

Latex Logical vs Physical Formatting.

- We use logical formatting (**sections & titles**) instead of physical formatting (bold, italic)
- Physical formatting only used within logical
- Logical commands ensure consistent formatting and easy global changes.
- Physical formatting = only in preamble definitions
- Document body = only logical commands.

Basic Document Structure Examples

- Document class defines template & bare formatting
- Preamble contains document-wide settings and definitions
- Document environment contains actual content.
- Preamble = settings, Doc. Env. = content

```

● ● ●

\documentclass[a4paper,11pt]{article} % Class with options
\title{Example 2} % Document
\author{My name}
\date{May 5, 2021}
\begin{document} % Start content
\maketitle % Print title block
\section{What's this?} % Section heading
Content text here.
\end{document} % End content

```

LaTeX Command Syntax;

- Commands starts with backslash + letter (descriptive names)
- Arguments: {} = mandatory, [] = optional
- Command can have multiple arguments in sequence.

```

● ● ●

\command % No arguments
\command{argument} % Mandatory
\command[optional]{mandatory} % Both types
\documentclass[a4paper,11pt]{article} % Multiple options

```

- {} = Required, [] = optional

LaTeX Environment Syntax;

- Environment use \begin{name} ... \end{name} structure
- Create local scope — effects end with environment.
- Can have arguments like commands (optional and mandatory)

```

● ● ●

\begin{document} % Simple environment
Content here
\end{document}

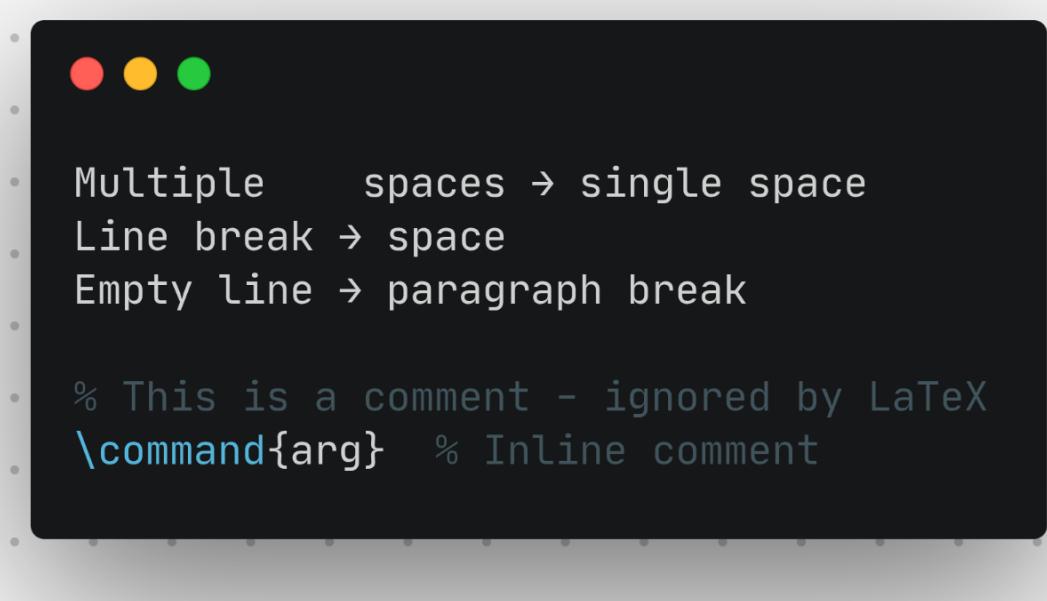
\begin{name}[optional]{mandatory} % Environment with
Content there
\end{name}

```

Environment = local scope with automatic cleanup.

How LaTeX Reads Input;

- LaTeX handles Unicode characters (ä, ü, ö, Greek, Russian) directly.
- Multiple spaces = single space, line break = space, empty lines = paragraph break.
- Special characters have reserved meanings in LaTeX syntax.
- \ = starts commands/macros
- {} and [] = command arguments
- \$ = math mode delimiters
- % = comment marker (ignores rest of line)



Special Symbols in LaTeX;

- Reserved character needs backslash command to print literally
- Common special symbols have simple \symbol commands

- Backslash itself is `\textbackslash`

commands

```
● ● ●
\#      % Hash/pound sign
\%      % Percent sign
\$      % Dollar sign
\&      % Ampersand
\_      % Underscore
\{      % Left brace
\}      % Right brace
\textbackslash % Backslash itself
```

- `\n` = line break (not backslash symbol.)

- Add `\` before special characters to print them.

Font Shape Commands ;

- Text formatting commands follow pattern `\text{\textbf{argument}}`

(`bf` = bold, `it` = italic, `sl` = slanted)

- Commands can be nested for combinations :

`\text{it}\{\text{\textbf{nested}}`

- `\textit{}` is intelligent - switches b/w italic & upright.

- `\textit{}` = context aware emphasis (italic \leftrightarrow upright)

```
● ● ●
\emph{emphasized}          % Italic (default)
\textbf{bold}                % Bold
\textit{italic}              % Italic
\textsl{slanted}             % Slanted
\textsc{Small Caps}          % Small caps
\textit{\textbf{both}}         % Nested
formatting
```

Font Families;

- Three main font families: serif (Roman), sans-serif, monospaced (typewriter)
- serif fonts have decorative strokes, sans-serif don't, monospaced have equal character.
- Different fonts serve different purpose (readability, screen display, code).
- serif = books/print, sans-serif = screen/heading, monospaced = code.



```
\textsf{sans-serif text}      % Sans-serif  
\texttt{typewriter text}    % Monospaced  
% default is serif/Roman family
```

Font Switching Declaration;

- Declaration works like switch (no arguments) vs command (with arguments)
- Use `\normalfont` to return to default font
- Switching declarations affect all following



```
% Declarations (switches)  
\sffamily\bfseries Bold sans-serif text  
\normalfont Back to default  
  
% vs Commands (arguments)  
\textsf{\textbf{Bold sans-serif text}}
```

Igrouping with Braces;

- curly braces {} create groups that confine command
- groups can be nested and must be properly closed.
- Alternative to using `\normalfont`
- {} local scope, every \ needs matching }



Normal text, {\sffamily sans serif {\bfseries and
bold}} text ends at closing brace

{\itshape italic text} back to normal

Font Sizes;

- 10 pre defined size declarations from `\tiny` to `\huge`.
- Size scale relative to document's base font size.
- No argument - based commands - must use braces for scope.
- 10 sizes: `\tiny` → `\huge`
- Size are relative to base font size.



`\tiny \scriptsize \footnotesize \small
\large \Huge \LARGE \huge \Huge`

{\large This is large} back to normal size

Creating Simple TeX Macros

- Macros are custom commands that expand to predefined text/commands
- Defined in preamble with
`\newcommand{\name}{content}`
- Act as abbreviations & placeholders for consistent formatting
- Define once in preamble, use everywhere in document.
- Change definition = change entire document