Learning LaTeX

Dr. G. H. J. Lanel

March 23, 2017

Outline

Outline

Introduction

- TEX is a low-level markup and programming language, created to typeset documents attractively and consistently.
- LATEX is a package based on TEX, its purpose is to simplify TeX typesetting.
- LATEXis a high-quality typesetting system.
- It includes features designed for the production of technical and scientific documentation. Today widely used for publications.

Introduction

- TEX is a low-level markup and programming language, created to typeset documents attractively and consistently.
- LATEX is a package based on TEX, its purpose is to simplify TeX typesetting.
- LATEXis a high-quality typesetting system.
- It includes features designed for the production of technical and scientific documentation. Today widely used for publications.

Introduction

- TEX is a low-level markup and programming language, created to typeset documents attractively and consistently.
- LATEX is a package based on TEX, its purpose is to simplify TeX typesetting.
- LATEXis a high-quality typesetting system.
- It includes features designed for the production of technical and scientific documentation. Today widely used for publications.

Introduction

- TEX is a low-level markup and programming language, created to typeset documents attractively and consistently.
- LATEX is a package based on TEX, its purpose is to simplify TeX typesetting.
- LATEXis a high-quality typesetting system.
- It includes features designed for the production of technical and scientific documentation. Today widely used for publications.

Introduction

- TEX is a low-level markup and programming language, created to typeset documents attractively and consistently.
- LATEX is a package based on TEX, its purpose is to simplify TeX typesetting.
- LATEXis a high-quality typesetting system.
- It includes features designed for the production of technical and scientific documentation. Today widely used for publications.

Introduction

- TEX is a low-level markup and programming language, created to typeset documents attractively and consistently.
- LATEX is a package based on TEX, its purpose is to simplify TeX typesetting.
- LATEXis a high-quality typesetting system.
- It includes features designed for the production of technical and scientific documentation. Today widely used for publications.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Introduction

- You can't see the final result straight away.
- Everything is controlled with written commands.
- The document is written in a self-chosen text editor and saved in a .tex-file and finally converts .tex-file into a .pdf/dvi-file.

Advantages:

- The layout, fonts, tables etc. are consistent throughout.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes and references are generated easily.

Software examples

Requirements:

1. Text Editor
2. LaTeX-binaries
3. Output viewer for DVI/PDF files

Windows:

1. Notepad, TeXnicCenter, Winshell
2. MiKTeX

Mac OS X:

1. TeXShop
2. MacTeX or TeX-live

Linux:

1. Emacsen, gvim, Texmaker, Kile
2. TeX-live

Software examples

Requirements:

1. Text Editor
2. LaTeX-binaries
3. Output viewer for DVI/PDF files

Windows:

1. Notepad, TeXnicCenter, Winshell
2. MiKTeX

Mac OS X:

1. TeXShop
2. MacTeX or TeX-live

Linux:

1. Emacsen, gvim, Texmaker, Kile
2. TeX-live

Software examples

Requirements:

1. Text Editor
2. LaTeX-binaries
3. Output viewer for DVI/PDF files

Windows:

1. Notepad, TeXnicCenter, Winshell
2. MiKTeX

Mac OS X:

1. TeXShop
2. MacTeX or TeX-live

Linux:

1. Emacsen, gvim, Texmaker, Kile
2. TeX-live

Software examples

Requirements:

1. Text Editor
2. LaTeX-binaries
3. Output viewer for DVI/PDF files

Windows:

1. Notepad, TeXnicCenter, Winshell
2. MiKTeX

Mac OS X:

1. TeXShop
2. MacTeX or TeX-live

Linux:

1. Emacsen, gvim, Texmaker, Kile
2. TeX-live

Software examples

Requirements:

1. Text Editor
2. LaTeX-binaries
3. Output viewer for DVI/PDF files

Windows:

1. Notepad, TeXnicCenter, Winshell
2. MiKTeX

Mac OS X:

1. TeXShop
2. MacTeX or TeX-live

Linux:

1. Emacsen, gvim, Texmaker, Kile
2. TeX-live

Outline

Document structure

The structure of a .tex-file:

```
\documentclass [ 12pt ] { article }
\begin{document }
. . .
% Document text
. . .
\end{document }
```

Page Layout

The document class command:

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

Example:

```
. . . [ a4paper , twoside , twocolumns ] {article}
```

or

```
\documentclass [ 1 1 pt , a4paper , oneside ] { report}
```

Some document classes: article, report, letter, beamer, book

Page Layout

The document class command:

documentclass [options] {class}

Example:

. . . [a4paper , twoside , twocolumns] {article}

or

documentclass [1 1 pt , a4paper , oneside] { report}

Some document classes: article, report, letter, beamer, book

Page Layout

The document class command:

documentclass [options] {class}

Example:

. . . [a4paper , twoside , twocolumns] {article}

or

documentclass [1 1 pt , a4paper , oneside] { report}

Some document classes: article, report, letter, beamer, book

Page Layout

The document class command:

\documentclass [options] {class}

Example:

. . . [a4paper , twoside , twocolumns] {article}

or

\documentclass [1 1 pt , a4paper , oneside] { report}

Some document classes: article, report, letter, beamer, book

Page Layout

The document class command:

\documentclass [options] {class}

Example:

. . . [a4paper , twoside , twocolumns] {article}

or

\documentclass [1 1 pt , a4paper , oneside] { report}

Some document classes: article, report, letter, beamer, book

Title

Title (inserted right after\begin{document}):

---

title {Introduction to LaTeX}

\ **author** {[Your name]}\ \

date {\today}

maketitle

Introduction to LaTeX
Your name
March 23, 2017

Title

Title (inserted right after\begin{document}):

---

title {Introduction to LaTeX}

\ **author** {[Your name]}\ \

date {\today}

maketitle

Introduction to LaTeX
Your name
March 23, 2017

Document structure

Sections and hierarchy

Sectioning commands:

\part { title}

\chapter { title }

\section { title}

\subsection { title}

\subsubsection { title}

\paragraph { title}

\subparagraph { title}

Document structure

Sections and hierarchy

Sectioning commands:

part { title}

chapter { title }

section { title}

subsection { title}

subsubsection { title}

paragraph { title}

subparagraph { title}

Document structure for the article

Sections and hierarchy

This article has nine sections and some subsections so, we need to use \ **section** { title} and \ **subsection** { title} commands.

For example if we look at the 7^{th} section of the article we can write it as follows:

\ **section** { Special Symbols}

\ **subsection** { Common Greek letters}

. . .

% Document text

. . .

\ **subsubsection** {Special symbols}

Document structure for the article

Sections and hierarchy

This article has nine sections and some subsections so, we need to use \ **section** { title} and \ **subsection** { title} commands.

For example if we look at the 7^{th} section of the article we can write it as follows:

```
\section { Special Symbols}
    \subsection { Common Greek letters}
    . . .
    % Document text
    . . .
    \subsubsection {Special symbols}
```

Document structure for the article

Sections and hierarchy

This article has nine sections and some subsections so, we need to use \ **section** { title} and \ **subsection** { title} commands.

For example if we look at the 7^{th} section of the article we can write it as follows:

\ **section** { Special Symbols}

subsection { Common Greek letters}

. . .

% Document text

. . .

subsubsection {Special symbols}

Document structure for the article

Sections and hierarchy

So, we can obtain the following output.

6 Special Symbols

6.1 Common Greek letters

These commands may be used only in math mode. Only the most common letters are included here.

$$\alpha, \beta, \gamma, \Gamma, \delta, \Delta, \epsilon, \zeta, \eta, \theta, \Theta, \kappa, \lambda, \Lambda, \mu, \nu, \xi, \Xi, \pi, \Pi, \rho, \sigma, \tau, \phi, \Phi, \chi, \psi, \Psi, \omega, \Omega$$

6.2 Special symbols

Formatting

paragraph spacing and indenting

Document text placed between

\begin {document} and \end {document}

- Line break : \ \ (2*backslashes)
- Paragraph separation : Two breakspaces (2*enter)
- Line of comment : Begins with %

Formatting

paragraph spacing and indenting

Document text placed between

\begin {document} and \end {document}

- Line break : \ \ (2*backslashes)
- Paragraph separation : Two breakspaces (2*enter)
- Line of comment : Begins with %

Formatting

paragraph spacing and indenting

Document text placed between

\begin {document} and \end {document}

- Line break : \ \ (2*backslashes)
- Paragraph separation : Two breakspaces (2*enter)
- Line of comment : Begins with %

Formatting

paragraph spacing and indenting

Document text placed between

\begin {document} and \end {document}

- Line break : \ \ (2*backslashes)
- Paragraph separation : Two breakspaces (2*enter)
- Line of comment : Begins with %

Formatting

paragraph spacing and indenting

Document text placed between

\begin {document} and \end {document}

- Line break : \ \ (2*backslashes)
- Paragraph separation : Two breakspaces (2*enter)
- Line of comment : Begins with %

Formatting

paragraph spacing and indenting

Document text placed between

\begin {document} and \end {document}

- Line break : \ \ (2*backslashes)
- Paragraph separation : Two breakspaces (2*enter)
- Line of comment : Begins with %

We can skip some space by using commands such as

```
\bigskip    \medskip    \smallskip    \vspace{1pc}
```

The space can be negative.

Document structure

Mathematics

- $ math $: Math in text paragraph

```
\section{Equations}

Let us see how easy it is to write equations

$\Delta =\sum_{i=1}^N w_i (x_i - \bar{x})^2 $ .

It is a good idea to number equations, but
we can have an equation without a number by writing.
```

Then the output will be as follows:

2 Equations

Let us see how easy it is to write equations, $\Delta = \sum_{i=1}^{N} w_i(x_i - \bar{x})^2$. It is a good idea to number equations, but we can have a equation without a number by writing.

Document structure

Mathematics

- \ begin{equation} math \end{equation}:
  Math centered, separated from text and labeled

```
\section{Equations}

Let us see how easy it is to write equations.

\begin{equation}
\Delta =\sum_{i=1}^N w_i (x_i - \bar{x})^2 .
\end{equation}
```

Then the output will be as follows:

2 Equations

Let us see how easy it is to write equations.

$$\Delta = \sum_{i=1}^{N} w_i (x_i - \bar{x})^2. \qquad (1)$$

- We can also have an equation without a number by writing using the following command.

```
\begin{equation}
P(x) = \frac{x - a}{b - a} , \nonumber
\end{equation}
```

Then the output will be as follows:

$$P(x) = \frac{x - a}{b - a}$$

- We can give an equation a label so that we can refer to it later, using the command \label{...} when writing the equation and then ~ \eqref{...} command in the paragraph to refer the equation.

```
\begin{equation}
\label{eq:ising}
E = -J \sum_{i=1}^N s_i s_{i+1} ,
\end{equation}

Equation~\eqref{eq:ising} expresses
the energy of a configuration
of spins in the Ising model.
```

Then the output will be as follows:

$$E = -J \sum_{i=1}^{N} s_i s_{i+1},$$

Equation (2) expresses the energy of a configuration of spins in the Ising model.

- We also can align several equations using the following commands:

```
\begin{align }
a & = b \\
c & = d,
\end{align }
```

Then we get the following output

$$a = b$$

$$c = d,$$

Examples of more complicated equations:

- $$I = \int_{-\infty}^{\infty} f(x)\,dx.$$

  In order to obtain the above equation we have to use the following commands.

```
\begin{equation*}
I = \! \int_{-\infty}^\infty f(x)\,dx
\label{eq:fine}.
\end{equation*}
```

Document structure

Mathematics – continued

$$m(T) = \begin{cases} 0 & T > T_c \\ \left(1 - [\sinh 2\beta J]^{-4}\right)^{1/8} & T < T_c \end{cases}$$

In order to obtain the above equation we have to use the following commands.

```
\begin{equation*}
\label{eq:mdiv}
m(T) =
\begin{cases}
0 & \text{$T > T_c$} \\
\bigl(1 - [\sinh 2 \beta J]^{-4} \bigr)^{\! 1/8}
& \text{$T < T_c$}
\end{cases}
\end{equation*}
```

Document structure

Mathematics – continued

$$\mathbf{T} = \begin{pmatrix} T_{++} & T_{+-} \\ T_{-+} & T_{--} \end{pmatrix},$$

In order to obtain the above matrix we have to use the following commands.

```
\begin{align*}
\textbf{T} &=
\begin{pmatrix}
T_{++} \hfill & T_{+-} \\
T_{-+} & T_{--} \hfill
\end{pmatrix} , \nonumber \\
\end{align*}
```

Document structure

Floats

- Free, floating objects; position is only partly defined by the user (here, bottom, top, p: separate page).
- Automatically numbered, label used for referencing.
- Captions can be added.

Float syntax:

\ **begin** { figure or table } [placement specifier]

. . . figure contents . . .

\ **end** { figure or table }

Document structure

Floats

- Free, floating objects; position is only partly defined by the user (here, bottom, top, p: separate page).
- Automatically numbered, label used for referencing.
- Captions can be added.

Float syntax:

\ **begin** { figure or table } [placement specifier]

. . . figure contents . . .

\ **end** { figure or table }

Document structure

Floats

- Free, floating objects; position is only partly defined by the user (here, bottom, top, p: separate page).
- Automatically numbered, label used for referencing.
- Captions can be added.

Float syntax:

\ **begin** { figure or table } [placement specifier]

. . . figure contents . . .

\ **end** { figure or table }

Document structure

Floats

- Free, floating objects; position is only partly defined by the user (here, bottom, top, p: separate page).
- Automatically numbered, label used for referencing.
- Captions can be added.

Float syntax:

\ **begin** { figure or table } [placement specifier]

. . . figure contents . . .

\ **end** { figure or table }

Document structure

Floats

- Free, floating objects; position is only partly defined by the user (here, bottom, top, p: separate page).
- Automatically numbered, label used for referencing.
- Captions can be added.

Float syntax:

\ **begin** { figure or table } [placement specifier]

. . . figure contents . . .

\ **end** { figure or table }

Document structure

Floats

- Free, floating objects; position is only partly defined by the user (here, bottom, top, p: separate page).
- Automatically numbered, label used for referencing.
- Captions can be added.

Float syntax:

```
\ begin { figure or table } [ placement specifier ]
. . . figure contents . . .
\ end { figure or table }
```

Document structure

Floats

- Free, floating objects; position is only partly defined by the user (here, bottom, top, p: separate page).
- Automatically numbered, label used for referencing.
- Captions can be added.

Float syntax:

\ **begin** { figure or table } [placement specifier]

. . . figure contents . . .

\ **end** { figure or table }

Table code example:

```
\begin{table}[h]
\begin{center}
\begin{tabular}{|l|l|r|l|}
\hline
lattice & $d$ & $q$ & $T_{\rm mf}/T_c$ \\
\hline
square & 2 & 4 & 1.763 \\
\hline
triangular & 2 & 6 & 1.648 \\
\hline
diamond & 3 & 4 & 1.479 \\
\hline
simple cubic & 3 & 6 & 1.330 \\
\hline
bcc & 3 & 8 & 1.260 \\
\hline
fcc & 3 & 12 & 1.225 \\
\hline
\end{tabular}
\end{center}
\end{table}
```

The result :

lattice	d	q	T_{mf}/T_c
square	2	4	1.763
triangular	2	6	1.648
diamond	3	4	1.479
simple cubic	3	6	1.330
bcc	3	8	1.260
fcc	3	12	1.225

Listing using bulleting

Bulleting code example:

```
\begin{itemize}
  \item Tom
  \item Dick
\end{itemize}
```

Listing using bulleting

The output is:

- Tom
- Dick

Listing using numbering

Numbering code example:

```
\begin{enumerate}
  \item bread
  \item cheese
\end{enumerate}
```

Listing using numbering

The output is:

1. bread
2. cheese

Supported image file types: JPG, PNG and PDF

Figure code example:

```
\begin{figure}[h]
\begin{center}
\includegraphics{fig1.jpg}
\caption{\label{fig:typical}Show me a sine.}
\end{center}
\end{figure}
```

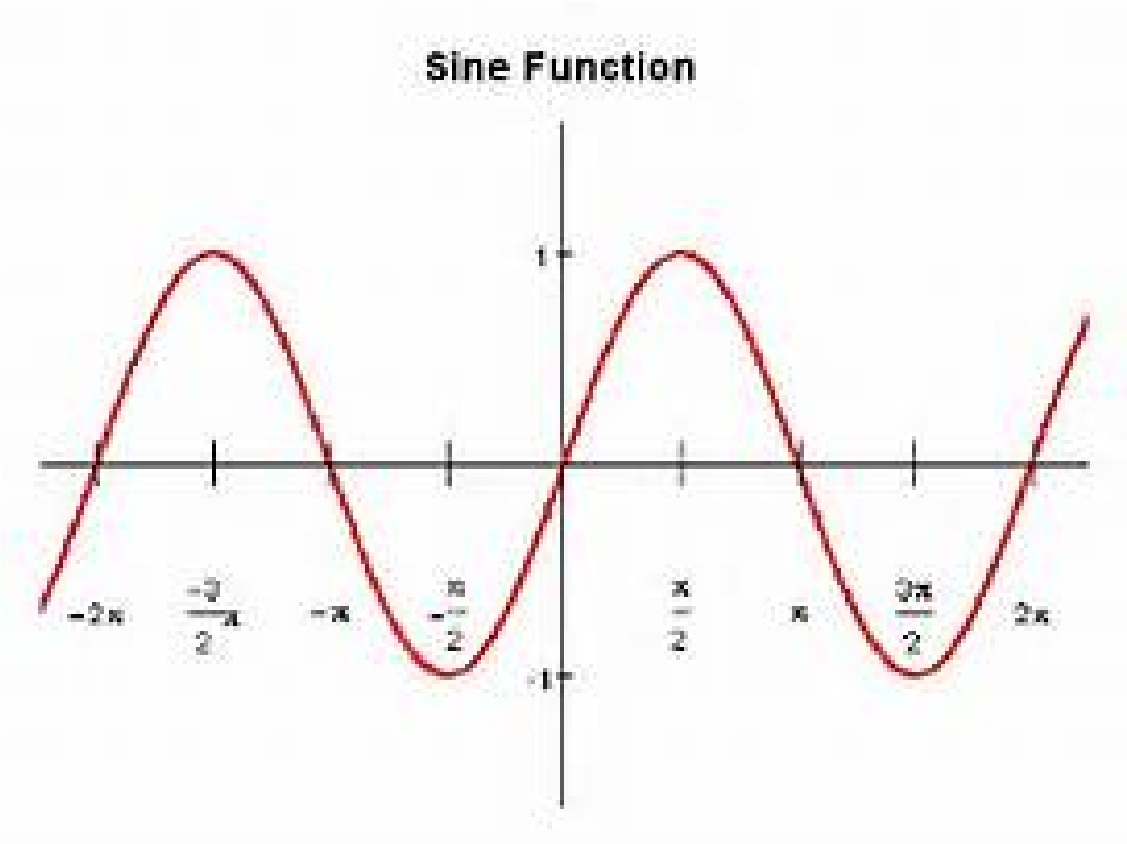


Figure : Sine curve.

Outline

Special Symbols

Common Greek Letters

These commands may be used only in math mode.

Only the most common letters are included here.

```
$\alpha,
\beta, \gamma, \Gamma,
\delta,\Delta,
\epsilon, \zeta, \eta, \theta, \Theta, \kappa,
\lambda, \Lambda, \mu, \nu,
\xi, \Xi,
\pi, \Pi,
\rho,
\sigma,
\tau,
\phi, \Phi,
\chi,
\psi, \Psi,
\omega, \Omega$
```

$$\alpha, \beta, \gamma, \Gamma, \delta, \Delta$$

$$\epsilon, \zeta, \eta, \theta, \Theta, \kappa$$

$$\lambda, \Lambda, \mu, \nu, \xi, \Xi$$

$$\pi, \Pi, \rho, \sigma, \tau, \phi$$

$$\Phi, \chi, \psi, \Psi, \omega, \Omega$$

The derivative and limit are defined as;

```
\begin{equation}
\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}
{\Delta x}
\end{equation}
\begin{equation}
f(x) \to y \quad \mbox{as} \quad x \to
x_{0}
\end{equation}
```

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} \tag{1}$$

$$f(x) \to y \quad \text{as} \quad x \to x_0 \tag{2}$$

$$f(x) \underset{x \to x_0}{\longrightarrow} y \tag{3}$$

Order of magnitude;

```
\log_{10}f \sim n
\end{equation}
\begin{equation}
f(x)\sim 10^{n}
\end{equation}
\end{frame}
```

$$\log_{10} f \sim n \tag{4}$$

$$f(x) \sim 10^{n} \tag{5}$$

Approximate equality;

```
\begin{equation}
f(x)\simeq g(x)
\end{equation}
```

$$f(x) \simeq g(x) \tag{6}$$

LaTeX is simple if we keep everything in proportion:

```
\begin{equation}
f(x) \propto x^3 .
\end{equation}
```

$$f(x) \propto x^3. \tag{7}$$

Outline

Formatting

Text styles

\emph{...} *I want to emphasize*

\textbf{...} **Bold**

\texttt{...} `Fixed width teletypefont`

\textsc{...} SMALL CAPITALS

Formatting

Text size

\tiny{...}	sample text
\Scriptsize{...}	sample text
\footnotesize{...}	sample text
\small{...}	sample text
\normalsize{...}	sample text
\large{...}	sample text
\Large{...}	sample text
\LARGE{...}	sample text
\huge{...}	sample text
\Huge{...}	sample text

Formatting

colours

There are several elements in LaTeX whose colour can be changed to improve the appearance of the document.

Colours can be manually defined to a desired tone using several models.

The simplest manner to use colours in your LaTeX document is by importing the package color or xcolor.

Formatting

colours

There are several elements in LaTeX whose colour can be changed to improve the appearance of the document.

Colours can be manually defined to a desired tone using several models.

The simplest manner to use colours in your LaTeX document is by importing the package color or xcolor.

Formatting

colours

There are several elements in LaTeX whose colour can be changed to improve the appearance of the document.

Colours can be manually defined to a desired tone using several models.

The simplest manner to use colours in your LaTeX document is by importing the package color or xcolor.

There are several elements in LaTeX whose colour can be changed to improve the appearance of the document.

Colours can be manually defined to a desired tone using several models.

The simplest manner to use colours in your LaTeX document is by importing the package color or xcolor.

There are several elements in LaTeX whose colour can be changed to improve the appearance of the document.

Colours can be manually defined to a desired tone using several models.

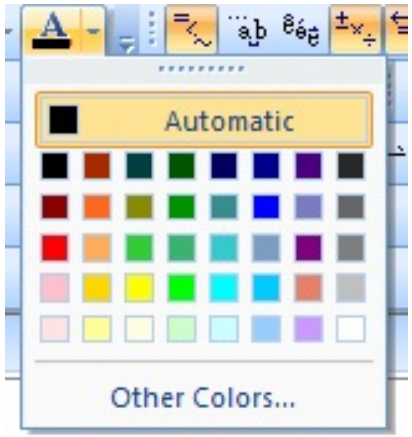
The simplest manner to use colours in your LaTeX document is by importing the package color or xcolor.

Some waysof using colours in LaTeX ;

1. Using the command

```
\begin{itemize}
\color{blue}
\item First item
\item Second item
\end{itemize}
```

2. Using toolbar

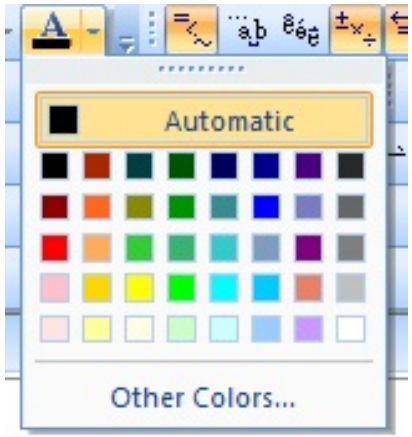


Some waysof using colours in LaTeX ;

1. Using the command

```
\begin{itemize}
\color{blue}
\item First item
\item Second item
\end{itemize}
```

2. Using toolbar



Outline

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Conclusions

- Learning LATEX is done by doing.
- Use external resources for command and syntax help.
- Compile the PDF frequently.
- Helps error debugging in the code.
- Let the floats (tables, figures) position themselves.
- Create a dedicated directory for each project.

Thank You!