A very short presentation about \LaTeX...

Merciadri Luca

University of Liège

November 17, 2008
Outline. I

1 Introduction
- What is \LaTeX? 
- What \LaTeX{} is not...
- Why using \LaTeX? 
- Quality of output 
- Portability 
- Focusing on your document 
- Writing quicker

2 Structure of a document
- A classical article 
  - Code 
  - Explanation 
  - Complements 
- A classical presentation 
  - Structure 
  - Example
A very short presentation about LaTeX...

Merciadri Luca

Introduction
What is LaTeX?
What LaTeX is not...
Why using LaTeX?
Quality of output
Portability
Focusing on your document
Writing quicker

Structure of a document
A classical article
Code
Explanation
Complements
A classical presentation
Structure
Example
Inserting images

Outline. II

1. Inserting images
   - Introduction
   - Example

3. Conclusion

4. Questions

5. Bibliography
What is \LaTeX? 

\LaTeX is...
What is \LaTeX? 

\LaTeX is... 

- a *free* and different word processor (not like Microsoft Word);
What is \LaTeX?

\LaTeX{} is...

- a *free* and different word processor (not like Microsoft Word);
- coming from \TeX{} (created by Donald E. Knuth);
What is \LaTeX?  

\LaTeX\ is...  

- a *free* and different word processor (not like Microsoft Word);  
- coming from T\TeX\ (created by Donald E. Knuth);  
- the evolution of T\TeX.  

---

Introduction

What is \LaTeX?
What \LaTeX\ is not...
Why using \LaTeX?  
Quality of output
Portability
Focusing on your document
Writing quicker

Structure of a document

A classical article  
Code
Explanation
Complements
A classical presentation
Structure
Example
Inserting images
What is \LaTeX? 

\LaTeX is... 

- a \textit{free} and different word processor (not like Microsoft Word); 
- coming from TEX (created by Donald E. Knuth); 
- the evolution of TEX. 

Here is Donald E. Knuth...
What \LaTeX\ is not...
What \LaTeX{} is not...

\LaTeX{} is not...

- WYSIWYG (What You See Is What You Get);
What \LaTeX is not...

- WYSIWYG (What You See Is What You Get);
- as easy to learn as other typesetting tools.
Why using \LaTeX?
Why using \LaTeX?  

- \textit{quality} of output (not only for formula’s);
Why using \LaTeX? 

- quality of output (not only for formula’s);
- portable (Unix, Dos, Mac, Windows);
Why using \LaTeX? 

- *quality* of output (not only for formula’s);
- *portable* (Unix, Dos, Mac, Windows);
- allows you to *concentrate* on your document’s *content*;
Why using \LaTeX{}?

- *quality* of output (not only for formula’s);
- *portable* (Unix, Dos, Mac, Windows);
- allows you to *concentrate* on your document’s *content*;
- once you know how it works, you *write quicker*;
Why using \LaTeX? 

- *quality* of output (not only for formula’s);
- *portable* (Unix, Dos, Mac, Windows);
- allows you to *concentrate* on your document’s content;
- once you know how it works, you *write quicker*;
- it is *free* of charges!

→ it is used by the most of scientists who work in Applied Sciences (Mathematics, Computer Sciences, Engineering, Chemistry).
Quality of output.

Quality of output...
Quality of output...

- non-\TeX \; document $\rightarrow$ PDF\(^1\) $\rightarrow$ *lack of quality*;
Quality of output...

- non-TEX document $\rightarrow$ PDF$^1$ $\rightarrow$ lack of quality;
- TEX document ($\rightarrow$ PS$^2$) $\rightarrow$ PDF $\rightarrow$ quality.

---

$^1$Portable Document Format

$^2$PostScript
Quality of output.

Quality of output...

- non-\text{T\hback EX} document $\rightarrow$ PDF\footnote{Portable Document Format} $\rightarrow$ lack of quality;
- \text{T\hback EX} document ($\rightarrow$ PS\footnote{PostScript}) $\rightarrow$ PDF $\rightarrow$ quality.

Just see this...
Quality of output.

Quality of output...

- non-\(\TeX\) document → PDF\(^1\) → lack of quality;
- \(\TeX\) document (→ PS\(^2\)) → PDF → quality.

Just see this...

Property

The Taylor serie of the hyperbolic cosecant is given by...

\[
\text{cosech}(x) = \frac{1}{x} + \sum_{n=1}^{\infty} \frac{2(1 - 2^{2n-1})B_{2n}x^{2n-1}}{(2n)!}.
\] (1)

---

\(^1\)Portable Document Format

\(^2\)PostScript
Quality of output...

- non-\TeX\ document $\rightarrow$ PDF$^1$ $\rightarrow$ lack of quality;
- \TeX\ document ($\rightarrow$ PS$^2$) $\rightarrow$ PDF $\rightarrow$ quality.

Just see this...

Property

*The Taylor serie of the hyperbolic cosecant is given by...*

$$\text{cosech}(x) = \frac{1}{x} + \sum_{n=1}^{\infty} \frac{2(1 - 2^{2n-1})B_{2n}x^{2n-1}}{(2n)!}.$$  \hspace{1cm} (1)

It is impossible to have such a finest equation in Word!

---

$^1$Portable Document Format

$^2$PostScript
Portable
Portable?

- PDF and PS formats are used everywhere $\rightarrow$ no compatibility problems!
Focusing on my document?

Focusing on my document?

Merciadri Luca

Introduction
What is \LaTeX?
What \LaTeX is not...
Why using \LaTeX?
Quality of output
Portability
Focusing on your document
Writing quicker

Structure of a document
A classical article
Code
Explanation
Complements
A classical presentation
Structure
Example
Inserting images

other typesetting programs have as output what you see; \LaTeX obliges you to divide in blocks your document: parts; chapters; sections; subsections; \LaTeX automatically makes the hyphenations when necessary; \LaTeX can send the reader dynamically (e.g. see p. XX);
\LaTeX automatically numbers equations, proofs, theorems, etc.; and so on...
other typesetting programs have as output what you see;
other typesetting programs have as output what you see;
\ Latex obliges you to divide in blocks your document:
Focusing on my document?

- other typesetting programs have as output what you see;
- \LaTeX{} obliges you to divide in blocks your document:
  - parts;
other typesetting programs have as output what you see;
\LaTeX\ obliges you to divide in blocks your document:
- parts;
- chapters;
Focusing on my document?

- other typesetting programs have as output what you see;
- \LaTeX\ obliges you to divide in blocks your document:
  - parts;
  - chapters;
  - sections;
Focusing on my document?

- other typesetting programs have as output what you see;
- LaTeX obliges you to divide in blocks your document:
  - parts;
  - chapters;
  - sections;
  - subsections;
Focusing on my document?

- Other typesetting programs have as output what you see;
- \LaTeX{} obliges you to divide in blocks your document:
  - parts;
  - chapters;
  - sections;
  - subsections;
  - subsubsections;
Focusing on my document?

- Other typesetting programs have as output what you see;
- \LaTeX{} obliges you to divide in blocks your document:
  - parts;
  - chapters;
  - sections;
  - subsections;
  - subsubsections;
- \LaTeX{} automatically makes the hyphenations when necessary;
Focusing on my document?

- other typesetting programs have as output what you see;
- \texttt{\LaTeX} obliges you to divide in blocks your document:
  - parts;
  - chapters;
  - sections;
  - subsections;
  - subsubsections;
- \texttt{\LaTeX} automatically makes the hyphenations when necessary;
- \texttt{\LaTeX} can send the reader dynamically (e.g. see p. XX);
other typesetting programs have as output what you see;
\LaTeX{} obliges you to divide in blocks your document:
- parts;
- chapters;
- sections;
- subsections;
- subsubsections;
\LaTeX{} automatically makes the hyphenations when necessary;
\LaTeX{} can send the reader dynamically (e.g. see p. XX);
\LaTeX{} automatically numbers equations, proofs, theorems, etc.;
Focusing on my document?

- other typesetting programs have as output what you see;
- \LaTeX{} obliges you to divide in blocks your document:
  - parts;
  - chapters;
  - sections;
  - subsections;
  - subsubsections;
- \LaTeX{} automatically makes the hyphenations when necessary;
- \LaTeX{} can send the reader dynamically (e.g. see p. XX);
- \LaTeX{} automatically numbers equations, proofs, theorems, etc.;
- and so on...
With \LaTeX, you write quicker.
With LaTeX, you write quicker. Once you know the rudimentary instructions...
With \LaTeX, you write quicker. Once you know the rudimentary instructions... 

- you type math. equations without worrying about it;
With \LaTeX{}, you write quicker. Once you know the rudimentary instructions. . .

- you type math. equations without worrying about it;
- you don’t have to verify your document’s presentation.
Structure of a classical article – code.

A classical article

\documentclass[10pt,a4paper,final,oneside]{article}
\usepackage[latin1]{inputenc}
\usepackage[francais,english]{babel}
\title{The Title}
\date{\today} % the comments are here
\author{The Author}
\location{The Location}
\email{name@provider.com}
\begin{document}
\maketitle
\chapter{\ldots}
Here is some text about the first chapter.
\begin{enumerate}
\item Example 1;
\item Example 2.
\end{enumerate}
After, we will speak about these things, but with no order:
\begin{itemize}
\item Example 1;
\item Example 2.
\end{itemize}
\section{\ldots}
As everybody knows, if $a=b$, and that $b=c$, we have the following formula:
\begin{equation}
  a=c.
\end{equation}
\include{externalexample}
\end{document}
Structure of a classical article – code.

A \LaTeX\ article has a code like this…

\documentclass[10pt,a4paper,final,oneside]{article}
\usepackage[latin1]{inputenc}
\usepackage[francais,english]{babel}
\title{The Title}
\date{\today} % the comments are here
\author{The Author}
\location{The Location}
\begin{document}
\maketitle
\chapter{\ldots}
Here is some text about the first chapter.

We will see the following things, in this order:
\begin{enumerate}
\item Example 1;
\item Example 2.
\end{enumerate}
After, we will speak about these things, but with no order:
\begin{itemize}
\item Example 1;
\item Example 2.
\end{itemize}
\section{\ldots}
As everybody knows, if $a=b$, and that $b=c$, we have the following formula:
\begin{equation}
a=c.
\end{equation}
\include{externalexample}
\end{document}
Structure of a classical article – code.

A \LaTeX{} article has a code like this...

\documentclass[10pt,a4paper,final,oneside]{article}
\usepackage[latin1]{inputenc}
\usepackage[francais,english]{babel}
\title{The Title}
\date{\today} % the comments are here
\author{The Author}
\location{The Location}
\email{name@provider.com}
\begin{document}
\maketitle
\chapter{\ldots}
Here is some text about the first chapter.
\section{\ldots}
As everybody knows, if $a=b$, and that $b=c$, we have the following formula:
\begin{equation}
a=c.
\end{equation}
\end{document}
Structure of a classical article – explanation (1/2).

Introduction
What is \LaTeX{}?
What \LaTeX{} is not.
Why using \LaTeX{}?
Quality of output
Portability
Focusing on your document
Writing quicker

Structure of a document
A classical article
Code
Explanation
Complements
A classical presentation
Structure
Example
Inserting images
Structure of a classical article – explanation (1/2).

Now, see what all these instructions mean...
Structure of a classical article – explanation (1/2).

Now, see what all these instructions mean...

`\documentclass[]{}`

→ represents the class of the document.
Now, see what all these instructions mean...

\documentclass{ }

→ represents the class of the document.

\usepackage{ }

→ allows the user to use a package.
Structure of a classical article – explanation (1/2).

Now, see what all these instructions mean...

\documentclass[]{}

→ represents the class of the document.

\usepackage[]{}

→ allows the user to use a package.

\title{}

→ allows the user to give a title to his document.
Structure of a classical article – explanation (1/2).

Now, see what all these instructions mean...

\documentclass[]{}
→ represents the class of the document.
\usepackage[]{}
→ allows the user to use a package.
\title{}
→ allows the user to give a title to his document.
\begin{document}
→ begins the document.
Now, see what all these instructions mean...

```latex
\documentclass[]{}
```

→ represents the class of the document.

```latex
\usepackage[]{}
```

→ allows the user to use a package.

```latex
\title{}
```

→ allows the user to give a title to his document.

```latex
\begin{document}
```

→ begins the document.

```latex
\maketitle
```

→ prints the title in the document.
Structure of a classical article – explanation (1/2).

Now, see what all these instructions mean...

\documentclass[]{}

→ represents the class of the document.

\usepackage[]{}

→ allows the user to use a package.

\title{}

→ allows the user to give a title to his document.

\begin{document}

→ begins the document.

\maketitle

→ prints the title in the document.

\part{nameofpart}

→ makes the text which follows this instruction being a part (named “nameofpart”), until the following instruction of part.
Structure of a classical article – explanation (2/2).

A very short presentation about \LaTeX\... 

Merciadri Luca

Introduction
What is \LaTeX? What \LaTeX\ is not... Why using \LaTeX? Quality of output Portability Focusing on your document Writing quicker

Structure of a document
A classical article Code Explanation Complements A classical presentation Structure Example Inserting images

\begin{document}

\section*{Introduction}

\subsection*{What is $\LaTeX$?}

\subsection*{What $\LaTeX$ is not...}

\subsection*{Why using $\LaTeX$?}

\subsection*{Quality of output}

\subsection*{Portability}

\subsection*{Focusing on your document}

\subsection*{Writing quicker}

\section*{Structure of a classical article – explanation (2/2)}

\subsection*{Explanation}

\subsection*{Complements}

\end{document}
Structure of a classical article – explanation (2/2).

\include{externalfile}

→ includes another \TeX file (here “externalfile.tex”) in the document, at this place.
\include{externalfile}

→ includes another \TeX file (here “externalfile.tex”) in the document, at this place.

$a$

→ tells \LaTeX that $a$ is a math. element. You must put the $$ between a math. element in \LaTeX.

\end{document}
Structure of a classical article – explanation (2/2).

```latex
\include{externalfile}
```

→ includes another \TeX file (here “externalfile.tex”) in the document, at this place.

\$a\$

→ tells \LaTeX\, that \(a\) is a math. element. You **must** put the $$ between a math. element in \LaTeX.\n
```latex\begin{equation}a=c.\end{equation}\n```

→ is a very important environment; it is useful for long equations, such as the one we wrote for \(\cosh(x)\).
Structure of a classical article – explanation (2/2).

\include{externalfile}  
→ includes another \TeX file (here "externalfile.tex") in the document, at this place.

$a$
→ tells \LaTeX that $a$ is a math. element. You **must** put the $$  \begin{equation} a=c. \end{equation}  \end{document}
→ is a very important environment; it is useful for long equations, such as the one we wrote for \cosh(x).
\end{document}
ends the document.
Structure of a classical article – complements.

There are lots of other commands; for typing equations (either in $ or in equation environment), there is a plain notation.

Here are a few examples.

\[ a^{u} \] → means \( a \) to the power \( u \) (\( a^u \)).

\[ a_{u} \] → means that \( u \) is below \( a \) (\( a_u \)).

\[ \sqrt{a} \] → means that we take the square root of \( a \) (\( \sqrt{a} \)).

\[ \int_{a}^{b} \sin(x)\mathrm{dx} \] → means that we take the integral of \( \sin(x) \) between \( a \) and \( b \) (\( \int_{a}^{b} \sin(x)\mathrm{dx} \)).
Structure of a classical article – complements.

- There are lots of other commands;
Structure of a classical article – complements.

- There are lots of other commands;
- for typing equations (either in $\$$ or in equation environment), there is a plain notation.

Here are a few examples...
Structure of a classical article – complements.

- There are lots of other commands;
- for typing equations (either in $$ or in equation environment), there is a plain notation.

Here are a few examples...

\[ a^{u} \rightarrow \text{means } a \text{ to the power } u \left( a^{u} \right). \]
Structure of a classical article – complements.

- There are lots of other commands;
- for typing equations (either in $$ or in equation environment), there is a plain notation.

Here are a few examples. . .

\[ a^{u} \]
→ means \( a \) to the power \( u \) (\( a^u \)).

\[ a_{u} \]
→ means that \( u \) is below \( a \) (\( a_u \)).
Structure of a classical article – complements.

- There are lots of other commands;
- for typing equations (either in $$ or in equation environment), there is a plain notation.

Here are a few examples. . .

\[ a^{u} \]
→ means \( a \) to the power \( u \) \( (a^u) \).

\[ a_{u} \]
→ means that \( u \) is below \( a \) \( (a_u) \).

\[ \sqrt{a} \]
→ means that we take the square root of \( a \) \( (\sqrt{a}) \).
There are lots of other commands; for typing equations (either in $$ or in equation environment), there is a plain notation.

Here are a few examples...

\( a^{u} \)
→ means \( a \) to the power \( u \) (\( a^{u} \)).

\( a_{u} \)
→ means that \( u \) is below \( a \) (\( a_{u} \)).

\( \sqrt{a} \)
→ means that we take the square root of \( a \) (\( \sqrt{a} \)).

\( \int_{a}^{b} \sin(x) \mathrm{d}x \)
→ means that we take the integral of \( \sin(x) \) between \( a \) and \( b \) (\( \int_{a}^{b} \sin(x) \mathrm{d}x \)).
Structure of a classical presentation (1/2).
Structure of a classical presentation (1/2).

A presentation follows nearly the same instructions as for an article, but there are a few new instructions. Some of these new instructions are:
Structure of a classical presentation (1/2).

A presentation follows nearly the same instructions as for an article, but there are a few new instructions. Some of these new instructions are:

```latex
\documentclass{beamer}
```

→ the document’s class is not the same as before.
A very short presentation about \LaTeX\ldots

Merciadri Luca

Introduction
What is \LaTeX? What \LaTeX{} is not. . .
Why using \LaTeX{}?
Quality of output
Portability
Focusing on your document
Writing quicker

Structure of a document
A classical article
Code
Explanation
Complements
A classical presentation
Structure
Example
Inserting images

Introduction

A presentation follows nearly the same instructions as for an article, but there are a few new instructions. Some of these new instructions are:

\documentclass{beamer}
→ the document’s class is not the same as before.

\begin{frame}
→ begins a frame.

Structure of a classical presentation (1/2).

A presentation follows nearly the same instructions as for an article, but there are a few new instructions. Some of these new instructions are:

\documentclass{beamer} 

→ the document’s class is not the same as before.

\begin{frame} 

→ begins a frame.

\frametitle{example} 

→ gives the name “example” to this frame.
Structure of a classical presentation (1/2).

A presentation follows nearly the same instructions as for an article, but there are a few new instructions. Some of these new instructions are:

\documentclass{beamer}
→ the document’s class is not the same as before.

\begin{frame}
→ begins a frame.

\frametitle{example}
→ gives the name “example” to this frame.

To make a specific item (e.g. the item “example 2”) only appearing on frames $x$, and $y$ (if they exist), you have to write:
Structure of a classical presentation (1/2).

A presentation follows nearly the same instructions as for an article, but there are a few new instructions. Some of these new instructions are:

\documentclass{beamer}
→ the document’s class is not the same as before.
\begin{frame}
→ begins a frame.
\frametitle{example}
→ gives the name “example” to this frame.

To make a specific item (e.g. the item “example 2”) only appearing on frames \( x \), and \( y \) (if they exist), you have to write:

\item<\(x,y\)> example 2
To put a specific text on a specific frame \((x)\), we will use:
Structure of a classical presentation (2/2).

To put a specific text on a specific frame ($x$), we will use:
\[\text{\only<x>{specific text}}\]
This presentation is totally written using \textsc{\LaTeX} thanks to MikTeX distribution on Windows XP.
Inserting images – Introduction.

Inserting images in \LaTeX{} documents is a little bit fair-haired but keep in mind that...
Inserting images – Introduction.

Inserting images in \LaTeX\ documents is a little bit fair-haired but keep in mind that...
If you convert \TeX\ → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:
don't forget the "figure" environment;
h stands for "here";
you must include the package "graphicx" in the preamble;
"scale" means "echelle" (ratio);
your picture must be in PNG format;
the "caption" command gives a legend to the image;
don't forget to close the "figure" environment!
Inserting images in \LaTeX documents is a little bit fair-haired but keep in mind that...

If you convert \TeX → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:
don't forget the "figure" environment;
h stands for "here";
you must include the package "graphicx" in the preamble;
"scale" means "échelle" (ratio);
your picture must be in PNG format;
the "caption" command gives a legend to the image;
don't forget to close the "figure" environment!
Inserting images in \LaTeX\ documents is a little bit fair-haired but keep in mind that... If you convert \TeX\ → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:
Inserting images in \LaTeX{} documents is a little bit fair-haired but keep in mind that... If you convert \TeX{} → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:
- don’t forget the “figure” environment;
Inserting images in \LaTeX{} documents is a little bit fair-haired but keep in mind that... If you convert \TeX{} → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:

- don’t forget the “figure” environment;
- $h$ stands for “here”;

Inserting images – Introduction.
Inserting images in \LaTeX documents is a little bit fair-haired but keep in mind that...
If you convert \TeX → \PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:

- don’t forget the “figure” environment;
- \textit{h} stands for “here”;
- you must include the package “graphicx” in the preamble;
Inserting images in \LaTeX documents is a little bit fair-haired but keep in mind that... If you convert \TeX → \PS → \PDF (normal conversion), you have to use a command like:
\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:
- don’t forget the “figure” environment;
- \textit{h} stands for “here”;
- you must include the package “graphicx” in the preamble;
- “scale” means “échelle” (ratio);
Inserting images in \LaTeX\ documents is a little bit fair-haired but keep in mind that... If you convert \TeX\ → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:
- don’t forget the “figure” environment;
- $h$ stands for “here”;
- you must include the package “graphicx” in the preamble;
- “scale” means “échelle” (ratio);
- your picture must be in PNG format;
Inserting images in \LaTeX \documentclass{} is a little bit fair-haired but keep in mind that...

If you convert TeX → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[scale=1, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:

- don’t forget the “figure” environment;
- \(h\) stands for “here”;
- you must include the package “graphicx” in the preamble;
- “scale” means “échelle” (ratio);
- your picture must be in PNG format;
- the “caption” command gives a legend to the image;
Inserting images in \LaTeX{} documents is a little bit fair-haired but keep in mind that...
If you convert \TeX{} → PS → PDF (normal conversion), you have to use a command like:

\begin{figure}[h]
\includegraphics[\textwidth, bb=0 0 829 397]{yourimage.png}
\caption{This is a caption.}
\end{figure}

Precisions:

- don’t forget the “figure” environment;
- \textit{h} stands for “here”;
- you must include the package “graphicx” in the preamble;
- “scale” means “échelle” (ratio);
- your picture must be in PNG format;
- the “caption” command gives a legend to the image;
- don’t forget to close the “figure” environment!
Inserting images – Example.

Here is an example of six different real functions. You can see that the graph is perfect.
Inserting images – Example.

Here is an example of six different real functions. You can see that the graph is perfect.
Conclusion.
\textsc{Conclusion.}

\textsc{LaTeX} . . .

- all the commands begin with a slash;
Conclusion.

**\LaTeX** . . .

- all the commands begin with a slash;
- very useful for reports, whatever you want to speak about;
Introduction

What is \LaTeX? What \LaTeX{} is not... Why using \LaTeX{}? Quality of output Portability Focusing on your document Writing quicker

Structure of a document

A classical article Code Explanation Complements A classical presentation Structure Example Inserting images

Con...
Conclusion.

\textsc{\LaTeX} ... 

- all the commands begin with a slash; 
- very useful for reports, whatever you want to speak about; 
- very easy to write equations if you are a little bit motivated. 

\textbf{Adopt \LaTeX}!
Questions.
Questions.

- Why is \LaTeX{} so nice?
Questions.

Why is \LaTeX{} so nice?

→ interesting characteristics:
Questions.

Why is \LaTeX so nice?
→ interesting characteristics:
  1. quality of output;
Why is \LaTeX{} so nice?

\rightarrow \text{interesting characteristics:}

1. quality of output;
2. portable;

Why is \LaTeX{} so nice?
Why is \LaTeX{} so nice?

→ interesting characteristics:

1. quality of output;
2. portable;
3. allows you to concentrate;
Why is \LaTeX\ so nice?

→ interesting characteristics:

1. \textit{quality} of output;
2. \textit{portable};
3. allows you to \textit{concentrate};
4. once you know how it works, you \textit{write quicker};
Why is \LaTeX{} so nice?

→ interesting characteristics:

1. \textit{quality} of output;
2. \textit{portable};
3. allows you to \textit{concentrate};
4. once you know how it works, you \textit{write quicker};
5. it is \textit{free of charges}!
Why is \LaTeX{} so nice?
→ interesting characteristics:
1. quality of output;
2. portable;
3. allows you to concentrate;
4. once you know how it works, you write quicker;
5. it is free of charges!

Can you guess what does WYSIWYG means?
Questions.

- Why is $\text{\LaTeX}$ so nice?
  → interesting characteristics:
  1. quality of output;
  2. portable;
  3. allows you to concentrate;
  4. once you know how it works, you write quicker;
  5. it is free of charges!

- Can you guess what does WYSIWYG means?
  → “What You See Is What You Get”. When you change the style of the document, you directly see it (Word). It is not the case in $\text{\LaTeX}$. 
Why is $\LaTeX$ so nice?

→ interesting characteristics:

1. quality of output;
2. portable;
3. allows you to concentrate;
4. once you know how it works, you write quicker;
5. it is free of charges!

Can you guess what does WYSIWYG means?

→ “What You See Is What You Get”. When you change the style of the document, you directly see it (Word). It is not the case in $\LaTeX$.

What are the output formats of $\LaTeX$?
Questions.

Why is \LaTeX so nice?
→ interesting characteristics:
1. quality of output;
2. portable;
3. allows you to concentrate;
4. once you know how it works, you write quicker;
5. it is free of charges!

Can you guess what does WYSIWYG means?
→ “What You See Is What You Get”. When you change the style of the document, you directly see it (Word). It is not the case in \LaTeX.

What are the output formats of \LaTeX?
→ they are 2:
Why is $\LaTeX$ so nice?

→ interesting characteristics:

1. quality of output;
2. portable;
3. allows you to concentrate;
4. once you know how it works, you write quicker;
5. it is free of charges!

Can you guess what does WYSIWYG means?

→ “What You See Is What You Get”. When you change the style of the document, you directly see it (Word). It is not the case in $\LaTeX$.

What are the output formats of $\LaTeX$?

→ they are 2:

1. PDF (Portable Document Format);
Why is \LaTeX{} so nice?

→ interesting characteristics:

1. **quality** of output;
2. **portable**;
3. allows you to **concentrate**;
4. once you know how it works, you **write quicker**;
5. it is **free of charges**!

Can you guess what does **WYSIWYG** means?

→ “What You See Is What You Get”. When you change the style of the document, you *directly* see it (Word). It is not the case in \LaTeX{}.

What are the output formats of \LaTeX{}? → they are 2:

1. **PDF** (Portable Document Format);
2. **PS** (PostScript).
Questions.

Why is \LaTeX{} so nice?

→ interesting characteristics:

1. quality of output;
2. portable;
3. allows you to concentrate;
4. once you know how it works, you write quicker;
5. it is free of charges!

Can you guess what does WYSIWYG means?

→ “What You See Is What You Get”. When you change the style of the document, you directly see it (Word). It is not the case in \LaTeX{}.

What are the output formats of \LaTeX{}?

→ they are 2:

1. PDF (Portable Document Format);
2. PS (PostScript).

What is the aim of the include command?
Questions.

- Why is **\LaTeX** so nice?
  → interesting characteristics:
  1. *quality* of output;
  2. *portable*;
  3. allows you to *concentrate*;
  4. once you know how it works, you *write quicker*;
  5. it is *free of charges*!

- Can you guess what does **WYSIWYG** means?
  → “What You See Is What You Get”. When you change the style of the document, you *directly* see it (Word). It is not the case in **\LaTeX**.

- What are the output formats of **\LaTeX**?
  → they are 2:
  1. *PDF* (Portable Document Format);

- What is the aim of the *include* command?
  → includes another **\TeX** file in the document, at the place where it is used.
There is only the point 3, page 12, which comes from the two following references. There are plenty other references on the Web!

- **Marc Baudoin.**
  Apprends latex !, 1997.
  *Ecole nationale supérieure de techniques avancées.*

- **Maxime Bonjean.**
  Introduction to latex, 2006.
  *ULg.*

This presentation is also available at [http://www.student.montefiore.ulg.ac.be/~merciadri](http://www.student.montefiore.ulg.ac.be/~merciadri).