

Uživatelem definované příkazy

K definování nových příkazů slouží příkaz `\newcommand`. Jeho prvním parametrem je název nového příkazu a druhým parametrem je text, který má být příkazem produkován.

Příklady

1. `\newcommand{\la}{\leftarrow}`

2. `\newcommand{\wa}{\widetilde{a}}`

3. $D^{[2]} \times D^{[3]}$ `\newcommand{\DD}{D^{[2]} \times D^{[3]}}`
`\newcommand{\DD}{\ensuremath{D^{[2]} \times D^{[3]}}}`
Příkaz `\ensuremath` vyhodnotí, zda se nenachází v matematickém prostředí a v případě potřeby dodá hranice matematiky.

Příkazy s parametrem

$$(f^2)^{\left[\left[\frac{A^2}{B-1}\right]\right]}$$

```
\newcommand{\fsq}{\ensuremath{(f^2)^{\left[\left[\frac{A^2}{B-1}\right]\right]}}}
```

Příkaz `\newcommand` má volitelný argument, kterým lze specifikovat počet parametrů nového příkazu.

```
\newcommand{\sq}[1]{\ensuremath{(#1^2)^{\left[\left[\frac{A^2}{B-1}\right]\right]}}}
```

Použití:

$$(r^2)^{\left[\left[\frac{A^2}{B-1}\right]\right]}$$

```
\[\sq{r}\]
```

$$\langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle$$

```
\newcommand{\vct}[2]{\langle \dots, 0, \dots, \overset{#1}{#2}, \dots, 0, \dots \rangle}

$$\mathbf{\langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle}$$

```

Volitelné argumenty

Uživatel si může definovat své příkazy s jedním volitelným argumentem.

$$a_1 + a_2 + \cdots + a_n$$

```
\newcommand{\Sum}{a_{1}+a_{2}+\cdots+a_{n}}  
$\Sum$
```

$$a_1 + a_2 + \cdots + a_n \text{ ale } a_1 + a_2 + \cdots + a_m$$

```
\newcommand{\NewSum}[1][n]{a_{1}+a_{2}+%  
\cdots+a_{#1}}  
$\NewSum$ ale $\NewSum[m]$
```

Příkaz `\newcommand` může mít až devět argumentů, *pouze první* však může být volitelný. Následující příkaz má dva argumenty, jeden volitelný.

$$x_1 + x_2 + \cdots + x_n$$

$$a_1 + a_2 + \cdots + a_n$$

$$a_1 + a_2 + \cdots + a_i$$

```
\newcommand{\NNSum}[2][n]{#2_{1}+#2_{2}+%  
\cdots+#2_{#1}}  
$\NNSum{x}$\  
$\NNSum{a}$\  
$\NNSum[i]{a}$
```

Změna již existujících příkazů

Příkaz `\renewcommand`

```
\renewcommand{\qedsymbol}{\ensuremath{\blacksquare}}  
\renewcommand{\indexname}{Malý rejstřík}
```

Příkaz `\providecommand` – pokud je již příkaz definován, nechá ho beze změny, pokud ne, funguje jako `\newcommand`.

Zjištění definice příkazu

Příkazy `\show` a `\showthe`

Příkaz `\def`

```
\def\la{\leftarrow}
```

$\langle a, b \rangle$

```
\def\v<#1>{\langle#1\rangle}  
$\vv<a,b>$
```

Uživatелеm definovaná prostředí

Modifikace existujících prostředí

```
\newenvironment{jmeno} [par] [implic] {zacatek}{konec}
```

Důkaz. Text.

```
□ \newenvironment{demo}  
  {\begin{proof}\em}  
  {\end{proof}}  
  \begin{demo}  
  Text.  
  \end{demo}
```

- První položka
- Druhá položka

```
\newenvironment{tucsez}{\begin{itemize}  
\setlength{\itemsep}{0pt}\bfseries  
}{\end{itemize}}  
\begin{tucsez}  
\item První položka  
\item Druhá položka  
\end{tucsez}
```

Nově definovaná prostředí mohou mít také parametry. Formální parametry mohou být umístěny pouze v definici začátku prostředí. Stejně jako u příkazu `\newcommand` může i zde být první parametr volitelný. Jeho implicitní hodnota se zapisuje do hranatých závorek za parametr označující počet parametrů nového prostředí.

Theorem 1. *Text.*

```
\newenvironment{theoremRef}[1]
{\begin{theorem}\label{T:#1}}
{\end{theorem}}
\begin{theoremRef}{label}
Text.
\end{theoremRef}
```

\Rightarrow *Podmínka:*

Důsledek.

```
\newenvironment{citat}[1]{\begin{quotation}
{\noindent$\Rightarrow$\textit{#1:}}\}[2mm]
}{\end{quotation}}
\begin{citat}{Podmínka}
Důsledek.
\end{citat}
```

Prostředí šířky šest centimetů,
bez odsazení.

Prostředí šířky
šest centimetů,
bez odsazení.

```
\newenvironment{narrow}[1][6cm]
{\noindent\begin{minipage}{#1}}
{\end{minipage}}
\begin{narrow}
Prostředí šířky šest centimetů, bez odsazení.
\end{narrow}
\vskip 2mm
\begin{narrow}[3cm]
Prostředí šířky šest centimetů, bez odsazení.
\end{narrow}
```

```
\newenvironment{vcentrpage}
{\newpage\vspace*{\fill}}
{\vspace*{\fill}\par\pagebreak}
```

Čítače

Každé číslo, které L^AT_EX samostatně vypisuje, je spojeno s nějakým čítačem. Čítače mají svá jména, některé jsou předdefinovány, jiné si může uživatel definovat sám.

Příkazem `\setcounter` můžeme do čítače vložit zvolenou hodnotu.

```
\setcounter{chapter}{2}
```

Příkazem `\addtocounter` můžeme přičíst k dosavadní hodnotě čítače zvolenou hodnotu.

```
\addtocounter{page}{2}
```

Obsah čítače můžeme v textu zobrazit v různých podobách:

<code>\arabic{page}</code>	→ 4		<code>\roman{page}</code>	→ iv		<code>\alph{page}</code>	→ d
<code>\$_\fnsymbol{page}\$</code>	→ §		<code>\Roman{page}</code>	→ IV		<code>\Alph{page}</code>	→ D

Pro výpis čísla vytváří L^AT_EX ke každému čítači příkaz, jehož název je tvořen předponou `\the`. Za ni se připojí jméno čítače (například `\thesection` pro výpis čísla sekce). Změna způsobu výpisu:

```
\renewcommand{\thesection}{\Roman{section}}
```

Definice vlastních čítačů:

```
\newcounter{mujcitac}  
\newcounter{mujcitac}[hlavnicitac]
```

Volitelný argument je jméno nadřazeného čítače. Jestliže se změní jeho hodnota, závislý čítač se vynuluje. Např.

```
\newcounter{section}[chapter]
```

Má-li nějaký čítač podřazené čítače, které mají být nulovány, pak zvyšování jeho hodnoty je nutné provádět příkazem `\stepcounter{čítač}` nebo `\refstepcounter{čítač}`. Oba příkazy zvyšují hodnotu čítače o jedničku, příkaz `\refstepcounter` navíc definuje hodnotu, kterou lze snímat pomocí návěští (příkazem `\label`).

Toto je text před příkladem.

Příklad 1.1: A zde máme text příkladu.

```
\newcounter{priklad}[section]
\newenvironment{priklad}
{\begin{quote}
  \refstepcounter{priklad}
  \textbf{Příklad \thesection.\thepriklad:}}
{\end{quote}}
Toto je text před příkladem.
\begin{priklad}
A~zde máme text příkladu.
\end{priklad}
```

Chceme-li vytvořit závislost již definovaného čítače na nějakém jiném, můžeme použít příkaz

```
\@addtoreset{závislý}{nadřizený}.
```

Např. vynulování čítače obrázků v každé sekci provedeme příkazem

```
\@addtoreset{figure}{section}.
```

Hodnota uložená v čítači je přístupná pomocí příkazu `\value`

```
\setcounter{citac}{\value{jinycitac}}
```

Speciální čítače

`tocdepth` a `secnumdepth`