

# Úvod do L<sup>A</sup>T<sub>E</sub>Xu

Kubo

18. septembra 2005

## Abstrakt

Na nasledujúcich pár stránkach nájdete dostatok informácií na to, aby ste boli schopní písať jednoduché dokumenty v L<sup>A</sup>T<sub>E</sub>Xu.

## 1 Prvý výplod

Napíšeme si prvý jednoduchý dokument. Bude vyzerat' asi takto:

```
\documentclass{article}
\usepackage{slovak}
\begin{document}
    Mój prvý výplod v \LaTeX u.
\end{document}
```

Mój prvý výplod v L<sup>A</sup>T<sub>E</sub>Xu.

Text vľavo treba napísať do textového editora (napr. metapad, na CD máte winshell, môj obľúbený editor je gvim – dá sa stiahnuť na [ksp.sk/~kuko/tex/](http://ksp.sk/~kuko/tex/)). Dôležité je, aby ste text uložili ako obyčajný text (plain text, nie wordovský dokument!), teda aby v súbore bolo presne to, čo napíšete (dobré je uložiť súbor s koncovkou `.tex`). Keď teraz z príkazového riadku spustíte `pdfcslatex` a napíšete názov súboru, program by mal vytvoriť súbor s rovnakým názvom, ale koncovkou `.pdf` – to je výsledok nášho snaženia, dokument, ktorý si môžeme pozrieť napr. v Acrobat Reader-i alebo v ghostscript-e.

Ak ide všetko hladko a nikde ste sa nepomýlili, obdržíte výsledok vpravo.

## 2 Kostra

Takmer každý náš dokument bude začínať tromi príkazmi:

1. `\documentclass{article}` si bližšie vysvetlíme a popíšeme neskôr;
2. `\usepackage{slovak}` hovorí T<sub>E</sub>Xu, aby použil balík (package) `slovak` – tu je zariadené to, aby sa L<sup>A</sup>T<sub>E</sub>X správal po slovensky. To znamená, aby fungovali dlžne a mäččene, aby L<sup>A</sup>T<sub>E</sub>X písal „abstrakt“ namiesto „abstract“, „kapitola“ namiesto „chapter“, 18. septembra 2005 namiesto anglického ekvivalentu, a.pod.

3. `\begin{document}` – tu sa začína samotný dokument; ak zvlášť neupozorním na opak, všetko, čo budeme písať, budeme písať až za tento príkaz.

Celý dokument končí príkazom `\end{document}` – všetko za týmto príkazom ako keby tam nebolo.

### 3 Medzery a riadky

Skúsme toho textu vyprodukovať trochu viac (nasledujúci text napíšte medzi `\begin{document}` a `\end{document}`):

Môj prvý dokument v <code>\LaTeX</code> u.	Môj prvý dokument v <code>L<sup>A</sup>T<sub>E</sub>X</code> u. Och,
Och, ani si neviete predstaviť,	ani si neviete predstaviť, ako som si
ako som si <code>\LaTeX</code> obľúbil.	<code>L<sup>A</sup>T<sub>E</sub>X</code> obľúbil.
Úplne ho žeriem.	Úplne ho žeriem.

Ako napovedá názov odseku, na tomto príklade si vysvetlíme dve veci: medzery a riadky. Všimnite si, že napríklad medzi slová „som“ a „si“ (3. riadok) som napísal až 5 medzier, avšak vôbec sa to neodzrkadlilo na výstupe. Či napíšem jednu, alebo viacej medzier, `TEX` to považuje za jednu. Tiež si všimnite si, že v pôvodnom texte som prešiel na nový riadok po slovách `LATEX`u, predstaviť a obľúbil, zatiaľčo vo výsledku `TEX` prešiel na nový riadok tam, kde to bolo podľa neho najlepšie (`TEX` hľadá miesta pre začiatok nového riadku veľmi dômyselne a dokáže dokonca väčšinu slov rozdeliť; keď si pozriete pravý okraj tohto dokumentu, všimnete si, že niektoré slová sú rozdelené do dvoch riadkov; ja som to nerobil – robil to `TEX`; už len týmto prekonáva textové editory ako Word, ktoré sú oveľa novšie; tie text jednoducho useknú tak ako to vyjde a medzislovné medzery rozťahnu, aby bol text zarovnaný na oboch stranách; môžu tak vzniknúť veľmi „voľné“, „riedke“ riadky).

Nový odstavec začneme tak, že v texte vynecháme voľný riadok. Začiatoku nového odstavca potom `TEX` sám odsadí (tzn. na začiatku prvého riadku nechá voľné miesto).

### 4 Príkazy

S príkazmi sme sa v príkladoch stretávali už od začiatku. Tu si povieme, čo presne príkazy sú. Asi ste si stihli všimnúť že obyčajný text sa píše veľmi jednoducho. Zaujímavé veci sa dejú, keď pred nejaké slovo napíšeme opačné lomítko, tzv. backslash. Ak napríklad napíšeme `LaTeX`, na výstup nedostaneme nič iné ako „LaTeX“. Keď však napíšeme `\LaTeX`, dostaneme „L<sup>A</sup>T<sub>E</sub>X“ (podobne `\TeX` vypíše `TEX`). Naozaj si myslíte, že ja mám pojem o čase? Nie je to tak. . . Dnešný dátum (resp. dátum, kedy tento text píšem) viem vypísať jednoducho pomocou príkazu `\today`: 18. septembra 2005. Pred-predchádzajúcu vetu som zakončil tromi bodkami; mohol som napísať jednoducho tri bodky; výsledok (...) však

nie je až taký dobrý, ako keď napíšem `\dots` (`\dots` dá bodky viac od seba; pre porovnanie `:::`).

Jazyk `TeX` sa skladá z veľmi jednoduchých príkazov; `TeX` sa však dokáže učiť nové. Nové príkazy sú vlastne skratky pre viacero príkazov. Ak by sme chceli pekne napísať tri bodky, museli by sme v `TeX`u napísať bodku, potom príkaz pre malú medzeru, potom bodku, opäť malú medzeru a nakoniec tretiu bodku. Ak však `TeX`u vysvetlíme (ako, to sa dozvieme oveľa neskôr), že keď napíšeme `\dots`, chceme, aby napísal pekne tri bodky (oddelené malými medzerami), dokáže sa to naučiť (rovnako ako oveľa zložitejšie príkazy). Takéto príkazy (skratky) sa volajú makrá a `LaTeX` nie je nič viac ako balík makier, kde je naprogramovaných veľa príkazov na ľahké použitie.

Vysvetlili sme si teda, čo je to príkaz (makro); teraz si povieme, čo `TeX` považuje za príkaz. Príkaz (resp. názov makra) začína backslashom (`\`) a pokračuje malými alebo veľkými písmenami (medzi malými a veľkými písmenami je rozdiel, t.j. `LaTeX` a `latex` nie je to isté). V názve makra nesmú byť žiadne iné znaky, resp. znaky iné ako písmená ukončujú názov. Napríklad text `LaTeX-ový TeX` chápe ako príkaz `LaTeX`, za tým pomlčka a za tým ďalšie tri písmená. Podobne `"\TeX"` sa chápe ako úvodzovka, príkaz `TeX` a úvodzovka.

Ako som ale napísal „v `LaTeXu`“? Ak tipujete v `LaTeXu`, tipujete zle. `LaTeXu` by totiž bolo považované za jeden príkaz (a taký v `LaTeXu` neexistuje, ak si ho sami nenaprogramujete, teda program by vám vynadal). `TeX` je však spravený tak, že ignoruje všetky medzery za príkazom; stačí teda napísať v `LaTeX` u (všimnite si príklad v treťom odseku – tam sme to tak naozaj spravili).

Otázka je potom, ako napísať za nejakým príkazom medzeru (napr. „`TeX` je... fajn.“)? Odpoveď opäť dostaneme, keď sa pozrieme do príkladu v treťom odseku. Treba napísať `\_` (lavórik za backslashom označuje medzeru; takto sa medzery, ak sú momentálne dôležité, zviditeľňujú). Napríklad „`TeX` je... fajn.“ som napísal takto: `TeX\_je\dots\_fajn`.

Povedali sme si, že ak za backslashom nasleduje nejaké slovo (reťazec písmen), `TeX` to považuje za príkaz. Čo keď ale za backslashom nenasleduje písmeno? V tom prípade `TeX` považuje za príkaz jediný znak. Napríklad v `TeX`; sa za príkaz považuje `TeX` (postupnosť písmen je končí bodkočiarkou); v `a\_tak` za backslashom nenasleduje písmeno, teda za príkaz sa považuje iba `\_`. Prvý typ príkazu sa označuje ako „control sequence“, druhý typ ako „control symbol“.

## 5 Špeciálne znaky

V `TeX`u sa niektoré znaky používajú pre špeciálne účely (už vieme, že backslashom začínajú príkazy). Ostatné špeciálne znaky sú:

#   \$   %   ^   &   -   {   }   ~

Ich význam si vysvetlíme, keď príde ten správny čas. Ak by sme však tieto znaky chceli napísať, tak mriežku vypíšeme pomocou `\#`, dolár pomocou `\$`, striešku pomocou `\^{}` alebo `\^\_` (strieška sa v texte používa na vysádzanie cirkumflexu: francúzske `ê` som napísal ako `\^e`; iný význam si ukážeme neskôr),

ampersand pomocou `\&` (napr. `rythm\&blues`), podčiariovník pomocou `\_`, zložené zátvorky pomocou `\{` a `\}` a tildu pomocou `\~{}` alebo `\~\_` (španielske `ñ` sa dá vysádzať, ak napíšeme `\~n`).

Keď už sme pri tých špeciálnych znakoch, tak `\copyright` nám vytvorí © a `\S` paragraf (§).

V typografii sa rozlišuje medzi viacerými druhmi pomlčiek. Je treba rozlišovať minimálne spojovník (ako-tak, sem-tam; v `TeX`u je to `ako-tak` a `sem-tam`) a pomlčku (strany 17–22; čo sme napísali ako `17--22`). V `TeX`u existuje ešte dlhšia pomlčka—tá sa používala hlavne vo Viktoriánskej ére (v `TeX`u `---`) teraz sa používa skôr pomlčka oddelená medzerami; stále sa však môže hodiť na uvádzanie priamej reči.

Čo sa týka akcentov, tie sa vo všeobecnosti robia takto: Obyčajný dĺžň ( `\'e`, fr. *accent aigu*) píšeme ako backslash a apostrof; opačný dĺžň píšeme `\`` (fr. *accent grave*, napr. `\'e1\`e` vypíše *é1ève*); `\^` je už spomínaný cirkumflex ( `\^e`, fr. *accent circonflexe*, napr. `cr\^epe` vypíše *crêpe*); `\"a` je v slovenčine prehlasované *a*, nemčina tento zázrak pozná ako *umlaut*; španielske `\~n` (napr. `se\~norita` – *señorita*) sme už tiež spomínali, používa sa tiež na označenie nosového *n* (napr. `at\~asj\~o` – *atásjō* (fr. *výslovnosť attention*); mäkčeň sa dá písať pomocou `\v`; maďarský *umlaut* (skôr dve čiarky ako dve bodky) píšeme `\H o` (ó); fr. *cédille* píšeme `\c c` (ç, `fran\c cais` je *français*); ostatné akcenty nájdete v tabuľke, za úlohu máte zistiť, v akých jazykoch sa používajú a najšť príklady slov, kde sa vyskytujú.

Z ďalších špeciálnych znakov stoja za zmienku nemecké ostré *s*, ktoré dostaneme príkazom `\ss` (ß), poľské *l*, ktoré dostaneme príkazmi `\l` a `\L` (ł a Ł) a písmená *i* a *j* bez bodky (`\i` a `\j` – *i* a *j*). Na čo sú nám *i* a *j* bez bodky? Odpoveďou sú rôzne akcenty. Ak chceme napísať *maître* alebo *naïve*, `\ma\~itre` alebo `na\~ive` by vypísalo aj bodku na *i*, aj ďalší akcent (*maître*, resp. *naïve*); správne treba napísať `ma\~i tre`, resp. `na\~i ve` (pozor, nie `na\~\ive`).

<code>\TeX</code>	<code>TeX</code>	<code>\dots</code>	...	<code>\o</code>	ø	<code>\'e</code>	é	<code>\u g</code>	ğ
<code>\LaTeX</code>	<code>L<sup>A</sup>TeX</code>	<code>\oe</code>	œ	<code>\O</code>	Ø	<code>\`a</code>	à	<code>\va</code>	ă
<code>\dag</code>	†	<code>\OE</code>	Œ	<code>\l</code>	ł	<code>\^e</code>	ê	<code>\H o</code>	ő
<code>\ddag</code>	‡	<code>\ae</code>	æ	<code>\L</code>	Ł	<code>\"a</code>	ä	<code>\t oo</code>	ö
<code>\S</code>	§	<code>\AE</code>	Æ	<code>\ss</code>	ß	<code>\~n</code>	ñ	<code>\c c</code>	ç
<code>\P</code>	¶	<code>\aa</code>	å	<code>\i</code>	ı	<code>\=a</code>	ā	<code>\d a</code>	ą
<code>\copyright</code>	©	<code>\AA</code>	Å	<code>\j</code>	ĵ	<code>\.a</code>	à	<code>\b a</code>	ą

To, že používate počeskoslovenčenú verziu (`cslatex` alebo `pdfcslatex`) a píšete `\usepackage{slovak}` na začiatok dokumentu zabezpečuje, že môžete písať jednoducho `áéíóú` namiesto `\'a\'e\'i\'o\'u`, `čšž` namiesto `\v c\v s\v z`, `ãõ` namiesto `\"a\~o` a pod. (to ešte nehovorím o písmenách ako *đ* a *ł*, kde treba mäkčeň zvlášť upravovať). Tiež tu je definovaný príkaz `\r`, ktorý nad nasledujúcim písmenom vysádza český krúžok (napr. `Císař\v r\v r` uv `pekař\v r` a `pekař\v r\v r` uv `císa\v r`). Tieto znaky si netreba pamätať, stačí vedieť, ako fungujú a kde sa dajú nájsť (napríklad tu; tiež v `TeX Cookbook` či `TeX Reference Card` na CD).

## 6 Fonty

Ak chcete nejaký text *zvýrazniť*, stačí jednoducho `text \emph{zvýrazniť}` (z angl. *emphasize* - *zvýrazniť*). Odporúčam používať výsostne tento spôsob – len aby ste neboli ohúrený, koľko rôznych druhov písma existuje a nemiešali ich všetky do jedného dokumentu. Ak použijete príliš veľa druhov písma, text bude krikľavý. Navyše, podľa môjho skromného názoru tučné písmo dosť kričí; zvážte jeho použitie inde ako v nadpisoch.

Ak predsalen niektorý font potrebujete:

<code>\emph{emphasize}</code>	<i>emphasize</i>	<code>\textrm{roman}</code>	roman
<code>\textit{italic}</code>	<i>italic</i>	<code>\textsl{slanted}</code>	<i>slanted</i>
<code>\textsf{sans serif}</code>	sans serif	<code>\textbf{bold face}</code>	<b>bold face</b>
<code>\texttt{typewriter}</code>	typewriter	<code>\textsc{Small Caps}</code>	SMALL CAPS

Niektoré typy sa dajú dokonca kombinovať, napr. tučnú kurzívu dostaneme ako `\textit{\textbf{abcdefg}}` (***abcdefg***).

Nasledujúci príklad ukazuje, ako sa dá nastaviť veľkosť písma.

<code>{\tiny Text \texttt{(tiny)}}</code>	Text (tiny)
<code>{\scriptsize Text \texttt{(scriptsize)}}</code>	Text (scriptsize)
<code>{\footnotesize Text \texttt{(footnotesize)}}</code>	Text (footnotesize)
<code>{\small Text \texttt{(small)}}</code>	Text (small)
<code>{\normalsize Text \texttt{(normal)}}</code>	Text (normal)
<code>{\large Text \texttt{(large)}}</code>	Text (large)
<code>{\Large Text \texttt{(LARGE)}}</code>	Text (Large)
<code>{\LARGE Text \texttt{(LARGE)}}</code>	Text (LARGE)
<code>{\huge Text \texttt{(huge)}}</code>	Text (huge)
<code>{\Huge Text \texttt{(HUGE)}}</code>	Text (HUGE)

## 7 Štýly dokumentov

Konečne si vysvetlíme, čo je to ten `\documentclass`, čo furt musíme písať na začiatku. Týmto príkazom sa nastavuje štýl dokumentu. Tých štandardných je na výber niekoľko, ale dajú sa vyrobiť aj vlastné štýly. Z tých štandardných poznáme tieto:

`article` na články, prezentácie, krátke správy, ...

`report` na dlhšie správy obsahujúce niekoľko kapitol, krátke knihy, dizertačné práce, ...

`book` na knihy

`letter` na písanie listov (korešpondenciu)

`slide` na slidy

Štýly upravujú celkový vzhľad dokumentu. Nastavujú napríklad okraje; v článku je titul a abstrakt na jednej strane, v reporte a knihe má každý zvlášť stranu; v reporte začína kapitola na novej strane, v knihe na nasledujúcej pravej strane (tak sa to má; v článku kapitoly vôbec nie sú); články a reporty sú jednostranné, knihy obojstranné (rozlišujú tzv. recto (pravé) a verso (ľavé) strany – napr. čísla strán sú na vonkajšom okraji (na pravej strane vpravo, na ľavej vľavo), vonkajší okraj je väčší...). Jednoducho povedané, v rôznych štýloch robia príkazy trochu odlišné veci.

Rôzne veci sa dajú nastaviť (prestaviť). Najlepšie bude, keď si to ukážeme priamo na príkladoch:

```
\documentclass[11pt,a4paper]{article} článok 11-bodovým písmom na a4
\documentclass[12pt,twocolumn]{book} kniha 12-bodovým písmom; text je
    v dvoch stĺpcoch
\documentclass[twoside]{report} report obojstranne
```

Teda 10pt/11pt/12pt nastavuje veľkosť písma; `titlepage/notitlepage` nastavuje, či má byť titul a abstrakt na samostatnej strane, alebo nie; `onecolumn/twocolumn` nastavuje počet stĺpcov; `oneside/twoside` nastavuje, či chceme jednostranný alebo obojstranný výstup; `openright/openany` nastavuje, či sa má kapitola začínať na najbližšej voľnej strane alebo na najbližšej pravej strane.

## 8 Členenie textu

V tomto odseku sa konečne naučíme tie najdôležitejšie príkazy  $\LaTeX$ u. Zatiaľ sme sa totiž zaoberali textom; na začiatku by však mal byť nejaký ten titul, meno autora, prípadne dátum. Samotný text sa väčšinou člení (čím technickejší text, tým viac sa člení ;) na kapitoly, sekcie, podsekcie... Ako teda na to?

Na začiatku som jednoducho napísal `\title{Úvod do \LaTeX u}`, čím som  $\LaTeX$ u oznámil, ako sa toto tu bude volať. Za tým nasledovalo `\author{Kubo}`, čím som  $\LaTeX$ u oznámil svoje ctené meno. Nakoniec som už len napísal príkaz `\maketitle` a  $\LaTeX$  za mňa nastavil písmo, vypísal titul, vyriešil (vertikálne) medzery, sám doplnil dátum výroby (spotreby). Ak by sme chceli iný dátum, stačí nastaviť `\date{24.12.2000}`. Ak nič nenapíšeme, bude nastavené `\date{\today}`, t.j. dnešný dátum; ak žiadny dátum nechceme, nastavíme `\date{}`.

Za titulom som napísal krátky abstrakt (samozrejme, nie je to povinné, ale môže sa vám to zísť; abstrakt slúži k tomu, aby ste sa na pár riadkoch dozvedeli, o čom text je; to sa zide, keď niečo hľadáte, aby ste nemuseli čítať celý text, keď je úplne mimo). Abstrakt dosiahnete jednoducho tak, že napíšete `\begin{abstract}`, za tým napíšete svoj abstrakt a ukončíte `\end{abstract}`.

Po abstrakte nasledoval nadpis prvej sekcie. Kvôli nemu som musel vyfukáť `\section{Prvý výplod}`. Príkaz `\section` nastaví písmo, ktoré sa používa na nadpis sekcie, pred a za nadpis nechá trochu voľného miesta (pred nadpis dá trochu viac). Všimnite si, že som nenapísal žiadne číslo; o to sa tiež stará príkaz

`\section`. Ak chceme sekciu bez čísla, stačí napísať `\section*{Prvý výplod}`. Hviezdička znamená „nedávaj pred to žiadne číslo“. Podobne ako `\section` môžeme použiť `\subsection`, `\subsubsection` pre „podsekcii“ a „podpodsekcii“ (tie som v tomto texte nepoužil, lebo je krátky a necítil som bohvie akú potrebu). Nadpis paragrafu dosiahneme príkazom `\paragraph{Nadpis paragrafu}`.

Tento prístup má viacero výhod. Hneď prvá je, že písmo a medzery okolo sa nastavujú automaticky, nemusíte to robiť ročne; písmo bude pre každý nadpis rovnaké (jednoducho sa nemôžete seknúť). Tiež je zariadené, aby sa nestalo, že nadpis sekcie je na jednej strane a text, ktorý pokračuje až na druhej – neprípustné. Neskôr si vysvetlíme ako sa dajú sekcie označiť, aby sa na ne dalo odkazovať. Do tretice, stačí napísať čarovné slovíčko `\tableofcontents` a  $\LaTeX$  vám sám od seba vyprodukuje obsah:

## Obsah

1 Prvý výplod	1
2 Kostra	1
3 Medzery a riadky	2
4 Príkazy	2
5 Špeciálne znaky	3
6 Fonty	5
7 Štýly dokumentov	5
8 Členenie textu	6

Začiatok tohto konkrétneho textu vyzerá zhruba takto:

```
\documentclass{article}
\usepackage{slovak}
```

```
\begin{document}
\title{Úvod do \LaTeX u}
\author{Kubo}
\maketitle
```

```
\begin{abstract}
Na nasledujúcich pár stránkach nájdete dostatok informácií na
to, aby ste boli schopní písať jednoduché dokumenty v \LaTeX u.
\end{abstract}
```

```
\section{Prvý výplod}
```

Napíšeme si prvý jednoduchý dokument. Bude vyzeráť asi takto: