Introduction to LaTeX

Tejas Sanap

December 28, 2019

How do we write a document?

The process of creating a document comprises of two components:

How do we write a document?

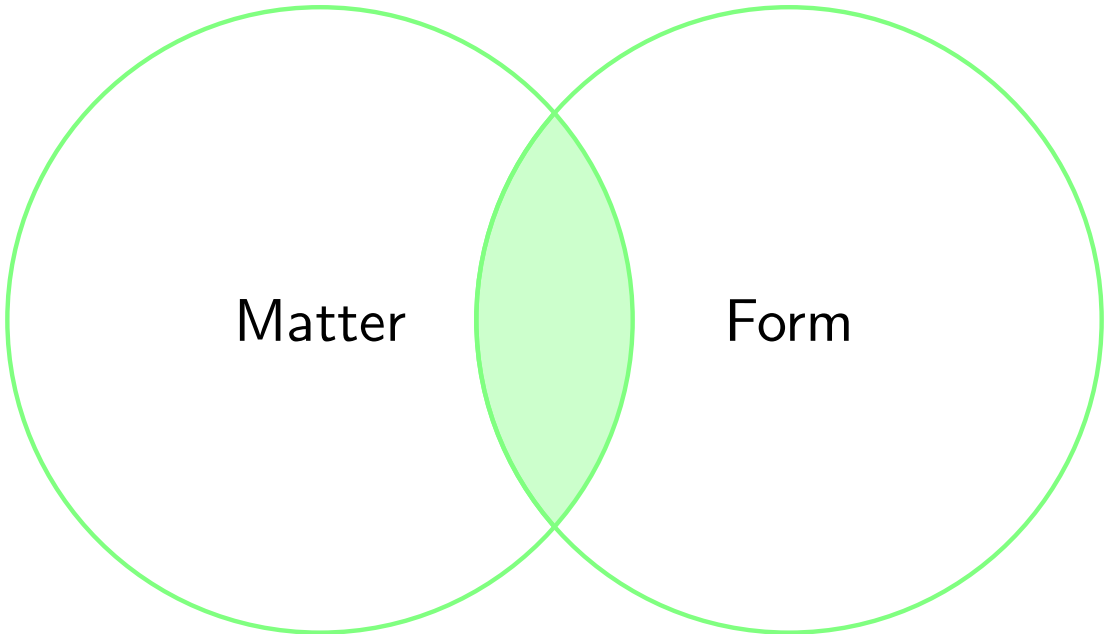
The process of creating a document comprises of two components:

- Deciding **what** the content **should be**. . .
- Deciding **how** the content **should look**. . .

How do we write a document?

The process of creating a document comprises of two components:

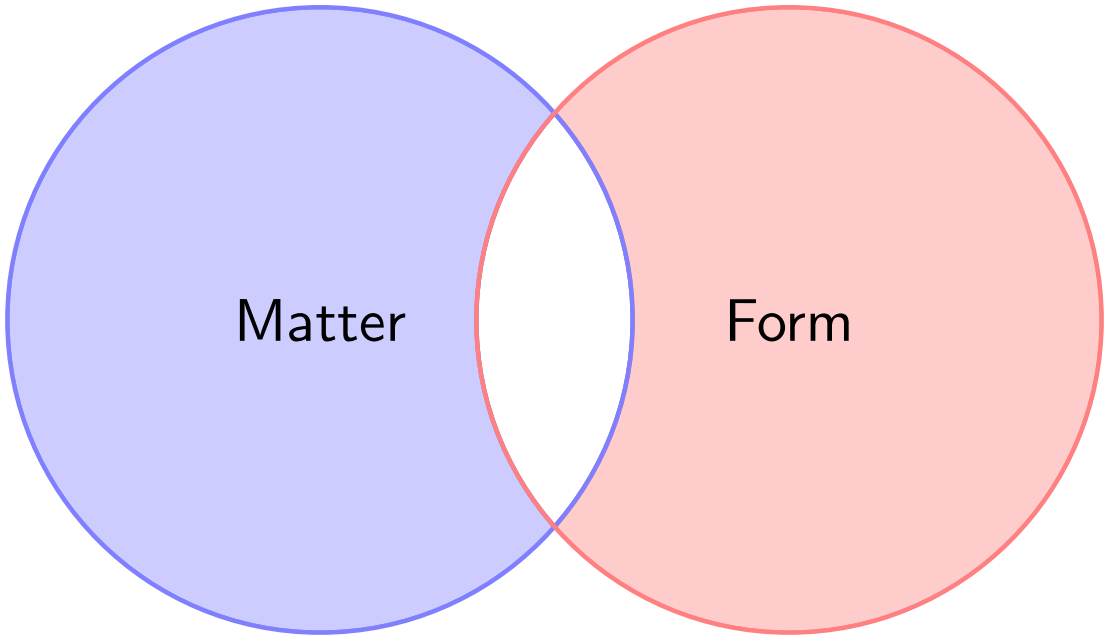
- Deciding **what** the content **should be**...
- Deciding **how** the content **should look**...



How do we write a document?

The process of creating a document comprises of two components:

- Deciding **what** the content **should be**...
- Deciding **how** the content **should look**...



Type-setting?

What is type-setting?

Type-setting?

What is type-setting?

- The process of arranging the various objects on a page.
- It is process that takes place after the manuscript has been written.

Type-setting?

What is type-setting?

- The process of arranging the various objects on a page.
- It is process that takes place after the manuscript has been written.

How do we typeset?

MS WORD!

Type-setting?

What is type-setting?

- The process of arranging the various objects on a page.
- It is process that takes place after the manuscript has been written.

How should we typeset?

LaTeX!

Type-setting?

What is type-setting?

- The process of arranging the various objects on a page.
- It is process that takes place after the manuscript has been written.

Unlike WYSIWYG softwares like MS Word and Open Office, LaTeX seperates the process of typesetting from the process of inserting content.

So, what exactly is LaTeX?

- LaTeXis a general type-setting system.

So, what exactly is LaTeX?

- LaTeXis a general type-setting system.
- It allows markup to describe the structure of the document, so that the user does not have to think about it.

So, what exactly is LaTeX?

- LaTeXis a general type-setting system.
- It allows markup to describe the structure of the document, so that the user does not have to think about it.
- Thus, by using document classes and additional packages the same document can be generatedd in several different formats.

So, what exactly is LaTeX?

- LaTeXis a general type-setting system.
- It allows markup to describe the structure of the document, so that the user does not have to think about it.
- Thus, by using document classes and additional packages the same document can be generatedd in several different formats.
- LaTeXis technically a TeX macro package.

So, what exactly is LaTeX?

- LaTeXis a general type-setting system.
- It allows markup to describe the structure of the document, so that the user does not have to think about it.
- Thus, by using document classes and additional packages the same document can be generatedd in several different formats.
- LaTeXis technically a TeX macro package.
- In computer-science-y terms, TeX is a *macroprocessor*.

What makes LaTeX different from WYSIWYG text processors?

- WYSIWYG systems make two claims:

What makes LaTeX different from WYSIWYG text processors?

- WYSIWYG systems make two claims:
  1. You type what you want to print.

What makes LaTeX different from WYSIWYG text processors?

- WYSIWYG systems make two claims:
  1. You type what you want to print.
  2. You see on your screen the closest approximation of what will finally be printed.

What makes LaTeX different from WYSIWYG text processors?

- WYSIWYG systems make two claims:
  1. You type what you want to print.
  2. You see on your screen the closest approximation of what will finally be printed.
- However, LaTeX is not a WYSIWYG text processor.

What makes LaTeX different from WYSIWYG text processors?

- WYSIWYG systems make two claims:
  1. You type what you want to print.
  2. You see on your screen the closest approximation of what will finally be printed.
- However, LaTeX is not a WYSIWYG text processor.
- Instead, LaTeX defines a logical model of a document.

What makes LaTeX different from WYSIWYG text processors?

- WYSIWYG systems make two claims:
  1. You type what you want to print.
  2. You see on your screen the closest approximation of what will finally be printed.
- However, LaTeX is not a WYSIWYG text processor.
- Instead, LaTeX defines a logical model of a document.
- Where, each object is labelled for what it is, *not what it should look like*.

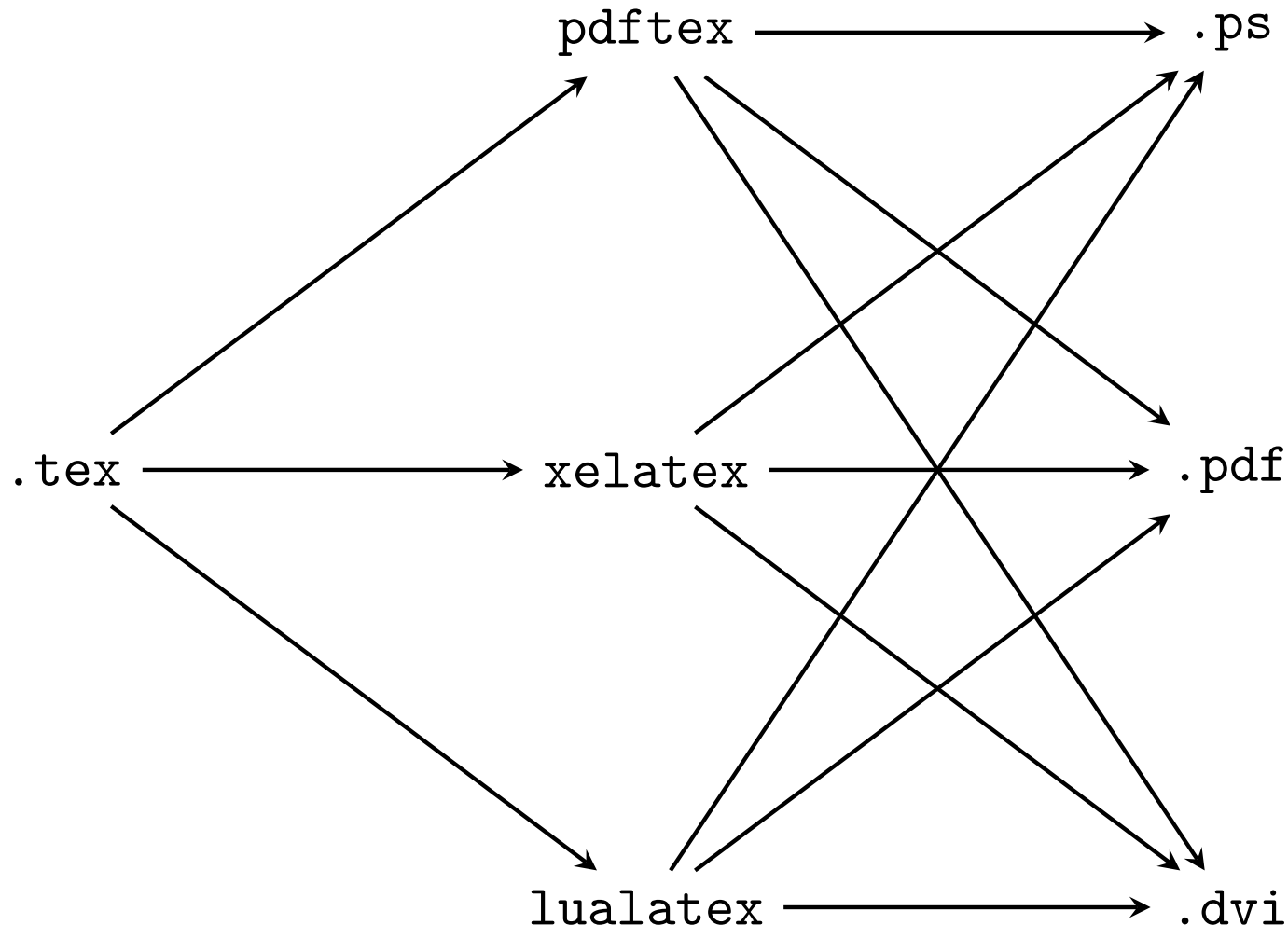
What makes LaTeX different from WYSIWYG text processors?

- WYSIWYG systems make two claims:
  1. You type what you want to print.
  2. You see on your screen the closest approximation of what will finally be printed.
- However, LaTeX is not a WYSIWYG text processor.
- Instead, LaTeX defines a logical model of a document.
- Where, each object is labelled for what it is, *not what it should look like*.
- This is the role that "markup" plays.

How do I use LaTeX?

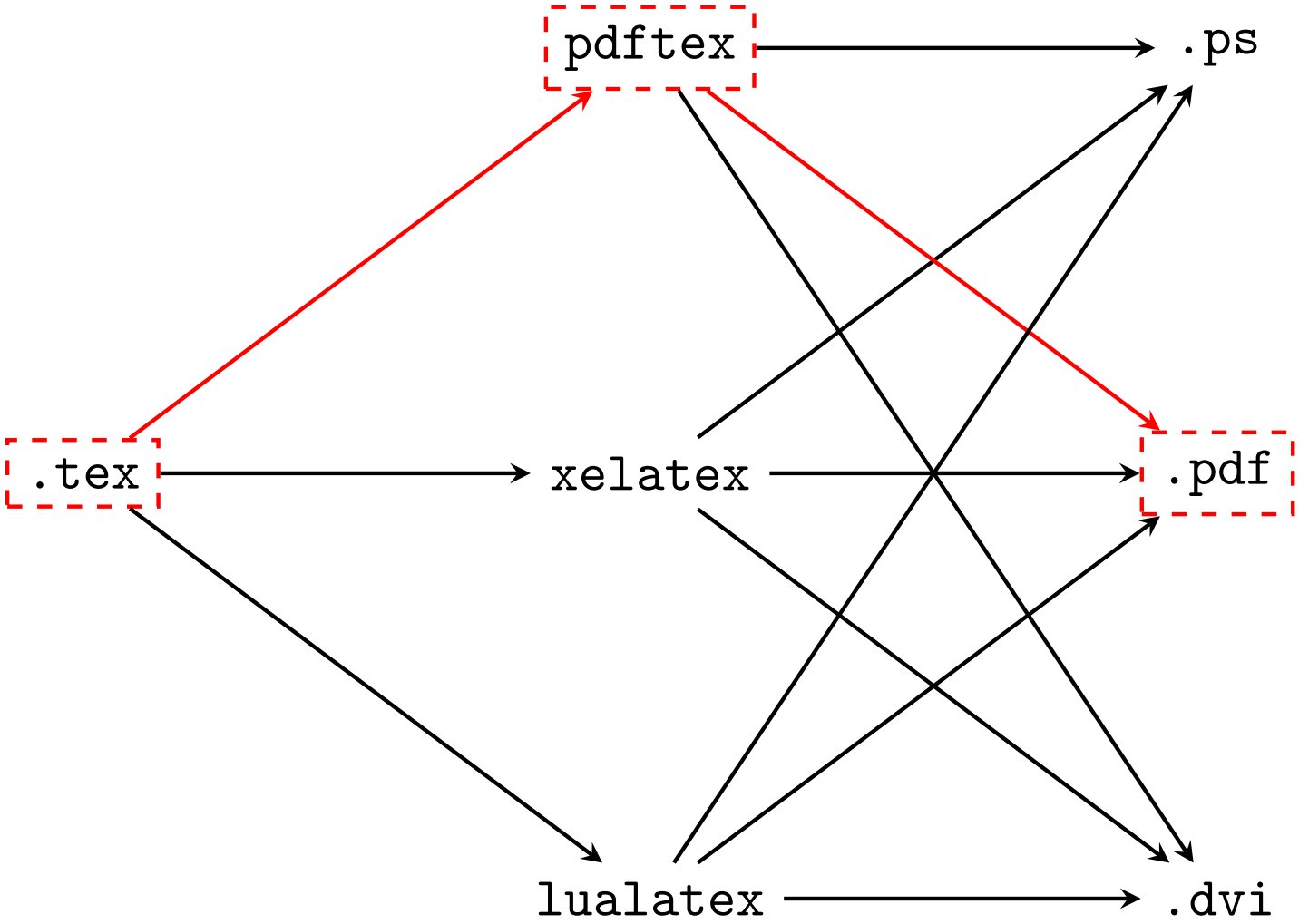
- A LaTeX file or source code is always a plaintext file, that is processed by a TeX engine.
- This is one of the most important features of LaTeX as it frees the user using a particular version or text editor.

How do I use LaTeX?



Source Code **LaTeX engine** **Final Output**

How do I use LaTeX?



Source Code **LaTeX engine** **Final Output**

How many TEX engines are available?

There are multiple TEXengines:

How many TEX engines are available?

There are multiple TEXengines:

- **Knuth's** TEX:
  This is original TEXengine which serves as the lowest layer of LATEX's software architecture.

How many TEX engines are available?

There are multiple TEXengines:

- **Knuth's** TEX:
  This is original TEXengine which serves as the lowest layer of LATEX's software architecture.
- `pdftex`:
  This engine adds a bunch of primitives related to the PDF and DVI extension.

How many TEX engines are available?

There are multiple TEXengines:

- **Knuth's** TEX:
  This is original TEXengine which serves as the lowest layer of LATEX's software architecture.
- `pdftex`:
  This engine adds a bunch of primitives related to the PDF and DVI extension.
- `xetex`:
  This engine provides better font support.

How many TEX engines are available?

There are multiple TEXengines:

- **Knuth's** TEX:
  This is original TEXengine which serves as the lowest layer of LATEX's software architecture.
- `pdftex`:
  This engine adds a bunch of primitives related to the PDF and DVI extension.
- `xetex`:
  This engine provides better font support.
- `luatex`:
  Originally, meant to replace `pdftex`, but, now moving in a very different direction. This engine also better font support (like, `xelatex`) through Lua code.

How many TEX engines are available?

There are multiple TEXengines:

- **Knuth's** TEX:
  This is original TEXengine which serves as the lowest layer of LATEX's software architecture.
- `pdftex`:
  This engine adds a bunch of primitives related to the PDF and DVI extension.
- `xetex`:
  This engine provides better font support.
- `luatex`:
  Originally, meant to replace `pdftex`, but, now moving in a very different direction. This engine also better font support (like, `xelatex`) through Lua code.
- and, a few more...

Structure of LaTeX source file.

```
1  \documentclass{article}
2
3  \usepackage{hyperref}
4
5  \title{A critical analysis of Naruto: the Manga}
6  \author{Tejas Sanap}
7
8  \begin{document}
9          \maketitle
10
11 \end{document}
```

Environments

- TEX give us the ability to directly use commands.

Environments

- TEX give us the ability to directly use commands.
- LATEX builds up on that and provides us with *environments*.

Environments

- TEX give us the ability to directly use commands.
- LATEX builds up on that and provides us with *environments*.
- Environments perform an action on a block rather than just doing something in one place.

Environments

- TEX give us the ability to directly use commands.
- LATEX builds up on that and provides us with *environments*.
- Environments perform an action on a block rather than just doing something in one place.
- Environments puts its contents inside a TEX group.

Environments

- TEX give us the ability to directly use commands.
- LATEX builds up on that and provides us with *environments*.
- Environments perform an action on a block rather than just doing something in one place.
- Environments puts its contents inside a TEX group.
- And, prevents commands from leaking out.

Environments

- TEX give us the ability to directly use commands.
- LATEX builds up on that and provides us with *environments*.
- Environments perform an action on a block rather than just doing something in one place.
- Environments puts its contents inside a TEX group.
- And, prevents commands from leaking out.
- Enviroments restrict their effects to their own contents.

Environments

Example:

```
\begin{center}
        The Jinchuriki.
\end{center}
```

Environments

Example:

```
\begin{center}
        The Jinchuriki.
\end{center}
```

A few of the most common environments we see are:

- `document`.
- `figure`.
- `align`.
- `table`.

What are floats?

- Some elements in a document cannot be broken over a page.

What are floats?

- Some elements in a document cannot be broken over a page.
- Floats are containers for such elements.

What are floats?

- Some elements in a document cannot be broken over a page.
- Floats are containers for such elements.
- Floats are always associated with a caption and label.

What are floats?

- Some elements in a document cannot be broken over a page.
- Floats are containers for such elements.
- Floats are always associated with a caption and label.
- By default, LaTeX identifies tables and figures as floats.

What are floats?

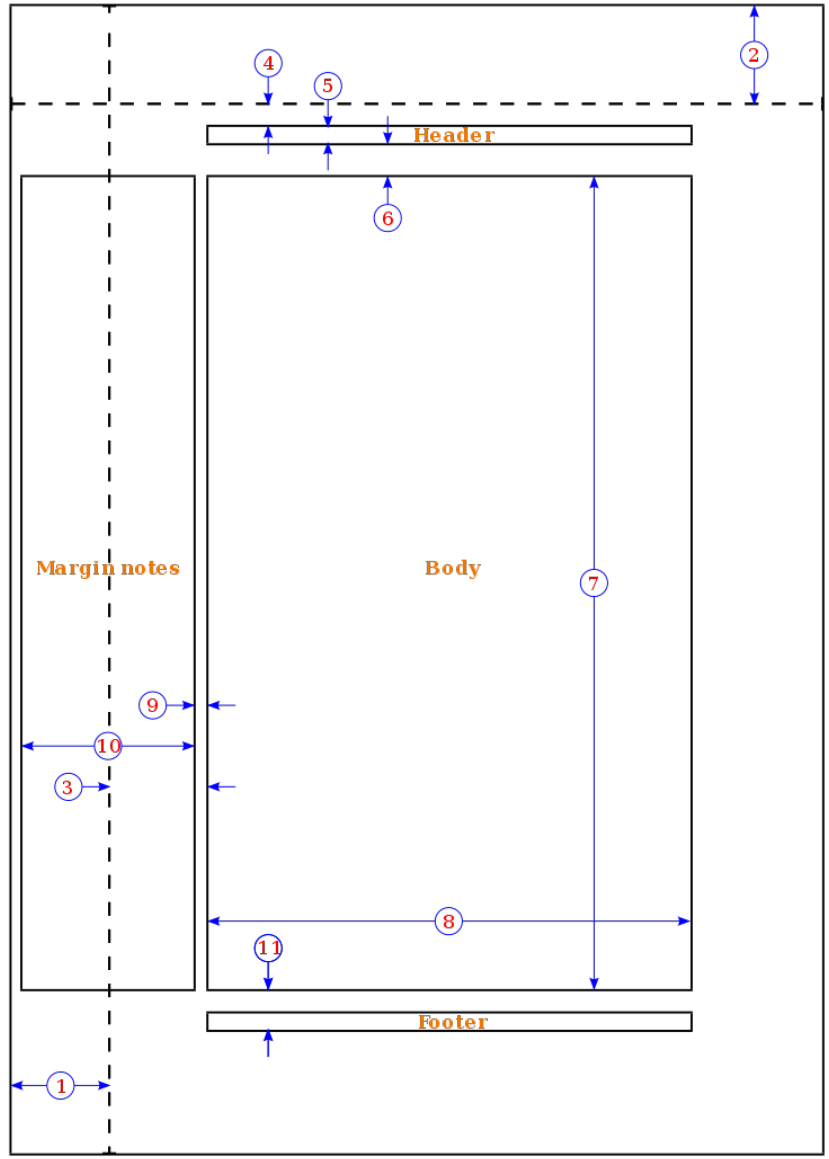
- Some elements in a document cannot be broken over a page.
- Floats are containers for such elements.
- Floats are always associated with a caption and label.
- By default, LaTeX identifies tables and figures as floats.
- The tricky part, is to figure out how to place/position floats.

What are floats?

- Some elements in a document cannot be broken over a page.
- Floats are containers for such elements.
- Floats are always associated with a caption and label.
- By default, LaTeX identifies tables and figures as floats.
- The tricky part, is to figure out how to place/position floats.
- LaTeX automatically moves floats across pages depending on how much space is left.

Page Geometry

4. \topmargin
7. \textheight
8. \textwidth
11. \footskip



Lengths in LaTeX

`\baselineskip`	Vertical distance between lines in a paragraph
`\columnsep`	Distance between columns
`\columnwidth`	The width of a column
`\evensidemargin`	Margin of even pages, commonly used in two-sided documents such as books
`\linewidth`	Width of the line in the current environment.
`\oddsidemargin`	Margin of odd pages, commonly used in two-sided documents such as books
`\paperwidth`	Width of the page
`\paperheight`	Height of the page
`\parindent`	Paragraph indentation
`\parskip`	Vertical space between paragraphs
`\tabcolsep`	Separation between columns in a table (tabular environment)
`\textheight`	Height of the text area in the page
`\textwidth`	Width of the text area in the page
`\topmargin`	Length of the top margin