

LATEX

Syllabus:-

1. Introduction to Latex:- & packages:-

- what is Latex
- Advantages of using Latex
- Explain about Latex software.
- Creating first Latex document.
- Latex basic type setting commands.
- font families
- font sizes
- font styles.
- Explanation of document class options.
- Introduction to packages.
- Advantages
- functionality of diff packages (Amsmath, amsfonts, tikz/pgf, geometry, graphics, color, multirow, chemfig).

Topic
⇒ What is Latex: -

①

- Latex (pronounced either ("lah-tech" or "lay-tech") is a macro package based on Tex created by Leslie Lamport. Its purpose is to simplify Tex type setting, especially for documents containing mathematical formulae. Within the type setting system, its name is formatted as latex.
- Tex is implemented by "donald knuth" in the year 1977. Later it can be edited to work for different codes for user the versions are implemented by "Latex2ε".
- Many later authors have contributed extensions, called packages or styles, to latex. Some of these are bundled with most Tex/Latex software distributions.

Topic
⇒ Advantages of Latex: -

- Document sources can be read with any text editor and understood, unlike the complex binary and XML formats used with WYSIWYG programs.
- you can concentrate purely on the structure and contents of the document, not get caught up with superficial layout issues.
- you don't need to manually adjust fonts, text sizes, line height, or text flow for readability, as latex takes care of them automatically.
- In latex the document structure is visible to the user, and can be easily copied to another document. In WYSIWYG applications it is often not obvious how a certain formatting was produced, and it might be impossible to copy it directly for use in another document.

①-①
→ The layout, font, tables and so on are consistent throughout the document.

→ mathematical formulae can be easily typeset.

→ Since the document source is plaintext, tables, figures, Equations, etc. can be generated programmatically with any language.

→ you are forced to structure your documents correctly.

⇒ Advantage of micro soft:-

→ micro soft word is "what you see is what you get" (WYSIWYG). This means that you can see how the final document will look as you are typing.

⇒ Disadvantage of latex:-

→ latex does not support "WYSIWYG" "what you see is what you get". you can't see the final document until you compile the code successfully.

⇒ Explain about latex software:-

→ (See the soft copy installation pdf).

- o -

⇒ Terms regarding Tex:-

a) Document preparation systems:-

Latex is a document preparation system based on Tex. So the system is the combination of the languages and the macros.

b) Tex distributions:- are collections of packages and programs (compilers, fonts, and macropackages) that enable you to typeset without having to manually fetch files and configure things.

c) Engines:-

An engine is an executable that can turn your source code into a printable output format.

→ The engine by itself only handles the syntax, it also needs to load fonts and macros to fully understand the source code and generate output properly.

→ The engine will determine what kind of source code it can read, and what format it can output (usually DVI (Device independent file format) or PDF (portable document format)).

→ All in all, distributions are an easy way to install what you need to use the engines and the systems you want. Distributions usually target specific operating systems.

⇒ Systems

Descriptions

② - ①

- Context - A TeX-based document preparation system (as LaTeX is) with a very consistent and easy syntax and support for pdfTeX, XeTeX and LuaTeX engines. It does not have the same objective as LaTeX however.
- LaTeX - A TeX-based document preparation system designed by Leslie Lamport. It is actually a set of macros for TeX. It aims at taking care of the formatting process.
- metafont - A high-quality font system designed by Donald Knuth along TeX.
- metapost - A descriptive vector graphics language based on metafont.
- TeX - The original language designed by Donald Knuth.

⇒ Engines

Descriptions

- luatex, lualatex - A TeX engine with Lua scripting engine embedded aiming at making TeX internals more flexible.
- pdftex, pdfLaTeX - the engines (PDF compilers)
- tex, latex - the engines (DVI compilers).

Topic ⇒ General structure of a file :-

(3)

- every latex file begins with `\documentclass{...}`
- Article, book, report etc. are example of different document class.
- This begins a section called the "preamble" which contains global instruction for the file.
- `\usepackage{...}` Commands and others are placed here.
- After the preamble, the body of the text begins with `\begin{document}`
- At the end of the file is `\end{document}`
- ⇒ `[...]` vs `{...}`
- `[...]` contains an optional argument.
- `{...}` contains a required argument.

Ex:- `\documentclass[a4paper, 12pt, oneside]{article}`

⇒ Preamble :-

- After `%`, latex ignores the rest of the line (comment).
- `\documentclass{article}` causes latex to format the text as an article with set formatting codes.
- `\usepackage{...}` Specifies an exception or addition to the set formatting rules multiple packages can be used.

⇒ Structure :-

(3) - (1)

```
\documentclass [option] {class}
\begin {document}
  \title { ... }
  \author { ... }
  \date { ... } (put . in front of it to suppress)
  \maketitle ( . title and name appear here )
\end {document}.
```

Ex:- \documentclass [a4paper, 12pt, oneside] {Article}

```
\begin {document}
\title {RGUKT}
\author {Ravi Reddy}
\date {28.01.19}
\maketitle
\end {document}.
```

another structure:-
Sectioning:-
1. Section
1.1. Subsection {
1.1.1. ^{sub} Subsection
output 1, 1.1, 1.1.1.

⇒ Structure:-

```
\documentclass [option] {class}
\begin {document}
  . . .
\end {document}
- . -
```

TOPIC ⇒ Creating a first latex document:-

④

→ latex is a great tool to create documents. It's based on the 'WYSIWYM' (what you see is what you mean) idea, meaning you only have to focus on the contents of your document and the computer will take care of the formatting.

→ with latex, it's very easy to create professional-looking material. This article presents the basics of how to create a document.

→ Introduction:-

let's start with the simplest working example.

```
\documentclass {article}
```

```
\begin {document}
```

first document . This is a simple example, with no extra parameters or packages included.

```
\end {document}
```

output.

first document.
This is a simple example
with no extra parameters
or packages includes

④-①
→ The input file is just a plain text file, with the extension .tex. It will contain code that the computer interprets to produce a pdf file.

→ The first line of code declares the type of document, in this case is an article.

→ Then, between the `\begin{document}` `\end{document}` tags you must write the text of your document.

⇒ The preamble of a document :-

→ In the previous example the text was entered after the `\begin{document}` command. The part of your .tex file before this point is called the preamble.

→ In the preamble, you define the type of document you are writing and the language, load extra packages you will need, and set several parameters. For instance, a normal document preamble would look like this:

```
\documentclass[12pt, letterpaper]{article}
\usepackage[utf8]{inputenc}
\title{First document}
\author{Knuth}
\thanks{Team LaTeX}
\date{Feb 2018}.
```

Description of each line:-

⑤

→ \documentclass [12pt, letterpaper] {article}

As said before, this defines the type of document. Some additional parameters inside brackets and comma-separated can be passed to the command.

→ In the example, the extra parameters set the font size (12pt) and the paper size (letterpaper). Of course other font sizes (9pt, 11pt, 12pt) can be used. The default size is 10pt.

→ As for the paper size, other possible values are included A4 and legal paper.

→ Note that overleaf uses a European LaTeX distribution, which produces documents in A4 size by default.

→ \usepackage [^{→ default} utf8] {^{→ encoding} inputenc}

This is the encoding for the document, to allow characters beyond ASCII (eg: a, u, ...) to be used in the text. It can be omitted or changed to another encoding but utf-8 is recommended.

→ unless you specifically need another encoding. Or if you are unsure about it, add this line to the preamble.

→ The next three lines are self-descriptive.

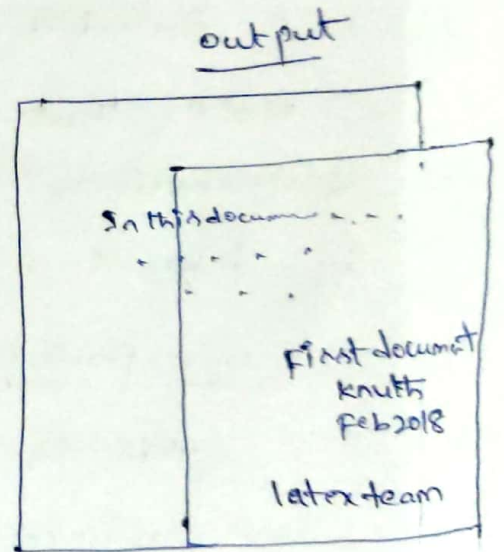
Anyway, you can see a description of what they actually do in the next section.

⑤-⑥
→ Another important parameter that can be passed to the `\documentclass` command is `twocolumn` if you want your text in a two-column format and two-side paper sheet printing.

→ Displaying the title of your document:-

To display the title of your document you have to declare its components in the preamble and then use some additional code:

```
\documentclass [12pt, letterpaper, twoside] {article}
\usepackage [utf8] {inputenc} no space encrypting
\title {first document}
\author {knuth}
\thanks {Latex team}
\begin {Feb 2018}
\begin {title page}
\maketitle
\end {title page}
```



In this document some extra packages and parameters were added. There is an encoding package and page size and font size parameters.

```
\end {document}
```

There is a block with three lines in the preamble^⑥ that defines the information to be included on the title page.

→ \title {first document}

This is the title.

→ \author {knuth}

Here you put the name(s) of the author(s) and, as an optional parameter, you can add the next command.

→ \thanks {latex team}

This can be added the name of the author, inside the braces of the title command. It will add a superscript and a footnote with the text inside the braces. useful if you need to thank an institution in your article.

→ \date {feb 2018}

you can enter the date manually or use the command \today so the date will be updated automatically at the time you compile your document.

→ Once you have that in the preamble now in the body of your document you can use the next commands for the information to be printed.

→ \begin { title page } \end { title page }

This declares an environment. a block of code with a specific behaviour depending on its type. In this case whatever you include in this title page environment will appear in the first page of your document.

→ \make title .

This Command will print the title. the author and the date in the format shown in the example. if it's not enclosed in a title page environment, it will be shown at the beginning of the document, above the first line.

→ Basic formatting: abstract, paragraphs and new lines :-

Everything included inside the \begin { document } \end { document } Commands will be rendered in the final document.

```
\documentclass [12pt, letterpaper, twoside] {article}
\usepackage [utf8] {inputenc}
\begin { document }
\begin { abstract }
```

this is a simple paragraph at the beginning of the document. A brief introduction to the main subject.

→ \end {abstract}

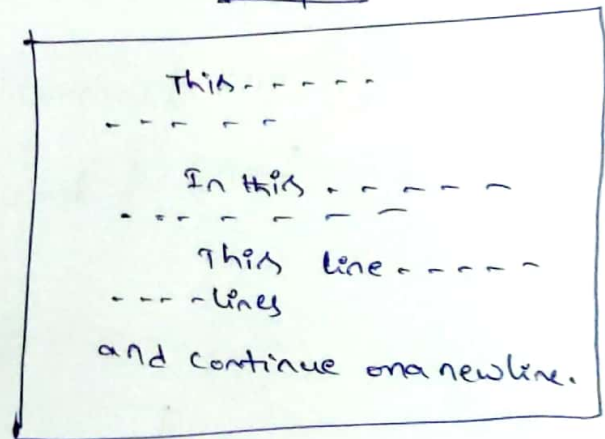
In this document some extra packages and parameters were added. There is an encoding package, and page size and font size parameters.

This line will start a Second paragraph.

And I can break the lines and continue on a new line.

→ \end {document}

output



This ----

In this ----

This line ----

and continue on a new line.

→ comments :-

Some times it's necessary to add Comments to your latex code for readability. This is straight-forward, put a % before the Comment and latex will ignore the text.

\documentclass {article}

\usepackage [utf8] {inputenc} % . codification of

\usepackage {Comment} % . Here begins the body of the document.

→ \begin {document}

⑦-①

This document Contains a lot of comments, none of them will appear here, only this text.

This document Contains a lot of comments, none of them will appear here, only this text.

→ \begin {comment}

This text won't show up in the compiled pdf
this is just a multi-line comment. useful to,
for instance, comment out slow-rendering parts
while working on a draft.

\end {comment}

\end {document}

output

This document contains ...
----- only this text.
This document ...
----- only this text.

\use package {comment}

The % symbol is a reserved character if you actually
use %. see the reference guide for a full list of

⇒ Latex basic type setting Commands:-

Document types available in the \documentclass command.

<u>Document type</u>	<u>Description.</u>
Article	→ for short documents and journal articles.
report	→ for longer documents and dissertations. is the most commonly used.
book	→ useful to write books.
letter	→ for letters.
slide	→ for slides, rarely used.
beamer	→ slides in the Beamer class format. See the beamer documentation for a better description.

⇒ Reserved characters:-

The following symbols can be characters reserved by latex because they introduce a command and have a special meaning.

\$ % ^ & - { } ~ \

These symbols can be printed with special commands (in some cases - inside mathematical environment).

<u>character</u>	<u>function</u>	<u>How to print it</u>
1. #	→ macroparameter	→ \#
2. \$	→ math mode	→ \$
3. %	→ Comment	→ \%
4. ^	→ Superscript (in math mode)	→ \wedge { } or \textasciisum
5. &	→ separate column entries in tables	→ \&
6. -	→ Subscript (in math mode)	→ _
7. { }	→ processing block	→ \{ \}
8. ~	→ unbreakable space, use it whenever you want to leave a space which is unbreakable	→ \~ { } (or) \textasciitilde
9. \	→ starting Commands, which extend until the first non-alphanumerical character	→ \textbackslash (or) \

Font Families :-

(3 types families are included in latex families)

By default, in Standard latex classes the

Script typeface (roman) font is used. The other font type faces (sans serif and typewriter, monospace) can be used by entering some specific commands.

Ex:-

```
\documentclass[a4paper]{article}
```

```
\begin{document}
```

```
FONT FAMILIES ||
```

→ latex → font styles

```
\textrm{font families} ||
```

```
\textsf{font families} ||
```

```
\texttt{font families} ||
```

```
\textmd{font families} ||
```

```
\end{document}
```

3 types are included.

output

FONT FAMILIES

font families.

font families

font families

font families

font families.

- o -

⇒ Font sizes :-

Font sizes are identified by special names, the actual size is not absolute but relative to the font size declared in the \documentclass statement.

Ex:-

```
\documentclass [options] {article}
```

```
\begin {document}
```

```
{\tiny font size} ||
```

```
{\scriptsize font size} ||
```

```
{\small font size} ||
```

```
{\large font size} ||
```

```
{\Large font size} ||
```

```
{\LARGE font size} ||
```

```
{\huge font size} ||
```

```
{\Huge font size} ||
```

```
\end {document}
```

output

```
font size
font size
font size
font size
font size
font size
font size
```

Font styles:-

(10)

The most common font styles in latex are bold, italics, and underlined, but there are a few more.

Ex:-

```
\documentclass [options] {article}
\begin{document}
\emph {font style} ||
\textit {font style} ||
\textsl {font style} ||
\textbf {font style} ||
\texttt {font style} ||
\textsc {font style} ||
\textsf {font style} ||
\underline {font style}
\end{document}
```

output

font style
font style *→ italic*
font style
font style **→ bold**
font style
FONT STYLE
font style
font style → underline

- 0 -