set of information we recommend the students to visit the Web page.

1.2 Contributors

We like to acknowledge the following contributors that helped on this document. Please notify us with your name and a brief command on what you contributed:

John Doe He contributed to none of teh sections as this is just an example.



Documenting Scientific Research

2	Documenting Scientific Research	13
2.1	Writing a Scientific Article or Conference Paper	
3	Introduction to \LaTeX	19
3.1	Installation	
3.2	Basic \LaTeX Elements	
3.3	Advanced topics	
3.4	Editing	
3.5	The \LaTeX Cycle	
3.6	Tips	
4	Managing Bibliographies	33
4.1	Entry types	
4.2	Integrating Bibtex entries into Other Systems	
4.3	Other Reference Managers	
5	Editors	51
5.1	Basic Emacs	
6	Other Formats	55
6.1	reStructuredText	
6.2	Markdown	
6.3	Communicating Research in Other Ways	



2. Documenting Scientific Research

F

chapter/doc/report-book.tex

2.1 Writing a Scientific Article or Conference Paper

An important part of any scientific research is to document it. This is often done through scientific conferences or journal articles. Hence it is important to learn how to prepare and submit such papers. Most conferences accept typically the papers in PDF format but require the papers to be prepared on MSWord or in LaTeX. While working with many students in the past we noticed however that those students using Word often spend unnecessarily countless hours on trying to make their papers beautiful while actually violating the template provided by the conference. Furthermore, we noticed that the same students had issues with bibliography management. Instead of Word helping the student it provided the illusion to be easier than LaTeX but when adding up the time spent on the paper we found that LaTeX actually saved time. This has been especially true with the advent of collaborative editing services such as sharelatex [??] and overleaf [??].

In this section we provide you with a professional template that is used for either system based on the ACM standard that you can use to write papers. Naturally this will be extremely useful if the quality of your research is strong enough to be submitted to a conference. We structure this section as follows. Although we do not recommend that you use MSWord for your editing of a scientific paper, we have included a short section about it and outline some of its pitfalls that initially you may not think is problematic, but has proven to be an issue with students. Next we will focus on introducing you to LaTeX and showcasing you the advantages and disadvantages. We will dedicate an entire section on bibliography management and teach you how to use Jabref which clearly has advantages for us.

Having a uniform report format not only helps the students but also allows the comparison of paper length and effort as part of teaching a course. We have added an entire section to this chapter that discusses how we can manage a *Class Proceedings* form papers that are contributed by teams

in the class.

2.1.1 Professional Paper Format

The report format we suggest here is based on the standard ACM proceedings format. It is of very high quality and can be adapted for your own activities. Moreover, it is possible to use most of the text to adapt to other formats in case the conference you intend to submit your paper to has a different format. The ACM format is always a good start.

Important is that you do not need to change the template but you can change some parameters in case you are not submitting the paper to a conference but use it for class papers. Certainly you should not change the spacing or the layout and instead focus on writing content. As for bibliography management we recommend you use jabref which we will introduce in Section ??.

We recommend that you carefully study the requirements for the report format. We would not want that your paper gets rejected by a journal, conference or the class just because you try to modify the format or do not follow the established publication guidelines.

The template we are providing is available from:

- <https://github.com/cloudmesh/classes/tree/master/docs/source/format/report>

Convenient compressed files are available at

- <https://github.com/cloudmesh/classes/tree/master/docs/source/format/report.tar.gz>
- <https://github.com/cloudmesh/classes/tree/master/docs/source/format/report.zip>

You will find in it a modified ACM proceedings templates for Word and for LaTeX that has an identification box removed on the lower left hand side of the first page. This is done for classes so that you have more space to write. In case you must submit to a conference you can use the original ACM template. This template can be found at

2.1.2 Submission Requirements

Although the initial requirement for some conferences or journals is the document PDF, in many cases you must be prepared to provide the source when submitting to the conference. This includes the submission of the original images in an images folder. You may be asked to package the document into a folder with all of its sources and submit to the conference for professional publication.

2.1.3 Microsoft Word vs. \LaTeX

Microsoft Word will provide you with the initial impression that you will save lots of time writing in it while you see the layout of the document. This will be initially true, but once you progress to the more challenging parts and later pages such as image management and bibliography management you will see some issues. These include that figure placement in Word needs to be done just right in order for images to be where they need to be. We have seen students spending hours with the placement of figures in a paper but when they did additional changes the images jumped around and were not at the place where students expected them to be. So if you work with images, make sure you understand how to place them. Also always use relative caption counters so that if an image gets placed elsewhere the counter stays consistent. So never use just the number, but a reference to the figure when referring to it. Recently a new bibliography management system was added

to Word. However, however it is not well documented and the references are placed in the system bibliography rather than a local managed bibliography. This may have severe consequences when working with many authors on a paper. The same is true when using Endnote. We have heard in many occasions that the combination of endnote and Word destroyed documents. You certainly do not want that to happen the day before your deadline. Also in classes we observed that those using LaTeX deliver better structured and written papers as the focus is on text and not beautiful layout.

For all these reasons we do not recommend that you use Word.

In LaTeX where we have an easier time with this as we can just ignore all of these issues due to relative good image placement and excellent support for academic reference management. Hence, it is in your best interest to use LaTeX. The information we provide here will make it easy for you to get started and write a paper in no time as it is just like filling out a form.

2.1.4 Working in a Team

Today research is done in potentially large research teams. This also includes the rewriting of a document. There are multiple ways this is done these days and depends on the system you choose.

In MSWord you can use skydrive, while for LaTeX you can use sharelatex and overleaf. However, in many cases the use of github is possible as the same groups that develop the code are also familiar with github. Thus we provide you here also with the introduction on how to write a document in github while group members can contribute.

Here are the options:

LaTeX and git: This option will likely save you time as you can use jabref also for managing collaborative bibliographies and

sharelatex: an online tool to write latex documents

overleaf: an online tool to write latex documents

MS onedrive: It allows you to edit a word document in collaboration. We recommend that you use a local installed version of Word and do the editing with that, rather than using the online version. The online editor has some bugs. See also (untested): <http://www.paulkiddie.com/2009/07/jabref-exports-to-word-2007-xml/>, <http://usefulcodes.blogspot.com/2015/01/using-jabref-to-import-bib-to-microsoft.html>

Google Drive: google drive could be used to collaborate on text that is then pasted into document. However it is just a starting point as it does not support typically the format required by the publisher. Hence at one point you need to switch to one of the other systems.

2.1.5 Timemanagement

Obviously writing a paper takes time and you need to carefully make sure you devote enough time to it. The important part is that the paper should not be an after thought but should be the initial activity to conduct and execute your research. Remember that

1. It takes time to read the information
2. It takes time understand the information
3. It takes time to do the research

For deadlines the following will get you in trouble:

1. *There are still 10 weeks left till the deadline, so let me start in 4 weeks* Procrastination is your worst enemy.
2. If you work in a team that has time management issues address them immediately

3. Do not underestimate the time it takes to prepare the final submission into the submission system. Prepare automated scripts that can deliver the package for submission in minutes rather than hours by hand.

2.1.6 Paper Checklist

In this section we summarize a number of checks that you may perform to make sure your paper is properly formatted and in excellent shape. Naturally this list is just a partial list and if you find things we should add here, let us know.

- Have you written the report in the specified format?
- Have you included an acknowledgement section?
- Have you included the paper in the submission system (In our class it is git)?
- Have you specified proper identification in the submission system. This is typically a form or ASCII text that needs to be filled out (In our case it is a README.md file that includes a homework ID, names of the authors, and e-mails)?
- Have you included all images in native and PDF format in the submission system?
- Have you added the bibliography file that you managed (In our case jabref to make it simple for you)?
- In case you used word have you also provided the jabref?
- In case of a class and if you do a multiauthor paper, have you added an appendix describing who did what in the paper?
- Have you spellchecked the paper?
- Are you using **a** and **the** properly?
- Have you made sure you do not plagiarize?
- Is the title properly capitalized?
- Have you not used phrases such as shown in the Figure below, but instead used as shown in Figure 3 when referring to the 3rd figure?
- Have you capitalized “Figure 3”, “Table 1”, ... ?
- Have you removed any figure that is not referred explicitly in the text (As shown in Figure ..)
- Are the figure captions below the figures and not on top. (Do not include the titles of the figures in the figure itself but instead use the caption or that information?)
- When using tables have you put the table caption on top?
- Make the figures large enough so we can read the details. If needed make the figure over two columns?
- Do not worry about the figure placement if they are at a different location than you think. Figures are allowed to float. If you want you can place all figures at the end of the report?
- Are all figures and tables at the end?
- In case you copied a figure from another paper you need to ask for copyright permission. IN case of a class paper you **must** include a reference to the original in the caption.
- Do not use the word “I” instead use we even if you are the sole author?
- Do not use the phrase “In this paper/report we show” instead use “We show”. It is not important if this is a paper or a report and does not need to be mentioned.
- Do not artificially inflate your paper if you are below the page limit and have nothing to say anymore.
- If your paper limit is 12 pages but you want to hand in 120 pages, please check first ;-)
- Do not use the characters & # % in the paper if you use LaTeX. If you use them you probably need a in front of them.
- If you want to say and do not use & but use the word and.
- Latex uses double single open quotes and double single closed quotes for quotes. Have you

made sure you replaced them?

- Pasting and copying from the Web often results in non ascii charaters to be used in your text, please remove them and replace accordingly.

In case of a class

- Check in your current work of the paper on a weekly basis to show consistent progress.
- Please use the dedicated report format for class. It may not be the ACM or IEEE format, but may have some additions that make management of bibliographies easier. Do follow our instructions for bibliographies.

In case you are allowed to use word in class, such as the one we teach at IU, the following applies in addition:

- Are you manageing your references in jabref and endnote (we need both)
- Are you using the right template we have a special 2 column template for the class that is a modified version from the 2 column ACM template
- Are you using build in numbered section management? MSWord has Sections that must be used
- Are you using real bulleted lists in Word and not just a "*" or a "-"?
- Have you carelessly pasted and copied into the document without using proper formats. E.g. in MSWord this is a problem. You need to fix the format and use the build in format. Not that if you paste wrong you effect the format styles.
- Have you created not only a docx document but also the PDF.
- Make sure you use .docx and not .doc

If you observe something missing let us know.

2.1.7 Example Paper

An example report in PDF format is available:

- report.pdf

2.1.8 Creating the PDF from LaTeX on your Computer

Latex can be easily installed on any computer as long as you have enough space. Furthermore if your machine can execute the make command we have provided in the standard report format a simple Makefile that allows you to do editing with immediate preview as documented in the LaTeX lesson.

2.1.9 Class Specific README.md

For the class we will manage all papers via github.com. You will be added to our github at

- <https://github.com/bigdata-i523>

and assigned an hid (homework index directory) directory with a unique hid number for you. In addition, once you decide for a project, you will also get a project id (pid) and a directory in which you place the projects. Projects must not be placed in hid directories as they are treated differently and a class proceedings is automatically created based on your submission.

As part of the hid directory, you will need to create a README.md file in it, that **must** follow a specific format. The good news is that we have developed an easy template that with common

sense you can modify easily. The template is located at

- <https://raw.githubusercontent.com/bigdata-i523/sample-hid000/master/README.md>

As the format may have been updated over time it does not hurt to revisit it and compare with your README.md and make corrections. It is important that you follow the format and not eliminate the lines with the three quotes. The text in the quotes is actually yaml. yaml is a data format the any data scientist must know. If you do not, you can look it up. However, if you follow our rules you should be good. If you find a rule missing for our purpose, let us know. We like to keep it simple and want you to fill out the *template* with your information.

Simple rules:

- replace the hid number with your hid number.
- naturally if you see sample- in the directory name you need to delete that as your directory name does not have sample- in it.
- do not ignore where the author is to be placed, it is in a list starting with a -
- there is always a space after a -
- do not introduce empty lines
- do not use TAB and make sure your editor does not by accident automatically creates tabs. This is probably the most frequent error we see.
- do not use any : & _ in the attribute text including titles
- an object defined in the README.md must have on a single type field. for example in the project section. Make sure you select only one type and delete the other
- in case you have long paragraphs you can use the > after the abstract
- Once you understood how the README.md works, please delete the comment section.
- Add a chapter topic that your paper belongs to

2.1.10 Exercise

Report.1: Install latex and jabref on your system

Report.2: Check out the report example directory. Create a PDF and view it. Modify and recompile.

Report.4: Learn about the different bibliographic entry formats in bibtex

Report.5: What is an article in a magazine? Is it really an Article or a Misc?

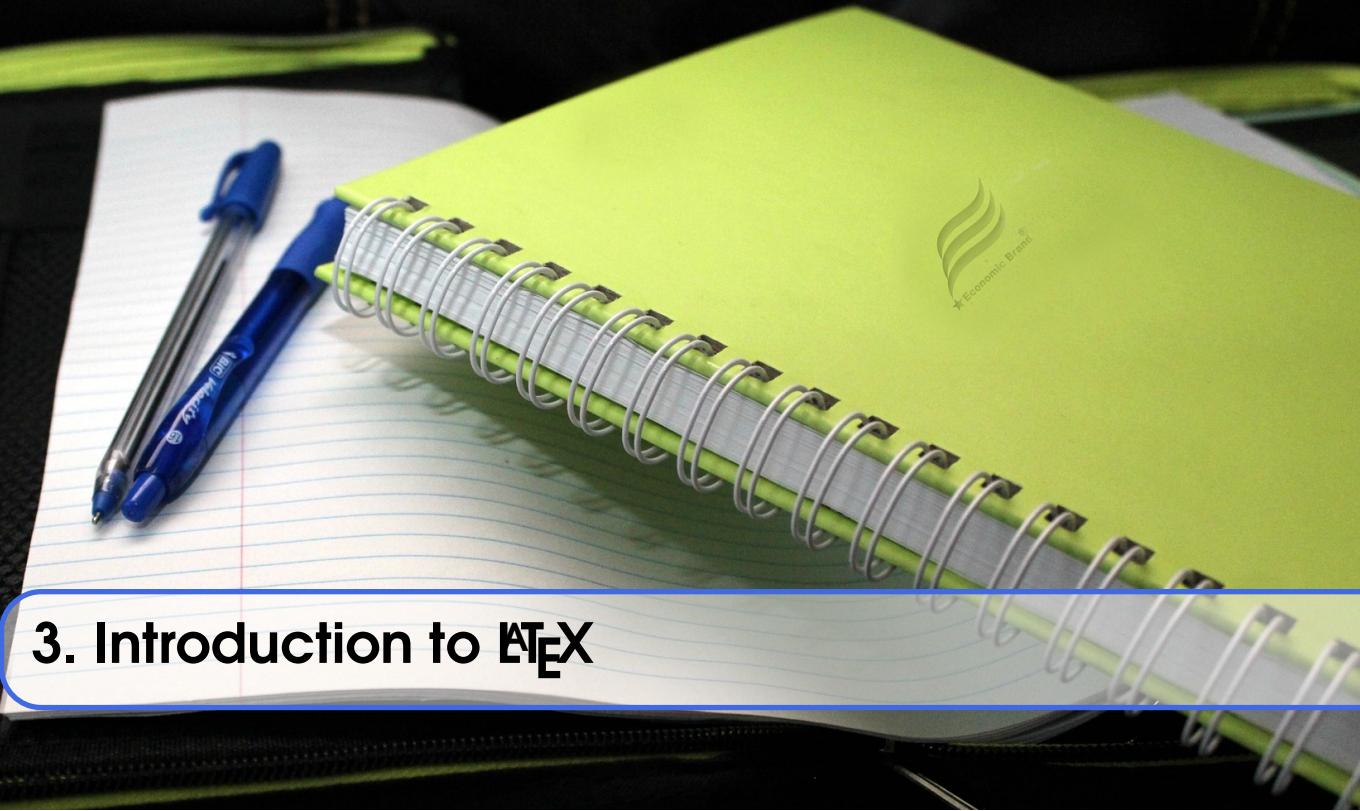
Report.6: What is an InProceedings and how does it differ from Conference?

Report.7: What is a Misc?

Report.8: Why are spaces, underscores in directory names problematic and why should you avoid using them for your projects

Report.9: Write an objective report about the advantages and disadvantages of programs to write reports.

Report.10: Why is it advantageous that directories are lowercase have no underscore or space in the name?



3. Introduction to \LaTeX

F chapter/doc/latex.tex

Mastering a text processing system is an essential part of a researcher's life. Not knowing how to use a text processing system can slow down the productivity of research drastically.

3.1 Installation

LaTeX is available on all modern computer systems. A very good installation for OSX is available at:

- <https://tug.org/mactex/>

However, if you have older versions on your systems you may have to first completely uninstall them.

3.1.1 Local Install

Installing LaTeX is trivial, and is documented on the internet very well. However, it requires sufficient space and time as it is a large environment. A system such as TeX Live takes in full install about 5.5 GB. In addition to LaTeX we recommend that you install jabref and use it for bibliography management.

Thus you will have the most of them on your system.

- pdflatex: the latex program producing pdf
- bibtex: to create bibliographies
- jabref: GUI application to bibtex files (<http://www.jabref.org/>)

Make sure you check that these programs are there, for example with the Linux commands:

```
which pdflatex
which bibtex
which jabref (on OSX you may have an icon for it)
```

If these commands are missing, please install them. For the newest documentation on installation of LaTeX we recommend you look up the installation for your specific OS.

Install on Ubuntu 16.04

The easiest way to install it on ubuntu is to use the terminal and type in (make sure you have enough space):

```
sudo apt-get install texlive-full
```

One of the best editors for LaTeX is emacs as you can also do bibliography management with it and not just LaTeX. However, other editors are available including:

- Kile, TeXworks, JLatexEditor, Gedit LaTeX Plugin, TeXMaker

Please look up how to install them if you like to use them. TeXMaker is popular, However I find the combination of emacs and latexmk superior. TeXmaker is installed with:

```
sudo apt-get install texmaker
```

Other installations:

- kile is installed by default
- <https://www.tug.org/texworks/> (Works on ubuntu, Windows, OSX)

LaTeX for OSX

- <https://www.latex-project.org/get/>

LaTeX for Windows

- <https://www.latex-project.org/get/>

3.1.2 Online Services

Sharelatex

Those that like to use latex, but do not have it installed on their computers may want to look at the following video:

Video: <https://youtu.be/PfhSOjuQk8Y>

Video with cc: <https://www.youtube.com/watch?v=8IDCGTFXoBs>

ShareLaTeX not only allows you to edit online, but allows you to share your documents in a group of up to three. Licenses are available if you need more than three people in a team.

Overleaf

Overleaf.com is a collaborative latex editor. In its free version it has a very limited disk space. However it comes with a Rich text mode that allows you to edit the document in a preview mode. The free templates provided do not include ACM template, but you are allowed to use the OSA template.

Features of overleaf are documented at: <https://www.overleaf.com/benefits>

Paperia

We do not know where this service is located. However it offers similar services as ShareLatex and Overleaf.

- <https://papeeria.com/>

3.2 Basic LaTeX Elements

Often researchers may be initially overwhelmed with all the features that \LaTeX provides. However, it is much simpler than you initially believe. In Chapter ?? we introduced you towards using an article template. As a template is provided you can just look at the elements in that article and modify or copy them while adapting the content. Thus, it is more like filling out a form. You do not have to learn much and you can learn as you go. We are providing in this chapter some basic \LaTeX elements that will help you getting started quickly while serving you as a reminder what how to do certain things in \LaTeX .

3.2.1 Characters

\LaTeX is a command language and as such uses some special characters as part of the language. Thus if you want to use these characters either in your text or bibliography you need to be especially careful about. These characters include % \$ # _

Other than in hypref links and urls you need to put a backslash in front of them. For example to print a % in the text you need to use:

\%

Furthermore the character " is not at all used as discussed in the next section.

3.2.2 Highlighting Text

Quotes are not written with the " character, but are embedded in two left single quotes and two right single quotes:

‘‘This is a quote’’

which will result in:

“This is a quote”

In many papers we see that the quote is misused while putting quotes around a word. However quotes are often just used to quote a text from another paper. Instead of using quotes authors may actually emphasize a word. \LaTeX has a special command for that using:

{\em this is emphasized}

resulting in

this is emphasized

To write a text as bold (which should also be avoided as bold is typically used in section headers), you can use:

{\em this is bold fett}

resulting in

this is bold fett

3.2.3 Sections

LaTeX provides a convenient mechanism to structure a paper with sections and subsections. This is achieved with the following commands:

```
\section{This is a Section}
\subsection{This is a Subsection}
\subsubsection{This is a Subsubsection}
```

Once you use one of these commands the next paragraph will start below the section command.

In addition you have the command:

```
\paragraph{This is a paragraph.}
```

The line is behind the paragraph heading

The command is special as it does not introduce a new line between the Heading and the next line even if you include empty lines

3.2.4 Empty Lines

Multiple empty lines will be reduced to a single empty line.

3.2.5 Itemize

Itemized lists can be written as:

```
\begin{itemize}
    \item First item
    \item Second item
\begin{itemize}
```

resulting in

- First item
- Second item

3.2.6 Enumerate

Enumerations can be written as:

```
\begin{enumerate}
    \item First item
    \item Second item
\begin{enumerate}
```

resulting in

1. First item
2. Second item

3.2.7 Descriptions

Description lists can be written as:

```
\begin{itemize}
    \item[Cloud] My definition of a Cloud.
    \item[Big Data] My definition of Big Data
\begin{itemize}
```

Cloud: My definition of a Cloud

Big Data: My definition of Big Data

3.2.8 Images

Figures are extremely easy to handle by including them from source. We never worry about the placement as LaTeX does typically a very good job of doing this.:

In Figure \ref{F:graph} we show a black and white graph about

```
\begin{figure}
    \includegraphics[width=\columnwidth]{images/graph.pdf}
    \caption{A sample black and white graphic. \cite{las17graph}}
    \label{F:graph}
\end{figure}
```

Note that las17graph must be a label of a valid bibtex entry. This is needed if you have copied the image from elsewhere to avoid plagiarism. However, if you came up with the graph yourself than you do not need a citation.

We recommend that you place in your paper drafts all images at the which can be done with the endfloat package

This can be enabled if you include the following lines before begin document command:

```
\usepackage{endfloat}
\renewcommand{\efloatseparator}{\mbox{}}
```



```
\begin{document}
```

3.2.9 Tables

tables from csv tables by hand

3.2.10 Labels

As we saw already for figures and tables it is recommended to use the label and ref commands to refer to figure or table numbers. This applies also to sections. Thus I can place a label after a section:

```
\section{Introduction}\label{S:introduction}
```

and write elsewhere in the paper:

```
As we showcased in Section \ref{S:introduction}
```

Furthermore to conveniently distinguish sections tables and figures, we use the prefix S T F followed

by a colon for the label. This helps organizing your paper in case you have many labels.

3.2.11 Mathematics

One of the strength of LaTeX is the ability to write easily sophisticated mathematical expressions on paper with high quality. A good online resource is provided by the following online resource from which we have copied some examples:

- <https://en.wikibooks.org/wiki/LaTeX/Mathematics>

Exponents are using the \wedge character:

```
$ (a + b)^2 = a^2 + 2ab + b^{c+2} $
```

$$(a - b)^2 = a^2 - 2ab + b^2$$

Greek letters are referred to by their name proceeded by the slash:

```
$ \alpha, \Alpha, \beta, \Beta, \gamma, \Gamma, \pi, \Pi, \phi $
```

Limits can be written as follows:

```
$ \lim_{x \rightarrow \infty} \exp(-x) = 0 $
```

$$\lim_{x \rightarrow \infty} \exp(-x) = 0$$

Fractions are indicated by the `frac` command, and binomials by `binom`:

```
$ \frac{n!}{k!(n-k)!} = \binom{n}{k} $
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Matrices can be created as follows:

```
A_{m,n} =
\begin{pmatrix}
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

3.3 Advanced topics

3.3.1 ACM and IEEE Proceedings Format

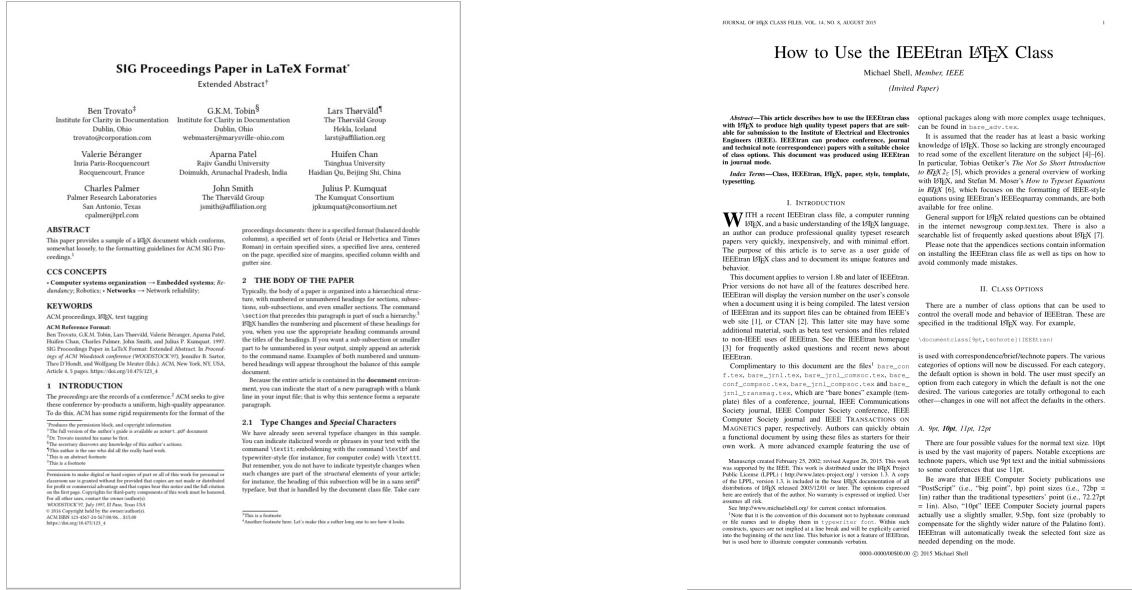


Figure 3.1: The look of the ACM and IEEE format templates

- <http://www.acm.org/publications/proceedings-template>
- https://www.ieee.org/conferences_events/conferences/publishing/templates.html

3.3.2 Generating and Managing Images

To produce high quality images the programs PowerPoint and omnigraffle on OSX are recommended. When using powerpoint please keep the image ratio to 4x3 as they produce nice size graphics which you also can use in your presentations. When using other ratios they may not fit in presentations and thus you may increase unnecessarily your work. We do not recommend vizio as it is not universally available and produces images that in case you have to present them in a slide presentation does not easily reformat if you do not use 4x3 aspect ratio.

Naturally, graphics should be provided in SVG or PDF format so they can scale well when we look at the final PDF. Including PNG, gif, or jpeg files often do not result in the necessary resolution or the files become real big. For this reason we for example can also not recommend tools such as tableau as they do not provide proper exports to high quality publication formats. For interactive display such tool may be good, but for publications it produces inferior formatted images.

We recommend that all images be stored into a folder called images in the same directory where your LATEXmain document resides.

3.3.3 Slides

Slides are best produced with the seminar package:

```
\documentclass{seminar}
```

```
\begin{slide}

Hello World on slide 1

\end{slide}
```

The text between slides is ignored

```
\begin{slide}

Hello World on slide 2

\end{slide}
```

However, in case you need to have a slide presentation we recommend you use ppt. Just paste and copy content from your PDF or your LaTeX source file into the ppt.

3.3.4 Useful Online Information about \LaTeX

Latex Sheet: <https://wch.github.io/latexsheet/latexsheet.pdf>

Latex Short: <http://tug.ctan.org/info/lshort/english/lshort.pdf>

Wikibook: <https://en.wikibooks.org/wiki/LaTeX>

Wikibook (PDF) : <https://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf>

Links to books: <https://latexforhumans.wordpress.com/2008/10/11/the-best-guides-to-latex/>

Links to books: <https://www.latex-project.org/help/books/>

LaTeX2e: The LaTeX Reference Manual provides a good introduction to Latex.

- LaTeX Users and Reference Guide, by Leslie Lamport https://www.amazon.com/LaTeX-Document-Preparation-System/dp/0201529831/ref=sr_1_2?s=books&ie=UTF8&qid=1507114870&sr=1-2&keywords=lamport
- LaTeX an Introduction, by Helmut Kopka https://www.amazon.com/Guide-LaTeX-4th-Helmut-Kopka/dp/0321173856/ref=pd_lpo_sbs_14_t_0?_encoding=UTF8&pse=1&refRID=2BB4APDFEX34A4JM65ZB
- The LaTeX Companion, by Frank Mittelbach <https://www.amazon.com/LaTeX-Companion-Techniques-Computer-Typesetting/dp/0201362996>

3.3.5 LaTeX vs. X

We will refrain from providing a detailed analysis on why we use LaTeX in many cases versus other technologies. In general, we find that LaTeX:

- is incredibly stable
- produces high-quality output
- is platform independent
- has lots of templates
- has been around for many years so it works well
- removes you from the pain of figure placements
- focusses you on content rather than the appearance of the paper
- integrates well with code repositories such as git to write collaborative papers.
- has superior bibliography integration
- has a rich set of tools that make using LaTeX easier
- authors do not play with layouts much so papers in a format are uniform

In case you need a graphical view to edit LaTeX or LateX exportable files you also find AucTeX and Lyx.

Word

Word is arguably available to many, but if you work on Linux you may be out of luck. Also Word often focusses not on structure of the text but on its appearance. Many students abuse Word and the documents in Word become a pain to edit with multiple users. Recently Microsoft has offered online services to collaborate on writing documents in groups which work well. Integration with bibliography managers such as endnote or Mendeley is possible.

However, we ran into issues whenever we use word:

- Word tends sometimes to crash for unknown reasons and we lost a lot of work
- Word has some issues with the bibliography managers and tends to crash sometimes for unknown reasons.
- Word is slow with integration to large bibliographies.
- Figure placement in Word in some formats is a disaster and you will spend many hours to correct things just to find out that if you make small changes you have to spend additional many hours to get used to the new placement. We have not yet experienced a word version where we have not lost images. Maybe that has changed, so let us know

However, we highly recommend the collaborative editing features of Word that work on a paragraph and not letter level. Thus saving is essential so you do not block other people from editing the paragraph.

Google Docs

Unfortunately, many useful features got lost in the new google docs. However, it is great to collaborate quickly online, share thoughts and even write your latex documents together if you like (just copy your work in a file offline and use latex to compile it ;-))

The biggest issue we have with Google Docs is that it does not allow the support of 2 column formats, that the bibliography integration is non-existent and that paste and copy from web pages and images encourages unintended plagiarism when collecting information without annotations (LaTeX and Word are prone to this too, but we found from experience that it tends to happen more with Google docs users).

A Place for Each

When looking at the tools we find a place for each:

Google docs: Short meeting notes, small documents, quick online collaborations to develop documents collaboratively at the same time.

Word: Available to many, supports 2 column format, supports paragraph based collaborative editing, Integrates with bibliography managers.

LaTeX: Reduces failures, great offline editing, superior bibliography management, superior image placement, runs everywhere. Great collaborative editing with sharelatex, allows easy generation of proceedings written by hundreds of people with shared index.

The best choice for your class: LaTeX

3.4 Editing

3.4.1 Emacs

The text editor emacs provides a great basis for editing TeX and LaTeX documents. Both modes are supported. In addition there exists a color highlight module enabling the color display of LaTeX and TeX commands. On OSX aquamacs and carbon emacs have build in support for LaTeX. Spell checking is done with flyspell in emacs.

Aquamacs

Aquamacs is an editor based on GNU Emacs that runs on OSX and integrates with the OSX desktop. This is for many the preferred editor on OSX for \LaTeX .

<http://aquamacs.org>

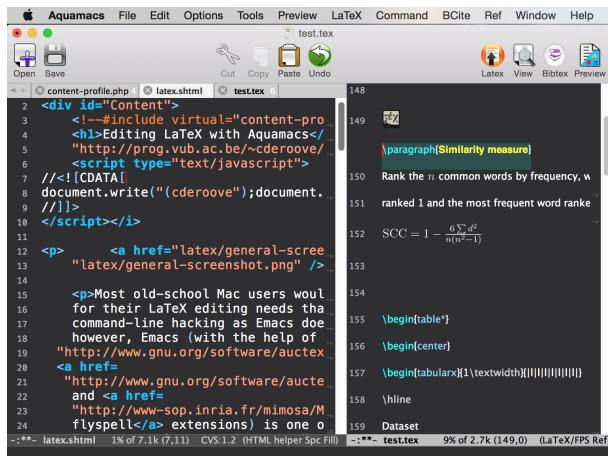


Figure 3.2: Aquamacs

3.4.2 Vi/Vim

Another popular editor is vi or vim. It is less feature rich but many programmers ar using it. As it can edit ASCII text you can edit LaTeX. With the LaTeX add-ons to vim, vim becomes similar powerful while offering help and syntax highlighting for LaTeX as emacs does. (The authors still prefer emacs)

3.4.3 TeXshop

Other editors such as TeXshop are available which provide a more integrated experience. However, we find them at times to stringent and prefer editors such as emacs.

3.4.4 LyX

We have made very good experiences with Lyx. You must assure that the team you work with uses it consistently and that you all use the same version.

Using the ACM templates is documented here:

- <https://wiki.lyx.org/Examples/AcmSiggraph>

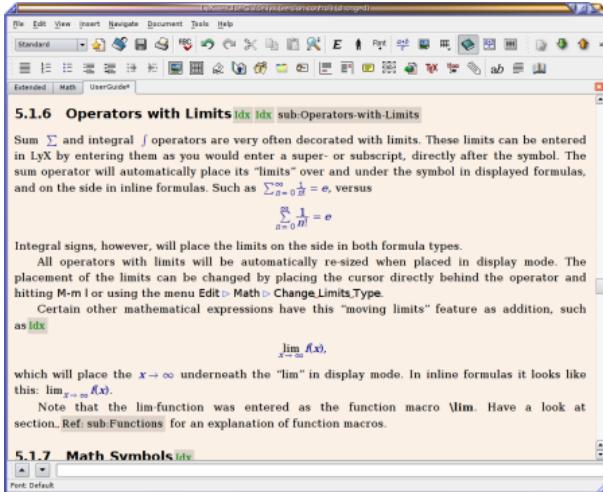


Figure 3.3: Lyx

On OSX it is important that you have a new version of LaTeX and Lyx installed. As it takes up quite some space, you may want to delete older versions. The new version of LyX comes with the acmsigplan template included. However on OSX and other platforms the .cls file is not included by default. However the above link clearly documents how to fix this.

3.4.5 WYSIWYG locally

We have found that editors such as Lyx and Auctex provide very good WYSIWYG alike features. However, we found an even easier way while using skim, a pdf previewer, in conjunction with emacs and latexmk. This can be achieved while using the following command assuming your latex file is called ‘report.tex’:

```
latexmk -pvc -view=pdf report
```

This command will update your pdf previewer (make sure to use skim) whenever you edit the file report.tex and save it. It will maintain via skim the current position, thus you have a real great way of editing in one window, while seeing the results in the other.

Skim can be found at: <http://skim-app.sourceforge.net/>

3.4.6 Markdown and L^AT_EX

It may come as a surprise to many that one can actually write simple LaTeX documents also in markdown Syntax or mix section written in markdown while others are written in LaTeX. To do so all you have to do is place the markdown text in a separate file. Let us call the file content.rst which has the following lines included in it:

```
# Section

* item a
* item b
```

Obviously, we would have to convert this to LaTeX. Luckily there is a very useful program called *pandoc* that does this for you. You could make the translation in the shell, but you could also make the translation locally on your computer while allowing L^AT_EX to start up external programs. This

is achieved with the `write18` command and allowing \LaTeX explicitly to call external programs. Please inspect the following `latex` file that includes a template on how to do this. We assume the file is called `markdown.tex` for our example.

```
\documentclass{article}

\include{graphicx}
\newcommand{\tightlist}{{}{}}

\begin{document}
\immediate\write18{pandoc content.md -o content.tex}

\input{markdown-latex}

\end{document}
```

Now to generate the PDF we simply have to call the following command that include the `-shell-escape` flag to allow the execution of `write18` embedded commands:

```
pdflatex -shell-escape markdown-test
```

The output will be `markdown.pdf` with the content from the `markdown` file translated. Doing this naturally allows you to write large portions in `markdown` and automatically include them in your \LaTeX document. Hence, you can use editors such as Macdown to initially work in semi WYSIWYG mode and do fairly straight forward edition. Naturally the same can be done in RST. Naturally the most elementary features are supported. For more sophisticated features, please use \LaTeX directly.

3.4.7 pyCharm

TODO: comment on how we can use pycharm for editing and what the limitations are.

3.4.8 MSWord

it is possible to use Word.

be careful with

3.5 The \LaTeX Cycle

To create a PDF file from `latex` yo need to generate it following a simple development and improvement cycle.

First, Create/edit ASCII source file with `file.tex` file:

```
emacs file.tex
```

Create/edit bibliography file:

```
jabref refs.bib
```

Create the PDF:

```
pdflatex file
bibtex file
```

```
pdflatex file  
pdflatex file
```

View the PDF:

```
open file
```

It not only showcases you an example file in ACM 2 column format, but also integrates with a bibliography. Furthermore, it provides a sample Makefile that you can use to generate view and recompile, or even autogenerate. A compilation would look like:

```
make  
make view
```

If however you want to do things on change in the tex file you can do this automatically simply with:

```
make watch
```

for make watch its best to use skim as pdf previewer

3.6 Tips

Including figures over two columns:

- <http://tex.stackexchange.com/questions/30985/displaying-a-wide-figure-in-a-two-column>
- positioning figures with textwidth and columnwidth https://www.sharelatex.com/learn/Positioning_images_and_tables
- An organization as the author. Assume the author is National Institute of Health and want to have the author show up, please do:
`key= {National Institute of Health},
author= {{National Institute of Health}},
Please note the {{ }}`
- words containing ‘fi’ or ‘ffi’ showing blank places like below after recompiling it: find as nd efficiency as e ciency
You copied from word or PDF ff which is actually not an ff, but a condensed character, change it to ff and ffi, you may find other such examples such as any non ASCII character. A degree is for example another common issue in data science.
- do not use l & and other latex characters in bibtex references, instead use , and the word and
- If you need to use _ it is _ but if you use urls leave them as is
- We do recommend that you use sharelatex and jabref for writing papers. This is the easiest solution and beats in many cases MSWord as you can focus on writing and not on formatting.



4. Managing Bibliographies

F <https://github.com/cloudmesh/classes/blob/master/docs/source/lesson/doc/bibtex.rst>

4.0.1 Integrating Bibliographies

LaTeX integrates very well with bibtex. There are several pre-formatted styles available. It includes also styles for ACM and IEEE bibliographies. For the ACM style we recommend that you replace abbrv.bst with abbrvurl.bst, add hyperref to your usepackages so you can also display URLs in your citations:

```
\bibliographystyle{IEEEtran}
\bibliography{references.bib}
```

Then you have to run latex and bibtex in the following order:

```
latex file
bibtex file
latex file
latex file
```

or simply call make from our makefile.

The reason for the multiple execution of the latex program is to update all cross-references correctly. In case you are not interested in updating the library every time in the writing progress just postpone it till the end. Missing citations are viewed as [?].

Two programs stand out when managing bibliographies: emacs and jabref:

- <http://www.jabref.org/>

Other programs such as Mendeley, Zotero, and even endnote integrate with bibtex. However their support is limited, so we recommend that you just use jabref. Furthermore its free and runs on all

platforms.

jabref

Jabref is a very simple to use bibliography manager for LaTeX and other systems. It can create a multitude of bibliography file formats and allows upload in other online bibliography managers.

- Installation: Go to <http://www.jabref.org/> and click download
- Video: <https://youtu.be/cMtYOHCHZ3k>
- Video with cc: <https://www.youtube.com/watch?v=QVbifcLgMic>

jabref and MSWord

According to others it is possible to integrate jabref references directly into MSWord. This has been conducted so far however only on a Windows computer.

We have not tried this ourselves, but give it as a potential option.

Here are the steps the need to be done:

1. Create the Jabref bibliography just like in presented in the Jabref video
2. After finishing adding your sources in Jabref, click File -> export
3. Name your bibliography and choose MS Office 2007(*.xml) as the file format. Remember the location of where you saved your file.
4. Open up your word document. If you are using the ACM template, go ahead and remove the template references listed under Section 7. References
5. In the MS Word ribbon choose ‘References’
6. Choose ‘Manage Sources’
7. Click ‘Browse’ and locate/select your Jabref xml file
8. You should now see your references appear in the left side window. Select the references you want to add to your document and click the ‘copy’ button to move them from the left side window to the right window.
9. Click the ‘Close’ button
10. In the MS Word Ribbon, select ‘Bibliography’ under the References tab
11. Click ‘Insert Bibliography’ and your references should appear in the document
12. Ensure references are of Style: IEEE. Styles are located in the References tab under ‘Manage Sources’

As you can see there is significant effort involve, so we do recommend you use LaTeX as you can focus there on content rather than dealing with complex layout decisions. This is especially true, if your papers have figures or tables, or you need to add references.

4.1 Entry types

In this section we will explain how to find and properly generate bibliographic entries. We are using bibtex for this as it is easy to use and generates reasonable entries that can be included in papers. What we like to achieve in this section is not to just show you a final entry, but to document the process on how that entry was derived. This will allow you to replicate or learn from the process to apply to your own entries.

We will address a number of important entry types which includes:

- wikipedia entries
- github entries

- books
- articles in a scientific journal
- articles in a conference
- articles in magazines (non scientific)
- blogs

4.1.1 Source code References

We will learn how to cite a source code from a publicly hosted repository. Such repositories are frequently used and include, for example github, bitbucket, sourcefore, or your Universities code repository as long as it is publicly reachable. As changes can occur on these repositories, it is important that the date of access is listed in the entry or even the release version of the source code.

Let us without bias chose a random source code entry that has been contributed by a student as follows:

```
@Misc{gonzalez_2015,
  Title = {Buildstep},
  Author = {Gonzalez, Jose and Lindsay, Jeff},
  HowPublished = {Web Page},
  Month = {Jul},
  Note = {Accessed: 2017-1-24},
  Year = 2015,
  Key = {www-buildstep},
  Url = {https://github.com/progium/buildstep}
}
```

Is this entry correct? Let us analyse.

Entry type Misc

First, it seems appropriate to use a `@misc` entry. We correctly identify this is a misc entry as it is online available. More recent version of bibtex include also the type `@online` for it. However, in order to maintain compatibility to older formats we chose simply `Misc` here and if we really would need to we could replace it easily

Label

Typically the Label should contain 3 letters from an author name, short year and the short name of the publication to provide maximum information regarding the publication. Underscores need to be replaced by dashes or removed. However as this is a github repository it is better to integrate this into the label. Hence, we simply use the github-projectname (in our case `github-buildstep`, out of convention we only use lower case letters).

Author

Unless the last name contains spaces, it should be first name followed by the last name with multiple authors separated with “and”.

Key

In this case the key field can be removed as the entry has an author field entry. If there was no author field, we could use key to specify the alphabetical ordering based on the specified key. Note

that a key is not the label. In fact in our original entry the key field was wrongly used and the student did not understand that the key is used for sorting.

Howpublished

Since the source is a github project repository, the howpublished field shall hold the value {Code Repository} rather than a web page. If the url specified was a normal webpage, the {Web Page} entry would be valid.

Month

The lowercase month is, used for international notation since months are not capitalized in some other languages.

Owner

In class we introduced the convention to put the student HID in it. If multiple students contributed, add them with space separation.

Accessed

As we do not yet typically an accessed field, we simply include it in the note field. This is absolutely essential as code can change and when we read the code we looked at a particular snapshot in time. In addition it is often necessary to record the actual version of the code. Typically this can also be done with the month and year field while relying on a release date

Final Entry

Filling out as many fields as possible with information for this entry we get:

```
@Misc{github-buildstep,
  Title = {Buildstep},
  Author = {Jose Gonzalez and Jeff Lindsay},
  HowPublished = {Code Repository},
  Year = {2015},
  Month = jul,
  Note = {Accessed: 2017-1-24},
  Url = {https://github.com/program/buildstep},
  Owner = {S17-I0-3025},
}
```

We are using the release date in the year and month field as this project uses this for organizing releases. However, other project may have release versions so you would have in addition to using the data also to include the version in the note field such as:

```
Note = {Version: 1.2.3, Accessed: 2017-1-24},
```

All those that helped should add your HID to this entry with a space separated from each other

4.1.2 Researching proper bibtex entries

Article in a journal

Many online bibtex entries are wrong or incomplete. Often you may find via google a bibtex entry that may need some more research. Lets assume your first google query returns a publication and

you cite it such as this:

```
@Unpublished{unpublished-google-sawzall,
  Title = {{Interpreting the Data: Parallel Analysis with Sawzall}},
  Author = {Rob Pike, Sean Dorward, Robert Griesemer, Sean Quinlan},
  Note = {accessed 2017-01-28},
  Month = {October},
  Year = {2005},
  Owner = {for the purpose of this discussion removed},
  Timestamp = {2017.01.31}
}
```

Could we improve this entry to achieve your best? We observe:

1. The author field has a wrong entry as the , is to be replaced by an and.
2. The author feild has authors and thus must not have a {{ }}
3. The url is missing, as the simple google search actually finds a PDF document.

Let us investigate a bit more while searching for the title. We find

- A) https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwj_ytSA-PDRAhUH8IMKHaomC-oQFggaMAA&url=https%3A%2F%2Fresearch.google.com%2Farchive%2Fsawzall-sciprog.pdf&usg=AFQjCNHSSfKBwbxVAVPQ0td4rTjitKucpA&sig2=vbiVzi36B3gGFjIzlUKBDA&bvm=bv.1
- B) <https://research.google.com/pubs/pub61.html>
- C) <http://dl.acm.org/citation.cfm?id=1239658>

Let us look at A)

As you can see from the url this is actually some redirection to a google web page which probably is replaced by B as its from google research. So let us look at B)

Now when you look at the link we find the url <https://research.google.com/archive/sawzall-sciprog.pdf> which redirects you to the PDF paper.

When we go to B) we find surprisingly a bibtex entry as follows:

```
@article{61,
  title = {Interpreting the Data: Parallel Analysis with Sawzall},
  author = {Rob Pike and Sean Dorward and Robert Griesemer and Sean Quinlan},
  year = 2005,
  URL = {https://research.google.com/archive/sawzall.html},
  journal = {Scientific Programming Journal},
  pages = {277–298},
  volume = {13}
}
```

Now we could say lets be satisfied, but C) seems to be even more interesting as its from a major publisher. So lats just make sure we look at C)

If you go to C, you find under the colored box entitled Tools and Resources a link called **bibtex**. Thus it seems a good idea to click on it. This will give you:

```
@article{Pike:2005:IDP:1239655.1239658,
  author = {Pike, Rob and Dorward, Sean and Griesemer, Robert and Quinlan, Sean},
  title = {Interpreting the Data: Parallel Analysis with Sawzall},
  journal = {Sci. Program.},
  issue_date = {October 2005},
```

```

volume = {13},
number = {4},
month = oct,
year = {2005},
issn = {1058-9244},
pages = {277--298},
numpages = {22},
url = {http://dx.doi.org/10.1155/2005/962135},
doi = {10.1155/2005/962135},
acmid = {1239658},
publisher = {IOS Press},
address = {Amsterdam, The Netherlands, The Netherlands},
}

```

Now we seem to be at a position to combine our search result as neither entry is sufficient. As the doi number properly specifies a paper (look up what a doi is) we can replace the url with one that we find online, such as the one we found in A) Next we see that all field sin B are already covered in C, so we take C) and add the url. Now as the label is great and uniform for ACM, but for us a bit less convenient as its difficult to remember, we just change it while for example using authors, title, and year information. lets also make sure to do mostly lowercase in the label just as a convention. Thus our entry looks like:

```

@article{pike05swazall,
author = {Pike, Rob and Dorward, Sean and Griesemer, Robert and Quinlan, Sean},
title = {Interpreting the Data: Parallel Analysis with Sawzall},
journal = {Sci. Program.},
issue_date = {October 2005},
volume = {13},
number = {4},
month = oct,
year = {2005},
issn = {1058-9244},
pages = {277--298},
numpages = {22},
url = {https://research.google.com/archive/sawzall-sciprog.pdf},
doi = {10.1155/2005/962135},
acmid = {1239658},
publisher = {IOS Press},
address = {Amsterdam, The Netherlands, The Netherlands},
}

```

As you can see properly specifying a reference takes multiple google queries and merging of the results you find from various returns. As you still have time to correct things I advise that you check your references and correct them. If the original reference would have been graded it would have been graded with a “fail” instead of a “pass”.

4.1.3 Article in a conference proceedings

Lets look at a second obvious example that needs improvement:

```

@InProceedings{wettinger-any2api,
Title          = {Any2API - Automated APIfication},

```

```

Author          = {Wettinger, Johannes and
                 Uwe Breitenb{"u}cher
                 and Frank Leymann},
Booktitle      = {Proceedings of the 5th International
                 Conference on Cloud Computing and
                 Services Science},
Year           = {2015},
Pages          = {475486},
Publisher      = {SciTePress},
ISSN           = {2326-7550},
Owner          = {S17-I0-3005},
Url            = {https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf}
}

```

As you can see this entry seems to define all required fields, so we could be tempted to stop here. But its good to double check. Lets do some queries against ACM, . and google scholar, so we just type in the title, and if this is in a proceedings they should return hopefully a predefined bibtex record for us.

Lets query:

```
google: googlescholar Any2API Automated APIfication
```

We get:

- https://scholar.google.de/citations?view_op=view_citation&hl=en&user=j6lIXt0AAAAJ&citation_for_view=j6lIXt0AAAAJ:8k81kl-MbHgC

On that page we see Cite

So we find a PDF at <https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf>

Lets click on this and the document includes a bibtex entry such as:

```
@inproceedings{Wettinger2015,
  author= {Johannes Wettinger and Uwe Breitenb{"u}cher and Frank
           Leymann},
  title = {Any2API - Automated APIfication},
  booktitle = {Proceedings of the 5th International Conference on Cloud
              Computing and Service Science (CLOUDSER)},
  year = {2015},
  pages = {475--486},
  publisher = {SciTePress}
}
```

Now lets add the URL and owner:

```
@inproceedings{Wettinger2015,
  author= {Johannes Wettinger and Uwe Breitenb{"u}cher and Frank
           Leymann},
  title = {Any2API - Automated APIfication},
  booktitle = {Proceedings of the 5th International Conference on Cloud
              Computing and Service Science (CLOUDSER)},
```

```

year = {2015},
pages = {475--486},
publisher = {SciTePress},
url ={https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf},
owner = {S17-I0-3005},
}

```

Should we be satisfied? No, even our original information we gather provided more information. So lets continue. Lets googlesearch different queries with ACM or IEEE and the title. When doing the IEEE in the example we find an entry called

dlp: Frank Leyman

Lets look at it and we find two entries:

```

@inproceedings{DBLP:conf/closer/WettingerBL15,
author    = {Johannes Wettinger and
             Uwe Breitenb\"{u}cher and
             Frank Leymann},
title     = {{\{ANY2API\}} - Automated APIfication - Generating APIs for Executables
             to Ease their Integration and Orchestration for Cloud Application
             Deployment Automation},
booktitle = {{\{CLOSER\}} 2015 - Proceedings of the 5th International Conference on
             Cloud Computing and Services Science, Lisbon, Portugal, 20-22 May,
             2015.},
pages     = {475--486},
year      = {2015},
crossref  = {DBLP:conf/closer/2015},
url       = {http://dx.doi.org/10.5220/0005472704750486},
doi       = {10.5220/0005472704750486},
timestamp = {Tue, 04 Aug 2015 09:28:21 +0200},
biburl   = {http://dblp.uni-trier.de/rec/bib/conf/closer/WettingerBL15},
bibsource = {dblp computer science bibliography, http://dblp.org}
}

@proceedings{DBLP:conf/closer/2015,
editor    = {Markus Helfert and
             Donald Ferguson and
             V{\'e}ctor M{\'e}ndez Mu{\~n}oz},
title     = {{\{CLOSER 2015 - Proceedings of the 5th International Conference on
             Cloud Computing and Services Science, Lisbon, Portugal, 20-22 May,
             2015\}}},
publisher = {SciTePress},
year      = {2015},
isbn      = {978-989-758-104-5},
timestamp = {Tue, 04 Aug 2015 09:17:34 +0200},
biburl   = {http://dblp.uni-trier.de/rec/bib/conf/closer/2015},
bibsource = {dblp computer science bibliography, http://dblp.org}
}

```

So lets look at the entry and see how to get a better one for our purpose to combine them. When using jabref, you see optional and required fields, we want to add as many as possible, regardless if

optional or required, so Lets do that (I I write here in ASCII as easier to document:

```
@InProceedings{,
    author = {},
    title = {},
    OPTcrossref = {},
    OPTkey = {},
    OPTbooktitle = {},
    OPTyear = {},
    OPTeditor = {},
    OPTvolume = {},
    OPTnumber = {},
    OPTseries = {},
    OPTpages = {},
    OPTmonth = {},
    OPTaddress = {},
    OPTorganization = {},
    OPTpublisher = {},
    OPTnote = {},
    OPTannote = {},
    url = {}
}
```

So lets copy and fill out the **form** from our various searches:

```
@InProceedings{Wettinger2015any2api,
    author = {Johannes Wettinger and
              Uwe Breitenb\\"{u}cher and
              Frank Leymann},
    title = {{ANY2API - Automated APIfication - Generating APIs for Executables
              to Ease their Integration and Orchestration for Cloud Application
              Deployment Automation}},
    booktitle = {{CLOSER 2015 - Proceedings of the 5th International Conference on
                  Cloud Computing and Services Science}},
    year = {2015},
    editor = {Markus Helfert and
              Donald Ferguson and
              V{\'e}ctor M{\'e}ndez Mu{\~n}oz},
    publisher = {SciTePress},
    isbn = {978-989-758-104-5},
    pages = {475--486},
    month = {20-22 May},
    address = {Lisbon, Portugal},
    doi = {10.5220/0005472704750486},
    url = {https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf},
    owner = {S17-I0-3005},
}
```

4.1.4 What are the differnt entry types and fields

We were asked what are the different entry types and fields, so we did a google query and found the following useful information. please remember that we also have fields such as doi, owner, we will add status ={pass/fail} at time of grading to indicate if the reference passes or fails. We may assign this to you so you get familiar with the identification if a refernece is ok or not.

Please see <https://en.wikipedia.org/wiki/BibTeX>

4.1.5 InProceedings

Please fill out

```
@InProceedings{,
    author =      {},
    title =       {},
    OPTcrossref = {},
    OPTkey =      {},
    OPTbooktitle = {},
    OPTyear =     {},
    OPTeditor =   {},
    OPTvolume =   {},
    OPTnumber =   {},
    OPTseries =   {},
    OPTpages =    {},
    OPTmonth =    {},
    OPTaddress =  {},
    OPTorganization = {},
    OPTpublisher = {},
    OPTnote =     {},
    OPTannote =   {},
    url =        {}
}

@inproceedings{vonLaszewski15tas,
    author =      {DeLeon, Robert L. and Furlani, Thomas R. and Gallo,
                  Steven M. and White, Joseph P. and Jones, Matthew
                  D. and Patra, Abani and Innus, Martins and Yearke,
                  Thomas and Palmer, Jeffrey T. and Sperhac, Jeanette
                  M. and Rathsam, Ryan and Simakov, Nikolay and von
                  Laszewski, Gregor and Wang, Fugang},
    title =       {{TAS View of XSEDE Users and Usage}},
    booktitle =   {Proceedings of the 2015 XSEDE Conference: Scientific
                  Advancements Enabled by Enhanced
                  Cyberinfrastructure},
    series =     {XSEDE '15},
    year =       2015,
    isbn =       {978-1-4503-3720-5},
    location =   {St. Louis, Missouri},
    pages =      {21:1--21:8},
    articleno =  21,
```

```

numpages = 8,
url = {http://doi.acm.org/10.1145/2792745.2792766},
doi = {10.1145/2792745.2792766},
acmid = 2792766,
publisher = {ACM},
address = {New York, NY, USA},
keywords = {HPC, SUPReMM, TAS, XDMoD, XSEDE usage, XSEDE users},
}

```

4.1.6 TechReport

Please fill out

```

@TechReport{,
author = {},
title = {},
institution = {},
year = {},
OPTkey = {},
OPTtype = {},
OPTnumber = {},
OPTaddress = {},
OPTmonth = {},
OPTnote = {},
OPTannote = {},
url = {}
}

@TechReport{las05exp,
title = {{The Java CoG Kit Experiment Manager}},
Author = {von Laszewski, Gregor},
Institution = {Argonne National Laboratory},
Year = 2005,
Month = jun,
Number = {P1259},
url = {https://laszewski.github.io/papers/vonLaszewski-exp.pdf}
}

```

4.1.7 Article

Please fill out

```

@Article{,
author = {},
title = {},
journal = {},
year = {},
OPTkey = {},
OPTvolume = {},
OPTnumber = {},
OPTpages = {}
}

```

```

OPTmonth =      {},
OPTnote =      {},
OPTannote =    {}.,
url = {}

}

@Article{las05gridhistory,
  title =  {{The Grid-Idea and Its Evolution}},
  author = {von Laszewski, Gregor},
  journal = {Journal of Information Technology},
  year =   2005,
  month =  jun,
  number = 6,
  pages =  {319-329},
  volume = 47,
  doi =    {10.1524/itit.2005.47.6.319},
  url =   {https://laszewski.github.io/papers/vonLaszewski-grid-idea.pdf}
}

```

4.1.8 Proceedings

Please fill out

```

@Proceedings{,
  title =      {},
  year =       {},
  OPTkey =     {},
  OPTbooktitle = {},
  OPTeditor =   {},
  OPTvolume =   {},
  OPTnumber =   {},
  OPTseries =   {},
  OPTaddress =  {},
  OPTmonth =    {},
  OPTorganization = {},
  OPTpublisher = {},
  OPTnote =     {},
  OPTannote =   {},
  url =        {}
}

@Proceedings{las12fedcloud-proc,
  title =  {{FederatedClouds '12: Proceedings of the 2012
             Workshop on Cloud Services, Federation, and the 8th
             Open Cirrus Summit}},
  year =   2012,
  address = {New York, NY, USA},
  editor =  {vonLaszewski, Gregor and Robert Grossman and Michael
             Kozuch and Rick McGeer and Dejan Milojicic},
  publisher = {ACM},
  ISBN =   {978-1-4503-1754-2},
}

```

```

location = {San Jose, California, USA},
url =
    {http://dl.acm.org/citation.cfm?id=2378975&picked=prox&cfid=389635474&cft
}

```

4.1.9 Wikipedia Entry

Please fill out

```

@Misc{,
OPTkey = {},
OPTauthor = {},
OPTtitle = {},
OPThowpublished = {},
OPTmonth = {},
OPTyear = {},
OPTnote = {},
OPTannote = {},
url = {}
}

@Misc{www-ode-wikipedia,
Title = {Apache ODE},
HowPublished = {Web Page},
Note = {Accessed: 2017-2-11},
Key = {Apache ODE},
Url = {https://en.wikipedia.org/wiki/Apache_ODE}
}

```

4.1.10 Blogs

Please fill out

```

@Misc{,
OPTkey = {},
OPTauthor = {},
OPTtitle = {},
OPThowpublished = {},
OPTmonth = {},
OPTyear = {},
OPTnote = {},
OPTannote = {},
OPTurl = {}
}

@Misc{www-clarridge-discoproject-blog,
title = {Disco - A Powerful Erlang and Python Map/Reduce
Framework},
author = {Clarridge, Tait},
howpublished = {Blog},
month = may,
}

```

```

note = {Accessed: 25-feb-2017},
year = 2014,
url = {http://www.taitclaridge.com/techlog/2014/05/disco-a-powerful-erlang-and-python-m}
}

```

4.1.11 Web Page

Please fill out

```

@Misc{,
OPTkey = {},
OPTauthor = {},
OPTtitle = {},
OPThowpublished = {},
OPTmonth = {},
OPTyear = {},
OPTnote = {},
OPTannote = {},
url = {}
}

@Misc{www-cloudmesh-classes,
OPTkey = {},
author = {von Laszewski, Gregor},
title = {Cloudmesh Classes},
howpublished = {Web Page},
OPTmonth = {},
OPTyear = {},
OPTnote = {},
OPTannote = {},
url = {https://cloudmesh.github.io/classes/}
}

@Misc{www-awslambda,
title = {AWS Lambda},
author = {{Amazon}},
key = {AWS Lambda},
howpublished = {Web Page},
url = {https://aws.amazon.com/lambda/faqs/}
}

```

4.1.12 Book

Given the following entry. What is the proper entry for this book. Provide rationale:

```

@Book{netty-book,
Title = {Netty in Action},
Author = {Maurer, Norman and Wolfthal, Marvin},
Publisher = {Manning Publications},
Year = {2016},
}

```

To obtain the record of a book you can look at many information sources. The can include:

- <https://www.manning.com/books/netty-in-action>
- <https://www.amazon.com/Netty-Action-Norman-Maurer/dp/1617291471>
- <http://www.barnesandnoble.com/w/netty-in-action-norman-maurer/1117342155?ean=9781617291470#productInfoTabs>
- <http://www.powells.com/book/netty-in-action-9781617291470/1-0>

Furthermore, we need to consider the entry of a book, we simply look it up in emacs where we find the following but add the owner and the url field:

```
@Book{,
  ALTAuthor =      {},
  ALTEditor =     {},
  title =        {},
  publisher =    {},
  year =         {},
  OPTkey =       {},
  OPTvolume =    {},
  OPTnumber =    {},
  OPTseries =    {},
  OPTaddress =   {},
  OPTedition =   {},
  OPTmonth =     {},
  OPTnote =      {},
  OPTannote =    {},
  ownwer =       {},
  url =          {}
}
```

In summary we find the following fields:

Required fields: author/editor, title, publisher, year

Optional fields: volume/number, series, address, edition, month, note, key

We apply the following to fill out the fields.

address: The address is the Publisher's address. Usually just the city, but can be the full address for lesser-known publishers.

author: The name(s) of the author(s) (in the case of more than one author, separated by and) Names can be written in one of two forms: Donald E. Knuth or Knuth, Donald E. or van Halen, Eddie. Please note that Eddie van Halen would result in a wrong name. For our purpose we keep nobelity titles part of the last name.

edition: The edition of a book, long form (such as "First" or "Second")

editor: The name(s) of the editor(s)

key: A hidden field used for specifying or overriding the alphabetical order of entries (when the "author" and "editor" fields are missing). Note that this is very different from the key that is used to cite or cross-reference the entry.

label: The label field should contain three letters from the auth field, a short year reference and a short name of the publication to provide the maximum information regarding the publication. Underscores should be replaced with dashes or removed completely.

month: The month of publication or, if unpublished, the month of creation. Use three-letter abbreviations for this field in order to account for languages that do not capitalize month

names. Additional information for the day can be included as follows: aug #“~10,”

publisher: The publisher’s name

series: The series of books the book was published in (e.g. “The Hardy Boys” or “Lecture Notes in Computer Science”)

title: The title of the work. As the capitalization depends on the bibliography style and the language used we typically use camel case. To force capitalization of a word or its first letter you can use the curly braces, ‘{ }’. To keep the title in camel case simple use title = { {My Title} }

type: The field overriding the default type of publication (e.g. “Research Note” for techreport, “{PhD} dissertation” for phdthesis, “Section” for inbook/incollection) volume The volume of a journal or multi-volume book year The year of publication (or, if unpublished, the year of creation)

While applying the above rules and tips we summarize what we have done for this entry:

1. Search for the book by title/Author on ACM (<http://dl.acm.org/>) or Amazon or barnes-andnoble or upcitemdb (<http://upcitemdb.com>). These services return bibtex entries that you can improve.
2. Hence one option is to get the ISBN of the book. For “Mesos in action” from upcitemdb we got the ISBN as “9781617 292927”. This is the 13 digit ISBN. The first 3 digits (GS1 code) can be skipped. Using the rest of 10 digits “1617 292927”, Add in JabRef in Optional Fields->ISBN.

However it is fine to just specify the full number.

We can also return a bibtex entry generated while using Click on the “Get BibTex from ISBN”.

Now we get more information on this book entry from ISBN. We can opt either the original or newly searched entry for the below bibtex fields or merge as appropriate. URL may not match from where we initially read the book, however there is option to put your original url or newly searched url. EAN, Edition, Pages,url,published date etc. Do a search on amazon for “ASIN”. Can skip if not available. Sometime we get ASIN for a different publication, maybe a paperback ASIN={B01MT311CU} We can add it as it becomes easier to search

doi: If you can find a doi number you should also add it. In this case we could not locate one.

As a result we obtain the entry:

```
@Book{netty-book,
  title = {Netty in Action},
  publisher = {Manning Publications Co.},
  year = {2015},
  author = {Maurer, Norman and Wolfthal, Marvin Allen},
  address = {Greenwich, CT, USA},
  edition = {1st},
  isbn = {1617291471},
  asin = {1617291471},
  date = {2015-12-23},
  ean = {9781617291470},
  owner = {S17-I0-3022 S17-I0-3010 S17-I0-3012},
  pages = {296},
  url = {http://www.ebook.de/de/product/21687528/norman_maurer_netty_in_action.html},
}
```

4.2 Integrating Bibtex entries into Other Systems

We have not tested any of this

4.2.1 Bibtex import to MSWord

XML import

Please respond back to us if you have used this and give feedback.

1. In JabRef, export the bibliography in MS Word 2008 xml format
2. Name the file Sources.xml (case sensitive)
3. In OSX with MS Word 2015: Go to /Library/Containers/com.microsoft.word/Data/Library/Application Support/Microsoft Word/
4. Rename the original Sources.xml file to Sources.xml.bak
5. Copy the generated Sources.xml in this folder
6. Restart MS Word.

We do not know what needs to be done in case you need to make changes to the references. Please report back your experiences. To avoid issues we recommend that you use LaTeX, and not MSWord.

BibTeX4Word

We have not tried this:

- <http://www.ee.ic.ac.uk/hp/staff/dmb/perl/index.html>

You are highly recommended to use Jabref for bibliography management in this class. Here is an introductory video on Jabref: <https://youtu.be/roi7vezNmfo?t=8m6s>

4.3 Other Reference Managers

Please note that you should first decide which reference manager you like to use. In case you for example install zotero and mendeley, that may not work with word or other programs.

4.3.1 Endnote

Endnote is a reference manager that works with Windows. Many people use Endnote. However, in the past, Endnote has caused complications when dealing with collaborative management of references. Its price is considerable. We have lost many hours of work because of instability of Endnote in some cases. As a student, you may be able to use Endnote for free at Indiana University.

- <http://endnote.com/>

4.3.2 Mendeley

Mendeley is a free reference manager compatible with Windows Word 2013, Mac Word 2011, LibreOffice, BibTeX. Videos on how to use it are available at:

- <https://community.mendeley.com/guides/videos>

Installation instructions are available at

- <https://www.mendeley.com/features/reference-manager/>

When dealing with large databases, we found the integration of Mendeley into word slow.

4.3.3 Zotero

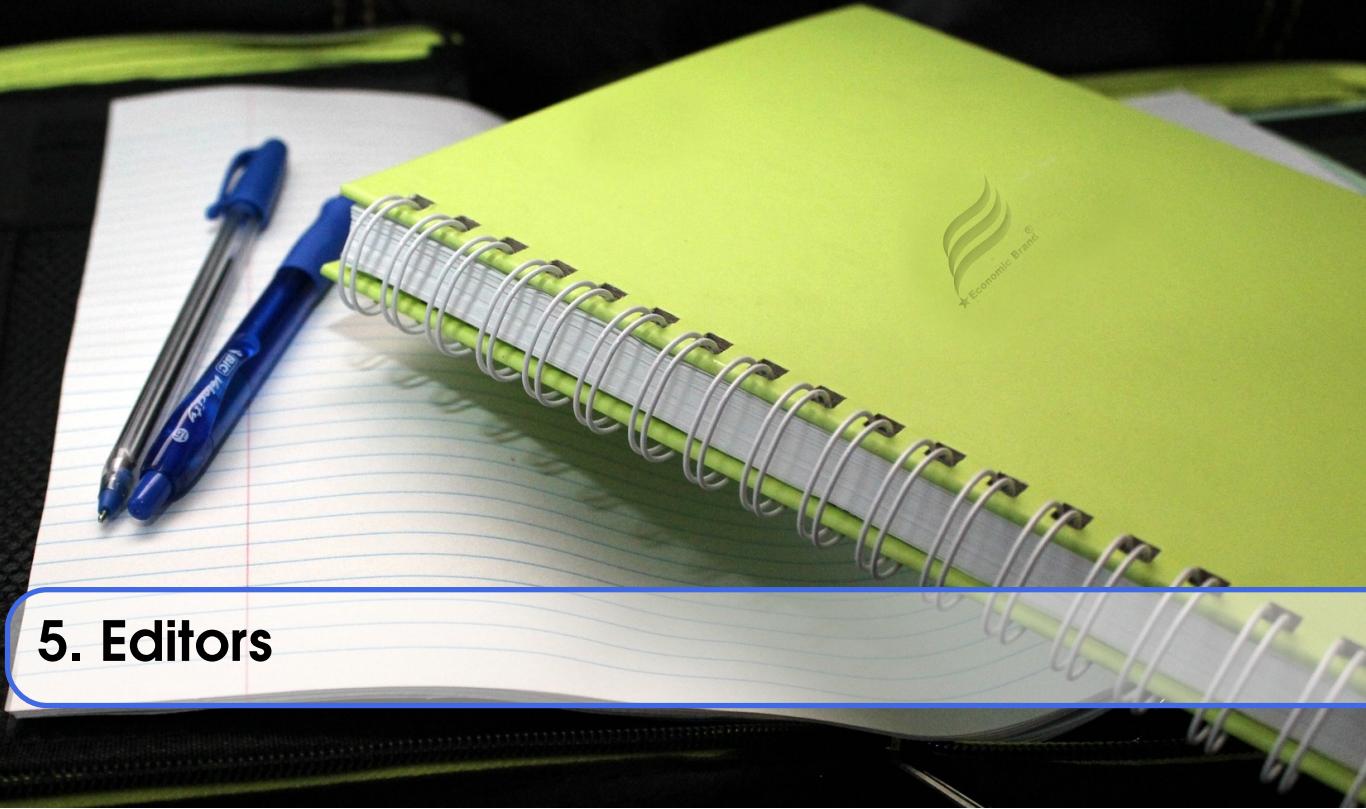
Zotero is a free tool to help you collect, organize, cite, and share your research sources. Documentation is available at

- <https://www.zotero.org/support/>

The download link is available from

- <https://www.zotero.org/>

We have limited experience with Zotero



5. Editors

F

<https://github.com/cloudmesh/classes/blob/master/docs/source/lesson/doc/emacs.rst>

5.1 Basic Emacs

One of the most useful short manuals for emacs is the following reference card. It takes some time to use this card efficiently, but the most important commands are written on it. Generations of students have literally been just presented with this card and they learned emacs from it.

- <https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf>

There is naturally also additional material available and a great manual. You could also look at

- <https://www.gnu.org/software/emacs/tour/>

From the last page we have summarized the most useful and **simple** features. And present them here. One of the hidden gems of emacs is the ability to recreate replayable macros which we include here also. You ought to try it and you will find that for data science and the cleanup of data emacs (applied to smaller datasets) is a gem.

Notation

Key	Description
C	Control
M	Esc (meta character)

In the event of an emergency...

Here's what to do if you've accidentally pressed a wrong key:

If you executed a command and Emacs has modified your buffer, use C-/ to undo that change. If you pressed a prefix key (e.g. C-x) or you invoked a command which is now prompting you for input (e.g. Find file: ...), type C-g, repeatedly if necessary, to cancel. C-g also cancels a long-running operation if it appears that Emacs has frozen.

Moving around in buffers can be done with cursor keys, or with the following key combinations:

Key	Description
C-f Forw	ard one character
C-n Next	line
C-b Back	one character
C-p Prev	ious line

Here are some ways to move around in larger increments:

Key	Description
C-a Begi	nning of line
M-f Forw	ard one word
M-a Prev	ious sentence
M-v Prev	ious screen
M-< Begi	nning of buffer
C-e End	of line
M-b Back	one word
M-e Next	sentence
C-v Next	screen
M-> End	of buffer

You can jump directly to a particular line number in a buffer:

Key	Description
M-g g	Jump to specified line

Searching is easy with the following commands

Key	Description
C-s Incr	emental search forward
C-r Incr	emental search backward

Replace

Key	Description
M-% Quer	y replace

Killing (“cutting”) text

Key	Description
C-k	Kill line

Yanking

Key	Description
C-y	Yank s last killed text

Macros

Keyboard Macros

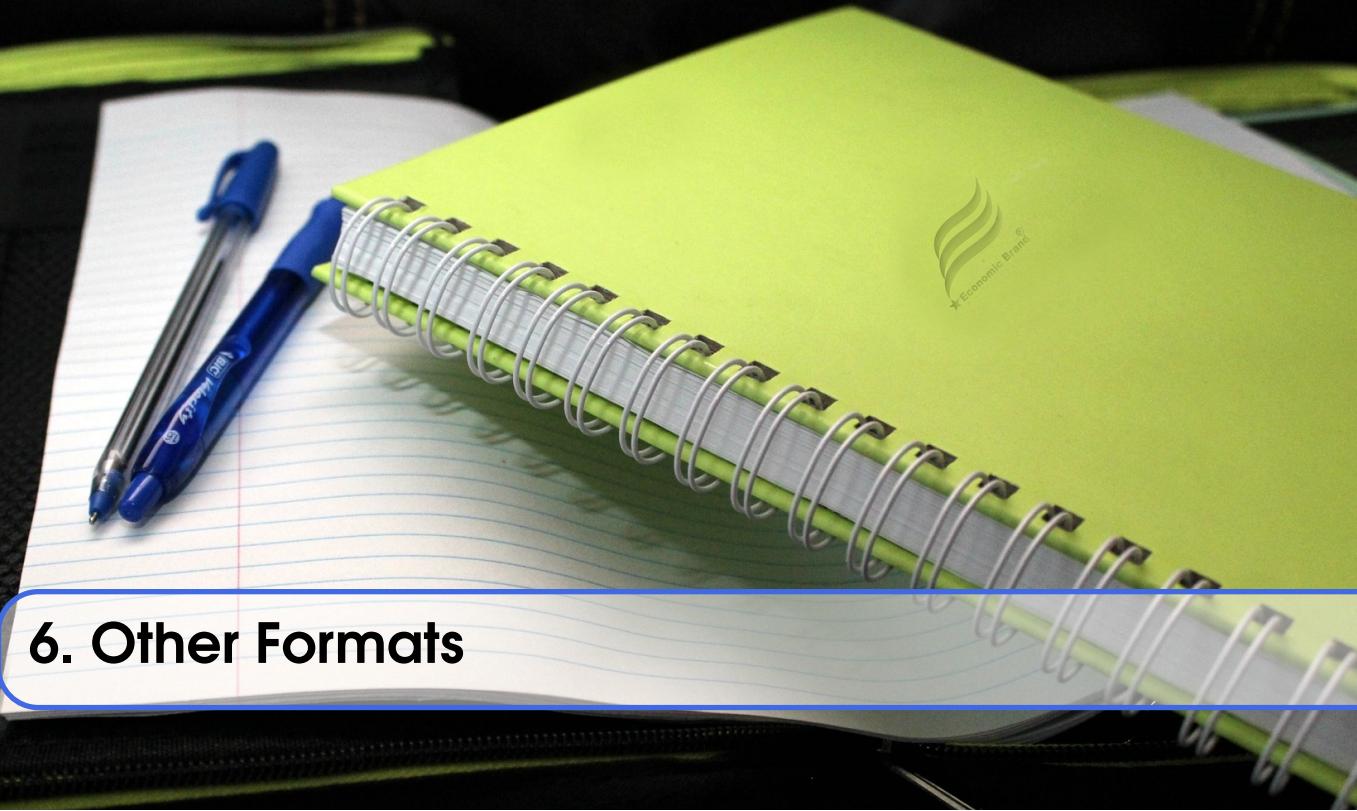
Keyboard macros are a way to remember a fixed sequence of keys for later repetition. They're handy for automating some boring editing tasks.

Key	Description
M-x (Start recording macro
M-x)	Stop recording macro
M-x e	Play back macro once
M-5 M-x-e	Play back macro 5 times

Modes

“Every buffer has an associated major mode, which alters certain behaviors, key bindings, and text display in that buffer. The idea is to customize the appearance and features available based on the contents of the buffer.” modes are typically activated by ending such as .py, .java, .rst, ...

Key	Description
M-x python-mode	Mode for editing Python files
M-x auto-fill-mode	Wraps your lines automatically when they get longer than 70 characters.
M-x flyspell-mode	Highlights misspelled words as you type.



6. Other Formats

F

<https://github.com/cloudmesh/classes/blob/master/docs/source/lesson/doc/rst.rst>

6.1 reStructuredText

reStructuredText (RST) purpose is to provide an easy-to-read, what-you-see-is-what-you-get plaintext markup syntax and parser system. With its help you can develop documentation not only for stand alone documentation, simple web pages, an in-line program documentation (such as Python). RST is extensible and new features can be added. It is used in sphinx as one of its supported formats.

6.1.1 Links

- RST Sphinx documentation: <http://www.sphinx-doc.org/en/stable/rest.html>
- RST Syntax: <http://docutils.sourceforge.net/rst.html>
- Important extensions: <http://sphinx-doc.org/ext/todo.html>

Cheatcheat:

- <http://github.com/ralsina/rst-cheatsheet/raw/master/rst-cheatsheet.pdf>
- <http://docutils.sourceforge.net/docs/ref/rst/directives.html>

6.1.2 Source

The source for this page is located at

- <https://raw.githubusercontent.com/cloudmesh/classes/master/docs/source/lesson/doc/rst.rst>

This way you can look at the source on how we create this page.

6.1.3 Sections

with overline, for parts * with overline, for chapters =, for sections -, for subsections ^, for subsubsections ", for paragraphs

RST allows to specify a number of sections. You can do this with the various underlines:

```
*****
Chapter
*****
Section
=====
Subsection
-----
Subsubsection
-----
Paragraph
~~~~~~
```

6.1.4 Listtable

```
.. csv-table:: Eye colors
  :header: "Name", "Firstname", "eyes"
  :widths: 20, 20, 10

  "von Laszewski", "Gregor", "gray"
```

6.1.5 Exceltable

we have integrated Excel table from <http://pythonhosted.org//sphinxcontrib-exceltable/> into our sphinx allowing the definition of more elaborate tables specified in excel. However the most convenient way may be to use list-tables. The documentation to list tables can be found at <http://docutils.sourceforge.net/docs/ref/rst/directives.html#list-table>

6.1.6 Boxes

Seealso

```
.. seealso:: This is a simple **seealso** note.
```

Note

This is a **note** box.

```
.. note:: This is a **note** box.
```

Warning

note the space between the directive and the text

```
.. warning:: note the space between the directive and the text
```

Others

This is an **attention** box.

```
.. attention:: This is an **attention** box.
```

This is a **caution** box.

```
.. caution:: This is a **caution** box.
```

This is a **danger** box.

```
.. danger:: This is a **danger** box.
```

This is a **error** box.

```
.. error:: This is a **error** box.
```

This is a **hint** box.

```
.. hint:: This is a **hint** box.
```

This is an **important** box.

```
.. important:: This is an **important** box.
```

This is a **tip** box.

```
.. tip:: This is a **tip** box.
```

6.1.7 Sidebar directive

It is possible to create sidebar using the following code:

```
.. sidebar:: Sidebar Title  
:subtitle: Optional Sidebar Subtitle
```

Subsequent indented lines comprise
the body of the sidebar, and are
interpreted as body elements.

Sidebar Title: Optional Sidebar Subtitle

Subsequent indented lines comprise the body of the sidebar, and are interpreted as body elements.

6.1.8 Sphinx Prompt

```
.. prompt:: bash, cloudmesh$
```

```
    wget -O cm-setup.sh http://bit.ly/cloudmesh-client-xenial  
    sh cm-setup.sh
```

6.1.9 Programm examples

You can include code examples and bash commands with two colons.

This is an example for python:

```
print ("Hallo World")
```

This is an example for a shell command:

```
$ ls -lisa
```

6.1.10 Hyperlinks

Direct links to html pages can be done with:

```
'This is a link to an html page <hadoop.html>'_
```

Note that this page could be generated from an rst page

Links to the FG portal need to be formulated with the portal tag:

```
:portal:'List to FG projects </projects/all>'
```

In case a subsection has a link declared you can use :ref: (this is the preferred way as it can be used to point even to subsections):

```
:ref:'Connecting private network VMs clusters <_s_vpn>'
```

A html link can be created anywhere in the document but must be unique. for example if you place:

```
.. _s_vpn:
```

in the text it will create a target to which the above link points when you click on it

6.1.11 Todo

```
.. todo:: an example
```

Todo

an example



<https://github.com/cloudmesh/classes/blob/master/docs/source/lesson/doc/markdown.rst>

6.2 Markdown

TBD. Section about Markdown

see: <https://en.wikipedia.org/wiki/Markdown>



<https://github.com/cloudmesh/classes/blob/master/docs/source/lesson/doc/type.rst>

6.3 Communicating Research in Other Ways

Naturally, writing papers is not the only way to communicate your research with others. We find that today we see additional pathways for communication including blogs, twitter, facebook, e-mail, Web pages, and electronic notebooks. Let us revisit some of them and identify when they are helpful.

6.3.1 Blogs

blog: noun, a regularly updated website or web page, typically one run by an individual or small group, that is written in an informal or conversational style.

Advantages:

- encourages spontaneous posts
- encourages small short contributions
- chronologically ordered
- standard software exists to set up blogs
- online services exist to set up blogs

Disadvantages:

- structuring data is difficult (some blog software support it)
- not suitable for formal development of a paper
- often lack of sophisticated track change features
- no collaborative editing features

6.3.2 Sphinx

Sphinx (<http://www.sphinx-doc.org/>) is a tool that creates integrated documentation from a markup language while.

Advantages:

- output formats: html, LaTeX, PDF, ePub
- integrates well with directory structure
- powerful markup language (reStructuredText)
- can be hosted on GitHub via GitHub pages
- can integrate other renderers such as Markdown
- automatic table of contents, table of index
- code documentation integration
- search
- written in Python and using bash, so extensions and custom automation are possible

Disadvantage:

- requires compile step
- When using markdown GitHub can render individual page

Others:

- Read the Docs (<https://readthedocs.org/>)
- Doxygen (<http://www.stack.nl/~dimitri/doxygen/>)
- MkDocs (<http://www.mkdocs.org/>)

6.3.3 Notebooks

Jupyter

The Jupyter Notebook (<http://jupyter.org/>) is an open-source web application allowing users to create and share documents that contain live code, equations, visualizations and explanatory text. Use cases include data cleaning and transformation, numerical simulation, statistical modeling, machine learning.

Advantages:

- Integrates with python
- Recently other programming languages have been integrated
- Allows experimenting with settings
- Allows a form of literate programming while mixing documentation with code
- automatically renders on github
- comes with web service that allows hosting

Disadvantage:

- mostly encourages short documents
- mark up language is limited
- editing in ASCII is complex and Web editing is preferred

Apache Zeppelin

A Web-based notebook that enables data-driven, interactive data analytics and collaborative documents with SQL, Scala and hadoop. It integrates a web-based notebook with data ingestion, data exploration, visualization, sharing and collaboration features to Hadoop and Spark.

Advantages:

- integration to various framework
- Web framework
- integration with spark, hadoop

Disadvantages:

- larger framework
- must leverages existing deployments of spark, hadoop