

**G**uardstone  
*new generation software*

# Příprava dokumentů

**TEX**

(formát L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>)

**Kamil Ekštein**

---

# Obsah

<b>1</b>	<b>Základní struktura dokumentu</b>	<b>2</b>
1.1	Příkaz <code>documentclass</code> . . . . .	2
1.1.1	Přepínače třídy upřesňující sazbu dokumentu . . . . .	3
1.2	Příkaz <code>usepackage</code> . . . . .	4
1.3	Příkazy <code>begin</code> a <code>end</code> — blok . . . . .	4
<b>2</b>	<b>Sazba prostého textu</b>	<b>6</b>
2.1	Řezy písma — fonty . . . . .	6
2.1.1	Skloněné řezy písma — příkaz <code>/</code> . . . . .	7
2.2	Velikost písma . . . . .	8
2.3	Zavádění fontů . . . . .	8
2.3.1	Zvětšení zaváděného fontu — fráze <code>scaled</code> a <code>at</code> . . . . .	9
2.4	Sazba akcentovaných znaků . . . . .	10
2.5	Sazba znaků národních abeced . . . . .	10
2.6	Sazba speciálních znaků . . . . .	10
2.7	Sazba znaků řecké abecedy . . . . .	11
<b>3</b>	<b>Formátování textu</b>	<b>12</b>
3.1	Zlom řádky . . . . .	12
3.1.1	Příkaz <code>newline</code> . . . . .	12
3.1.2	Příkaz <code>\\</code> . . . . .	12
3.1.3	Příkaz <code>linebreak</code> . . . . .	13
3.2	Dělení textu na odstavce . . . . .	13
3.2.1	Příkaz <code>par</code> . . . . .	13

# [1]

## Základní struktura dokumentu

```
\documentclass[velikost-písma, parametry-třídy, ...]{třída-dokumentu}
\usepackage{seznam-stylů}
\begin{document}
...
\end{document}
```

### 1.1 Příkaz `documentclass`

Příkaz `\documentclass` slouží k určení všeobecných charakteristik, tzv. *třídy* vytvářeného dokumentu. Různé třídy dokumentů se pak liší způsobem sazby některých typografických elementů (např. odstavců, kapitol, atd.). Příkaz `\documentclass` také jednoznačně určuje, že se jedná o dokument ve formátu  $\text{\LaTeX}2_{\epsilon}$ , a nikoli v některém ze starších formátů (např.  $\text{\LaTeX}$  2.09).

První — **nepovinný** — parametr příkazu `\documentclass` je seznam přepínačů, které blíže specifikují způsob sazby dokumentu dané třídy. Jako první člen tohoto seznamu se obvykle uvádí základní velikost písma v dokumentu. Udává se v jednotkách *point*<sup>1</sup>, zkráceně *pt*. Dokument v  $\text{\LaTeX}$ u může mít základní velikost písma **10pt**, **11pt** nebo **12pt**. Pokud není základní velikost písma uvedena (parametr nemusí být uveden vůbec, případně je ponechán prázdný), provede  $\text{\TeX}$  sazbu písmem velikosti 10 bodů, tj. běžnou velikostí písma v knihtisku.

Další přepínače mohou například určovat, pro jakou velikost papíru má  $\text{\TeX}$  sazbu optimalizovat. Má-li být dokument vysázen na papír formátu A4, uvedeme přepínač **a4paper**. Jednotlivé přepínače, které je možno v příkazu `\documentclass` použít, se teoreticky mohou různit (zejména tehdy, pokud půjde o uživatelsky definované třídy) podle uvedené třídy — jestliže ale budeme využívat pouze standardní třídy, které  $\text{\LaTeX}2_{\epsilon}$  nabízí, je sada přepínačů jednotná ( $\Rightarrow$  1.1.1).

Druhý — **povinný** — parametr specifikuje třídu dokumentu.  $\text{\LaTeX}$  nabízí celkem čtyři různé standardní<sup>2</sup> třídy dokumentů, které lze použít v závislosti na

<sup>1</sup>základní typografická jednotka, 1 pt  $\doteq$  0,2416 mm

<sup>2</sup>Při dostatečné znalosti makrojazyka  $\text{\TeX}$ u lze vytvářet vlastní *uživatelské třídy* a řídit jimi sazbu svých dokumentů. Například pro tento dokument byla vytvořena třída **refp** — referenční příručka.

tom, jaký je rozsah připravovaného materiálu a jaké tedy bude jeho strukturování (dělení na kapitoly, podkapitoly, atd.). Třída může být uvedena vždy jen jedna, nelze je kombinovat (a ani pro to není žádný rozumný důvod).

❖ **Tabulka 1: Třídy dokumentů**

třída	parametr	specifikace
dopis	<code>letter</code>	typografické parametry dopisu dle národní* normy
článek	<code>article</code>	příprava krátkých dokumentů
zpráva	<code>report</code>	příprava rozsáhlejších dokumentů
kniha	<code>book</code>	příprava (velmi) rozsáhlé publikace

\*Použití třídy `letter` do značné míry ovlivňuje následné zavedení stylu příslušného národního jazyka (v našem případě `czech`) a je to jen logické, protože sazba např. dopisu v americké angličtině se značně liší od požadavků na dopis v češtině.

„Rozsahem“ dokumentu se v kontextu  $\text{\LaTeX}$  míní zejména stupeň jeho strukturování. Například kniha (`book`) se může (ale nemusí) dělit na části (`part`) a kapitoly (`chapter`), zatímco článek (`article`) nanejvýš na sekce (`section`) — o tom ale až později.

### 1.1.1 Přepínače třídy upřesňující sazbu dokumentu

Následující tabulka uvádí přehled nejběžnějších přepínačů (v původní anglické terminologii *class options*), které se mohou vyskytovat v nepovinném parametru příkazu `\documentclass`.

❖ **Tabulka 2: Přepínače třídy — Options**

přepínač	popis
<code>a4paper</code>	sazba stránky na papír formátu A4
<code>a5paper</code>	sazba stránky na papír formátu A5
<code>b5paper</code>	sazba stránky na papír formátu A5
<code>letterpaper</code>	sazba stránky na papír formátu Letter
<code>legalpaper</code>	sazba stránky na papír formátu Legal
<code>executivepaper</code>	sazba stránky na papír formátu Executive
<code>landscape</code>	sazba stránky „na šířku“
<code>oneside</code>	sazba pouze z jedné strany stránky
<code>twoside</code>	sazba z obou stran stránky
<code>onecolumn</code>	sazba textu do jednoho sloupce na stránce
<code>twocolumn</code>	sazba textu do dvou sloupců na stránce
<code>draft</code>	náhled, sazba bez úpravy stránky
<code>final</code>	konečná verze, plná úprava stránky

Pokud není nepovinný parametr příkazu `\documentclass` uveden, nastaví  $\text{\TeX}$  automaticky přepínače tak, jako kdyby měl nepovinný parametr tento tvar:

```
\documentclass[10pt, letterpaper, oneside, onecolumn,
final, openany]{ ... }
```

Z toho je zřejmé, že v našich — českých — podmínkách je přinejmenším nutné uvést přepínač `a4paper`, neboť americký formát Letter má poněkud odlišné rozměry a na stránce formátu A4 by sazba upravená pro formát Letter nevypadala příliš dobře (Letter je kratší a širší než A4).

## 1.2 Příkaz `usepackage`

Příkaz `\usepackage` **nemusí** být v záhlaví dokumentu uveden. Zajišťuje zavedení tzv. *stylů* nebo *balíků*<sup>3</sup>. Styly jsou zdrojové texty makrojazyka  $\TeX$  uložené v souborech, které mají obvykle příponu `.STY`<sup>4</sup>.

Styly dodávají  $\TeX$ u důležité informace o tom, jak provést sazbu dokumentu a většinou také obsahují tzv. *makra*, čili definice nových příkazů zapsané v makrojazyce  $\TeX$ u. Ty styly, které je potřeba k sazbě dokumentu použít (může jich být i více), se uvádí jako parametry příkazu `\usepackage`.

### ▶ Příklad 1: Použití stylů

Je-li třeba vysázet dokument v češtině, je nutné použít styl `czech`, aby  $\TeX$  správně „česky“ dělil slova, odsazoval odstavce a čísloval stránky podle českých typografických zvyklostí, sestavoval obsah, sázel datum a podobně tak, jak to vyžadují normy české typografie. Styl `czech` zavedeme takto:

```
\usepackage{czech}
```

Provádí-li se sazba českého dokumentu na papír formátu A4, bude příkaz pro zavedení stylů vypadat takto:

```
\usepackage{czech, a4}
```

Případně můžeme požadovat, aby  $\TeX$  zcela využil šířku papíru A4 a nenechával okraje tak široké:

```
\usepackage{czech, a4wide}
```

Přehled těch nejběžnějších stylů, jejich účel a způsob využití je uveden v dodatku **Styly**. Množství různých stylů se může lišit podle jednotlivých instalací sázecího systému  $\TeX$ . V případě některých „exotických“ stylů se může docela snadno stát, že v určitých instalacích soubory s jejich definicemi prostě nebudou. Jestliže tedy bude dokument využívat nějaké vysoce specializované styly, je nanejvýš vhodné je dodávat (resp. archivovat) spolu se zdrojovým textem takového dokumentu.

S dobrou znalostí makrojazyka  $\TeX$ u lze jednoduše vytvářet vlastní styly, a tak ještě lépe přizpůsobovat vzhled dokumentu svým představám.

## 1.3 Příkazy `begin` a `end` — blok

$\LaTeX$  — což je vlastně jakási sada tříd dokumentů, balíků a stylů, zapsaná v makrojazyce  $\TeX$ u — velmi důsledně vyžaduje dodržování tzv. *bloků*. Začátek

<sup>3</sup> $\LaTeX$ 2<sub>ε</sub> označuje vstupní soubor pojmem *balík* (package) — termín *styl* pochází a přetrvává z názvosloví starších formátů  $\LaTeX$ u.

<sup>4</sup>Tak je tomu ve většině případů. Tato přípona ovšem není povinná, takže se lze setkat i se styly s jinou příponou (např. `.TEX`).

bloku je označen příkazem `\begin` a konec bloku příkazem `\end`. Bloky mohou být různého typu — to je dáno parametrem obou příkazů.

Celý dokument je z pohledu  $\text{\LaTeX}$  jediný blok typu dokument (`document`), a proto je omezen příkazy `\begin` a `\end` takto:

```
\begin{document}
...
\end{document}
```

Vymezením bloku příkazy `\begin` a `\end` vzniká tzv. *prostředí*. Myšlenka různých prostředí v rámci dokumentu je vlastně prapůvodní důvod vzniku formátu  $\text{\LaTeX}$ .

Každý typografický element (např. tabulka, výčet, rovnice, atd.) má své vlastní prostředí, dané parametrem příkazů `\begin` a `\end`. Rozumí se samo sebou, že parametr uvozujícího příkazu `\begin` musí být stejný jako parametr k němu příslušného ukončujícího příkazu `\end`.

### ▶ **Příklad 2: Sazba tabulky pomocí prostředí `tabular`**

```
\begin{tabular}{|c|c|}
\hline
1 & 2 \\
3 & 4 \\
\hline
\end{tabular}
```

Výsledek takové posloupnosti příkazů  $\text{\TeX}$  vypadá následovně:

1	2
3	4

# [2]

## Sazba prostého textu

Text dokumentu musí ležet mezi oběma příkazy, omezujícími blok typu dokument — tedy v prostředí dokumentu (→ 1.3). Následuje-li text za koncem bloku — za příkazem `\end{document}`, L<sup>A</sup>T<sub>E</sub>X jej zcela ignoruje. Naopak předchází-li text dokumentu uvozující příkaz bloku `\begin{document}`, hlásí L<sup>A</sup>T<sub>E</sub>X chybu.

Formátování textu ve zdrojovém souboru<sup>1</sup> (mezery, tabelátory, atd.) **nemá** vliv na rozvržení textu ve výsledném dokumentu po zpracování T<sub>E</sub>Xem. T<sub>E</sub>X například automaticky vhodně zalamuje řádky, dělí slova, atd. bez ohledu na to, jak vypadá zdrojový text.

### ▶ Příklad 3: Nejjednodušší dokument

```
\documentclass[12pt]{article}
\begin{document}
Vergilius: Šťasten ten, komu se dostalo poznání příčiny věcí.
\end{document}
```

Taková posloupnost příkazů T<sub>E</sub>Xu má za následek vysázení jediné stránky s textem v následující podobě:

Vergilius: Šťasten ten, komu se dostalo poznání příčiny věcí.

## 2.1 Řez písma — fonty

Řez písma, tzv. *font*<sup>2</sup> se volí příslušným příkazem (→ tabulka) změny řezu písma, který má tvar `\xx`. Ona dvě písmena v příkazu jsou vždy mnemotechnickou zkratkou anglického názvu toho kterého řezu.

Po uvedení příkazu provádí T<sub>E</sub>X sazbu zvoleným řezem písma až do okamžiku, kdy je uveden příkaz nastavující jiný řez. To ale nemusí vždy vyhovovat, a proto lze použít příkaz změny řezu i tzv. *lokálně*, což znamená, že je takový příkaz a za ním následující text, který má být zvoleným řezem písma vysázen, uzavřen do složených závorek (→ příklad). Zvolený řez písma pak „platí“ pouze

<sup>1</sup>s příponou `.TEX`

<sup>2</sup>Zde panuje — jako ostatně v mnoha jiných oblastech vědy — terminologický chaos. Podle české typografické terminologie se určitý „tvar“ písma nazývá *řez písma*, výraz *font* lze tedy chápat spíše jako označení souboru s daty příslušného řezu.

na úseku textu omezeného závorkami.

### ❖ Tabulka 3: Základní řezy písma

příkaz	řez písma	české označení	ukázka
<code>\rm</code>	roman	patkové	<i>Žít znamená bojovat.</i>
<code>\sl</code>	slanted roman	skloněné patkové	<i>Žít znamená bojovat.</i>
<code>\it</code>	italic	kurzíva	<i>Žít znamená bojovat.</i>
<code>\tt</code>	typewriter	strojopis	<b>Žít znamená bojovat.</b>
<code>\bf</code>	bold face	tučné patkové	<b>Žít znamená bojovat.</b>
<code>\sf</code>	sans serif	bezpatkové	<i>Žít znamená bojovat.</i>
<code>\sc</code>	small caps	kapitálky	<b>ŽÍT ZNAMENÁ BOJOVAT.</b>
<code>\em*</code>	emphasized	zvýrazněné	<i>Žít znamená bojovat.</i>

\*Příkaz `\em` slouží k nastavení zvýrazněného písma — to však ale není řez v pravém slova smyslu. Je to ve skutečnosti jakýsi přepínač mezi dvěma různými, dostatečně odlišnými řezy písma. Je-li „nezvýrazněný“ text sázen řezem roman, pak je zvýrazněný sázen kurzívou a naopak.

#### ▶ Příklad 4: Sazba textu různými řezy písma

```
\bf Vergilius:  
\it Štasten ten, komu se dostalo poznání \sf příčiny \it věci.
```

**Vergilius:** *Štasten ten, komu se dostalo poznání příčiny věci.*

#### ▶ Příklad 5: Lokální změna řezu písma

```
{\bf Vergilius:}  
\it Štasten ten, komu se dostalo poznání {\sf příčiny} věci.
```

**Vergilius:** *Štasten ten, komu se dostalo poznání příčiny věci.*

Výše uvedené řezy písma jsou v  $\text{T}_{\text{E}}\text{X}$ u běžně k dispozici a  $\text{T}_{\text{E}}\text{X}$  „zná“ příslušné příkazy pro jejich nastavení. To ale **neznamená**, že by jiné řezy písma v  $\text{T}_{\text{E}}\text{X}$ u neexistovaly, že by nebylo možné vysázet dokument nějakým jiným písmem. Jiné řezy písma lze dodatečně zavádět (→ 2.3).

Po spuštění je vždy nastaven řez písma roman (jako kdyby byl na začátku dokumentu uveden příkaz `\rm`).

### 2.1.1 Skloněné řezy písma — příkaz /

Při použití skloněných řezů písma (`\it`, `\sl`) v kombinaci s neskloněnými obvykle dochází k optickému zúžení mezery, která je odděluje (zejména se to projevuje u velkých písmen nebo písmen, která jsou akcentovaná):

```
\it Skloněný font \rm přechází v~neskloněný font.
```

*Skloněný font* přechází v neskloněný font.

Tomuto typografickému „zmetku“ lze předejít užitím příkazu `\/` v místě, kde skloněné písmo přechází v neskloněné (před příslušným příkazem změny řezu):

`\it Skloněný font\/\ \rm přechází v~neskloněný font.`

*Skloněný font* přechází v neskloněný font.

## 2.2 Velikost písma

Příkazy pro změnu velikosti písma se používají stejným způsobem, jako příkazy pro změnu řezu písma. Lze je také použít globálně i lokálně.

❖ **Tabulka 4: Běžné velikosti písma**

příkaz	ukázka
<code>\tiny</code>	velikost tiny
<code>\scriptsize</code>	velikost script
<code>\footnotesize</code>	velikost footnote
<code>\small</code>	velikost small
<code>\normalsize</code>	velikost normal
<code>\large</code>	velikost large
<code>\Large</code>	velikost Large
<code>\LARGE</code>	velikost LARGE
<code>\huge</code>	velikost huge
<code>\Huge</code>	velikost Huge

Tyto velikosti písma  $\TeX$  „zná“, aniž by bylo nutné je jakkoli definovat. Stejně jako v případě řezů písma to však **neznamená**, že by nebylo možné přinutit  $\TeX$  vysázet písmo jinou velikostí (→ 2.3.1).

Po spuštění je vždy nastavena velikost `\normalsize` (jako kdyby byl na začátku dokumentu uveden příkaz `\normalsize`).

## 2.3 Zavádění fontů

*Font* je v kontextu  $\TeX$ u (i jiných typografických systémech) sada všech znaků určitého řezu písma. K tomu, aby bylo možno v  $\TeX$ u určitý font (jiný než některý ze základních) použít, je třeba font nejprve tzv. *zavést* (seznámit  $\TeX$  s jeho parametry). Je nutné, aby existovala jeho tzv. *metrika*<sup>3</sup> a aby existovaly definice jednotlivých znaků fontu v knihovně fontů<sup>4</sup>.

V praxi je obvykle font obsažen v knihovně fontů právě tehdy, je-li k dispozici soubor s jeho metrikou.

Názvy souborů s metrikami vždy představují určitou mnemotechnickou pomůcku — odvozenou od jména, případně vlastností toho kterého fontu — podle

<sup>3</sup>Metriky jsou uloženy v souborech s příponou `.TFM` ( $\TeX$  font metric). Tyto soubory se ve valné většině instalaci systému  $\TeX$  nalézají v adresáři `\TEXINPUT` nebo v některém z jeho případných podadresářů.

<sup>4</sup>soubory s příponou `.FLI` (font library)

níž je při troše cviku možné poznat, o jaký font se jedná.

### ▶ Příklad 6: Zavedení nestandardního fontu

Např. český bezpatkový tučný font o velikosti 10pt má metriku uloženou v souboru `CSSSBX10.TFM` (CS je jasné — to označuje, že je to český font, tj. obsahuje háčkované a čárkované znaky; SS říká, že se jedná o bezpatkové písmo (sans serif); BX označuje písmo tučné (bold extended); 10 je velikost v bodech). Zavedeme ho takto:

```
\font\tucny=csssbox10
\tucny Takto vypadá český desetibodový bezpatkový tučný řez.
```

**Takto vypadá český desetibodový bezpatkový tučný řez.**

Tímto způsobem lze např. provést sazbu **azbukou**. Zavedeme font, ve kterém jsou uloženy znaky ruské abuky (`wncyr10`) a pak zapíšeme text v anglické fonetické transkripci ruštiny:

```
\font\azbuka=wncyr10
\azbuka Rossiya --- samaya boljshaya strana mira.
```

Россия — самая большая страна мира.

### 2.3.1 Zvětšení zaváděného fontu — fráze `scaled` a `at`

1. `\font\přepínač-fontu = metrika scaled zvětšení`
2. `\font\přepínač-fontu = metrika at velikost`

Pomocí fráze `scaled` uvedené za příkazem `font` je možné „přinutit“  $\TeX$ , aby znaky fontu zvětšil nebo zmenšil podle našeho požadavku. Zvětšení 1000 představuje původní velikost fontu, tak jak je definována v souboru s metriku. Např. zvětšení 2000 znamená, že znaky budou 2× větší, než je jejich původní velikost, tzn. 10pt u desetibodového písma.

### ▶ Příklad 7: Zavedení zvětšeného fontu

```
\font\velky=csssbox10 scaled 2400
\velky 2,4x zvětšené bezpatkové tučné písmo.
```

**2,4x zvětšené tučné písmo.**

Chceme-li připravit font pro sazbu písma o velikosti např. 0,5 cm, použijeme frázi `at`, která umožňuje zadat přesně požadovanou výšku znaků řezu:

```
\font\pulcm=csssbox10 at 0.5cm
```

V  $\TeX$ u je k dispozici celá řada jednotek délky, které lze spolu s frází `at`<sup>5</sup> užít (např. `in` — palce, `mm` — milimetry, `pt` — typografické body, `dd` — didoty).

<sup>5</sup>Pochopitelně, že nejen s touto frází. Tyto jednotky lze použít až na drobné výjimky kdekoli.

## 2.4 Sazba akcentovaných znaků

Mnohé instalace  $\TeX$ u jsou pro sazbu akcentovaných znaků národních abeced, tj. znaků ě, š, č, ř, ž, atd., přímo přizpůsobeny; např. instalace  $\text{CST}\TeX$ , která dovede pracovat s oběma nejpoužívanějšími kódy češtiny, s kódem bratří Kamenických i s kódem 852 Slavic (Latin-2) podporovaným velkými výrobci softwaru.

To znamená, že akcentované znaky lze vkládat přímo do zdrojového textu a  $\TeX$  je správně interpretuje a vysází. Chceme-li ale připravit dokument, který nebude obsahovat znaky některého z výše uvedených kódů, nebo naopak bude obsahovat takové znaky, které v těchto českých kódech nejsou podporovány (např. francouzský ç – cédille), je třeba využít příkazů  $\TeX$ u pro sazbu akcentů.

❖ **Tabulka 5: Příkazy pro sazbu akcentů**

příkaz	název akcentu	ukázka	použití (sazba ukázky)
<code>\'znak</code>	accent grave	à, ò	<code>\'a, \'o</code>
<code>\'znak</code>	akut, čárka	á, í	<code>\'a, \'r</code>
<code>\^znak</code>	cirkumflex, vokáň	â, ô	<code>\^a, \^o</code>
<code>\~znak</code>	tilda, vlnovka	ã, ñ	<code>\~a, \~n</code>
<code>\=znak</code>	břevno, pruh	ā, p̄	<code>\=a, \=p</code>
<code>\.znak</code>	tečka	á, ẋ	<code>\.a, \.x</code>
<code>\"znak</code>	umlaut	ä, ü	<code>\"a, \"u</code>
<code>\b{znak}</code>	pruh dole	ī, f̄	<code>\b{i}, \b{f}</code>
<code>\c{znak}</code>	cédille	ç, ẓ	<code>\c{c}, \c{z}</code>
<code>\d{znak}</code>	tečka dole	ł, ṃ	<code>\d{l}, \d{m}</code>
<code>\H{znak}</code>	dvojakut	ǎ, ǒ	<code>\H{a}, \H{o}</code>
<code>\t{znak}</code>	tie-after, vazba	ā, ō	<code>\t{a}, \t{o}</code>
<code>\u{znak}</code>	breve	ă, ș	<code>\u{a}, \u{s}</code>
<code>\v{znak}</code>	háček	ǎ, ě	<code>\v{a}, \v{e}</code>

## 2.5 Sazba znaků národních abeced

Většina běžně používaných znaků národních abeced, které se nevyskytují na klávesnici počítače, se dá jednoduše vysázet pomocí některého z následujících příkazů  $\TeX$ u:

❖ **Tabulka 6: Příkazy pro sazbu znaků národních abeced**

příkaz	znak	příkaz	znak	příkaz	znak	příkaz	znak
<code>\ae</code>	æ	<code>\AE</code>	Æ	<code>\oe</code>	œ	<code>\OE</code>	Œ
<code>\aa</code>	å	<code>\AA</code>	Å	<code>\o</code>	ø	<code>\O</code>	Ø
<code>\l</code>	ł	<code>\L</code>	Ł				

## 2.6 Sazba speciálních znaků

Některé znaky není možné zapsat do zdrojového souboru dokumentu vytvářeného v  $\TeX$ u, neboť  $\TeX$  tyto znaky interpretuje jako znaky řídicí (např. %, &, {, }, atd.). Tyto speciální znaky se dají do dokumentu vložit pomocí příslušných příkazů.

Následující tabulka uvádí přehled příkazů  $\TeX$ u, které umožňují vysázet

některé speciální znaky, jež nelze jednoduše zapsat do zdrojového textu dokumentu:

❖ **Tabulka 7: Příkazy pro sazbu speciální znaků**

příkaz	znak	příkaz	znak	příkaz	znak	příkaz	znak
<code>\copyright</code>	©	<code>\pounds</code>	£	<code>\\$</code>	\$	<code>\%</code>	%
<code>\{</code>	{	<code>\}</code>	}	<code>\_</code>	-	<code>\&amp;</code>	&
<code>\#</code>	#	<code>\S</code>	§	<code>\P</code>	¶		

## 2.7 Sazba znaků řecké abecedy

V základní sadě fontů<sup>6</sup> jsou nadefinované také znaky řecké abecedy – alfabety, protože představují jeden ze základních kamenů sazby matematických vzorců. Proto také lze tyto znaky sázet pouze v tzv. *matematickém prostředí* (→ dále) ohraničeném z obou stran znaky \$ (dolar).

Sazba znaků řecké abecedy se provádí příkazy, které jsou tvořené jménem požadovaného řeckého písmene v anglické transkripci, přičemž je-li příkaz zapsán s počátečním písmenem malým, vysází  $\TeX$  malé řecké písmeno. Je-li naopak příkaz zapsán s velkým počátečním písmenem, vysází se velké řecké písmeno. Některá písmena řecké abecedy se svým tvarem velmi přibližují nebo se zcela shodují s písmeny latinky — pro taková písmena (např. omíkrón)  $\TeX$  příkaz nemá.

❖ **Tabulka 8: Příkazy pro sazbu malých řeckých písmen**

příkaz	znak	příkaz	znak	příkaz	znak	příkaz	znak
alpha	$\alpha$	beta	$\beta$	gamma	$\gamma$	delta	$\delta$
epsilon	$\epsilon$	varepsilon	$\varepsilon$	zeta	$\zeta$	eta	$\eta$
theta	$\theta$	vartheta	$\vartheta$	iota	$\iota$	kappa	$\kappa$
lambda	$\lambda$	mu	$\mu$	nu	$\nu$	xi	$\xi$
pi	$\pi$	varpi	$\varpi$	rho	$\rho$	varrho	$\varrho$
sigma	$\sigma$	varsigma	$\varsigma$	phi	$\phi$	varphi	$\varphi$
chi	$\chi$	psi	$\psi$	omega	$\omega$	tau	$\tau$
upsilon	$\upsilon$						

❖ **Tabulka 9: Příkazy pro sazbu velkých řeckých písmen**

příkaz	znak	příkaz	znak	příkaz	znak	příkaz	znak
Gamma	$\Gamma$	Delta	$\Delta$	Theta	$\Theta$	Lambda	$\Lambda$
Xi	$\Xi$	Pi	$\Pi$	Sigma	$\Sigma$	Upsilon	$\Upsilon$
Phi	$\Phi$	Psi	$\Psi$	Omega	$\Omega$		

<sup>6</sup>Základní sadou rozumíme soubor tzv. *Computer Modern* fontů, které byly vytvořené spolu se systémem  $\TeX$ . V případě přizpůsobených národních instalací  $\TeX$ u se pak základní sadou rozumí příslušné národní mutace *Computer Modern* fontů.

# [3]

---

## Formátování textu

### 3.1 Zlom řádky

#### 3.1.1 Příkaz `newline`

Okamžitý, bezpodmínečný zlom řádky, tj. započítí další sazby až na následující nové řádce, se provádí příkazem `newline`. Veškerý text uvedený za tímto příkazem se vysází na novou řádku:

▶ **Příklad 8: Použití příkazu `newline`**

řádka první`\newline` řádka druhá

řádka první řádka druhá
----------------------------

#### 3.1.2 Příkaz `\\`

Příkaz `\\` pracuje v zásadě stejně jako příkaz `newline`. Rozdíl mezi příkazem `newline` spočívá především v tom, že jej lze použít s nepovinným parametrem, kterým je možno specifikovat, jak velkou *vertikální mezeru* má  $\TeX$  mezi dvě řádky vložit:

▶ **Příklad 9: Použití příkazu `\\`**

Využijeme nepovinného parametru a vložíme mezi řádky mezeru o velikosti 3 milimetry.

řádka první`\\[3mm]`řádka druhá`\\`řádka třetí

řádka první řádka druhá řádka třetí
---

Další podstatnou výhodou tohoto příkazu je fakt, že jej na klávesnici počítače napíšeme o mnoho rychleji než příkaz `newline`.

Ještě existuje jedna možnost použití tohoto příkazu, a to ve tvaru:

řádka první\\\*řádka druhá

Tato varianta říká T<sub>E</sub>Xu, že v místě zlomu řádky **nesmí** být zalomena stránka, tj. že nesmí nastat situace, kdy první řádka bude na jedné stránce a druhá řádka na jiné.

### 3.1.3 Příkaz linebreak

Příkaz `linebreak` pracuje v zásadě stejně jako příkaz `newline`. Rozdíl mezi příkazem `newline` spočívá především v tom, že T<sub>E</sub>X tento příkaz interpreтуje spíše jako doporučení k zalomení řádky. Naléhavost tohoto doporučení je možné stupňovat nepovinným parametrem příkazu. Rozsah parametru je 0 až 4, přičemž hodnota 4 znamená bezpodmínečně zalomit řádku, tj. vlastně příkaz `newline`. Není-li nepovinný parametr uveden, chová se příkaz `linebreak` jako příkaz `newline`.

## 3.2 Dělení textu na odstavce

Rozdělení textu na odstavce se zajistí tak, že se do zdrojového textu dokumentu vloží v místě zamýšleného nového odstavce **prázdná řádka**. T<sub>E</sub>X na tuto prázdnou řádku zareaguje tak, že zalomí předcházející řádku a začátek nové řádky posune vpravo, přičemž velikost odsazení odstavce je dána příslušnou třídou dokumentu, resp. použitými styly.

Jinou možností vytvoření odstavce je použití příkazu `par`.

### 3.2.1 Příkaz par

Příkaz `par` je zcela ekvivalentní použití prázdné řádky ve zdrojovém textu. Smyslem používání tohoto příkazu je zabezpečení optické kompaktnosti zdrojového textu v místech, kde by jí prázdná řádka mohla narušit.