

# L<sup>A</sup>T<sub>E</sub>Xação

Mais uma apostila de L<sup>A</sup>T<sub>E</sub>X

Tássio Naia dos Santos

11 de Abril de 2013

## Sobre

Este documento nasce como um material de apoio a oficinas de  $\LaTeX$ . As referidas oficinas são oferecidas pelo Grupo de Estudos de Software e Cultura Livres da Escola Politécnica da Universidade de São Paulo, o PoliGNU, com o apoio do Centro de Competência em Software Livre (CCSL) do Instituto de Matemática e Estatística da Universidade de São Paulo (IME).

Este texto está licenciado sob a *GNU Free Documentation License* (uma cópia está anexa ao fim deste documento). Resumidamente, tens o direito de distribuir cópias, modificadas ou não, com a condição de mantê-las sob a mesma licença. O texto é hoje um projeto hospedado no Gitorious©. Todos os arquivos empregados para produzir o texto (exceto as referências) podem ser baixados em <https://gitorious.org/material-latex/>.

Vimos a saber que, apesar de todo o cuidado, os filhos de uma gafe e um senhor descuido perderam-se por estas páginas, enquanto o café era servido. Caso encontre algum desses erros, errando por aí afora, pedimos o favor de identificá-lo, para que possamos apaziguar uma mãe aflita. Nosso contato é [poli@gnu.org](mailto:poli@gnu.org).<sup>1</sup>

Grato!  
o autor

---

<sup>1</sup>E nosso sítio web é [polignu.org](http://polignu.org).

a José Augusto

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>6</b>
1.1	A metáfora . . . . .	6
1.2	Sinopse da Ópera . . . . .	6
1.3	O que dá pra fazer . . . . .	6
<b>2</b>	<b>Rotina de trabalho</b>	<b>6</b>
<b>3</b>	<b>Primeiro documento</b>	<b>7</b>
3.1	Texto e sequências de controle . . . . .	7
3.2	Um documento simples . . . . .	9
3.3	Parâmetros . . . . .	10
3.4	Regiões . . . . .	10
3.5	Acentuação: para além do ASCII . . . . .	11
3.6	Pacotes . . . . .	12
3.7	Caracteres reservados . . . . .	13
3.8	Palavras de controle e texto . . . . .	13
3.9	Texto sem formatação . . . . .	14
3.10	Alguma tipografia . . . . .	15
3.10.1	Hífens e hifenação . . . . .	16
3.10.2	Ligaduras e kerning . . . . .	17
3.10.3	Sobre espaçamento horizontal . . . . .	17
<b>4</b>	<b>Estruturando o texto</b>	<b>18</b>
4.1	Títulos, autor e data de documentos . . . . .	18
4.2	Marcando a anatomia . . . . .	19
4.3	Sumários . . . . .	19
4.4	Referenciando elementos do texto . . . . .	20
<b>5</b>	<b>Alguns elementos do texto</b>	<b>20</b>
5.1	Listas . . . . .	20
5.2	Citando textualmente . . . . .	21
5.3	Alinhamento . . . . .	21
5.3.1	Texto não-justificado . . . . .	21
5.4	Elementos flutuantes . . . . .	21
5.4.1	Figuras . . . . .	22
5.4.2	Tabelas . . . . .	23
<b>6</b>	<b>Expandindo o L<sup>A</sup>T<sub>E</sub>X</b>	<b>23</b>
6.1	Criando comandos . . . . .	25
6.2	Modificando comandos . . . . .	25
6.3	Criando e modificando ambientes . . . . .	26
6.4	Estender pra quê? . . . . .	26
<b>7</b>	<b>Múltiplos arquivos</b>	<b>26</b>
7.1	\input . . . . .	27
7.2	\include e \includeonly . . . . .	28

<b>8</b>	<b>Símbolos</b>	<b>28</b>
<b>9</b>	<b>Matemática</b>	<b>29</b>
9.1	Entrando no modo matemático . . . . .	29
9.2	Índices e expoentes . . . . .	29
<b>10</b>	<b>Aspectos estruturais</b>	<b>31</b>
10.1	Parágrafos marginais . . . . .	31
10.2	Arquivos auxiliares . . . . .	32
10.3	Comandos frágeis . . . . .	32
<b>11</b>	<b>Bibliografia &amp; Cia.: Bib<math>\TeX</math></b>	<b>33</b>
11.1	Fazendo no muque . . . . .	33
11.2	Bib $\TeX$ . . . . .	34
<b>12</b>	<b>Índices remissivos &amp; Cia.</b>	<b>35</b>
12.1	Índices remissivos . . . . .	35
12.2	Glossários . . . . .	36
<b>13</b>	<b>E agora José?</b>	<b>37</b>
<b>14</b>	<b>Utilidades</b>	<b>37</b>
	<b>GNU Free Documentation License</b>	<b>39</b>
	<b>Índice Remissivo</b>	<b>48</b>
	<b>Referências</b>	<b>49</b>

# 1 Introdução

## 1.1 A metáfora

L<sup>A</sup>T<sub>E</sub>X se apoia fortemente numa certa relação entre apresentação e conteúdo do texto, cujo conhecimento pode poupar-nos (ou melhor, auxiliar-nos a lidar com) algumas dores de cabeça: é o princípio de que o *ritmo visual* de um texto deve enfatizar sua estrutura. Por exemplo: a formatação consistente de títulos de seções, destacando em que ponto se iniciam, realça a coordenação entre os trechos que compõem o documento.

Esse pressuposto parece válido para a vasta maioria dos escritos, livros, páginas da internet, publicações de caráter técnico, tais como relatórios, monografias, cartas, etc. A fatia deixada de fora abarca produções caracterizadas por alguma inconstância, defasagem intencional ou arritmia entre o conteúdo e a formatação — como trabalhos artísticos.

## 1.2 Sinopse da Ópera

Donald E. Knuth criou T<sub>E</sub>X, um sistema de tipografia digital muito (!! ) poderoso, e extremamente flexível.

[T<sub>E</sub>X is] *a new typesetting system intended for the creation of beautiful books—and especially for books that contain a lot of mathematics.*

Knuth—The T<sub>E</sub>Xbook [Knu84]

Leslie Lamport criou o L<sup>A</sup>T<sub>E</sub>X, que, a grosso modo, é uma interface mais simplista para o uso do T<sub>E</sub>X. Uma preocupação do L<sup>A</sup>T<sub>E</sub>X é que, ao usá-lo, tenhamos foco no conteúdo, na estrutura do que estamos a compor<sup>2</sup>. Busca separar as etapas de composição conceitual e visual do texto — em contraposição ao modelo de edição de texto manifesto nos programas mais populares hoje. Nos últimos, o paradigma é: *o que você vê é o que você obtém*<sup>3</sup>; ao usar L<sup>A</sup>T<sub>E</sub>X, *o que você vê é o que você quis dizer*<sup>4</sup>.

## 1.3 O que dá pra fazer

Compor textos belíssimos. (E, por que não? Compor textos horrorosos.) Na prática, veremos em breve, é simples produzir documentos pdf, ps, e dvi; documentos com diagramas (que podem ser desenhados usando o próprio sistema, ou importando imagens jpg, eps, pdf, etc.), tabelas, versos, referências bibliográficas, índices, hiperlinks, e muitas outras coisas.

# 2 Rotina de trabalho

Escrever um documento usando L<sup>A</sup>T<sub>E</sub>X não é muito diferente de escrever um documento numa máquina de escrever, embora o resultado seja bastante diverso. Em geral, você irá abrir um programa para edição de texto<sup>5</sup>, escreverá o texto (também chamado de texto *fonte*, por

---

<sup>2</sup>Como consequência dessa abordagem, uma série de decisões sobre a formatação do documento é tomada de antemão pelo L<sup>A</sup>T<sub>E</sub>X, fornecendo-nos uma primeira aparência para o texto — que, não obstante bastar em geral, é passível de alteração.

<sup>3</sup>Conhecido pela sigla em inglês w<sub>Y</sub>S<sub>I</sub>W<sub>X</sub>G, *what you see is what you get*.

<sup>4</sup>Do inglês: *what you see is what you mean* (w<sub>Y</sub>S<sub>I</sub>W<sub>Y</sub>M).

<sup>5</sup>Há programas sofisticados para a edição de documentos L<sup>A</sup>T<sub>E</sub>X, mas este não é nosso foco aqui.

ser a origem de onde emana o documento em sua forma final), e pedirá ao  $\LaTeX$  que gere o documento `pdf` (ou `ps`, ou `dvi`) que desejar. Simples assim.

Não abordaremos aqui o processo de instalação do  $\LaTeX$ , ou como preparar o seu computador para processar os arquivos `tex`. A boa notícia é que essa é uma tarefa simples. Há várias páginas na internet que explicam detalhadamente como instalar o programa, independentemente de qual seja o sistema do seu computador. A seguir listamos algumas referências úteis para interessados em  $\LaTeX$ .

- *$\LaTeX$ , A Document Preparation System*, de Leslie Lamport (criador do  $\LaTeX$ ) [[Lam94](#)],
- *The Not So Short Introduction to  $\LaTeX$* , de Tobias Oetiker [[O+10](#)],
- *The  $\TeX$ book*, de Donald E. Knuth (criador do  $\TeX$ , que é a base sobre a qual se assenta o  $\LaTeX$ ) [[Knu84](#)],
- Wiki brasileiro de  $\TeX$ : [www.tex-br.org](http://www.tex-br.org),
- Getting to Grips with  $\LaTeX$ , de Andrew Roberts: <http://www.andy-roberts.net/misc/latex/>,
- Apostila de  $\LaTeX$  da Universidade Federal Fluminense: [www.telecom.uff.br/pet/petws/downloads/apostilas/LaTeX.pdf](http://www.telecom.uff.br/pet/petws/downloads/apostilas/LaTeX.pdf),
- $\TeX$  Users Group: [www.tug.org](http://www.tug.org),
- Comprehensive  $\TeX$  Archive Network: [www.ctan.org](http://www.ctan.org).

## 3 Primeiro documento

### 3.1 Texto e sequências de controle

Quando você escreve um texto  $\LaTeX$ , já dissemos, a maior parte do tempo você está escrevendo como se usasse uma máquina de escrever comum (talvez você nunca tenha usado uma, mas provavelmente pode imaginar como é). Mas não todo o tempo.

Uma primeira diferença das máquinas de escrever é o espaçamento. Muitos textos são feitos de modo a formarem retângulos na página (o afamado *alinhamento justificado*). Mas, para que as linhas tenham o mesmo comprimento, é preciso hifenar<sup>6</sup> algumas palavras, e ainda alargar ou comprimir ligeiramente os espaços vazios do texto (essa é uma tarefa complicada, que o  $\LaTeX$  desempenha exemplarmente). Não é surpresa que o sistema tenha um modo diferente de lidar com o espaçamento que colocamos no texto. Por exemplo: colocar um espaço entre palavras faz com que elas fiquem separadas por um espaço (até aí, nenhuma surpresa). Mas colocar dois, três, ou cinquenta espaços entre um par de palavras tem o mesmo efeito que colocar apenas um. E mais: quebrar a linha no texto não causa uma quebra de linha no texto final. Observe atentamente o exemplo abaixo. O caractere ‘`\_`’ indica um espaço em branco.

---

<sup>6</sup>Dividir em sí-la-bas, vide seção [3.10.1](#).

Separado por um espaço	
Separado por um espaço	Separado por um espaço
Separado por dois espaços	Separado por um espaço Separado por dois espaços
Separado por vários espaços!	Separado por vários espaços! Separado por ai caramba!
Separado por ai caramba!	Duas quebras de linha seguidas (criando uma linha em branco) iniciam um novo parágrafo, o que não acontece quando há apenas uma.
Duas quebras de linha seguidas (criando uma linha em branco) iniciam um novo parágrafo, o que não acontece quando há apenas uma.	Existe o comando barra-barra.
Se quiser forçar uma quebra de linha \\ Existe o comando barra-barra.	

Até o momento, falamos de texto puro e simples. Caracteres e espaços. Eventualmente, no entanto, você desejará acrescentar algo ao texto além de palavras. Pode ser que queira *ênfatizar alguma passagem*, ou

“... citar algo que, alguma vez, muito apropriadamente, foi dito ou escrito, e que ilustra bem o que quer que seja.”

Autor Conhecido

Em situações como essas, empregam-se *sequências de controle*, que especificam o papel de alguma palavra, região ou ponto do texto.

Por exemplo, empreguei uma certa palavra de controle (*control word*) pouco acima, para dizer ao L<sup>A</sup>T<sub>E</sub>X que “Texto e sequências de controle” é um título de seção. Sabendo disso, o sistema pode fazer várias coisas, como

1. descobrir o número da seção,
2. alterar o tamanho e peso da fonte empregada para escrever as palavras do título (com o número da seção ao lado), e
3. acrescentar uma linha ao sumário do texto com o número da página em que a seção começa.

Sequências de controle iniciam por uma barra ‘\’. A maior parte delas, a que chamamos *palavras de controle*, é formada pela barra seguida por letras (consideramos aqui letras os caracteres ‘A’ a ‘Z’, e ‘a’ a ‘z’). Há um outro tipo de sequência de controle, que chamaremos aqui de *caractere de controle* (control character), que consiste de uma barra seguida de um caractere não-letra, por exemplo ‘\-’, e ‘\{’ (a função dessas sequências será explicada nas seções 3.10.1 e 9).

Naturalmente, surge a pergunta: mas e se eu quiser usar uma \ no meu texto? De fato, se você digitar “amigo\inimigo” para obter amigo\inimigo, terá uma surpresa: muito provavelmente o L<sup>A</sup>T<sub>E</sub>X reclamará de uma *undefined control sequence \inimigo*. Veremos que alguns caracteres são “reservados” pelo L<sup>A</sup>T<sub>E</sub>X para algumas funções especiais. Citamos



aqui os caracteres ‘%’, ‘\$’ e ‘\_’, além, claro, do nosso amigo ‘\’. Se você deseja usá-los no seu texto, será preciso recorrer a alguma sequência de controle que os coloque lá. A propósito, as sequências de controle necessárias para esses caracteres em particular são

‘\%’ para ‘%’    ‘\\$’ para ‘\$’    ‘\\_’ para ‘\_’    ‘\textbackslash’ para ‘\’

### 3.2 Um documento simples

Um texto preparado para o L<sup>A</sup>T<sub>E</sub>X em geral é precedido por um *preâmbulo*, em que geralmente são descritas características do texto (por exemplo, se ele é uma carta, um livro, um relatório; quem é o seu autor; se o documento será impresso frente e verso, ou se apenas uma página por folha.

O trecho abaixo tem três sequências de controle. Vejamos o que significam.

```
\documentclass{article}
\begin{document}

Olá mundo! % Colocar um conteúdo de verdade.

\end{document}
```

Primeiro definimos a *classe* do documento, com a sequência de controle `\documentclass`. Essa sequência requer um parâmetro, (qual a classe do documento, no caso `article`) que é posto entre chaves. Teremos mais a falar sobre parâmetros, ou *argumentos* daqui a pouco.

A classe `article`, define uma série de coisas, como o tamanho das margens e a formatação de muitos elementos do texto, p. ex., a formatação dos números das páginas. Outras classes comumente usadas incluem `letter`, para cartas, `beamer` para apresentações de slides, `report` para relatórios, `book` para livros, `a0poster` para pôsteres em papel A0, e ainda modelos para teses disponibilizados por universidades, muitos dos quais se pode obter gratuitamente na internet.

A seguir, demarca-se o início do documento propriamente dito. O par de sequências de controle `\begin` e `\end` delimita uma *região* (falaremos mais delas em breve). Aqui, a região é o próprio documento, seu conteúdo visível. Assim, `\begin{document}` delimita o início de uma região do tipo *document*, que é encerrada por `\end{document}`.

Finalmente, o conteúdo do documento: a frase “Olá mundo!”, seguida de um *comentário*. Se você é um programador, a noção de comentário (como aliás muitas outras que abordaremos aqui) deve ser-lhe bem familiar. Em nosso exemplo, o comentário é

*Colocar um conteúdo de verdade.*

Comentários iniciam-se por um caractere ‘%’, e vão até o fim da linha. Eles são ignorados pelo L<sup>A</sup>T<sub>E</sub>X: são anotações no texto que o autor pode fazer para lembrar-se de algo ou temporariamente remover um trecho do texto, por exemplo.

Um detalhe importante: todo o texto que faz parte do comentário é como se não existisse para o L<sup>A</sup>T<sub>E</sub>X quando ele processa o texto: tudo o que está entre o caractere ‘%’ e a primeira quebra de linha é ignorado, incluindo o caractere de porcentagem e a quebra de linha!

Um comentário serve para inibir o%	Um comentário serve para inibir o
início de um parágrafo novo, já q%	início de um parágrafo novo, já que engole a que-
ue engole a quebra de linha também.	bra de linha também.

### 3.3 Parâmetros

As sequências de controle (também chamadas aqui de *comandos*) encontradas, até agora foram sempre seguidas de algum texto entre chaves. Em  $\text{\LaTeX}$ , as chaves servem para agrupar coisas, para que sejam vistas como uma unidade só.

De modo geral, sequências de controle operam de acordo com os parâmetros, ou argumentos, que passamos para elas. Se uma sequência emprega um certo número de parâmetros (digamos, 2), ela considera que eles são os (dois) agrupamentos imediatamente depois dela. Mas atenção: o  $\text{\LaTeX}$  sempre considera *agrupamento* a menor unidade indivisível que encontra ao ler um texto! Letras que não estejam envolvidas em chaves são, cada uma, um elemento diferente, assim como sequências de controle o são. Por outro lado, um texto envolvido entre chaves conta como um único agrupamento, um único elemento.

Por exemplo, suponhamos que haja um comando `\importante` para destacar texto, que opere sobre um único parâmetro (o texto importante). O que cada uma das linhas a seguir destaca?

```
\importante Lembre-se de usar chaves!
\importante{fazer as compras}
\importante{Destacar textos {importantes}}
```

Respostas: (Você tentou os exercícios? Vá lá, mais uma chance!) Respectivamente: “L”; “fazer as compras”, e “Destacar textos importantes”.

Comandos nem sempre precisam de argumentos. Por exemplo, `\newpage` termina a página atual e continua o texto na página seguinte, e `\maketitle` mostra o título, autor e data do texto.

### 3.4 Regiões

Você já deve ter reparado que há uma certa “anatomia” no texto. Por exemplo, há imagens, citações, tabelas, poemas, listas, enumerações, e descrições, só para citar alguns. Todos são elementos de natureza diferente do texto, tanto visual como conceitualmente.

Essas regiões, também chamadas de *ambientes*, são trechos do texto que têm um papel diferente, e, assim, provavelmente demandam um tratamento diferente.

Já usamos regiões uma vez nesta apostila: o corpo do texto, o *document*, onde vivem seus elementos visíveis. Neste ponto, você já deve imaginar como fazer para delimitar um ambiente. Digamos que uma parte de nosso relatório seja pura magia. Para que isso seja de fato incorporado ao texto, basta fazer:

```
\begin{pura-magia}
Chirrin-chirrion!
\end{pura-magia}
```

### 3.5 Acentuação: para além do ASCII

Ao experimentar os exemplos dados até agora (se não fez, esta é uma boa oportunidade! Vá gerar documentos a partir dos exemplos, eu fico aqui esperando) você deve ter reparado que os caracteres acentuados não aparecem no documento final. Mas experimente o seguinte

```
\documentclass{article}
\begin{document}
Olá mundo! Você come\c cou a notar algo?
\end{document}
```

Não desespere. Acentuar é muito mais fácil do que isso. Tentemos outra coisa

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\begin{document}
Olá mundo! Você começou a notar algo?
\end{document}
```

Qual o resultado? E se você tentar o seguinte?

```
\documentclass{article}
\usepackage[T1]{fontenc}
\begin{document}
Olá mundo! Você começou a notar algo?
\end{document}
```

Uma das alternativas acima deve solucionar a questão dos acentos em seu computador, a depender de como estão armazenadas as letras no seu texto. Mais precisamente, cada uma das linhas novas, que começam por `\usepackage`, tenta dizer ao  $\LaTeX$  como interpretar a *codificação* do arquivo que ele irá processar.

O leitor atento poderá se perguntar: mas o texto que salvei é *puro*<sup>a</sup>, sem formatação alguma — como ele pode ser armazenado de mais de um modo? quem determina que codificação o arquivo tem? A resposta direta a essa pergunta é a seguinte: arquivos são armazenados como sequências de zeros e uns no computador (ao menos até este momento, em 2010). A *codificação* de um arquivo é o conjunto de regras que associa a determinadas sequências de zeros e uns a cada uma das letras de um texto.

---

<sup>a</sup>Usamos aqui *texto puro* como tradução da expressão em inglês *plain text*: texto sem formatação.

Apesar de os comandos para acentuação serem dispensáveis na maioria dos casos, há situações em que pode ser útil saber um truque ou outro. Principalmente quando o que se deseja é escrever algum nome estrangeiro em algum ponto particular do texto, e não se sabe como obter o caractere a partir do seu teclado.

O trecho a seguir é um excerto do  $\TeX$ book.

Erd\H os, B\=askara, Gabor Szeg\"o.

'\`o' (grave accent)  
'\'o' (acute accent)  
'\^o' (circumflex or \hat")  
'\"o' (umlaut or dieresis)  
'\~o' (tilde or \squiggle")  
'\=o' (macron or \bar")  
'\.'o' (dot accent)  
'\u o' (breve accent)  
'\v o' (há\v cek or \check")  
'\H o' (long Hungarian umlaut)  
'\t oo' (tie-after accent)  
'\c o' (cedilla)  
'\d o' (dot-under accent)  
'\b o' (bar-under accent)  
'\oe', '\OE' (French ligature OE)  
'\ae', '\AE' (Latin and  
Scandinavian ligature AE)  
'\aa, \AA' (Scandinavian A-with-circle)  
'\o', '\O' (Scandinavian O-with-slash)  
'\l', '\L' (Polish suppressed-L)  
'\ss' (German \es-zet" or sharp S)

Erdős, Baskara, Gabor Szegő.

'ò' (grave accent) 'ó' (acute accent) 'ô' (circumflex or "hat") 'ö' (umlaut or dieresis) 'õ' (tilde or "squiggle") 'ō' (macron or "bar") 'ó' (dot accent) 'ö' (breve accent) 'č' (háček or "check") 'ő' (long Hungarian umlaut) 'ô' (tie-after accent) 'ç' (cedilla) 'ø' (dot-under accent) 'ø' (bar-under accent) 'œ', 'Œ' (French ligature OE) 'æ', 'Æ' (Latin and Scandinavian ligature AE) 'å', 'Å' (Scandinavian A-with-circle) 'ø', 'Ø' (Scandinavian O-with-slash) 'ł', 'Ł' (Polish suppressed-L) 'ß' (German "es-zet" or sharp S)

Mas o que faz o comando '\usepackage'? Veremos a seguir.

### 3.6 Pacotes

Uma característica importantíssima do L<sup>A</sup>T<sub>E</sub>X é sua expansibilidade, que permite que ele se adapte às necessidades dos mais variados usuários. Assim como é possível estender as capacidades de um programa acrescentando-lhe 'plugins', 'add-ons', ou, em mais baixo-nível, bibliotecas, é possível dotar o L<sup>A</sup>T<sub>E</sub>X de mais comandos, pela inclusão de *pacotes*.<sup>7</sup>

Pacotes são documentos de texto (como os que você escreve ao seguir esta apostila). Certo, eles não são *exatamente* documentos de texto como os que você escreve agora: os pacotes possuem diversas definições de comandos, macros e ambientes, que agregam funcionalidade ao L<sup>A</sup>T<sub>E</sub>X. (Mas é texto, apenas.) Pacotes têm muitas vezes a extensão `sty`, embora você não precise se preocupar com esse detalhe (ao menos enquanto você não estiver escrevendo seus próprios pacotes, ou investigando as fascinantes entranhas do sistema).

Para aplicar um pacote, basta usar o comando `\usepackage`. Esse comando faz com que o L<sup>A</sup>T<sub>E</sub>X procure pelo arquivo do pacote e torne sua funcionalidade disponível para que você dela disponha como quiser. O argumento do comando é o nome do pacote. Pouco atrás usamos um comando para poder usar acentos em arquivos codificados em `utf8`.

```
\usepackage[utf8]{inputenc}
```

Este comando tem um *parâmetro opcional*, `utf8`. Parâmetros opcionais estão presentes em vários comandos. Um parâmetro opcional pode ser omitido; ele geralmente representa alguma configuração ou pequena alteração no modo de funcionamento do comando.

Assim, é comum que pacotes possam ser configurados por meio de parâmetros opcionais passados a eles.

<sup>7</sup>Há outras maneiras...

Da mesma maneira, classes de documento também podem ser configuradas por meio da passagem de parâmetros opcionais. Alguns exemplos: pode-se passar os parâmetros opcionais `11pt`, `twocolumn`, `twoside`, `draft` para a declaração da classe `article`. Assim, para um documento a ser impresso frente-e-verso, em duas colunas, podemos escrever

```
\documentclass[twocolumn,twoside]{article}
\begin{document}
...
\end{document}
```

É importante notar que separamos os parâmetros opcionais por vírgulas. Isso acontece para comandos como `documentclass` e `usepackage`, mas não é válido para outros comandos.

Há pacotes para as mais diversas coisas: acrescentar cor ao texto, usar capitulares (letras grandes, muitas vezes cheias de adornos, no início de parágrafos), para descrever palavras-cruzadas, jogos de xadrez, para desenhar, para fazer tabelas grandes, colocar trechos de texto em colunas, acrescentar marcas d'água, personalizar cabeçalhos e rodapés. Existem até mesmo “meta-pacotes.”<sup>8</sup>

### 3.7 Caracteres reservados

São dez os caracteres reservados pelo  $\LaTeX$  para funções especiais (ou seja, é preciso alguma ginástica para obtê-los). Eles são os seguintes.

`\` `_` `^` `~` `&` `#` `{` `}` `%` `$`

A barra marca o início de um comando; o “underscore” e o circunflexo são usados no modo matemático (seção 9); o “til” para gerar espaços *duros* (seção 3.10.3); o “e comercial” é usado em tabelas (seção 5.4.2); o “jogo da velha” é usado na definição de comandos (seção 6.1); as chaves agrupam texto; o caractere de porcentagem marca o início de comentários; e o cifrão delimita o modo matemático.

Esses caracteres podem ser usados em um documento prefixando-os por uma barra.

---

<code>\_</code> <code>\^{} </code> <code>\~{} </code> <code>\&amp; </code> <code>\# </code> <code>\{ </code> <code>\} </code> <code>\% </code> <code>\\$ </code>	<code>_</code> <code>^</code> <code>~</code> <code>&amp;</code> <code>#</code> <code>{</code> <code>}</code> <code>%</code> <code>\$</code>
--	---

---

A exceção é a barra, que pode ser obtida por meio do comando `\textbackslash`.

### 3.8 Palavras de controle e texto

Para construir um texto usamos aqui nada mais que palavras e comandos, simplesmente. Nesta seção veremos como eles se coordenam.

Durante o processamento de seu texto, a maior parte do tempo o  $\LaTeX$  apenas encontra letras comuns, que prepara para colocar em um parágrafo. Algumas vezes, no entanto, ele encontra uma barra — o que significa que uma sequência de controle foi encontrada. Se o caractere seguinte não for uma letra, trata-se de um *caractere de controle* (vide seção 3.1), e o  $\LaTeX$  continua processando o texto, levando em conta, claro, o significado do comando que

<sup>8</sup>Pacotes que auxiliam a escrita de outros pacotes. Esses pacotes geralmente são de um gênero mais técnico, parecendo às vezes “coisa de programador.”

encontrou. Já se após a barra há uma letra, o sistema se prepara para ler uma palavra de controle: continua a ler caracteres do texto até encontrar o primeiro caractere que não seja uma letra<sup>9</sup>. Se a palavra de controle é seguida de espaços em branco, *eles são ignorados*; e se é seguida de *uma* quebra de linha, esta é ignorada também. O que acontece com mais quebras de linha? Experimente! Exercício: como você faria para escrever `TeX`emplo?

Se os espaços em branco são ignorados, como faço para que uma palavra de controle como `LaTeX` seja seguida por um espaço (como foi aqui)? Os espaços são necessários após uma palavra de controle para definir seu fim — caso contrário, o `LaTeX` consideraria que `TeX`emplo é uma palavra de controle só. Mas qualquer coisa que permita ao sistema identificar que a palavra de controle terminou serve para o mesmo propósito. Assim, se você colocar um grupo vazio seguindo o comando (`\TeX{} emplo`), ou colocar um grupo envolvendo o comando (`{\TeX} emplo`), o espaço que segue o fim do grupo será preservado. Há ainda um outro modo de colocar um espaço logo depois de uma palavra de controle: basta usar o comando `\` , que é uma barra seguida de um espaço. Esse comando simplesmente produz um espaço em branco, e podemos escrever `\TeX\ emplo` para obter o `TeX`emplo.

### 3.9 Texto sem formatação

Por vezes o que queremos é que o texto digitado apareça exatamente como o escrevemos. Veremos a seguir que o `LaTeX` toma algumas decisões por conta própria na hora de compor o texto, e os importantes benefícios que esse comportamento traz consigo. Por hora, mencionemos um importante exemplo: nem todo espaço no arquivo-fonte corresponderá a um espaço na formatação final. Calma, as palavras não serão coladas. Mas experimente usar dois espaços entre um par de palavras. O que acontece<sup>10</sup>?

Em algumas situações, como por exemplo em listagens de programas, pode ser útil usar o `LaTeX` como se ele não fosse mais do que uma máquina de escrever digital. Queremos que o texto seja posto *verbatim*, isto é, exatamente como foi escrito. Para isso, podemos usar (sic) o ambiente `verbatim`.

<pre>\begin{verbatim} int main(int argc, char argv) {     int resposta = 42;     /* TODO: calcular a pergunta */     return 0; } \end{verbatim}</pre>	<pre>int main(int argc, char argv) {     int resposta = 42;     /* TODO: calcular a pergunta */     return 0; }</pre>
---	---

Há um outro método para “cancelar” a interpretação de caracteres, para trechos menores, destinados a viver dentro de uma frase comum. Por exemplo, as várias vezes em que me referi a comandos `LaTeX`, precisei fazer com que a interpretação do comando fosse abortada (caso contrário, teria obtido `LaTeX`). O comando que faz isso é o `\verb`, que possui uma sintaxe especial. O comando é seguido por um caractere qualquer (espaço vale!). Esse caractere servirá para delimitar o fim do argumento de `\verb`. A esse caractere se segue o texto a ser “verbatimizado,” que é todo o texto até a próxima ocorrência do delimitador. Exemplo: `\verb!\LaTeX!` resulta em `\LaTeX`, mas `\LaTeX` resulta `LaTeX`.

<sup>9</sup>Lembre-se: letras são os caracteres de `a a z` e de `A a Z`.

<sup>10</sup>Não há resposta aqui =).

Um último comentário. Tanto o comando quanto o ambiente `verbatim` possuem uma versão “estrelada”, que exhibe os espaços em branco `deste_jeito_aqui`. O ambiente é chamado `verbatim*` e o comando `\verb*`.

### 3.10 Alguma tipografia

Já dissemos que o  $\text{\LaTeX}$  tem um jeito particular de dispor o texto que escrevemos. Veremos agora que história é essa.

<p>As grandiloquência exibicionista são pouco persuasiva para aqueles honestamente curioso.</p> <p>Verdade <span style="float: right;">isso.</span> Para quem já tanto circunvaga o sentido, cheio de dedos no pântano dos significados, um pouco de tento com o que passa a ser floreio decorativo é no mínimo cortês.</p> <p>E tudo <math>\text{\LaTeX}</math> ado apropriadamente. <math>\text{\emph{Muito}}</math> apropriadamente. Usando alguns comandos <math>\text{\LaTeX}</math> que já foram vistos\dots.</p>	<p>As grandiloquência exibicionista são pouco persuasiva para aqueles honestamente curioso.</p> <p>Verdade isso. Para quem já tanto circunvaga o sentido, cheio de dedos no pântano dos significados, um pouco de tento com o que passa a ser floreio decorativo é no mínimo cortês.</p> <p>E tudo <math>\text{\LaTeX}</math>ado apropriadamente. <i>Muito</i> apropriadamente. Usando alguns comandos <math>\text{\LaTeX}</math> que já foram vistos....</p>
---	---

Algo que salta à vista de primeira é que as quebras de linha não são respeitadas. Também parece que os espaços a mais são desconsiderados... e a realidade não está mesmo longe disso: um espaço ou vários espaços são a mesma coisa para o  $\text{\LaTeX}$ . Uma (única) quebra também é equivalente a um espaço. Duas quebras de linha, por outro lado, fazem com que um novo parágrafo seja iniciado.

Notável também é o fato de que o primeiro parágrafo não tem recuo, enquanto que os demais o têm. Isto se deve ao fato de que para algumas culturas (em particular na tipografia de língua inglesa), não é costumeiro marcar a primeira linha de um parágrafo com recuo a menos que este seja precedido imediatamente por outro parágrafo. Afinal, esse recuo tem por objetivo facilitar a identificação visual do novo parágrafo, o que não é necessário por exemplo se o parágrafo é o primeiro de uma seção ou capítulo do texto.

Encontramos também os comandos  $\text{\LaTeX}$ , que escreve  $\text{\LaTeX}$ , e  $\text{\emph{}}$ , que *ênfatiza* o texto que lhe é passado como parâmetro. Note que o que o comando faz é enfatizar: o jeito como ele faz isso não é a nossa preocupação nesse momento.

*O que importa aqui é que o trecho tem que ser destacado.*

E isso é diferente de dizer que o texto deve ser posto em negrito, ser sublinhado, ser escrito em fúcsia,  $\text{\textit{no 9b murgls otieĵ ordnartse9}}$ . Afinal, o paradigma aqui é que a aparência do texto refletirá a função, o papel semântico desempenhado por cada um de seus elementos. Assim, descreve-se num primeiro momento o que cada um *significa*, deixando-se para outra etapa (quando pertinente) o ajuste do modo pelo qual essa função é realçada visualmente.

$\text{\LaTeX}$  lida com uma granularidade maior de conceitos do que comumente nos é dado controlar em ambientes usuais de edição de texto; conceitos que, a princípio, podem surpreender os não iniciados ao universo dos cuidados tipográficos. A partir de agora, e à medida que adquire experiência com um sistema tipográfico de alta qualidade como o  $\text{\LaTeX}$ , você notará

uma série de mudanças na sua percepção de um texto. Seu vocabulário vai crescer, seus olhos e atenção serão exercitados em novas direções, e muito provavelmente você se surpreenderá com a influência que “detalhes” têm no ritmo e facilidade de leitura de um texto. Mãos à obra!

### 3.10.1 Hífens e hifenação

Muito embora haja apenas um tipo de hífen em seu teclado, existem muito mais hífen na tipografia. Há aquele usado em palavras compostas, como “guarda-chuva” ou ainda “resguardar-se”, e que também servem para marcar a quebra de uma palavra no fim de uma linha (sua *hifenação*); há o traço usado para indicar um intervalo de números, por exemplo 12–14; há o travessão — o mais longo entre os hífen; e há o sinal de menos, usado em equações, como em  $20 - 3 = 17$ . É fácil produzir cada um desses símbolos em  $\LaTeX$ .

- guarda-chuva, resguardar-se
- exercícios das páginas 12--14
- no dia de hoje --- véspera de amanhã
- diga-me também que  $\(2-2=5\)$ , Winston

O último dos exemplos acima usa o chamado *modo matemático*, assunto da seção 9.

Mas há ainda o que falar sobre hifenação. Na maior parte dos casos, o  $\LaTeX$  sabe hifenas corretamente as palavras de diversos idiomas (o português entre eles). Para isso basta usar o pacote `babel`, passando como parâmetro `brazil`. Algumas vezes, porém, usamos termos que possuem uma hifenação pouco comum, ou usamos palavras que o  $\LaTeX$  não consegue hifenas a contento. Quando isso ocorre, podemos dizer explicitamente em que pontos uma palavra pode ser hifenada. Há dois modos de fazê-lo: pode-se, no preâmbulo, adicionar um comando `\hyphenation`, que leva como parâmetro uma lista de hifenações, separadas por espaços, como abaixo. Assim terá ensinado ao  $\LaTeX$  como hifenas o termo, e ele aplicará a regra aprendida sempre que achar conveniente. Note que não se podem usar comandos ou caracteres especiais no argumento do comando.

```
\hyphenation{FNAC A-bra-cur-six}
```

No exemplo acima, FNAC, fnac e Fnac não serão jamais hifenadas, ao passo que Abracursix e abracursix o serão, segundo os hífen especificados.

Outro modo é explicar onde uma determinada ocorrência de uma palavra pode ser hifenada, quando ela ocorre no texto. Nesse caso, a sugestão de hifenação vale naquele ponto somente. O  $\LaTeX$  não se lembrará dela se a palavra for usada novamente.

---

É algo assim, como direi?  
`su\per\ca\li\frag\i\lis%`  
`\-tic\ex\pi\-a\li\do\cious`

---

É algo assim, como direi? supercalifragilisticexpialidocious

---



### 3.10.2 Apurando os sentidos: ligaduras, kerning

As letras por vezes requerem pequenas modificações no espaçamento entre si, ou mesmo em sua forma, a depender dos símbolos que estão próximos de si. Por exemplo, alguns pares de letras são aproximados, enquanto outras vezes, partes de letras se fundem. Observe os exemplos abaixo.

fi      fi      Ta      Ta  
fl                      fl

Ligaduras (do inglês, *ligatures*), ocorrem quando um agrupamento de letras é substituído por algum outro símbolo, quer para melhorar sua legibilidade, quer para tornar o texto mais belo.

Já o *kerning* é um aumento ou diminuição do espaço entre letras, que varia de acordo com o entorno de cada caractere.

Uma Torta Para Jaiminho  
Uma Torta Para Jaiminho  
AVARO      AVARO      PARA      PARA

### 3.10.3 Sobre espaçamento horizontal

Nem todos os espaços são iguais. Não só variam em tamanho, mas possuem comportamentos distintos. Falaremos a seguir dos *espaços duros* e de espaços um pouco mais largos, embora isso esteja longe de esgotar o assunto.<sup>11</sup> Falaremos dos espaços mais comuns no texto, como os que separam palavras. Algumas vezes (principalmente quando abordarmos a escrita de expressões matemáticas), outros tipos de espaçamento serão necessários.

Todo parágrafo justificado, isto é, que tem as margens direita e esquerda alinhadas verticalmente, exige que o espaçamento entre palavras seja “elástico”, aumentando ou diminuindo conforme a necessidade. O  $\text{\LaTeX}$  possui um mecanismo interno elaborado para o gerenciamento desses espaços (que não descreveremos aqui). Ainda assim, é importante saber que alguns espaços são mais elásticos do que outros, e que os espaços podem possuir limites de compressão e expansão.

Por exemplo, o espaço que segue o ponto final (ou a interrogação, ou a exclamação) em uma frase é mais elástico que o espaço que une as demais palavras. Mas como o fim de uma frase é identificado?

Por padrão, o  $\text{\LaTeX}$  assume que um ponto final — ou outra pontuação como ‘?’ ou ‘!’, ou ‘...’ (reticências são produzidas usando o comando `\ldots`) — marca o fim de uma frase sempre que, e somente quando, for precedida por uma letra minúscula. Na maior parte dos casos esse comportamento é exatamente o que precisamos, mas nem sempre.

---

<sup>11</sup>Espaço preenchido e espaço em branco são igualmente importantes na composição visual.

Títulos como Dr. não terminam uma frase, na maior parte dos casos. Isso é resolvido usando ‘Dr.\\_’.<sup>12</sup> Por outro lado, existem casos em que uma letra maiúscula seguida de pontuação *termina* uma frase: URSS. Para indicar que o ponto final após uma letra maiúscula termina a frase, existe o comando ‘\@’, correto?

Entendi, OK\@. A frase terminou no último ponto. E nesse também. E nesses.

Falaremos agora dos espaços *duros*. Existem palavras que estão naturalmente ligadas, e não toleram quebras de linha entre si. Isto acontece em expressões como “seção 3.10.3”, “Dr. House” (viu?). Frequentemente é preciso prestar atenção a expressões como “Teorema de Kuratowski”, “Associação Contra os Maus-tratos a Espécies”, em que nem todos os espaços são duros, mas alguns são. Para produzir um espaço duro em L<sup>A</sup>T<sub>E</sub>X, usa-se o til ‘~’. Por exemplo, “5~cm”. Com um pouco de prática se torna natural a introdução desses espaços quando apropriado.

## 4 Estruturando o texto

Textos, assim como animais, possuem uma anatomia. Essa anatomia é o que permite ao leitor se localizar em sua leitura, identificar algo que procura. A estrutura do texto, além disso, carrega uma mensagem em si, ao menos em potencial, ao refletir o encadeamento do texto.

A depender da classe do documento, há uma certa variedade de tipos de segmentações à nossa disposição para organizar o texto.<sup>13</sup> Artigos podem ser particionados em seções, subseções, subsubseções, apêndices. Livros possuem, adicionalmente ao que está disponível em artigos, capítulos (contendo um certo número de seções). Relatórios possuem adicionalmente (a livros) *partes* (que contém capítulos). E por aí vai.

Você pode mesmo criar seu próprio nível hierárquico, como parágrafos.

Neste capítulo, abordaremos, a título de exemplo, seccionamento (segmentação) de um texto em artigos (documentos da classe *article*). O comportamento apresentado em livros, relatórios etcétera é análogo, e em caso de dúvida basta recorrer aos manuais da respectiva classe (que, por padrão, vêm juntamente com o pacote quando a sua distribuição L<sup>A</sup>T<sub>E</sub>X é instalada).

### 4.1 Títulos, autor e data de documentos

Em muitas classes de documentos, estão disponíveis os comandos para definir o título, o(s) autor(es) e a data do documento. Cada classe exhibe essa informação de um modo, mas em boa parte delas você define o título com um comando `\title{Minhas Férias}`, o autor usando o comando `\author{YoMoiIchEu}`. A data é composta automaticamente com a data em que o documento for processado (no idioma do documento). Você pode escolher (fixar) a data usando o comando `\date{Muito, muito tempo atrás}`.

Depois de especificados o título e o autor (mais de um autor pode ser declarado, separando-se seus nomes por `\and`), você escolhe o ponto do texto no qual quer que apareçam, e usa o comando `\maketitle`. Voilà!

---

<sup>12</sup>Como vimos na seção 3.8.

<sup>13</sup>E, como tudo o mais quando se trata da família T<sub>E</sub>X, esse conjunto pode ser estendido e modificado como melhor nos aprouver.

## 4.2 Marcando a anatomia

O exemplo a seguir ilustra o uso de seções, subseções, subsubseções, seções não numeradas e apêndices.

```
\section*{Preface}
% texto...
\section{Introdução}
% texto...
\subsection{Contexto histórico}
% texto...
\subsection{Proposta investigativa}
% texto...
\section{Argumentação fantástica}
% texto...
\section{Conclusão bombástica}
% texto...
\appendix
% A partir daqui os capítulos são ‘‘numerados’’
% com letras em vez de números
\section{Prova incontestável}
```

Dizemos que uma seção inicia a partir do comando `\section`. O argumento que este comando leva é o título da seção. O mesmo acontece para sub-seções, e as demais divisões do texto.

Seções são numeradas por padrão. Para obter uma seção, sub-seção, etc. não numerada, use o respectivo comando em sua versão com asterisco, por exemplo `\section*{Prefácio}`.

A classe de documento e os pacotes que você usa definem quais os comandos de secionamento disponíveis. Livros, por exemplo, têm `\chapter`, relatórios têm `\part`, e por aí vai.

## 4.3 Sumários

Falemos agora do acompanhamento natural de um texto secionado: sumários (ou índices). Fazer um sumário, com o  $\text{\LaTeX}$  é muito simples. Marque os títulos das partes usando os comandos de secionamento que acabamos de ver, e, no ponto do texto em que deseja acrescentar o índice, coloque o comando `\tableofcontents`.

Uma vez marcadas as seções do texto e solicitado o índice, o  $\text{\LaTeX}$  anota (em um arquivo auxiliar) as páginas em que começam as seções do texto, à medida que o processa. Essas informações são usadas para escrever o sumário. A depender da parte do texto na qual sumário foi posto, pode ser necessário processar o texto duas vezes<sup>14</sup> (na primeira as páginas em que ocorrem as seções são anotadas, e na segunda as entradas no sumário são atualizadas com os valores corretos). Além das informações escritas no arquivo auxiliar (que tem a extensão `aux`), o comando `\tableofcontents` faz ser gerado um outro arquivo, com a extensão `toc` (*table of contents*), que contém o sumário em si.

É possível — e igualmente fácil — gerar listas de figuras, tabelas ou quaisquer outros elementos usando  $\text{\LaTeX}$ .

---

<sup>14</sup>Para sermos estritamente precisos, é possível construir documentos anômalos que “ludibriam” o índice e requerem mais processamento, mas não se preocupe: se seu texto tiver essa propriedade, certamente você a terá causado conscientemente.

Os comandos de seccionamento possuem em geral um parâmetro opcional, que é uma versão “mais compacta” do título, para ser usada no sumário (ou, por vezes, no cabeçalho ou rodapé de páginas).

```
\section[Prova Documental]{%
  Documento provando a corretude do argumento %
  que concebi em uma longa insônia alcoolizada}
```

## 4.4 Referenciando elementos do texto

Assim como sumários são elementos importantes para a orientação do leitor-explorador, existem outros tipos de referências que ocorrem com frequência. Outro modo de remeter o leitor a um trecho, página — em geral, a um *elemento* qualquer — do texto é usando *referências*, que são o assunto desta seção.

Há vários tipos de referências, e várias maneiras de se referir a alguma coisa. Podemos fazer referência a uma *figura* ou a um *capítulo*; assim como podemos identificá-los por um número próprio, ou pelo número da página em que se iniciam.<sup>15</sup> Em sua essência, porém, existem apenas dois componentes imprescindíveis em uma referência: um *indicador* e um *indicado*.

Para referenciar algo em  $\text{\LaTeX}$ , usamos *rótulos*. Rótulos são nomes que damos a algum elemento do texto. Para criar um rótulo, use o comando `\label{nome do rotulo}`, e para referenciá-lo use o comando `\ref{nome do rotulo}`.

Quando usados em um ponto do texto, o label fica automaticamente associado à página, seção (subseção e etcétera) a que pertence aquele ponto no texto. Em enumerações, associa-se ainda ao item correspondente, e assim vale para figuras, tabelas e ambientes em geral.

Note que no exemplo o nome do rótulo não tem acentos. Quando criar seus rótulos, use apenas caracteres simples: mais especificamente, caracteres ASCII<sup>16</sup>.

## 5 Alguns elementos do texto

Aqui veremos exemplos de três ambientes que delimitam construções comuns em textos: listas (de vários tipos), trechos com lateral rasgada (não-justificados), e elementos “flutuantes”.

### 5.1 Listas & Cia.

Listas encarnam uma função que é misto de destaque e segregação. A um só tempo, o conteúdo de uma lista é apartado da corrente do texto, enquanto que cada um de seus itens tem sua unidade e individualidade reforçadas, como você pode ver experimentando com os exemplos abaixo. Eles mostram, respectivamente, listas não-numeradas e numeradas e descrições.

---

<sup>15</sup>Outros tipos de referência incluem: referência a notas de rodapé e a elementos “externos”, como itens de bibliografia, glossário e índices remissivos, que são explorados nas seções 11 e 12.

<sup>16</sup>Entre os caracteres ASCII estão as letras de ‘a’ a ‘z’ (maiúsculas e minúsculas), os dígitos, o espaço em branco, assim como os caracteres “@#%&\*‘ ’ ! ( ) - \_ = + [ ] { } ^ ~ , . ; \ | < > ?”.

```

\begin{itemize}
\item cebola,
\item açafrão, e
\item alho.
\end{itemize}

```

- cebola,
- açafrão, e
- alho.

```

\begin{enumerate}
\item cebola,
\item açafrão, e
\item alho.
\end{enumerate}

```

1. cebola,
2. açafrão, e
3. alho.

```

\begin{description}
\item[cebola] Muito empregada p/ temperar.
\item[açafrão] Também.
\item[alho] Idem. Hmm!
\end{description}

```

**cebola** Muito empregada p/ temperar.  
**açafrão** Também.  
**alho** Idem. Hmm!

## 5.2 Citando textualmente

Existem dois ambientes comumente usados para incluir excertos de outros textos no seu. Os ambientes `quote` para trechos curtos, e `quotation` para trechos mais longos.

## 5.3 Alinhamento

Boa parte dos textos possui alinhamento justificado, i.e., possui ambas as margens retas e paralelas. Nem sempre isso é desejado. Existem muitas maneiras de definir o alinhamento do texto: falamos de duas delas aqui.

### 5.3.1 Texto não-justificado

No ambiente `flushleft`, o texto é “empurrado” para a esquerda. Os espaços não são nem esticados nem comprimidos. O efeito resultante são linhas de comprimento variável, por vezes uma opção interessante de diagramação.

Simetricamente, `flushright` tem o comportamento que seria de se esperar, fazendo parágrafos ficarem com a esquerda *rasgada*, isto é, sejam empurrados para a direita.

Mais não é preciso dizer: `center`.

## 5.4 Elementos flutuantes

Tipógrafos atentam para uma série de características na disposição do texto que frequentemente passam despercebidas ao nosso consciente. Uma delas é o equilíbrio entre o texto que se espalha pelas páginas e os demais elementos, como figuras e tabelas, que pontuam a paisagem aqui e ali.

O  $\LaTeX$  toma várias precauções na disposição desses elementos, ditos *flutuantes* (porque sua posição não é fixa no texto como a de uma palavra em uma frase). É como se os elementos fossem troncos de árvore à deriva sobre uma correnteza de palavras. Figuras e tabelas são somente alguns exemplos de elementos flutuantes. Eles são ambientes (respectivamente `figure`

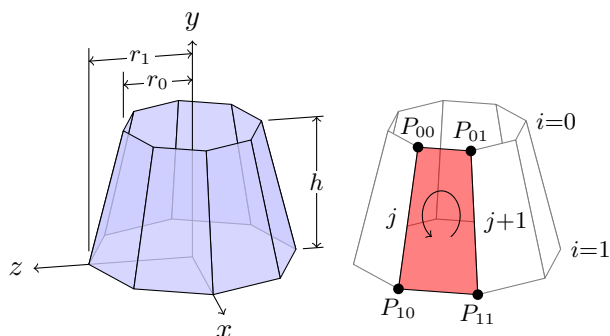


Figura 1: Uma figura gerada com o pacote Tikz.

e table) que encapsulam o conteúdo que irá flutuar.<sup>17</sup> Trocando em miúdos, ele (conteúdo) nem sempre será posto no exato ponto do texto (fonte) em que aparece.

É frequente que elementos flutuantes apareçam ainda em um índice próprio (listas de figuras ou tabelas, por exemplo), e que possuam uma legenda (para que, mesmo extraídos de sua vizinhança textual, remetam ao assunto de que tratam).

#### 5.4.1 Figuras

Figuras são uma ferramenta poderosa na composição de textos, quando usadas com parcimônia. É possível colocar imagens no documento dizendo ao  $\LaTeX$  sua localização (ou apenas seu nome, se estiverem na mesma pasta que o documento). Também é possível desenhar usando o próprio  $\LaTeX$ , por exemplo, com o pacote Tikz.

Para colocar figuras em um documento  $\LaTeX$ , basta usar o comando

$$\backslash includegraphics\{nome-do-arquivo\}$$

em que a extensão do tipo de arquivo não precisa ser incluída. Mas atenção: nem toda extensão de imagem é conhecida pelo  $\LaTeX$  nativamente, embora baste usar um pacote para superar o problema, na maior parte dos casos. Acrescente  $\backslash usepackage\{graphicx\}$  no preâmbulo de seu documento e você poderá incluir imagens png, jpg e pdf, para citar algumas.

O trecho a seguir inclui a figura 1 em um ambiente flutuador.

```
\begin{figure}
  \begin{center}
    \includegraphics{exemplos-externos/foto-noel}
    \caption{Foto de Papai Noel.}\label{fig:prova-noel}
  \end{center}
\end{figure}
```

<sup>17</sup>Falamos de *o que* acontece com um elemento flutuante: ele deriva. Como ele o faz, isto é, como o  $\LaTeX$  tenta determinar um posicionamento adequado, é um assunto um pouco mais delicado, que exige uma bagagem técnica (tanto tipográfica quanto  $\TeX$ nica mesmo) que extrapola nosso escopo. O leitor interessado não encontrará dificuldade para acessar uma miríade de discussões sobre *algoritmos de posicionamento de floats do  $\LaTeX$*  na internet, ou em livros: (respectivamente) no *TUGboat* [flo93], e *The  $\LaTeX$  Companion* [Mit05] (por exemplo).

Note que o comando `\caption` é usado *dentro* do ambiente `figure` para acrescentar uma *legenda*. Repare ainda que o `label` que segue a legenda permite fazer referência (seção 4.4) ‘a figura. Assim, por exemplo “a figura 2 foi deslocada pelo L<sup>A</sup>T<sub>E</sub>X” é escrito:

```
a figura~\ref{fig:layers} foi deslocada pelo \LaTeX
```

em que “`fig:layers`” é o rótulo da figura.

## 5.4.2 Tabelas

Tabelas são um dos maiores desafios de diagramação enfrentados corriqueiramente por tipógrafos. Cada tabela é única, e uma leve alteração de espaçamento tem o poder de alterar bastante o destaque de uma e outra informação.

Começemos pelo ambiente `tabular`, que é o que permite escrever as tabelas. Ele é um ambiente que recebe um parâmetro, indicando algumas propriedades das colunas da tabela (vide exemplo).

O argumento do ambiente `tabular` é uma sequência de letras, entre as quais estão ‘`c`’, ‘`l`’ e ‘`r`’. Elas indicam o alinhamento do texto nas colunas da tabela. As letras, da esquerda para a direita, referem-se, respectivamente às colunas, da esquerda para a direita. A seguir, cada linha da tabela contém o texto de uma “célula” da tabela, separada da seguinte por um ‘`&`’. A célula mais à direita deve ser sucedida por `\\`, que indica o fim da linha da tabela.

Existem outras letras que podem aparecer como parte do parâmetro de `tabular`. Uma delas é a barra vertical ‘`|`’, que traça uma barra entre colunas.

Há ainda comandos que permitem traçar linhas que separam apenas algumas das células, que fixam a largura de uma célula (para que seja possível escrever parágrafos de mais de uma linha na tabela).

Não chegamos a mencionar todas as opções à disposição para a composição de tabelas. É possível estender o conjunto de comandos disponíveis, incluindo pacotes no preâmbulo do documento. A tabela 1 foi composta usando o pacote `booktabs`. Outro pacote comumente usado é o `longtable`, quando estamos tratando de tabelas que se espalham por mais de uma página.

<pre>\begin{center} \begin{tabular}{clcr l} a &amp; b &amp; c &amp; d &amp; e \\ fg &amp; hi &amp; jk &amp; lm &amp; no \\ \hline p &amp; q &amp; r &amp; s &amp; t \end{tabular} \end{center}</pre>	<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">a</td> <td style="padding: 5px;">b</td> <td style="padding: 5px;">c</td> <td style="padding: 5px;">d</td> <td style="border-right: 1px solid black; padding: 5px;">e</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">fg</td> <td style="padding: 5px;">hi</td> <td style="padding: 5px;">jk</td> <td style="padding: 5px;">lm</td> <td style="border-right: 1px solid black; padding: 5px;">no</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">p</td> <td style="padding: 5px;">q</td> <td style="padding: 5px;">r</td> <td style="padding: 5px;">s</td> <td style="border-right: 1px solid black; padding: 5px;">t</td> </tr> </table>	a	b	c	d	e	fg	hi	jk	lm	no	p	q	r	s	t
a	b	c	d	e												
fg	hi	jk	lm	no												
p	q	r	s	t												

## 6 Expandindo o L<sup>A</sup>T<sub>E</sub>X

Há uma grande quantidade de comandos disponíveis ao usuário de L<sup>A</sup>T<sub>E</sub>X (e ainda mais são criados em pacotes novos continuamente). Embora a maior parte das coisas que se pode querer fazer em L<sup>A</sup>T<sub>E</sub>X já exista na forma de algum comando, não raro podemos nos valer, com proveito, do poder de *estender o L<sup>A</sup>T<sub>E</sub>X* — como elástico que é.

Há uma série de razões por que usar comandos é uma boa. Falaremos delas... ao fim desta seção.

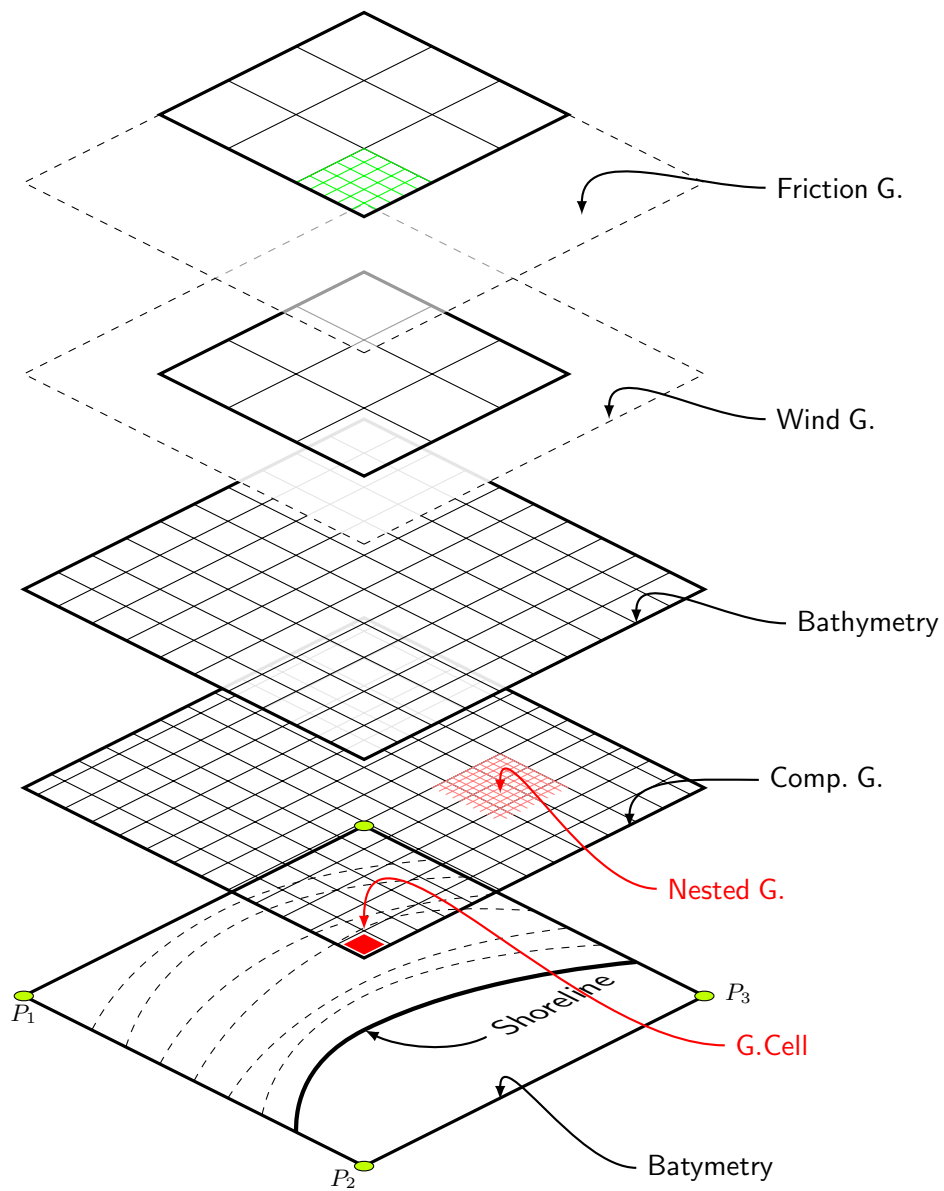


Figura 2: Outra figura gerada com o pacote Tikz.

Nome		
Nome	Sobrenome	nota
Pablo	Guerra	7.5
César	Bento	5
Eliás	Ribeiro	12

Tabela 1: Nota não reflete o aprendizado.



## 6.1 Criando comandos

O mecanismo primário de programação do T<sub>E</sub>X (e logo o L<sup>A</sup>T<sub>E</sub>X) é a definição de macros. Se isso soa grego, então ignore. Criar, ou *definir*, comandos é instruir o sistema sobre o que fazer quando ele encontra determinada sequência de controle.

Definimos comandos usando o comando `\newcommand`. O tipo mais simples de comando que há é a mera substituição. É como se definíssemos um apelido para algo que escrevemos com frequência.

Por exemplo, se estou a compor um estudo sobre a genealogia da família de feiticeiros do castelo Rá-tim-bum, faz sentido definir os comandos `\stradv`, e `\nino`

```
\newcommand{\stradv}{Stradivarius Victorius}
\newcommand{\nino}{Antonino}
```

que define `\stradv` como sinônimo para “Stradivarius Victorius”, e `\nino` analogamente. Então para escrever “Antonino Stradivarius Victorius II é o mais novo Stradivarius Victorius”, basta escrever

```
\nino\ \stradv\ II é o mais novo \stradv.
```

Esse tipo de comando é particularmente útil (para definir siglas grandes, por exemplo, ou nomes comuns). Mas por vezes não nos basta. Queremos mais. Queremos que o comando seja parametrizável, ou seja, que parte dele seja alterável. Fazemos isso dizendo ao L<sup>A</sup>T<sub>E</sub>X que o comando leva parâmetros, passando para `\newcommand` um parâmetro opcional: o número de argumentos (parâmetros) que o comando usa. Por exemplo,

```
\newcommand{\estrangeiro}[1]{‘#1’}
```

Este comando coloca aspas em torno de termos que sejam marcados como estrangeiros. Note o uso do caractere reservado ‘#’. Seguido de um número *n* entre 1 e 9, ele referencia o *n*-ésimo argumento que foi passado para o comando. Além disso, observe que o número de parâmetros é posto entre colchetes, *entre* o nome da macro<sup>18</sup> e a sua definição.

Mais um último exemplo. O que o comando `\formal` abaixo faz? (*Dica: é possível passar argumentos vazios para um comando.*)

```
\newcommand{\formal}[2]{Sr#1 #2}
```

## 6.2 Modificando comandos

Só é possível definir um comando uma vez. Se após isso queremos mudar seu significado, o que fazemos é *redefiní-lo*.

```
\renewcommand{\emph}{PRESTA ATENÇÃO AQUI!}
```

A sintaxe é a mesma da criação usual de comandos. Ao redefinir um comando, emph não é preciso continuar usando o mesmo número de parâmetros que originalmente ele usava. É como definir pela primeira vez um comando, só que você acrescenta “re” antes de `newcommand`.<sup>19</sup>

---

<sup>18</sup>Macro e comando são sinônimos aqui.

<sup>19</sup>A vantagem de ocorrer um erro se você tenta definir um comando que já existe é que desse modo o L<sup>A</sup>T<sub>E</sub>X garante que, se você está redefinindo um comando, está fazendo-o consciente de que está sobrescrevendo um comando existente. Isso é importante quando você está manipulando macros que serão usadas por outras pessoas, ou por você mesmo daqui a algum tempo.

### 6.3 Criando e modificando ambientes

Você cria um ambiente usando `\newenvironment`. Um ambiente é um comando com dois argumentos: o texto que você quer que seja inserido *antes* do início da região, e o que quer que seja inserido *depois*. É possível (e simples) definir ambientes que recebam parâmetros. Deixamos por conta do leitor descobrir como fazê-lo. (E já que estamos a distribuir tarefas, adivinhe como se faz para alterar o comportamento de um ambiente que já esteja definido.)

```
\newenvironment{com fofoca}{Você não acredita o que...}{...e essa agora!}
\begin{com fofoca}
Casaram!
\end{com fofoca}
```

### 6.4 Não sou mandão. Por que ficar criando comandos?

Terminamos esta seção com alguma discussão sobre o que vimos aqui. Definir comandos é uma parte importante do uso do  $\LaTeX$ . É com eles que você conseguirá marcar a estrutura e o significado de cada um dos elementos do seu texto. Comandos não precisam fazer nada (por um bom tempo) para que sejam úteis. Por exemplo

```
\newcommand{\palavraChave}[1]{#1}
```

O importante aqui, a princípio, é que o *significado* seja explicitado. Futuramente, à medida que você ganhar fluência em  $\LaTeX$ , verá que um texto semanticamente marcado transforma-se facilmente em uma obra de encher os olhos. Por exemplo, se pudermos confiar que todas as palavras-chave do texto estão marcadas com o comando `\palavraChave`, e toda palavra estrangeira com `\estrangeira`, será fácil colocar em um parágrafo todas as palavras-chave do texto automaticamente,<sup>20</sup> ou fazer um glossário dos termos em outro idioma. A anotação do significado de um elemento é um ponto de referência ao qual podemos acrescentar formatação e ações.

Comandos estáticos (substituições sem parâmetros) agilizam a digitação, e são ferramentas importantes para garantir a uniformidade de nomenclatura no texto. Conceitos que ainda estamos concebendo ou batizando, termos de cuja tradução não estamos certos, siglas e elementos que desejamos referenciar em glossários ou índices, todos esses são fortes candidatos a serem parcial ou totalmente encapsulados por um comando ou ambiente personalizado. Esse é um dos modos pelos quais o  $\LaTeX$  apoia o paradigma wsiwym.<sup>21</sup>

## 7 Projetos com vários arquivos

Nem todas as pessoas já tiveram a experiência de trabalhar em projetos em que vários arquivos de texto são necessários — donde o título desta seção pode soar estranho. Aqui discutiremos como (e por que) separar um documento em arquivos diferentes, que geram ainda um único arquivo `pdf` ou `dvi`.

---

<sup>20</sup>Isto é, de modo que novas anotações sejam agregadas a ele toda vez que o texto for processado, sem necessidade de interferência manual no processo

<sup>21</sup>Além disso, o ponto do texto em que ocorre a definição do comando “encapsulador” é o lugar mais seguro (de fato, é o lugar) para se colocar marcações visuais, como **negrito**, **sem serifa** ou **VERSALETE**, já que assim as mesmas transformações visuais são assim aplicadas consistentemente a todos os elementos com igual marcação semântica.

Há diversas situações em que é vantajoso ter um texto em vários pedaços. Uma bastante comum é o reuso. Dependendo do tipo de documento que você costuma escrever, determinados conjuntos de pacotes serão imprescindíveis, e você se verá acrescentando sempre os mesmos e definindo os mesmos comandos de novo e de novo e de novo... Mantendo um arquivo com o seu preâmbulo, você só precisa dizer ao  $\text{\LaTeX}$  (uma vez) onde encontrá-lo.

Arquivos menores são mais fáceis e rápidos de transmitir, imprimir, e de editar (é rápido encontrar o lugar no texto que se quer modificar). Ganha-se ainda em organização: em trabalhos de médio e grande porte, não se pode menosprezar o benefício de ter arquivos relacionados agrupados em uma mesma pasta. Essa vantagem é crucial se há mais de uma pessoa participando do projeto.

Uma possibilidade que a quebra em arquivos traz é processar apenas parte do documento por vez: somente o capítulo que se está editando, por exemplo. (A “compilação” de um projeto complexo pode levar alguns minutos — e podem ser necessárias várias iterações seguidas durante revisões e reestruturações.)

Alguma separação, é inevitável. Os pacotes, e classes de documento, por exemplo, são arquivos de texto que são incluídos no seu texto dissimuladamente. Listas de figuras e o sumário são outros exemplos.<sup>22</sup> Muitas vezes você irá acrescentar imagens, que, sendo ou não arquivos de texto, são externos ao documento.

## 7.1 $\backslash\text{input}$

O modo mais simples e “puro” de acrescentar um arquivo, digamos, `agradecimentos.tex` ao texto é usar o comando `\input{agradecimentos}`. Quando o texto é processado, esse comando tem o efeito de fazer com que o conteúdo do arquivo seja enxertado no texto, na posição exata em que ele ocorre: *para o  $\text{\LaTeX}$ , é como se o conteúdo sempre houvesse estado ali.*

Se chamado via linha de comando, o  $\text{\LaTeX}$  procura pelo arquivo no diretório (pasta) a partir do qual foi invocado. Ambientes mais elaborados para a edição de documentos têm provavelmente alguma opção de configuração do diretório de “referência”. Num projeto em que todos os arquivos estão na mesma pasta, isso é indiferente.<sup>23</sup> O diretório de referência passa a ser importante quando o projeto usa arquivos que estão em pastas diferentes. Isso porque o argumento do comando `\input` é mais do que o nome do arquivo. É o *caminho* até o arquivo.

Digamos que arquivo “principal” do texto (i.e., aquele que o  $\text{\LaTeX}$  irá processar), chama-se `carta-a-dulcineia.tex`. Ele será uma pequena narrativa das aventuras e desventuras que se esteve a enfrentar em honra de sua amada. Cada trecho dessa narração está em um arquivo, e digamos que já estão escritos os arquivos `o-gigante.tex` e `terrivel-feitico.tex`, guardados na pasta `capitulos`; há também um prólogo etilicamente enamorado. O projeto como um todo está numa pasta chamada `carta-a-dulcineia`, que está organizado conforme mostra a tabela 2.

Para que todos esses arquivos apareçam no documento final, eles precisam ser incluídos na `carta-a-dulcineia.tex`, que poderia ser escrito como segue.

---

<sup>22</sup>Estes são arquivos auxiliares criados pelo sistema durante o processamento de seu texto, e que são incluídos no documento quando ao processá-lo o  $\text{\LaTeX}$  percebe sua existência.

<sup>23</sup>Claro, há casos em que o diretório de referência *faz* diferença. Se esse é o seu caso, sugiro que procure alguém que use o mesmo ambiente que você, ou mesmo peça ajuda na internet. É provável que a solução do problema seja bem simples.

---

carta-a-dulcineia:	carta-a-dulcineia.tex, capitulos (diretório) prologo.tex
capitulos:	o-gigante.tex, terrivel-feitico.tex

---

Tabela 2: Arquivos do projeto *Carta a Dulcinéia*

```
\documentclass{letter}
\begin{document}
\input{prologo}
\input{capitulos/o-gigante}
\input{capitulos/terrivel-feitico}
\end{document}
```

## 7.2 `\include` e `\includeonly`

Outro modo de incluir arquivos é com o comando `\include`. Ele se comporta de maneira quase idêntica ao `\input`, só que cada arquivo enxertado começa em uma nova página. Outra diferença é que você pode usar o comando `\includeonly` no preâmbulo para dizer exatamente quais dos arquivos incluídos (por um `\include`) devem ser processados e aparecer no arquivo final. Considerando o exemplo do projeto de carta para a doce Dulcinéia, pode-se processar apenas o capítulo sobre o gigante e o prólogo enquanto se está trabalhando neles, bastando formatar o arquivo como mostrado abaixo.

```
\documentclass{letter}
\includeonly{o-gigante,prologo} % <-- aqui!
\begin{document}
\include{prologo}
\include{capitulos/o-gigante}
\include{capitulos/terrivel-feitico}
\end{document}
```

Note a lista de documentos que se quer processar, e que os nomes são separados por uma vírgula, sem nenhum espaço entre eles.

## 8 Símbolos

Diagramação é a disposição de símbolos. E há uma infinidade deles. Citamos nesta apostila alguns deles, mas certamente não o suficiente para atender à sua necessidade. Recomendamos fortemente que mantenha à mão uma cópia do excelente trabalho de Scott Pakin, *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List* [Pak08], que, muito provavelmente, já está em alguma parte de sua instalação do sistema<sup>24</sup>, e que exibe uma lista organizada de aproximadamente cinco mil símbolos que estão a sua disposição.

<sup>24</sup>Em algumas instalações o arquivo é chamado `symbols-a4.pdf`.

Entre os símbolos disponíveis, estão elementos decorativos, símbolos fonéticos, matemáticos, de linguagens arcaicas, musicais, genealógicos, enxadrísticos, químicos, diacríticos incomuns ou compostos, de diagramas de Feynman, de segurança, de legenda em mapas, etcétera. Nada inesperado para um sistema que permite escrever em  $\text{\LaTeX}$ .

## 9 Matemática

Ah, a matemática... Ela é em grande parte a razão pela qual temos o  $\text{\LaTeX}$  (e os computadores!). Aqui, em particular, o  $\text{\LaTeX}$  brilha.

Existem dois “modos” principais nos quais o  $\text{\LaTeX}$  pode operar quando escreve expressões matemáticas: o *modo matemático inline* e o *modo matemático “display”*. Ele está no primeiro, em geral, quando está escrevendo uma fórmula que deverá ocupar um espaço limitado (no meio de um parágrafo, por exemplo), mas também em índices ou em frações, como veremos adiante.

É importante perceber que as regras de espaçamento entre letras são diferentes quando se está trabalhando no modo matemático. As letras são postas em um tipo itálico, e os espaços são desconsiderados entre letras; o espaço entre caracteres como  $=$ ,  $+$  e  $-$  mudam, e parágrafos (duas quebras de linha consecutivas) não são permitidos.

Nesta seção faremos uma pequena incursão na composição de fórmulas usando  $\text{\LaTeX}$ . Tenha em mente que há uma série de parâmetros que afetam a legibilidade de uma expressão matemática — e mencionaremos apenas alguns deles. Em todos os casos, lance mão de seu bom-senso, pergunte a opinião de amigos, e você não deve ter problemas. Mãos à obra!

### 9.1 Entrando no modo matemático

Já mencionamos a existência de dois modos matemáticos. Para escrever uma fórmula em meio a um parágrafo, basta escrevê-la entre  $\backslash ( e \backslash )$ <sup>25</sup>. A fórmula será então tratada pelo  $\text{\LaTeX}$  como qualquer outra palavra no parágrafo:  $\backslash (a + b\backslash )$  resulta em  $a + b$ . A presença de espaços é indiferente no modo matemático. ‘ $a+b$ ’, ‘ $a +b$ ’, e ‘ $a + b$ ’ são todas formas equivalentes. Uma grande vantagem disso é que o autor pode formatar a expressão como melhor lhe convier em termos de legibilidade (para si) quando está a escrever o texto, e o resultado não será “estragado” por isso. Esse é o modo de escrever matemática no meio da linha (*inline*).

Outro modo é colocar a fórmula em um banner, com destaque: é o modo de exibição (*display*). Para usá-lo, coloque a fórmula entre ‘ $\backslash [$ ’ e ‘ $\backslash ]$ ’: a expressão  $\backslash [a\backslash \times b = c.\backslash ]$  faz o  $\text{\LaTeX}$  produzir

$$a \times b = c.$$

(O ponto final foi colocado só para terminar a frase, poderia ser omitido sem maiores consequências que uma frase interminada.)

### 9.2 Para cima e para baixo

É simples: para gerar uma expressão “superscrita” a outra, ( $a^b$ ) usa-se o comando ‘ $\text{\textasciicircum}$ ’ (acento circunflexo, ou chapéu). Para “subscritos”, usamos ‘ $\text{\textasciitilde}$ ’ (*underscore*). Observe os

---

<sup>25</sup>Há quem use cifrões, ‘ $\text{\textdollar}$ ’, envolvendo a expressão matemática, mas essa é uma prática que desencorajamos — é uma herança de versões antigas do  $\text{\LaTeX}$ , e deve cair em desuso.

exemplos, e note a necessidade de agrupar alguns conjuntos de símbolos para obter certos resultados.

$\backslash[ a^b = c^{de} \backslash]$	$a^b = c^d e$
$\backslash[ a^b = c^{\{de\}} \backslash]$	
$\backslash[ a_b = \log c \approx f(b) \backslash]$	$a^b = c^{de}$
$\backslash[ a^{\{c + d\}} = \lim_{a \rightarrow 0} \gamma^{a\tau} \backslash]$	$a_b = \log c \approx f(b)$
$\backslash[ \sum_{i=1}^n i = n(n+1)/2 \backslash]$	$a^{(c+d)} = \lim_{a \rightarrow 0} \gamma^{a\tau}$
	$\sum_{i=1}^n i = n(n+1)/2$

Repare que as expressões “lim” e “log” têm texto escrito de modo diferente. Para escrever texto na fonte romana (i.e., fonte do texto corrente), é preciso sair temporariamente do modo matemático, para que os espaços voltem a valer. Isso pode ser feito usando-se uma *caixa*. (Preste atenção aos espaços!)

$\backslash[ \mbox{para todo } a \in A \mbox{ há um } b \in B \backslash]$	para todo $a \in A$ há um único $b \in B$
--	---

Alguns comandos para digitar expressões matemáticas, como o `\frac` levam mais de um parâmetro. Outros, como o `\choose` operam sobre todo o conteúdo do grupo que o contém, usando como operandos o texto que se encontra à sua esquerda e à sua direita no grupo.

$\backslash[ \frac{a}{b} = \sqrt[n]{cD} \backslash]$	$\frac{a}{b} = \sqrt[n]{cD} = \sqrt[n]{cD} = \sqrt[n]{p}$
$\backslash[ \{a \choose b + c\} = \{a \choose c + d\} \backslash]$	$\binom{a}{a+b} = \binom{a}{b+c}$

Nos exemplos anteriores vimos situações em que parênteses esticaram ou encolheram, junto com o conteúdo que envolvem. Para obter esse efeito, usamos os comandos `\left` e `\right` (sempre aos pares), seguidos do caractere que se deseja expandir (colchetes, chaves, barra vertical ou parênteses). O exemplo a seguir já exhibe um ou outro requinte a mais.

<pre>\[ \left(\int f\right) \stackrel{\text{def}}{=} \left(\int_0^{+\infty} f(x) dx\right) \]</pre>	$\left(\int f\right) \stackrel{\text{def}}{=} \left(\int_0^{+\infty} f(x) dx\right)$
---	--

Primeiro, usamos `\textrm` para que o ‘d’ em  $dx$  (e o “def” em  $\stackrel{\text{def}}{=}$ ) seja escrito em texto romano (ou seja, texto normal). O comando `\stackrel` empilha pequenos textos, e trata o símbolo resultante como um operador relacional (como maior, menor, menor ou igual etc.). Usamos ainda os comandos ‘\!’ e ‘\,’, correspondentes respectivamente aos comandos ‘`\negthinspace`’ e ‘`\thinspace`’ em modo texto, usados (também respectivamente) para aproximar ou afastar elementos do texto horizontalmente<sup>26</sup>.

Você pode obter letras gregas no modo matemático facilmente. Vários outros símbolos estão disponíveis, e a internet é sua amiga para encontrá-los.

<pre>\[ \partial\delta\alpha\beta\gamma &gt; \Gamma &lt; \epsilon \geq\varepsilon \] \[\leq\psi=\sim\neq\leq\geq\in\cup\cap\oplus\cdot\times\div\equiv\] \[\forall\exists\mapsto\rightarrow\nu\]</pre>	$\partial\delta\alpha\beta\gamma > \Gamma < \epsilon \geq \varepsilon$ $\leq \psi = \sim \neq \leq \geq \in \cup \cap \oplus \cdot \times \div \equiv$ $\forall \exists \mapsto \rightarrow \nu$
--	--

E isto é apenas parte do que se pode fazer com o  $\LaTeX$ , apenas tocamos a superfície. Uma boa referência é [Grä00].

## 10 Aspectos estruturais

Chegou a hora de aproximar-mo-nos um pouco mais do  $\LaTeX$ , conhecer-lhe melhor os trejeitos e manhas, falar mais a sua língua. Esse conhecimento é valioso quando quisermos convencê-lo a fazer algo diferente para nós.

### 10.1 Parágrafos marginais

Usar notas marginais no texto pode ser uma maneira interessante de destacar algum conceito. O comando `\marginpar{parágrafo}` acrescenta um parágrafo à margem do parágrafo atual.

É possível mudar drasticamente a aparência de um parágrafo lateral (assim como de qualquer outro tipo de parágrafo): diminuir a fonte em que é escrito, deixá-lo rasgado à direita ou esquerda (seção 5.3), envolvê-lo em uma caixa, rotacioná-lo, etc. — em suma,

<sup>26</sup>Comumente dizemos que esses comandos *produzem espaço horizontal*: `\negthinspace` produz espaço fino negativo, e `\thinspace` produz um espaço fino.

notas  
marginais

Assim.

qualquer transformação. Por exemplo, parágrafos de páginas pares e ímpares são por padrão colocados de modo a que estejam na lateral da folha que ficaria “para fora” caso o texto seja encadernado. Esse comportamento, para ser mais preciso, depende de algumas definições na classe do documento.<sup>27</sup>

---

```
Houve um tempo\footnote{Por volta
de 1920.} em que as pessoas viviam
como se estivessem na década de 20.
```

Houve um tempo<sup>a</sup> em que as pessoas viviam como se estivessem na década de 20.

---

<sup>a</sup>Por volta de 1920.

---

## 10.2 Arquivos auxiliares

O  $\text{\LaTeX}$  se vale de um bom número de arquivos auxiliares para realizar seu trabalho. Tomemos um tempo para observar como funciona o processo de uso de um arquivo auxiliar.

Alguns desses arquivos são produzidos pelo próprio  $\text{\LaTeX}$ , durante a compilação do documento. O índice do documento, suas listas de tabelas e figuras, assim como vários outros, são criados para armazenar as linhas dos índices de elementos de certos tipos. Ao processar um documento com índice, por exemplo, os números das páginas em que as seções se iniciam são armazenados para posterior uso. Cada vez que o texto passa pelo  $\text{\LaTeX}$ , os números de página mais recentemente armazenados nos arquivos auxiliares são colocados nos índices.

Um processo parecido acontece com arquivos externos que são gerados por programas como  $\text{\BibTeX}$  (seção 11),  $\text{\makeindex}$  e  $\text{\makeglossaries}$  (seção 12). A diferença, então, é que para produzi-los é preciso processar o texto fonte com o  $\text{\LaTeX}$ , passá-lo ainda por um (ou vários) desses programas. Depois, ainda deve ser  $\text{\LaTeX}$ ado ao menos mais duas vezes ao menos, para que as referências sejam atualizadas.

## 10.3 Comandos frágeis

Alguns comandos, como todos nós, precisam por vezes de atenção e cuidado, é preciso protegê-los. Você pode fazê-lo com  $\text{\protect}$ , que tem como argumento algum texto que precise ser protegido.

Essa necessidade advém do fato que alguns comandos fazem sentido apenas se presentes em determinada parte do texto, e, se não protegidos, podem causar tumulto se são inadvertidamente removidos de seu contexto-natal por outros comandos.

Exemplos de comandos “transportadores” são  $\text{\section}$  e  $\text{\caption}$ , por exemplo. O texto que é passado como parâmetro para essas sequências de controle não aparece apenas *in loco*, mas é carregado para índices, listas de figuras, ou mesmo para o cabeçalho da página.

Mas não entremos em detalhes ainda. O importante por agora é que, havendo perigo à vista, pode ser necessária proteção.

---

<sup>27</sup>Por exemplo, se você está usando alguma classe de documento padrão, como `article` ou `book`, a opção `twoside` implica que o documento será impresso frente-e-verso, o que geralmente implica que parágrafos marginais serão colocados à direita ou esquerda dependendo de a página a que pertencem ser par ou ímpar (a opção `oneside` faz todo paragrafo marginal aparecer no mesmo lado da página).



## 11 Bibliografia & Cia.: BibTeX

Veremos nesta seção duas abordagens para a composição de bibliografias. Em uma delas, escrevemos a bibliografia linha por linha, assim como escrevemos o texto. O  $\LaTeX$  automaticamente numera as entradas da bibliografia, e você pode referenciá-las com o comando `\cite{rotulo}`.

Outro jeito, muito popular a propósito, de trabalhar com bibliografias, é usando o programa BibTeX. Nessa abordagem, as entradas bibliográficas são escritas em um arquivo de extensão `bib`, seus campos (autor, edição, editora, etc.) são marcados semanticamente, e a formatação é deixada a encargo do programa BibTeX<sup>28</sup>.

### 11.1 Fazendo no muque

O mecanismo original de composição de bibliografias pressupõe que elas estejam postas em um ambiente próprio, o `thebibliography`. Cada entrada possui opcionalmente um rótulo público, que aparecerá entre colchetes quando for citada, e ainda um rótulo interno, que funciona como os rótulos definidos com `\label`, podendo ser referenciado usando o comando `\cite{rotulo}`. Se nenhum rótulo público é fornecido, o  $\LaTeX$  numera as entradas, que coloca na citação.

Outra característica importante deste método é que as referências aparecem exatamente na ordem em que foram declaradas, como seria de se esperar. Isso não acontece, veremos, quando se usa o BibTeX, que automatiza a ordenação dos itens da referência.

O processo de compilação do documento se altera quando se acrescenta bibliografias em um documento, do mesmo modo como acontece quando usam-se referências internas: o processamento do arquivo gera alguns arquivos auxiliares, que são usados para escrever as citações.

```
\begin{thebibliography}{longuissimo}
\bibitem[Tahan83]{malba-tahan} TAHAN, Malba. \emph{O Homem que
Calculava}. Ed. Círculo do Livro. Edição integral. 1983.

\bibitem[Calvino03]{se-um-viajante} CALVINO, Ítalo.
\emph{Se um Viajante numa Noite de Inverno}. Ed. Schwarcz. 2003.
\end{thebibliography}
```

## Referências

[Tahan83] TAHAN, Malba. *O Homem que Calculava*. Ed. Círculo do Livro. Edição integral. 1983.

[Calvino03] CALVINO, Ítalo. *Se um Viajante numa Noite de Inverno*. Ed. Schwarcz. 2003.

Vejamos o papel de cada um dos elementos no exemplo. `longuissimo` é qualquer texto que tenha tamanho maior (ou igual) ao rótulo da mais longa entrada da bibliografia. Ele é usado pelo  $\LaTeX$  para reservar espaço para os rótulos quando ele compõe os itens da bibliografia.

Tanto `Tahan83` quanto `Calvino03` são rótulos que aparecerão, por exemplo, quando se usar o comando `\cite{malba-tahan}`.

<sup>28</sup>Em conjunto com pacotes que porventura você acrescente para configurar esse comportamento.

## 11.2 BibTeX

O programa BibTeX foi criado por Oren Patashnik, há mais de vinte e cinco anos. Com ele produzimos referências bibliográficas a partir de um arquivo (geralmente com a extensão `bib`), em que estão listadas diversas entradas. Essa prática traz-nos diversas vantagens.

- a ordenação dos itens da bibliografia é automatizada,
- a composição visual é uniformizada,
- é possível manter um único banco de referências (algo como fichas catalográficas) — não é preciso re-escrever entradas na bibliografia que se usa com frequência,
- uma formatação diferente pode ser aplicada a tipos diferentes de entradas (livros, revistas, etc.).

Cada entrada do arquivo `bib` tem um tipo, e uma série de campos com seus respectivos valores. Os tipos de entradas padrão incluem livros, revistas, propostas de palestras em conferências (*inproceedings*), artigos (*article*), parte de livros (*inbook* — capítulo, seção etc.), manuais (*manual*), dissertações de mestrado (*masterthesis*), teses de doutorado (*phdthesis*), relatórios (*report*), textos que não foram publicados (*unpublished*) e alguns outros (entre eles miscelâneo — *misc* — quando nada mais servir). Para uma lista de todos os tipos, nada como olhar a documentação do BibTeX em seu sistema.<sup>29, 30</sup>

Para acrescentar a bibliografia ao documento, podemos usar `\bibliography{arquivo}` (`.bib` não é necessário), onde se quer que ela seja incluída.

O comum é que a bibliografia cresça juntamente com o texto. Assim, eventualmente você escreverá um texto em que citará um determinado texto. Você acrescenta o `\cite` correspondente, e depois acrescenta ao arquivo com as entradas bibliográficas a nova entrada. Para que ela apareça no documento, será preciso que o BibTeX processe o documento, após o que o documento ainda deverá ser processado como de costume, para que as referências sejam atualizadas. Para dar conta do recado, processe o documento antes e ao menos duas vezes depois de processá-lo com o BibTeX:

1. Processe o documento uma vez (LaTeXe o documento),
2. BibTeXe o documento, e
3. processe o documento mais duas vezes (a primeira produz as referências corretas e a segunda as coloca no documento produzido).

O pacote `babelbib`, usado em conjunto com `babel` facilita a composição de entradas bibliográficas quando as referências contém itens em diversos idiomas. Problemas com acentuação podem ser solucionados amiúde com o uso de comandos para acentuação como os vistos na seção 3.5.

Terminamos a seção com um exemplo simplório de um arquivo com as entradas bibliográficas para o BibTeX. (O comando `\LaTeXe` produz LaTeX 2<sub>ε</sub>.)

<sup>29</sup>Geralmente o arquivo se chama `btzdoc`.

<sup>30</sup>Também recomendamos a páginas <http://www.bibtex.org> e <http://data.bibbase.org/>. O formato BibTeX é bastante difundido, e diversos programas possuem expansões para facilitar a busca e conversão de entradas de bibliografia usando esse formato.

```

@BOOK{malba-tahan,
  author={Malba Tahan},
  title={O Homem que Calculava},
  publisher={Círculo do Livro}
  year={1983},
  note={edição integral}
}

@MANUAL{lshort,
  author = "Tobias Oetiker and others",
  title = "The Not So Short Introduction to  $\LaTeX$ :
    Or  $\LaTeX$  in 157 minutes",
  month = june,
  note = "on ctan \url{/info/lshort}",
  year = 2010
}

```

## 12 Índices remissivos & Cia.

Glossários e índices remissivos são elementos delicados, por sua natureza distribuída. São como que amarrações de palavras, uma teia que se espalha por toda a tessitura do documento, associando conceitos. Veremos agora como usar esses elementos no  $\LaTeX$ .

Aqui arquivos auxiliares continuam desempenhando um papel importante. Mantemos um fluxo de trabalho análogo ao da preparação de bibliografias (seção 11). Acrescentamos ao documento diretrizes declarando a existência do elemento em questão, e marcamos ainda a as suas manifestações (expressões, indexadas ou definidas num glossário). Glossários demandam uma preparação a mais: é preciso escrever as definições dos termos, ou declarar o arquivo externo em que estão contidas. Tendo feito isso, o procedimento é padrão:

1.  $\LaTeX$ ar o documento,
2. processar o documento para gerar glossário ou índice remissivo,
3.  $\LaTeX$ ar o documento mais duas vezes, para atualizar os índices e referências de páginas.

### 12.1 Índices remissivos

O primeiro passo é acrescentar um pacote que permita gerar índices, como por exemplo `makeidx` (que usamos aqui). Ele funciona em conjunto com o programa `makeindex`, que processa o documento em busca dos índices, anotando as páginas em que ocorrem. Além disso, é preciso colocar o comando `\makeindex`, caso contrário o índice não será produzido (é um jeito fácil de “desligar” o índice, quando oportuno).

O segundo passo é criar as entradas do índice. O que é uma entrada do índice? É uma referência para um ponto (em geral sua página) do texto, associado a uma palavra. Como índices podem ser extensos, é comum que sejam ordenados alfabeticamente. Para associar a um termo uma entrada no índice existe o comando `\index{nome no índice}`, que faz com que, no índice remissivo, seja criada uma entrada apontando para a página corrente, rotulado com “nome no índice”.

Acontece que `makeindex` nem sempre tem pela frente uma tarefa fácil quando se propõe a organizar o índice em ordem alfabética. Por exemplo, como você ordenaria “néctar”, “nervo”, “`\newcommand`”, e “ $\pi$ ”? Para isso existe um modo alternativo de declarar uma entrada no índice, usando a seguinte sintaxe.

```
\index{versao ascii para ordenacao@versão \com{firulas}}
```

(Não, `\com` não é um comando. Agora é.)

Uma última característica: algumas vezes queremos expressar uma hierarquia de conceitos no índice. Isso também é possível (os conceitos-subordinados aparecem todos organizados “dentro” da entrada do elemento hierarquicamente superior, no índice). Como chegar a esse efeito almejadíssimo é algo a respeito de que nos reservamos o direito de calar. Tire a caneta da orelha e tome cá, esta é a tua pulga. Poderá tirá-la daí com uma olhadela na documentação do pacote `makeidx`.

A terceira e última etapa antes que se passe ao processamento do texto é definir em que ponto daquele estará o índice. É nesse exato lugar que deve deixar o comando

```
\printindex
```

Agora é a dança de processamento de sempre, mudando os parezinhos: uma com  $\text{\LaTeX}$ , outra com `makeindex`, e mais duas com  $\text{\LaTeX}$  para terminar bem a farra.

## 12.2 Glossários

A maior parte dos passos envolvidos na composição de um glossário já foi abordada em alguma parte deste documento. Um pacote muito usado para compor glossários é `glossaries`. Como acontece para índices remissivos, é preciso haver `\makeglossaries` em algum ponto do preâmbulo do documento para que o glossário seja de fato processado. Para compor um glossário, definem-se as entradas (dotando-lhes de um rótulo para futura referência); esses rótulos são então usados no texto para referir aos termos que representam; por fim, o documento é processado, pelo  $\text{\LaTeX}$  e por algum programa que permita extrair do documento as ocorrências de termos do glossário.

Definimos rótulos com `\newglossaryentry{rotulo}{configuracao}` (há outros meios). O rótulo é apenas um nome pelo qual nos referiremos ao conceito, como na discussão sobre referências internas na seção 4.4. A configuração é um texto que define várias propriedades do item do glossário (por exemplo, sua forma plural, sua descrição, seu nome, caso texto do item seja diferente de seu rótulo). Há ainda a possibilidade de fornecer um texto opcional que é usado para a ordenação das entradas no glossário. Observe o exemplo. (`\ensuremath` é um comando que garante que seu argumento esteja em modo matemático.)

```
\newglossaryentry{pi}
{
  name={\ensuremath{\pi}},
  description={razão entre o comprimento de uma circunferência
               e seu diâmetro},
  sort=pi
}
```

É possível colocar acrônimos (e siglas) em uma lista separada no glossário. Como fazê-lo é um mistério revelado apenas aos interessados que se dão ao trabalho de investigar o assunto.

Os itens do glossário são assessíveis pelas macros `\gls`, `\Gls`, `\glspl`, `\Glspl`, entre outras, que recebem como parâmetro um rótulo. Os comandos iniciando com letra minúscula apresentam a entrada tal como ela foi declarada (ou o rótulo, se nenhum `nome` foi configurado). Com letra maiúscula, deixam a inicial do item maiúscula (para uso em início de frases, por exemplo). A variante desses comandos que termina por “`pl`” representa uma versão pluralizada do item.

Definidos e empregados os termos do glossário, determinamos a sua localização no texto invocando (i.e., usando) `\printglossaries`. A partir daí, é só processar o documento:

1. `LaTeXar`,
2. processar com o programa `makeglossaries` (ou equivalente), e
3. `LaTeXar` mais duas vezes.

## 13 E agora José?

Uma parte importante de qualquer processo de aprendizado é saber como dar-lhe continuidade, sempre. As pessoas que usam `LaTeX` fazem-no com os objetivos mais diversos, e tem interesses dos mais variados. Constituem uma comunidade ativa e prestativa, e a rede pulula de fóruns, blogs e pessoas interessadas em ensinar e aprender mais sobre a composição de documentos bem-feitos, seja usando `LaTeX`, `TeX` ou mesmo outra tecnologia qualquer.

Algo frequente, quando ainda se está a habituar-se ao `LaTeX`, é precisar fazer algo e não saber como fazê-lo. Além de excelentes livros sobre o assunto (veja as referências bibliográficas!), não deixe de integrar ativamente a comunidade de `LaTeXistas`. Pergunte, investigue, colabore. Queremos conhecê-lo!

## 14 Utilidades

A partir de agora estás outorgado o título de `LaTeXnico`! O que vem adiante são apenas adendos ao teu cinto de utilidades, mas havendo dominado o material até aqui, deves estar apto a enfrentar a maior parte das quimeras tipográficas que o aventureiro compositor encontrará em uma jornada habitual. No mais, não hesite em convocar a comunidade, que não se fará surda a qualquer pedido de auxílio.

A lista abaixo contém algumas (poucas!) sugestões de pacotes que você pode achar interessante investigar. Existem vários pacotes que possuem finalidades parecidas, quando não idênticas — fica à sua escolha. A ideia é que você conheça um pouco do que dá para fazer com o `LaTeX`, sua nova ferramenta. Sem mais delongas, à lista.

**hyperref** Cria hiperlinks dentro do próprio documento, além de controlar seu aspecto. Tem forte integração com a estrutura de documentos `pdf`, permitindo controlar propriedades como *autor*, *língua*, etc.

**url** Cria o comando `url`, que encapsula páginas na *internet* faz sua quebra de modo “inteligente”.

**xcolor** Deixa seu texto mais colorido!

**fancyhdr** Personaliza o cabeçalho e rodapé de páginas, exibindo, por exemplo, a seção atual, nome do autor ou qualquer texto.

**tikz** Desenhe figuras com texto! Visite <http://www.texample.net/> para ver do que ele é capaz.

**beamer** uma classe de documento para compor slides.

**amsmath** Pacote da *American Mathematical Society* com vários comandos para facilitar a composição de expressões matemáticas.

**a0poster** Pôsteres em a0!

**microtype** Microtipografia.

**memoir** Uma classe de documento que estende e aprimora grandemente as classes documento tradicionais, acrescentando uma série de outras categorias.

**multicolumn** Permite usar um número variável de colunas no texto.

**indentfirst** Recua a primeira linha do primeiro parágrafo de seções.

**belbib** Traduz palavras da bibliografia, como “edição”, “ano”, etc.

**helvet** Permite usar a fonte Helvética no texto.

# GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

[<http://fsf.org/>](http://fsf.org/)

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under

this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add



no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

### **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

## Índice

- \\, 8
- alinhamento
  - alinhado à direita, 21
  - alinhado à esquerda, 21
- ASCII, 20
- `\caption`, 23
- center (ambiente), 21
- centralizado, 21
- comando, 10
- espaços duros, 17
- figuras, 22
  - legenda, 23
- figure
  - legenda, 23
- flushleft, 21
- flushright, 21
- flutuante (elemento), 21
- fonte (texto), 6
- `\index`, 35
- `\label`, 20
- letras, 8
- listas, 20
  - não numeradas, 20
  - numeradas, 20
- macro, 25
- marcação semântica, 15
- modo matematico
  - display, 29
  - inline, 29
- modo matemático, 16, 29
- `\newcommand`, 25
- parágrafo, 15
- `\printindex`, 36
- `\ref`, 20
- reticências, 17
- seção, 19
- sumário, 19
- tabelas, 23
- texto fonte, 6
- verbatim, 14
- versão `\com{firulas}`, 36
- WYSIWYG, 6
- WYSIWYM, 6





## Referências

- [flo93] *Formatting documents with floats*, volume 21, 1993.
- [Grä00] Grätzer, George: *Math into L<sup>A</sup>T<sub>E</sub>X*. Birkhäuser Boston, 3<sup>a</sup> edição, 2000.
- [Knu84] Knuth, Donald E.: *The T<sub>E</sub>Xbook*. Addison-Wesley, 1984, ISBN 0-201-13448-9.
- [Lam94] Lamport, Leslie: *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison-Wesley, 1994, ISBN 0-201-52983-1.
- [Mit05] Mittelbach, Frank: *The L<sup>A</sup>T<sub>E</sub>X companion*. Addison-Wesley, Boston, 2<sup>a</sup> edição, 2005.
- [O<sup>+</sup>10] Oetiker, Tobias *et al.*: *The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>: Or L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> in 157 minutes*, 2010. on ctan [/info/lshort](#).
- [Pak08] Pakin, Scott: *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List*. 2008.