

Velmi stručný úvod do L^AT_EXu

V. Stránský

24. ledna 2015

Abstrakt

Tento text slouží jako návod k L^AT_EXu pro úplně začátečníky. Jeho cílem je seznámit studenty středních škol (řešitele Semináře vědy a techniky) se základy tohoto publikačního systému, a proto jsou některé informace v něm obsažené poměrně zjednodušené. Také ani zdaleka neobsahuje všechny možnosti, které L^AT_EX nabízí. Pro podrobnější a přesnější informace doporučuji [1] nebo [4].

1 Úvod

L^AT_EX (čti [latech] nebo [lejtek]) je program pro tvorbu vysoce kvalitních textových dokumentů a prezentací. Mezi jeho výhody oproti MS Word (se kterým jsme se seznámili v [7]) patří především sazba textu ve vyšší kvalitě, snadnější tvoření úhlednějších matematických vzorců, automatické číslování objektů a možnost se na ně odkazovat. Další výhodou je to, že L^AT_EX pomáhá uživateli přehledně strukturovat dokument. Naopak jeho nevýhodou je obtížná sazba plovoucích objektů, jako jsou obrázky a tabulky. Začátky práce s L^AT_EXem mohou být těžké, jeho používání je velmi podobné např. tvorbě webových stránek pomocí HTML. Pro vytvoření dokumentu tak musíme nejdříve napsat zdrojový kód, který nám L^AT_EX přeloží do požadovaného formátu, jež si můžeme zobrazit.

2 Instalace L^AT_EXu

Nejprve je nutné stáhnout kompilátor L^AT_EXu. Pro **Mac** existuje kompilátor MacTeX, který lze stáhnout z adresy tug.org/mactex/. Stáhneme instalační soubor MacTeX.pkg, nainstalujeme jej a vše by mělo fungovat. Na **Linuxu** se nejčastěji používá TeX Live, který lze stáhnout z www.tug.org/texlive/.

Pro operační systém **Windows** (jemuž se budeme věnovat primárně) je jednou z možností software MiKTeX, který kompilátor L^AT_EXu obsahuje. Instalační soubor lze nalézt na oficiálních stránkách na adrese www.miktex.org/download. Jeho spuštěním nainstalujeme MiKTeX do námi zvolené složky (doporučuji použít výchozí nastavení). Po dokončení instalace by měl být MiKTeX připraven k použití, je však vhodné ještě zkontrolovat, zda je nainstalována i čeština (slovenština). To provedeme ve verzích Windows 7 a starších tak, že otevřeme seznam programů a spustíme okno *MiKTeX* → *Maintenance* → *Settings*. Ve verzi 8 pak zmáčkneme tlačítko Windows a mezi programy najdeme dlaždici *Settings* s logem MiKTeXu a tu otevřeme. V záložce *Languages* pak zkontrolujeme, zda je zaškrtnuté políčko u češtiny (slovenštiny). Stejně tak ověříme, že je zaškrtnuté v záložce *Packages* → *Language Support*. Pokud není, zaškrtneme jej a potvrdíme OK.

V tuto chvíli je na našem počítači nainstalován L^AT_EX. Teoreticky bychom mohli napsat zdrojový kód v Notepadu a přeložit jej do formátu pdf pomocí konzole příkazem

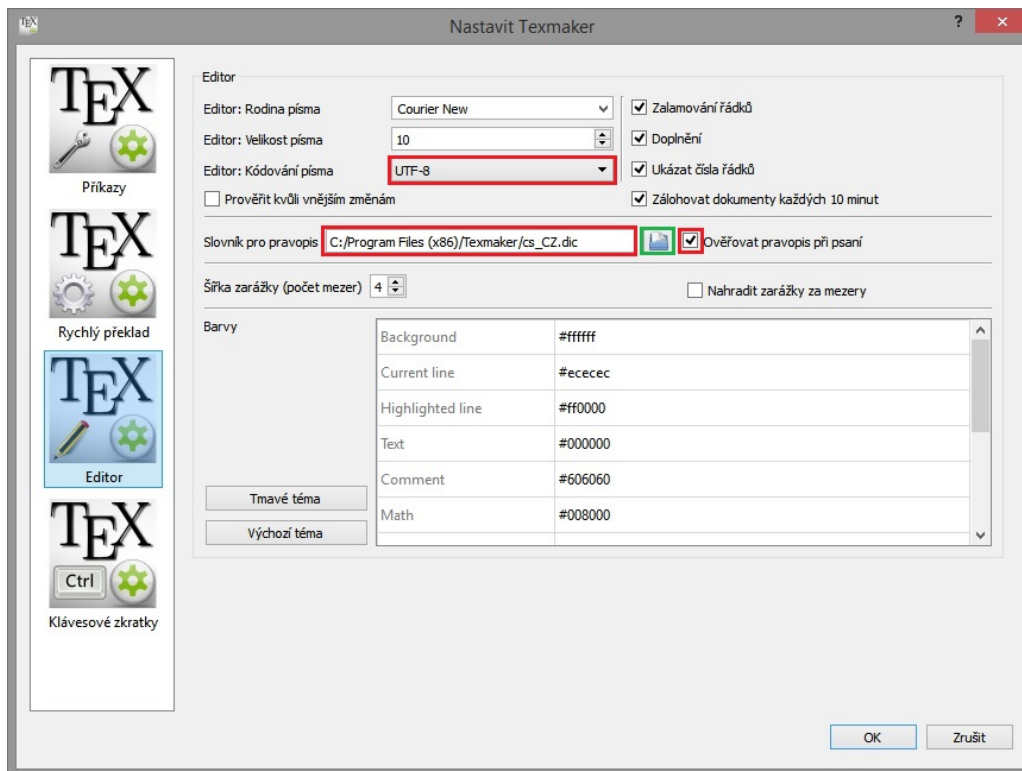
```
pdflatex <jméno_dokumentu>.
```

Protože by ale práce v Notepadu byla velmi nepřehledná, existují pro L^AT_EX editory, které uživateli pomáhají s orientací ve zdrojovém kódu. Těchto editorů je celá řada, jejich porovnání nalezneme na en.wikipedia.org/wiki/Comparison_of_TeX_editors. V tomto návodu se zmíníme o dvou, volbu editoru ale necháváme na čtenáři.

Prvním z nich je TeXworks, který je součástí distribuce MiKTeXu, tedy na Windows jsme ho již nainstalovali společně s MiKTeXem. Jedná se o poměrně jednoduchý editor, jednou z jeho nevýhod je absence kontroly pravopisu (v základní verzi).

Poněkud kvalitnější je pak Texmaker, který lze stáhnout z www.xm1math.net/texmaker/download.html. Tento editor patří mezi nejpoužívanější (funguje na všech zmíněných operačních systémech), a tak ho budeme v tomto návodu používat.

Po instalaci jej spustíme, otevřeme *Volby* → *Nastavit Texmaker* → *Editor* a zkontrolujeme, zda je kódování písma nastavené na UTF-8. Texmaker již obsahuje ověřování pravopisu při psaní. Nastavíme jej tak, že vybereme ve stejné záložce slovník *cs_CZ.dic* a zaškrtneme příslušné políčko, viz Obrázek 1.



Obrázek 1: Nastavení Texmakeru. Zeleně je označené tlačítko, kterým můžeme vybírat mezi nainstalovanými slovníky.

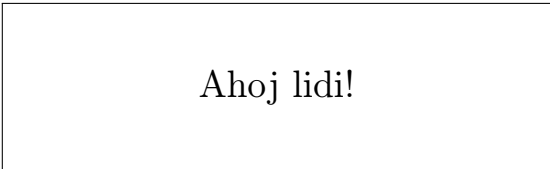
3 Psaní dokumentu

3.1 Základní struktura

Protože se syntax \LaTeX u nejlépe vysvětluje na příkladech, pokusme se rozebrat nejjednodušší možný zdrojový kód (později si ho zkusíme sami napsat a přeložit). Ten vidíme nalevo, napravo pak výsledek po přeložení.

```
\documentclass[10pt]{article}

\begin{document}
{\Large Ahoj lidi!}
\end{document}
```



Jak si můžeme všimnout, ne všechno se vysázelo. Konkrétně se nevysázela ta slova, před kterými se nacházelo zpětné lomítko. To proto, že zpětným lomítkem začínají příkazy, které mají jednu z následujících struktur.

- `\příkaz[volitelný parametr]{povinný parametr}`, kde počet povinných a volitelných parametrů je různý, třeba i nulový
- `{\příkaz parametr}`
- `\begin{prostředí} \end{prostředí}`

3.1.1 Příkaz documentclass

Každý dokument psaný v \LaTeX u musí povinně začínat příkazem `\documentclass[] {typ dokumentu}`, kterým programu říkáme, jaký typ dokumentu se chystáme tvořit. Základní 3 typy jsou `article`, `report` a `book`. Ty se liší především ve formátování záhlaví a zápatí, dále umožňují každý jiné příkazy. Pro tvorbu několikastránkových dokumentů se nejvíc hodí `article`, budeme se tedy věnovat právě jemu.

Mohli jsme si všimnout, že jsou za příkazem hranaté závorky, do kterých lze psát volitelný parametr. Touto formou lze zadat další detaily, jako je velikost písma (např. `10pt`, `12pt`), jaký používáme typ papíru (např. `a4paper`, `letterpaper`) a zda pak chceme tisknout jednostranně či oboustranně (`oneside`, `twoside`). Tyto parametry píšeme všechny do jedné závorky, oddělujeme je čárkami. Nenapišeme-li do hranatých závorek

nic (nebo nenapišeme-li ani tyto závorky), L^AT_EX použije výchozí nastavení. Pro article je výchozím nastavením [letterpaper,10pt,oneside], tedy příkazy \documentclass[letterpaper,10pt,oneside]{article} a \documentclass{article} jsou totožné.

3.1.2 Prostředí document

Dalším povinným příkazem v každém zdrojovém kódu L^AT_EXu je \begin{document} \end{document}. Text, který se nachází mezi begin a end, bude vysázen v prostředí, které je dáno povinným parametrem. V tomto případě je prostředí samotný dokument, tedy to, co bude mezi begin a end, se vysází, naopak cokoli, co napíšeme za \end{document}, bude L^AT_EX ignorovat.

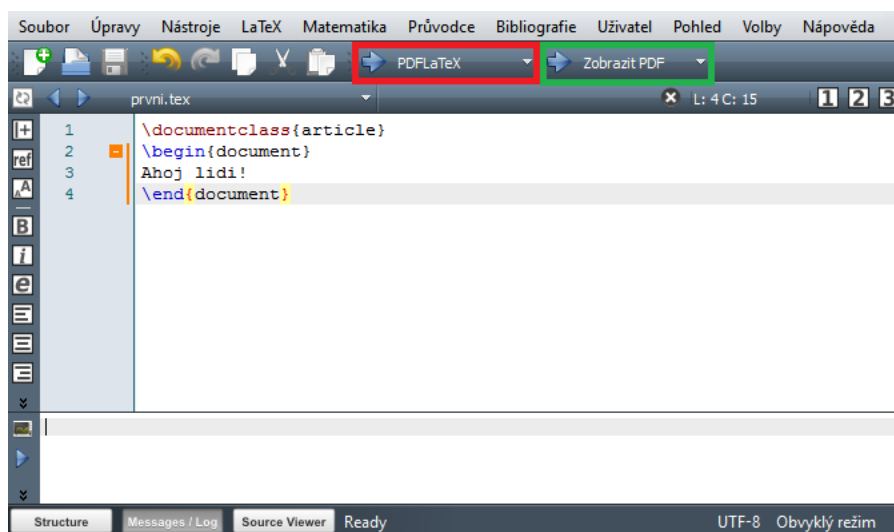
3.1.3 Shrnutí

Každý zdrojový kód v L^AT_EXu je ve tvaru

```
\documentclass[doplňující volby]{typ dokumentu}
Preamble
\begin{document}
Text dokumentu
\end{document}
```

Co je to preamble si povíme později.

Cvičení Nyní si už konečně zkusíte vysázet text. Otevřete si svůj editor, zvolte *Soubor* → *Nový* a uložte si jej na vámi zvolené místo na disku. Do zdrojového kódu napište minimální kód, uvedený v sekci 3.1. Zvolte v nabídce překlad PDFLaTeX a spusťte jej (viz Obr. 2). Na výsledné PDF se pak můžete podívat přímo z prostředí editoru pomocí tlačítka Zobrazit PDF, nebo si ho můžete otevřít ve složce, kam jste uložili zdrojový soubor.



Obrázek 2: Minimální kód v napsaný v prostředí Texmakeru. Červený obdélník značí výběr možnosti překladu, zelený pak možnosti zobrazení. Kliknutím na příslušnou šipku pak spustíte překlad, respektive zobrazíte dokument.

Jakmile se vám to podaří, zkuste změnit text dokumentu na „Schválně, jestli umíš česky!“ a zkuste přeložit. Zobrazíte-li si pak výsledné PDF, pravděpodobně zjistíte, že se nevysázely české znaky. Jak se s tímto problémem vypořádat se dovíte v další kapitole.

3.2 Preamble

Preamble je místo ve zdrojovém kódu nacházející se mezi příkazy \documentclass{} a \begin{document}. V této části můžeme přidávat tzv. balíčky a měnit globální nastavení dokumentu.

3.2.1 Balíčky

Balíčky nám umožní rozšířit funkce \LaTeX u, např. přidají nové příkazy. Pomocí nich taky můžeme naučit \LaTeX načítat a sazet v češtině. Balíček načteme tak, že do preamble vložíme řádek

```
\usepackage[volitelné parametry]{jméno balíčku}
```

Pokud balíček nemáme nainstalovaný, MikTeX ho (pokud jej má v databázi) stáhne z internetu a nainstaluje jej. Základní balíčky jsou uvedeny v následující tabulce.

Jméno balíčku	Volitelný parametr	Funkce
inputenc	utf8	umožní načítání českých znaků
fontenc	IL2	umožní přesnou sazbu českých znaků
babel	czech	přeloží vestavěné formulace do češtiny (např. datum)
indentfirst		odsazení prvního odstavce (v AJ se první odstavec neodsazuje)
graphicx		umožní vkládání obrázků
geometry	požadované okraje	nastaví okraje stránky dle parametrů
amsmath		pokročilá sazba matematiky
amsfonts		pokročilé matematické fonty

Cvičení Prozatím se spokojme s balíčky, které zaručují fungování češtiny. Zkuste tedy vysázet text z předešlého cvičení, a to tak, že přidáme balíčky pro načítání a sazbu českých znaků. Použijte tedy zdrojový kód

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[IL2]{fontenc}
\usepackage[czech]{babel}
\begin{document}
Schválně, jestli umíš česky!
\end{document}
```

Teď už by vše mělo fungovat správně. Tyto balíčky budete pravděpodobně používat ve všech svých česky psaných dokumentech.

Bodovaný úkol 1 (0,25 b.) Stáhněte si zdrojový kód `bodovane_ukoly.tex` a doplňte do něj příkazy tak, aby se text vysázel. Dále v něm do příkazu `\author{}` uveďte své jméno a příjmení.

3.3 Text dokumentu

3.3.1 Mezery a odstavce

Nejprve prozkoumejme, jak se sází mezery.

jedna mezera
spousta mezer

jedna mezera spousta mezer

Jak si můžeme všimnout, \LaTeX chápe libovolný počet mezer za sebou jako jednu mezeru. Rozhodně nám tedy nedovolí vytvářet velké mezery mezi slovy pouze tím, že budeme donekonečna mačkat mezerník. Podobná je situace, snažíme-li se napsat nový odstavec. To provedeme tak, že v kódu vynecháme prázdný řádek. Vynecháme-li jich více, \LaTeX to pochopí jako jeden vynechaný řádek a provede to samé. Toho lze využít pro zpřehlednění zdrojového kódu, ať už pomocí vynechaných řádků či tabulátorů.

Co tedy dělat, pokud si přejeme dostat do textu větší mezeru? Využijeme příkazy `\hspace{velikost}` pro horizontální mezeru a `\vspace{velikost}` pro vertikální. Velikost můžeme zadat v různých jednotkách, například pt, px nebo cm. Např. horizontální mezeru s velikostí 5 cm se sází příkazem `\hspace{5cm}`.

Horizontální `\hspace{1cm}` mezera.
Vertikální
`\vspace{1cm}`
mezera.

Horizontální mezera. Vertikální mezera.

Bodovaný úkol 2 (0,25 b.) Doplňte do zdrojového kódu `bodovane_ukoly.tex` příkazy tak, aby se v druhém odstavci vysázely vertikální a horizontální mezera. Délku obou volte 2 cm.

3.3.2 Rozdělení textu do částí

Jednou z výrazných předností \LaTeX u je možnost dělit text na jednotlivé sekce, které jsou automaticky číslované. Přehlednost zvyšují také různé úrovně těchto sekcí. Lze tvořit různé kapitoly, podkapitoly a podobně. Základní tři příkazy, které budeme používat, jsou: `\section{}`, `\subsection{}` a `\subsubsection{}`, kde povinným parametrem je jméno dané ((pod)pod)sekke. Nemáme-li o číslování zájem, doplníme příkazy o hvězdičku, např. `\section*{}`. Obsah vysázíme příkazem `\tableofcontents`.

```
Takto
\section{Úvod}
se
\subsection{Podúvod}
tvoří
\subsubsection{Prapůvod}
sekce.
```

<p>Takto</p> <p>1 Úvod</p> <p>se</p> <p>1.1 Podúvod</p> <p>tvoří</p> <p>1.1.1 Prapůvod</p> <p>sekce.</p>

Bodovaný úkol 3 (0,25 b.) Doplňte do zdrojového kódu `bodovane_ukoly.tex` příkazy tak, abyste jej rozdělili na sekce a podsekke. Na začátek dokumentu vysázejte obsah.

3.3.3 Pravidlo dvojího překládání

V předchozím úkolu jste nejspíše narazili na problém, že se obsah napoprvé nevysázal správně. Je to proto, že \LaTeX nejdříve sází obsah, přičemž jména sekcí načítá ze zvláštního souboru. Až teprve po vysázení obsahu se vytvoří (aktualizuje) seznam sekcí na základě právě vysázeného dokumentu. Z toho důvodu je pravidlem překládat zdrojový kód dvakrát.

3.3.4 Speciální znaky

Některé znaky mají speciální význam. Napíšeme-li je ve zdrojovém kódu, sice se nevysázejí, ale výsledný text nějakým způsobem ovlivní. Jak jsme si už všimli, takovým znakem je třeba `\`. Speciální znaky jsou:

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

Uvedme si význam dvou z nich. Pomocí `%` píšeme do zdrojového kódu komentáře. Napíšeme-li do řádku `%`, tak cokoli, co následuje, bude \LaTeX ignorovat.

```
text %komentář
```

text

Posledním znakem, který si v tuto chvíli vysvětlíme, je `~`, tzv. tilda. Slouží jako příkaz pro nedělitelnou mezeru. V místě, kde ji použijeme, \LaTeX nezalomí řádek. Používá se z důvodu typografických pravidel, např. lze s její pomocí ošetřit, aby na konci řádku nezůstala jednopísmenná předložka.

Další speciální znaky jsou využívány v tzv. matematickém módu, o kterém si více povíme později. Pokud si přejeme speciální znaky vysázet, musíme použít příkazy, které jsou obvykle ve tvaru `\znak`:

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

\$ % ^ & _ { } ~ \

3.3.5 Další možnosti sazby textu

Velikost písma pro celý dokument nastavujeme pomocí volitelného parametru příkazu `\documentclass`. Přejeme-li si, aby byla část textu psaná jinou velikostí, využijeme příkazy ve tvaru `{\velikost text}`. Závorky musíme opravdu použít takto, jedná se totiž o příkaz ve tvaru `{\příkaz parametr}`. Pokud je vynecháme nebo napíšeme nesprávně příkaz ve tvaru `\velikost {text}`, vše od příkazu dál se bude sázet danou velikostí.

Parametry velikosti jsou:

`tiny` `small` `normalsize` `large` `Large` `LARGE` `huge` `Huge`.

L^AT_EX při výchozím nastavení sází text do bloku. Zarovnání textu na střed, doleva a doprava lze nastavit pomocí příkazů `\begin{center} text \end{center}`, `\begin{flushleft} text \end{flushleft}` a `\begin{flushright} text \end{flushright}`. Příkazem `\textit{text}` píšeme kurzívou, podobně příkazem `\textbf{text}` sázíme tučné písmo a `\underline{text}` podtržené.

Přechod na nový řádek provádíme příkazem `\\`, na novou stránku pak pomocí `\pagebreak`.

Bodovaný úkol 4 (0,5 b.) Vysázejte následující text do dokumentu Bodované úkoly.

Chceme-li *vysázet* logo L^AT_EXu, použijeme příkaz `\LaTeX{}`.

3.3.6 Seznamy

Existují dva základní typy seznamů, číslovaný (`enumerate`) a nečíslovaný (`itemize`). Chovají se jako prostředí, vytvoříme je tedy příkazem `\begin{enumerate}`, resp. `\begin{itemize}` a ukončíme díky `\end{enumerate}`, resp. `\end{itemize}`. Jednotlivé položky seznamu pak musíme uvést příkazem `\item`. Oba typy se dají libovolně kombinovat, jak si ukážeme na příkladě.

```
\begin{enumerate}
  \item první položka
  \begin{itemize}
    \item první bod první položky
    \item druhý bod první položky
  \end{itemize}
  \item druhá položka
\end{enumerate}
```

1. první položka
 - první bod první položky
 - druhý bod první položky
2. druhá položka

Bodovaný úkol 5 (0,25 b.) V dokumentu Bodované úkoly upravte dané položky do seznamu.

3.4 Plovoucí objekty

Některé prvky dokumentu (nejčastěji tabulky a obrázky) se nesází podle umístění ve zdrojovém kódu, ale jejich pozice se určuje až při sazbě. Tyto prvky nazýváme plovoucí objekty. L^AT_EX se je snaží vysázet na vhodnou pozici, zabraňuje tak rozdělení objektu na více stran a podobně.

3.4.1 Tabulky

Tabulky se v L^AT_EXu sázejí poměrně obtížně. Některé editory, včetně Texmakeru, proto mají integrované funkce, které jejich tvorbu usnadňují. Pokusme se ale vytvořit tabulku bez nich. Zdrojový kód si opět vysvětlíme na příkladě.

```
\begin{table}[!ht]
\centering
\begin{tabular}{|l|c|r|}
\hline
Zarovnání vlevo & Na střed & Vpravo \\
\hline
1 & 2 & 3 \\
\hline
\multicolumn{3}{|c|}{Tabulka} \\
\hline
\end{tabular}
\caption{Popisek tabulky} \label{tab:oznaceni}
\end{table}
```

Zarovnání vlevo	Na střed	Vpravo
1	2	3
Tabulka		

Tabulka 1: Popisek tabulky

Příkazem `\begin{table}` se zahajuje prostředí plovoucího objektu. Pomocí volitelného parametru pak máme možnost specifikovat, kam má L^AT_EX tento objekt umístit. Protože s umístěním bývají problémy, je vhodné používat volitelné parametry tohoto příkazu. Ty jsou následující.

- `!` příkáže L^AT_EXu, aby ignoroval některá svá omezení pro umístění plovoucího objektu
- `h` příkazuje umístit objekt „tady“, tedy dle pořadí ve zdrojovém kódu

- `t` umístí objekt nahoru na stránku
- `b` umístí objekt dolů na stránku
- `p` umístí objekt na stránku, kde jsou pouze plovoucí objekty

Tyto parametry můžeme libovolně kombinovat, \LaTeX je pak bude načítat od prvního k poslednímu. V našem případě, kdy využíváme kombinaci `!ht`, se \LaTeX snaží tabulku nejprve umístit na dané místo. Pokud se mu to nepodaří, pokusí se jí umístit nahoru na stránku.

Dále pomocí příkazu `\centering` říkáme, že si přejeme daný objekt zarovnat na střed.

Samotná tabulka začíná až příkazem `\begin{tabular}{}`. Do povinného parametru pak uvádíme, jak mají být zarovnané sloupce tabulky (tedy i kolik jich je) a jak mají být oddělené, tedy `l` znamená sloupec zarovnaný vlevo, `c` na střed a `r` vpravo. Každé písmeno reprezentuje jeden sloupec, tedy pokud bychom chtěli sloupec přidat, musíme přidat i písmeno. `|` značí oddělení sloupců jednou svislou čarou. Následující příkaz `\hline` vysází vodorovnou čáru na vrchu tabulky. Jednotlivé buňky (v řádku) oddělujeme pomocí `&`, přechod na nový řádek zařídíme pomocí `\\`. Přejeme-li si sloučit více buněk dohromady, můžeme využít `\multicolumn{počet sloupců}{zarovnaní}{text}`. Po ukončení tabulky (`\end{tabular}`) můžeme přidat její popis pomocí příkazu `\caption{popisek}`. Tím se tabulka zároveň očísluje. Pokud bychom se pak chtěli na toto číslo odkazovat, je nutné si tabulku nějak označit. To provedeme pomocí `\label{tab:oznaceni}`¹.

Bodovaný úkol 6 (0,5 b.) V dokumentu Bodované úkoly upravte dané položky do tabulky.

3.4.2 Obrázky

Obrázky se do dokumentu implicitně nahrávají ze složky, kde je uložený zdrojový kód, pro přehlednost je ale zvykem je nahrávat z podsložky `img`. Základní syntaxe je podobná jako u tabulek, viz příklad.

```
\begin{figure}[!ht]
\centering
\includegraphics[velikost a další možnosti]{img/jméno_obrázku.přípona}
\caption{obr:popisek obrázku}
\label{označení_obrázku}
\end{figure}
```

Oproti tabulkám tedy používáme prostředí `figure`. Obrázek nahráváme příkazem `\includegraphics[]{}`, který ale nepatří mezi základní příkazy, proto je nutné includovat balíček `graphicx`. Pomocí volitelného parametru můžeme měnit velikost a orientaci obrázku, zatím se ale soustředíme pouze na velikost. Protože je poněkud neohrabané udávat velikost obrázku ve standardních jednotkách, budeme využívat spíše násobky šířky textu. Šířku textu nám dává příkaz `\textwidth`, jeho použití pak vypadá např. takto:

```
\includegraphics[width = 0.7\textwidth]{jméno_obrázku.přípona}.
```

\LaTeX akceptuje obrázky typu JPG, PNG, PDF a EPS (při použití balíčku `epstopdf`). Popisek obrázku a jeho označení jsou stejné jako u tabulek.

3.4.3 Odkazování

Protože se v textu často potřebujeme odkazovat na informace uvedené v tabulkách nebo případně na obrázky, nabízí \LaTeX možnost označit tyto plovoucí objekty příkazem `\label{označení}` a později se na ně odkázat pomocí `\ref{označení}`, resp. `\eqref{označení}` pro rovnice. Tento příkaz vysází číslo daného objektu, použití tedy vypadá takto:

```
...viz informace uvedené v Tabulce \ref{označení}...
```

Pokud se na objekt odkazujeme dříve, než ho pojmenujeme, \LaTeX při prvním překladu vypíše varování, že nenašel pojmenování. Při druhém překladu už ho pak má načtené a pojmenuje objekt správně.

Stejným způsobem se lze odkazovat i na sekce textu. Pojmenujeme je opět pomocí příkazu `\label{}`, který napíšeme do dané sekce. Je zvykem ho psát rovnou za příkaz `\section{}` (u plovoucích objektů za `\caption{}`), neboť příkaz `\label{}` vytvoří odkaz na nejbližší odkazovatelný objekt, který se před ním vyskytl v kódu.

Bodovaný úkol 7 (0,25 b.) Vysázejte do dokumentu Bodované úkoly logo Semináře vědy a techniky, jeho šířku volte jako polovinu šířky textu. Dejte mu popisek a označení. Napište větu, ve které se na něj odkážete pomocí `\ref{}`.

¹Označení uvozujeme typem objektu, abychom se později v označeních vyznali.

3.5 Matematický mód

Největší přednost \LaTeX se skrývá v sazbě matematických vzorců. Nabízí obrovské množství symbolů (které se snadno sázejí), takže není problém během chvilky napsat například

$$\lim_{n \rightarrow \infty} a_n = a \iff (\forall \varepsilon > 0)(\exists n_o \in \mathbb{N})(\forall n > n_o)(|a_n - a| < \varepsilon).$$

Mnoho užitečných příkazů a prostředí nám nabízí balíčky $\mathcal{AMS}\text{-}\text{\LaTeX}$, které budeme dále používat.

3.5.1 Matematická prostředí

Do matematického módu uprostřed textu vstoupíme pomocí symbolu $\$$ a tím ho i ukončíme. Pokud bychom do něj tedy chtěli na tomto místě vstoupit a napsat třeba x^2 , napíšeme $\$x^2\$$. Pokud bychom chtěli napsat matematický vzorec na samostatný řádek, provedeme to příkazem $\$\$vzorec\$\$$.

Vzorec $c^2 = a^2 + b^2$ v textu.

Vzorec na
 $\$c^2 = a^2 + b^2\$$
samostatném řádku.

Vzorec $c^2 = a^2 + b^2$ v textu. Vzorec na

$$c^2 = a^2 + b^2$$

samostatném řádku.

Pokud píšeme matematický výraz do řádku, tak se jej \LaTeX snaží upravit tak, aby příliš nenarušoval formátování textu. Rozdíl vzorce v textu a na samostatném řádku je patrný v následujícím příkladě.

$\$\lim_{n \rightarrow \infty} \sum_{k=1}^n a_k\$$
 $\$ \$\lim_{n \rightarrow \infty} \sum_{k=1}^n a_k \$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n a_k$$
$$\lim_{n \rightarrow \infty} \sum_{k=1}^n a_k$$

Občas se také potřebujeme na daný vzorec odkazovat. K tomu slouží číslované rovnice, které můžeme označit příkazem $\backslash label\{}$. V \LaTeX u je tvoříme pomocí prostředí $equation$, lepší je ale používat $align$ z balíčku $amsmath$. Toto prostředí existuje i ve variantě bez číslování ($align*$), můžeme jím tedy nahradit i $\$ \$$. Jeho syntaxe pak vypadá takto.

$\backslash begin\{align\} \backslash label\{eq:teorie_relativity\}$
 $E=mc^2$
 $\backslash end\{align\}$

$$E = mc^2 \tag{1}$$

3.5.2 Text v matematickém módu

Každé písmeno je v matematickém módu chápáno jako proměnná, a tak se sází odděleně kurzívou. Konstanty je zvykem psát vzpřímeně, k tomu využíváme příkaz $\backslash mathrm\{znak\}$. V případě některých funkcí, které je taky zvykem psát vzpřímeně (např. sinus), můžeme využít přímo příkazu pro danou funkci ($\backslash sin\{x\}$), která vysází text vzpřímeně a s příslušnou mezerou za ní. Číselné množiny se pak značí pomocí $\backslash mathbb\{mnozina\}$. Např. pro množinu reálných čísel $\backslash mathbb\{R\}$, vysází se \mathbb{R} . Přejeme-li si vysázet větší část textu v matematickém módu, užijeme $\backslash text\{text\}$.

3.5.3 Několik příkladů sazby v matematickém módu

Matematických symbolů, které je možné sázet, je obrovské množství. My si uvedeme pouze ty základní příkazy, obsáhlejší seznam (se spoustou dalších užitečných informací) najdete v [3].

Řecká písmena se sází velmi jednoduše, obvykle příkazem $\backslash pismeno$. Například $\backslash alpha$, $\backslash beta$, $\backslash gamma$. Velká řecká písmena vysázíme tak, že v kódu napíšeme první písmeno velké, např. $\backslash Lambda$.

$\backslash alpha$, $\backslash beta$, $\backslash gamma$, $\backslash lambda$, $\backslash Lambda$

$\alpha, \beta, \gamma, \lambda, \Lambda$

Indexy píšeme pomocí speciálních znaků \wedge a $_$. Chceme-li tedy například napsat kubickou rovnici, použijeme následující kód.

`$a_3x^3 + a_2x^2 + a_1x + a_0 = 0$`

$$a_3x^3 + a_2x^2 + a_1x + a_0 = 0$$

Pokud chceme do indexu napsat více než jeden znak, je nutné znaky uzavřít do složených závorek.

`$a^{11}, a^{\{11\}}$`

$$a^{11}, a^{11}$$

Zlomky sázíme pomocí příkazu `\frac{čitatel}{jmenovatel}`. Složený zlomek vytvoříme tak, že do čitatele i jmenovatele zadáme opět příkaz `\frac{}{}`.

`$$\frac{\frac{a}{b}}{\frac{c}{d}}$`

$$\frac{\frac{a}{b}}{\frac{c}{d}}$$

Pokud se pokusíme uzavřít zlomek do závorek, dostáváme následující.

`$(\frac{a}{b})^2$`

$$\left(\frac{a}{b}\right)^2$$

Takovéto závorky se nám jistě nelíbí, z toho důvodu musíme použít příkaz `\left(\frac{a}{b} \right)^2`. Prostředí `\left \right` upraví závorky tak, aby byly stejně velké jako zlomek mezi nimi. Zároveň umí pracovat i s jinými typy závorek.

`$$\left(\frac{a}{b} \right)^2$`

$$\left(\frac{a}{b}\right)^2$$

Velikost závorek můžeme nastavit i manuálně. To se hodí zejména v případě, kdy píšeme několik závorek za sebe.

`$$\Bigg(\bigg(\Big(\big(\frac{a}{b} \big) \Big) \bigg) \Bigg) $`

$$\left(\left(\left(\left(\frac{a}{b}\right)\right)\right)\right)$$

Odmocniny tvoříme příkazem `\sqrt{a}`.

`$$\sqrt{225}=15$`

$$\sqrt{225} = 15$$

Pomocí volitelného parametru pak můžeme dodat stupeň odmocniny.

`$$\sqrt[n]{a}$`

$$\sqrt[n]{a}$$

Balíčky $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ (mimo jiné `amsmath`, `amsfonts`, `amsthm`) nám dávají k dispozici spoustu dalších příkazů, například umožňují jednoduše sázet matice. Ty se sázejí pomocí prostředí `pmatrix` podobně jako tabulky.

`$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} $$$`

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Další příklady Uvedme si nyní několik různých příkladů matematických vzorců. Začněme definicí limity, která uvozovala tuto kapitolu.

```
\usepackage{amsmaths}

$\lim_{n \to \infty} a_n = a \quad \Longleftrightarrow \quad (\forall \varepsilon > 0)
(\exists n_o \in \mathbb{N})(\forall n > n_o)(|a_n - a| < \varepsilon).$
```

$$\lim_{n \rightarrow \infty} a_n = a \iff (\forall \varepsilon > 0) (\exists n_o \in \mathbb{N})(\forall n > n_o)(|a_n - a| < \varepsilon).$$

```

$\int_{-\infty}^{+\infty} \frac{1}{\sigma \sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x = 1$
```

$$\int_{-\infty}^{+\infty} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1$$

```

$\mathrm{sgn}(x) = \left\{ \begin{array}{ll} -1 & x < 0, \\ 0 & x = 0, \\ 1 & x > 0. \end{array} \right.$
```

$$\mathrm{sgn}(x) = \begin{cases} -1 & x < 0, \\ 0 & x = 0, \\ 1 & x > 0. \end{cases}$$

Bodovaný úkol 8 (0,75 b.) Vysázejte do dokumentu Bodované úkoly rovnici

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(c)}{n!} (x - c)^n.$$

3.6 Tvorba vlastních příkazů

V dokumentech často potřebujeme napsat mnohokrát za sebou to samé. To sice můžeme vyřešit pomocí zkratk `Ctrl-C` a `Ctrl-V`, mnohem praktičtější je ale vytvořit si vlastní příkaz, který vysází daný text. \LaTeX nám to umožňuje pomocí příkazu `\newcommand{příkaz}{definice}`, který píšeme do preambule.

```
\newcommand{\svat}{Seminář vědy a techniky}

\svat
```

$$\text{Seminář vědy a techniky}$$

Možnosti `\newcommand` jsou ale mnohem větší, můžeme totiž zadat i parametry našeho příkazu. Syntaxe pak musí být takováto: `\newcommand{příkaz}[počet parametrů][implicitní nastavení]{definice}`. V definici se pak na parametr odkazujeme pomocí `#` a číslem parametru.

```
\newcommand{\integral}[4]
{
  $\int_{\#1}^{\#2} \#3 \mathrm{d}\#4$
}

\integral{1}{2}{f(x)}{x}
```

$$\int_1^2 f(x) dx$$

3.7 Seznam použité literatury

Ve vědeckých člancích je pravidlem uvádět seznam literatury, kterou jsme k tvorbě daného článku využili. V \LaTeX u nám k vytvoření této sekce slouží prostředí `thebibliography`, kterému pomocí povinného parametru předáváme informaci, kolik článků maximálně citujeme. Tento údaj slouží především k formátování seznamu použité literatury, podle počtu číslic v zarovnávané položce v dané vzdálenosti od kraje stránky. Pokud je tedy citovaných článků nejvýše 9, použijeme devítku jako parametr. Je-li jich nejvýše 99, použijeme 99 jako parametr. Prostředí se pak dále chová stejně jako seznam, jen místo příkazu `\item` musíme využít `\bibitem{označení}`. Pomocí tohoto příkazu citovaný článek rovnou označíme. Pokud bychom se na něj chtěli v textu odkazovat, použijeme `\cite{označení}`.

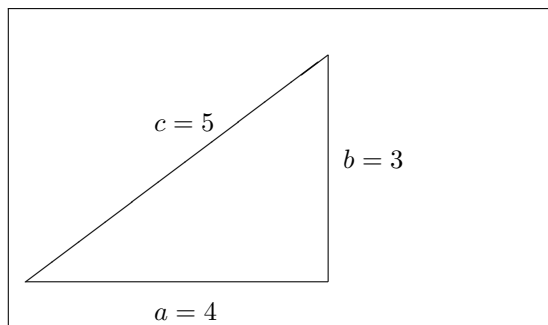
```
\begin{thebibliography}{9}
\bibitem{notsosshort} T. Oetiker, H. Partl, I. Hyna, E. Schlegl,
\textit{The Not So Short Introduction to  $\LaTeX$ , version 5.03},
\url{tobi.oetiker.ch/lshort/lshort.pdf}
\end{thebibliography}
```

3.8 Grafika

V \LaTeX u můžeme vytvářet vlastní jednoduché obrázky. Slouží k tomu prostředí `picture(x,y)`, kde x a y jsou rozměry obrázku. Abychom se nemuseli neustále opakovat s jednotkami, na začátku je nastavíme příkazem `\setlength{\unitlength}{jednotka}`. Pro vložení objektu do obrázku do bodu (x,y) pak využijeme `\put(x,y){objekt}`. Jako objekt může sloužit například text, dostupné jsou ale i geometrické tvary.

- **úsečku** vytvoříme pomocí `\line(x,y){délka}`, kde (x,y) značí směr, kterým budeme úsečku kreslit (x a y musí být v intervalu $\langle -6;6 \rangle$) a délka značí délku průmětu do osy x . V případě úsečky kolmé na osu x pak délka znamená průmět do osy y
- **šipku** získáme velmi podobně jako úsečku, příkazem `\vector(x,y){délka}`, x a y nyní musí být v intervalu $\langle -4;4 \rangle$
- **kružnici** vysázíme díky `\circle{průměr}`, kde průměr nesmí být větší než cca 14 mm. Pro větší průměry je nutné použít balíček `pspicture`.

```
\setlength{\unitlength}{1mm}
\begin{picture}(50,40)
\put(0,5){\line(1,0){40}}
\put(40,5){\line(0,1){30}}
\put(0,5){\line(4,3){40}}
\put(17,0){\textit{a=4}}
\put(42,20){\textit{b=3}}
\put(17,25){\textit{c=5}}
\end{picture}
```



Pro tvorbu složitějších obrázků je vhodné najít balíčky, které se nám pro daný obrázek hodí, protože tvorba v prostředí `picture` je poměrně náročná.

3.9 Prezentace

Pokud napíšeme v L^AT_EXu článek, ke kterému potřebujeme vytvořit prezentaci, můžeme jej rovnou využít a za pomoci balíčku `beamer` transformovat do prezentace, aniž bychom museli obtížně přepisovat článek do Power-Pointu. V tomto návodu se budeme věnovat pouze naprostým základům `beameru`, podrobněji jej probereme až v dalších sériích Semináře vědy a techniky. Jeho použití a základní příkazy si probereme na příkladu.

```
\documentclass{beamer}
\usetheme{Warsaw}
\usepackage[utf8]{inputenc}

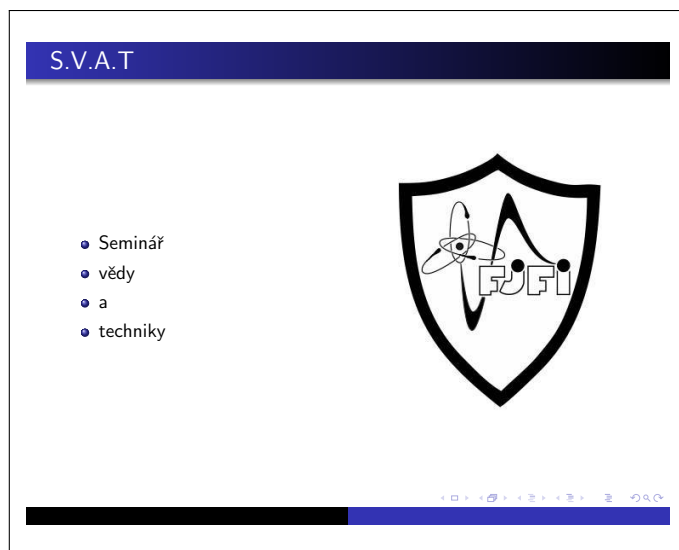
\begin{document}
\begin{frame}
\frametitle{S.V.A.T}
\begin{columns}

\begin{column}{0.5\textwidth}
\begin{itemize}
\item Seminář
\item vědy \pause
\item a
\item techniky
\end{itemize}
\end{column}

\begin{column}{0.5\textwidth}
\includegraphics[width = \textwidth]{img/logo_svat}
\end{column}

\end{columns}
\end{frame}

\end{document}
```



Pomocí `\usetheme{}` nastavujeme schéma prezentace. Ta jsou pojmenována po městech, kromě `Warsaw` tak existují ještě např. schémata `Frankfurt`, `Madrid` a `Berlin`. Slide vytvoříme pomocí prostředí `frame`, jeho název pak příkazem `\frametitle`. Slide můžeme rozdělit na sloupce pomocí prostředí `columns`, jednotlivé sloupce jsou pak prostředí `column`. Velmi užitečný je příkaz `\pause`, kterým nastavujeme postupně se objevující prvky na slidu. V `beameru` pak fungují všechny příkazy, které fungují při tvorbě textového dokumentu.

4 Závěr

S \LaTeX em během svého studia ještě nejspíše strávíte dlouhé večery, možná i noci. Ačkoli se jeho používání zpočátku může jevit dost obtížné, po pár napsaných dokumentech si zvyknete na jeho syntaxi a budete schopni psát rychle i takové vzorce, jejichž tvorba ve Wordu trvá věčnost. Zároveň vaše dokumenty budou vypadat profesionálně, už na první pohled lze rozpoznat texty sázené Wordem a \LaTeX em. Schopnost psát v \LaTeX u je taky, mimo jiné, ceněnou položkou v životopisu. Přeji Vám příjemné chvíle strávené s tímto sázečím programem. A i v těch nejhorších chvílích, kdy na Vás vyskočí 10 errorů, nepropadejte panice!

– autor

Reference

- [1] T. Oetiker, H. Partl, I. Hyna, E. Schlegl, *The Not So Short Introduction to $\text{\LaTeX} 2_{\epsilon}$, version 5.03*, tobi.oetiker.ch/lshort/lshort.pdf
- [2] Matematický korespondenční seminář, *TEX*, <http://mks.mff.cuni.cz/info/tex/>, citováno 27. 9. 2014
- [3] *TEXovnik*, <http://frcatel.fri.uniza.sk/~beerb/latex/latexovnik.pdf>
- [4] P. Satrapa, *TEX pro pragmatiky, verze 1.1*, <http://www.nti.tul.cz/~satrapa/docs/latex/latex-pro-pragmatiky.pdf>
- [5] P. Sopouch, P. Formánek, *TEX v kostce, verze 0.9*, <http://www.it.cas.cz/manual/latex/>
- [6] L. Lomtatidze, R. Plch, *Sázíme v TEXu diplomovou práci z matematiky*, Brno 2003
- [7] V. Paštyková, *Prezentace vlastních prací – formátování textu*, http://svat.fjfi.cvut.cz/files/SVAT_zpracovani_dat_word.pdf