

L^AT_EX E_aD

AULA 7

Criando Comandos e Ambientes

Sumário

Criando Comandos.....	2
Comandos Simples.....	2
Comandos com Parâmetros	3
Comandos Duais.....	4
Redefinindo Comandos	5
O comando <i>renewcommand</i>	5
Comando <i>def</i>	5
Criando Ambientes	6
Ambientes sem Argumentos.....	6
Ambientes com Argumentos.....	6
Redefinindo ambientes	8
REFERÊNCIAS	9



Olá Pessoal, voltamos ao nosso layout padrão das aulas, espero que tenham gostado de trabalhar com o BEAMER, eu, particularmente, a cada vez que estudo um pouco sobre LaTeX fico mais fascinado com as interessantes e inacabáveis ferramentas que podemos utilizar em nosso meio científico. Vocês hoje vão aprender a facilitar a escrita em LaTeX, com comandos criados por vocês mesmos, onde sua maior utilização se dá ao utilizar comandos já existentes mas usados com maior simplicidade. Vamos lá!

Criando Comandos

A criação de novos comandos é feita ao se colocar a seguinte estrutura no preâmbulo:

```
\newcommand{comando}[nº de parâmetros][valor opcional]{definição}
```

No nome do comando não se pode inserir números, delimitadores, espaços e nem caracteres especiais, lembrando ainda que o LaTeX distingue letras maiúsculas de minúsculas (o que é um ponto positivo!).

Comandos Simples

Os comandos criados sem parâmetros são os mais simples, a sua estrutura básica é:

```
\newcommand{comando}{definição}
```

Um exemplo clássico de seu uso é para o comando *seno*, onde no LaTeX, até aqui usamos o `\sin`, uma vez que a sigla para o *seno* em inglês é o `\sin` e em português é `\sen`, portanto, no preâmbulo coloca-se:

```
\newcommand{\sen}{\mathrm{sen}}
```

Lembrando de aulas anteriores, que quando precisávamos escrever `\sen`, sempre devíamos utilizar o comando `\mathrm` que faz o romano reto.



Abra um arquivo, crie um preâmbulo e crie os comandos:

```
\newcommand{\tg}{\mathrm{tg}}
```

```
\newcommand{\e}{\mathrm{e}}
```

```
\newcommand{\max}{\mathrm{m`ax}}
```

E utilize os comandos no texto:

```
$$\tg(x)$$
```

```
$$\e^x$$
```

```
$$\max(a,b)$$
```

Comandos com Parâmetros

Os comandos com parâmetros são mais sofisticados que os anteriores, é indicado qual o número de parâmetros e estes são especificados pelo símbolo # seguidos da posição, ou seja, #1, #2, #3 especificam o primeiro, segundo e terceiro parâmetros, respectivamente.

Vamos fazer um exemplo prático:

Temos que $\|x\|$ é chamado norma de x . Então vamos criar (detalhadamente!) um comando para não precisar digitar os comandos dos delimitadores para norma, olhando a estrutura básica mostrada anteriormente temos

`\newcommand{comando}[nº de parâmetros][valor opcional]{definição}`

Vamos fazer o comando que se chamará `\norma` o número de parâmetros será 1 (neste caso temos que o parâmetro nada mais é que o argumento do que estamos criando), neste caso não temos o valor opcional e a definição é dada pelo argumento, logo temos que a sintaxe ficará

`\newcommand{\norma}[1]{||#1||}`

Ou seja, temos a `\norma{x}` irá gerar $\|x\|$, assim como `\norma{x+y}` irá gerar $\|x + y\|$, percebemos então que o #1 significa o valor a ser substituído pelo que colocarmos dentro dos colchetes.

Outro exemplo, porém agora com dois parâmetros, temos que uma n -upla é um pequeno conjunto com n elementos, ou seja, podemos representar por (x_1, x_2, \dots, x_n) , e para não precisarmos do código

`(x_1,x_2,\ldots,x_n)`

toda vez que precisarmos adicionar uma n -upla, vamos criar um comando para facilitar nossa vida, então agora temos dois parâmetros, o x e o n .

Então temos o comando

`\newcommand{\nupla}[2]{${\#1}_1,{\#1}_2,\ldots,{\#1}_{\#2}}$`

Então se fizemos `\nupla{a}{j}` obteremos (a_1, a_2, \dots, a_j) . Observe ainda que neste caso colocamos os cifrões junto com a definição do comando, o que faz com que não precisemos colocar ao digitar o comando, basta utilizá-lo sem os cifrões

mesmo, porém como isso as vezes pode confundir um pouco aconselhamos utilizar o novo comando sempre SEM os cifrões. Veja o Exemplo 7.1.4, página 50 de Apostila de LaTeX (Cariello, 2011) onde é mostrado como criar um comando para fazer com maior facilidade a subtração de duas frações, este exemplo consiste da criação de um comando com quatro parâmetros. Os comando com parâmetros podem possuir no máximo nove parâmetros.



Alguns comandos possuem parâmetros opcionais, por exemplo o comando `\sqrt` que possui o parâmetro opcional que é o índice da raiz, ou seja `\sqrt[índice]{argumento}` onde o índice pode ser simplesmente ignorado, tornando o comando na raiz quadrada.

Comandos Duais

Basicamente, consistem de comandos que funcionam tanto no modo matemático como no modo de texto normal, são utilizados com o comando `\ensuremath`.

Toda vez que queremos criar os símbolos dos conjuntos numéricos, precisamos escrever:

`\mathbb{N, Z, Q, I, R, C}` para obter $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{I}, \mathbb{R}, \mathbb{C}$.

Então para facilitar nossa vida podemos definir:

`\newcommand{\N}{\mathbb{N}}` para gerar \mathbb{N} com o comando `(\N)` e assim sucessivamente para os outros conjuntos numéricos. Porém, este tipo de símbolo precisamos utilizar em diversos momentos, seja em uma escrita estritamente matemática ou em outra como simplesmente citar o conjunto \mathbb{N} dos naturais como acabei de fazer, para isso, ao definir estes comandos podemos usar

`\newcommand{\N}{\ensuremath{\mathbb{N}}}`

E assim podemos usar o `\N` tranquilamente em qualquer lugar, tanto em texto normal ou texto matemático.

Redefinindo Comandos

Aprendemos a criar comandos, agora vamos aprender a redefinir alguns comandos.

O comando *renewcommand*

Anteriormente utilizamos o comando *newcommand* neste vamos utilizar o *renewcommand* que intuitivamente é utilizado para redefinir alguns comandos já existentes, novamente voltemos ao exemplo do \sin , no lugar de criar um comando novo como fizemos anteriormente para o \sen , poderíamos fazer

```
\renewcommand{\sin}{\mathrm{sen}}
```

Assim, toda vez que fosse digitado o comando \sin , apareceria o \sen , só não é aconselhável utilizar a redefinição de comando quando for necessário utilizar ambos os comandos no documento.

Já o comando \providecommand serve para criar e redefinir comandos, desde que os mesmos não existam, ou seja, você pode criar um comando, se ele já existir ficará como esta, caso não exista ficará como você criou. Exemplo:

```
\providecommand{\sin}{\mathrm{sen}}
```

 não funcionará pois o comando \sin existe.

```
\providecommand{\sen}{\mathrm{sen}}
```

 funcionará caso não exista o comando \sen .

Comando *def*

O comando \def serve para criação de comandos bem como redefinição, porém não se importando em saber se o comando já existe ou não, ele simplesmente o substitui. Também não é possível a criação de comandos com parâmetros.

Exemplos do uso do \def :

```
\def\N{\mathbb{N}}
```

```
\def\ex{e^x}
```

Criando Ambientes

Os ambientes são muito usados em LaTeX e possuem configurações específicas, por exemplo o ambiente *center* que centraliza o texto que se encontra dentro do `\begin{center} ... \end{center}`. Outros ambientes conhecidos são os de criação de matrizes, ambientes de equações, tabelas, enumerações e outros.

A estrutura básica para criação de um ambiente é a seguinte:

```
\newenvironment{ambiente}[número de parâmetros][valor do opcional]
{início do ambiente}{fim do ambiente}
```

Ambientes sem Argumentos

São ambientes mais simples, onde você apenas digita o texto dentro do ambiente sem precisar estabelecer outras configurações. Veja o exemplo a seguir.

Vamos criar um ambiente que deixe o texto inserido em negrito e centralizado, vamos chama-lo de *negritocentro*, portanto devemos acrescentar no preâmbulo o comando

```
\newenvironment{negritocentro}{\begin{flushright}\bf}{\end{flushright}}
```

e no texto coloque

```
\begin{negritocentro}
```

TEXTO

```
\end{negritocentro}
```

Faça o teste!!

Você também pode adicionar junto ao *begin* assim como esta o `\bf` os comandos `\it` para itálico e todos os comandos para tamanho de fontes como por exemplo o `\Huge`.

Ambientes com Argumentos

Assim como para os comandos com parâmetros (argumentos) temos que os ambientes também podem ter até 9 parâmetros sendo que o primeiro deles pode ser opcional.

Vamos criar um ambiente para demonstração (para os não matemáticos, uma demonstração sempre possui um quadradinho no final, as vezes preenchido as vezes não} existe o ambiente `\begin{proof} ... \end{proof}` que já o faz, mas o quadradinho é feito transparente, supondo que queiramos que o quadradinho seja preenchido então vamos criar um ambiente para que faça a demonstração e coloque este quadradinho.

O comando `\rule{altura}{largura}` produz um retângulo em preto com a altura e largura estipulados e aqui vamos usar a unidade de medida *ex* própria do LaTeX que equivale ao tamanho de um “*x*” na fonte usada e o comando `\hfill` faz com que o retângulo seja colocado ao final da linha.

Então temos o comando

```
\newenvironment{dem}[1][Demonstração]{\textbf{#1:}\ }{\hfill\rule{1ex}{1ex}}
```

O “dem” consiste do nome do nosso ambiente, o [1] diz que temos um parâmetro, o [Demonstração] é o nome que já estará no início do ambiente, e o `{\textbf{#1:}\ }` é a variação do parâmetro, suponha que queiramos colocar a palavra Prova do lugar de Demonstração, para isso seria necessário variar o parâmetro de um modo que já veremos, o `{\hfill\rule{1ex}{1ex}}` faz com que apareça o quadradinho no final no ambiente, que será usado da seguinte maneira:

```
\begin{dem}
```

```
DEMONSTRAÇÃO
```

```
\end{dem}
```

ou com a variação do parâmetro temos

```
\begin{dem}[Prova]
```

```
DEMONSTRAÇÃO
```

```
\end{dem}
```



Crie um ambiente com dois parâmetros, que são textos que ficarão dentro de molduras, centralizadas e com espaço horizontal de 5cm entre elas. Veja resposta em Cariello (2011) página 53.

Redefinindo ambientes

Resume-se em utilizar o comando `\renewenvironment` para mudar alguns ambientes, ironicamente vamos mudar o ambiente *center* para alinhado à direita.

Vamos usar os códigos

```
\renewenvironment{center}{\begin{flushright}}{\end{flushright}}
```

Agora toda vez que se usar o comando `\begin{center} ... \end{center}` o texto ficará alinhado à direita (Não façam isso em casa!!!).

Pessoal, só por curiosidade, a palavra *environment* é ambiente em inglês.

Até a próxima aula.

REFERÊNCIAS

CARIELLO, Daniel. CARNEIRO, Evaneide Alves. REZENDE, Germano Abud de. Apostila de LaTeX. Universidade Federal de Uberlândia: Uberlândia-MG, 2011. 70p.