Computer Practicum 1

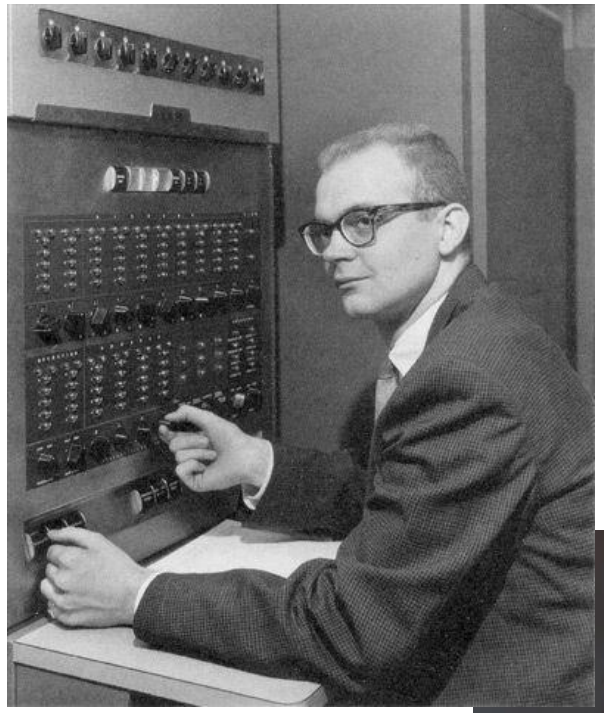
Introduction to LaTeX

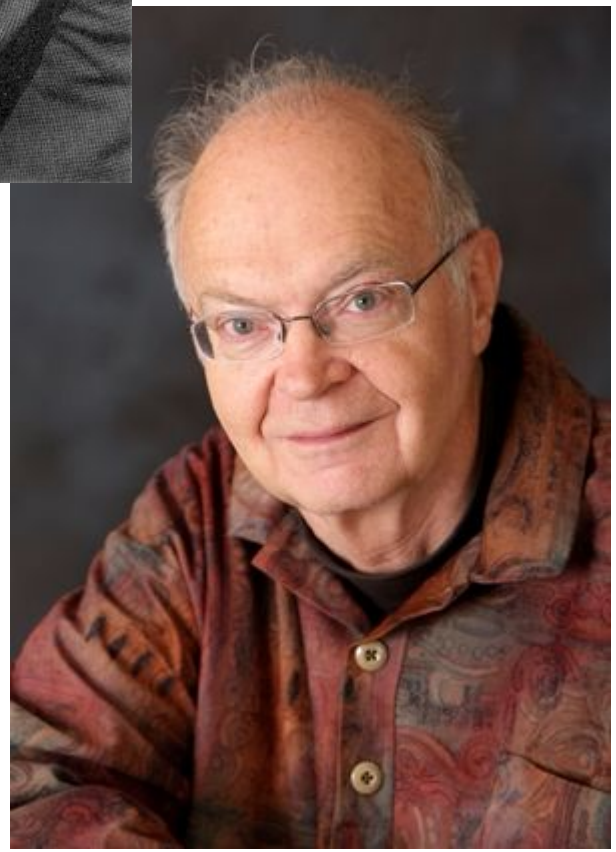
Jordan Deja
@jrdndj

Based on LaTeX tutorial by Claudio Vellage and the Latex for Beginners workbook (University of Edinburgh).

Let’s test your skills.

https://play.typeracer.com/





Donald Ervin Knuth, PhD (1938-)

"Father" of software engineering.

Author of the ***bible*** for all computers: **The Art of Computer Programming**

Creator of **TeX** system for document preparation.

What is LaTeX?

- LaTeX is a **document preparation system** for professional-looking documents.
- suited for making long, structured documents.
- It is very good at typesetting **equations**.
- It is available as **free software** for most operating systems.
- LaTeX was created by **Leslie Lamport** (thus the term Lampert's TeX) in 1985, is based on TeX a typesetting system designed by **Donald E. Knuth** in 1978 for high quality digital typesetting.

Why LaTeX?

- Microsoft Word is ‘What You See Is What You Get’ (WYSIWYG) type of editor.
- With LaTeX you do not see how the final document will look while you are typing it - **this allows you to concentrate on the content rather than appearance**.
- A LaTeX file has **.tex** file extension.
- As you type, you **mark** the document structure (title, chapters, subheadings, lists etc.) **with tags**. It’s like HTML but for documents.

Online LaTeX editor

https://www.overleaf.com/

My first LaTex file - Hello World!

```
\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```

A backslash \ tells LaTeX this is not an actual text you want to see printed in your document but instead is an instruction or command for the LaTeX compiler. All commands share the structure:

```
\commandname{option}
\begin{document}
  ...
\end{document}
```

Text, images, tables etc. That you want to actually see printed in the document, should be put between the `\begin{document}` and `\end{document}` statements. This defines an environment – an area of the document where certain typesetting rules apply.

My first LaTex file - environments

```
\begin{document}
  \begin{environment1}
    \begin{environment2}
    \end{environment2}
  \end{environment1}
\end{document}
%Invalid:
\begin{document}
  \begin{environment1}
    \begin{environment2}
  \end{environment1}
    \end{environment2}
\end{document}
```

```
\begin{env1}
  ...
\end{env2}
```

It is possible (and usually necessary) to have multiple environments in a document.

The `document` environment is always the topmost environment.

There are numerous different environments and we will need them for inserting mathematical formulas and figures to the document.

It is possible to define your own environments, but it is very likely that the environment you desire already exists. LaTeX itself comes with a few predefined environments.

My first LaTex file - title page

this is a random
godzilla image

```
\documentclass{article}
\title{My first document}
\date{19 November 2021}
\author{Godzilla}
\begin{document}
  \maketitle
  \newpage
  Hello World!
\end{document}
```

The area before our main document is called ***preamble***.
In this specific example we use it to set up the values for the `\maketitle` command for later use in our document.
This command will create a titlepage for us.
The `\newpage` command speaks for itself.
If you don‘t want a number at the bottom of the first page, You can do so by adding the `\pagenumbering{gobble}` command and then changing it back to `\pagenumbering{arabic}` on the next page numbers:

```
\begin{document}
  \pagenumbering{gobble}
  \maketitle
  \newpage
  \pagenumbering{arabic}
    Hello World!
\end{document}
```

My first LaTex file - sections

```
\documentclass{article}
 \title{My first document}
 \date{20 November 2021}
 \author{Gospod Godzilla}
 \begin{document}
   \maketitle
    \pagenumbering{gobble}
   \newpage
   \pagenumbering{arabic}
   \section{Section}
   Hello World!
   \subsection{Subsection}
   Structuring a document is easy!
 \end{document}
```

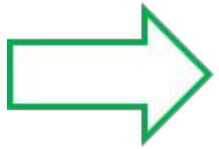
LaTeX offers commands to structure the content of the documents into logic units.

```
\section{}
\subsection{}
\subsubsection{}
\paragraph{}
\subparagraph{}
```

Using the `\section{}` command it generates section headings and numbers them automatically.
The *`section`* commands are numbered and will appear in The table of contents of your document. *`Paragraphs`* aren't numbered and won't show in the table of contents.

My first LaTex file - sections (continued)

```
\documentclass{article}
\begin{document}
  \section{Section}
  Hello World!
  \subsection{Subsection}
  Structuring a document is easy!
  \subsubsection{Subsubsection}
  More text.
  \paragraph{Paragraph}
  Some more text.
  \subparagraph{Subparagraph}
  Even more text.
  \section{Another section}
\end{document}
```



1 Section

Hello World!

1.1 Subsection

Structuring a document is easy!

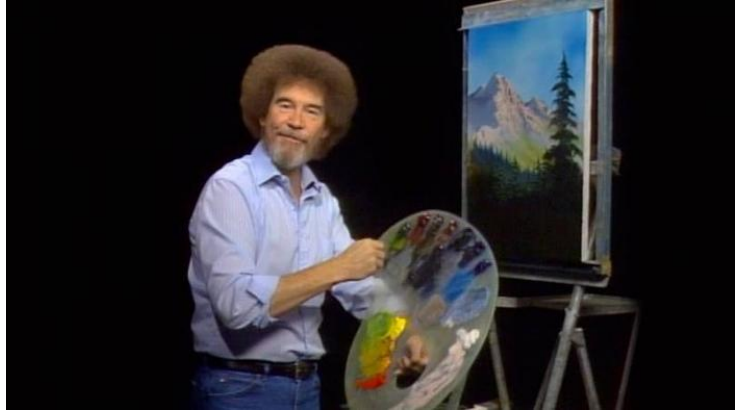
1.1.1 Subsubsection

More text.

Paragraph Some more text.

Subparagraph Even more text.

2 Another section



My first LaTex file - packages

```
\documentclass{article}
\usepackage{PACKAGENAME}
\begin{document}
...
\end{document}
```

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
  \begin{equation*}
    f(x) = x^2
  \end{equation*}
```

LaTeX offers a lot of functions by default, but in some situations it can become in handy to use so called *packages*.
To import a package in LaTeX, add the `\usepackage` command to the *preamble* of your document.

To typeset math, LaTeX offers (among others) an ***environment*** called ***equation***. Everything inside this environment will be printed in *math mode*, a special typesetting environment for math.

LaTeX automatically numbers the equations in the text. But if you don‘t want to have the equations automatically numbered, you have to include a package that allows you to do that.

My first LaTex file - math features

```
...
This formula $f(x) = x^2$ is
an example.
...
```

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
  \begin{equation*}
    1 + 2 = 3
  \end{equation*}
  \begin{equation}
    1 = 3 - 2
  \end{equation}
  \begin{align*}
    1 + 2 &= 3\\
    1 &= 3 - 2
  \end{align*}
\end{document}
```

There are two major modes of typesetting math in LaTeX:

- embedding the math directly into your text by ***encapsulating*** a formula in ***dollar signs*** and
- using a predefined ***math environment***.

The most useful *math environments* are:

- ***equation*** *environment* for typesetting single equations and
- ***align*** environment for multiple equations and automatic alignment.

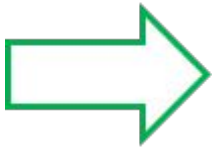
The ***align*** environment aligns the equations at the ***sign &***. Single equations have to be *separated* by a *linebreak* \\. There is no alignment when using the simple *equation* environment. It is not possible to enter two equations in that environment, it will result in a *compilation error*. The asterisk (e.g. equation*) indicates, that we don't want the equations to be numbered.

My first LaTex file - fractions

```
\documentclass{article}

  \usepackage{amsmath}

  \begin{document}

    \begin{align*}
      f(x) &= x^2\\
      g(x) &= \frac{1}{x}\\
      F(x) &= \int^a_b
\frac{1}{3}x^3\\
      G(x) &= \frac{1}{\sqrt{x}}
    \end{align*}

  \end{document}
```

LaTeX can display any mathematical notation. It's possible to typeset integrals, fractions and more. Every command has a specific syntax to use.

It's important to take care of opening and closing the **braces *{}*.**

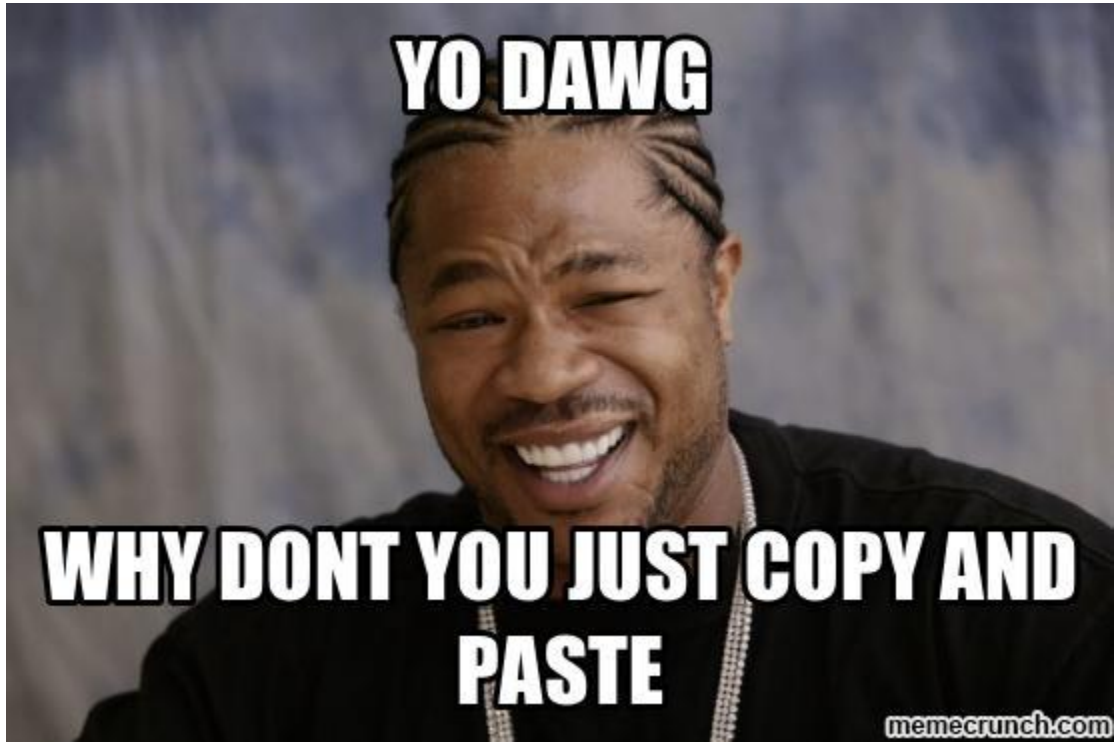


$$f(x) = x^2$$
$$g(x) = \frac{1}{x}$$
$$F(x) = \int_b^a \frac{1}{3}x^3$$
$$G(x) = \frac{1}{\sqrt{x}}$$

..or easier alternatives

https://latex.codecogs.com/eqneditor/editor.php

https://latexeditor.lagrida.com/



My first LaTex file - matrices

```
\documentclass{article}

\usepackage{amsmath}

\begin{document}
  $
  \begin{matrix}
  1 & 0\\
  0 & 1
  \end{matrix}
  $
\end{document}
```

There is a special matrix environment for displaying matrices in LaTeX.

To surround the matrix by brackets, it's necessary to use special statements, because the plain [ ] symbols do not scale as the matrix grows.
To scale them up, we must use the following code:

```
\left[
\begin{matrix}
1 & 0\\
0 & 1
\end{matrix}
\right]
```

This also works for parentheses and braces and is not limited to matrices. It can be used to scale for fractions and other expressions.

My first LaTex file - images

```
\documentclass{article}
\usepackage{graphicx}
\begin{document}
  \begin{figure}
\includegraphics[width=\linewidth]{boat.jpg}
    \caption{A boat.}
    \label{fig:boat1}
  \end{figure}
  Figure \ref{fig:boat1} shows a boat.
\end{document}
```

LaTeX automatically indexes all pictures and tags them with successive numbers when using the ***figure environment*** and the ***graphicx package***.

The *figure* environment takes care of the numbering and positioning of the image within the document. To include a figure, use the `\includegraphics` command. It takes the image width as an option in brackets and the path to your image file.
`\linewidth` - the picture will be scaled to fit the width of the document.
`\caption` – sets the text shown below the image
`\label` - is invisible, but useful for refering to the figure in the document. To refer to the figure, use `\ref{label}` in the text. It will be replaced by the correct number.

You need to include the ***graphicx*** package in order to use this code.

My first LaTex file - images and positioning!

```
\documentclass{article}

\usepackage{graphicx}

\begin{document}

  \begin{figure}[h!]
    \includegraphics[width=\linewidth]{boat.jpg}
    \caption{A boat.}
    \label{fig:boat1}
  \end{figure}

  Figure \ref{fig:boat1} shows a boat.

\end{document}
```

The figure doesn't necessarily show up in the place where you put your code in the .tex file. LaTeX will put the picture on the page where it finds sufficient space.
To prevent this behavior, it's necessary to set the ***float*** value for the figure environment.

```
%...
\begin{figure}[h!]
%...
```

Setting the float by adding *[h!]* behind the figure environment
\begin tag will force the figure to be shown at the location in the document. Possible values are:

- h (here) - same location
- t (top) - top of page
- b (bottom) - bottom of page
- p (page) - on an extra page
- ! (override) - will force the specified location

My first LaTex file - tables

```
\documentclass{article}

\begin{document}

  \begin{table}[h!]
    \centering
    \caption{Caption for the table.}
    \label{tab:table1}
    \begin{tabular}{l|c||r}
      1 & 2 & 3\\
      \hline
      a & b & c\\
    \end{tabular}
  \end{table}

\end{document}
```

LaTeX offers an environment for table creation. For this purpose we use the ***table***, ***tabular*** and the ***center*** *environment*.

The ***table*** *environment* holds our other environments and allows to add a caption to our table.
The data is contained in the ***tabular*** *environment.*
Use the ***center*** *environment* to center the table on the page.

The *ampersands* & are ***column separators*** and *newline symbols* \\ are ***row separators***.
Vertical lines, passed as an argument to the tabular environment (e.g. `\begin{tabular}{l|c||r}`) and the letters tell whether we want to align the content to the **left (l)**, to the **center (c)** or to the **right (r)** for each column.

Row separators can be added with the `\hline` command.
`\caption` and `\label` commands can be used in the same way as for pictures.

..or an easier alternative

Make the table in Excel/GSheets, copy and paste in this website:

https://www.tablesgenerator.com/

My first LaTex file - table of contents

```
\documentclass{article}

\begin{document}

  \tableofcontents
  \newpage

  \section{Section}

  Dummy text

  \subsection{Subsection}

  Dummy text

\end{document}
```

Generating a ***table of contents*** can be done with a few commands. LaTeX uses the section headings to create the *table of contents.*

```
\tableofcontents
```

You can also create a ***list of figures*** and a ***list of tables***.

```
\listoffigures
\listoftables
```

Practice Task - make something fun but elegant in LaTeX

Step 1: Create a new document project in overleaf.
Step 2: The .tex file should be named something like:

<Surname_GivenName_Memes>.tex e. g. Godzilla_Gospod.tex

Step 3: Combine images, math equations and other LaTeX elements (see samples in next slides) discussed today. Add your name somewhere in the document so it is visible.

<optional>
Step 4: Submit the .tex code, other files and screenshot of the preview as a contribution to CP1Preps repository. You can do this by creating your own branch, and submitting a pull request with your updated code.

Homework

$\sum_{n=1}^{\infty} n = -\frac{1}{12}$

Advertising

Reposts

$\pi = e = 3$

"Let"

Nonsensical proofs

Very elementary math

Disrespecting other academic disciplines

High quality OC related to mathematics

$$69 = \sum_{n=1}^{9} \sum_{d|n} d$$

Theoretical computer science:

27.5 Proposition. $\vdash_{\mathbf{K}+(\mathrm{A3})} \boxdot(A \leftrightarrow B) \rightarrow \boxdot(F(A) \leftrightarrow F(B))$.

27.16 Lemma. $w \models \boxdot(p \leftrightarrow A) \rightarrow \boxdot(\Box C_i(p) \rightarrow \Box C_i(H_i))$.

Also theoretical computer science:

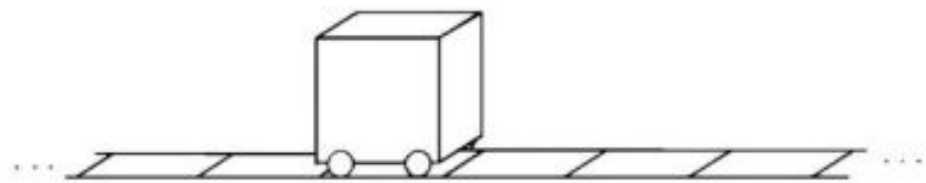
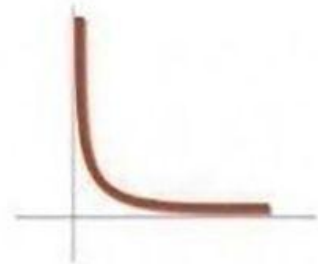


Figure 3-1. A Turing machine.

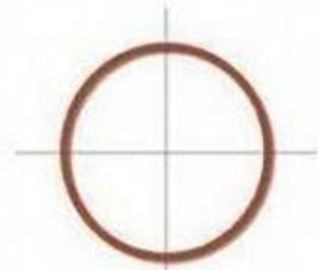
Appropriate for all ages

ALL YOU NEED IS

$$y = \frac{1}{x}$$



$$x^2 + y^2 = 9$$



$$y = |-2x|$$

$$x = -3|\sin y|$$

Learning LaTeX so your reports are more nicely typeset

Learning LaTeX because your tutor marks down Word-users

Learning LaTeX for that smug sense of elitist superiority

Learning LaTeX so you can understand memes