

```
\documentclass{article}
```

```
% o comando \documentclass{} define uma série de características do  
% documento (margens, tipo de numeração das páginas etc.).
```

```
% É muito frequente querermos fazer alguns ajustes finos (nem sempre  
% tão finos assim), ou precisarmos incrementar nosso documento com  
% alguma estrutura mais específica. É nesse momento que os pacotes  
% ('`packages'') e estilos ('`style files'') aparecem.
```

```
%  
% O uso de pacotes assemelha-se à inclusão de bibliotecas em  
% programação, tendo a função de tornar disponíveis alguns comandos  
% que estendem a funcionalidade dos comandos básicos. Assim como  
% ocorre em várias linguagens de programação, é também possível  
% criarmos nossos próprios pacotes (ou estilos). Esse assunto será  
% abordado (oxalá) futuramente.
```

```
%  
% A sintaxe para inclusão de pacotes ou estilos é  
% \usepackage[argumentos opcionais]{nome do pacote ou estilo}
```

```
\usepackage[utf8]{inputenc} % Permite processar este arquivo, que usa  
% a codificação utf8.  
\usepackage[brazil]{babel} % Carrega padrões de hifenação do português.
```

```
% Estamos agora prestes a iniciar o documento propriamente dito. O  
% documento é o exemplo do que se chama ambiente  
% ('`environment''). Os ambientes têm a seguinte sintaxe  
%  
% \begin[argumentos opcionais]{nome do ambiente}  
% o conteúdo do ambiente vem aqui...  
% end{ambiente}
```

```
\begin{document} % Todo o conteúdo propriamente dito de um documento é  
% colocado entre \begin{document} e \end{document}
```

```
% Os próximos dois comandos definem variáveis da classe ``article''  
% (\documentclass{article}, na primeira linha do documento). O título  
% e o autor. Essas definições poderiam ter sido feitas antes  
% de \begin{document}.
```

```
\title{Aula zero}  
\author{Tássio Naia dos Santos}
```

```
\maketitle % Esse comando efetivamente diz ao LaTeX para imprimir a  
% ``capa'' do texto.
```

Finalmente, o primeiro parágrafo do texto! Mais uma frase! E ainda
outra\footnote{Certo, nesse passo podemos ir bem longe\dots}!

O \LaTeX\ não se importa com o número de linhas que você deixa
entre os parágrafos no arquivo original. Basta deixar uma, e ele
iniciará um novo parágrafo, com espaçamento correto e tudo o mais.

Eis um novo parágrafo.
Note que, se não houver uma linha em branco entre as frases (no
texto-fonte), elas serão colocadas no mesmo parágrafo.

Uma coisa parecida acontece com os espaços. Deixar um ou
muitos espaços entre as palavras, não fará diferença na
saída gerada pelo \LaTeX.

```
\section{Mãos à obra---um pouco de filosofia} % \section é apenas um  
% dos vários comandos que permitem a  
% estruturação do nosso texto.
```

Até o presente momento, o \LaTeX\ só se mostrou um jeito deselegante e
bagunçado de escrever documentos. Algumas perguntas (e brados de
indignação) são bastante naturais nesse ponto. Provavelmente você já
pensou que gastaria muito menos tempo para gerar a mesma saída com o
seu velho e conhecido editor de texto, e é bem possível que tenha
torcido o nariz para essa complexidade desnecessária que enche o
documento de sequências (ai, o trema caiu!) de comandos.

Por outro lado, com alguma sorte você ficou agradavelmente surpreso
com a facilidade com que o \LaTeX\ gerou uma nota de rodapé, e efetuou
a correta divisão silábica de uma palavra ou outra alguns parágrafos
atrás.

Pode ser que você esteja um pouco encucado a respeito da quantidade de

mudanças que o `\LaTeX` fez no texto. Se você, como a grande maioria das pessoas que usam processadores de texto, está familiarizado com a rotina de seleção de estilos, tipos`\footnote`{As famosas `\emph`{fontes}.}, e configurações de margens e espaçamentos, é bem provável que esteja se perguntando `\emph`{``mas quem é esse `\LaTeX` para dizer o que é melhor para o `\Huge` meu} texto?''}

Vamos por partes. Se você já deu uma olhadela no arquivo-fonte deste texto, deve ter notado que, excluindo-se os comentários, o documento está recheado de comandos que dão um `\emph`{significado} à pedaços do texto. Essa abordagem é fundamentalmente diferente da visão orientada a formatação, que é a base de editores WYSIWYG`\footnote`{What You See Is What You Get.} de texto.

Para tornar uma longa história curta, ao empregarmos `\LaTeX` como ferramenta de geração de nossos textos, estamos basicamente separando a `\emph`{composição} e a `\emph`{formatação} do texto. E que vantagem nós levamos nisso? Bem, todo documento estruturado carrega um determinado padrão. Na esmagadora maioria dos casos, queremos que todas as seções de nossos relatórios, ou capítulos de nossos livros, ou legendas de nossas tabelas sejam impressas com a mesma formatação. Não importa que sejam da fonte `\emph{x}`, tamanho `\emph{y}`, desde que sigam consistentemente o padrão.

Ora, quando dotamos os elementos de nosso texto de `\emph`{significado}, fica muito mais simples fazer isso`\footnote`{Tão simples que até um computador pode fazer!}. Basta dizer que `\emph`{as notas de rodapé devem ser numeradas sequencialmente, e colocadas nem uma pilha (de altura máxima `\emph{z}`) na parte inferior da página; devem ser escritas em uma fonte `\emph{w\%}` menor do que a da fonte usada no corpo do texto; e devem apresentar um pequeno traço sobre a primeira delas, para dividi-las do texto}. Sacou?

Mas a história não acaba aí. Imagine que pudéssemos apenas nos preocupar com o conteúdo do texto, e que depois fosse possível entregar o manuscrito a um diagramador profissional---amigo nosso---que cuidaria de todos os detalhes da aparência do texto final: do espaçamento entre as palavras em uma linha até a distribuição das imagens pelo texto e a confecção dos índices, da conferência das inter-referências das seções até a formatação das referências bibliográficas! Bem, você sabe aonde eu quero chegar`\dots` o `\LaTeX` é camarada.

Se essa carreira de argumentos não foi o suficiente para impressionar, chegou a hora de pegar pesado---estou falando da escrita de fórmulas e expressões matemáticas. A necessidade de um sistema que permitisse a geração automática de equações de forma decente foi uma das motivações da criação do `\TeX`, base sobre a qual o `\LaTeX` se apoia`\footnote`{(Ai essa reforma ortográfica!) O `\LaTeX` é um conjunto de macros `\TeX` que facilita seu uso e estende sua funcionalidade.}. O resultado disso é que podemos exclamar`\footnote`{Após uma rápida consulta ao livro-texto de física III`\dots`}, sem a menor dificuldade:

```
\[
\oint{\vec{B}}\cdot\emph{d}\vec{s}}=\mu_0\epsilonpsilon_0\frac{\emph{d}\phi_E}{\emph{dt}}+\mu_0 i_{enc}.
\]
```

% calma, calma...

Mas chega de enrolação---vejamos um pouco de `\LaTeX` pra valer. As próximas duas seções introduzem algumas ideias básicas, fundamentais para o uso dessa ferramenta poderosa.

Um último ponto antes de passarmos à técnica. Gosto de fazer a seguinte comparação: aprender a usar o `\LaTeX` é como aprender a usar uma calculadora científica (ou uma planilha). Não é trivial, mas faz toda a diferença.

`\section`{Ambientes}

Ambientes são uma parte importante de um documento. Não é incomum identificarmos partes do texto que apresentam características peculiares, que as diferenciam do texto que as rodeia. Ambientes (`\emph`{environments}) em `\LaTeX` são sempre iniciados por ```\textbackslash begin''` e terminados por ```\textbackslash % o`
% caractere \ é reservado para uso
% pelo LaTeX, e é preciso usar o
% comando \textbackslash para
% imprimi-lo.

end''. Alguns exemplos de ambientes são listas, tabelas, citações e figuras. Em geral, ambientes

```
\begin{itemize}
\item envolvem pequenos trechos do documento (que devem receber
tratamento especial);
\item são aninháveis (podemos ter um ambiente dentro de outro
ambiente\footnote{Basta observar que o próprio documento é um
ambiente: \emph{document}.}); e
\item similarmente a boa parte dos comandos, como
```

```
\begin{enumerate} % este é um ambiente de numeração que está aninhado
                  % de um ambiente para listagem.
```

```
\item \textbackslash emph;
\item \textbackslash footnote;
\item \textbackslash section; etc.
```

```
\end{enumerate}
podem receber parâmetros.
\end{itemize}
```

```
\section{Alguns comandos}
```

Antes de dar a aula por terminada, há ainda alguns comandos a apresentar. Usamos o `\LaTeX` para formatação de texto, mas, por vezes, há trechos do texto que não queremos processados por ele. Um típico caso ocorre quando queremos inserir o trecho de um programa em nosso documento. Programas geralmente são escritos em uma fonte monoespçada `\footnote`{Um tipo monoespçado é aquele em que todos os caracteres têm a mesma largura.}, e frequentemente usamos caracteres em branco para estruturar (endentar) o código, tornando mais fácil a sua compreensão. Mas se apenas escrevermos o código, digamos, de uma função escrita em C:

```
int fatorial (int n) {
    if (n == 1)
        return 1;
    return n * fat (n - 1);
}
```

não é nenhuma surpresa que o resultado não seja o desejado.

Quando queremos que parte do texto seja mantida sem formatação, ou seja, quando queremos que o `\LaTeX` não o processe, recorremos ao ambiente `\emph{verbatim}`.

```
\begin{verbatim}
int fatorial (int n) {
    if (n == 1)
        return 1;
    return n * fat (n - 1);
}
\end{verbatim}
```

Existe ainda outro modo de inserir texto não-formatado no documento, sem que para isso tenhamos que iniciar um ambiente. É o comando `\verb|\verb|`. Para usá-lo, basta encerrar a sequência de caracteres que queremos ```\emph{verbatimizar}''` entre duas ocorrências de um caractere delimitador.

```
% Por exemplo, \verb+abc+ diz ao processador do texto para imprimir
% sem formatação todos os caracteres entre os dois sinais +. Note que
% qualquer caractere pode ser um delimitador, e que o delimitador não
% pode fazer parte do texto a ser impresso sem formatação.
```

Outro exemplo: `\verb:SOMA MOVE.B #5,D2:.`

Outro comando que é usado com frequência é a ênfase (`\verb+\emph+`). Esse comando põe em destaque o texto que lhe é passado como argumento.

```
\section{Resumo da ópera}
```

Segue uma lista dos comandos apresentados nessa ``aula'', com uma pequena descrição.

```
\begin{itemize}
\item \verb+%+ permite inserir comentários que se estendem até o fim da linha;
\item \verb+\documentclass{}+ especifica as configurações básicas do
    documento que vamos escrever;
\item \verb+begin{}+ (document, itemize, enumerate, verbatim) e
    \verb+end{}+ delimitam ambientes;
\item \verb+\section{}+ marca o início de uma seção;
\item \verb+\emph{}+ enfatiza;
\item \verb+\footnote{}+ insere uma nota de rodapé;
\item \verb+\dots+, \verb+\textbackslash+, \verb+\TeX+, \verb+\LaTeX+,
    \verb+\%+ peritem imprimir \verb+\dots\ \textbackslash\ \TeX\ \LaTeX\
    (respectivamente); e
\item \verb+\verb+ impede que um ceto texto seja processado.
\end{itemize}
```

E por hora basta. Até a próxima!

```
\end{document}
```