
Základy sazby dokumentů v systému L^AT_EX

Učební texty k semináři

Autor:

Jiří Rybička (Mendelova univerzita v Brně)

Datum:

30. 8. 2017

Obsah

1	Úvod	3
2	Zpracování textů na principu systému T _E X	4
3	Práce se systémem, zdrojový text, příkazy	7
4	Začínáme s dokumentem	10
5	Hladká sazba	13
6	Smíšená sazba	16
7	Odstavcová sazba	19
8	Stránkový design a členění dokumentu	25
9	Svislé zarovnání, tabulky	29
10	Matematická sazba	32
11	Kreslení a vkládání obrázků, práce s barvami	36
12	Literatura k dalšímu studiu	40

Zvládnutí libovolného programového systému určité složitosti s sebou nese především nutnost systematického seznámení se základními principy činnosti a se všemi podstatnými službami, které program nabízí. To také předpokládá určitou časovou investici, která se začne vyplácet až po určité době. Pokud se mezitím ukáže, že služby daného programu nevyhovují potřebám uživatele, lze konstatovat, že tato investice nebyla dobrá.

V případě systémů postavených na principu $\text{T}_{\text{E}}\text{X}$ je situace obdobná, ale ze zkušeností lze odvodit, že byť se časová investice zdá *poněkud* větší než v případě jiných programů obdobné třídy, její návratnost bývá *podstatně* větší. Systém $\text{T}_{\text{E}}\text{X}$ je totiž natolik specifický, že v mnoha ohledech nemá ani v dnešní době významnou konkurenci a jeho zvládnutí tedy přináší unikátní efekty.

Tento stručný text zabývající se $\text{T}_{\text{E}}\text{X}$ ovou nadstavbou $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ nemůže obsáhnout vše, co by bylo vhodné pro praxi vědět. Tvoří však úvod, na jehož základech lze snadno pokračovat v dalším studiu, a to jak z literatury uvedené v závěrečném seznamu, tak částečně i z elektronických zdrojů na internetu nebo dokumentačních textů v použité distribuci.

Základním konceptem tohoto textu je především prezentace principů a případně zásad, na kterých jsou tyto principy postaveny. Nebylo možné zde popsat stovky příkazů, které jsou v systému a jeho komponentách k dispozici, ale jsou zde uvedeny některé ilustrační příklady, na nichž lze prezentované principy vyzkoušet v praxi.

Zpracování textů na principu systému $\text{T}_{\text{E}}\text{X}$

Oblast zpracování textů je bezesporu jednou z nejrozšířenějších aplikací – prakticky každý uživatel počítače musí produkovat texty nejrůznějšího druhu. Proto také vznikla řada počítačových systémů pokrývajících tyto potřeby. Koncepty těchto programů jsou velmi rozdílné, můžeme zde však vysledovat alespoň tři základní směry:

- programy původně modelující psací stroj, vhodné pro přípravu zdrojových textů programů, dokumentů HTML nebo podobných souborů; lze je označit pojmem **programový editor** (příklady: vi, vim, NotePad, PSPad, KEdit, Emacs a mnoho dalších),
- programy původně modelující psací stroj, později doplněné o různé funkce vhodné pro přípravu textových dokumentů (různé typy a řezy písma, práce s odstavci, stránkami a dokumentními celky); označují se pojmem **textový editor** nebo **textový procesor** (příklady: WordPad, Word, OO Writer atd.),
- programy modelující knižní sazbu, nebo alespoň některé její části; pro ně se používá pojem **systém DTP** (Desk Top Publishing – publikování na stole). Typickými představiteli jsou InDesign, QuarkXPress, $\text{T}_{\text{E}}\text{X}$.

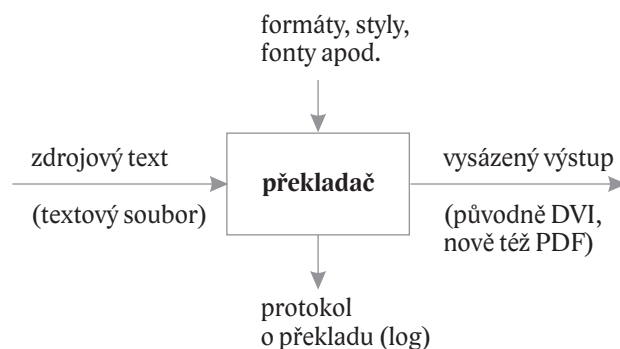
Žádný z programových systémů neumí zcela všechno, je tudíž potřebné pro každou aplikaci vybrat vhodnou alternativu i mezi systémy v rozsahu stejné kategorie. Potřebujeme také vědět alespoň rámcové vymezení funkcí jednotlivých systémů, abychom mohli odpovídající variantu zvolit. Cílem tohoto textu je seznámení se systémy postavenými na principu $\text{T}_{\text{E}}\text{X}$ – jsou to systémy patřící do kategorie počítačových modelů knižní sazby, ale vyznačující se zcela rozdílnými vlastnostmi oproti jiným představitelům této kategorie.

Základní princip systému $\text{T}_{\text{E}}\text{X}$

Původní myšlenkou tvůrce systému $\text{T}_{\text{E}}\text{X}$, Donalda E. Knutha ze Stanfordské univerzity, byl počítačový model práce kvalitního sazeče pro matematickou sazbu. Ta vždycky představovala nejnáročnější část tvorby odborných knih zahrnující jak precizní zvládnutí řemesla, tak i schopnost vnést do sazby cit pro výsledek. Název $\text{T}_{\text{E}}\text{X}$ tak symbolizuje tři počáteční řecká písmena $\tau\epsilon\chi$ slova označujícího technologii a zároveň umění, čte se tedy „tech“.

V době, kdy tento systém vznikal (1982), byly k dispozici zcela jiné počítačové technologie než dnes. Z toho také do značné míry vyplývá i základní princip – autor (sazeč) píše svůj text, do něhož vkládá příkazy pro sazbu. Tento zdroj je následně analyzován a zpracován do podoby výsledné sazby materiálu v jednotlivých stránkách.

Základní princip práce systému je znázorněn na obr. 2.1.



Obrázek 2.1: Princip systému \TeX

Jak je zde vidět, jádro systému spočívá ve zpracování zdrojového textu, k němuž jsou přibrány informace o předdefinovaných příkazech, sázecích stylech, použitých fontech atd., hlavním výsledkem překladu je vysázený materiál ve formátu DVI (DeVice Independent – nezávislý na zařízení), z něhož lze získat formát PostScript, nebo nověji ve formátu PDF. Kromě vysázeného výsledku překladač pořizuje komplexní protokol o překladu, z něhož lze vyčíst všechny podstatné informace, případné chyby nebo nedostatky sazby. Jádro systému je zpravidla obklopeno velkým množstvím dalších podpůrných programů – programové editory pro pořizování zdrojového textu nebo automatizované generátory zdrojových textů, programy pro následné zpracování vysázeného výstupu, programy dodávající potřebné komponenty (styly, fonty, obrázky) apod.

D. Knuth se soustředil hlavně na vysokou kvalitu výstupu. Systém velmi precizně dodržuje všechny detaily práce kvalifikovaného sazeče a umožňuje aplikovat typografické zásady vytvářené po několik staletí. V tomto aspektu nemá systém do dnešní doby prakticky žádnou konkurenci.

Druhou zcela zásadní vlastností je možnost téměř jakkoliv přizpůsobit systém příkazů potřebám uživatele – existuje způsob, jak vytvořit libovolný nový příkaz nebo změnit chování již existujícího příkazu včetně schopnosti změny vnímání jednotlivých symbolů zapisovaných ve zdrojovém textu. I v tomto směru se jedná o unikátní přístup.

Nadstavby, distribuce

Samotný \TeX disponuje desítkami příkazů, které jsou schopny zcela detailně řídit sazbu. Jsou však pro běžného uživatele dost složité a mnohdy také zbytečně podrobné. Proto vznikla řada **nadstaveb** umožňujících vytvořit pro uživatele příznivější prostředí, ale současně využít všech výhod a vynikajících vlastností \TeX u. Jednou z nejznámějších a nejpoužívanějších nadstaveb je \LaTeX , jehož původním autorem je L. Lamport (1984). Poskytuje řadu příkazů výrazně usnadňujících tvorbu běžných dokumentů (zprávy, články, knihy, dopisy, prezentace).

Vývoj však probíhá do dnešní doby a vzniká tak řada dalších rozšíření. Jedním z nejvýraznějších je XeTeX/XeLaTeX, umožňující zpracování textů v mnoha jazycích, sazbu zleva doprava i zprava doleva, využití systémových fontů atd. V poslední době je hodně v popředí rovněž systém ConTeXt představující zcela odlišný pohled na způsob reprezentace dokumentu. Cílem všech rozšíření je zejména přizpůsobení systému současným potřebám a možnostem při zachování základního principu, umožňujícího získávat precizní a typograficky kvalitní dokumenty.

Kolekce několika variant překladačů, mnoha podpůrných programů, dostupných stylů, fontů a dalších potřebných součástí tvoří tzv. **distribuci**. Existuje několik distribucí,

jednou z nejrozšířenějších je T_EXLive. Vyznačuje se nejen rozsáhlostí a komplexností, ale také faktem, že všechny komponenty jsou bezplatně dostupné. Lze ji nainstalovat z distribučního DVD nebo přímo z internetu, z adresy www.texlive.org. Podporovány jsou všechny běžné operační systémy (Windows, Linux, MacOS X). Distribuce je každoročně aktualizována a o její vývoj se stará rozsáhlý tým lidí.

Další známou a rozšířenou distribucí je MikT_EX, specializovanou na operační systémy typu Windows.

Kromě instalace na lokální počítač je možné využít i přístup přes internet. Na adrese <http://tex.mendelu.cz> se nachází webové rozhraní, pomocí něhož lze velmi snadno pořídit zdrojový text a získat hotový vysázený výstup ve formátu PDF nebo PostScript.

V celém dalším textu budeme popisovat práci s překladačem nadstavby L^AT_EX a výstupním formátem bude PDF. Jedná se o alternativu, která je velmi rozšířená a pokrývá velké množství uživatelských potřeb.

Práce se systémem, zdrojový text, příkazy

Po instalaci zvolené distribuce má uživatel k dispozici zvolený překladač, pro pořizování zdrojového textu si může zvolit libovolný programový editor a pro manipulaci s vysázeným výstupem použije vhodný prohlížeč. Práce se systémem spočívá v opakované posloupnosti tří kroků:

1. pořizení nebo úprava zdrojového textu v editoru (jednou z mnoha možností je například volně dostupný editor PSPad),
2. překlad zdrojového textu zvoleným překladačem do zvoleného výstupního formátu (například překladač pdflatex s výstupem do formátu PDF),
3. prohlédnutí výsledku prohlížečem výstupního formátu (v případě formátu PDF lze využít například všeobecně dostupný Acrobat Reader, velmi výhodný je program GSView v OS Windows nebo další alternativy v systémech typu Linux a MacOS).

V porovnání například se systémem InDesign se na první pohled systém \LaTeX odlišuje vlastností „non WYSIWYG“¹. Je však potřebné si uvědomit, že tato vlastnost je velmi diskutabilní – tzv. práce v náhledu (tedy situace, kdy přímo zasahujeme do výsledného tvaru dokumentu) má celou řadu zásadních nevýhod, počínaje velmi problematickou orientací ve výsledné sazbě (řada věcí není na obrazovce počítače dobře vidět), přes nemožnost optimalizovat a automatizovat řadu procesů až po velmi obtížné manipulace se speciálními sazebními prvky – zejména speciálními znaky v hladké sazbě. Jde tedy v případě \LaTeX u o nevýhodu, kterou citelně vnímá většinou jen začátečník, který ještě nepoznal nesmírnou sílu automatizace sazby a rozsáhlé možnosti přesného řízení výstupu.

Typy příkazů, skupiny, prostředí

Klíčovou součástí práce se systémem \LaTeX je tedy pořizování zdrojového textu. V celém dalším textu se budeme věnovat příkazům umístovaným do vstupního textového materiálu, jejich koncepci a možnostem.

Příkaz pro překladač může mít tři tvary:

1. tvar tzv. aktivního znaku – příkaz je tvořen jediným znakem, např. \$, % nebo &;
2. tvar jednoznakového příkazu – příkaz je uvozen obráceným lomítkem, za nímž je jeden neabecední znak, např. \; nebo \=,

¹WYSIWYG = What You See Is What You Get, tj. uživatel pracuje přímo s výsledným tvarem dokumentu.

3. tvar slovního příkazu – příkaz je uvozen obráceným lomítkem, za nímž je posloupnost abecedních znaků, např. `\item` nebo `\footnote`. Za slovním příkazem musí být uveden oddělovač – nějaký neabecední znak nebo mezery nebo konec řádku. Je-li oddělovačem skupina mezer nebo konec řádku, je tento oddělovač pohlcen a nedostane se do výstupu. Toto chování umožňuje efektivně vkládat slovní příkazy do textu, aniž by se na výstupu kumulovaly parazitní mezery.

Příkazy mohou mít **parametry**. Parametry mohou být povinné (obvykle se zapisují do složených závorek `{ ... }`) nebo nepovinné (zapisují se vždy do hranatých závorek `[...]`). Nepovinný parametr lze vynechat, pak se vynechají i jeho závorky. Pořadí povinných a nepovinných parametrů je dáno definicí příkazu. Příklad: `\makebox[2em][r]{kk}`

Příkazy mají rozdílné **rozsahy platnosti**. Některé mají jen místní účinek (projeví se jen v místě, kde jsou uvedeny), některé naopak mají účinek zcela globální (bez ohledu na místo uvedení ovlivní celý dokument). Většina příkazů však platí v určité ohraničené oblasti, která se nazývá **skupina**.

Skupinu můžeme v dokumentu vymežit dvěma způsoby:

1. složenými závorkami `{...}`
2. tzv. **prostředím**, tj. úsekem dokumentu ohraničeným příkazy `\begin{název}... \end{název}`

Skupiny lze do sebe vnořovat podle potřeby, avšak skupiny nelze křížit, tj. nelze vnitřní skupinu uzavřít později než skupinu vnější.

Platnost příkazu s rozsahem skupiny začíná okamžikem uvedení tohoto příkazu a končí zavřením příslušné skupiny.

Vlastní definice

Příkazy, které jsou k dispozici a jsou již předdefinovány, mohou být ještě obohaceny o další, které si uživatel může vytvořit sám. Vlastní definice jsou poměrně velmi jednoduché, definicemi lze vytvořit nové příkazy i změnit definici již existujících.

Jistě může být na první pohled podivné, že se v této chvíli zabýváme tvorbou a modifikací příkazů, aniž byly prezentovány příkazy předdefinované, ale koncept přizpůsobování systému jednotlivým dokumentům a konkrétním uživatelským potřebám vyžaduje, abychom se ruku v ruce věnovali oběma skupinám současně.

Vlastní příkaz včetně parametrů lze vytvořit několika způsoby:

- příkazem `\def` a jeho variantami, tuto metodu lze použít jak pro nové příkazy, tak i pro redefinici existujících;
- příkazem `\newcommand` pro vytvoření nového příkazu nebo příkazem `\renewcommand` pro redefinici již existujícího.

Příklady

Př. 3.1: `\def\pozdrav{Dobrý den}` – vytvoří příkaz `\pozdrav`, jehož uvedením se vypíše text „Dobrý den“.

Př. 3.2: `\newcommand{\pozdrav}{Dobrý den}` – totéž jako v předchozím příkladu.

Př. 3.3: `\def\osloveni#1{Vážený pane #1}` – vytvoří příkaz `\osloveni` s jedním parametrem, do něhož můžeme uvést jméno. Příklad použití: `\osloveni{Nováku}` vyprodukuje text „Vážený pane Nováku“. Maximální počet parametrů může být 9. Místa, kam se v definici vloží text určitého parametru, jsou naznačena znakem `#` a odpovídajícím číslem.

Př. 3.4: `\newcommand{\osloveni}[1]{Vážený pane #1}` – totéž jako v předchozím příkladu.

Př. 3.5: `\renewcommand{\osloveni}[1]{Dobrý den, pane #1}` – změni definici dříve vytvořeného příkazu.

Princip strukturního značkování dokumentů

Chceme-li vidět na papíře nebo na obrazovce hotový dokument, předpokládáme, že není sestaven jen z jednotlivých znaků, ale má také určitou vizuální podobu. Je tedy jasné, že tuto podobu musíme nějak specifikovat a že k tomu slouží určité příkazy.

Zároveň je však taky dobré vědět, *proč* je určitý úsek dokumentu zobrazen právě v takové podobě – vizuální stránka s sebou bezesporu nese i svůj význam, podle něhož se často čtenář snadno orientuje. Lze říct, že z významu textového prvku lze vizuální podobu i odvodit, například víme-li, že určité sousloví je nadpisem první úrovně, dokážeme odvodit, jakou podobu takový nadpis má mít. Opačně to však neplatí – uvidíme-li například v textu sousloví tučným řezem písma, nemusí být jasné, zda se jedná o vyznačení, nadpis nebo významný pojem, definici apod.

Z toho tedy vyplývá, že pro správné formátování dokumentu je důležitější znát význam textového prvku (úseku, oddílu), z něhož se vizuální podoba dá odvodit a dá se taky podle potřeby a kontextu ostatních prvků vhodně měnit.

Příkaz, který nese informaci o významu textového prvku, pracovně nazveme **strukturní značka**. Naopak příkaz, který nese informaci o vizuální podobě, nazveme **vizuální značka**. Vzhledem k tomu, že v systému \LaTeX můžeme vyrobit libovolné vlastní příkazy, můžeme celý dokument sestavit jen pomocí strukturních značek, jejichž vizuální podoba bude obsažena v definicích. Tyto definice můžeme kdykoliv později modifikovat podle potřeby, aniž bychom museli zasahovat do dokumentu.

Příklad strukturně značkování dokumentu

```
\adresa{Vážený pan}{Jaromír Nováček}
      {Velehradská 654/11}{674 11}{Uherský Ostroh}
\vec{Avízo platby}
\osloveni{Nováčku},
\text{s~potěšením Vám můžeme sdělit, že zboží, které jste
nám poskytl, bylo prodáno podle Vašich pokynů, získané
finanční prostředky Vám tedy posíláme na Váš účet.}
\pozdrav{Eva Housková}{obchodní úsek}
\priloha{Doklad o~provedené transakci}
```

Uvedený text nepotřebuje komentář, významy jednotlivých částí jsou srozumitelné na základě vhodné volby názvů příkazů. Zároveň však vidíme, že v dokumentu není jediná zmínka o vzhledu a způsobu formátování. To je soustředěno v definicích jednotlivých příkazů.

Začínáme s dokumentem

Typografické zásady a principy

Po čistě technickém úvodu se nyní soustředíme na hlavní cíl našeho snažení – tedy na správně a esteticky dokonale upravený dokument. Jak na to?

Na tuto zdánlivě jednoduchou otázku už odpovídají generace lidí od vynálezu knihtisku – odpověď leží v mnoha typografických pravidlech, která mají stálou platnost i dnes, kdy je sice technologie úplně jiná, ale způsob vnímání čteného textu člověkem zůstává v podstatě pořád stejný.

Proto se vždy před uvedením technického řešení budeme (alespoň stručně) zabývat typografickými zásadami, na něž pak technické řešení vždy navazuje. Budeme čerpat zejména z normy Základní pravidla sazby (ON 88 2503).

Jedním z nejzákladnějších parametrů dokumentů je **písmo** – jeho typ, řez a stupeň. Celý dokument je zpravidla vytvořen jedním typem písma (kombinace typů jsou z hlediska úpravy často jen zbytečnou komplikací). Tento typ písma nazveme **základní písmo**. Nejčastěji se použije základní, tzv. **obyčejný řez** tohoto typu a také **základní stupeň** (velikost).

Pro různé prvky se pak v dokumentu používají různé řezy a stupně, typ písma však obvykle zůstává pro celý dokument stejný.

Koncept technického řešení

System \LaTeX má možnost nastavit základní písmo dokumentu tak, abychom se o to už pak nemuseli nikde starat. Tento koncept je obohacen ještě o poměrně promyšlené možnosti změn řezů a stupňů tak, aby dokument jako celek vypadal dobře. To je však vyváženo tím, že taková centrální volba nemůže být úplně libovolná, jsou předdefinovány určité možnosti, z nichž je pak možné vybírat.

Realizace v systému \LaTeX

Dokument v systému \LaTeX má svou strukturu – skládá se z úvodního příkazu zavádějícího tzv. třídu dokumentu, za tímto příkazem následuje preambule, kde lze uvádět různé definice globálních příkazů a nastavení a připojovat styly (v terminologii systému se jim říká obecně balíčky), za preambulí následuje prostředí dokumentu. Zjednodušeně můžeme strukturu naznačit tímto schématem:

Struktura dokumentu

```
\documentclass[volby]{třída}
...preambule dokumentu; sem se uvádějí různé definice
\usepackage{styl} ...připojení stylu (balíčku definic)
\begin{document}
...tělo dokumentu; sem se zapisuje text
\end{document}
```

Volby – nepovinný parametr umožňuje upřesnit některá globální nastavení v dané třídě dokumentu: `twoside`, `oneside` nastavuje dvoustranný nebo jednostranný dokument, volby `11pt`, `12pt` nastavují velikost základního písma na 11 nebo 12 anglických typografických bodů. Implicitní stupeň písma je 10 bodů.

Třída dokumentu – definuje typ dokumentu. Předdefinované třídy jsou `article`, `book`, `report`, `letter`, `slides`. Třidu si může vyrobit uživatel podle svých potřeb. Je to soubor s rozšířením `.cls` s definicemi příkazů.

Styl – název stylu (balíčku) s definicemi příkazů. Těchto balíčků je k dispozici nepřeberné množství. Pro českou sazbu potřebujeme alespoň balíček `babel` umožňující definovat všechny jazyky použité v dokumentu. Balíček si může vyrobit uživatel, je to textový soubor s rozšířením `.sty`. V definici třídy dokumentu a v definici stylu dokumentu lze vytvořit všechny strukturní značky potřebné pro zápis daného typu dokumentu.

Základní písmo má implicitně typ odvozený z Knuthova písma Computer Modern. Je to písmo velmi charakteristické pro systémy postavené na principu \TeX , protože v každé distribuci toto písmo vždy bylo k dispozici, včetně národních modifikací. Jedná se o statickou antikvu, která je vhodná pro odborné dokumenty, má k dispozici standardní řezy, bezserifovou variantu, strojopisné písmo a písma pro matematickou sazbu. Je v několika směrech optimalizováno pro sazbu systémem \TeX a jeho nadstavbami.

Není však vhodné všechny dokumenty sázet stejným typem písma. Existuje řada kvalitních písem vhodných pro beletristické texty, pro knihy nejrůznějších zaměření a dokumenty rozdílných účelů. Volby základního písma dokumentu byly tedy rozšířeny o možnost připojit jakékoliv písmo, které máme k dispozici v systému. V následujícím příkladu dokumentu je zvoleno písmo Palatino, základní stupeň 12 bodů:

Příklad dokumentu

```
\documentclass[12pt]{article}
\usepackage[a4paper]{geometry}    ... papír A4
\usepackage{xltextra}            ... XLaTeX, UTF-8
\usepackage{polyglossia}        ... sazba v různých jazycích
\setdefaultlanguage{czech}      ... čeština je výchozí jazyk
\setmainfont{Palatino Linotype} ... základní písmo bude Palatino
\textwidth=16cm                 ... šíře sazby je 16 cm
\begin{document}
Čeština je zapnuta jako první jazyk. Chceme-li přepnout
do slovenštiny, kterou máme rovněž k dispozici, napíšeme
příkaz \selectlanguage{slovak} a následující text
můžeme psát v slovenčine.
\end{document}
```

Po vysázení dostaneme přibližně tento výsledek:

Čeština je zapnuta jako první jazyk. Chceme-li přepnout do slovenštiny, kterou máme rovněž k dispozici, napíšeme příkaz a následující text můžeme psát v slovenčine.

Na procvičení:

1. Vytvořte dokument na papír B5 se základním písmem Bookman velikosti 11 bodů. Napište několik řádků českého textu a k tomu připojte alespoň tři řádky v angličtině. Odpovídajícím způsobem přepněte jazyk, sledujte případné dělení slov.
2. Vyzkoušejte vysázet tentýž dokument písmem Times a sledujte rozdíl mezi prostorem zabraným písmem Bookman a Times.
3. Vysázejte stejný text na papír formátu A5 písmem stupně 10 bodů implicitním typem písma.
4. Experimentujte s šíří sazby – vyzkoušejte například, jak bude vypadat sazba do úzkého sloupce (kolem 5 cm). Sledujte dělení slov v použitých jazycích.

Hladká sazba

Typografické zásady a principy

Pojmem **hladká sazba** je myšlen text vysazený jedním typem, řezem a stupněm písma. Jedná se obvykle o základní text dokumentu. Na první pohled by se zdálo, že hladká sazba vlastně nic není – jen slova poskládaná do vět, odstavců a stránek. V běžném textu se však vyskytuje řada znaků, které v sazbě vyjadřují různé významové a estetické rozdíly a umožňují přesné a rychlé čtení textu. Část těchto znaků je přesně definována v Pravidlech českého pravopisu, některé jsou pak součástí typografických pravidel. Uvedeme alespoň stručný přehled těchto znaků.

- Mezery – jsou nejčastějším znakem vůbec. Můžeme je rozdělit na meziznakové a mezislovní. Meziznakové mezery obvykle systém obhospodařuje sám (slitky, vyrovnání), v ojedinělých případech se s nimi provádějí určité změny (například prostrkání – dnes již nepoužívané vyznačování).

Mezislovní mezery pak mohou být základní, rozšířené a zúžené. Základní mezislovní mezery jsou využívány k zarovnání sazby odstavce do bloku, mají tedy pružnou velikost (správně by se měla její velikost pohybovat od čtvrtiny do poloviny stupně písma). Zúžené a rozšířené mezery jsou pevné.

- Spojovník – vodorovná čárka spojující dvě slova; používá se v souhlase s Pravidly českého pravopisu. Kolem spojovníku nikdy nejsou mezery. Spojovník, který se dostane na konec řádku, musí být opakován na začátku řádku následujícího.
- Pomlčka – významově i graficky se liší od spojovníku. Má dva významy: buď nahrazuje interpunkční čárku, pak se kolem ní vkládají mezery, nebo nahrazuje slova „až“, „až do“, „versus“, pak se sází bez okolních mezer. V tomto druhém významu se nesmí dostat na řádkový zlom – musí se nahradit odpovídajícím slovem.
- Uvozovky – jsou dvojí: počáteční a koncové. V běžných knižních písmech mají počáteční uvozovky tvar devítek a jsou umístěny na základní lince, koncové uvozovky mají tvar šestek a jsou „pověšeny“ na horní dotažnici.
- Stupeň, procento a promile – tyto znaky mají společnou vlastnost v mezerování: jsou-li připojeny bez mezery k předchozímu číslu (slovu), stávají se jeho součástí a tvoří pak jako celek přídavné jméno: např. 10% čteme „desetiprocentní“. Psány s mezerou tvoří s předchozím slovem dva samostatné celky, podstatná jména: např. 10 % čteme „deset procent“.
- Znak pro násobení – je často nesprávně nahrazován písmenem „x“. Přisazuje se bez mezery, tvoří-li vyjádření opakování (např. 5×), ale s okolními mezerami se sází ve významu operace násobení nebo vyjádření rozměrů ploch apod. (např. 10 × 15).

- Další znaky – patří sem výpustek (tři tečky), znak et (&), hvězdička, mečik, paragraf apod. Většinou mají tyto znaky zvláštní mezerování.

Koncept technického řešení

Z technického hlediska se u speciálních znaků řeší způsob jejich vkládání, protože většina z nich se nenachází přímo na klávesnici počítače. Proto existují různé způsoby jejich vyjádření, kterými lze příslušný znak do počítače vložit a také mu nastavit odpovídající chování.

Realizace v systému L^AT_EX

Realizaci speciálních znaků uvádí následující tabulka. V ní jsou uvedeny některé příkazy, jejichž definici systém implicitně nemá, musíme ji provést sami. Jedná se o tyto případy:

- Zúžené mezery – potřebujeme minimálně dvě: jednu čtvrtinovou pro většinu případů a jednu šestinovou pro mezerování výpustku. Definice jsou následující:

```
\def\,{\penalty 10000\hskip 0.25em}
\def\;{\penalty 10000\hskip 0.16667em}
```

- Spojovník s automatickou detekcí konce řádku a opakováním na dalším řádku:

```
\def\spoj{\discretionary{-}{-}{-}}
```

- Pomlčka s automatickou detekcí konce řádku a případným nahrazením původním slovem:

```
\def\az{\discretionary}{\hbox{až\ }}{--}}
```

Znak	Zápis v textu	Příklad	Vysázeno
Mezera	Stisk mezerníku (i několikanásobný)		
Zúžená mezera	\,	10\,mm J.\,K.\,Tyl	10 mm, J. K. Tyl
Nezlomitelná mezera	~ (ručně nebo spuštěním programu pro automatické vložení za předložky)	v~Brně a~v~Blansku	v Brně a v Blansku
Široká mezera	\quad (čtverčik – 1 em) \quadquad (2 em)	1\quad Úvod	1 Úvod
Odstavec	Vynechaný řádek (i vícenásobně)		
Tři tečky	... nebo \dots	zanikl\;\dots	zanikl...
Spojovník	- (přímo z klávesnice)	bude-li	bude-li
Spojovník	\spoj (tento spojovník se při řádkovém zlomu přetahuje na začátek následujícího řádku)	bude\spoj{}li	bude-li
Pomlčka	-- (půlčtverčiková) --- (čtverčiková)	6--12	6-12
Pomlčka (rozsahová)	\az (tato pomlčka se při řádkovém zlomu nahradí slovem „až“)	6\az 12	6-12, 6 až 12
Znak minus	\$\$-\$\$ (v matematickém režimu)	\$\$-10\$\$	-10
Znak násobení	\$\$\times\$\$	\$\$2\times 3\$\$\,mm	2 × 3 mm

(pokračování na další straně)

Znak	Zápis v textu	Příklad	Vysázeno
Stupeň	<code>\$^\circ\$</code>	<code>5\,\$^\circ\$C</code>	5 °C
Paragraf	<code>\S</code> (jen s číslem)	<code>\S\,36</code>	§ 36
Značky #, \$ a &	<code>\#, \\$, \&</code>		
Závorky { a }	<code>\{, \}</code>		
Značky < a >	<code>\$<\$, \$>\$</code>	<code>\$a>b\$</code>	$a > b$
Procento	<code>\%</code>	<code>10\,\%</code>	10 %
Uvozovky	<code>\uv{text}</code>	<code>\uv{Něco}</code>	„Něco“
Uvozovky (úhlové)	<code>\flqq text\frqq</code>	<code>\uvv{Něco}</code>	»Něco«

Na procvičení:

1. Vysázejte speciálními znaky: Vlak na trase Tanvald až Harrachov překonává i za teplot pod minus 15 stupňů Celsia stoupání až dvě celá osm desetin procenta. S padesátiprocentní pravděpodobností bude dvakrát použit v uvozovkách vousatý paragraf deset Studijního a zkušebního řádu.
2. Vyjádřete speciálními znaky, že paní Lída používá příjmení Nová i Pálková, že stožár je natřen žlutými a zelenými pruhy, pracovní doba je od 9 do 17 hodin, v sobotu se hraje utkání mezi Realem Madrid a Barcelonou a že osvětlení bude stát asi 150 amerických dolarů.

Smíšená sazba

Typografické zásady a principy

Ve smíšené sazbě se používají různé typy písma, různé řezy a stupně. Protože ve většině případů budeme celý dokument realizovat v jednom typu písma, jsou nejčastěji potřeba změny řezů a stupňů. Změnu typu písma – písmové rodiny – lze akceptovat například v případě, že použijeme bezserifovou variantu téhož typu.

Proč měníme řez písma?

Nejčastějším důvodem změny řezu je tzv. **vyznačení**. Jedná se o zvětšení důrazu na určitém úseku textu. Základním způsobem vyznačování je použití *kurzívy*. Kurzívní řez má dostatečnou schopnost odlišit při podrobném čtení zdůrazněnou a obyčejnou část textu, ale při pohledu na stránku jako celek nevytváří výrazné plochy.

Silný důraz lze realizovat **tučným řezem**. Ten na sebe strhává pozornost i při letmém pohledu na celou stránku, používá se tedy v případě, že tuto vlastnost považujeme za užitečnou (například nové pojmy v učebnici jsou ideálním případem). Často se tučný řez používá také pro nadpisy nebo jiné důležité a záchytné prvky stránky.

Co když chceme v kurzívním textu ještě něco zdůraznit? Podle typografických pravidel k tomu použijeme opět obyčejný řez.

Chceme-li něco zdůraznit v tučném textu, použijeme k tomu tučnou kurzívu.

Při všech vyznačováním si musíme uvědomit dvě základní věci:

1. vyznačovací řezy mají vždy horší čitelnost než řez obyčejný, vyznačováním tedy *neplýtváme*;
2. každý vyznačovací řez má v textu svou funkci, snažíme se tedy vyznačování vždy realizovat *důsledně jednotně*, aby význam vyznačení byl v celém dokumentu podobný.

Důležitým pravidlem je, že je-li úsek textu vyznačen určitým řezem, je stejným řezem opatřena i drobná interpunkce, která k tomuto textu patří, rovněž vyznačený text v uvozovkách nebo závorkách má stejným řezem vysázeny i obklopující uvozovky a závorky.

Dalšími dostupnými řezy jsou *skloněné písmo* a **KAPITÁLKY**. Skloněný řez má slabou odlišovací schopnost (porovnejte s kurzívou), jeho použití pro vyznačování tedy není doporučeno, lze jej využít pro grafickou úpravu dokumentu. Kapitálky se často používají pro svůj důstojný charakter pro vyznačování jmen osob, významných názvů apod.

Pro vyznačování NEPOUŽÍVÁME podtržení!

Podtržení je rekvizita pevně spojená s psacím strojem a jeho možnostmi, v sazbě prakticky nemá co dělat. Linka v sazbě má zcela jiný účel – nejčastěji oddělení dvou celků. Podtržení navíc silně snižuje čitelnost textu, neboť koliduje s dolními dotahy znaků.

Pro vyznačování rovněž v dnešní době již nepoužíváme tzv. **prostrkání** (zvětšení mezinakových mezer) a už vůbec ne tzv. **proložení**, tj. vkládání mezinakových mezer mezi všechny znaky (to je výhradně strojopisná rekvizita). Prostrkání se v době sazby kovovými písmeny používalo z úsporných důvodů, aby se nemuselo vyrábět písmo jiného řezu. Dnes již tento důvod zcela minul, prostrkání se svou horší čitelností tedy pro běžné vyznačovací účely není vůbec potřeba.

Proč měníme stupeň písma?

Vhodně použité rozdílné stupně písma významně přispívají k dobré orientaci v textu. Logická a střídavá volba stupňů v souladu s běžnými pravidly tedy vede čtenáře a pomáhá mu při čtení. Obecně menším stupněm snižujeme důležitost, větším stupněm zvyšujeme důležitost textu. Všechny stupně písma vztahujeme ke stupni základního písma.

Jaký úsek textu může mít například rozdílný stupeň? Menším stupněm se sázejí poznámky pod čarou, popisky obrázků a tabulek, tabulková data, údaje ve stránkovém záhlaví apod., větší stupně jsou používány hlavně na nadpisy.

Změnu stupně písma **nepoužíváme pro vyznačování!**

O kolik větší nebo menší má příslušný stupeň být? Zde platí pravidlo, že pro dostatečné odlišení jednotlivých stupňů je potřebná změna alespoň o 20 % velikosti. Není tedy přípustné, abychom například nadpis jedné úrovně vysadili písmem 14 bodů a nadpis nižší úrovně písmem 13 bodů.

Koncept technického řešení

Změna řezu nebo stupně je velmi často chápána jako změna materiálu v určité části odstavce nebo určité části dokumentu. Tuto část lze vymezit obvykle skupinou (vyjádřenou svorkami nebo prostředím). Příkazy pro změnu typu, řezu a stupně písma mají tedy platnost v rámci skupiny, jsou však k dispozici i některé varianty, které pracují s materiálem ve svém parametru.

Příkazy, které ovlivňují text do konce skupiny, lze použít prakticky na libovolný úsek textu. Příkazy, které pracují s textem v parametru, lze použít jen na materiál v rámci jednoho odstavce.

Realizace v systému L^AT_EX

Změny typu a řezu písma

<code>\textrm{...}</code>	<i>nebo</i> <code>{\rmfamily ...}</code>	<i>serifový typ písma</i>
<code>\textsf{...}</code>	<i>nebo</i> <code>{\sffamily ...}</code>	<i>bezserifový typ písma</i>
<code>\texttt{...}</code>	<i>nebo</i> <code>{\ttfamily ...}</code>	<i>strojopisný typ písma</i>
<code>\textit{...}</code>	<i>nebo</i> <code>{\itshape ...}</code>	<i>kurzívní řez</i>
<code>\textup{...}</code>	<i>nebo</i> <code>{\upshape ...}</code>	<i>vzpřímený řez (z kurzívy)</i>
<code>\textsc{...}</code>	<i>nebo</i> <code>{\scshape ...}</code>	<i>kapitálky</i>
<code>\textsl{...}</code>	<i>nebo</i> <code>{\slshape ...}</code>	<i>skloněný řez</i>
<code>\textbf{...}</code>	<i>nebo</i> <code>{\bfseries ...}</code>	<i>tučný řez</i>
<code>\textmd{...}</code>	<i>nebo</i> <code>{\mdseries ...}</code>	<i>netučný řez (z tučného)</i>

`\emph{...}` *vyznačení kurzívou, které lze vnořovat*

Příkaz `\emph{}` je poněkud zvláštní. Má pouze tvar s parametrem a chová se takto: Je-li v jeho okolí obyčejný řez, zapne kurzívu. Je-li v okolí kurzívní text, zapne vzpřímený řez. Přesně realizuje typografické pravidlo o vnořování vyznačovacích řezů, lze jej tedy s výhodou použít pro běžné vyznačování, aniž bychom museli „ručně“ kontrolovat správnost nastavení řezů.

Pro nastavení stupně písma jsou k dispozici příkazy, jejichž účinek závisí na nastavení stupně základního písma. Všechny příkazy mají jen jeden tvar, který má účinek v rámci dané skupiny. Automaticky s nastavením příslušného stupně písma se také upraví správné řádkování v odstavci. V přehledu uvedeme pro základní orientaci rovněž bodové velikosti, které se vztahují k implicitnímu nastavení základního písma na 10 bodů:

Změny stupně písma

<code>{\normalsize ...}</code>	<i>základní stupeň 10 bodů (implicitně nastaveno)</i>
<code>{\large ...}</code>	<i>stupeň 12,00 bodů</i>
<code>{\Large ...}</code>	<i>stupeň 14,40 bodů</i>
<code>{\LARGE ...}</code>	<i>stupeň 17,28 bodů</i>
<code>{\huge ...}</code>	<i>stupeň 20,74 bodů</i>
<code>{\Huge ...}</code>	<i>stupeň 24,88 bodů</i>
<code>{\small ...}</code>	<i>stupeň 9,00 bodů</i>
<code>{\footnotesize ...}</code>	<i>stupeň 8,00 bodů</i>
<code>{\scriptsize ...}</code>	<i>stupeň 7,00 bodů</i>
<code>{\tiny ...}</code>	<i>stupeň 5,00 bodů</i>

Kromě uvedených příkazů, které pracují s relativními velikostmi, lze využít příkaz `\fontsize{velikost}{řádkování}\selectfont` a zadat tak libovolnou velikost a libovolné řádkování.

Na procvičení:

1. Vysázejte: Obvod Země vypočítal již ERATOSTHENÉS zhruba 300 let př. n. l., když využil změny **pozorovacího úhlu** hvězdy SÍRIUS ve stejný den *na dvou různých zeměpisných místech* – v ATHÉNÁCH a v ALEXANDRII. S tehdejšími přístroji dospěl *k překvapivě přesnému výsledku*.
2. Nastavte základní písmo na Bookman a vysázejte zvolenou větu ve třech velikostech: 7, 10 a 17 bodů. Pak totéž proveďte s implicitním písmem Computer Modern. V obou případech prostudujte tvary písmen v malém a velkém stupni. Jaký závěr z tohoto pozorování plyne?

Odstavcová sazba

Typografické zásady a principy

Odstavec je považován za základní prvek dokumentu. Jeho tvar a vzájemné vizuální oddělení má zásadní vliv na celkový vzhled a působení dokumentu na čtenáře. Způsob sazby je ovlivněn několika parametry, jejichž vhodné hodnoty uvedeme (viz též obr. 7.1).



Obrázek 7.1: Parametry odstavcové sazby

Zarovnání – celková úprava jednotlivých řádků odstavce. Základní možností je zarovnání **do bloku** (je zarovnaný levý i pravý okraj). Zarovnání do bloku se docíluje rovnoměrnými změnami mezislovních mezer a dělením slov.

Další možnosti zarovnání jsou **na prapor** (zarovnaný je jen levý okraj, nebo zarovnaný je jen pravý okraj), případně zarovnání **na střed**.

Řádkování – vzdálenost účaří dvou po sobě jdoucích řádků. Pro optimální čitelnost se nastavuje zhruba na 120 % stupně písma.

Odstavcová zarážka – mezeru před prvním znakem prvního řádku odstavce. Slouží k vizuálnímu oddělení odstavců. Někdy se nepoužívá tam, kde k vizuálnímu oddělení už dochází – typicky za nadpisem. Její velikost se pohybuje mezi jedním až dvěma stupni písma (pro desetibodové písmo tedy 10–20 bodů), a to podle šíře sazby.

Odsazení – přídatná svislá mezeru mezi odstavci – tvoří jiný způsob vizuálního oddělení odstavců. Nepoužívá se, pokud jsou použity zarážky. Jeho velikost je typicky polovinou řádkování. Pro dosažení identického řádkování na všech stránkách se odsazení nastavuje na hodnotu rovnou řádkování.

Levý a pravý okraj – přídavná mezera mezi okrajem sazby a okrajem odstavce vlevo, resp. vpravo. Používá se u zvláštních odstavců, například u výčtů (levý okraj), citátů (oba okraje) apod. Velikost koresponduje s velikostí zarážky.

Koncept technického řešení

Odstavcová sazba v počítačových programech je řízena algoritmem odstavcového zlomu. Kvalita tohoto algoritmu rozhoduje o celkové kvalitě sazby celého dokumentu. Základním cílem správné sazby je získání pokud možno rovnoměrného rozmístění mezer a celkově upraveného vzhledu. Algoritmy pro odstavcovou sazbu mohou obecně pracovat s každým řádkem zvlášť, nebo pracují s celým odstavcem, případně s dalšími parametry. Často je do odstavcového zlomu zahrnut i problém dělení slov a manipulace s některými speciálními znaky.

Realizace v systému L^AT_EX

V systémech postavených na principu T_EXu je implementován velmi propracovaný algoritmus odstavcového zlomu. V tomto směru nemá systém T_EX mezi ostatními programy pro počítačovou sazbu vážnějšího konkurenta.

Pro zahájení nového odstavce je potřebné vložit prázdný řádek (nebo více prázdných řádků – chovají se jako jeden), nebo použít explicitní příkaz `\par`.

Pro přechod na nový řádek uvnitř odstavce lze použít příkaz `\\`. Tento příkaz má nepovinný parametr, který určuje vzdálenost k dalšímu řádku. Příklad: `\\[20mm]` – příkaz způsobí vertikální mezeru 20 mm.

Odstavce po vyplnění stránky automaticky přecházejí na další stránku. Je hlídáno, aby stránkový zlom nenastal v nevhodném místě odstavce, toto hlídání lze ovlivnit následujícími příkazy:

`\widowpenalty=číslo` – číslo v rozsahu 0–10 000 určuje „sílu“ zákazu tzv. vdovy – východového řádku odstavce samostatně na stránce.

`\clubpenalty=číslo` – číslo v rozsahu 0–10 000 určuje „sílu“ zákazu tzv. sirotka – zarážkového řádku odstavce samostatně na stránce.

`\brokenpenalty=číslo` – číslo v rozsahu 0–10 000 určuje „sílu“ zákazu zlomu stránky za řádkem, který má na konci rozdělené slovo.

Nepodmíněný přechod na novou stránku lze provést příkazy `\newpage` (obyčejný přechod), `\clearpage` (vysazení neumístěných plovoucích objektů a přechod na novou stránku) nebo `\cleardoublepage` (vysazení neumístěných plovoucích objektů a přechod na další lichou stranu).

Implicitně je sazba nastavena se zarovnáním do bloku, odstavcová zarážka je rovna 15 bodům, řádkování 12 bodů, odsazení a oba okraje nulové. Pro změnu implicitního způsobu sazby potřebujeme nastavit některé délkové parametry, k tomu však potřebujeme vědět něco o rekvizitě zvané **délkový registr** a vůbec o možnostech zadání **délky** v systému T_EX.

Délka, jednotky

Všechny délkové rozměry jsou vnitřně zpracovávány jako celočíselné hodnoty. Nedochozí tedy při manipulacích k zaokrouhlovacím chybám. Jednotkou je tzv. **scaled point** (sp), představující vzdálenost $5,35 \cdot 10^{-9}$ m, tedy asi 5 nanometrů. Všechny další jednotky tedy představují jen násobky tohoto rozměru.

Systém \TeX je ojedinělý i v možnostech používat různé měrné systémy. Lze pracovat s anglickou typografickou soustavou, evropskou typografickou soustavou, s palcovými nebo metrickými hodnotami. Použitelné jednotky jsou:

<i>název</i>	<i>zkratka</i>	<i>poznámka</i>
historický anglický typografický bod (point)	pt	0,351 mm
monotypový typografický bod (big point)	bp	0,353 mm
pica	pc	1 pc = 12 pt
evropský Didôtův typografický bod	dd	0,376 mm
cicero	cc	1 cc = 12 dd
palec	in	1 in = 25,4 mm
centimetr	cm	
milimetr	mm	
scaled point	sp	65 536 sp = 1 pt

Kromě těchto absolutních jednotek má systém ještě dvě jednotky **relativní**, jsou odvozeny od aktuálně nastaveného stupně písma. Jsou to:

<i>název</i>	<i>zkratka</i>	<i>poznámka</i>
čtverčík	em	1 em = stupeň písma
půlčtverčík	ex	1 ex = 1/2 em

Pružná délka je taková délka, která má kromě své přirozené velikosti definován i interval stažitelnosti a roztažitelnosti. Pružné délky se uplatňují v místech, kde funguje zarovnávací algoritmus (například odstavcový zlom, vyplňování stránek). Stažitelnost a roztažitelnost se zapisuje klíčovými slovy plus a minus.

Příklady

Př. 7.1: Pružná délka 12 pt plus 1 pt minus 2pt může nabývat minimálního rozměru 11 pt a maximálního 14 pt.

Př. 7.2: Délka 2cm minus 0.35cm může být pouze stažitelná.

Př. 7.3: Délka 2in plus 35bp může být pouze roztažitelná.

Př. 7.4: Speciální délka `\fill` má přirozenou velikost nulovou, ale má nekonečnou roztažitelnost.

Délkové registry

Délku můžeme uložit do **délkového registru**. Některé registry jsou již předdefinovány, můžeme si však vytvořit i vlastní. Registr má název, který má tvar běžného příkazu. Vložení hodnoty do registru se provede příkazem

```
\registr=délka
```

Nastavení délkové hodnoty platí uvnitř skupiny.

Chceme-li hodnotu registru použít, zapíšeme jméno registru, které může být předcházeno reálným koeficientem, kterým se hodnota registru vynásobí.

Registry, které se vztahují k sazbě odstavců:

`\textwidth` – šíře sazby (používá se v preambuli),

`\textheight` – výška sazby (používá se v preambuli),

`\parindent` – odstavcová zarážka,

`\parskip` – odsazení (pružná délka),

`\baselineskip` – řádkování (nelze přímo nastavit, ovlivňuje se nastavením stupně písma nebo koeficientem `\baselinestretch`, který je nutno předefinovat),

`\leftskip` – levý okraj,

`\rightskip` – pravý okraj.

Příklady

Př. 7.5: `\textwidth=125mm` – nastavení šíře sazby na 125 mm.

Př. 7.6: `\parindent=2em` – nastavení zarážky na dva čtverčíky.

Př. 7.7: `\footskip=5\baselineskip` – nastavení vzdálenosti stránkové paty na pětinašobek řádkování.

Př. 7.8: `\parskip=20pt plus 3pt minus 2pt` – nastavení pružného odsazení.

Př. 7.9: Ovlivnění řádkování: `\def\baselinestretch{1.5}\normalsize` – řádkování se zvětší na jedenapůlnásobek normální velikosti. Příkaz pro nastavení stupně písma způsobí uplatnění nově definovaného koeficientu.

Př. 7.10: `\leftskip=2\parindent \rightskip=\leftskip` – nastavení okrajů zúžené sazby (například citát) na dvě zarážky.

Mezery

`\hspace` – v parametru tohoto příkazu je libovolná míra představující požadovanou mezeru. Mezera se vytvoří uvnitř řádku, nikoliv na začátku nebo na konci. Příklad: `\hspace{2\parindent}`

`\hspace*` – příkaz pro mezeru, která se vytvoří vždy. Příklad: `\hspace*{20pt}`

`\vspace` – v parametru tohoto příkazu je libovolná míra představující požadovanou mezeru. Mezera se vytvoří uvnitř stránky, nikoliv na začátku nebo na konci. Příkaz lze použít mezi odstavci, nikoliv uvnitř odstavce. Příklad: `\vspace{5.5\baselineskip}`

`\vspace*` – příkaz pro mezeru, která se vytvoří vždy. Příklad: `\vspace*{\fill}`

`\smallskip` – příkaz pro svislou mezeru o velikosti 1/4 řádkování. Používá se jen mezi odstavci.

`\medskip` – příkaz pro svislou mezeru o velikosti 1/2 řádkování. Používá se jen mezi odstavci.

`\bigskip` – příkaz pro svislou mezeru o velikosti řádkování. Používá se jen mezi odstavci.

Odstavcová prostředí

Pro ovlivnění sazby odstavců se používá několik prostředí upravujících implicitní způsob sazby:

`center` – sazba se zarovnáním na střed.

`flushleft` – sazba se zarovnáním vlevo (pravý prapor).

`flushright` – sazba se zarovnáním vpravo (levý prapor).

`quote` – sazba kratšího citátu; levý a pravý okraj je zaražen implicitně o 25 bodů. Má-li citát více odstavců, je mezi nimi odsazení.

`quotation` – sazba delšího citátu (obvykle více odstavců). Odstavce mají zarážky, jsou bez vertikálních mezer.

`verse` – sazba veršů; přechod na nový verš se provede příkazem `\\`, mezi strofami se vynechá alespoň jeden řádek.

`itemize` – výčet se zarážkovým znakem; každá položka výčtu začíná příkazem `\item`.

`enumerate` – výčet s číslovanými položkami; položky začínají příkazem `\item`.

`description` – popisný výčet; položky začínají příkazem `\item[heslo]`.

`verbatim` – prostředí pro doslovný text. Speciální prostředí, v němž se neinterpretují žádné příkazy a sazba strojopisným písmem přesně odpovídá zápisu ve zdrojovém textu.

Příklady

Př. 7.11: Číslovaný seznam:

Dnešní úkoly:

```
\begin{enumerate}
\item Odnést kabát do čistírny.
\item Koupit 2~kg brambor.
\item Stavit se na poště a~zjistit, zda už nedorazil očekávaný balík.
\end{enumerate}
```

Po vysázení dostaneme přibližně tento vzhled:

Dnešní úkoly: <ol style="list-style-type: none">1. Odnést kabát do čistírny.2. Koupit 2 kg brambor.3. Stavit se na poště a zjistit, zda už nedorazil očekávaný balík.

Př. 7.12: Popisný seznam:

Co jsme se naučili včera ve škole:

```
\begin{description}
```

```
\item [Kódování národních znaků] -- existuje několik přístupů, jak zobrazit národní znaky v počítači.
```

```
\item [Výpočet vlastní informace] -- lze zjistit, kolik informace nese určitá zpráva.
```

```
\item [Datová redundance] -- je to rozdíl mezi maximální možnou entropií a entropií daného kódu.
```

```
\end{description}
```

Co jsme se naučili včera ve škole:

Kódování národních znaků – existuje několik přístupů, jak zobrazit národní znaky v počítači.

Výpočet vlastní informace – lze zjistit, kolik informace nese určitá zpráva.

Datová redundance – je to rozdíl mezi maximální možnou entropií a entropií daného kódu.

Na procvičení:

1. Vhodným nastavením parametrů sazby vysázejte zvolený text do šíře 80 mm se zarážkami jeden čtverčík, nulovým odsazením s roztažitelností 2 body, u jednoho odstavce nastavte levý okraj na 1,5násobek zarážky.
2. Vysázejte dvě strofy Máchova Máje (nebo jiné zvolené básně).
3. Na zvoleném textu skládajícím se z minimálně tří odstavců vyzkoušejte činnost prostředí pro sazbu na střed a na prapor.
4. Vysázejte stručný popis alespoň dvou součástí osobního počítače.

Stránkový design a členění dokumentu

Typografické zásady a principy

Umístění sazby na listu papíru

Stránka se obvykle dělí na **sazební obrazec** (obdélník, v němž se nachází vysázený materiál) a na **okraje** (volný prostor na listu papíru). Sazební obrazec se dále dělí na **záhlaví**, **sazební zrcadlo** a **patu**.

Volba velikosti okrajů je jednou z klasických úloh řešených už od středověku. Lidské vnímání stránky není rovnoměrné – stránka má svoje optické těžiště jinde než geometrické těžiště (přibližně v horní třetině). Rovněž pravá část stránky se jeví těžší než levá. Z toho vyplývá, že okraje nikdy nemohou být všude stejné.

Jednou z užívaných konstrukcí okrajů je poměr malých celých čísel. Označíme-li l levý okraj, h horní okraj, p pravý okraj a d dolní okraj, pak můžeme psát:

$$l : h : p : d = 3 : 4 : 5 : 7$$

Příklady

Př. 8.1: Chceme stanovit okraje a šíři sazby na papíru formátu B5, když je zadán pravý okraj o velikosti 25 mm.

Z poměru okrajů určíme levý okraj: $25 \cdot \frac{3}{5} = 15$ mm. Z rozměrů formátu B5 (176×250) určíme šíři sazby: $176 - (25 + 15) = 136$ mm. Horní a dolní okraj má velikost 20 mm a 35 mm.

Př. 8.2: Máme list papíru formátu A5 a chceme sázet text o šířce 116 mm. Jaké okraje máme nastavit?

Formát A5 má rozměry 148×210 mm. Součet velikostí levého a pravého okraje získáme rozdílem $148 - 116 = 32$ mm. Těchto 32 mm musíme rozdělit v poměru 3 : 5, odsud dostáváme levý okraj o velikosti 12 mm a pravý okraj o velikosti 20 mm. Dále z poměru okrajů snadno vypočteme horní okraj (16 mm) a dolní okraj (28 mm).

Stránkové prvky

Záhlaví stránky slouží především k orientaci v dokumentu. V tom případě se jedná o tzv. **živé záhlaví**, tj. materiál, jehož obsah se mění podle kontextu obsahu stránek. Nejčastěji se do živého záhlaví vkládá číslo stránky, dále v knihách název kapitoly (sekce), ve sbornících název článku a jméno autora, v časopisech název rubriky apod. Číslo stránky se zásadně sází na vnější okraj, aby bylo při listování v dokumentu co nejlépe dosažitelné.

V záhlaví, které má opticky velmi výhodnou pozici, používáme méně nápadné sazební prvky – menší stupeň písma, doplňkový řez. Záhlaví je potřebné oddělit od hlavního

textu vertikální mezerou velikosti nejméně jednoho řádku hlavního textu, velmi často se oddělení zdůrazňuje linkou.

Podobnou úlohu má i stránková pata, avšak vzhledem k jejímu opticky nevýhodnému postavení se obvykle hodí jen pro číslování stránek, k tomu se někdy využívá nápadnějšího řezu (tučné) nebo zvětšeného stupně.

Členění dokumentu

Text dokumentu složený z posloupnosti odstavců se často člení na logické celky, které mají svoje nadpisy. Správný systém nadpisů výrazně přispívá k dobré orientaci v logice dokumentu, proto jeho volba a realizace vyžaduje odpovídající pozornost.

Systémy nadpisů mohou vypadat různě, jedním z nejpoužívanějších je princip jediného rozdílného parametru. Tedy nadpisy mají společný typ písma a společný řez (všechny jsou například tučné), liší se jen stupněm a tomu odpovídajícími vertikálními mezerami. Čím důležitější nadpis (vyšší úroveň), tím větší stupeň.

Druhou možností je stejný typ a stejný stupeň, odlišnost je v řezech – nejdůležitější nadpis je například tučným řezem, druhá úroveň je kurzívou, třetí úroveň obyčejným řezem.

Nadpisy mohou mít číslování. Často se používá desetinná hierarchie (1 – 1.1 – 1.1.1).

Koncept technického řešení

Rozložení stránky na listu papíru má v systémech typu $\text{T}_{\text{E}}\text{X}$ poněkud jiné ovládání. Zatímco v jiných programech se určují přímo velikosti okrajů, z nichž pak vyplyne šíře sazby, v systému $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ se šíře sazby definuje explicitně registrem `\text{textwidth}` a správné umístění na list papíru se dosáhne posunem implicitně daného levého a horního bodu sazby do potřebné pozice pomocí délkových registrů `\hoffset` a `\voffset`. Implicitně daný levý horní bod sazby má vzdálenost od okrajů papíru 1 palec (25,4 mm), to je však ovlivněno ještě nastavením programů pro zpracování a prohlížení vysázeného tvaru. Výšku sazby, která je nastavitelná registrem `\text{textheight}`, je potřebné vypočítat z požadovaných okrajů.

Příklady

Př. 8.3: Stanovení potřebných hodnot pro úpravu horizontálního nastavení sazby: chceme sázet na formát A4 šíří 162 mm s okraji $l = 18$ mm, $p = 30$ mm. Jak nastavíme příslušné registry?

Nastavíme šíři sazby `\text{textwidth}=162mm` a vysadíme dokument. Změřením z výsledného dokumentu zjistíme, že $l = 47$ mm. Nastavíme tedy `\hoffset=-29mm`.

Př. 8.4: Využijeme-li hodnot předchozího příkladu, nastavíme odpovídajícím způsobem i vertikální rozměry. Potřebujeme mít $h = 24$ mm, $d = 42$ mm a z toho vyplývá výška sazby $297 - (24 + 42) = 231$ mm. Nastavíme `\text{textheight}=231mm`. Po sazbě zjistíme, že současný horní okraj je 44 mm, takže bude `\voffset=-20mm`.

Záhlaví a paty dokumentu jsou řešeny speciálními příkazy, které lze podle potřeby libovolně předefinovat, nebo lze využít propracovaného balíčku `fancyhdr`. Lze poměrně jednoduše definovat, jaké informace a odkud se mají do záhlaví (nebo paty) umístit. Vzhledem k programovatelnosti všech součástí je živost záhlaví velmi dobře zajištěna a v tomto směru je systém $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ daleko vpředu mezi sázecími systémy.

Podobně propracovaný je i systém nadpisů. Předdefinovaná instantní varianta systému nadpisů, odpovídající typografickým pravidlům (tučné řezy s rozdílnými stupni), je použitelná na mnoho dokumentů, ale často je potřeba tento systém poněkud přizpůsobit. Opět je to otázka vhodné redefinice nebo využití již připravených nástrojů.

Realizace v systému L^AT_EX

Uvedeme nyní konkrétní příkazy a registry týkající se umístění sazebního obrazce a kompozice stránky, záhlaví a pat, systému nadpisů a s tím související nástroje.

Sazební obrazec a list papíru

Šíře a výška sazby již byla zmíněna v předchozí kapitole, další nástroje jsou:

`\hoffset` – registr pro vodorovný posun sazebního obrazce; záporné hodnoty znamenají posun vlevo.

`\voffset` – registr pro svislý posun sazebního obrazce; záporné hodnoty znamenají posun nahoru.

`\oddsidemargin` – registr pro nastavení levého okraje na lichých stránkách dvoustranného dokumentu.

`\evensidemargin` – registr pro nastavení levého okraje na sudých stránkách dvoustranného dokumentu.

`\headheight` – registr nastavující svislou velikost záhlaví.

`\headsep` – registr nastavující vzdálenost záhlaví od textu.

`\footskip` – registr nastavující vzdálenost účaří paty od textu.

`\marginwidth` – registr nastavující šířku sazby okrajových poznámek (ty se sázejí příkazem `\marginpar{...}`).

`\marginparsep` – registr nastavující mezeru mezi textem a okrajovou poznámkou.

`\usepackage[landscape]{geometry}` – připojení balíčku `geometry` s volbou umožňující sazbu na šířku papíru.

Záhlaví, paty a číslování stránek

`\pagestyle` – volba způsobu zobrazování záhlaví, pat a čísel stránek. Má jeden povinný parametr, jehož předdefinované hodnoty jsou: `empty` – žádné záhlaví, pata ani číslo stránky; `plain` – číslo stránky je uprostřed paty; `headings` – zobrazují se živá záhlaví s čísly stránek na vnějším okraji; `myheadings` – živá záhlaví s mírnou možností uživatelské modifikace.

Po připojení balíčku `fancyhdr` je dostupná volba `\pagestyle{fancy}` a je možné provést řadu nastavení umožňující přesné řízení záhlaví i pat. Více informací v dokumentaci k tomuto balíčku.

Často se u paty stránky (ale v rámci hlavního textu) objevují poznámky pod čarou. Ty jsou dostupné příkazem `\footnote{text}`, který se umístí přímo do textu. Automaticky vytvoří na dané stránce v dolní části čáru a pod ni umístí text v parametru. Do textu vloží odkazovací číslo, které se automaticky přiděluje.

Nadpisy

System nadpisů je dostupný sadou příkazů: `\part`, `\chapter`, (je dostupný v dokumentních třídách `book` a `report`), `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph`. Všechny tyto příkazy mají podobné chování a stejnou formu – jeden nepovinný a jeden povinný parametr. Povinným parametrem je vlastní text nadpisu, nepovinným parametrem může být jiná varianta textu, která se použije do záhlaví a obsahu.

Tato sada příkazů dělá čtyři činnosti: sází vlastní text nadpisu z povinného parametru (příslušným stupněm písma a s příslušnými vertikálními mezerami); nadpis obdařuje odpovídajícím číslem desetinné hierarchie, vytváří položku obsahu a umísťuje text do živého záhlaví (u nadpisů nejvyšších úrovní).

Použijeme-li libovolný z uvedených příkazů s hvězdičkou (například `\section*{...}`), pak tento příkaz dělá pouze jedinou činnost – sazbu příslušného nadpisu. Nadpis je bez čísla a žádné další akce nejsou prováděny. Tato varianta nemá nepovinný parametr.

Pokud bychom potřebovali nečíselný nadpis (například úroveň sekce) vložit do obsahu, použijeme příkaz:

```
\addcontentsline{toc}{section}{text}
```

Vložení materiálu do živého záhlaví lze zařídit příkazy `\markboth{levá}{pravá}` a `\markright{pravá}`.

Do místa, kde chceme mít obsah dokumentu, vložíme příkaz `\tableofcontents`.

Příklady

Př. 8.5: Nadpis úrovně sekce s variantním textem v obsahu:

```
\section[Integrované obvody TTL]{Vybrané integrované obvody  
bipolární technologie}
```

Př. 8.6: Nečíselovaný nadpis umístěný v obsahu dokumentu:

```
\subsection*{Poznámky k~technologii CMOS}  
\addcontentsline{toc}{subsection}{Poznámky k~technologii CMOS}
```

Př. 8.7: Nečíselovaný nadpis, jehož text je vložen do živého záhlaví:

```
\section*{Přílohy a~komentáře}  
\markboth{Přílohy a~komentáře}{Přílohy a~komentáře}
```

Na procvičení:

1. Zvolte dvoustranný dokument na formát papíru A5 s šíří sazby 100 mm a nastavte všechny ostatní rozměry pro umístění sazebního obrazce.
2. Vysázejte titulní stránku dokumentu (sazba na střed, hlavní titul do jedné třetiny výšky sazby, velké tučné písmo, u dolního okraje sazby město a rok).
3. Vyzkoušejte na zvoleném materiálu nadpisy sekce, podsekce a podpodsekce ve třídě dokumentu `article`. Vytvořte obsah dokumentu. Nastavte záhlaví typu `headings`.

Svislé zarovnání, tabulky

Na rozdíl od strojopisu se v sazbě zarovnání nikdy nedociluje mezislovními mezerami, ale k tomu určenými prostředky. Konkrétně v systému \LaTeX jsou k dispozici dvě prostředí – `tabbing` (pro tabelační zarážky) a `tabular` (pro tabulky).

Prostředí `tabbing`

Nejdůležitějšími prvky jsou příkaz pro nastavení tabelační zarážky a příkaz pro použití pozice tabelační zarážky. Prostor mezi tabelačními zarážkami tvoří samostatnou skupinu. Příklad:

Příklad prostředí `tabbing`

```
\begin{tabbing}
\bfseries Město \hspace{30mm} \=      nastavení tabelační zarážky
      \bfseries Teplota v~7 h \|[5pt]
Ústí nad Labem \> 17,5 $^\circ$C \\použití pozice zarážky
Pardubice \> 16,8 $^\circ$C \\nastavení druhé zarážky
Ostrava \> 15,9 \=$^\circ$C \\nastavení druhé zarážky
  \> \> nekalibrovaná stupnice \\nastavení druhé zarážky
Břeclav \> 19,5 $^\circ$C
\end{tabbing}
```

Po vysázení dostáváme:

Město	Teplota v 7 h
Ústí nad Labem	17,5 °C
Pardubice	16,8 °C
Ostrava	15,9 °C
	nekalibrovaná stupnice
Břeclav	19,5 °C

Prostředí `tabular`

Tabulky v prostředí `tabular` mají různé zajímavé možnosti. Ve srovnání s interaktivními programy se práce s tabulkami dost odlišuje, některé případy jsou poměrně obtížné, ale na druhé straně existuje řada nástrojů umožňující sazbu automatizovat a pro běžné případy tak výrazně zjednodušit práci.

Ze širokého repertoáru možností ukážeme několik příkladů.

Základní tvar tabulky

```
\begin{tabular}[vertikální poloha]{definice sloupců} \hline
pole & pole & ... \\ \hline
...
pole & pole & ... \\ \hline
\end{tabular}
```

vertikální poloha – udává polohu vůči okolí. Implicitně se tabulka umístí vertikálně středem k okolí, zadáním tohoto parametru v podobě písmene **b** se tabulka umístí spodním okrajem, písmenem **t** horním okrajem vzhledem k okolí.

definice sloupců – posloupnost znaků symbolicky znázorňujících sloupce tabulky, jejich zarovnání a materiál mezisloupcového prostoru. Písmena **l**, **c**, **r** po řadě znamenají sloupec zarovnaný vlevo, na střed a vpravo. Zápis `p{míra}` znamená sloupec zarovnaný do bloku na šířku *míra*. Svislá čára znázorňuje mezisloupcový prostor vyplněný svislou čarou. Příkaz `@{...}` umožňuje definovat materiál mezisloupcového prostoru, který se automaticky bude opakovat v každém řádku.

pole – pole tabulky je samostatnou skupinou s libovolným materiálem.

Příklad jednoduché tabulky

```
\begin{tabular}{rc}
\bfseries Body & \bfseries Znamka \\[5pt]
$<$50 & nevyhověl \\
50--69 & dobře \\
70--89 & velmi dobře \\
90--100 & výborně
\end{tabular}
```

Po vysázení dostáváme:

Body	Znamka
<50	nevyhověl
50–69	dobře
70–89	velmi dobře
90–100	výborně

V následující tabulce ukážeme dvě potřebné rekvizity – spojení tabulkových polí a dvě metody zarovnání.

Potřebujeme-li tabulkové pole přes více jiných, použijeme příkaz `\multicolumn{počet}{zarovnání}{text}`.

Tento příkaz umožňuje v nově vytvořeném sloučeném poli definovat jiný způsob zarovnání a mezisloupcových materiálů. Proto jej lze využít i pro pouhou redefinici způsobu zarovnání.

Správné svislé zarovnání materiálu tabulkových polí je zejména u číselných hodnot zcela zásadním prvkem. V příkladu jsou ukázány dvě metody zarovnání. Jedna spočívá v rozdělení na dva sloupce (ukázáno na sloupci „skóre“). Druhá metoda ukázaná ve sloupci „body“ doplňuje každou hodnotu přídatnou mezerou na stejnou šíři, sloupec pak lze zarovnat na střed. Mezeru, která je přesně rovna šířce číslice, zajišťuje příkaz `\hphantom{0}` a pro lepší čitelnost a použitelnost jsme tuto mezeru definovali kratším příkazem `\fm`.


```

\def\fm{\hphantom{0}}
\begin{tabular}{|r||l|r@{:}r|c|} \hline
\bfseries p. & \bfseries země & & & \\
\multicolumn{2}{c|}{\bfseries skóre} & & & \\
\bfseries body & & & & \\ \hline \hline
1. & Itálie & 23 & 4 & 14 \\ \hline
2. & Francie & 18 & 7 & 10 \\ \hline
3. & Švédsko & 14 & 10 & \fm 8 \\ \hline
4. & Švýcarsko & 3 & 18 & \fm 3 \\ \hline
\end{tabular}

```

Po vysázení dostáváme:

p.	země	skóre	body
1.	Itálie	23:4	14
2.	Francie	18:7	10
3.	Švédsko	14:10	8
4.	Švýcarsko	3:18	3

Ovlivnění velikosti řádkové rozteče v tabulce lze provést redefinicí koeficientu `\arraystretch`, například `\def\arraystretch{1.4}`. Velikost mezisloupcového prostoru lze ovlivnit hodnotou registru `\tabcolsep`, jehož implicitní hodnota je 6 pt. Předcházející tabulku jsme upravili nastavením

```
\def\arraystretch{1.3} \tabcolsep=15pt.
```

p.	země	skóre	body
1.	Itálie	23:4	14
2.	Francie	18:7	10
3.	Švédsko	14:10	8
4.	Švýcarsko	3:18	3

Na procvičení:

1. Pomocí prostředí `tabbing` vysázejte:

```

procedure Cisluj(var C: longint);
var R: string;
    Pozice: byte;
begin Pozice:=5;
    while not eof do begin
        readln(R);
        writeln(C: Pozice, '-', R)
    end
end;

```

2. Vysázejte tabulku malé násobilky.
3. Vysázejte svůj osobní rozvrh hodin.

Matematická sazba

Typografické zásady a principy

Matematická sazba byla vždy pro sazeče největším oříškem a byli ji schopni kvalitně provést jen ti nejlepší. To byl také údajně jeden z hlavních důvodů, proč chtěl D. Knuth tento problém vyřešit počítačovou podporou. Jeho systém je zcela jednoznačně to nejlepší, co lze v počítačové typografii použít.

Sazba matematických (ale i fyzikálních a chemických) výrazů podléhá velkému množství specifických pravidel. Potřebná jednoznačnost a přesnost těchto výrazů vyžaduje jejich důsledné respektování, protože na rozdíl od běžného textu může velmi často dojít k zásadnímu významovému posunu.

Každý matematický symbol má svůj jednoznačný a stálý tvar (typ písma, řez, velikost, okolní mezerování). Je přitom nepodstatné, zda se příslušný symbol vyskytuje v samostatném výrazu, nebo jako součást odstavce. Jak mají symboly správně vypadat? Odpověď na tuto klíčovou otázku asi nejlépe řeší norma ČSN ISO 31, v jejíž části 0 jsou všeobecné zásady a v části 11 tabulka matematických symbolů.

Koncept technického řešení

Většinu zmíněných vlastností umí systém pohlídat, ale v některých případech je potřebné implicitní způsob sazby poněkud usměrnit. Sazba matematiky je soustředěna do **matematických prostředí**. Uvnitř odstavce lze použít tzv. **textovou matematiku** (text math), mezi odstavci pak **vysazenou matematiku** (display math). Symbol použitý v textové matematice je pokud možno přizpůsoben odstavcové sazbě, aby co nejméně narušil řádkování. Ve vysazené matematice tedy může být větší nebo může mít odlišný tvar.

Realizace v systému L^AT_EX

Systém L^AT_EX má čtyři matematická prostředí:

1. Pro textovou matematiku, jsou tři možnosti: \dots nebo $\langle \dots \rangle$ nebo $\begin{math} \dots \end{math}$
2. Pro vysazenou matematiku, jsou opět tři možnosti:
$$\dots$$
 nebo
$$\langle \dots \rangle$$
 nebo
$$\begin{displaymath} \dots \end{displaymath}$$
3. Pro číslovanou vysazenou matematiku, $\begin{equation} \dots \end{equation}$
Toto prostředí automaticky čísloje daný vztah, číslo se vypisuje do závorky na pravý

okraj sazby. Příklad:

`\begin{equation} a+b=c \end{equation}` se vysází jako

$$a + b = c \tag{10.1}$$

4. Pro číslovanou víceřádkovou vysazenou matematiku,

`\begin{eqnarray} \dots \end{eqnarray}`

Prostředí umožňuje zarovnávat pod sebe soustavy vztahů (například rovnic) a každý řádek číslovat (případně číslo vynechat). Jedná se v podstatě o tabulku, která má tři sloupce, levý sloupec je automaticky zarovnáván vpravo, prostřední sloupec na střed a pravý sloupec vlevo. Příklad:

```
\begin{eqnarray}
2y + 3x & = & 13 \\
5y - 10x & = & 15 \\
y - 2x & = & 3
\end{eqnarray}
```

po vysázení dává

$$2y + 3x = 13 \tag{10.2}$$

$$5y - 10x = 15 \tag{10.3}$$

$$y - 2x = 3 \tag{10.3}$$

V libovolném z uvedených prostředí lze stejným způsobem aplikovat matematické prvky a symboly:

Proměnné a konstanty – zápisem písmene se automaticky volí matematická kurzíva, zápisem číslice se volí obyčejný řez. Chceme-li tento implicitní způsob změnit, například pro vysazení písmenné konstanty, použijeme změnu řezu `\mathrm{}`, například `a+\mathrm{e}` dává $a + e$ a znamená součet proměnné a se základem přirozeného logaritmu.

Minus – v matematickém režimu se zapisuje spojovníkem, např. `-b-3` dává $-b-3$. Všimněte si, že znak minus má dvojí význam – unární (představuje indikaci záporné nebo opačné hodnoty, sází se bez mezery) a binární (naznačuje operaci odečítání, sází se s okolními mezerami). Systém sám rozpozná, o jakou funkci tohoto znaku jde.

Desetinná čárka – zdánlivě bezproblémový znak s sebou nese drobnou záludnost: v anglické typografii má v matematice jiný význam, a to je oddělovač v seznamu hodnot s přídatnými mezerami. Proto zápis `1,2,3` dává 1, 2, 3. Při zápisu desetinného čísla v češtině však nemůžeme za desetinnou čárkou trpět mezeru, nemůžeme tedy jednoduše zapsat například `3,141`, což vede na 3, 141. Je nutné psát `3{,}141` (vysází se jako 3,141), kdy skupina kolem čárky zamezí nežádoucímu mezerování.

Exponent – zapisuje se znakem `^`, např. `c^2` dává c^2 nebo `x^{a+b}` dává x^{a+b} .

Index – zapisuje se podtržítkem, např. `x_i` dává x_m nebo `a_{ij}` dává a_{ij} .

Odmocnina – příkaz `\sqrt{}`, např. `\sqrt{a+b}` dává $\sqrt{a+b}$ nebo `\sqrt[3]{a^2+b^2}` dává $\sqrt[3]{a^2+b^2}$.

Zlomek – příkaz `\frac{čitatel}{jmenovatel}`, např. `\frac{1}{2}` dává $\frac{1}{2}$ nebo `\frac{a+b}{c-d}` dává $\frac{a+b}{c-d}$.

Řecká abeceda – každé řecké písmeno má svůj příkaz, `\alpha` (α) až `\omega` (ω), velká řecká písmena mají příkazy začínající velkým písmenem `\Gamma` (Γ), ... `\Omega` (Ω). Některá chybějí, protože jsou shodná s latinkovými písmeny.

Různé symboly – těch je velké množství, například `\infty` (∞ , nekonečno), `\emptyset` (\emptyset , prázdná množina), `\circ` (\circ , kolečko), `\times` (\times , násobení), `\rightarrow` (\rightarrow , šipka vpravo) atd.

Znamé funkce – sázejí se vzpřímeným řezem a obvykle mají specifické mezerování, jsou k nim k dispozici odpovídající příkazy, například `\sin x` ($\sin x$), podobně kosinus atd.

Sumy, integrály, limity... – symboly s možností sazby mezi nad a pod symbol. U nich se projevuje rozdíl mezi textovou a vysazenou matematikou. Příklad: `\sum_{i=1}^{n-1} a_i` se vysází v textové matematice jako $\sum_{i=1}^{n-1} a_i$ a ve vysazené matematice jako

$$\sum_{i=1}^{n-1} a_i$$

Podobně pracuje symbol pro integrál `\int`, součin `\prod` atd. Limita `\lim_{x \rightarrow \infty} \frac{2x}{x-3}` se vysází jako

$$\lim_{x \rightarrow \infty} \frac{2x}{x-3}$$

Velké oddělovače – závorky, svíslé čáry a některé jiné značky je potřebné někdy zvětšit tak, aby správně obklopovaly uzavřený výraz. Oddělovače, které mají schopnost se takto zvětšit, jsou pak sázeny speciální dvojicí příkazů `\left` a `\right`. Tyto příkazy musí být vždy v páru. Pokud na jedné straně oddělovač nechceme, zapíšeme místo něj k příkazu `left` nebo `right` jen tečku. Například

`\left\{\left[\frac{x^2}{\sqrt{x-2}}\right]\right\}` se vysází jako

$$\left\{ \left[\frac{x^2}{\sqrt{x-2}} \right] \right\}$$

Text – v matematickém výrazu jsou občas potřeba doprovodná vysvětlení nebo vyjádření proměnných slovně apod., která jsou sázena stejným písmem jako okolní text. K tomu účelu můžeme použít příkaz `\mbox{}`, materiál v jeho parametru bude sázen jako obyčejný text. Příklad:

`x + 2k^2 - y \leq 0 \quad \mbox{a odsud} \quad y \geq x + 2k^2` dává

$$x + 2k^2 - y \leq 0 \quad \text{a odsud} \quad y \geq x + 2k^2$$

Matice a jiná pole – podobně, jako je v textu k dispozici prostředí `tabular` pro sazbu tabulek, je v matematickém prostředí k dispozici `array`. Všechny jeho možnosti jsou totožné s prostředím `tabular`, včetně způsobu zápisu parametru a jednotlivých tabulkových polí. Jednou z mála drobných odlišností je registr, který řídí mezisloupcové mezery, jmenuje se `\arraycolsep` (v prostředí `tabular` se jmenuje `\tabcolsep`). Například

```

\left(
\begin{array}{r|ll}
a & b+c & 17 \\
b & a+c & 22 \\ \hline
c & a+b & 6
\end{array}
\right)

```

po vysázení vytvoří

$$\left(\begin{array}{c|cc} a & b+c & 17 \\ b & a+c & 22 \\ \hline c & a+b & 6 \end{array} \right)$$

Na procvičení:

Vysázejte:

1.

$$P = \sum_{i=1}^n g(A + ik) \quad (10.4)$$

2.

$$y = \left[\frac{x(x^2 - 1)}{(x + 1)} - \frac{x + 1}{x(x - 1)^2} \right]^3$$

3.

$$C \leq 2^N \quad (10.5)$$

$$\log C \leq N \log 2$$

$$N = \left\lceil \frac{\log C}{\log 2} \right\rceil \quad (10.6)$$

4.

$$\text{úspěšnost} = \frac{\text{počet správných odpovědí}}{\text{počet příkladů}}$$

5.

$$\lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x + \Delta x)}{\Delta x}$$

6.

$$\text{nespojité funkce } g(\xi) = \begin{cases} 0 & \text{pro } \xi < 0 \\ 2\xi & \text{pro } \xi \in (0, 1) \\ 2 & \text{pro } \xi \geq 1 \end{cases}$$

7.

$$\text{přenosová matice: } \begin{bmatrix} \alpha_{11} & \beta_{12} \\ \gamma_{21} & \delta_{22} \end{bmatrix} \quad (10.7)$$

Kreslení a vkládání obrázků, práce s barvami

Typografické zásady a principy

Obrázky, fotografie, grafy, schémata, náčrtky apod. významně doplňují zejména odborné texty. Ne nadarmo se říká, že „jeden obrázek je lepší než tisíc slov“. Provedení obrázků však musí odpovídat určitým pravidlům, aby se jejich potenciální přínos nezměnil na zátěž.

Jedním ze zásadních pravidel typografie je jednota významově stejných nebo podobných prvků. To plně platí i u vkládané grafiky. Měli bychom dodržet stejnou koncepci, rozměry, barevnost a způsob vkládání do textového materiálu. K tomu můžeme s výhodou využít příkazy, které si sami nadefinujeme a které při důsledném používání tuto jednotnost zaručí.

Koncept technického řešení

Grafiku můžeme z technického hlediska rozdělit na rastrovou a vektorovou. Pro dosažení maximální kvality je samozřejmostí v maximální míře využívat vektorových zdrojů. Při zpracování rastrových zdrojů bychom měli dopředu vědět, na jaká výstupní zařízení bude výsledek posílán, v jejich hustotě a kvalitě bychom měli připravit rastrové podklady.

Realizace v systému L^AT_EX

V systému L^AT_EX jsou většinou grafické prvky a práce s barvami řešeny až na úrovni prohlížeče vysázeného výstupu. Samotný systém tedy vlastně o grafice a barvách „nic neví“ a vkládá jen příkazy, které bude realizovat výstupní program (nejčastěji prohlížeč PDF).

Výjimkou je určitá možnost kreslení jednoduché vektorové grafiky pomocí prostředí `picture`. Toto prostředí je použitelné nejen na jednoduché obrázky, ale také na libovolné umístění jakéhokoliv materiálu v ploše sazby. Jeho popis přesahuje možnosti tohoto textu, podívejme se tedy alespoň v příkladu na základní koncept.

Prostředí `picture`

Prostředí pracuje s relativními rozměry, jejichž základem je míra uložená v registru `\unitlength`. Jeho změnou lze tedy snadno škálovat nakreslený obrázek.

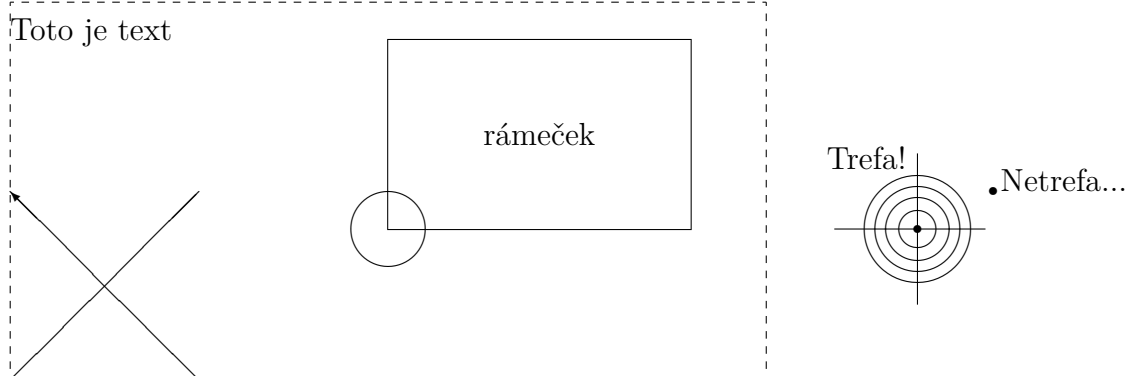
Příklad prostředí picture

```

\begin{picture}(100,50)  kreslicí plocha 100 × 50 jednotek
\put(0,45){Toto je text}  příkaz \put umísťuje objekty na plochu
\put(50,20){\framebox(40,25){rámeček}}  rámeček s textem uvnitř
\put(0,0){\dashbox(100,50){}}  čárkovaný rámeček kolem celé pracovní plochy
\put(50,20){\circle{10}}  kružnice
\put(0,0){\line(1,1){25}}  úsečka
\put(25,0){\vector(-1,1){25}}  vektor (úsečka se šipkou)
\put(120,10){\line(0,1){20}}  Materiál umístěný mimo pracovní plochu
\put(109,20){\line(1,0){20}}  překročení rozměrů plochy není hlídáno
\put(108,28){Trefa!}
\put(120,20){\circle*{1}}  soustředné kružnice tvořící terč
\put(120,20){\circle{5}}
\put(120,20){\circle{8}}
\put(120,20){\circle{11}}
\put(120,20){\circle{14}}
\put(130,25){\circle*{1}}
\put(131,25){Netrefa...}
\end{picture}

```

Nastavíme jednotku `\unitlength=1mm` a po vysázení dostaneme:



Vkládání obrázků z externích souborů

Práce s grafikou je soustředěna do balíčku `graphicx`, který je potřebné k dokumentu připojit. Pak je k dispozici základní příkaz `\includegraphics{}`, který umožňuje vložit obrázek ze souboru, jehož název je vložen do parametru. Formát souboru je závislý na verzi překladače, v překladači `pdflatex` je možné použít soubory formátu PDF (hlavně pro vektorovou grafiku), JPG a PNG (pro rastrovou grafiku). Vzhledem k univerzálnosti těchto formátů se však jimi pokryjí prakticky všechny potřeby.

Příklad: `\includegraphics{bardotka_nahled}` vysází



Balíček `graphicx` však umožňuje ještě další manipulace. Obsahuje například tyto příkazy:

`\resizebox` – úprava velikosti materiálu v parametru. Například

`\resizebox{20mm}{3mm}{Ahoj}` vytvoří **Ahoj**. Nechceme-li tvarově zkreslit upravovaný materiál, uvedeme v jednom z rozměrů vykřičník, například:

`\resizebox{20mm}{!}{Ahoj}` vysází již nezkreslené **Ahoj**.

`\scalebox` – úprava velikosti materiálu násobkem. Hodnoty menší než 1 zmenšují, hodnoty větší než jedna zvětšují. Příklad: `\scalebox{2.5}{\xi}` vysází ξ.

`\reflectbox` – zrcadlové převrácení materiálu v parametru. Příklad:

`\reflectbox{Správná odpověď}` vysází řevonqbo šnvšiqǫ.

`\rotatebox` – otočení materiálu o zadaný úhel ve stupních (kladná hodnota je proti směru hodinových ručiček). Příklad: `\rotatebox{45}{\xi}` dává ξ.

Práce s barvami

Veškeré příkazy pro práci s barvami jsou součástí balíčku `color`. Po jeho připojení k dokumentu máme k dispozici tyto možnosti:

`\definecolor` – definice identifikátoru barvy. Řada barev je již předdefinována, ale pro vlastní potřebu je velmi často nutné definovat nový identifikátor, který symbolicky označuje zvolený odstín. Příkaz má tři parametry:

`\definecolor{název}{model}{definice barvy}` Barevný model může být `rgb`, `cmym` nebo `gray`, podle toho se v definici barvy použítí tři, čtyři nebo jedna intenzitní hodnota. Příklad:

`\definecolor{mendelgreen}{cmyk}{0.53,0,0.85,0}`

`\color` – příkaz nastavuje kreslicí barvu, nastavení je platné do konce skupiny. Například: `{\color{blue}Modrý nápis}` vysází **Modrý nápis**.

`\textcolor` – nastavuje barvu pro materiál ve svém parametru.

`\textcolor{blue}{Modrý nápis}` dává **Modrý nápis**.

`\colorbox` – příkaz umožňuje nastavit barvu pozadí materiálu v parametru. Příklad:

`\colorbox{yellow}{Toto je rámeček}` vysází **Toto je rámeček**.

`\fcolorbox` – příkaz obarví ohraničující čáru a pozadí rámečku daného materiálem v parametru. Příklad:

`\fcolorbox{blue}{yellow}{Toto je rámeček}` dává **Toto je rámeček**.

`\pagecolor` – příkaz obarví pozadí celé dané stránky, například

`\definecolor{podklad}{gray}{0.9}\pagecolor{podklad}` vytvoří 10% šedou podkladovou barvu aktuální stránky a následujících stránek.

Na procvičení:

1. Vytvořte v prostředí `picture` organizační schéma fakulty.
2. Připravte libovolný obrázek a vložte jej do sazby připraveného textu s vhodnou úpravou velikosti.
3. Zjistěte definici barvy Vaší organizace a napište touto barvou její název.

Literatura k dalšímu studiu

- ČSN 01 6910 *Úprava dokumentů psaných textovými procesory*. Praha: Český normalizační institut, 2014.
- LAMPORT, L. *L^AT_EX – A Document Preparation System: User’s Guide and Reference Manual*. Reading: Addison-Wesley Publ. Comp., 1994.
- GOOSENS, M., MITTELBAACH, F., SAMARIN, A. *The L^AT_EX Companion*. Reading: Addison-Wesley, 1994.
- OLŠÁK, P. *Typografický systém T_EX*. Brno: Konvoj, 2000.
- RYBIČKA, J. *L^AT_EX pro začátečníky*. Brno: Konvoj, 2003.