

# Definice prostředí

- `\newenvironment{název}[param]{začátek}{konec}`
- `\begin{název}` bude pak nahrazeno začátkem, `\end{název}` koncem
- parametry lze používat jen v začátku
- `\newenvironment{příklad}%  
{\begin{quote}\textbf{Příklad: }}%  
\end{quote}}`
- ...
- `\begin{příklad}Text příkladu.\end{příklad}`

# Pojmenované parametry (1)

- časté u moderních balíčků
- existuje několik balíčků (pgfkeys, xkeyval), nejjednodušší rozhraní má **keycommand**:
- **`\newkeycommand{\příkaz}[jméno1=default1, jméno2=default2]{význam}`**
- v těle příkazu se parametry využívají pomocí **`\commandkey{jméno}`**
- použití: **`\příkaz[jméno=hodnota]`**

# Pojmenované parametry (2)

- lze i prostředí:

```
\newkeyenvironment{název}[jméno1=default1,  
jméno2=default2]{zahájení}{zakočnění}
```

- například centrováný blok dané šířky:

```
\newkeyenvironment{blok}[sirka=10cm]  
{\begin{center}  
  \begin{minipage}{\commandkey{sirka}}  
  \end{minipage}\end{center}}
```

použití: **\begin{blok} ... \end{blok}**

**\begin{blok}[sirka=8cm] ... \end{blok}**

# Čítače

- vhodné pro automatické číslování
- **`\newcounter{čítač}`** vytvoří nový čítač
- **`\setcounter{čítač}{hodnota}`** přiřadí mu hodnotu
- **`\addtocounter{čítač}{hodnota}`** přičte k čítači danou hodnotu
- **`\stepcounter{čítač}`** zvětší o 1
- **`\refstepcounter{čítač}`** zvětší o 1 a pokud se po něm objeví **`\label`**, **`\ref`** pro něj vydá hodnotu čítače

# Hodnoty čítačů

- **`\thečítač`** vysází do textu hodnotu
- **`\arabic{čítač}`** vysází hodnotu daným stylem
- **`\Roman{čítač}`**
- **`\roman{čítač}`**
- **`\alph{čítač}`**
- **`\Alph{čítač}`**
- **`\fnsymbol{čítač}`** (symbol poznámky pod čarou)
- **`\value{čítač}`** – vrátí hodnotu jako číslo, např. pro přiřazení do jiného čítače

# Čítače – příklad

- vylepšená verze prostředí pro příklady – automatické číslování

```
\newcounter{priklad}
```

```
\newenvironment{priklad}%
```

```
{\refstepcounter{priklad}%
```

```
\begin{quote}\textbf{Příklad~\arabic{priklad}: }}%
```

```
{\end{quote}}
```

# Automatický reset čítače

- nepovinným parametrem při vytvoření čítače lze určit **nadřazený čítač** – nový se resetuje, kdykoli se změní hodnota nadřazeného

- např. chceme číslovat příklady v kapitolách (5.1):

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

```
\newenvironment{priklad}%
```

```
{\refstepcounter{priklad}%
```

```
\begin{quote}
```

```
\textbf{Příklad~\thechapter.\thepriklad: }}%
```

```
\end{quote}}
```

# Délky

- pro ukládání rozměrů a manipulace s nimi

- vytvoření:

**`\newlength{\název}`**

- změny:

**`\setlength{\název}{délka}`**

nastaví délku

**`\addtolength{\název}{délka}`**

přičte délku

**`\settowidth{\název}{text}`**

nastaví šířku textu

**`\settoheight{\název}{text}`**

nastaví výšku textu

**`\settodepth{\název}{text}`**

nastaví hloubku textu





# Balík ifthen

- pro vytváření podmíněných konstrukcí  
`\usepackage{ifthen}`
- základní příkaz:  
`\ifthenelse{podmínka}{když platí}{když neplatí}`
- příklad – podle hodnoty čítače `příklad` zhodnotíme počet příkladů:  
`\ifthenelse{\value{příklad} < 2}`  
    {Máme nanejvýš jeden příklad.}  
    {Máme více příkladů.}

# Podmínky

- `=, <, >` porovnání čísel
- `\isodd{číslo}` je liché?
- `\equal{řet1}{řet2}` porovnání řetězců
- `\isundefined{příkaz}` je příkaz nedefinován?
- `\lengthtest{ ... }` porovnání délek
- `\boolean{jméno}` pravdivostní hodnota (viz dále)
  
- kombinování: `\and, \or, \not, \l( ... \l)`

# Příklad

- příkaz s nepovinným parametrem, není-li uveden, přebírá hodnotu povinného
- např. pro náš příkaz `\pojem`, aby umožnil do textu vložit skloňované slovo, zatímco do rejstříku 1. pád

```
\newcommand{\pojem}[2][ ]{%  
\ifthenelse{\equal{#1}{}}{\index{#2}}{\index{#1}}%  
\emph{#2}}
```

*výchozí hodnota prázdná*

Všechna `\pojem{data}` jsou uložena  
v `\pojem[soubor]{souborech}`.

# Pravdivostní hodnoty

- vytvoření pravdivostní hodnoty (příznaku):  
**`\newboolean{název}`**  
**`\provideboolean{název}`** (nevadí, když existuje)
- nastavení hodnoty:  
**`\setboolean{název}{hodnota}`**  
kde *hodnota* je **true** nebo **false**
- využití:  
**`\ifthenelse{\boolean{název}}{...}{...}`**

# Příklad

- centrální řízení, zda příklady číslovat

```
\newcounter{priklad}
```

```
\newboolean{cisla}
```

```
\setboolean{cisla}{true}
```

```
\newenvironment{priklad}%
```

```
{\refstepcounter{priklad}%
```

```
\begin{quote}\textbf{Příklad%
```

```
\ifthenelse{\boolean{cisla}}{~\thepriklad}{}}%
```

```
: }}{\end{quote}}
```

# Cyklus

- **`\whiledo{podmínka}{tělo}`**
- dokud platí *podmínka*, opakuje *tělo*
- **`\newcounter{odpocet}`**  
**`\newcommand{\odpocet}[1]{`**  
    **`\setcounter{odpocet}{#1}\theodpocet`**  
    **`\whiledo{\value{odpocet}>1}{ \ldots \`**  
        **`\addtocounter{odpocet}{-1}`**  
        **`\theodpocet`**  
    **`}`**  
**`}`**

`\odpocet{3}` → 3 ... 2 ... 1

# Úložné boxy

- `\newsavebox{\název}` vytvoří úložný box
- `\savebox{\název}{obsah}` vysázený *obsah* uloží do boxu, nemá viditelný výstup
- `\usebox{\název}` vysází obsah boxu
- příklad:  
`\newsavebox{\ostrava}`  
`\savebox{\ostrava}{!!!}`  
`\usebox{\ostrava}` Nazdar `\usebox{\ostrava}`



# Balíky

- soubory s příponou **.sty**, vlastní jméno odpovídá názvu balíku
- zahájit  
**`\ProvidesPackage{název}[datum verze]`**
- místo `\usepackage` používat **`\RequiresPackage`**
- neměl by generovat výstup (vkládá se před začátkem dokumentu), jen definovat příkazy apod.
- **`\AtBeginDocument{akce}`**  
provést *akci* při `\begin{document}`

# Volby balíku

- **`\DeclareOption{název}{akce}`**
- deklarace volby – pokud ji přidáte ke vložení balíku, provede se akce
- zpracování voleb provede (volat za deklaracemi)
- **`\ProcessOptions\relax`**
- **`\DeclareOption*{akce}`**  
*akce* se provede pro nedeklarované volby

# Příklad – balík tul

```
\ProvidesPackage{tul}[2021/01/18 version 1.10]
\RequirePackage{xcolor}
\definecolor{TULcolor}{cmyk}{0.5, 1, 0.5, 0.2}
\definecolor{FMcolor}{cmyk}{0, 0.6, 1, 0}
\newcommand{\barva}{TULcolor}
\DeclareOption{FM}{
  \renewcommand{\barva}{FMcolor}}
\DeclareOption*{
\PackageWarning{tul}{Unknown ‘\CurrentOption’}}
\ProcessOptions\relax
```