

Zpravodaj Československého sdružení uživatelů TeXu

Hans Hagen; Ton Otten; Vít Zýka; Ján Buša; Jiří Hrbek; Martina Plachá; Petr Tesařík
Exkurze do ConTeXtu

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 16 (2006), No. 2-4, 1–195

Persistent URL: <http://dml.cz/dmlcz/150013>

Terms of use:

© Československé sdružení uživatelů TeXu, 2006

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*
<http://dml.cz>

Exkurze do ConTeXtu

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

česka
verze

Ton Otten & Hans Hagen
PRAGMA ADE

konec

vyhledej

+

krok zpět

+

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

Z anglického originálu ConTeXt an Excursion přeložili Vít Zýka, Ján Buša, Jiří Hrbek, Martina Plachá a Petr Tesářík.
Překlad tohoto manuálu vznikl za podpory grantu Československého sdružení uživatelů \TeX (CSTUG, CSTUG).
Kapitola věnovaná sazbě českých a slovenských textů a příloha Další informační zdroje je původní text překladatelů. Děkujeme správčům serveru **Supelec** za hostování zdrojových textů pomocí SVN.

Tento dokument verze 6. prosince 2006 je generován pomocí nástrojů $\text{CON}\text{\TeX}\text{T}$, $\text{\TeX}\text{EDIT}$, $\text{\TeX}\text{UTIL}$ a $\text{Web2C}\ \text{PDF}\text{\TeX}$. Je sázen písmem Palladio. K testování jsme používali GHOSTSCRIPT a Acrobat Reader .

\TeX a $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ jsou obchodní značky American Mathematical Society; METAFONT je obchodní značka Addison-Wesley Publishing Company; PostScript , $\text{Portable Document Format}$ a Acrobat jsou obchodní značky Adobe Systems Incorporated; DVIPSONE a DVIWINDO jsou obchodní značky Y&Y Incorporated; IBM je obchodní značka International Business Machines Corporation; MSDOS je obchodní značka MicroSoft Corporation; všechna další jména výrobků jsou obchodní značky jejich výrobců.

© 1991–2006 PRAGMA ADE, Ridderstraat 27, 8061GH Hasselt, Nizozemsko, www.pragma-ade.com

konec

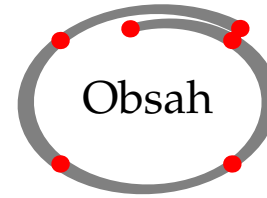
vyhledej

+

krok zpět

-

+



1	Jak vytvořit dokument	3	26	Barva	97
2	Jak zpracovat vstupní soubor	5	27	Pozadí textu	100
3	Zvláštní znaky	7	28	Pozadí stránky	101
4	Vymezení dokumentu	8	29	Zarovnání	102
5	Nadpisy	9	30	Interaktivní mód	
6	Odrážky a výčty	14		v elektronických dokumentech	104
7	Sazba matematiky	21	31	Písmo a přepínače fontů	111
8	Příkazy v matematickém módu	30	32	Složené znaky	119
9	Jednotky	33	33	Formátování stránky	121
10	Obrázky	34	34	Odsazení odstavců	126
11	Tabulky	42	35	Nastavovací příkazy	131
12	Tabelace / formátování odstavce	57	36	Definování příkazů / maker	133
13	Sloupce	61	37	Nezařazené	136
14	Poznámky pod čarou	64	38	Externí moduly	155
15	Definice pojmů	68	39	Grafické rozšíření / METAPOST	158
16	Číslované poznámky	70	40	Uživatelská nastavení	159
17	Orámování textu	73	41	Postup zpracování	159
18	Orámování odstavce	76	42	Pomocné soubory	161
19	Marginálie	78	43	Sazba českých/slovenských textů	162
20	Zalamování a číslování stránek	80			
21	Záhlaví a úpatí	84	A	Další informační zdroje	165
22	Obsah (seznamy)	85	B	Definice příkazů	168
23	Rejstříky	90	C	Seznam příkazů	185
24	Synonyma	93	D	Rejstřík	189
25	Křížové odkazy	95			

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

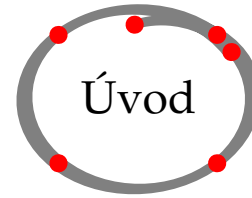
vyhledej

+

krok zpět

-

+



CONTEXT je inženýrský systém pro dokumentování založený na T_EXu. T_EX je sázecí systém a programovací jazyk zaměřený na sazbu a generování dokumentů. CONTEXT je snadno použitelný a umožňuje vytvářet složité tištěné i elektronické dokumenty.

Tento manuál popisuje možnosti CONTEXTu, popisuje dostupné příkazy a jejich použití.¹

CONTEXT je vyvíjen na praktických aplikacích: sazba a generování dokumentů počínaje jednoduchými beletristickými knihami a konče složitými technickými manuály a odbornými publikacemi v tištěné i elektronické podobě. Tento začátečnický manuál popisuje funkcionalitu CONTEXTu nezbytnou k formátování základních textových elementů používaných v manuálech nebo příručkách. CONTEXT je však schopný vytvářet mnohem více a pro toho, kdo se nespokojí s těmito základními funkcemi, je připraveno mnoho dalších detailnějších manuálů a jiných informací, Viz **přílohu A**.

CONTEXT má lokalizovatelné rozhraní umožňující uživatelům pracovat s CONTEXTem v jejich vlastním jazyce. V současné době existuje rozhraní holandské, německé, anglické, italské, francouzské, rumunské a také české. Tento manuál je dostupný v holandštině, němčině, angličtině a češtině.

¹ Všechny papírové a elektronické dokumenty o CONTEXTu jsou připraveny CONTEXTem. Všechny zdrojové texty těchto dokumentů jsou nebo budou elektronicky dostupné, aby umožnily vhlad do práce s tímto systémem.

obsah
rejstřík
seznam příkazů
definice příkazů
tiráž

konec	
vyhledej	+
krok zpět	
-	+

1 | Jak vytvořit dokument

Řekněme, že chceme vytvořit jednoduchý dokument. Má nějakou strukturu a obsahuje titulní stranu, několik kapitol, sekcí a podsekcí. Přirozeně má také obsah a rejstřík.

CONTEXT dokáže takový dokument vytvořit automaticky, když mu poskytneme správný vstup ve formě souboru. Nejprve tedy musíme vytvořit vstupní soubor. Úplný název vstupního souboru se skládá ze jména a přípony. Jméno můžeme zvolit libovolné, ale přípona musí být `tex`. Když vytvoříme soubor s názvem `muj_soubor.tex`, nebudeme mít s během CONTEXTu žádné potíže.

Vstupní soubor může vypadat následovně:

```
\starttext
\startstandardmakeup
  \midaligned{Jak vytvořit dokument}
  \midaligned{Autor}
\stopstandardmakeup
\completecontent
\chapter{Předmluva}
... vlastní text\index{prvek rejstříku} ...
\chapter{První kapitola}
\section[prvnisekce]{První sekce}
... vlastní text ...
\section{Druhá sekce}
\subsection{První podsekce}
... vlastní text\index{další prvek rejstříku} ...
\subsection{Druhá podsekce}
... vlastní text ...
```

obsah
rejstřík
seznam příkazů
definice příkazů
tiráž

konec	
vyhledej	+
krok zpět	
-	+

```

\section{Třetí podsekce}
... vlastní text ...
\chapter{Další kapitola}
... vlastní text ...
\chapter[poslednikapitola]{Poslední kapitola}
... vlastní text ...
\completeindex
\stoptext
    
```

CONTEX_T očekává na vstupu prostý soubor v kódování ASCII. Můžeme samozřejmě používat libovolný textový editor nebo procesor, ale nesmíme zapomenout, že CONTEX_T dokáže načíst jenom ASCII vstup. Většina editorů nebo textových procesorů umí exportovat soubory do prostého ASCII.

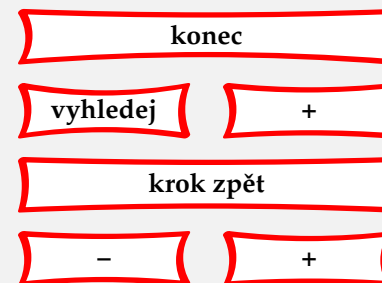
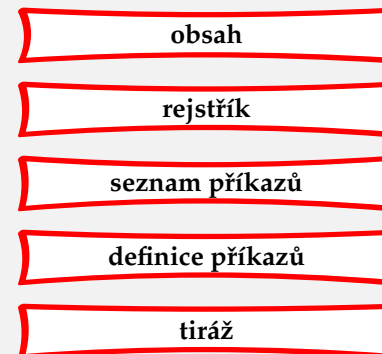
Vstupní soubor by měl obsahovat text, který chceme zpracovat programem CONTEX_T, a příkazy CONTEX_Tu. Příkazy CONTEX_Tu začínají znakem \. Příkazem \starttext označujeme začátek textu. Oblast před příkazem \starttext se nazývá deklarační oblast a používá se k definování nových příkazů a nastavení vzhledu dokumentu.

Za příkazem obvykle následuje dvojice hranatých závorek [] a/nebo levá a pravá složená závorka {}. V sekvenci \chapter [poslednikapitola] {Poslední kapitola} například povel \chapter po CONTEX_Tu požaduje, aby vykonal několik činností týkajících se úpravy, typografie a struktury. Těmito činnostmi může být:

1. začátek nové strany,
2. zvětšení počítadla kapitol o jednotku,
3. umístění čísla kapitoly před její název,
4. vynechání určitého vertikálního prostoru,
5. použití většího písma nebo
6. uložení názvu kapitoly (a čísla stránky) do obsahu.

Tyto činnosti budou vykonány s argumentem zadaným mezi levou a pravou svorkou: *Poslední kapitola*.

Dosud jsme se nezmínili o parametru [poslednikapitola] mezi hranatými závorkami. Je to návěští s referenčním jménem, které je možné použít pro odkaz na odpovídající kapitolu. Toho se docílí dalšími příkazy CONTEX_Tu: \in{kapitole} [poslednikapitola] vysází číslo kapitoly, zatímco \about [poslednikapitola] vrátí její název.



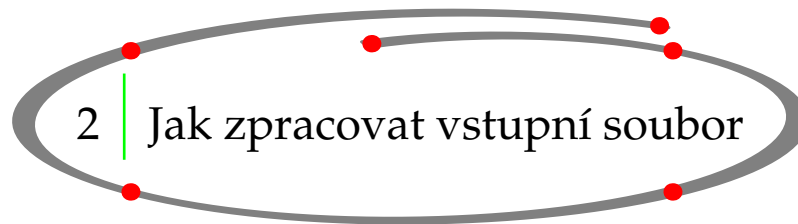
Seznam činností tedy může být rozšířen o:

7. zavedení referenčního jména `poslednikapitola`, které reprezentuje číslo a název kapitoly (a uloží se pro pozdější použití).

Dalším činnostem týkajícím se průběžných nadpisů, nastavení čítačů nebo interaktivních vlastností se nyní nebudeme věnovat.

Když necháme `CONTEXT` zpracovat uvedený vzorový soubor, získáme velmi jednoduchý dokument s několika očíslovanými kapitolami a s hlavičkami oddílů.

Během zpracování souboru se `CONTEXT` stará o spoustu věcí. Jednou z nich je například číslování stránek. Ovšem k vytvoření obsahu `CONTEXT` potřebuje znát čísla stránek, která mu při prvním průběhu zatím nejsou známá. Proto musíme tento soubor zpracovat dvakrát (dvojprůchodová úloha). K uložení těchto a podobných informací `CONTEXT` vytvoří několik pomocných souborů, které se pak zpracovávají programem `TEXUTIL`. V některých případech musíme vstupní soubor zpracovat třikrát (trojprůchodová úloha). Je také možné spouštět `CONTEXT` z příkazové řádky příkazem `TEXEXEC`. Tento skript, dříve napsaný v PERLU dnes v RUBY, se postará i o vícenásobné průchody. `TEXEXEC` je součástí standardní distribuce `CONTEXTu`.

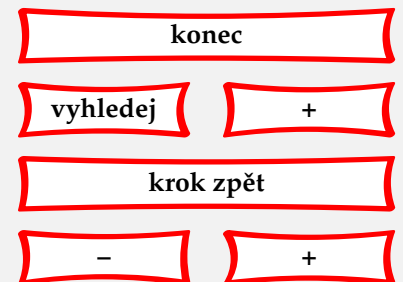
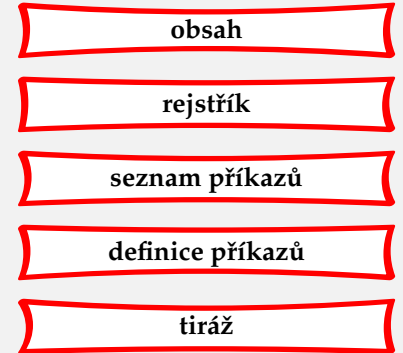


Jestliže chceme zpracovat vstupní soubor `CONTEXTu`, můžeme na příkazové řádce napsat:

```
context jmenosouboru
```

Dostupnost dávkového příkazu `context` závisí na použitém systému. Na to, jakým příkazem se spouští `CONTEXT`, se zeptejte správce svého systému. Pokud se náš soubor jmenuje `muj_soubor.tex`, může to být:

```
context muj_soubor
```



nebo, když je řádně nainstalován $\text{T}_{\text{E}}\text{XMFSTART}$ a $\text{T}_{\text{E}}\text{XEXEC}$:

```
texmfstart texexec --pdf muj_soubor
```

Příponu `.tex` není potřeba zadávat.

Po stlačení klávesy `Enter` začne zpracovávání. $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ vypisuje na obrazovku informace o průběhu zpracování. Když je soubor zpracován úspěšně, znovu se objeví výzva v příkazové řádce a $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ vytvoří soubor `dvi` nebo `pdf`.

Jestliže zpracování úspěšné není – například proto, že jsme napsali `\stptext` namísto `\stoptext` — $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ na obrazovku vypíše znak `?` a oznámí nám, že právě narazil na chybu. Poskytneme nám základní informace o druhu chyby a číslo řádky, na kterém se chyba projevila.

V okamžiku, kdy se objeví znak `?` můžeme napsat:

H	pro nápovědu k chybě,
I	pro vložení opraveného příkazu $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ u,
Q	pro ignorování chyb bez výpisu,
R	pro ignorování chyb s výpisem,
X	pro ukončení běhu, nebo
Enter	pro ignorování chyby.

Nejčastěji stlačíme klávesu `Enter` a zpracování bude pokračovat. Pak můžeme editovat vstupní soubor a chybu odstranit.

Některé chyby způsobí, že se na obrazovku vypíše znak `*` a současně se zpracování zastaví. To je důsledek závažné chyby ve vstupním souboru. Tuto chybu nemůžeme ignorovat a jedinou možností je napsat `\stop` nebo `Ctrl Z`. Program se ukončí a my můžeme chybu odstranit.

Během zpracování vstupního souboru nás $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ také informuje o činnostech, které na našem dokumentu provádí. Zobrazuje například čísla stránek a informace o jednotlivých krocích. Mimo to vypisuje varování. Ta se týkají typografie a oznamují nám neúspěšné zalomení řádku. Veškeré informace o zpracování se ukládají do záznamového souboru s příponou `.log`, kde lze přezkoumat varování i chyby a nalézt čísla příslušných řádek vstupního souboru.

Při úspěšném zpracování $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ vytvoří nový soubor s příponou `.dvi` nebo `.pdf`, v našem příkladu bude tedy vytvořen soubor `muj_soubor.dvi` nebo `muj_soubor.pdf`. Zkratka `dvi` znamená `Device Independent` (nezávislý na zařízení). To znamená, že soubor může být dále převeden patřičným ovladačem `PostScriptové (PS)` tiskárny na soubor vhodný k tisku nebo k prohlížení. Zkratka `PDF` znamená `Portable Document Format` (přenosný formát dokumentu). Je to formát pro tisk a prohlížení, nezávislý na operačním systému.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+



Už jsme si ukázali, že každý povel `CONTEXTU` je uvozen znakem `\` (obrácené lomítko). To znamená, že `\` má pro `CONTEXT` zvláštní význam. Kromě `\` existují ještě další znaky, které vyžadují zvláštní pozornost, pokud je chceme použít v doslovném ‚verbatim‘ módu nebo v módu textovém. **Tabulka 3.1** uvádí přehled těchto zvláštních znaků a povely potřebné k jejich vysázení.

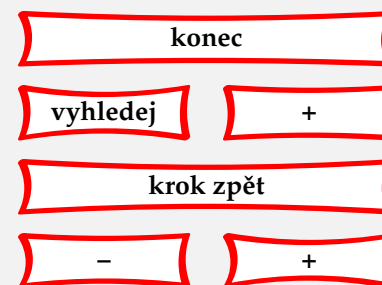
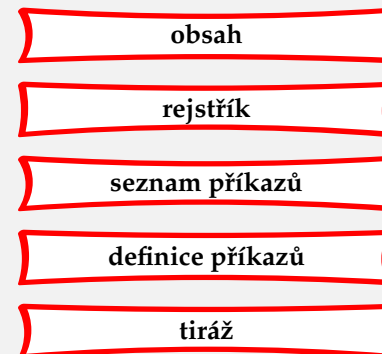
Zvláštní znak	Verbatim		Text	
	Vstup	Vysází se	Vstup	Vysází se
#	<code>\type{#}</code>	#	<code>\#</code>	#
\$	<code>\type{\$}</code>	\$	<code>\\$</code>	\$
&	<code>\type{&}</code>	&	<code>\&</code>	&
%	<code>\type{%}</code>	%	<code>\%</code>	%

Tabulka 3.1 Zvláštní znaky (1).

Další znaky mají zvláštní význam při sázení matematických výrazů a některé lze použít jedině v matematickém módu (viz **kapitulu 7**).

Zvláštní znak	Verbatim		Text	
	Vstup	Vysází se	Vstup	Vysází se
+	<code>\type{+}</code>	+	<code>+\$+\$</code>	+
-	<code>\type{-}</code>	-	<code>-\$-\$</code>	-
=	<code>\type{=}</code>	=	<code>=\$=\$</code>	=
<	<code>\type{<}</code>	<	<code>\$\$<\$</code>	<
>	<code>\type{>}</code>	>	<code>\$\$>\$</code>	>

Tabulka 3.2 Zvláštní znaky (2).



4 | Vymezení dokumentu

Každý dokument začíná povelom `\starttext` a je ukončen povelom `\stoptext`. Veškerý textový vstup je umístěn mezi tyto dva povely a jedině zde jej `CONTExT` bude zpracovávat.

Nastavení parametrů se umísťuje do deklarační oblasti, která se nachází těsně před povelom `\starttext`.

```
\setupbodyfont[12pt]
\starttext
Toto je jednořádkový dokument.
\stoptext
```

Uvnitř `\starttext ... \stoptext` lze dokument rozdělit na čtyři hlavní úseky:

1. úvodní část (angl. front matter)
2. hlavní část (angl. body matter)
3. závěrečná část (angl. back matter)
4. přílohy (angl. appendices)

Jednotlivé úseky se definují následovně:

```
\startfrontmatter ... \stopfrontmatter
\startbodymatter ... \stopbodymatter
\startbackmatter ... \stopbackmatter
\startappendices ... \stopappendices
```

V úvodní a v závěrečné části vytváří povel `\chapter` nečíslovaná záhlaví v obsahu. Tyto úseky se používají nejčastěji pro obsah, seznam obrázků a tabulek, předmluvu, poděkování apod. Strany jsou zde často číslovány římskými číslicemi.

Úsek příloh se používá – kdo by to byl řekl? – pro přílohy. Záhlaví zde mohou být sázena jiným způsobem, např. `\chapter` může místo čísel používat písmena abecedy.

Styl sekcí lze nastavit pomocí:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

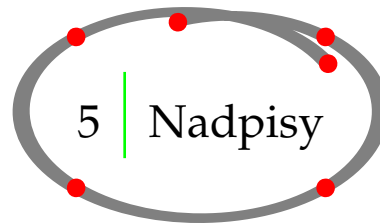
krok zpět

-

+

```
\setupsectionblock [.1.] [...,2.,...]
```

```
1 IDENTIFIER
2 number = yes no
   page  = yes right
   before = COMMAND
   after  = COMMAND
```



Struktura dokumentu je určena nadpisy. Nadpisy (záhlaví) se vytvářejí pomocí povelů uvedených v **tabulce 5.1**:

Číslovaný nadpis	Nečíslovaný nadpis
<code>\chapter</code>	<code>\title</code>
<code>\section</code>	<code>\subject</code>
<code>\subsection</code>	<code>\subsubject</code>
<code>\subsubsection</code>	<code>\subsubsubject</code>
...	...

Tabulka 5.1 Nadpisy.

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +

```
\chapter [...1;...] {.2.}  
                OPTIONAL
```

1 REFERENCE

2 TEXT

```
\section [...1;...] {.2.}  
                OPTIONAL
```

1 REFERENCE

2 TEXT

```
\subsection [...1;...] {.2.}  
                OPTIONAL
```

1 REFERENCE

2 TEXT

```
\title [...1;...] {.2.}  
                OPTIONAL
```

1 REFERENCE

2 TEXT

```
\subject [...1;...] {.2.}  
                OPTIONAL
```

1 REFERENCE

2 TEXT

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\subsubject [...1;...] {.2.}
                OPTIONAL
```

1 REFERENCE

2 TEXT

Tyto povely vytvoří nadpis v předem dané velikosti a řezu písma a přidají vertikální mezeru před a za nadpis.

Povelům pro nadpis je možné zadat několik parametrů, jako např.:

```
\title[hasselt-v-noci]{Hasselt v noci}
```

a

```
\title{Hasselt v noci}
```

Hranaté závorky jsou nepovinné a používají se pro odkazování uvnitř dokumentu. Pokud se chceme odkázat na tento nadpis, můžeme napsat např. `\at{str.}[hasselt-v-noci]`.

Tyto nadpisy lze samozřejmě přizpůsobit našemu vkusu a dají se dokonce definovat vlastní nadpisy. To se dělá povelům `\setuphead` a `\definehead`.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setuphead [...1,...] [...2,...]
```

```
1 SECTION
2 style          = normal bold slanted boldslanted type cap small...
                  COMMAND
   textstyle     = normal bold slanted boldslanted type cap small...
                  COMMAND
   numberstyle   = normal bold slanted boldslanted type cap small...
                  COMMAND
   color         = IDENTIFIER
   textcolor     = IDENTIFIER
   numbercolor   = IDENTIFIER
   number        = yes no
   ownnumber     = yes no
   page          = left right yes
   continue      = yes no
   header        = none empty high nomarking
   text          = none empty high nomarking
   footer        = none empty high nomarking
   before        = COMMAND
   inbetween     = COMMAND
   after         = COMMAND
   alternative   = normal inmargin middle TEXT
   hang          = none broad fit line NUMBER
   command       = \...#1#2
   numbercommand = \...#1
   textcommand   = \...#1
   deepnumbercommand = \...#1
   deeptextcommand = \...#1
   prefix        = + - TEXT
   placehead     = yes no empty
   incrementnumber = yes no LIST FILE
   resetnumber   = yes no
   file          = IDENTIFIER
   expansion     = yes no command
   margintext    = yes no
   inherits from \setupheads
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\definehead [.1.] [.?.]
```

```
1 IDENTIFIER
```

```
2 SECTION
```

```
\definehead
[mujnadpis]
[section]
```

```
\setuphead
[mujnadpis]
[numberstyle=bold,
textstyle=bold,
before=\hairline\blank,
after=\nowhitespace\hairline]
```

```
\mujnadpis[nadpis]{Hasselt má velkou publicitu}
```

Tím se definuje nový typ nadpisu `\mujnadpis`, který zdědí vlastnosti `\section`. Vypadal by nějak takto:

5.1 Hasselt má velkou publicitu

Ještě o jednom povelu bysme měli vědět, a to o `\setupheads`. Používá se k nastavení číslování pro číslované nadpisy. Pokud napíšeme:

```
\setupheads
[alternative=inmargin,
separator=--]
```

pak se všechna čísla objeví na okraji. Sekce 1.1 by vypadala jako 1–1.

Povely jako `\setupheads` se píšou do deklarační oblasti vstupního souboru.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+


```
\setupheads [...,.*,...]
```

```
* sectionnumber = yes NUMBER no
  alternative   = normal margin middle TEXT paragraph
  separator    = TEXT
  stopper      = TEXT
  align        = inner outer left right middle normal no yes
  alightitle   = yes float no
  tolerance    = verystrict strict tolerant verytolerant stretch
  indentnext   = yes no
  command     = \...#1#2
  margin      = DIMENSION
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

6 | Odrážky a výčty

Jedním ze způsobů, jak rozčlenit text, je vyjmenování nebo shrnutí jednotlivých položek. Příkaz pro vytvoření seznamu vypadá takto:

```
\startitemize [...,1...] [...,2,...] ... \stopitemize
      OPTIONAL      OPTIONAL
1 a A KA n N m r R KR NUMBER continue standard broad serried packed
  stopper joinedup atmargin inmargin intro columns text paragraph repeat
2 inherits from \setupitemize
```

Příklad:

konec

vyhledej

+

krok zpět

-

+

```

\startitemize[R,packed,broad]
\item Hasselt byl založen ve 14.století.
\item Hasselt je známý jako tzv. hanzovní město.
\item Jméno Hasselt vychází z-názvu stromu.
\stopitemize

```

Uvnitř dvojice `\startitemize ... \stopitemize` začínáme každou novou položku příkazem `\item` a za `\item` nezapomeňme na mezeru. V našem příkladu parametr `R` specifikuje římské číslování a `packed` minimální mezeru mezi řádky. Parametr `broad` zajistí rozumnou horizontální mezeru mezi značkou a textem. Náš příklad by vypadal:

- I. Hasselt byl založen ve 14.století.
- II. Hasselt je známý jako tzv. hanzovní město.
- III. Jméno Hasselt vychází z názvu stromu.

Rozčlenění textu do položek je dvoufázový proces. To znamená, že pro získání optimálního rozložení budete muset svůj soubor překompilovat dvakrát. V závorkách je obsažena informace o symbolu odrážky a lokální nastavení proměnných.

Argument	Značka před položkou
1	•
2	–
3	★
⋮	⋮
n	1 2 3 4 ...
a	a b c d ...
A	A B C D ...
r	i ii iii iv ...
R	I II III IV ...

Tabulka 6.1 Nastavení formátu odrážek.

Můžete také nadefinovat svoji vlastní značku pomocí příkazu `\definesymbol`. Vyzkoušejte například

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

–

+

```

\definesymbol[5] [$\clubsuit$]
\startitemize[5,packed]
\item Hasselt byl zbudován na písčném přesypu.
\item Hasselt leží na soutoku dvou řek.
\stopitemize

```

Dostanete pak tento výstup:

- ♣ Hasselt byl zbudován na písčném přesypu.
- ♣ Hasselt leží na soutoku dvou řek.

Někdy je potřeba mít jednotlivé položky nečíslované. V takovém případě použijte místo `\item` příkaz `\head`.

Hasselt leží v~provincii Overijssel, pro kterou je typické množství různých zvyklostí.

```
\startitemize
```

```
\head kraamschudden \hfill (vítání novorozeněte)
```

Po narození dítěte sousedé navštíví novopečené rodiče. Zatímco ženy přicházejí dítě obdivovat, muži jej (pokud je to chlapec) posoudí v ostatních ohledech. Je zvykem, že s sebou přinášejí `{\em krentenwegge}`, což je asi 1~\Meter\ dlouhý hrozinkový chléb. Nedílnou součástí oslav narození je samozřejmě tradiční alkoholický nápoj.

```
\head nabuurschap (naberschap) \hfill (společenství)
```

Prosperita malých komunit bývala velmi závislá na vzájemné pomoci jejích členů. Tito členové `{\em nabuurschap}`ů si pomáhali v~časích žní, pohřbů a~vůbec při všech těžkostech, které na komunitu doléhaly.

```
\head Abraham \& Sarah \hfill (stáří)
```

Když lidé v~Hasseltu dosáhnou padesáti let, říká se o~nich, že se vídají s~Abrahamem a~Sárou. Je zvykem dát jim tzv. `{\em speculaas}`, což je zvláštní druh kořeněných keksů.

```
\stopitemize
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Způsob, jakým se příkaz `\head` vysází, může být ovlivněn pomocí `\setupitemize`. V případě zlomu stránky, se `\head` objeví vždy na nové stránce. (Příkaz `\Meter` bude vysvětlen v kapitole 9.)

Příklad o starých zvycích vypadá takto:

Hasselt leží v provincii Overijssel, pro kterou je typické množství různých zvyklostí.

- kraamschudden (vítání novorozence)
Po narození dítěte sousedé navštíví novopečené rodiče. Zatímco ženy přicházejí dítě obdivovat, muži jej (pokud je to chlapec) posoudí v ostatních ohledech. Je zvykem, že s sebou přináší *krentenwegge*, což je asi 1 m dlouhý hrozinkový chléb. Nedílnou součástí oslav narození je samozřejmě tradiční alkoholický nápoj.
- nabuurschap (naberschap) (společenství)
Prosperita malých komunit bývala velmi závislá na vzájemné pomoci jejích členů. Tito členové *nabuurschapů* si pomáhali v časích žní, pohřbů a vůbec při všech těžkostech, které na komunitu doléhaly.
- Abraham & Sarah (stáří)
Když lidé v Hasseltu dosáhnou padesáti let, říká se o nich, že se vídají s Abrahamem a Sárrou. Je zvykem dát jim tzv. *speculaas*, což je zvláštní druh kořeněných keksů.

Parametry pro nastavení způsobu zobrazení položek jsou uvedeny v **tabulce 6.2**.

Způsob zobrazení položek je sice možné měnit u každého `\startitemize`, ale pro zachování jednotného vzhledu je vhodné jej nastavit pro celý dokument pomocí `\setupitemize`.

Parameter `columns` se používá společně s označením počtu, například kód:

```
\startitemize[n,columns,four]
\item Achter 't Werk
.
.
\item Justitiebastion
\stopitemize
```

vysází:

- | | | | |
|-------------------|-------------------|-----------------|---------------------|
| 1. Achter 't Werk | 4. Eikenlaan | 7. Heerengracht | 10. Julianakade |
| 2. Baangracht | 5. Eiland | 8. Hofstraat | 11. Justitiebastion |
| 3. Brouwersgracht | 6. Gasthuisstraat | 9. Hoogstraat | |

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Parametr	Význam
standard	standardní (globalní) nastavení
packed	bez vertikální mezery mezi položkami
serried	žádná horizontální mezera mezi značkou a textem
joinedup	žádné odsazení před a po bloku položek
broad	horizontální mezera mezi značkou a textem
inmargin	umístit značku položek do okraje
atmargin	umístit značku položek k okraji
stopper	umístí ukončení značky (implicitně '.')
columns	rozmístí položky do sloupců
intro	brání stránkovému zlomu mezi odrážkami a jejich uvozujícím odstavcem
continue	pokračování v předchozím číslování

Tabulka 6.2 Parametry pro `\startitemize`.

Někdy chceme po krátké přestávce v číslování položek pokračovat. Pak stačí napsat například `\startitemize[continue,columns,three,broad]` a číslování bude pokračovat ve formátu tří sloupců.

- | | | |
|------------------|--------------------|----------------------|
| 12. Kaai | 18. Prinsengracht | 24. Vicariehof |
| 13. Kalverstraat | 19. Raamstraat | 25. Vissteeg |
| 14. Kastanjelaan | 20. Ridderstraat | 26. Watersteeg |
| 15. Keppelstraat | 21. Rosmolenstraat | 27. Wilhelminalaan |
| 16. Markt | 22. Royenplein | 28. Ziekenhuisstraat |
| 17. Meestersteeg | 23. Van Nahuijsweg | |

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\setupitemize [...1] [...2;...] [...3.,...]
                OPTIONAL   OPTIONAL   OPTIONAL
1  NUMBER each
2  standard broad serried packed unpacked stopper joinedup atmargin inmargin
   autointro loose repeat SECTION paragraph intext random
3  margin      = no standard DIMENSION
   leftmargin  = no standard DIMENSION
   rightmargin = no standard DIMENSION
   width       = DIMENSION
   distance    = DIMENSION
   factor      = NUMBER
   items       = NUMBER
   start       = NUMBER
   before      = COMMAND
   inbetween   = COMMAND
   after       = COMMAND
   left        = TEXT
   right       = TEXT
   beforehead  = COMMAND
   afterhead   = COMMAND
   headstyle   = normal bold slanted boldslanted type cap small... COMMAND
   marstyle    = normal bold slanted boldslanted type cap small... COMMAND
   symstyle    = normal bold slanted boldslanted type cap small... COMMAND
   stopper     = TEXT
   n           = NUMBER
   symbol      = NUMBER
   align       = inner outer left right middle normal no yes
   indentnext  = yes no
    
```

Vnořené položky se automaticky vysázejí správně. Pokud například napíšete:

V Nizozemí mohou výši celé řady daní určovat města, takže životní náklady se město od města liší. Rozdíly dosahují až 50\,% v~daních jako:

```

\setupitemize[2][width=5em]
\startitemize[n]
\item Daň z~nemovitostí.
    
```

Daně z~nemovitostí se dělí do dvou složek:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\startitemize[a,packed]
\item daň z-vlastnictví nemovitosti a
\item daň z-nájmu.
\stopitemize

```

Pokud nemovitost není v-nájmu, platí obě složky vlastník.

\item Poplatek za psa.

Vlastník jednoho nebo více psů platí poplatek. Pokud pes zemře, nebo je prodán, je vlastník povinen nahlásit to na radnici.

```

\stopitemize

```

potom horizontální mezeru mezi značkou a textem vnořené položky lze nastavit pomocí:
`\setupitemize[2][width=5em]`.

Náš příklad pak bude vypadat takto:

V Nizozemí mohou výši celé řady daní určovat města, takže životní náklady se město od města liší. Rozdíly dosahují až 50% v daních jako:

1. Daň z nemovitostí.

Daně z nemovitostí se dělí do dvou složek:

- a. daň z vlastnictví nemovitosti a
- b. daň z nájmu.

Pokud nemovitost není v nájmu, platí obě složky vlastník.

2. Poplatek za psa.

Vlastník jednoho nebo více psů platí poplatek. Pokud pes zemře, nebo je prodán, je vlastník povinen nahlásit to na radnici.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

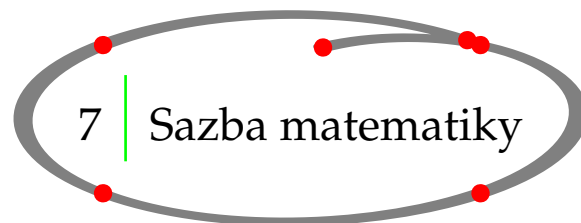
vyhledej

+

krok zpět

-

+



7.1 Úvod

\TeX je *především* program pro sazbu matematiky. Přesto tato kapitola týkající se sazby matematiky není rozsáhlá, jak byste možná očekávali. Pokud potřebujete v \TeX u sázet vzorce, doporučujeme vám ke studiu:²

- *The \TeX Book*, autor: D. E. Knuth
- *The Beginners Book of \TeX* , autoři: S. Levy a R. Seroul

Kromě toho o sazbě matematických rovnic pojednává $\text{CON}\text{\TeX}$ TOVÝ magazín ‚My Way: Using $\backslash\text{startalign}$ and friends‘ od Aditya Mahajana, viz **přílohu A**.

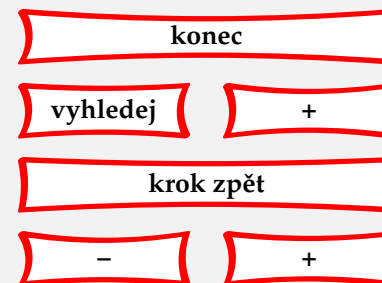
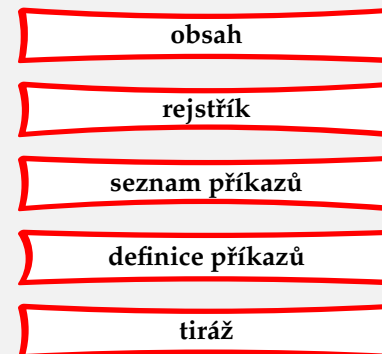
7.2 Sazba matematiky

Zvyklosti pro sazbu běžného a matematického textu se obvykle liší. \TeX tyto zvyklosti ‚zná‘ a během vytváření dokumentu je přesně aplikuje. Pokud je požadována vysoká kvalita matematické sazby, můžeme se na \TeX plně spolehnout.

Několik zvyklostí pro sazbu matematiky:

1. Znaky jsou sázeny v *matematické kurzíve* (neplést s *běžnou kurzívou* při použití fontů).
2. Užívají se symboly jako řecké znaky (α , χ) a matematické značky (\leq , \geq , \in).
3. Délky mezer jsou také jiné než v běžném textu.
4. V matematických výrazech se také užívá jiný způsob zarovnání textu.
5. Horní a dolní indexy jsou umístěny automaticky. Například: a_c^b .
6. Určité symboly mají různý vzhled podle toho, zda jsou uvedeny ve vnitřním nebo v display matematickém módu.

² V tomto manuálu se při sazbě matematiky opíráme o knížku *\TeX niques*, kterou napsal Arthur Samuel.



Při sazbě matematiky musíte pracovat v tak zvaném matematickém módu, ve kterém jsou výrazy definovány pomocí příkazů plain $\text{T}_{\text{E}}\text{X}$.

Matematický mód má dvě alternativy: vnitřní matematický mód a display matematický mód. Vnitřní matematický mód je ohraničen znaky $\$$ a $\$,$ zatímco display mód je aktivován pomocí $\$\$$ a $\$\$.$

Město Hasselt pokrývá plochu $42,05 \text{ \Square \Kilo \Meter}.$

Pokud budeme uvažovat kruhovou oblast této velikosti s-hasseltským tržištěm v-geometrickém středu $\$T\$,$ můžeme spočítat průměr této oblasti pomocí vzorce $\$\{\{1\}\over{4}\} \pi d^2\$.$

Tento text bude vypadat po vysázení takto:

Město Hasselt pokrývá plochu $42,05 \text{ km}^2.$ Pokud budeme uvažovat kruhovou oblast této velikosti s hasseltským tržištěm v geometrickém středu $T,$ můžeme spočítat průměr této oblasti pomocí vzorce $\frac{1}{4}\pi d^2.$

Větší množství $\{\}$ (složených závorek) ve výrazu $\frac{1}{4}\pi d^2$ je potřeba pro určení priority jednotlivých operací. Pokud zapomeneme na vnější závorky a zapíšeme: $\$\{1\}\over{4} \pi d^2\$,$ dostaneme nežádoucí výsledek: $\frac{1}{4\pi d^2}.$

Písmena a číslice jsou sázena ve třech velikostech (stylech): velikost textu $a + b,$ velikost indexu (script): $a+b$ a velikost indexu v indexu (scriptscript) $a+b.$ Velikost lze ovlivnit příkazy \scriptstyle a $\scriptscriptstyle.$

Symbols jako \int a \sum vypadají jinak ve vnitřním a jinak v display matematickém módu. Pokud napíšeme $\$\sum_{n=1}^m\$$ nebo $\$\int_{-\infty}^{+\infty}\$,$ dostaneme $\sum_{n=1}^m$ a $\int_{-\infty}^{+\infty}.$ To samé v display módu: $\$\$\sum_{n=1}^m\$\$$ a $\$\$\int_{-\infty}^{+\infty}\$\$$ však bude vypadat takto:

$$\sum_{n=1}^m a \int_{-\infty}^{+\infty}$$

Pomocí příkazů \nolimits a \limits můžeme ovlivnit vzhled symbolů \sum a $\int:$

$$\sum_{n=1}^m a \int_{-\infty}^{+\infty}$$

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Pro sazbu zlomků můžeme použít příkaz `\over`. V `CONTEXT`u můžeme používat také alternativu `\frac`. Například pro vysázení $\frac{a}{1+b} + c$ stačí zapsat: `$$\frac{a}{1+b}+c$`.

Další příkazy pro umístění různých částí výrazu nad sebe:

<code>\atop</code>	<code> \${a} \atop {b}\$</code>	$\frac{a}{b}$
<code>\choose</code>	<code> \${n+1} \choose {k}\$</code>	$\binom{n+1}{k}$
<code>\brack</code>	<code> \${m} \brack {n}\$</code>	$\left[\begin{matrix} m \\ n \end{matrix} \right]$
<code>\brace</code>	<code> \${m} \brace {n-1}\$</code>	$\left\{ \begin{matrix} m \\ n-1 \end{matrix} \right\}$

`TeX` může přizpůsobit velikost závorek jako `()` a `{}` automaticky, pokud levou závorku předchází příkaz `\left` a podobně pravou `\right`. Například zápis `$$1+\left(\frac{1}{1-x^{x-2}}\right)^3$$` bude vypadat:

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

Sazby horního a dolního indexu dosáhneme pomocí `,`, `'` a `^`. Tyto značky mají vliv pouze na první následující znak, takže víceznakové indexy je nutné ohraničovat složenými závkami `{}`.

V určitých situacích mohou být závorky zvětšeny pomocí příkazů `\bigl`, `\Bigl`, `\biggl`, `\Biggl` a jejich pravostranných ekvivalentů. Pokud to nestačí, tak ještě větších závorek můžeme dosáhnout spoluprací příkazů `\left` a `\right` s konstrukcí `\vbox`. Více než popis to objasní příklad, byť poněkud méně čitelný: `$$\left(\vbox to 16pt{x^{2^{2^{2^2}}}}\right)$$`

$$\left(x^{2^{2^{2^2}}} \right)$$

V `display` matematickém módu podléhají automatickému přizpůsobení velikosti následující symboly:

<code>\lfloor</code>	<code> [</code>	<code>\langle</code>	<code>\left[</code>	<code>\lvert</code>	<code> </code>	<code>\downarrow</code>	<code>↓</code>
<code>\rfloor</code>	<code>]</code>	<code>\rangle</code>	<code>\right]</code>	<code>\lVert</code>	<code> </code>	<code>\Downarrow</code>	<code>⇓</code>
<code>\lceil</code>	<code> [</code>	<code>/</code>	<code>\uparrow</code>	<code>\Uparrow</code>	<code>↑</code>	<code>\updownarrow</code>	<code>↕</code>
<code>\rceil</code>	<code>]</code>	<code>\backslash</code>	<code>\Uparrow</code>	<code>\Updownarrow</code>	<code>↑</code>	<code>\Updownarrow</code>	<code>⇕</code>

Pokud v `display` módu chceme použít složené zlomky, pak bychom měli zapsat běžným způsobem pouze ten vnější a pro ostatní by se měla zvolit forma `a/b`. Pro vysázení:

$$a_0 + \frac{a}{a_1 + \frac{1}{a_2}}$$

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

bychom neměli psát $a_0 + \frac{a}{a_1 + \frac{1}{a_2}}$ ale spíše $a_0 + \frac{a}{a_1 + 1/a_2}$, abychom dostali přehlednější

$$a_0 + \frac{a}{a_1 + 1/a_2}$$

Mimoto lze použít příkaz `\displaystyle`. Pokud zapíšeme $a_0 + \frac{a}{a_1 + \frac{1}{a_2}}$, tak dostaneme:

$$a_0 + \frac{a}{a_1 + \frac{1}{a_2}}$$

Níže ukážeme použití příkazů `\matrix`, `\pmatrix`, `\ldots`, `\cdots` a `\cases` bez dalšího vysvětlování.

\$\$

```
A=\left(\matrix{x-\lambda & 1 & & 0 & & \\ 0 & & x-\lambda & & 1 & \\ 0 & & 0 & & & x-\lambda \cr}\right)
```

\$\$

$$A = \begin{pmatrix} x - \lambda & 1 & 0 \\ 0 & x - \lambda & 1 \\ 0 & 0 & x - \lambda \end{pmatrix}$$

\$\$

```
A=\left|\matrix{x-\mu & 1 & & 0 & & \\ 0 & & x-\mu & & 1 & \\ 0 & & 0 & & & x-\mu \cr}\right|
```

\$\$

$$A = \begin{vmatrix} x - \mu & 1 & 0 \\ 0 & x - \mu & 1 \\ 0 & 0 & x - \mu \end{vmatrix}$$

\$\$

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
A=\pmatrix{a_{11} & a_{12} & \ldots & a_{1n} \cr
a_{21} & a_{22} & \ldots & a_{2n} \cr
\vdots & \vdots & \ddots & \vdots \cr
a_{m1} & a_{m2} & \ldots & a_{mn} \cr}
```

\$\$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

\$\$

```
|x|=\cases{\hphantom{-}x, & pokud $x\geq 0$; \cr
-x, & jinak. \cr}
```

\$\$

$$|x| = \begin{cases} x, & \text{pokud } x \geq 0; \\ -x, & \text{jinak.} \end{cases}$$

Pokud chceme vysázet v matematickém výrazu běžný text, musíme uvážit dvě věci. Za prvé, mezera se v matematickém módu se nevysází běžným způsobem a je třeba ji vynutit pomocí `\` (obráceného lomítka). Za druhé, je nutné změnit font, protože běžný text by se neměl sázet *matematickou kurzivou*, ale aktuálním fontem.

$$x^3 + \text{termy nižších řádů}$$

Matematické funkce jako `\sin` a `\tan`, které se musí vysázet aktuálním fontem, jsou v $\text{T}_\text{E}_\text{X}$ u předdefinované:

```
\arccos \cos \csc \exp \ker \limsup \min \sinh
\arcsin \cosh \deg \gcd \lg \ln \Pr \sup
\arctan \cot \det \hom \lim \log \sec \tan
\arg \coth \dim \inf \liminf \max \sin \tanh
```

Pokud zapíšeme funkci sinus `$$\sin 2\theta=2\sin\theta\cos\theta$$` nebo limitu `$$\lim_{x\to 0}\{\frac{\sin x}{x}\}=1$$` dostaneme:

$$\sin 2\theta = 2 \sin \theta \cos \theta \quad \text{nebo} \quad \lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Zarovnání v matematických výrazech si zaslouží zvláštní pozornost. Ve víceřádkových výrazech někdy potřebujeme zarovnaní podle znaménka ‚=‘. To se dělá pomocí příkazu `\eqalign`. Pokud zapíšeme:

```


$$\begin{aligned} ax^2+bx+c &= 0 & \backslash\text{cr} \\ x &= \frac{-b \pm \sqrt{b^2-4ac}}{2a} & \backslash\text{cr} \end{aligned}$$


```

vysází se:

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Někdy je potřeba zarovnávat podle více různých míst. Jak lze tento požadavek uskutečnit, je vidět na druhém řádku následujícího příkladu:

```


$$\begin{aligned} ax+bx+\dots+yx+zx & & = & x(a+b+\dots) & \backslash\text{cr} \\ & & & \phantom{=} x(a+\dots)+y+z & \backslash\text{cr} \\ & & & = y & \backslash\text{cr} \end{aligned}$$


```

Výsledkem pak je:

$$ax + bx + \dots + yx + zx = x(a + b + \dots + y + z)$$

$$= y$$

Vedle příkazu `\phantom` je zde ještě `\hphantom` bez výšky a `\vphantom` bez šířky.

Na velikosti mezer v matematickém výrazu se většinou můžeme na $\text{T}_\text{E}_\text{X}$ plně spolehnout. Nicméně v některých situacích se nám může hodit jejich velikost ovlivňovat. To se dělá pomocí:

```

\!   -\frac{1}{6}\quad
\,   -\frac{1}{5}\quad
\>  -\frac{2}{5}\quad
\;   -\frac{3}{8}\quad

```

Velikost těchto ‚mezer‘ je určena relativně k mezeře, kterou lze vyvolat příkazem `\quad` a která má šířku velkého písmene ‚M‘.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Užití příkazu `\prime`³ hovoří sama za sebe. Pokud bychom například chtěli vysázet $y'_1 + y''_2$, měli bychom zapsat `$y_1^{\prime}+y_2^{\prime\prime}`.

Výraz jako $\sqrt[3]{x^2 + y^2}$ lze získat pomocí zápisu `$$\root 3 \of {x^2+y^2}$$`.

Na konci této sekce si ukážeme použití příkazu `\mathstrut`, kterým můžeme dosáhnout vyrovnání velikostí, například symbolů odmocnin. Pomocí `$$\sqrt{\mathstrut a}+\sqrt{\mathstrut d}+\sqrt{\mathstrut y}$$` dostaneme $\sqrt{a} + \sqrt{d} + \sqrt{y}$. Bez příkazu `\mathstrut` bychom dostali $\sqrt{a} + \sqrt{d} + \sqrt{y}$.

7.3 Umístění matematických formulí

Jednotlivé matematické vzorce můžeme číslovat pomocí:

```
\placeformula [...1,...] {..2.} $$..3.$$
                OPTIONAL  OPTIONAL
```

- 1 REFERENCE
- 2 TEXT
- 3

```
\startformula ... \stopformula
```

Dva příklady:

```
\placeformula[formula:aformula]
\startformula
  y=x^2
\stopformula

\placeformula
\startformula
  \int_0^1 x^2 dx
\stopformula
```

³ „prime“ je anglický výraz pro čárku v indexu. Pozn. překl.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

$$y = x^2 \quad (7.1)$$

$$\int_0^1 x^2 dx \quad (7.2)$$

CONTEXTové příkazy `\startformula ... \stopformula` nahrazují počáteční a koncové `$$`. Když totiž napíšeme:

```
$$
\int_0^1 x^2 dx
$$
```

dostaneme výraz, který je sice *zobrazen* ve středu stránky, ale není tak dobře zarovnan jako v předcházejících příkladech.

$$\int_0^1 x^2 dx$$

Příkaz `\placeformula` zajistí mezeru okolo matematického vzorce a také její označení číslem. Hranaté závorky nejsou povinné a slouží pro uvedení referenčního jména pro odkazy, a také pro zapnutí a vypnutí číslování.

$$y = x^2 \quad (7.3)$$

$$y = x^3 \quad (7.4)$$

$$y = x^4 \quad (7.5)$$

Rovnice 7.4 byla vysázena takto:

```
\placeformula[rovnice:prostredni]
\startformula
y=x^3
\stopformula
```

Nápis v hranaté závorce `[rovnice:prostredni]` slouží pro odvolání se na tuto rovnici v textu. Tuto referenci jsme provedli příkazem `\in{Rovnice}[rovnice:prostredni]`.

Jestliže nevyžadujeme číslování, zapíšeme:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\placeformula[-]`

Číslování rovnic se nastavuje pomocí `\setupnumbering`. V tomto manuálu je číslování nastaveno příkazem `\setupnumbering[way=bychapter]`. To znamená, že před číslem rovnice je uvedeno číslo kapitoly a v každé další kapitole je číslování opět vynulováno. Pro zachování jednotnosti jsou tabulky, obrázky apod. číslovány stejným způsobem. Proto se `\setupnumbering` používá v deklarační oblasti vstupního souboru.

Rovnice mohou být nastaveny pomocí příkazu:

```
\setupformulae [...,.*,...]
```

```
* location = left right
left = TEXT
right = TEXT
align = inner outer left right middle normal no yes
option = middle
strut = yes no
distance = DIMENSION
margin = DIMENSION standard yes no
leftmargin = DIMENSION
rightmargin = DIMENSION
indentnext = yes no
alternative = IDENTIFIER
spacebefore = DIMENSION
after = DIMENSION
separator = TEXT
conversion = numbers characters Characters romannumerals Romannumerals TEXT
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

8 Příkazy v matematickém módu

8.1 Řecká abeceda

α	<code>\alpha</code>	ι	<code>\iota</code>	ϱ	<code>\varrho</code>
β	<code>\beta</code>	κ	<code>\kappa</code>	Σ	<code>\sigma</code>
Γ	<code>\gamma</code>	Λ	<code>\lambda</code>	ς	<code>\varsigma</code>
Δ	<code>\delta</code>	μ	<code>\mu</code>	τ	<code>\tau</code>
ϵ	<code>\epsilon</code>	ν	<code>\nu</code>	υ	<code>\upsilon</code>
ε	<code>\varepsilon</code>	Ξ	<code>\xi</code>	Φ	<code>\phi</code>
ζ	<code>\zeta</code>	o	<code>o</code>	φ	<code>\varphi</code>
η	<code>\eta</code>	Π	<code>\pi</code>	χ	<code>\chi</code>
Θ	<code>\theta</code>	ω	<code>\varpi</code>	Ψ	<code>\psi</code>
ϑ	<code>\vartheta</code>	ρ	<code>\rho</code>	Ω	<code>\omega</code>

8.2 Zvláštní symboly

\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>
\jmath	<code>\jmath</code>	$\sqrt{\quad}$	<code>\sqrt{\quad}</code>	\flat	<code>\flat</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>
\wp	<code>\wp</code>	\perp	<code>\perp</code>	\sharp	<code>\sharp</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\clubsuit	<code>\clubsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	\diamondsuit	<code>\diamondsuit</code>
∂	<code>\partial</code>	\triangle	<code>\triangle</code>	\heartsuit	<code>\heartsuit</code>
∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\spadesuit	<code>\spadesuit</code>

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

8.3 Operátory doplňující +, - a *

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\vee	<code>\vee</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\wedge	<code>\wedge</code>
\setminus	<code>\setminus</code>	\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>
\cdot	<code>\cdot</code>	\sqcap	<code>\sqcap</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\sqcup	<code>\sqcup</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\triangleleft	<code>\triangleleft</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\triangleright	<code>\triangleright</code>	\dagger	<code>\dagger</code>
\diamond	<code>\diamond</code>	\wr	<code>\wr</code>	\ddagger	<code>\ddagger</code>
\circ	<code>\circ</code>	\bigcirc	<code>\bigcirc</code>	\amalg	<code>\amalg</code>
\bullet	<code>\bullet</code>	\bigtriangleup	<code>\bigtriangleup</code>		
\div	<code>\div</code>	\bigtriangledown	<code>\bigtriangledown</code>		

8.4 Operátory

Σ	<code>\sum</code>	\prod	<code>\prod</code>	\coprod	<code>\coprod</code>
\int	<code>\int</code>	\oint	<code>\oint</code>	\bigcap	<code>\bigcap</code>
\bigcup	<code>\bigcup</code>	\bigsqcup	<code>\bigsqcup</code>	\bigvee	<code>\bigvee</code>
\bigwedge	<code>\bigwedge</code>	\bigodot	<code>\bigodot</code>	\bigotimes	<code>\bigotimes</code>
\bigoplus	<code>\bigoplus</code>	\biguplus	<code>\biguplus</code>		

8.5 Relace doplňující >, < a =

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>
$<$	<code>\prec</code>	$>$	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\smile	<code>\smile</code>	\mid	<code>\mid</code>	\doteq	<code>\doteq</code>
\frown	<code>\frown</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

8.6 Negované relace

$\not<$	$\not>$	$\not=$
$\not\leq$	$\not\geq$	$\not\equiv$
$\not\prec$	$\not\succ$	$\not\sim$
$\not\preceq$	$\not\succeq$	$\not\approx$
$\not\subset$	$\not\supset$	$\not\cong$
$\not\subseteq$	$\not\supseteq$	$\not\asymp$
$\not\sqsubset$	$\not\sqsupseteq$	

8.7 Šipky

\leftarrow	\longleftarrow	\uparrow
\Leftarrow	\Longleftarrow	\Uparrow
\rightarrow	\longrightarrow	\downarrow
\Rightarrow	\Longrightarrow	\Downarrow
\leftrightarrow	\longleftrightarrow	\updownarrow
\Leftrightarrow	\Longleftrightarrow	\Updownarrow
\mapsto	\longmapsto	\nearrow
\searrow	\swarrow	\nwarrow
\hookrightarrow	\hookrightarrow	

8.8 Alternativní příkazy

\neq	$\{$	$\{$	\wedge	\land	\rightarrow	mediummedium	$ $	\vert
\leq	\leq	$\}$	\vee	\lor	\leftarrow	\gets	$\ $	\Vert
\geq	\geq	\supseteq	\neg	\lnot				

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

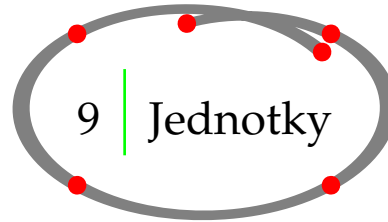
vyhledej

+

krok zpět

-

+



Abychom se přiměli používat míry a jednotky v celém dokumentu jednotně, můžeme vytvořit vlastní seznam těchto jednotek a umístit ho do deklarační oblasti (před příkaz `\starttext`) vstupního souboru.

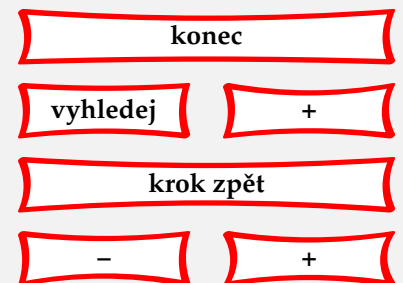
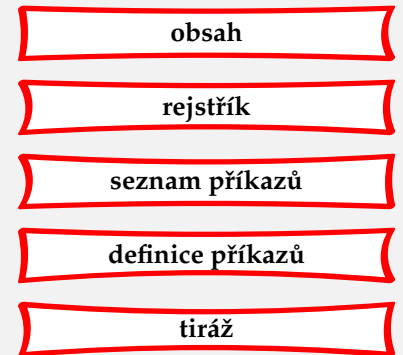
V `CONTEXTU` je dostupný vnější modul, který obsahuje téměř všechny jednotky SI. Pokud se na tento modul odvoláme příkazem `\usemodule[units]`, můžeme pak používat jednotky následujícím způsobem:

```
\Meter \Per \Square \Meter
\Cubic \Meter \Per \Sec
\Square \Milli \Meter \Per \Inch
\Centi \Liter \Per \Sec
\Meter \Inverse \Sec
\Newton \Per \Square \Inch
\Newton \Times \Meter \Per \Square \Sec
```

Trocha více psaní je vyvážena zárukou jednotného přístupu v užívání jednotek. Příkaz `\unit` také zaručuje, že od sebe nebudou odděleny hodnota a její jednotka. Pokud se totiž hodnota nachází na konci řádku a následující řádek začíná měrnou jednotkou, má to k dokonalosti daleko. Náš předchozí příklad se vysází takto:

```
m/m2
m3/s
mm2/inch
cl/s
ms-1
N/inch2
N·m/s2
```

Můžeme také nadefinovat své vlastní jednotky:

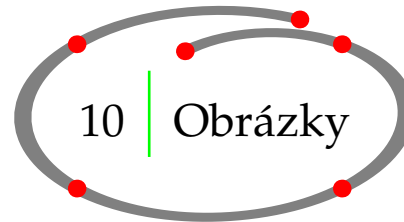


```
\unit [Ounce] {oz}{}
```

Později ve svém dokumentu můžeme napsat `15.6 \Ounce`, což bude vysázeno jako 15.6 oz.

Pro zapsání % a ‰ jednotným způsobem slouží dva zvláštní příkazy:

```
\percent
\permille
```



Obrázky a fotografie vložíme do dokumentu následujícím příkazem:

```
\placefigure
[] [obr:kostel]
{Kostel sv.~Štěpána}
{\externalfigure[ma-cb-24] [width=.4\textwidth]}
```

Po zpracování se na první dostupné místo vysází **obrázek 10.1**

Příkaz `\placefigure` se postará o číslování a vertikální odsazení obrázku. Dále tento příkaz inicializuje mechanismus umístění plovoucích objektů, což znamená, že se `CONTEXT` podívá, zda je na dané stránce pro náš obrázek dostatek místa. Pokud není, bude obrázek umístěn na jiné vhodné místo. V takovém případě bude text pokračovat a obrázek bude proplovat naším dokumentem, dokud pro něj nebude nalezeno optimální umístění. Tento mechanismus můžeme ovlivnit parametrem v první dvojici závorek.

Příkaz `\placefigure` je předdefinovanou instancí:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+



Obrázek 10.1 Kostel sv. Štěpána

```
\placefloat [.1.] [...2,...] {...3} {...4}
```

OPTIONAL OPTIONAL

1 left right here top bottom inleft inright inmargin margin leftmargin
rightmargin leftedge rightedge innermargin outermargin inneredge outeredge
inner outer line high low fit page leftpage rightpage opposite always
auto force tall reset line height depth split

2 REFERENCE

3 TEXT

4 TEXT

Zde použitelné parametry jsou popsány v **tabulce 10.1**.

Druhý pár hranatých závorek je určen pro referenční jméno odkazu. Na konkrétní obrázek se můžete odkázat příkazem:

```
\in{obrázek}[obr:kostel]
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Parametr	Vysvětlení
here	pokud je to možné, umístí obrázek na toto místo
force	vynutí umístění obrázku na tomto místě
page	umístí obrázek na zvláštní stránku
top	umístí obrázek k hornímu okraji stránky
bottom	umístí obrázek k dolnímu okraji stránky
left	umístí obrázek k levému okraji
right	umístí obrázek k pravému okraji
margin	umístí obrázek do marginálie

Tabulka 10.1 Parametry v příkazu `\placefigure`.

První pár složených závorek tohoto příkazu obsahuje popis obrázku. Můžeme tam napsat libovolný text. Pokud si nepřejeme žádný popis, ani číslování, můžeme použít `{none}`. Popisek je formátován pomocí `\setupcaptions` a nastavení (případně vynulování) číslování se řídí příkazem `\setupnumbering` (viz **sekce 37.2**).

Druhý pár závorek je pro vložení vlastního obrázku. Nejčastěji zde uvádíme příkaz pro vložení externího souboru `\externalfigure`, viz níže.

V následujícím příkladu vidíme použití funkce `\placefigure{}{}`. Všimněme si obsahu druhé závorky, nutné pro vysázení `Hasselt`.

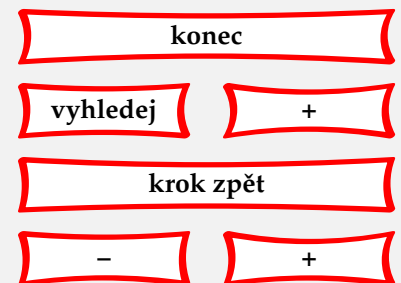
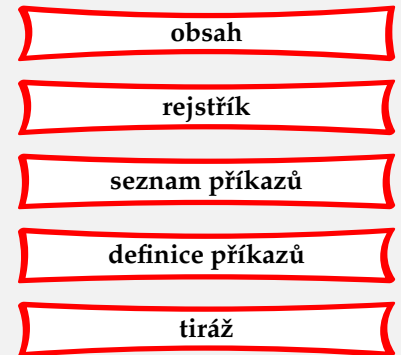
```
\placefigure
  {Orámovaný Hasselt}
  {\framed{\tfd Hasselt}}
```

Tento příklad vysází:

Hasselt

Obrázek 10.2
Orámovaný Hasselt

Obrázky nicméně často vytváříme v programech jako Corel Draw nebo Illustrator. Fotky – po oskenování – se obvykle upraví v GIMPu nebo PhotoShopu. Po této práci máme obrázky ve formě



souborů. CONTEXT podporuje všechny formáty souborů, se kterými umí nakládat ovladač v pozadí CONTEXTu. Pokud používáme PDFTEX, můžeme vkládat JPG, PNG a jednotlivé stránky ze souborů PDF jakož i výstup METAPOSTu, (soubory typu MPS). Uživatelé obvykle mohou CONTEXTu důvěřovat, že nalezne nejlepší možný formát souboru.

Na **obrázku 10.3** vidíme fotografii a vektorovou grafiku kombinovanou do jednoho obrázku.



oskenovaný obrázek



vektorová grafika

Obrázek 10.3 Hasseltské kanály.

Tento obrázek můžeme vysázet pomocí kódu podobného tomuto:

```
\placefigure
[here,force]
[fig:canals]
{Hasseltské kanály.}
{\startcombination[2*1]
{\externalfigure[ma-cb-03][width=.4\textwidth]}
{oskenovaný obrázek}
{\externalfigure[ma-cb-00][width=.4\textwidth]}
{vektorová grafika}
\stopcombination}
```

V tomto jednom plovoucím objektu jsou dva obrázky kombinovány pomocí:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+


```
\startcombination [...] ... \stopcombination
* N*M
```

Dvojice příkazů `\startcombination ... \stopcombination` je užívána pro kombinaci dvou obrázků do jednoho plovoucího objektu. Pomocí hranatých závorek můžeme ovlivnit vysázení většího množství obrázků. Jeden obrázek pod druhým se vysází pomocí `[1*2]`. Můžeme si představit, co se stane, když zkombinujeme 6 obrázků pomocí `[3*2]` ([řádky*sloupce]).

Uvedené příklady jsou dostačující pro vytváření ilustrovaných dokumentů. Nicméně někdy je potřeba více ovlivnit uspořádání obrázku a textu. Pro tento účel můžete použít:

```
\startframedtext [1.] [.,.2.,...] {3.} {4.} ... \stopframedtext
          OPTIONAL      OPTIONAL
1 left right middle none
2 inherits from \setupframedtexts
3 TEXT
4 TEXT
```

Místo slova `framed` můžeme použít dříve definovaný rámovaný objekt. Obrázek (`figure`) a tabulka (`table`) jsou již předdefinované, jak je vidět z následujícího příkladu kombinujícího obrázek s textem.

```
\startfiguretext
  [left]
  [fig:citizens]
  {none}
  {\externalfigure[ma-cb-18][width=.5\makeupwidth]}
Počet obyvatel Hasseltu se odjakživa měnil v~závislosti
na ekonomických podmínkách. Příkladem může být kopání
kanálu Dedemsvaart kolem roku 1810. Protože vedl přímo
skrz Hasselt, obchod vzkvétal.
To vedlo k růstu populace o~téměř 40\,\% během 10~let.
V~současnosti již Dedemsvaart nemá obchodní význam;
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

zdejší kanály se staly turistickou atrakcí. Ale
upomínky na dřívější časy prosperujícího obchodu,
mohou být vidět všude.

`\stopfiguretext`

Po vysázení uvidíme:



Počet obyvatel Hasseltu se odjakživa měnil
v závislosti na ekonomických podmínkách.
Příkladem může být kopání kanálu Dedem-
svaart kolem roku 1810. Protože vedl přímo
skrz Hasselt, obchod vzkvétal. To vedlo k
růstu populace o téměř 40% během 10 let.
V současnosti již Dedemsvaart nemá obchod-
ní význam; zdejší kanály se staly turistickou
atrakcí. Ale upomínky na dřívější časy prosperu-
jícího obchodu, mohou být vidět všude.

Poslední pár složených závorek většinou obsahuje příkaz `\externalfigure`.

```
\externalfigure [..1.] [...,2,...]
```

1 FILE

2 *inherits from* \useexternalfigure

Tento příkaz nám dává svobodu vysázet obrázky jakýmkoliv způsobem si budeme přát.
`\externalfigure` obsahuje dva páry hranatých závorek. První se užívá pro přesný název souboru bez přípony, druhý pro požadované rozměry a ev. formát souboru. Není obtížné si představit co způsobí vysázení zdrojového textu:⁴

```
\inmargin  
{\externalfigure
```

⁴ Viz stranu 78.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
[ma-cb-23]
[width=\marginwidth]}
```

Umístění a rozměry obrázku můžeme ovlivnit pomocí:

```
\setupfloats [...,*,...]
```

```
* location          = left right middle
width              = fit DIMENSION
before            = COMMAND
after             = COMMAND
margin           = DIMENSION
spacebefore      = small medium big none
spaceafter       = small medium big none
sidespacebefore = small medium big none
sidespaceafter  = small medium big none
indentnext       = yes no
ntop             = NUMBER
nbottom         = NUMBER
nlines          = NUMBER
default         = IDENTIFIER
tolerance        = 0 1 2
leftmargindistance = DIMENSION
rightmargindistance = DIMENSION
sidealign       = normal line
numbering       = yes nocheck
inherits from \setupframed
```

Můžeme také nastavit číslování obrázků a jejich popisek:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupcaptions [...,*.,...]
```

```
* location = top bottom none high low middle
width = fit broad max DIMENSION
minwidth = fit DIMENSION
headstyle = normal bold slanted boldslanted type cap small... COMMAND
style = normal bold slanted boldslanted type cap small... COMMAND
number = yes no
inbetween = COMMAND
align = inner outer left right middle normal no yes
conversion = numbers characters Characters romannumerals Romannumerals
way = bytext bycd:section
separator = TEXT
stopper = TEXT
command = COMMAND
distance = DIMENSION
```

Následující příkazy patří do deklarační oblasti vstupního souboru a mají tedy vliv na všechny plovoucí objekty.

```
\setupfloats
  [location=right]
\setupcaptions
  [location=top,
   style=boldslanted]
\placefigure
  {Typický obrázek Hasseltu.}
  {\externalfigure[ma-cb-12] [width=8cm]}
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

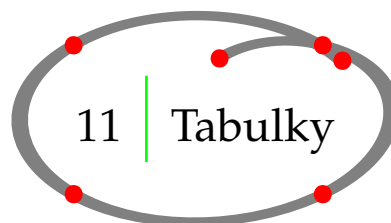
+

krok zpět

-

+

Obrázek 10.4 Typický obrázek Hasseltu.



- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +

Tabulka se obecně skládá ze sloupců. Jejich obsah může být zarovnán vlevo, vpravo, na střed či k desetinné čárce. Záhlaví může zaujímat jeden nebo více sloupců. Buňky tabulky mohou obsahovat rovnice nebo několik řádků textu. Horizontální a vertikální linky mohou vést přes celou tabulku nebo jen přes její část.

Toto napsal J. Wichura v předmluvě manuálu o `TABLE` (`TABLE manual`, 1988). Michael Wichura je také autorem jedné ze sady maker `TABLE`, kterou `CONTEXT` používá k sazbě tabulek.⁵

Do definic `TABLE` bylo oproti originálu přidáno několik maker zajišťujících konzistentnější řádkování a bylo zjednodušeno uživatelské rozhraní.⁶

Tabulku umístíme příkazem `\placetable`. Definujeme ji pomocí

```
\starttable [...] ... \stoptable
* TEXT IDENTIFIER
```

Definice tabulky může vypadat následovně:

```
\placetable[here][tab:ships]{Lodě kotvící v~Hasseltu.}
\starttable[|c|c|]
\HL
\NC \bf Rok \NC \bf Počet lodí \NC\SR
\HL
\NC 1645 \NC 450 \NC\FR
\NC 1671 \NC 480 \NC\MR
\NC 1676 \NC 500 \NC\MR
\NC 1695 \NC 930 \NC\LR
\HL
\stoptable
```

Tato tabulka je vysázena jako **tabulka 11.1**.

⁵ `CONTEXT` nabízí ještě jiné sady. Velmi zajímavé jsou „přirozené tabulky“ `\bTABLE`. Umožňují podobnou syntaxi jako HTML tabulky s velkou variabilitou sdružování buněk a bohatými grafickými možnostmi. Řádkování je však méně přirozené. Viz manuál [enattab.pdf](#) a magazín [NaturalTables.pdf](#). Pozn. překl.

⁶ `CONTEXT` byl vyvinut (také) pro uživatele bez technického nadání v éře WYSIWIG. Proto bylo doplněno přívětivé uživatelské rozhraní se snadnou manipulací se soubory a byly vypuštěny kryptické příkazy a programové i logické konstrukce.

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +

Rok	Počet lodí
1645	450
1671	480
1676	500
1695	930

Tabulka 11.1 Lodě kotvící v Hasseltu.

První příkaz `\placetable` má stejnou funkci jako `\placefigure`. Postará se o vertikální odsazení tabulky a o její číslo. Dále spustí mechanismus umístění plovoucího objektu a ten nalezne optimální polohu tabulky na stránce.

Položky tabulky se zadávají dovnitř dvojice `\starttable ... \stoptable`. Do hranatých závorek vkládáme značky pro formátování tabulky, viz **tabulku 11.2**.

Značka	Význam
	oddělovač sloupců
c	sloupec se zarovnáním na střed
l	sloupec se zarovnáním vlevo
r	sloupec se zarovnáním vpravo
s<n>	mezisloupcová mezera o hodnotě $n = 0, 1, 2$
w<rozměr>	minimální šířka sloupce o dané hodnotě

Tabulka 11.2 Značky pro formátování tabulky.

Jak jsme již viděli, existují kromě formátovacích značek ještě formátovací příkazy. Vkládají se přímo do jednotlivých buněk. Ty základní ukazuje **tabulka 11.3**.

V dalších příkladech uvidíme formátovací příkazy `CONTEXTU`. Jejich jména jsou delší a méně kryptická a řídí další formátovací funkce tabulek. V **tabulce 11.4** je jejich přehled.

Tabulky ukázané dále jsou uvedené se zdrojovými texty. Pro složitější ukázky si lze také přečíst `TABLE` manuál M. J. Wichury.

```
\placetable
[here,force]
```

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +

Příkaz	Význam
<code>\NR</code>	další řádka udělej řádku bez vertikálního vyrovnání
<code>\FR</code>	první řádka udělej řádku s horním vyrovnáním
<code>\LR</code>	poslední řádka udělej řádku s dolním vyrovnáním
<code>\MR</code>	prostřední řádka udělej řádku s horním i dolním vyrovnáním
<code>\SR</code>	oddělovací řádka udělej řádku s horním i dolním vyrovnáním
<code>\VL</code>	svislá linka nakresli svislou linku a jdi na další sloupec
<code>\NC</code>	další sloupec jsi na další sloupec
<code>\HL</code>	vodorovná linka nakresli vodorovnou linku
<code>\DL</code>	oddělovací linka* nakresli linku přes následující sloupec
<code>\DL[n]</code>	oddělovací linka* nakresli oddělovací linku přes <i>n</i> sloupců
<code>\DC</code>	oddělovací linka* udělej mezeru přes následující sloupec
<code>\DR</code>	oddělovací řádka* udělej řádku s horním i dolním vyrovnáním
<code>\LOW{text}</code>	— napiš <i>text</i> níže
<code>\TWO, \THREE</code> etc. —	udělej mezeru přes <i>dva, tři, ...</i> sloupce

Příkazy * `\DL`, `\DC` a `\DR` lze kombinovat.

Tabulka 11.3 Příkazy pro formátování tabulky.

Příkaz	Význam
<code>\JustLeft</code>	zarovnej vlevo a ignoruj formátování sloupce
<code>\JustRight</code>	zarovnej vpravo a ignoruj formátování sloupce
<code>\JustCenter</code>	vycentruj a ignoruj formátování sloupce
<code>\SetTableToWidth{}</code>	nastav pevnou šířku tabulky
<code>\use{n}</code>	přeskoč příštích <i>n</i> sloupců

Tabulka 11.4 `CONTEXT`ové formátovací příkazy tabulky.

```
[tab:effects of commands]
{Vliv formátovacích příkazů.}
{\startcombination[2*1]
  {\starttable[|c|c|]
    \HL
    \VL \bf Rok \VL \bf Počet obyvatel \VL\SR
    \HL
    \VL 1675 \VL ~428 \VL\FR
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej +

krok zpět

- +


```

\VL 1795      \VL 1124      \VL\MR
\VL 1880      \VL 2405      \VL\MR
\VL 1995      \VL 7408      \VL\LR
\HL
\stoptable}{standardní}
{\starttable[|c|c|]
\HL
\VL \bf Rok \VL \bf Počet obyvatel \VL\NR
\HL
\VL 1675      \VL ~428      \VL\NR
\VL 1795      \VL 1124      \VL\NR
\VL 1880      \VL 2405      \VL\NR
\VL 1995      \VL 7408      \VL\NR
\HL
\stoptable}{pouze \type{\NR}}
\stopcombination}

```

Ve výše uvedeném příkladu jsou v první tabulce použity \SR, \FR, \MR a \LR. Tyto příkazy se starají o vkládání mezer uvnitř tabulky. Jak je vidět, příkaz \NR pouze začíná novou řádku.

Rok	Počet obyvatel
1675	428
1795	1124
1880	2405
1995	7408

standardní

pouze \NR

Tabulka 11.5 Vliv formátovacích příkazů.

V dalším příkladě je ukázáno vyrovnání sloupců pomocí značek s0 a s1.

```

\startbuffer[one]
\starttable[|c|c|]
\HL
\VL \bf Rok \VL \bf Obyvatel \VL\SR
\HL

```

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej
- +
- krok zpět
-
- +

```

\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer

\startbuffer[two]
\starttable[s0 | c | c |]
\HL
\VL \bf Rok \VL \bf Obyvatel \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer

\startbuffer[three]
\starttable[| s0 c | c |]
\HL
\VL \bf Rok \VL \bf Obyvatel \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer

\startbuffer[four]
\starttable[| c | s0 c |]
\HL

```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\VL \bf Rok \VL \bf Obyvatel \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer
\startbuffer[five]
\starttable[s1 | c | c |]
\HL
\VL \bf Rok \VL \bf Obyvatel \VL\SR
\HL
\VL 1675 \VL ~428 \VL\FR
\VL 1795 \VL 1124 \VL\MR
\VL 1880 \VL 2405 \VL\MR
\VL 1995 \VL 7408 \VL\LR
\HL
\stoptable
\stopbuffer
\placetable
[here,force]
[tab:example-format-commands]
{Vliv formátovacích značek.}
{\startcombination[3*2]
  {\getbuffer[one]} {standardní}
  {\getbuffer[two]} {\type{s0}}
  {\getbuffer[three]} {\type{s0} ve sloupci~1}
  {\getbuffer[four]} {\type{s0} ve sloupci~2}
  {\getbuffer[five]} {\type{s1}}
  {} {}
\stopcombination}

```

Po zpracování těchto příkladů dostaneme **tabulku 11.6**. Implicitní tabulka má mezerování sloupců pomocí značek s2.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Rok	Obyvatel
1675	428
1795	1124
1880	2405
1995	7408

standardní

Rok	Obyvatel
1675	428
1795	1124
1880	2405
1995	7408

s0

Rok	Obyvatel
1675	428
1795	1124
1880	2405
1995	7408

s0 ve sloupci 1

Rok	Obyvatel
1675	428
1795	1124
1880	2405
1995	7408

s0 ve sloupci 2

Rok	Obyvatel
1675	428
1795	1124
1880	2405
1995	7408

s1

Tabulka 11.6 Vliv formátovacích značek.

Sloupce bývají občas odděleny svislými linkami | a řádky vodorovnými.

```
\placetable
[here,force]
[tab:divisions]
{Vyvolání dojmu volby.}
\starttable[|c|c|c|]
\NC Steenwijk \NC Zwartsluis \NC Hasselt \NC\SR
\DC \DL \DC \DR
\NC Zwartsluis \VL Hasselt \VL Steenwijk \NC\SR
\DC \DL \DC \DR
\NC Hasselt \NC Steenwijk \NC Zwartsluis \NC\SR
\stoptable
```

Praktičtější příklad je v **tabulce 11.8**.

```
\placetable
[here,force]
[tab:example contextcommands]
```

obsah
rejstřík
seznam příkazů
definice příkazů
tiráž

konec	
vyhledej	+
krok zpět	
-	+

Steenwijk	Zwartsluis	Hasselt
Zwartsluis	Hasselt	Steenwijk
Hasselt	Steenwijk	Zwartsluis

Tabulka 11.7 Vyvolání dojmu volby.

```
{Vliv formátovacích příkazů \CONTEXT{u.}
\starttable[|l|c|c|c|]
\HL
\VL \FIVE \JustCenter{Volby do městské rady v~roce 1994} \VL\SR
\HL
\VL \LOW{Strana} \VL \THREE{Okrsky} \VL \LOW{Celkem} \VL\SR
\DC \DL[3] \DC \DR
\VL \VL 1 \VL 2 \VL 3 \VL \VL\SR
\HL
\VL PvdA \VL 351 \VL 433 \VL 459 \VL 1243 \VL\FR
\VL CDA \VL 346 \VL 350 \VL 285 \VL ~981 \VL\MR
\VL VVD \VL 140 \VL 113 \VL 132 \VL ~385 \VL\MR
\VL HKV/RPF/SGP \VL 348 \VL 261 \VL 158 \VL ~767 \VL\MR
\VL GPV \VL 117 \VL 192 \VL 291 \VL ~600 \VL\LR
\HL
\stoptable
```

V posledním sloupci je znak ~ použit k simulaci čtyřciferného čísla. Znak ~ má šířku cifry.

Někdy se nám tabulka příliš rozroste. Abychom ji vysadili s textem, můžeme například zmenšit použité písmo. Pomocí příkazu

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Volby do městské rady v roce 1994				
Strana	Okrsky			Celkem
	1	2	3	
PvdA	351	433	459	1243
CDA	346	350	285	981
VVD	140	113	132	385
HKV/RPF/SGP	348	261	158	767
GPV	117	192	291	600

Tabulka 11.8 Vliv formátovacích příkazů `CONTEXTu`.

```
\setuptables [...,*,...]
```

```
* distance      = small medium big
bodyfont        = 5pt ... 12pt small big
HL              = NUMBER small medium big none
VL              = NUMBER small medium big none
depth           = NUMBER strut
height          = NUMBER strut
rulethickness   = DIMENSION
rulecolor       = IDENTIFIER
align           = inner outer left right middle normal no yes
commands        = COMMAND
align           = inner outer left right middle normal no yes
background      = screen color none
backgroundscreen = NUMBER
backgroundcolor = IDENTIFIER
bodyfont        = 5pt ... 12pt small big
```

to uděláme takto:

```
\placetable
[here,force]
[tab:setuptable]
{Použití \type{\setuptables}.}
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

{\startcombination[1*3]
{\setuptables[bodyfont=10pt]
\starttable[|c|c|c|c|c|c|]
\HL
\VL \use6
  \JustCenter{Pokles bohatství v~holandských guldenech (Df1)}
  \VL\SR
\HL
\VL Year \VL 1.000--2.000
  \VL 2.000--3.000
  \VL 3.000--5.000
  \VL 5.000--10.000
  \VL přes 10.000 \VL\SR
\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptable}{\tt bodyfont=10pt}
{\setuptables[bodyfont=8pt]
\starttable[|c|c|c|c|c|c|]
\HL
\VL \use6
  \JustCenter{Pokles bohatství v~holandských guldenech (Df1)}
  \VL\SR
\HL
\VL Year \VL 1.000--2.000
  \VL 2.000--3.000
  \VL 3.000--5.000
  \VL 5.000--10.000
  \VL over 10.000 \VL\SR
\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR

```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

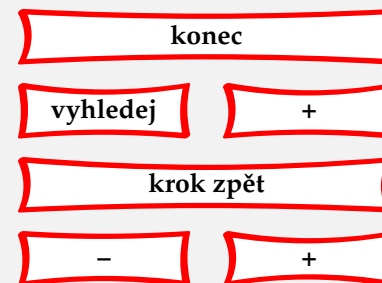
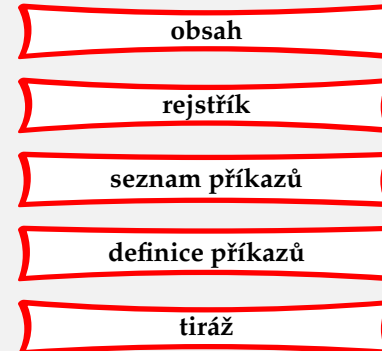
+

```

\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptable}{\tt bodyfont=8pt}
{\setuptables[bodyfont=6pt,distance=small]}
\starttable[|c|c|c|c|c|c|]
\HL
\VL \use6
\JustCenter{Pokles bohatství v~holandských guldenech (Dfl)}
\VL\SR
\HL
\VL Year \VL 1.000--2.000
\VL 2.000--3.000
\VL 3.000--5.000
\VL 5.000--10.000
\VL přes 10.000 \VL\SR
\HL
\VL 1675 \VL 22 \VL 7 \VL 5 \VL 4 \VL 5 \VL\FR
\VL 1724 \VL ~4 \VL 4 \VL -- \VL 4 \VL 3 \VL\MR
\VL 1750 \VL 12 \VL 3 \VL 2 \VL 2 \VL -- \VL\MR
\VL 1808 \VL ~9 \VL 2 \VL -- \VL -- \VL -- \VL\LR
\HL
\stoptable}{\tt bodyfont=6pt,distance=small}
\stopcombination}

```

Nastavit umístění tabulek můžeme také pomocí již známého



Pokles bohatství v holandských guldenech (Dfl)					
Year	1.000–2.000	2.000–3.000	3.000–5.000	5.000–10.000	přes 10.000
1675	22	7	5	4	5
1724	4	4	–	4	3
1750	12	3	2	2	–
1808	9	2	–	–	–

bodyfont=10pt

Pokles bohatství v holandských guldenech (Dfl)					
Year	1.000–2.000	2.000–3.000	3.000–5.000	5.000–10.000	over 10.000
1675	22	7	5	4	5
1724	4	4	–	4	3
1750	12	3	2	2	–
1808	9	2	–	–	–

bodyfont=8pt

Pokles bohatství v holandských guldenech (Dfl)					
Year	1.000–2.000	2.000–3.000	3.000–5.000	5.000–10.000	přes 10.000
1675	22	7	5	4	5
1724	4	4	–	4	3
1750	12	3	2	2	–
1808	9	2	–	–	–

bodyfont=6pt,distance=small

Tabulka 11.9 Použití \setuptables.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej +

krok zpět

– +

```
\setupfloats [...,*.,...]
```

```
* location          = left right middle
width              = fit DIMENSION
before            = COMMAND
after             = COMMAND
margin           = DIMENSION
spacebefore      = small medium big none
spaceafter       = small medium big none
sidespacebefore = small medium big none
sidespaceafter  = small medium big none
indentnext      = yes no
ntop            = NUMBER
nbottom        = NUMBER
nlines         = NUMBER
default        = IDENTIFIER
tolerance      = 0 1 2
leftmargindistance = DIMENSION
rightmargindistance = DIMENSION
sidealign      = normal line
numbering      = yes nocheck
inherits from \setupframed
```

Číslování a popisky nastavíme příkazem

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupcaptions [...,*.*,...]
```

```
* location = top bottom none high low middle
width = fit broad max DIMENSION
minwidth = fit DIMENSION
headstyle = normal bold slanted boldslanted type cap small... COMMAND
style = normal bold slanted boldslanted type cap small... COMMAND
number = yes no
inbetween = COMMAND
align = inner outer left right middle normal no yes
conversion = numbers characters Characters romannumerals Romannumerals
way = bytext bycd:section
separator = TEXT
stopper = TEXT
command = COMMAND
distance = DIMENSION
```

Tyto příkazy používáme v deklarační oblasti našeho vstupního souboru a mají globální platnost.

```
\setupfloats[location=left]
\setupcaption[style=boldslanted]
\placetable[here][tab:opening hours]{Otevírací hodiny knihovny.}
\starttable[|l|c|c|]
\HL
\VL \bf Den \VL \use2 \bf Otevírací hodiny \VL\SR
\HL
\VL pondělí \VL 14.00 - 17.30 \VL 18.30 - 20.30 \VL\FR
\VL úterý \VL \VL \VL \VL\MR
\VL středa \VL 10.00 - 12.00 \VL 14.00 - 17.30 \VL\MR
\VL čtvrtek \VL 14.00 - 17.30 \VL 18.30 - 20.30 \VL\MR
\VL pátek \VL 14.00 - 17.30 \VL \VL\MR
\VL sobota \VL 10.00 -- 12.30 \VL \VL\LR
\HL
\stoptable
```

Výsledek zobrazuje **tabulka 11.10**.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

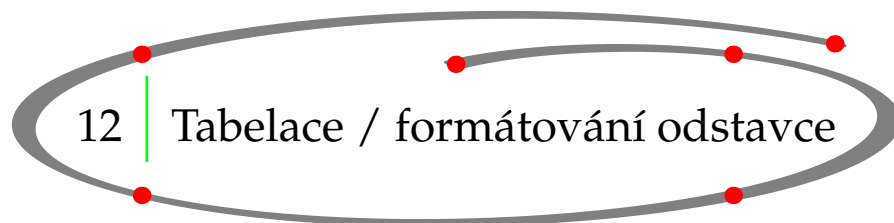
krok zpět

-

+

Den	Otevírací hodiny	
pondělí	14.00 – 17.30	18.30 – 20.30
úterý		
středa	10.00 – 12.00	14.00 – 17.30
čtvrtek	14.00 – 17.30	18.30 – 20.30
pátek	14.00 – 17.30	
sobota	10.00 – 12.30	

Tabulka 11.10 Otevírací hodiny knihovny.



Někdy chceme vysázet odstavce zformátované určitým přesně daným způsobem. Toho lze dosáhnout pomocí:

```
\starttabulate [...] [...] ... \stoptabulate
      OPTIONAL  OPTIONAL
1  TEXT
2  TEXT
```

Technika tabelace je úzce svázána s technikou tvorby tabulek. Tabelaci lze využít v těch případech, kdy chcete do buňky tabulky vysázet kompletní odstavec. Tabelace funguje dobře i při stránkovém zlomu.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Definice tabelace vypadá nějak takto:

```
\starttabelle[|w(1cm)B|p(.4\textwidth)|p|]
\NC 1252
  \NC Hasselt získává městská privilegia od biskupa Hendrika
    van Viandena.
  \NC Ostatní města na Hendrika van Viandena naléhala,
    aby tato privilegia neschvaloval. Hasseltu trvalo
    dlouho, než biskupa přesvědčil. Poté, co biskupovi
    poskytl pomoc v~konfliktu proti Drenthe, byla listina
    vydána. \NC\NR
\NC 1350
  \NC Hasselt se připojuje k~Hanze, aby tak ochránil svůj
    mezinárodní obchod.
  \NC Hanza měla pro obchodníky v~Hasseltu veliký význam.
    Ze zboží se tehdy platilo clo v~každém městě,
    na každé silnici nebo mostě. Poté, co se Hasselt
    stal členem Hanzy, byly mu k~dispozici nezaplatněné
    cesty po celé Evropě. Přes tento veliký význam Hanzy
    zůstal Hasselt vždy jejím méně významným členem. \NC\NR
\stoptabelle
```

V tomto případě má první sloupec šířku 1 cm a je vysázen tučným řezem (B). Druhý sloupec má šířku 40 % šířky odstavce a je vysázen jako odstavec. Zbývající horizontální místo je využito posledním odstavcem.

Uvedený příklad se vysází takto:

obsah
rejstřík
seznam příkazů
definice příkazů
tiráž

konec	
vyhledej	+
krok zpět	
-	+

1252	Hasselt získává městská privilegia od biskupa Hendrika van Viandena.	Ostatní města na Hendrika van Viandena naléhala, aby tato privilegia neschvaloval. Hasseltu trvalo dlouho, než biskupa přesvědčil. Poté, co biskupovi poskytl pomoc v konfliktu proti Drenthe, byla listina vydána.
1350	Hasselt se připojuje k Hanze, aby tak ochránil svůj mezinárodní obchod.	Hanza měla pro obchodníky v Hasseltu veliký význam. Ze zboží se tehdy platilo clo v každém městě, na každé silnici nebo mostě. Poté, co se Hasselt stal členem Hanzy, byly mu k dispozici nezaplatněné cesty po celé Evropě. Přes tento veliký význam Hanzy zůstal Hasselt vždy jejím méně významným členem.

Stejně jako v tabulkách i zde je k dispozici mnoho formátovacích povelů a parametrů. Jejich seznam je uveden v **tabulce 12.1**.

l	zarovnání vlevo	I	<i>italika</i>
c	zarovnání na střed	R	antikva
r	zarovnání vpravo	S	<i>skloněné</i>
in	odsazení vlevo	T	psací stroj
jn	odsazení vpravo	m	vnitřní matematický mód
kn	mezera okolo	M	display matematický mód
w(d)	1 řádka, pevná šířka	f \command	určení řezu písma
p(d)	odstavec, pevná šířka	barg . .	umístit . . před položku
p	odstavec, maximální šířka	aarg . .	umístit . . za položku
B	tučný řez	h \command	aplikovat na položku \command

Tabulka 12.1 Povelý pro tabelaci.

Jiný příklad formátování odstavce by mohl vypadat takto:

```
\definetabulate[ChemOdstavec] [[l|p|l|]]
\startChemOdstavec
\NC Vápenky
  \NC Hasselt ma své vlastní vápenky, které byly vystavěny
    v~roce 1504 a pálené vápno se v~nich vyrábělo až do
```

obsah
rejstřík
seznam příkazů
definice příkazů
tiráž

konec	
vyhledej	+
krok zpět	
-	+

roku 1956. Dnes slouží jako turistická atrakce.
`\NC \chemical{CaCO_3,~,GIVES,~,CaO,~,+,~,CO_2} \NC\NR`
`\stopChemOdstavec`

Vysází se to následovně:

Vápenky Hasselt ma své vlastní vápenky, které byly vystavěny v ro- $\text{CaCO}_3 \rightarrow \text{CaO} + \text{CO}_2$
 ce 1504 a pálené vápno se v nich vyrábělo až do roku 1956.
 Dnes slouží jako turistická atrakce.

Modul pro chemickou sazbu je vysvětlen v jiném manuálu, protože ne každý se zajímá o chemii.

Také jsme zde uvedli povel pro definici tabulačního schématu ChemOdstavec

```
\definetabulate [.1.] [.2.] [.3.]
                OPTIONAL OPTIONAL
1 IDENTIFIER
2 IDENTIFIER
3 TEXT
```

a existuje také:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\setuptabulate [..1.] [..,2.,..]
                OPTIONAL
1 IDENTIFIER
2 unit          = DIMENSION
  indenting     = never none not no yes always first next small medium
                big normal odd even DIMENSION
  before        = COMMAND
  after         = COMMAND
  inner         = COMMAND
  EQ            = TEXT
  rulecolor     = IDENTIFIER
  align         = inner outer left right middle normal no yes
  rulethickness = DIMENSION
  distance      = blank grid depth DIMENSION small medium big none
  bodyfont      = 5pt ... 12pt small big
  rule          = normal line
  split        = yes no
    
```

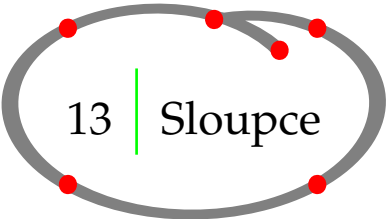
obsah

rejstřík

seznam příkazů

definice příkazů

tiráž



Nepříliš složité dokumenty je možné sázet do sloupců. Jestliže kus textu zahájíme příkazem `\startcolumns` a ukončíme `\stopcolumns`, veškerý takto obklopený materiál bude formátován do sloupců.⁷

konec

vyhledej

+

krok zpět

-

+

⁷ Složitější dokument, např. s plovoucími obrázky, je lépe sázet pomocí `\startcolumnsets ... \stopcolumnsets`. Popis lze najít v manuálu [columns.pdf](#), viz přílohu A. Pozn. překl.


```
\startcolumns [...,*.,...] ... \stopcolumns
                OPTIONAL
* inherits from \setupcolumns
```

```
\startcolumns[n=3,tolerance=verytolerant]
```

Hasselt je staré hanzovní město, položené 12\,km severně od Zwolle na řece Zwartewater.

Má bohatou historii, která se začíná kolem roku 1252, kdy Hasselt obdržel městská práva. Tuto historii můžeme rozkrývat pomocí velkého počtu památek v~centru města.

Nalezneme zde pozdněgotický kostel Sv. Štěpána založený roku 1479 s~velkolepými varhanami. Budova bývalé radnice, která se nalézá na Tržním náměstí, byla postavena mezi roky 1500 a~1550 a dnes v~něm sídlí muzeum zbraní. Za zmínku stojí zejména sbírka děl na černý střelný prach (haakhussen), která se řadí k největším na světě.

Dále je zde větrný mlýn \quote{De Zwaluw} (Vlaštovka) na mletí mouky z~roku 1748, unikátní nábřeží \quote{Stenendijk} a~v~Evropě poslední stará plně funkční vápenka.

Centrum města s~kanály lemovanými lípami, oblast Van Stolkspark a dok plný ruchu a~hemžení jsou ideální místa k~procházkám.

Okolí Hasseltu stojí ovšem též za zmínku. V~zimě hostí polder Mastenbroek velká stáda ovcí. V~létě tvoří vesničky Genne, Streukel a Celle\~mui\~den s~velice vzácnou květinou řebčíkem kostkovaným (lat. *Fritillaria meleagris*) výborné prostředí pro pěší i~cyklistické výlety.

Hasselt je také velice důležité město vodních sportů. Jezera severozápadně od Overijssel, řeky IJssel, Overijsselse Vecht a Randmeren jsou snadno dosažitelné z~přístavu jacht \quote{De Molenwaard}. Mořeplavbu, rybolov, plavání a jízdu na kanoi si v~Hasseltu můžete vychutnat naplno.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

A~na závěr uvedeme jednu událost obzvláštního významu. Každoročně se na konci srpna v~Hasseltu slaví dožínky - \quote{Eui Festival}.

\stopcolumns

Výsledkem bude třísloupcový text.

Hasselt je staré hanzovní město, položené 12 km severně od Zwolle na řece Zwartewater.

Má bohatou historii, která se začíná kolem roku 1252, kdy Hasselt obdržel městská práva. Tuto historii můžeme rozkrývat pomocí velkého počtu památek v centru města.

Nalezneme zde pozdněgotický kostel Sv. Štěpána založený roku 1479 s velkolepými varhanami. Budova bývalé radnice, která se nalézá na Tržním náměstí, byla postavena mezi roky 1500 a 1550 a dnes v něm sídlí muzeum zbraní. Za zmínku stojí zejména sbírka děl na černý stříelný prach (ha-

akhusen), která se řadí k největším na světě.

Dále je zde větrný mlýn ‚De Zwaluw‘ (Vlaštovka) na mletí mouky z roku 1748, unikátní nábreží ‚Stenendijk‘ a v Evropě poslední stará plně funkční vápenka.

Centrum města s kanály lemovanými lípami, oblast Van Stolkspark a dok plný ruchu a hemžení jsou ideální místa k procházkám.

Okolí Hasseltu stojí ovšem též za zmínku. V zimě hostí polder Mastenbroek velká stáda ovcí. V létě tvoří vesničky Genne, Streukel a Cellemuiden s velice vzácnou květinou řebčíkem kostkovaným

(lat. *Fritillaria meleagris*) výborné prostředí pro pěší i cyklistické výlety.

Hasselt je také velice důležité město vodních sportů. Jezera severozápadně od Overijssel, řeky IJssel, Overijsselse Vecht a Randmeren jsou snadno dosažitelné z přístavu jachet ‚De Molenwaard‘. Mořeplavbu, rybolov, plavání a jízdu na kanoi si v Hasseltu můžete vychutnat naplno.

A na závěr uvedeme jednu událost obzvláštního významu. Každoročně se na konci srpna v Hasseltu slaví dožínky – ‚Eui Festival‘.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Je-li to potřeba, pak přechod do nového sloupce může být vynucen pomocí `\column`. Sloupce nastavíme příkazem

```
\setupcolumns [...,*...]
```

```
* n          = NUMBER
  ntop       = NUMBER
  rule       = on off
  height     = DIMENSION
  tolerance  = verystrict strict tolerant verytolerant stretch
  distance   = DIMENSION
  balance    = yes no
  align      = inner outer left right middle normal no yes
  blank      = fixed halfline line flexible big medium small
  option     = background
  direction  = left right
  inherits from \setupframed
```

Ve většině případů dosáhneme při sazbě do sloupců lepších výsledků, zapneme-li sazbu do *řádkového rejstříku*. To se nastavuje parametrem `grid=yes` v příkazu `\setuplayout`.



Pokud chceme opatřit text poznámkou, můžeme použít `\footnote`. Příkaz vypadá takto:

- obsah**
- rejstřík**
- seznam příkazů**
- definice příkazů**
- tiráž**

- konec**
- vyhledej**
- +**
- krok zpět**
-
- +**

```
\footnote [1.] {2.}
                OPTIONAL
```

- 1 REFERENCE
- 2 TEXT

Hranaté závorky jsou nepovinné a obsahují referenční jméno poznámky pod čarou. Složené závorky obsahují text, který chcete zobrazit pod čarou.

Číslo poznámky pod čarou můžeme získat napsáním referenčního jména do

```
\note [1.]
```

- * REFERENCE

Pokud napíšeme text

Hanza byla pozdně středověká obchodní aliance měst v oblasti Severního a Baltského moře. Tento spolek byl sestaven na podporu a ochranu obchodování svých členů.\footnote[war]{Tento systém byl také zdrojem závisti a strachu mezi ostatními městy a zapříčinil mnohé válečné střety.}

V-období hanzovního systému zde panoval čilý obchodní ruch s veškerým zbožím jako dřevo, vlna, kovy, látky, sůl, víno a pivo.\note[war] Fungující obchod způsobil enormní růst prosperity a blahobytu v-hanzovních městech.\footnote{Jedním z těchto měst je Hasselt.}

pak bude vypadat takto:

Hanza byla pozdně středověká obchodní aliance měst v oblasti Severního a Baltského moře. Tento spolek byl sestaven na podporu a ochranu obchodování svých členů.⁸ V období hanzovního systému zde panoval čilý obchodní ruch s veškerým zbožím jako dřevo, vlna, kovy, látky, sůl, víno a pivo.⁸ Fungující obchod způsobil enormní růst prosperity a blahobytu v hanzovních městech.⁹

⁸ Tento systém byl také zdrojem závisti a strachu mezi ostatními městy a zapříčinil mnohé válečné střety.

⁹ Jedním z těchto měst je Hasselt.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Číslování poznámek se provádí automaticky. Příkaz `\setupfootnotes` nám umožní ovlivnit vzhled poznámky.

```
\setupfootnotes [...,*.,...]
```

```
* conversion = numbers characters Characters romannumerals Romannumerals
way          = bytext bycd:section
location     = page TEXT columns firstcolumn lastcolumn high none
rule        = on off
before      = COMMAND
after       = COMMAND
width       = DIMENSION
height      = DIMENSION
bodyfont    = 5pt ... 12pt small big
style       = normal bold slanted boldslanted type cap small...
            = COMMAND
distance    = DIMENSION
columndistance = DIMENSION
margindistance = DIMENSION
n           = NUMBER
numbercommand = \...#1
textcommand  = \...#1
split        = tolerant strict verystRICT NUMBER
textstyle    = normal bold slanted boldslanted type cap small...
            = COMMAND
textcolor    = IDENTIFIER
interaction  = yes no
factor       = NUMBER
inherits from \framed
```

Poznámky mohou být umístěny ve spodní části stránky, ale také jinde, např. na konci kapitoly. To provádíme nastavením `location=none` a uvedením `\placefootnotes` na konci kapitoly.

```
\placefootnotes [...,*.,...]
```

OPTIONAL

```
* inherits from \setupfootnotes
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Poznámka se také může přiřadit k tabulce. V takovém případě hovoříme o lokálních poznámkách. Pro tento účel existují následující příkazy:

```
\startlocalfootnotes ... \stoplocalfootnotes
```

```
\placelocalfootnotes [...,.*,...]
```

OPTIONAL

* *inherits from \setupfootnotes*

Ilustrující příklad použití lokálních poznámek:

```
\startlocalfootnotes[n=0]
\placetable
{Pokles produktivity Hasseltu.}
{\starttable[|l|c|c|c|c|]
\HL
\NC
\NC Kamnářství\footnote{Zdroj: Uit de geschiedenis van Hasselt.}
\NC Kovárny \NC Pivovary \NC Keramické dílny \NC\SR
\HL
\NC 1682 \NC 15 \NC 9 \NC 3 \NC 2 \NC\FR
\NC 1752 \NC ~6 \NC 4 \NC 0 \NC 0 \NC\LR
\HL
\stoptable
\placelocalfootnotes}
\stoplocalfootnotes
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

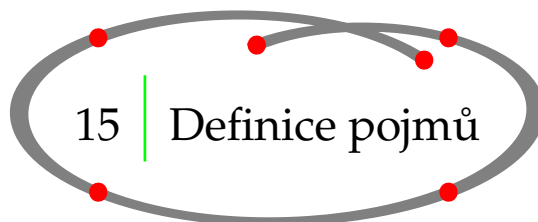
-

+

	Kamnářství ¹	Kovářny	Pivovary	Keramické dílny
1682	15	9	3	2
1752	6	4	0	0

¹ Zdroj: Uit de geschiedenis van Hasselt.

Tabulka 14.1 Pokles produktivity Hasseltu.



Pokud chceme poznámky, koncepty a myšlenky sázet jednotným způsobem, můžeme použít:

```
\definedescription [.1.] [...,2...]
                                OPTIONAL
1 IDENTIFIER
2 inherits from \setupdescriptions
```

Příklad:

```
\definedescription
[concept]
[location=serried,headstyle=bold,width=broad]
\concept{Hasseltské juffers:} Druh sušenky, vyrobený z~nadýchaného
těsta, sypaný cukrem. Je velmi sladký. \par
```

bude vypadat takto:

Hasseltské juffers: Druh sušenky, vyrobený z nadýchaného těsta, sypaný cukrem. Je velmi sladký.

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +

Můžeme také použít jiné formátování:

Hasseltská hořká:

Silný alkoholický nápoj (40 %) s přidanými bylinkami a velmi specifickou chutí. Prodává se v kameninových lahvičkách a podává se ijskoud (ledová).

Euifeest: Dožínky, oslava sklizně a konce období lopotné práce. Tyto slavnosti se konají poslední týden v srpnu.

Máme-li v takové definici více odstavců, použijeme dvojici `\start...-\stop...`:

```
\definedescription
  [concept]
  [location=right,
   headstyle=bold,
   width=broad]
```

```
\startconcept{Euifeest:} Dožínky, oslava sklizně a konce období lopotné
práce. Tato událost se koná na konci srpna a trvá přesně týden. Město
je celé osvětlené a ulice ozdobené. Tento sváteční týden je zakončen
{\em Braderií} (bazarem).
\stopconcept
```

Bude to vypadat následovně:

Dožínky, oslava sklizně a konce období lopotné práce. Tato událost se koná na konci srpna a trvá přesně týden. Město je celé osvětlené a ulice ozdobené. Tento sváteční týden je zakončen *Braderií* (bazarem).

Euifeest:

Formátování je nastaveno druhou hranatou závorkou v `\definedescription [] []`.

Po `\definedescription [] []` lze ještě změnit jeho formátování pomocí:

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +


```

\setupdescriptions [...1;...] [...2.,...]
                        OPTIONAL
1 IDENTIFIER
2 headstyle = normal bold slanted boldslanted type cap small... COMMAND
  style     = normal bold slanted boldslanted type cap small... COMMAND
  color     = IDENTIFIER
  width     = fit broad DIMENSION
  distance  = DIMENSION
  sample    = TEXT
  text      = TEXT
  align     = inner outer left right middle normal no yes
  margin    = standard yes no DIMENSION
  location  = left right top serried inmargin inleft inright hanging
  headcommand = COMMAND
  hang      = fit broad NUMBER
  before    = COMMAND
  inbetween = COMMAND
  after     = COMMAND
  indentnext = yes no
  indenting = never none not no yes always first next small medium big
            normal odd even DIMENSION
    
```

16 | Číslované poznámky

Užitím příkazu `\defineenumeration` můžeme očíslovat části textu, jako poznámky nebo otázky. Pokud chceme takovéto očíslované poznámky v dokumentu vytvořit, použijeme:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\defineenumeration [...1;...] [...2.] [...3.,...]
                        OPTIONAL      OPTIONAL
1 IDENTIFIER
2 IDENTIFIER
3 inherits from \setupenumerations
```

Například:

```
\defineenumeration
[remark]
[location=top,
text=Poznámka,
inbetween=\blank,
after=\blank]
```

Nyní jsou k dispozici příkazy `\remark`, `\subremark`, `\resetremark` a `\nextremark`, takže můžeme psát poznámky typu:

```
\remark Na začátku středověku byl Hasselt poutním místem.
Toto {\em Heilige Stede} (Posvátné místo) bylo strženo a zničeno
během období reformace. V roce 1930 po 300 letech
bylo {\em Heilige Stede} znovu vybudováno a otevřeno.
```

```
\subremark Nyní je {\em Heilige Stede} opět uzavřeno,
ale jednou ročně se na tomtéž místě koná bohoslužba
pod širým nebem. \par
```

Poznámka 1

Na začátku středověku byl Hasselt poutním místem. Toto *Heilige Stede* (Posvátné místo) bylo strženo a zničeno během období reformace. V roce 1930 po 300 letech bylo *Heilige Stede* znovu vybudováno a otevřeno.

Poznámka 1.1

Nyní je *Heilige Stede* opět uzavřeno, ale jednou ročně se na tomtéž místě koná bohoslužba pod širým nebem.

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej
- +
- krok zpět
-
- +

Pomocí `\resetremark` nebo `\resetsubremark` můžeme opět číslovat od začátku. Číslo zvýšíme pomocí `\nextremark` nebo `\nextsubremark`. Normálně se poznámky číslují automaticky po kapitolách nebo částech knihy.

Úpravu `\defineenumeration` si můžeme nastavit takto:

```
\setupenumerations [...1...] [...2...]
                        OPTIONAL
1 IDENTIFIER
2 inherits from \definedescription
```

Úpravu **Poznámky** a **Podpoznámky** ve výše uvedeném příkladu nastavíme pomocí:

```
\setupenumeration[remark] [headstyle=bold]
\setupenumeration[subremark] [headstyle=slanted,text=Podpoznámka]
```

Obsahuje-li poznámka více než jeden odstavec, musíme použít párový příkaz `\startremark` ... `\stopremark`, se kterým lze pracovat po předchozím nadefinování **Poznámky** užitím `\defineenumeration[remark]`.

Předchozí příklad bude vypadat takto:

```
\startremark
Na začátku středověku byl Hasselt poutním místem.
Toto {\em Heilige Stede} (Posvátné místo) bylo strženo
a zničeno během období reformace.

V roce 1930 po 300 letech bylo {\em Heilige Stede}
znovu vybudováno a otevřeno. Nyní je {\em Heilige Stede}
opět uzavřeno, ale jednou ročně se na tomtéž
místě koná bohoslužba pod širým nebem.
\stopremark
```

Poznámka 1

Na začátku středověku byl Hasselt poutním místem. Toto *Heilige Stede* (Posvátné místo) bylo strženo a zničeno během období reformace.

V roce 1930 po 300 letech bylo *Heilige Stede* znovu vybudováno a otevřeno. Nyní je *Heilige Stede* opět uzavřeno, ale jednou ročně se na tomtéž místě koná bohoslužba pod širým nebem.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Pokud nechceme danou poznámku číslovat, použijeme:

```
\remark[-] nebo \startremark[-]
```

Například:

```
\remark[-]
Na začátku středověku byl Hasselt poutním místem.
Toto {\em Heilige Stede} (Posvátné místo) bylo strženo
a zničeno během období reformace.\par
```

Poznámka

Na začátku středověku byl Hasselt poutním místem. Toto *Heilige Stede* (Posvátné místo) bylo strženo a zničeno během období reformace.



Text můžeme **orámovat** užitím `\framed`. Příkaz vypadá takto:

```
\framed [.,.1.,.] {.2.}
          OPTIONAL
1  inherits from \setupframed
2  TEXT
```

Hranaté závorky jsou nepovinné a obsahují nastavení parametrů orámování. Složené závorky obsahují text. Po pravdě řečeno, orámovaný text v posledním odstavci byl vytvořen příkazem `\inframed`, který se postará i o správné mezery mezi řádky.

```
\framed[height=3em,width=fit]{Hasselt potřebuje více prostoru}
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Tento zápis se zobrazí následujícím způsobem:

Hasselt potřebuje více prostoru

Další příklady `\framed` a nastavení jeho parametrů jsou ukázány dále. Tentokrát používáme alternativu `\inframed`.

```
\leftaligned
  {\inframed[width=fit]{Historie Hasseltu}}
\midaligned
  {\inframed[height=1.5cm,frame=off]{je}}
\rightaligned
  {\inframed[background=screen]{barvitá.}}
```

Takový zápis vytvoří:

Historie Hasseltu

je

barvitá.

Příkaz `\framed` je velmi důmyslný a používá se uvnitř mnoha maker. Syntaxe parametrů orámování je následující:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupframed [..,.*,..]
```

```
* height          = fit broad DIMENSION
width            = fit broad fixed local DIMENSION
autowidth       = yes no force
offset          = none overlay default DIMENSION
location        = depth hanging high lohi low top middle bottom keep
option          = none empty
strut           = yes no global local
align           = inner outer left right middle normal no yes
bottom          = COMMAND
top             = COMMAND
frame           = on off none overlay
topframe       = on off
bottomframe    = on off
leftframe      = on off
rightframe     = on off
frameoffset    = DIMENSION
framedepth     = DIMENSION
framecorner    = round rectangular
frameradius    = DIMENSION
framecolor     = IDENTIFIER
rulethickness  = DIMENSION
background     = screen color none foreground IDENTIFIER
backgroundscreen = NUMBER
backgroundcolor = IDENTIFIER
backgroundoffset = frame DIMENSION
backgrounddepth = DIMENSION
backgroundcorner = round rectangular
backgroundradius = DIMENSION
depth          = DIMENSION
corner         = round rectangular
radius        = DIMENSION
empty         = yes no
foregroundcolor = IDENTIFIER
foregroundstyle = normal bold slanted boldslanted type cap small...
               COMMAND
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+



Celý odstavec můžeme orámovat pomocí:

```
\startframedtext [1.] [OPTIONAL..OPTIONAL.2..] {3..4..} ... \stopframedtext
1 left right middle none
2 inherits from \setupframedtexts
3 TEXT
4 TEXT
```

```
\definefloat [vsuvka]
```

```
\placevsuvka [here] [block:bridge]{Příklad vsuvky.}
```

```
\startframedtext [width=.8\makeupwidth]
```

Pro město Hasselt bylo velmi důležité mít most přes řeku Zwarte. Biskup utrechtský dal Hasseltu svolení k jeho stavbě v roce 1486.

```
\blank
```

Ostatní města v okolí se však obávala poplatků, které by bylo nutné platit při přechodu přes tento most, a~tak mnoho let bránili jeho stavbě.

```
\stopframedtext
```

Tento příklad také ukazuje použití příkazu `\definefloat`. Více informací k tomuto příkazu najdete v **sekcí 37.2**. Příkaz `\blank` vytvoří prázdný řádek.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Pro město Hasselt bylo velmi důležité mít most přes řeku Zwarte. Biskup utrechtský dal Hasseltu svolení k jeho stavbě v roce 1486.

Ostatní města v okolí se však obávala poplatků, které by bylo nutné platit při přechodu přes tento most, a tak mnoho let bránili jeho stavbě.

Vsuvka 18.1 Příklad vsuvky.

Orámování můžete nastavit pomocí

```
\setupframedtexts [.,.,*.,.]
```

```
* bodyfont      = 5pt ... 12pt small big
  style         = normal bold slanted boldslanted type cap small...
                COMMAND
  left          = COMMAND
  right         = COMMAND
  before        = COMMAND
  after         = COMMAND
  inner         = COMMAND
  linecorrection = on off
  depthcorrection = on off
  margin        = standard yes no
  location       = left right middle none
  indenting     = never none not no yes always first next small medium
                big normal odd even DIMENSION
  inherits from \setupframed
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

19 | Marginálie

Vložit text do *marginálie* (okraje stránky) je velmi snadné. Stačí použít `\inmargin`.

```
\inmargin [.1.] [.2.] {.3.}
          OPTIONAL OPTIONAL
1 + - low
2 REFERENCE
3 TEXT
```

Připomeňme si jeden z předcházejících příkladů:

```
\inmargin
  {\externalfigure
   [ma-cb-23]
   [width=\marginwidth]}
```

Tato ukázka vysází obrázky do okraje. Jistě si umíme představit, že v některých dokumentech může tento efekt působit velice dobře, ale buďme opatrní. Okraje jsou většinou velmi úzké a obrázky tak budou vsutku okrajové!

Několik dalších příkladů je ukázáno v následujícím textu.

"Ridderstraat" (Rytířská ulice) je název velmi výmluvný. Ve 14. a 15. století zde bydleli šlechtici a prominentní obyvatelé. Z některých jejich velkých domů se později `\inright{chudobinec}` staly chudobince a starobince.

Až do roku `\inleft[low]{\tfc 1940}` 1940 se zde nacházela synagoga. Asi 40 židů se zde scházelo ke slavení svého šabat. Během války byli všichni židé deportováni do Westborku a pak do koncentračních



- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +

táborů v~Německu a~Polsku. Žádná z~židovských rodin se již nevrátila a~synagoga byla zbořena v~roce 1958.

Příkazy `\inmargin`, `\inleft` a `\inright` plní stejnou funkci. V oboustranně tištěném dokumentu bude příkaz `\inmargin` sázet vždy do správného okraje. Dvojitě lomítko `\\` způsobí zalomení řádky. Předchozí příklad by tedy vypadal takto:

"Ridderstraat" (Rytířská ulice) je název velmi výmluvný. Ve 14. a 15.století zde bydleli šlechtici a prominentní obyvatelé. Z některých jejich velkých domů se pozdějistaly chudobince a starobince.

1940

Až do roku 1940 se zde nacházela synagoga. Asi 40 židů se zde scházelo ke slavení svého šabatů. Během války byli všichni židé deportováni do Westborku a pak do koncentračních táborů v Německu a Polsku. Žádná z židovských rodin se již nevrátila a synagoga byla zbořena v roce 1958.

Nastavit text v okraji lze pomocí `\setupinmargin`.

```
\setupinmargin [..1.] [..2.,..]
                OPTIONAL
1 left right NUMBER
2 location = left right both
  style = normal bold slanted boldslanted type cap small... COMMAND
  before = COMMAND
  after = COMMAND
  align = inner outer left right middle normal no yes
  line = NUMBER
  distance = DIMENSION
  separator = TEXT
  width = DIMENSION
  distance = DIMENSION
  stack = yes no
  inherits from \setupframed
```

chudobinec

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

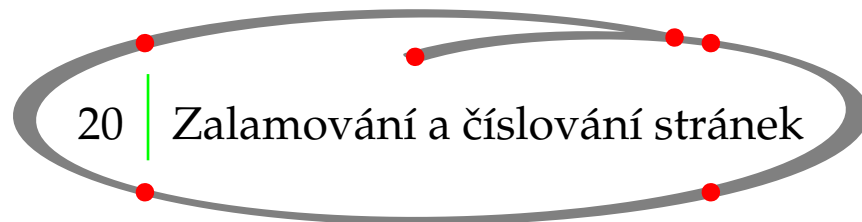
vyhledej

+

krok zpět

-

+



Zalomení stránky lze vynutit nebo potlačit příkazem:

```
\page [...;*...]
```

```
* yes  makeup  no  preference  bigpreference  left  right  disable  last  quadruple
   even  odd  blank  empty  reset  start  stop
```

Parametry se zadávají uvnitř hranatých závorek. Parametry a jejich smysl jsou uvedeny v **tabulce 20.1**.

Číslování stránek je automatické, ale je možné ručně nastavit číslo stránky příkazem:

```
\page [25]
```

Někdy je lepší zadat číslo strany relativně, například [+2] nebo [-2].

Umístění čísla stránek na straně závisí na našich preferencích a na typu dokumentu (jednostranný nebo dvojstranný). Číslování stránek se nastavuje příkazem:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Volba	Význam
yes	začátek nové stránky
makeup	začátek nové stránky bez plovoucích objektů
no	potlačení stránkového zlomu
preference	na tomto místě upřednostňujeme novou stránku
bigpreference	nová stránka má na tomto místě vysokou prioritu
left	následující stránka je levá (sudá)
right	následující stránka je pravá (lichá)
disable	následující příkazy nemají účinek
reset	následující příkazy se provedou
empty	vložení prázdné stránky
last	vkládání stránek až do dosažení sudého čísla strany
quadruple	vkládání stránek až do dosažení násobku čtyř

Tabulka 20.1 Volby příkazu \page.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setuppagenumbering [...,*.,...]
```

```
* alternative      = singlesided doublesided
  location         = header footer left right middle margin marginedge
                  inleft inright
  conversion       = numbers characters Characters romannumerals Romannumerals
  style            = normal bold slanted boldslanted type cap small...
                  COMMAND
  left             = TEXT
  right            = TEXT
  way              = bytext bycd:section
  text             = TEXT
  numberseparator = TEXT
  textseparator   = TEXT
  cd:sectionnumber = yes no
  separator        = TEXT
  strut           = yes no
  state           = start stop
  width           = DIMENSION
  command         = \...#1
```

Parametry se umísť ují do hranatých závorek.

Někdy může být výhodné odložit nějaký materiál až na začátek další strany, tedy po provedení nejbližšího stránkového zlomu. To se provede příkazem:

```
\startpostponing ... \stoppostponing
```

```
\startpostponing
\placefigure
  {Odložený obrázek.}
  {\externalfigure[ma-cb-16] [width=\textwidth]}
\stoppostponing
```

Obrázek bude umístěn ve vrchní části následující strany, aby co nejméně narušoval průběžný text.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+



Obrázek 21.1 Odložený obrázek.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+



Někdy chceme dokument opatřit záhlavím a úpatím. K tomu se používají následující příkazy:

```
\setupfootertexts [.1.] [.2.] [.3.]
                   OPTIONAL OPTIONAL OPTIONAL
1   text margin edge
2   TEXT SECTION date MARK pagenumber
3   TEXT SECTION date MARK pagenumber
```

```
\setupheadertexts [.1.] [.2.] [.3.]
                  OPTIONAL OPTIONAL OPTIONAL
1   text margin edge
2   TEXT SECTION date MARK pagenumber
3   TEXT SECTION date MARK pagenumber
```

První dvojice závorek udává, kam se má záhlaví nebo úpatí umístit (text, margin=marginálie, edge=vnější okraj, viz **obrázek 123**). Úpatí a záhlaví jsou určeny druhou a třetí závorkou. V oboustranných dokumentech dodáme čtvrtou a pátou závorku pro nastavení záhlaví/úpatí levých a pravých stránek odděleně. Ve většině případů však můžeme tyto poslední dvě závorky vynechat.

```
\setupfootertexts [Manuál] [section]
```

V tomto případě se text *Manuál* objeví v levém dolním rohu stránky a jméno aktuální kapitoly bude vpravo.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Čáru oddělovací záhlaví a úpatí můžeme nastavit pomocí:

```
\setupheader [..] [.,.,2.,..]
              OPTIONAL
```

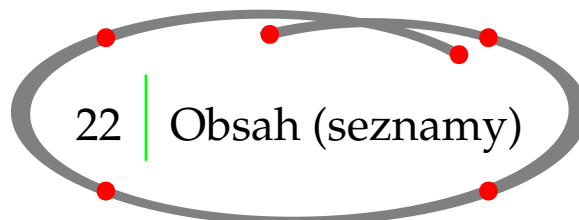
```
1 TEXT margin edge
2 state      = normal stop start empty high none nomarking IDENTIFIER
  strut      = yes no
  style      = normal bold slanted boldslanted type cap small... COMMAND
  leftstyle  = normal bold slanted boldslanted type cap small... COMMAND
  rightstyle = normal bold slanted boldslanted type cap small... COMMAND
  leftwidth  = DIMENSION
  rightwidth = DIMENSION
  before     = COMMAND
  after      = COMMAND
```

```
\setupfooter [..] [.,.,2.,..]
              OPTIONAL
```

```
1 inherits from \setupheader
2 inherits from \setupheader
```

Pokud chceme záhlaví a úpatí zcela vypustit, napíšeme:

```
\noheaderandfooterlines
```



obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Obsah zahrnuje čísla kapitol, názvy kapitol a čísla stránek; může být rozšířen o sekce, podsekce atd. Obsah se generuje automaticky napsáním:

```
\placecontent
```

Jaký obsah se vytvoří, záleží na umístění tohoto příkazu v dokumentu. Na začátku dokumentu se vygeneruje seznam všech kapitol, sekcí atd. Ale umístěním na začátku jednotlivých kapitol, tj.:

```
\chapter{Hasselt v létě}
\placecontent
\section{Hasselt v červenci}
\section{Hasselt v srpnu}
```

se pouze vytvoří seznam názvů sekcí s odpovídajícím očíslováním a s čísly stránek.

Příkaz `\placecontent` můžeme použít poté, co nadefinujeme nový seznam pomocí:

```
\definecombinedlist [.1.] [...2;...] [...,3.,...]
                                OPTIONAL
1 IDENTIFIER
2 LIST
3 inherits from \setupcombinedlist
```

Tento příkaz spolu s `\definelist` nám dovolí nadefinovat vlastní seznam, potřebný ke strukturování celého dokumentu.

Použití tohoto a dalších příbuzných příkazů je ukázáno na následujícím příkladu:

```
\definelist[kapitola]
\setuplist
[kapitola]
[before=\blank,
after=\blank,
style=bold]
\definelist[podkapitola]
\setuplist
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
[podkapitola]
[alternative=d]
```

Nyní máme dva seznamy: seznam kapitol a seznam podkapitol. Oba budou sloučeny do jediného obsahu příkazem `\definecombinedlist`.

```
\definecombinedlist
[obsah]
[kapitola,podkapitola]
[level=subsection]
```

Po tomto úkonu máme k dispozici dva další příkazy: `\placeobsah` a `\completeobsah`. Oba způsobí vysázení obsahu, ten druhý ještě přidá nadpis ‚Obsah‘.

Vzhled seznamu lze nastavit parametrem `alternative`.

Nastavení	Formátování
a	číslo – název – číslo stránky
b	číslo – název – mezery – číslo stránky
c	číslo – název – tečky – číslo stránky
d	číslo – název – číslo stránky (v odstavci)
e	rezervováno pro interaktivní účely
f	rezervováno pro interaktivní účely

Tabulka 22.1 Nastavení formátování seznamů.

Seznamy se nastavují pomocí:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setuplist [...1,...] [...2,...]
```

```
1 IDENTIFIER
2 state = start stop
  alternative = a b c ... none command
  coupling = on off
  criterium = SECTION local previous current all
  pageboundaries = LIST
  style = normal bold slanted boldslanted type cap small...
  COMMAND
  numberstyle = normal bold slanted boldslanted type cap small...
  COMMAND
  textstyle = normal bold slanted boldslanted type cap small...
  COMMAND
  pagestyle = normal bold slanted boldslanted type cap small...
  COMMAND
  color = IDENTIFIER
  command = \...#1#2#3
  numbercommand = \...#1
  textcommand = \...#1
  pagecommand = \...#1
  interaction = cd:sectionnumber TEXT pagenumber all
  before = COMMAND
  after = COMMAND
  inbetween = COMMAND
  left = TEXT
  right = TEXT
  label = yes no
  prefix = yes no none
  pagenumber = yes no
  headnumber = yes no
  cd:sectionnumber = yes no
  aligntitle = yes no
  margin = DIMENSION
  width = DIMENSION fit
  height = DIMENSION fit broad
  depth = DIMENSION fit broad
  distance = DIMENSION
  separator = TEXT
  stopper = TEXT
  symbol = none 1 2 3 ...
  expansion = yes no command
  maxwidth = DIMENSION
  inherits from \framed
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupcombinedlist [.1.] [...,.2.,...]
```

```
1 IDENTIFIER
2 level = 1 2 3 4 SECTION current
   inherits from \setuplist
```

Pokud chceme změnit vzhled generovaného obsahu, musíme pamatovat na to, že je to seznam.

```
\setupcombinedlist
[obsah]
[alternative=c,
aligntitle=no,
width=2.5cm]
```

Od této chvíle se bude formátování obsahu lišit od jeho standardního nastavení.

Seznamy se vyvolávají a umísťují pomocí:

```
\placelist [...,1,...] [...,.2.,...]
                                     OPTIONAL
```

```
1 IDENTIFIER
2 inherits from \setuplist
```

Chceme-li tedy pouze část obsahu, napíšeme:

```
\placecontent[level=section]
```

nebo

```
\completecontent[level=section]
```

a zobrazí se pouze kapitoly a sekce. To se může hodit, pokud máme složitý a strukturovaný obsah, který má spoustu pod pod a pod odstavců, které v celkovém obsahu nechceme vypisovat.

Dlouhý seznam nebo obsah se nevejde na jednu stránku. Potřebujeme-li ve vhodném místě vynutit přechod na novou stránku, napíšeme:

```
\completecontent[2.2,8.5,12.3.3]
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Zalomení stránky se objeví po sekci 2.2 a 8.5 a podsekcí 12.3.3.

V některých případech chceme v automaticky generovaném seznamu napsat vlastní text. Toho dosáhneme pomocí:

```
\writetolist [1..] {2..} {3..}
```

```
1 SECTION IDENTIFIER
2 TEXT
3 TEXT
```

```
\writebetweenlist [1..] {2..}
```

```
1 SECTION IDENTIFIER
2 TEXT
```

Například, chceme-li udělat poznámku v obsahu po sekci nazvané *Hotely v Hasseltu*, napíšeme:

```
\section{Hotely v Hasseltu}
\writebetweenlist[section]{\blank}
\writetolist[section]{}{---ve výstavbě--}
\writebetweenlist[section]{\blank}
```



obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

V `CONTEXTU` je možné vytvořit jeden nebo více rejstříků. Implicitně je k dispozici příkaz `\index`. Jestliže chceme do rejstříku přidat nějaké slovo, napíšeme například:

```
\index{radnice}
```

Výraz *radnice* se objeví jako heslo rejstříku. Rejstřík je pomocným programem seřazen podle abecedy. Pokud je některé heslo třeba zatřídit mimo běžné abecední pořadí, například v případě symbolu, musí být doplněno o další třídící informaci:

```
\index[mínus]{$-}$}
```

Někdy máme pod- nebo podpodhesla. Ty se zadávají následujícím způsobem:

```
\index{radnice+poloha}
\index{radnice+architektura}
```

Rejstřík vygenerujeme příkazem

```
\placeindex
```

nebo

```
\completeindex
```

Příkaz `\index` je v `CONTEXTU` předdefinovaný, ale samozřejmě můžeme také definovat své vlastní rejstříky:

```
\defineregister [..1.] [..2.]
```

1 SINGULAR NAME

2 PLURAL NAME

Chceme-li například vytvořit rejstřík ulic v Hasseltu, můžeme napsat:

```
\defineregister [ulice] [ulice]
```

Nyní je k dispozici nový příkaz pro tvorbu rejstříku `\ulice`. Heslem tohoto nového rejstříku může být třeba `\ulice{Ridderstraat}`. Samotný *seznam ulic* nyní vytvoříme použitím příkazů:

```
\placeulice
\completeulice
```

Vzhled rejstříků můžeme měnit příkazem:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupregister [...1] [...2] [...,3.,...]
                        OPTIONAL
```

1 SINGULAR NAME

2 IDENTIFIER

3 n = NUMBER

balance = yes no

align = inner outer left right middle normal no yes

style = normal bold slanted boldslanted type cap small...
COMMAND

pagestyle = normal bold slanted boldslanted type cap small...
COMMAND

textstyle = normal bold slanted boldslanted type cap small...
COMMAND

indicator = yes no

coupling = yes no

cd:sectionnumber = yes no

criterium = SECTION local all

distance = DIMENSION

symbol = 1 2 ... n a ... none

interaction = pagenumber TEXT

expansion = yes no command

referencing = on off

command = \...#1

location = left middle right

maxwidth = DIMENSION

unknownreference = empty none

alternative = a b A B

prefix = both first none

compress = no yes

deeptextcommand = \...#1

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

24 | Synonyma

V mnoha případech chtějí autoři používat určitá slova shodně v celém dokumentu. K dosažení jednotnosti je k dispozici následující příkaz:

```
\definesynonyms [.1.] [.2.] [.3.] [.4.]
                                OPTIONAL
1  SINGULAR NAME
2  PLURAL NAME
3  COMMAND
4  COMMAND
```

První dvojice hranatých závorek obsahuje tvar synonyma v jednotném čísle a druhá dvojice obsahuje tvar množného čísla. Třetí dvojice hranatých závorek obsahuje povel.

Například povel `\abbreviation` (zkratka) je definován následovně:

```
\definesynonyms [abbreviation] [abbreviations] [\infull]
\setupsynonyms [style=cap]
```

Nyní je k dispozici příkaz `\abbreviation` a můžeme jím definovat své vlastní zkratky:

```
\abbreviation{ANWB}{Holandská automobilová asociace}
\abbreviation{VVV}{Úřad turistických informací}
\abbreviation{NS}{Holandské dráhy}
```

Jestliže napíšeme:

```
Holandský \VVV\ (\infull{VVV}) nám může poskytnout turistické
informace o~Hasseltu.
```

Dostaneme něco jako:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Holandský vvv (Úřad turistických informací) nám může poskytnout turistické informace o Has-seltu.

Seznam synonym a zkratk je nejlepší definovat v deklarační oblasti našeho vstupního souboru z důvodu jeho jednoduché údržby. Tento typ informace můžeme také uložit do externího souboru (např. abbrev.tex) a tento načíst příkazem:

```
\input abbrev.tex
```

Když chcete do vašeho dokumentu umístit seznam zkratk, můžete napsat:

```
\placelistofabbreviations
```

nebo

```
\completelistofabbreviations
```

Vytvoří se tak úplný abecedně uspořádaný seznam zkratk a jejich významů.

Sazba synonym se dá ovlivnit příkazem \setupsynonyms.

```
\setupsynonyms [..1..] [..,2..,..]
```

```
1 IDENTIFIER
2 textstyle = normal bold slanted boldslanted type cap small... COMMAND
  synonymstyle = normal bold slanted boldslanted type cap small... COMMAND
  location = left right top serried inmargin inleft inright
  width = DIMENSION
  state = start stop
  criterium = all used
  conversion = yes no
  expansion = yes no command
  command = \...#1#2#3
```

Další aplikace synonym je popsána v kapitole 9.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

25 | Křížové odkazy

Na vytvoření křížového odkazu z jednoho místa dokumentu na druhé použijeme příkaz:

```
\in {.1.} {.2.} [.3.]
```

- 1 TEXT
- 2 TEXT
- 3 REFERENCE

Složené závorky mohou obsahovat text, který má být vysázen před nebo za odkazem. To je výhodné pro zvětšení klikatelné oblasti v interaktivních dokumentech. Hranaté závorky pak nesou referenční jméno. Když napíšeme název kapitoly následujícím způsobem

```
\chapter [hotel]{Hotely v~Hasseltu}
```

tak se na tuto kapitolu můžeme odkázat povelom

```
v~\in{kapitole}[hotel]
```

Po zpracování dostaneme číslo kapitoly a odkaz vypadá například takto: v kapitole 23. Příkaz `\in` můžeme použít s referenčním jménem libovolného textového prvku, například kapitoly, sekce, obrázku, tabulky, vzorce, apod.

Další příklad:

V~Hasseltu můžeme dělat spoustu věcí:

```
\startitemize[n,packed]
\item plavat,
\item plachtit,
\item [tramp] trampovat,
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\item jezdit na kole.
```

```
\stopitemize
```

```
\in{Činnost}[tramp] popsaná na \at{straně}[tramp] je velmi namáhavá.
```

Bude to vypadat takto:

V Hasseltu můžeme dělat spoustu věcí:

1. plavat,
2. plachtit,
3. trampovat,
4. jezdit na kole.

Činnost 3 popsaná na **straně 96** je velmi namáhavá.

Jak můžeme vidět, je také možné vytvořit odkaz na stránku. To se dělá příkazem:

```
\at {1.} {2.} [3.]
```

```
1 TEXT
```

```
2 TEXT
```

```
3 REFERENCE
```

Tento příkaz lze kombinovat s povely:

```
\pagereference [...]
```

```
* REFERENCE
```

a

```
\textreference [1.] {2.}
```

```
1 REFERENCE
```

```
2 TEXT
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Když se chceme odkázat na kapitolu *Hotely v Hasseltu*, můžeme napsat:

V~\in{kapitole}[hotel] na \at{straně}[hotel] nalezneme úplný přehled
možností ubytování v~\pagereference[bydleni]Hasseltu.

Při zpracování zdrojového souboru bude vygenerováno číslo kapitoly a číslo strany. Na jiném
místě dokumentu se můžeme odkázat na bydleni takto: na \at{straně}[bydleni].

Můžeme rovněž definovat několik návěští oddělených čárkami.

```
\placefigure
[here]
[obr:kanaly,obr:cluny]
{Typický obrázek Hasseltu.}
{\externalfigure[ma-cb-08][width=10cm]}
```

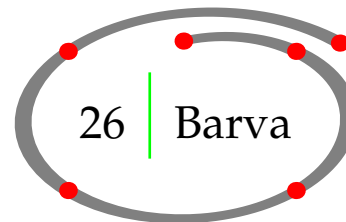
V~Hasseltu je mnoho kanálů (viz \in{obrázek}[obr:kanaly]).

.

V~hasseltských kanálech mohou kotvit čluny
(viz \in{obrázek}[obr:cluny]).

Výsledek může vypadat následovně:

V Hasseltu je mnoho kanálů (viz **obrázek 25.1**). . . . V hasseltských kanálech mohou kotvit čluny
(viz **obrázek 25.1**).



Text může být sázen barevně.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+



Obrázek 25.1 Typický obrázek Hasseltu.

```
\setupcolors [...,.*,...]
```

```
* state      = start  stop  global  local
  conversion = yes   no   always
  reduction  = yes   no
  rgb        = yes   no
  cmyk       = yes   no
  mpcmyk     = yes   no
  mpspot     = yes   no
  textcolor  = IDENTIFIER
  split      = c   m   y   k   p   s   no  IDENTIFIER
  criterium  = all  none
```

Používání barev se musí zapnout povelom:

```
\setupcolors[state=start]
```

Nyní máme k dispozici základní barvy (červenou – red, zelenou – green a modrou – blue).

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\startcolor[red]
Hasselt je velice \color[green]{barevné} město.
\stopcolor
```

Hasselt je velice **barevné** město.

Při použití černobílého tisku uvidíme jenom šedé odstíny. V elektronickém dokumentu barvy dopadnou podle očekávání.

Můžete také definovat své vlastní barvy příkazem:

```
\definecolor [.1.] [.,.2.,...]

1 IDENTIFIER
2 r = TEXT
  g = TEXT
  b = TEXT
  c = TEXT
  m = TEXT
  y = TEXT
  k = TEXT
  s = TEXT
  h = TEXT
  t = TEXT
  a = TEXT
  p = TEXT
  e = TEXT
```

Například:

```
\definecolor [temneruda] [r=.5,g=.0,b=.0]
\definecolor [tmavezelena] [r=.0,g=.5,b=.0]
```

Nyní jsou k dispozici barvy **temneruda** a **tmavezelena**.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

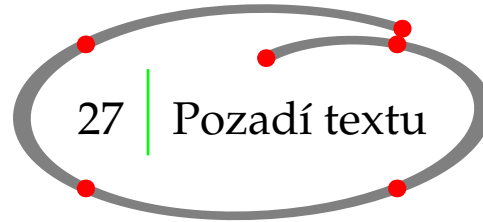
vyhledej

+

krok zpět

-

+



Ke zvýraznění kapitoly nebo několika odstavců můžeme použít *pozadí*. Takový text se vloží do prostředí

```
\starttextbackground ... \stoptextbackground
```

Použití ilustruje tento příklad:

```
\setuptextbackground
  [frame=off,location=paragraph,
   leftoffset=.5\bodyfontsize,rightoffset=.5\bodyfontsize]
\starttextbackground
V-Hasseltu vyrůstalo několik významných lidí. Teprve nedávno se
ukázalo, že Kilian van Rensselaer hrál významnou roli při založení státu
New York.
\stoptextbackground
```

Výsledkem je:

V Hasseltu vyrůstalo několik významných lidí. Teprve nedávno se ukázalo, že Kilian van Rensselaer hrál významnou roli při založení státu New York.

Pozadí může zabírat několik stránek. Pomocí

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

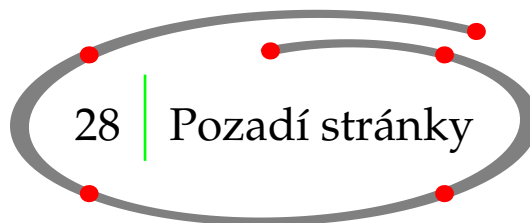
-

+

```
\setuptextbackground [...,*.,...]
```

```
* leftoffset = DIMENSION
  rightoffset = DIMENSION
  topoffset = DIMENSION
  bottomoffset = DIMENSION
  before = COMMAND
  after = COMMAND
  state = start stop
  location = text paragraph none
  alternative = 0 1 2
  inherits from \setupframed
```

Lze ovlivnit jeho formátování.



Pozadí stránky může být nastaveno pomocí:

```
\setupbackgrounds [.1.] [...2;...] [...,3.,...]
                   OPTIONAL  OPTIONAL
```

```
1 top header TEXT footer bottom page paper leftpage rightpage
2 leftedge leftmargin TEXT rightmargin rightedge
3 state = start stop cd:repeat
  inherits from \setupframed
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

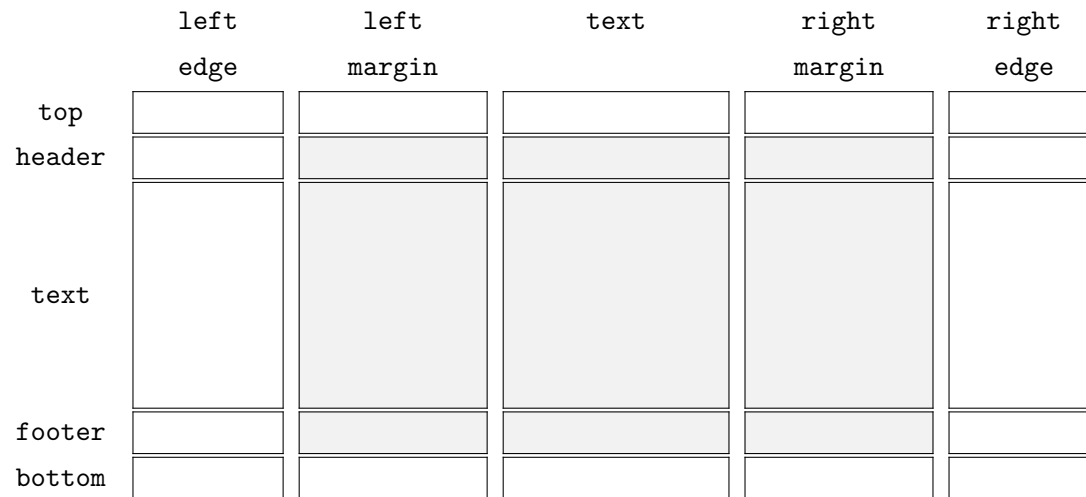
+

krok zpět

-

+

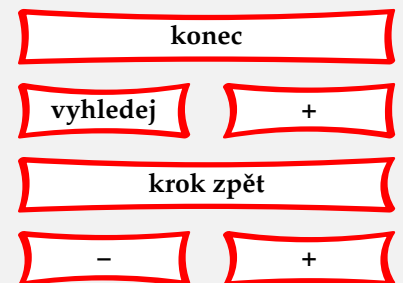
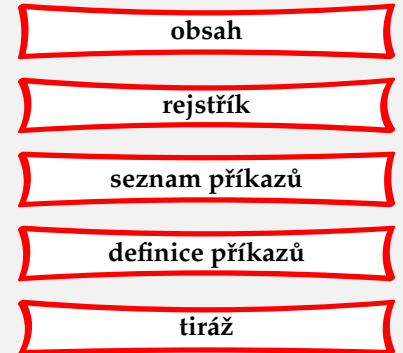
První dvě hranaté závorky definují část plochy stránky, které chceme nastavit pozadí. Poslední závorka obsahuje jeho parametry.



Obrázek 28.1 Plochy stránky použitelné v `\setupbackgrounds`.

Pokud např. chcete mít šedé pozadí jako na **obrázku 28.1**, napíšeme:

```
\setupbackgrounds
 [header,text,footer]
 [leftmargin,text,rightmargin]
 [background=screen]
```



Horizontální a vertikální zarovnání textu se nastavuje pomocí

```
\setupalign [...]
```

```
* width left right middle inner outer wide broad height bottom line reset
   hanging nothanging hyphenated nothyphenated
```

Pro zarovnání jedné řádky stačí uvést:

```
\rightaligned{}
\leftaligned{}
\midaligned{}

```

Např.

```
\leftaligned {Hasselt byl vybudován na písččné hoře.}
\midaligned  {Hasselt byl vybudován na soutoku dvou řek.}
\rightaligned {Hasselt byl pojmenován podle dřeva
               stromu ambroně (hazelwood).}
```

Po zpracování bude tento příklad vypadat takto:

Hasselt byl vybudován na písččné hoře.

Hasselt byl vybudován na soutoku dvou řek.

Hasselt byl pojmenován podle dřeva stromu ambroně (hazelwood).

K zarovnání odstavce se použije:

```
\startalignment [...] ... \stopalignment
                   OPTIONAL
```

```
* inherits from \setupalign
```

Zarovnání můžeme upřesnit zadáním směru (direction) a tolerance pružných výplňků (tolerance). Směr může být vertikální (vertical) a horizontální (horizontal). Tolerance je standardně velmi přísná (verystriict). Při sazbě do užších sloupců je užitečné povolit schovíva-
větší přístup k zalamování odstavce (tolerant nebo verytolerant). Pro sazbu tohoto manuálu je tolerance nastavena takto:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

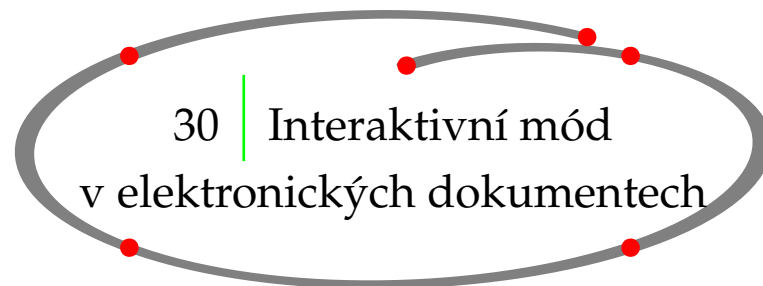
+

krok zpět

-

+

`\setuptolerance[horizontal,verystrict]`



30.1 Úvod

Dokumenty dnes mohou být zpřístupňovány elektronicky prostřednictvím obrazovky počítače.

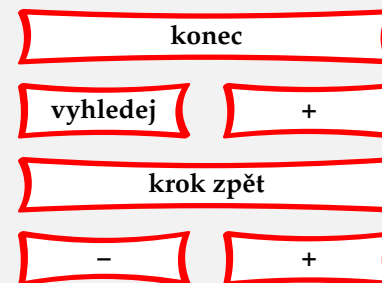
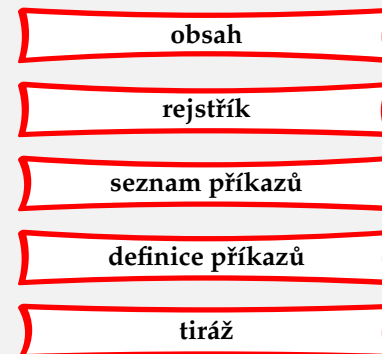
Po kliknutí myší na určitou plochu interaktivního dokumentu, se nám zobrazí jeho jiná část – ta, která byla odkazována. Příkladem může být kliknutí na číslo strany v rejstříku a následné zobrazení odpovídající stránky dokumentu.

Interakce se týkají:

- aktivních čísel kapitol v obsahu;
- aktivních čísel stran v rejstříku;
- aktivních čísel stran, čísel kapitol a čísel obrázků ve vnitřních odkazech na strany, kapitoly, obrázky ap.;
- aktivních názvů, čísel stran a čísel kapitol v externích referencích na jiné interaktivní dokumenty;
- aktivní menu v navigačních panelech.

Interaktivní chování do jisté míry závisí na programu, kterým dokument prohlížíme. Pokud předpokládáme, že použijeme PDF_TE_X pro přímé generování PDF dokumentu nebo použijeme Acrobat Distiller pro převedení POSTSCRIPTového souboru na soubor PDF, je pak zřejmé, že pro zobrazení použijeme PDF prohlížeč, například Acrobat Reader, Acrobat Exchange nebo GHOSTVIEW.

CONTEXT je velmi mocný nástroj pro generování elektronických nebo interaktivních dokumentů PDF. V této kapitole je nicméně popsáno jenom pár standardních vlastností. Protože autoři tohoto



manuálu chtějí všechny manuály věnované CONTEXTu zpřístupnit elektronicky včetně zdrojových textů, je jednou z možností, jak se důvěrně obeznámit se schopnostmi CONTEXTu, zpětné inženýrství.¹⁰

30.2 Interaktivní mód

Interaktivní mód se aktivuje pomocí:

```

\setupinteraction [...,*,...]

* state          = start stop
  menu           = on off
  page           = yes no
  click          = yes no
  split          = yes no
  display        = new
  openaction     = REFERENCE
  closeaction    = REFERENCE
  openpageaction = REFERENCE
  closepageaction = REFERENCE
  calculate      = IDENTIFIER
  strut          = yes no
  width          = DIMENSION
  height         = DIMENSION
  depth          = DIMENSION
  style          = normal bold slanted boldslanted type cap small...
                COMMAND
  color          = IDENTIFIER
  contrastcolor  = IDENTIFIER
  symbolset      = IDENTIFIER
  title          = TEXT
  subtitle       = TEXT
  author        = TEXT
  date          = TEXT
  keyword       = TEXT
  fieldlayer    = auto IDENTIFIER
    
```

Například:



¹⁰ Pozn. překl.: Interaktivitě je věnován manuál `mwidget.pdf`, viz přílohu A.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupinteraction
[state=start,
color=green,
style=bold]
```

způsobí, že se začnou automaticky generovat hyperlinky a aktivní slova se zobrazí zeleně.

Interaktivní dokumenty jsou podstatně větší (v MB) než jejich papíroví bratřenci, protože hyperlinky zabírají nějaké místo. Také si všimněme, že zpracování dokumentu probíhá pomaleji. Proto je výhodné interaktivní mód během přípravných fází tvorby dokumentu deaktivovat.

30.3 Interaktivita v rámci jednoho dokumentu

Již dříve jsme viděli, jak vytvořit odkaz pomocí `\in` a `\at`. Možná jste se divili, proč píšeme `\in{kapitola}[chap:uvod]`. Zprvce se tak zamezí, aby slovo *kapitola* a korespondující číslo bylo rozděleno řádkovým zlomem. A zadruhé bude jak slovo *kapitola*, tak i její číslo sázeno v interaktivním módu aktivně. Díky tomu získá čtenář větší klikatelnou oblast.

V interaktivním módu máme k dispozici jeden příkaz, který v papírové variantě dokumentu nemá příliš význam. Je jím

```
\goto {..1.} {..2.} [...3,...]
```

- 1 TEXT
- 2 TEXT
- 3 REFERENCE

Složené závorky obsahují uvozující nebo ukončující text reference, hranatý odkaz (referenční jméno nebo pokyn).

V `\goto{Hasseltu}[fig:cityplan]` jsou všechny ulice vybudovány do kruhu.

V interaktivním dokumentu bude slovo *Hasseltu* zelené a aktivní. Skočíte jím na obrázek s mapkou Hasseltu.

30.4 Interaktivita mezi dokumenty

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Je možné odkazovat z jednoho dokumentu na jiný. Nejdříve musíme registrovat vnější dokument. Dělá se to pomocí

```
\useexternaldocument [.1.] [.2.] [.3.]
```

- 1 IDENTIFIER
- 2 file
- 3 TEXT

První hranaté závorky musí obsahovat logické jméno dokumentu, druhý pár závorek obsahuje jméno souboru a poslední závorky obklopují pojmenování vnějšího dokumentu.

Pro odkazování takto registrovaného dokumentu použijeme

```
\from [.*.]
```

- * REFERENCE

Závorky obsahují logické jméno dokumentu.

Podívejme se na následující příklad:

```
\useexternaldocument [hia] [hasbook] [Hasselt v~srpnu]
```

Většina turistických atrakcí je popsána v~knize
`\from[hia]`. Podrobnosti o~Eui||feestu je možno nalézt
 v~knize `\from[hia::euifeest]`. Popis `\goto{Eui||feestu}[hia::euifeest]`
 je v~průvodci `\from[hia]`. Eui||feest je popsán na
`\at{straně}[hia::euifeest]` v~materiálu `\from[hia]`.
 Viz další informace v~kapitole `\in{kapitole}[hia::euifeest]` knihy
`\from[hia]`.

Příkaz `\useexternaldocument` bývá použit v úvodní oblasti vstupního souboru.

Po zpracování našeho vstupního souboru (přinejmenším dvakrát, aby byly odkazy v pořádku) a souboru `hasbook.tex`, budeme mít dva PDF dokumenty. Výše uvedené odkazy mají následující význam:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

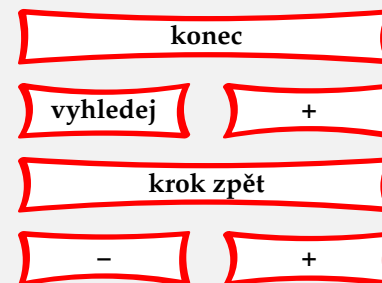
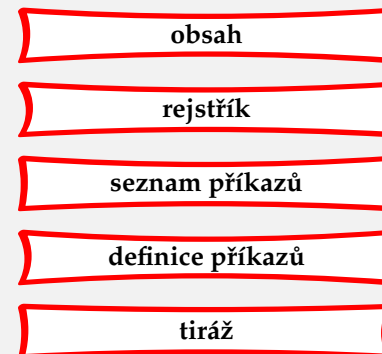
+

- `\from[hia]` vyrobí aktivní titulek, který jsme uvedli v třetím páru hranatých závorek příkazu `\useexternaldocument` a svázán bude s první stranou `hasbook.pdf`.
- `\from[hia::euifeest]` vyrobí aktivní titulek a sváže jej se stranou, kde začíná kapitola `Eui-feest`.
- `\goto{Eui||feest}[hia::euifeest]` vyrobí aktivní slovo `Eui-feest` a sváže jej se stranou, kde začíná kapitola `Eui-feest`.
- `\at{strana}[hia::euifeest]` vyrobí aktivní slovo `strana`, připojí číslo strany a sváže je s touto stranou.
- `\in{kapitola}[hia::euifeest]` vyrobí aktivní slovo `kapitola`, připojí číslo kapitoly a sváže je s příslušnou kapitolou.

Jak jsme mohli vidět, `::` odděluje (logické) jméno souboru a referenční jméno doskoku v tomto dokumentu.

30.5 Menu

Definovat můžeme také navigační panel pomocí



```

\defineinteractionmenu [1.] [2...;...] [3...,...OPTIONAL]

1 IDENTIFIER
2 left right top bottom IDENTIFIER
3 before = COMMAND
  after = COMMAND
  inbetween = COMMAND
  left = COMMAND
  right = COMMAND
  middle = COMMAND
  state = start stop none local
  style = normal bold slanted boldslanted type cap small...
        COMMAND
  color = IDENTIFIER
  distance = overlay DIMENSION
  samepage = yes empty no none
  unknownreference = yes empty no none
  leftoffset = DIMENSION
  rightoffset = DIMENSION
  topoffset = DIMENSION
  bottomoffset = DIMENSION
  position = yes no
  inherits from \setupframed
    
```

První pár závorek definuje referenční jméno, které může později být použito pro vyvolání menu. Druhé závorky určují polohu na obrazovce. Třetí nastavují menu.

Typická definice menu vypadá takto:

```

\setupcolors
 [state=start]

\setupinteraction
 [state=start,
  menu=on]

\defineinteractionmenu
 [mymenu]
 [right]
 [state=start,
    
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+


```

align=middle,
background=screen,
frame=on,
width=\marginwidth,
style=smallbold,
color=]
\startinteractionmenu[mymenu]
  \but [content] Obsah \\
  \but [index] Rejstřík \\
  \vfill \\
  \but [ExitViewer] Konec \\
\stopinteractionmenu
\setupheadertexts[{\interactionmenu[mymenu]}]

```

Tato definice vytvoří na pravé straně každé strany menu. Tlačítka menu budou obsahovat texty *Obsah*, *Rejstřík* a *Konec* s odpovídajícími funkcemi, po řadě: skok na obsah, skok na rejstřík a ukončení prohlížeče. Návěští na obvyklé doskoky, jakými jsou např. `content` a `index`, jsou předdefinovány, pokud obsah a rejstřík vložíme pomocí `\completecontent` a `\completeindex`. Jiné předdefinované funkce jsou `FirstPage` (první strana), `LastPage` (poslední strana), `NextPage` (další strana) a `PreviousPage` (předchozí strana). O vložení menu se stará `\interactionmenu`.

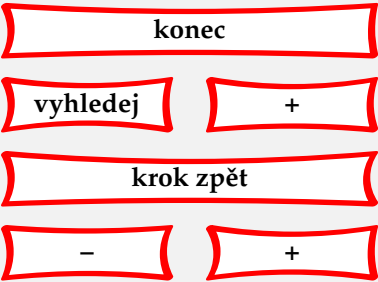
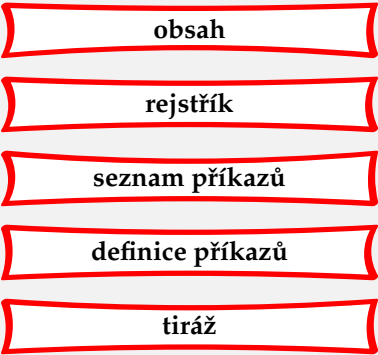
Akce typu `ExitViewer` (ukončí prohlížeč) je nezbytná, aby náš elektronický dokument byl samoobsažný. Jiné předdefinované akce, které můžeme použít jsou `PrintDocument` (tiskni), `SearchDocument` (prohledej dokument) a `PreviousJump` (skok na předchozí zobrazení).

Menu jsou nastavována pomocí:

```

\startinteractionmenu [...] ... \stopinteractionmenu
* IDENTIFIER

```



31 | Písmo a přepínače fontů

31.1 Úvod

Implicitní font v `CONTEXTu` je *Computer Modern Roman* (`cmr`). Můžete také použít font *Lucida Bright* (`lbr`) a symboly *American Mathematical Society* (`ams`). Dostupné jsou PostScriptové fonty (`pos`) a mnohé další.

31.2 Řezy písma a jejich velikost

Rodinu, řez a velikost fontu pro celý dokument vybereme pomocí:

```
\setupbodyfont [...,*,...]
```

```
* IDENTIFIER serif regular roman sans support sansserif mono type teletype  
handwritten calligraphic 5pt ... 12pt
```

Pokud použijeme `\setupbodyfont[cmr,sansserif,9pt]`

v úvodní deklarační oblasti vstupního souboru, text by měl vypadat asi takto.

Pro změnu uprostřed dokumentu, nebo na úrovni sekce, bychom měli použít:

```
\switchtobodyfont [...,*,...]
```

```
* 5pt ... 12pt small big global
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

10.\,listopadu (jeden den před dnem svatého Martina) chodí omladina města Hasselt dům od domu a zpívá zvláštní píseň určenou jen pro tuto příležitost. Svůj zpěv doprovází na nástroj {\em foekepot}. Neodejdou, dokud jim nedáte nějaké peníze, nebo sladkosti. Píseň zní asi takto:

```
\startnarrower
\switchtobodyfont[small]
\startlines
Foekepotterij, foekepotterij,
Geef mij een centje dan ga'k voorbij.
Geef mij een alfje dan blijf ik staan,
'k Zal nog liever naar m'n arrenmoeder gaan.
Hier woont zo'n rieke man, die zo vulle g\grave{e}ven kan.
G\grave{e}f wat, old wat, g\grave{e}f die arme stumpers wat,
'k Eb zo lange met de foekepot elopen.
'k Eb gien geld om brood te kopen.
Foekepotterij, foekepotterij,
Geef mij een centje dan ga'k voorbij.
\stoptlines
\stopnarrower
```

Poznamenejme ještě, že `\startnarrower … \stopnarrower` se také užívá jako začátek a konec přepnutí fontu. Funkce `\startlines` a `\stoptlines` v tomto příkladu je zřejmá.

10. listopadu (jeden den před dnem svatého Martina) chodí omladina města Hasselt dům od domu a zpívá zvláštní píseň určenou jen pro tuto příležitost. Svůj zpěv doprovází na nástroj *foekepot*. Neodejdou, dokud jim nedáte nějaké peníze, nebo sladkosti. Píseň zní asi takto:

Foekepotterij, foekepotterij,
 Geef mij een centje dan ga'k voorbij.
 Geef mij een alfje dan blijf ik staan,
 'k Zal nog liever naar m'n arrenmoeder gaan.
 Hier woont zo'n rieke man, die zo vulle gèven kan.
 Gèf wat, old wat, gèf die arme stumpers wat,
 'k Eb zo lange met de foekepot elopen.
 'k Eb gien geld om brood te kopen.
 Foekepotterij, foekepotterij,
 Geef mij een centje dan ga'k voorbij.

Pokud chceme získat přehled dostupných druhů fontu zvolené rodiny, zkusme vysázet:

```
\showbodyfont[cmr]
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

	[modern] [cmr]							\mr : Ag					
	\tf	\sc	\sl	\it	\bf	\bs	\bi	\tfx	\tfxx	\tfa	\tfb	\tfc	\tfd
\rm	Ag	AG	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\ss	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
\tt	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

31.3 Přepínače řezu a velikosti v parametrech příkazů

V mnoha příkazech jeden z parametrů indikuje požadovaný styl sazby. Například:

```
\setuphead[chapter][style=\tfd]
```

V tomto případě je velikost písma nadpisu kapitol nastavena na \tfd. Místo takového příkazu bychom ale měli používat předdefinované volby, které se vztahují k aktuálnímu druhu fontu:

```
normal bold slanted boldslanted type mediaeval
small smallbold smallslanted smallboldslanted smalltype
capital cap
```

31.4 Lokální řez písma a jeho velikost

V běžném textu můžeme měnit *styl písma* na roman, bezpatkové a neproporcionální (strojopisné) pomocí příkazů \rm, \ss a \tt.

Řez písma, jako kurzíva, skloněné nebo tučné, můžeme měnit pomocí \it, \sl a \bf.

Velikost písma lze měnit v rozmezí od 4pt do 12pt a nastavuje se pomocí \switchtobodyfont.

Aktuální styl je v příkazu přednastaven pomocí \tf. Pokud chceme zvětšit velikost písma, použijeme \tfa, \tfb, \tfc a \tfd. Příponu a, b, c a d lze přidávat také k příkazům \it, \sl nebo \bf.

```
{\tfc Ražba mincí}
```

V období od roku {\tt 1404} do {\tt 1585} vlastnil Hasselt {\sl právo ražení mincí}. Proti této výsadě Hasseltu ostatní města protestovala, ale

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
{\switchtobodyfont[7pt] biskup z Utrechtu} tyto protesty  
nevyslyšel.
```

Složená závorka indikuje začátek a konec oblasti, která je ovlivněna přepínačem stylu nebo velikosti.

Ražba mincí

V období od roku 1404 do 1585 vlastnil Hasselt *právo ražení mincí*. Proti této výsadě Hasseltu ostatní města protestovala, ale `biskup z Utrechtu` tyto protesty nevyslyšel.

31.5 Předefinování velikosti fontu

Ve zvláštních případech můžeme definovat svou vlastní velikost fontu.

```
\definebodyfont [...1,...] [...2.] [...3.,...]
```

```
1 5pt ... 12pt default
```

```
2 rm ss tt mm hw cg
```

```
3 tf = FILE
```

```
bf = FILE
```

```
sl = FILE
```

```
it = FILE
```

```
bs = FILE
```

```
bi = FILE
```

```
sc = FILE
```

```
ex = FILE
```

```
mi = FILE
```

```
sy = FILE
```

```
ma = FILE
```

```
mb = FILE
```

```
mc = FILE
```

Tato definice by měla vypadat asi takto:

```
\definebodyfont [10pt] [rm] [tfe=Regular at 36pt]
```

```
{\tfe Hasselt!}
```

Nyní příkaz `\tfe` vysází velikostí 36pt nápis: **Hasselt!**

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

31.6 Kapitálky

Zkratky jako PDF (Portable Document Format) jsou tištěny pomocí malých kapitálek. Malé kapitálky jsou o něco menší, než počáteční písmena vět v aktuálním fontu a mohou být vysázeny pomocí:

```
\cap {...}
```

```
* TEXT
```

Pokud porovnáme PDF, `\cap{PDF}` a `\sc pdf`
 PDF, PDF a pdf
 PDF, PDF a PDF (Computer Modern Roman)

můžeme vidět zřetelně rozdíl. Příkaz `\sc` vytváří skutečné kapitálky (musí existovat jako samostatný font), zatímco pseudokapitálky `\cap` jsou vytvořeny geometrickou deformací verzálek.

31.7 Zvýraznění

Pro jednotné zvýraznění slov v dokumentu používáme příkaz:

```
\em
```

Zvýrazněná slova se implicitně vysází skloněným stylem.

Při procházce Hasseltem bychom měli `{\bf \em dávat pozor}` na `{\em Amsterodamy}`. Slovem `{\em Amsterodamy}` `{\bf \em nejsou}` míněni obyvatelé Amsterodamu, ale malé kamenné pilíře, oddělující chodník od silnice. Jsou zamýšleny jako ochrana chodců před projíždějícími auty, ale ve skutečnosti se lidé častěji zraní, když o ně zakopnou.

Při procházce Hasseltem bychom měli **dávat pozor** na *Amsterodamy*. Slovem *Amsterodamy nejsou* míněni obyvatelé Amsterodamu, ale malé kamenné pilíře, oddělující chodník od silnice. Jsou zamýšleny jako ochrana chodců před projíždějícími auty, ale ve skutečnosti se lidé častěji zraní, když o ně zakopnou.

Pokud něco dvakrát zdůrazníme, vysází se to opět nezdůrazněně.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

31.8 Strojopis alias režim verbatim

Pokud si přejeme zobrazit text přesně tak, jak jsme ho uvedli do vstupního souboru, s ukončením řádek přesně tam, kde jste použili ENTER, použijeme:

```
\starttyping ... \stoptyping
```

V textu můžeme použít:

```
\type {.*.}
```

```
* TEXT
```

Složené závorky ohraničují strojopisný text. Sestavujme jej opatrně, protože v něm již nefunguje běžný mechanismus zalamování řádků.

Nastavení režimu verbatim můžeme provádět pomocí:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setuptyping [..1.] [..,2.,...]  
                OPTIONAL
```

```
1 file typing IDENTIFIER  
2 space      = on off  
  page      = yes no  
  option    = slanted normal commands color none  
  text      = yes no  
  icommand  = COMMAND  
  vcommand  = COMMAND  
  ccommand  = COMMAND  
  before    = COMMAND  
  after     = COMMAND  
  margin    = DIMENSION standard yes no  
  evenmargin = DIMENSION  
  oddmargin = DIMENSION  
  blank     = DIMENSION small medium big standard halflines line  
  escape    =  
  space     = on off  
  tab       = NUMBER yes no  
  page      = yes no  
  indentnext = yes no  
  style     = normal bold slanted boldslanted type cap small... COMMAND  
  color     = IDENTIFIER  
  palet     = IDENTIFIER  
  lines     = yes no hyphenated  
  empty     = yes all no  
  numbering = line file no
```

```
\setuptype [..,*.,...]
```

```
* space = on off  
  option = slanted normal none  
  style = normal bold slanted boldslanted type cap small... COMMAND  
  color = IDENTIFIER
```

31.9 Kódování znaků

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Každý font je připraven v jednom nebo více kódováních. Při zavedení fontu pomocí fontových definičních tabulek, tzv. `typescriptů`, je nutné toto kódování specifikovat. Pro kódování `ec`, používané v Evropě nejčastěji, vypadá načtení písma `Computer Modern` takto:

```
\usetypescript [modern] [ec]
\setupbodyfont [modern, 10pt, rm]
```

Typ kódování fontu volíme podle toho, v jakém kódování máme vzory dělení slov v použitém jazyce. Toto kódování se dovíme zapsáním `\showpatterns`, viz **tabulka 31.1**.

language	encoding	mapping	number	left _{min}	right _{min}
en	ec	ec	1	2	3
uk	ec	ec	2	3	3
de	texnansi	texnansi	3	2	2
de	ec	ec	4	2	2
fr	texnansi	texnansi	5	2	2
fr	ec	ec	6	2	2
es	ec	ec	7	2	2
pt	texnansi	texnansi	8	2	2
pt	ec	ec	9	2	2
it	texnansi	texnansi	10	2	2
it	ec	ec	11	2	2
nl	texnansi	texnansi	12	2	2
nl	ec	ec	13	2	2
cz	il2	il2	14	2	2
cz	ec	ec	15	2	2
sk	il2	il2	16	2	2
sk	ec	ec	17	2	2
pl	pl0	pl0	18	2	2
pl	ec	ec	19	2	2
pl	qx	qx	20	2	2

Tabulka 31.1 Výpis příkazu `\showpatterns`.

Přípravu fontu se správným kódováním i s předgenerováním `typescriptů` lze svěřit programku `texfont`.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Pokud náš editor pracuje v jiném kódování, než je kód fontu, pak je nutné text překódovat. O to se postará `CONTEXT`, pokud vstupní kódování našeho dokumentu uvedeme v hranaté závorce příkazu:

```
\enableregime [...]
```

```
* Latin1 Latin2 Latin7 Latin15 iso-8859-1 iso-8859-2 iso-8859-7 iso-8859-15
  cp1250 cp1251 cp1252 cp1253 cp1254 cp1257 viscii cyr grk mac utf8
  unicode
```



32.1 Úvod

V **kapitole 3** jste již viděli, že pro vysázení speciálních znaků jako # \$ % & _ { a } musíme napsat více než jeden symbol. Akcentovaná písmena nebo cizokrajné symboly musí být složeny z více znaků, aby byl vytvořen správně vysázený znak.

Nad rámeček této příručky vychází problematika akcentovaných písmen v matematickém režimu. Informace o této problematice najdete v `TEX`Booku Donalda E. Knutha.

32.2 Akcenty

Kromě přímého vstupu znaků s diakritikou pomocí vhodného kódování (viz **kapitolu 31.9**), můžeme v `CONTEXT`u akcentovaná písmena složit z jednoduchých znaků a diakritických znamének. **Tabulka 32.1** ukazuje, jak kompozit vytvoříme. Písmeno *u* je uvedeno jenom jako příklad.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Napíšeme	Dostaneme	Napíšeme	Dostaneme
\`{u}	ù	\u{u}	ű
\' {u}	ú	\v{u}	ű
\^{u}	û	\H{u}	ű
\" {u}	ü	\b{u}	u
\~{u}	ü	\d{u}	u
\={u}	ü	\c{u}	u
\. {u}	ü	\k{u}	u
\r{u}	ü		

Tabulka 32.1 Akcenty

Asi nechceme psát písmena ì nebo ĵ (s tečkami). Z písmen *i* a *j* vytvoříme složené znaky následujícím způsobem:

\{"\i} ì

\^{j} ĵ (j bez tečky nemusí být ve všech písmech k dispozici)

32.3 Cizokrajné symboly

Jak se sestavují písmena a znaky, které se vyskytují v cizích jazycích, uvádí **tabulka 32.2**.

Napíšeme	Dostaneme	Napíšeme	Dostaneme
\oe	œ	\O	Ø
\OE	Œ	\l	ł
\ae	æ	\L	Ł
\AE	Æ	\SS	ß
\aa	å	?´	ı
\AA	Å	!´	ı
\o	ø	\texteuro	€

Tabulka 32.2 Cizokrajná písmena a znaky

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Mnoho dalších symbolů je definováno v externích množinách. Zavádějí se příkazem

```
\usesymbols [.*.]
```

```
* IDENTIFIER
```

V hranatých závorkách je troj písmenný název množiny, např. eur, nav, mvs, jmn, uni. Seznam znaků dané množiny se zobrazí příkazem

```
\showsymbolset [.*.]
```

```
* IDENTIFIER
```



Formát stránky je definován příkazem:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej +

krok zpět

- +

```
\setuplayout [...,*...]
```

```
* width          = DIMENSION fit middle
height          = DIMENSION fit middle
backspace       = DIMENSION
topspace        = DIMENSION
margin          = DIMENSION
leftmargin      = DIMENSION
rightmargin     = DIMENSION
header          = DIMENSION
footer         = DIMENSION
top             = DIMENSION
bottom          = DIMENSION
leftedge        = DIMENSION
rightedge       = DIMENSION
headerdistance = DIMENSION
footerdistance = DIMENSION
topdistance     = DIMENSION
bottomdistance = DIMENSION
leftmargindistance = DIMENSION
rightmargindistance = DIMENSION
leftedgedistance = DIMENSION
rightedgedistance = DIMENSION
horoffset       = DIMENSION
veroffset       = DIMENSION
style           = normal bold slanted boldslanted type cap small...
                COMMAND

color           = IDENTIFIER
marking         = on off color screen TEXT
location        = left middle right bottom top singlesided doublesided
scale           = DIMENSION
nx              = NUMBER
ny              = NUMBER
dx              = DIMENSION
dy              = DIMENSION
lines           = NUMBER
columns         = NUMBER
columndistance = DIMENSION
grid            = yes no
bottomspace     = DIMENSION
cutspace       = DIMENSION
textdistance    = DIMENSION
textwidth       = NUMBER
textmargin     = DIMENSION
clipoffset      = DIMENSION
page            = IDENTIFIER
paper           = IDENTIFIER
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Tento povel se píše do deklarační oblasti vstupního souboru.

Měli bychom se obeznámit s parametry, které popisují formát stránek (stránkové zrcadlo). Stránka je rozdělena na několik oblastí jako například text, záhlaví, úpatí atd. Velikosti těchto částí se dají nastavit příkazem `\setuplayout`.

Různé oblasti ve formátu stránky jsou popsány na **obrázku 33.1**. Okraj je také nazýván marginálií.

	left edge	left margin	text	right margin	right edge
top					
header					
text					
footer					
bottom					

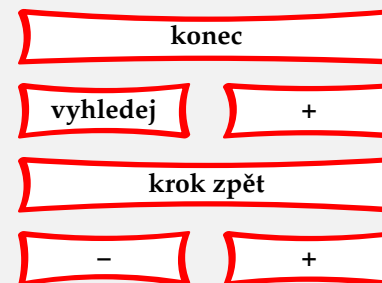
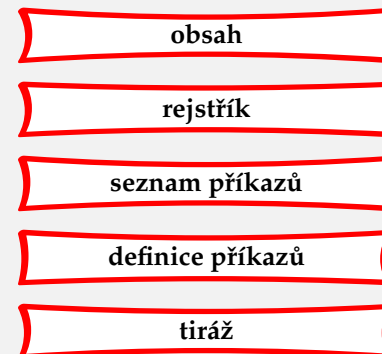
Obrázek 33.1 Formát stránky.

Pokud chceme vidět formát stránkového zrcadla, můžeme napsat příkaz `\showframe` a zpracovat jednu stranu nebo celý soubor. Oblasti jsou pak na stránce vyznačeny několika barevnými rámečky.

Příkaz `\showsetups` ukáže hodnoty parametrů. Kombinací obou příkazů je povel `\showlayout`.

Hodnoty formátovacích parametrů jsou dostupné jako příkazy (viz **tabulku 33.2**). To nám umožňuje přesněji definovat rozměry sloupců, obrázků a tabulek. Některé z těchto parametrů jsou vysvětleny v **tabulce 33.1**.

Jestliže chceme definovat šířku sloupce nebo výšku obrázku, můžeme to udělat relativně k hodnotám `\makeupwidth` nebo `\makeupheight`. Změny nastavení šířky nebo výšky stránky pak proporcionalně změní rozměry sloupce nebo obrázku.



Příkazy	Význam
<code>\makeupwidth</code>	šířka zrcadla sazby
<code>\makeupheight</code>	výška zrcadla sazby
<code>\textwidth</code>	šířka textové oblasti
<code>\textheight</code>	výška textové oblasti

Tabulka 33.1 Několik parametrů ve formě příkazů.

```
\placefigure
[here]
[fig:stepgable]
{Stupňovitá štítová zeď.}
{\externalfigure[ma-cb-19][width=.6\textwidth]}
```

Po zpracování dostaneme:



Obrázek 33.2 Stupňovitá štítová zeď.

Ostatní vzdálenosti a rozměry jsou uvedeny v **tabulce 33.2**.

obsah
rejstřík
seznam příkazů
definice příkazů
tiráž

konec	
vyhledej	+
krok zpět	
-	+

Příkaz `\setuplayout` je definován v deklarační oblasti vstupního souboru, před povel `\starttext`. To znamená, že hodnoty parametrů jsou globální a platné pro celý dokument. Nicméně pokud pro některé stránky chceme vykonat drobné změny rozvržení stránky, použijeme

```
\adaptlayout [21,38] [height=+.5cm]
```

V tomto případě budou mít strany 21 a 38 výšku rovnou 0,5 cm + výška stránky.

Na lokální změny formátu stránky můžete použít příkaz:

```
\startlocal ... \stoplocal
```

Například:

```
\start
\startlocal
\setuplayout [height=+.5cm]
\stoplocal
```

Hasselt má úplně jiné uspořádání než většina měst kvůli svým opevněním a-hradním příkopům.

```
\stop
```

Neradíme používat tyto lokální změny příliš často. Vždycky je lepší pozměnit text nežli měnit formát stránky.

Parametr	Příkaz
vzdálenost vršku	<code>\topdistance</code>
výška vršku	<code>\topheight</code>
vzdálenost záhlaví	<code>\headerdistance</code>
výška záhlaví	<code>\headerheight</code>
horní mezera	<code>\topspace</code>
úroveň hlavičky	<code>\headlevel</code>
vnitřní okraj	<code>\backspace</code>

Tabulka 33.2.a Parametry pro úpravu stránky.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

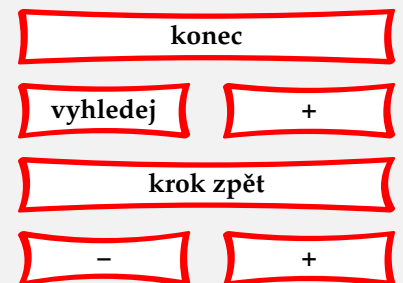
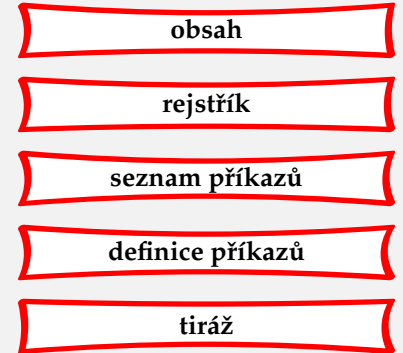
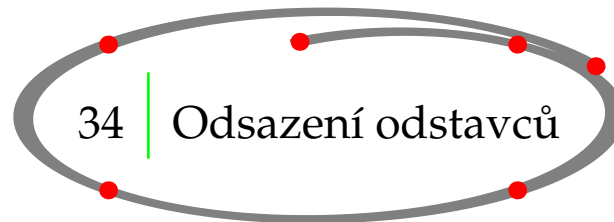
krok zpět

-

+

vzdálenost marginálií	<code>\margindistance</code>
šířka každé z marginálií	<code>\marginwidth</code>
šířka levé marginálie	<code>\leftmarginwidth</code>
šířka pravé marginálie	<code>\rightmarginwidth</code>
vzdálenost hran	<code>\edgedistance</code>
šířka každé z hran	<code>\edgewidth</code>
šířka levé hrany	<code>\leftedgewidth</code>
šířka pravé hrany	<code>\rightedgewidth</code>
výška papíru	<code>\paperheight</code>
šířka papíru	<code>\paperwidth</code>
výška zrcadla	<code>\makeupheight</code>
šířka zrcadla	<code>\makeupwidth</code>
výška textu	<code>\textheight</code>
šířka textu	<code>\textwidth</code>
vzdálenost úpatí	<code>\footerdistance</code>
výška úpatí	<code>\footerheight</code>
vzdálenost spodku	<code>\bottomdistance</code>
výška spodku	<code>\bottomheight</code>

Tabulka 33.2.b Parametry pro úpravu stránky.



V `TeXu` a `CONTEXTu` je nejdůležitější jednotkou textu odstavec. Nový odstavec můžeme začít

- prázdným řádkem,
- příkazem `\par`.

Ve vstupním `ASCII` souboru je vhodné jako oddělovače odstavců využívat prázdné řádky. Dostaneme tak přehlednější a lépe strukturovaný text, ve kterém se snadněji odhalují případné chyby.

Pokud si přejeme ukončit odstavec výslovně příkazem, použijeme `\par`.

```
Jednou byl ve válce Hasselt obléhán. Po nějaké
době na město dolehl hladomor a~vše, co se dalo sníst,
už bylo snědeno, kromě jedné krávy. Ta byla ponechána naživu
a~navíc se jí dostávalo velice slušného zacházení. \par
Jednou denně vyváděli obyvatelé Hasseltu tuto krávu
na procházku po hradbách. Když obléhatelé viděli
dobře živenou krávu, propadali malomyslnosti. Nakonec
obléhání ukončili a~Hasselt byl zachráněn.\par
V~Hoogstraatu se dnes na památku tohoto obležení
a~moudrosti obyvatel nachází kamenná deska
s~vyobrazením krávy.
```

Místo příkazů `\par` bychom mohli také použít několik prázdných řádků:

```
Jednou byl ve válce Hasselt obléhán. Po nějaké
době na město dolehl hladomor a~vše, co se dalo sníst,
už bylo snědeno, kromě jedné krávy. Ta byla ponechána naživu
a~navíc se jí dostávalo velice slušného zacházení.

Jednou denně vyváděli obyvatelé Hasseltu tuto krávu
na procházku po hradbách. Když obléhatelé viděli
dobře živenou krávu, propadali malomyslnosti. Nakonec
obléhání ukončili a~Hasselt byl zachráněn.

V~Hoogstraatu se dnes na památku tohoto obležení
a~moudrosti obyvatel nachází kamenná deska
s~vyobrazením krávy.
```

34.2 Mezera mezi odstavci

Velikost mezery mezi odstavci lze nastavit pomocí:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupwhitespace [...]  

OPTIONAL  

* none small medium big line fixed fix DIMENSION
```

Tento dokument je vysázen s nastavením `\setupwhitespace[medium]`.

Když je velikost meziodstavcové mezery nastavena, máme k dispozici dva příkazy, které jsou nicméně potřeba jen zřídka:

```
\nowhitespace  

\whitespace
```

Pokud odstavec obsahuje vodorovné linky, nebo orámovaný text, jako například

```
Ridderstraat 27, 8061GH Hasselt
```

pak se stává, že rozestup není optimální. V takovém případě lze provést korekci pomocí:

```
\startlinecorrection ... \stoplinecorrection
```

Pokud tedy napíšeme:

```
\startlinecorrection  

\framed{Ridderstraat 27, 8061GH Hasselt}  

\stoplinecorrection
```

dostaneme lepší výstup. Tyto příkazy ale použijeme, jen pokud je to opravdu nezbytné!

```
Ridderstraat 27, 8061GH Hasselt
```

Další příkaz pro ovládání mezery mezi odstavci je:

```
\blank [...*...]  

OPTIONAL  

* small medium big nowhite back white disable force reset line halflin  

FORMULA fixed flexible none
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Hranaté závorky za příkazem jsou nepovinné a umožňují nastavit velikost mezery mezi odstavci. Klíčová slova jako `small`, `medium` a `big` nastavují mezeru relativně vzhledem k velikosti daného fontu.

Oficiální název Hasselt má vždy příponu `Ov`, což je zkratka provincie `{\em Overijssel}`.

`\blank[2*big]`

Úsměvné na tom je, že v~Holandsku je Hasselt pouze jeden a~tato přípona je tedy nadbytečná.

`\blank`

Tato přípona je pozůstatkem z~dob, kdy Holandsko a~Belgie byly jednou zemí pod vládou španělského krále Filipa II.

`\blank[2*big]`

Hasselt v~Belgii leží v~provincii Limburg, ale Belgičané na své dopisy Hasselt (Li) zřejmě nepíší.

Příkaz `\blank` bez hranatých závorek představuje obvyklou mezeru.

Náš příklad by vypadal:

Oficiální název Hasselt má vždy příponu `Ov`, což je zkratka provincie *Overijssel*.

Úsměvné na tom je, že v Holandsku je Hasselt pouze jeden a tato přípona je tedy nadbytečná.

Tato přípona je pozůstatkem z dob, kdy Holandsko a Belgie byly jednou zemí pod vládou španělského krále Filipa II.

Hasselt v Belgii leží v provincii Limburg, ale Belgičané na své dopisy Hasselt (Li) zřejmě nepíší.

Velikost obvyklé mezery lze ovládat pomocí:

```
\setupblank [.*.]
              OPTIONAL
* normal standard line DIMENSION big medium small fixed flexible
```

Pokud chceme potlačit tuto meziodstavcovou mezeru, můžeme použít:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\startpacked [...] ... \stoppacked
                OPTIONAL
* blank
```

V tomto dokumentu je mezera nastavena na medium ale na následujících řádcích je tato mezera potlačena a řádky jsou tedy semknuty k sobě.

```
\startpacked
Hasselt (Ov) leží v~provincii Overijssel.
Hasselt (Li) leží v~provincii Limburg.
Pozor: Mluvíme o~Limburgu v~Belgii. Existuje také
Limburg v~Holandsku.
\stoppacked
```

Vysázeno bude:

Hasselt (Ov) leží v provincii Overijssel.
Hasselt (Li) leží v provincii Limburg.
Pozor: Mluvíme o Limburgu v Belgii. Existuje také Limburg v Holandsku.

Není těžké si představit, k čemu slouží:

```
\startunpacked ... \stopunpacked
```

Vertikální mezeru si lze také vynutit pomocí příkazu `\godown`. Její velikost je nastavena v závorkách.

```
\godown [...]
* DIMENSION
```

34.3 Odsazení

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Velikost odsazení nastavíme pomocí:

```
\setupindenting [...*...]
```

```
* never none not no yes always first next small medium big normal odd
  even DIMENSION
```

Rozumného odsazení se dosáhne pomocí:

```
\setupindenting[medium]
```

Tímto bude první řádka odstavce odsazená. Jen po mezeře (vysázené příkazem `\blank`) je v základním nastavení odsazení potlačeno.

Lokálně na určitém místě můžeme odsazování ovládat pomocí:

```
\indenting [...*...]
```

```
* never none not no yes always first next small medium big normal odd
  even DIMENSION
```

Když například použijeme `never`, budou od toho okamžiku všechna odsazení potlačena. Příkaz `none` ovlivní pouze následující odstavec.

Pokud používáme odsazování a na určitém místě si odsazení nepřejeme, můžeme také použít:

```
\noindenting
```



obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej +

krok zpět

- +

Globální příkazy se umístí ují v deklarační oblasti vstupního souboru, před povelem `\starttext`. V **příloze B** je uveden přehled dostupných příkazů a jejich parametrů.

Nastavovací příkazy mají stejnou strukturu. Vypadají třeba takto:

```

\setupparagraphs [.1.] [.2.] [..., .3., ...]
                                OPTIONAL

1 IDENTIFIER
2 NUMBER each
3 style      = normal bold slanted boldslanted type cap small... COMMAND
  width      = DIMENSION
  height     = DIMENSION
  align      = inner outer left right middle normal no yes
  tolerance  = verystRICT strict tolerant verytolerant stretch
  distance   = DIMENSION
  before     = COMMAND
  after      = COMMAND
  inner      = COMMAND
  command    = COMMAND
  rule       = on off
    
```

Deklarační povely se více nebo méně skládají z logického jména a určitého počtu dvojic hranatých závorek. Tyto můžou být nepovinné, v tom případě je pod nimi napsáno slovo `OPTIONAL`. Dvojice hranatých závorek mohou v definici obsahovat:

```
\setupacommand [.1.] [.2.] [..., . = ., ...]
```

Čárky ve třetí záorce naznačují, že zde může být vložen seznam parametrů, rovnítko značí, že jde o parametry typu *klíč=hodnota*. Čísla `.1.` a `.2.` odkazují na vysvětlivky za definicí. Ty nám říkají, jaké volby můžeme uvést v první, druhé a třetí hranaté záorce.

Implicitní volby a hodnoty parametrů jsou podtrženy. Navíc si můžeme všimnout, že některé hodnoty jsou vysázeny verzálkami: `SECTION`, `NAME`, `DIMENSION`, `NUMBER`, `COMMAND`, and `TEXT`. Toto naznačuje, že nejde o konstantní hodnoty, ale o hodnoty určitého typu:

- `SECTION` název oddílu, například `chapter`, `section`, `subsection` atd.;
- `NAME` identifikátor (logické jméno);
- `DIMENSION` rozměr v jednotkách `cm`, `pt`, `em`, `ex`, `sp` nebo `in`;
- `NUMBER` celé číslo;

- obsah
- rejstřík
- seznam příkazů
- definice příkazů
- tiráž

- konec
- vyhledej +
- krok zpět
- +

COMMAND příkaz s daným počtem parametrů;
 TEXT text.

36 | Definování příkazů / maker

CONTEXT je sada maker založená na T_EXu. T_EX je programovací jazyk a zároveň typografický systém. To znamená, že si můžeme sami naprogramovat určité vlastnosti, pokud takovou flexibilitu potřebujeme.

Nový příkaz můžeme nadefinovat pomocí

```
\define [.1.] \2. {3.}
      OPTIONAL
1  NUMBER
2  NAME
3  TEXT
```

Následující příklad objasní jeho použití.

Můžeme mít bohatě ilustrovaný dokument a pokud jsme již unavení psaním

```
\placefigure
[here,force]
[fig:logické jméno]
{Popisek.}
{\externalfigure[soubor] [width=5cm]}
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

můžeme si definovat vlastní příkaz s několika proměnnými:

- logické jméno;
- popisek;
- soubor.

Naše definice příkazu a jeho volání může vypadat takto:

```
\define [3] \mujobrazek
  {\placefigure
   [here,force] [fig:#1]
   [#2]{\externalfigure [#3] [width=5cm]}}
\mujobrazek{lev}{Holandský lev na stráží.}{ma-cb-13}
```

V hranatých závorkách [3] stanovujeme, že chceme tři proměnné #1, #2 a #3. Při volání příkazu \mujobrazek musíme vložit tyto tři proměnné hodnoty mezi složené závorky. Výsledkem bude:



Obrázek 36.1
Holandský lev na stráží.

Programování sofistikovanějších příkazů je ponecháno na čtenáři.

Kromě definování příkazů můžeme také navíc definovat dvojici \start ... \stop.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\definestartstop [..1.] [..,2.,..]
```

```
1 IDENTIFIER
2 before = COMMAND
  after  = COMMAND
  style  = normal bold slanted boldslanted type cap small... COMMAND
  commands = COMMAND
```

Například:

```
\definestartstop
[hvezdicky]
[commands={\inleft{\hbox to \leftmarginwidth
{\leaders\hbox{$\star$}\hfill}}},
before=\blank,
after=\blank]
```

```
\starthvezdicky
{\em Hasselter Juffers} jsou sladké koláčky, ale jejich
název není hříčka náhody. V~červenci roku 1233 byl založen
klášter {\em Zwartewaterklooster} (Černá voda). Klášter byl
zamýšlen pro neprovdané dívky a~ženy patřící k~šlechtě Hasseltu.
Tyto dívky a~ženy byly nazývány {\em juffers}.
\stophvezdicky
```

Výsledkem bude:

Hasselter Juffers jsou sladké koláčky, ale jejich název není hříčka náhody. V červenci roku 1233 byl založen klášter *Zwartewaterklooster* (Černá voda). Klášter byl zamýšlen pro neprovdané dívky a ženy patřící k šlechtě Hasseltu. Tyto dívky a ženy byly nazývány *juffers*.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

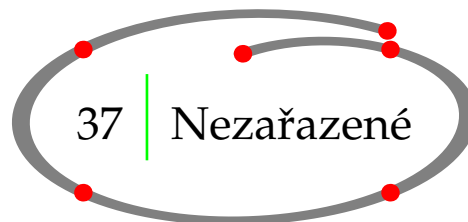
vyhledej

+

krok zpět

-

+



37.1 Titulní strana

V prvním příkladu tohoto manuálu na **straně 3** jsme použili příkaz

```
\startnamemakeup [...,*... ] ... \stopnamemakeup
                    OPTIONAL
* inherits from \startmakeup
```

Tento příkaz může být použit k definování *titulních stran*, protože mívají jinak velká stránková zrcadla než hlavní textová část dokumentu. Často k návrhu vzhledu takové výjimečné strany využíváme předdefinované dvojice `\start ... \stopstandardmakeup`.

Jednoduchá titulní strana může vypadat takto:

```
\startstandardmakeup
\blank
\rightaligned{\tfd Hasselt v~21.\,století}
\blank
\rightaligned{\tfb Budoucnost}
\vfill
\rightaligned{\tfa C. van Marle}
\rightaligned{Hasselt, 2001}
\stopstandardmakeup
```

V oboustranném dokumentu musíme vykonat některé další akce, abychom vysázeli zadní stranu titulního listu. Především příklad v takovém případě změníme na:

```
\startstandardmakeup [doublesided=no]
\blank
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\rightaligned{\tfd Hasselt v~21.\,století}
\blank
\rightaligned{\tfb Budoucnost}
\vfill
\rightaligned{\tfa C. van Marle}
\rightaligned{Hasselt, 2001}
\stopstandardmakeup
\startstandardmakeup[page=no]
\vfill
\copyright 2001

```

Tato kniha je věnována lidem žijícím v~Hasseltu. Chceme poděkovat fotografovi J.~Jonkerovi za zfalšování fotografií do této knihy tak, že čtenář získá jasnou představu o~budoucí podobě Hasseltu.

```
\stopstandardmakeup
```

Naše vlastní titulní strana může být navržena pomocí příkazů

```
\definemakeup [..1.] [..,2..,..]
```

- 1 IDENTIFIER
- 2 *inherits from* \setupmakeup

a

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupmakeup [.1.] [.,.2.,...]
```

```
1 IDENTIFIER
2 width      = DIMENSION
  height     = DIMENSION
  voffset    = DIMENSION
  hoffset    = DIMENSION
  page       = left yes right
  commands  = COMMAND
  doublesided = yes no empty
  headerstate = normal stop start empty none nomarking
  footerstate = normal stop start empty none nomarking
  textstate  = normal stop start empty none nomarking
  topstate   = stop start
  bottomstate = stop start
  pagestate  = stop start
  color      = IDENTIFIER
```

37.2 Plovoucí objekty

Plovoucím objektem (floating block) se v `CONTEXT`u nazývá textový element, např. tabulka či obrázek, který je zpracován speciálním způsobem. Již jsme viděli použití `\placefigure` a `\placetable`. Oba jsou příkladem plovoucích objektů popsanych v **kapitolách 10 a 11**.

Svůj vlastní plovoucí objekt si můžeme definovat pomocí

```
\definefloat [.1.] [.2.]
```

```
1 SINGULAR NAME
2 PLURAL NAME
```

Složené závorky se používají pro jméno objektu v jednotném a množném čísle. Například:

```
\definefloat [vsuvka] [vsuvky]
```

Nyní jsou k dispozici následující příkazy:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\placevsuvka[] []{}{}
\startvsuvkatext ... \stopvsuvkatext
\placelistofvsuvky
\completelistofvsuvky

```

Nově definovaný plovoucí objekt může být nastaven pomocí

```
\setupfloat [.1.] [...,.2,...]
```

```

1 IDENTIFIER
2 height          = DIMENSION
  width           = DIMENSION
  maxheight       = DIMENSION
  maxwidth        = DIMENSION
  minwidth        = DIMENSION
  default         = IDENTIFIER
  pageboundaries  = LIST
  leftmargindistance = DIMENSION
  rightmargindistance = DIMENSION
  location        = left middle right
  inherits from \setupframed

```

Dále můžeme nastavit formátování všech plovoucích objektů pomocí příkazu

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupfloats [...,*,...]
```

```
* location           = left right middle
width                = fit DIMENSION
before               = COMMAND
after                = COMMAND
margin               = DIMENSION
spacebefore          = small medium big none
spaceafter           = small medium big none
sidespacebefore      = small medium big none
sidespaceafter       = small medium big none
indentnext           = yes no
ntop                 = NUMBER
nbottom              = NUMBER
nlines               = NUMBER
default              = IDENTIFIER
tolerance             = 0 1 2
leftmargindistance   = DIMENSION
rightmargindistance  = DIMENSION
sidealign             = normal line
numbering             = yes nocheck
inherits from \setupframed
```

Číslování a popis objektu lze nastavit pomocí

```
\setupcaption [..1..] [...,2,...]
```

```
1 IDENTIFIER
2 inherits from \setupcaptions
```

a

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\setupcaptions [...,*.,...]

* location    = top bottom none high low middle
  width       = fit broad max DIMENSION
  minwidth    = fit DIMENSION
  headstyle   = normal bold slanted boldslanted type cap small... COMMAND
  style       = normal bold slanted boldslanted type cap small... COMMAND
  number      = yes no
  inbetween   = COMMAND
  align       = inner outer left right middle normal no yes
  conversion  = numbers characters Characters romannumerals Romannumerals
  way         = bytext bycd:section
  separator   = TEXT
  stopper     = TEXT
  command     = COMMAND
  distance    = DIMENSION

```

Tyto příkazy používáme v úvodní části vstupního souboru. `\setupfloats` a `\setupcaptions` mají globální vliv na všechny plovoucí objekty. Následuje kompletní ukázka nového plovoucího objektu.

```

\setupfloats[location=middle]
\setupcaption[location=bottom,headstyle=boldslanted]
\definefloat[vsuvka][vsuvky]
\placevsuvka{Ukázka vsuvky.}
\startframedtext
Na začátku tohoto století existovala tramvajová linka z~Zwolle do
Blokzijl přes Hasselt. Později nabývaly na významu jiné dopravní
prostředky a~tramvajová trať byla před druhou světovou válkou
zrušena. Nyní by taková tramvajová linka možná byla zase zisková.
\stopframedtext

```

37.3 Blok textu

Jiným stavebním kamenem dokumentu je *blok textu*. Příkladem bloku textu je jeden nebo více odstavců, které chceme použít několikrát.

Blok textu musíme definovat pomocí

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Na začátku tohoto století existovala tramvajová linka z Zwolle do Blokzijl přes Hasselt. Později nabývaly na významu jiné dopravní prostředky a tramvajová trať byla před druhou světovou válkou zrušena. Nyní by taková tramvajová linka možná byla zase získávána.

Vsuvka 37.1 Ukázka vsuvky.

```
\defineblock [...]
```

```
* IDENTIFIER
```

Mezi hranaté závorky vložíme jeho referenční jméno.

```
\defineblock[holandstina]
```

Také můžeme najednou definovat několik bloků textu, pokud mezi jednotlivá jména napíšeme čárku.

Po zadefinování máme k dispozici následující dvojici:

```
\beginholandstina ... \endholandstina
```

S blokem můžeme pracovat pomocí příkazů

```
\hideblocks [...1;...] [...2;...]
                        OPTIONAL
```

```
1 IDENTIFIER
```

```
2 IDENTIFIER
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\useblocks [...1;...] [...2;...]
                                OPTIONAL
```

```
1 IDENTIFIER
```

```
2 IDENTIFIER
```

```
\keepblocks [...1;...] [...2;...]
                                OPTIONAL
```

```
1 IDENTIFIER
```

```
2 all IDENTIFIER
```

```
\selectblocks [...1;...] [...2;...] [...3.]
                                OPTIONAL OPTIONAL
```

```
1 IDENTIFIER
```

```
2 IDENTIFIER
```

```
3 criterium = all SECTION
```

Zde je příklad jejich použití:

```
\defineblock[holandstina,cestina]
```

```
\hideblocks[holandstina,cestina]
```

```
\begincestina[blok-a]
```

Po roce 1810 přinesl kanál Dedemsvaart Hesseltu prosperitu.

Všechny lodě proplouvaly kanálem v-Hasseltu a obchody na obou stranách kanálu bohatly.

```
\endcestina
```

```
\beginholandstina[blok-a]
```

Sinds 1810 veroorzaakte de Dedemsvaart enige welvaart in Hasselt.

Alle schepen voeren door de grachten en de winkels aan weerszijden van de gracht floreerden.

```
\endholandstina
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\useblocks [cestina] [blok-a]
```

Výsledkem bude:

Po roce 1810 přinesl kanál Dedemsvaart Hesseltu prosperitu. Všechny lodě proplouvaly kanálem v Hasseltu a obchody na obou stranách kanálu bohatly.

Jestliže budeme pokračovat v takovémto používání bloku textu, můžeme vytvořit dvojjazyčný manuál. Pro tyto účely je také možno uložit bloky textu do externích souborů:

```
\setupblock [holandstina] [file=store-h]
```

Holandský blok textu je uložen do souboru `store-h.tex` a fragmenty textu mohou být vyvolány jejich referenčními jmény.

37.4 Uložení textu pro pozdější použití

Dočasně můžeme uložit informace pro budoucí použití v dokumentu pomocí

```
\startbuffer [...] ... \stopbuffer
      OPTIONAL
* IDENTIFIER
```

Například:

```
\startbuffer [navsteva]
```

Chcete-li vidět, co vám Hasselt může nabídnout, měli byste jej někdy navštívit. Jestliže si s sebou vezmete tento manuál, některá místa poznáte.

```
\stopbuffer
```

```
\getbuffer [navsteva]
```

Příkazem `\getbuffer [navsteva]` se vyvolá uložený text. Referenční jméno můžeme vynechat, `CONTEXT` pak pracuje s interním jménem. Přetisk uložené informace získáme pomocí `\typebuffer [navsteva]`.

Nastavení se provádí příkazem

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupbuffer [1.] [.,.,2.,...]
```

```
1 IDENTIFIER
2 paragraph = NUMBER
   before   = COMMAND
   after    = COMMAND
```

37.5 Skrytý text

Text může být skryt pomocí

```
\starthiding ... \stophiding
```

Vložený text nebude zpracováván.

37.6 Linky

Máme k dispozici mnoho příkazů na kreslení linek. Jednu linku vysázíme pomocí

```
\hairline
```

nebo

```
\thinrule
```

Více linek najednou nakreslíme pomocí

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\thinrules [.*.]
              OPTIONAL
* inherits from \setupthinrules
```

Možná je také kombinace textu a linek:

Hasselt – Amsterdam

Kdybychom nakreslili spojnici Hasseltu s Amsterdamem, prošli bychom vzdálenost téměř 145 km.

Kdybychom nakreslili dvě spojnice Hasseltu s Amsterdamem, prošli bychom vzdálenost téměř 290 km.

Amsterdam _____

_____ Hasselt

```
\starttextrule{Hasselt -- Amsterdam}
Kdybychom nakreslili spojnici Hasseltu s~Amsterdamem, prošli bychom
vzdálenost téměř 145 \Kilo \Meter.
\stoptextrule

Kdybychom nakreslili dvě spojnice Hasseltu s~Amsterdamem, prošli bychom
vzdálenost téměř 290 \Kilo \Meter.

Amsterdam \thinrules[n=3] Hasselt
```

Při kreslení linek musíme být vždy opatrní. Prázdné řádky okolo `\thinrules` nesmí být opomenuty a výsledné vertikální odsazení by mělo být vždy předmětem naší pozornosti.

Vertikální odsazení můžeme nastavit pomocí

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupthinrules [.*.]
```

```
* interlinespace = small medium big
  n              = NUMBER
  before         = COMMAND
  inbetween      = COMMAND
  after          = COMMAND
  color          = IDENTIFIER
  backgroundcolor = IDENTIFIER
  height         = DIMENSION max
  depth         = DIMENSION max
  alternative    = a c
  rulethickness = DIMENSION
  color          = IDENTIFIER
  background     = color none
  backgroundcolor = IDENTIFIER
```

Dále máme k dispozici několik doplňkových příkazů, které mohou být velice užitečné:

```
\setupfillinrules [...,*.,...]
```

```
* width          = fit broad DIMENSION
  distance        = DIMENSION
  before          = COMMAND
  after           = COMMAND
  style           = normal bold slanted boldslanted type cap small...
                  COMMAND
  n               = NUMBER
  interlinespace = small medium big
  separator       = TEXT
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupfillinlines [...,*.,...]
```

```
* width = DIMENSION
margin = DIMENSION
distance = DIMENSION
before = COMMAND
after = COMMAND
```

Uvedeme je v následujícím příkladu:

```
\setupfillinrules[width=2cm]
\setupfillinlines[width=3cm]
\fillinrules[n=1]{\bf jméno}
\fillinrules[n=3]{\bf adresa}
\fillinline{Uved'te prosím \underbar{počet} domů v~Hasseltu.} \par
V~tomto textu přeškrtněte \overstrikes{Hasselt}\periods[18]
```

Výsledkem bude:

jméno _____
adresa _____

Uved'te prosím počet domů v Hasseltu. _____

V tomto textu přeškrtněte ~~Hasselt~~.

Tyto příkazy bývají používány v dotaznících. Omezením je, že podtržený nebo přeškrtnutý text nebude na konci řádky rozdělen.

37.7 Horní a dolní index v textu

Vložit ^{horní index} a _{dolní index} do dokumentu je jednoduché. A co říkáte tomuto: ^{horní index?}_{dolní index} To vypadá podivně, co?!

Tento ošklivý text byl vysázen pomocí `\low{}`, `\high{}` a `\lohi{ }{ }`. Index je vkládán do složených závorek.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

37.8 Datum

Systémové datum vložíme do textu pomocí

```
\currentdate
```

37.9 Umístění do předepsané polohy

Občas máme potřebu umístit text na konkrétní místo uvnitř strany či jiného textového elementu. Uděláme to pomocí

```
\position (...1,...) {...2.}
```

```
1 POSITION
```

```
2 TEXT
```

Kulaté závorky určují souřadnice x, y , složené závorky ohraničují text, který má být umístěn.

Souřadnice nastavíme pomocí

```
\setuppositioning [...,*...,...]
```

```
* state = start overlay
  unit  = cm pt em mm ex es in
  factor = NUMBER
  scale = NUMBER
  offset = yes no
  xstep  = absolute relative
  ystep  = absolute relative
```

Zvolit můžeme jednotky a měřítko. Zde je ilustrační příklad:

```
\def\dicefive%
{\framed
 [width=42pt,height=42pt,offset=0pt]
 {\setuppositioning
 [unit=pt,factor=12,xoffset=-11pt,yoffset=-8pt]%
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

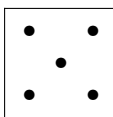

```

\startpositioning
\position(1,1){$\bullet$}%
\position(1,3){$\bullet$}%
\position(2,2){$\bullet$}%
\position(3,1){$\bullet$}%
\position(3,3){$\bullet$}%
\stoppositioning}}

```

```
\placefigure{Toto je pět.}{\dicefive}
```

Výsledkem tohoto relativně složitého příkladu je toto:



Obrázek 37.1 Toto je pět.

37.10 Otočení textu, obrázků a tabulek

V mnoha případech chceme otočit text nebo obrázek. Provedeme to pomocí

```

\rotate [..., 1, ...] {2.}
          OPTIONAL
1  inherits from \setuprotate
2  TEXT

```

Hranaté závorky jsou volitelné. Uvnitř nich specifikujeme úhel natočení: `rotation=90`. Složené závorky vymezují text či objekt, který chceme otočit. Příklad:

V roce 1252 Hasselt získal městská práva. Od té doby měl `\rotate[rotation=90]{právo}` používat vlastní pečeť na oficiálních dokumentech. Tato pečeť zobrazuje svatého Štěpána, jednoho z prvních křesťanským mučedníků. Byl `\rotate[rotation=270]{patronem}` Hasseltu. Po reformaci byla pečeť pozměněna, svatý Štěpán ztratil svou svatost a byl zobrazován bez svatozáře.

Výsledkem je velice ošklivý odstavec:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

V roce 1252 Hasselt získal městská práva. Od té doby měl ^{právo} používat vlastní pečeť na oficiálních dokumentech. Tato pečeť zobrazuje svatého Štěpána, jednoho z prvních křesťanských mučední-

^{patronem} ků. Byl ^{patronem} Hasseltu. Po reformaci byla pečeť pozměněna, svatý Štěpán ztratil svou svatost a byl zobrazován bez svatozáře.

Stejně jednoduše můžeme otočit i obrázek:

```
\placefigure
  [] [fig:rotation]
  {Rybářský přístav (de Vispoort) otočený o~180 \Degrees.}
  \rotate[rotation=180]{\externalfigure[ma-cb-15] [width=10cm]}
```

Na **obrázku 37.2** vidíte, že není vždy úplně zřejmé, na co se díváme, když obrázek otočíme.



Obrázek 37.2 Rybářský přístav (de Vispoort) otočený o 180°.

Otočení nastavíme pomocí:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setuprotate [...,*,...]
```

```
* rotation = NUMBER
   location = normal high fit broad depth
   inherits from \setupframed
```

37.11 Ukončení řádky

Nová řádka může být vynucena příkazem

```
\crlf
```

Pokud chceme, aby konce řádek kopírovaly ukončení ve vstupním souboru, napíšeme:

```
\startlines ... \stoptlines
```

Na radnici města Hasselt je dřevěná deska, kde se můžeme dočíst:

```
\startlines
Heimelijcken haet
eigen baet
jongen raet
Door diese drie wilt verstaen
is het Roomsche Rijck vergaen.
\stoptlines
```

Tato malá básnička obsahuje varování pro radní Hasseltu: {\it Nedovol, aby osobní prospěch nebo pocity ovlivňovaly tvůj rozum při rozhodování.}

Na radnici města Hasselt je dřevěná deska, kde se můžeme dočíst:

```
Heimelijcken haet
eigen baet
jongen raet
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Door diese drie wilt verstaen
is het Roomsche Rijck vergaen.

Tato malá básnička obsahuje varování pro radní Hasseltu: *Nedovol, aby osobní prospěch nebo pocity ovlivňovaly tvůj rozum při rozhodování.*

V několika příkazech se konce řádků označují \\. Například když napíšeme
`\inmargin{uvnitř}\marginálie}`, pak se text v marginálním sloupci zlomí do dvou řádků.

37.12 Dělení slov

Pokud připravujeme vícejazyčný text, musíme si být vědomi faktu, že dělení slov se jazyk od jazyku liší.

Hlavní jazyk aktivujeme pomocí

```
\mainlanguage [.*.]
```

```
* nl fr en uk de es cz ..
```

Mezi závorky můžeme vložit af (afrikánština), ca (katalánština), cn (čínština), cz (čeština), da (dánština), de (němčina), en (angličtina), es (španělština), fi (finština), fr (francouština), gr (řečtina), it (italština), hu (maďarština), hr (srbochorvačtina), ja (japonština), nl (holandština), no (norština), pl (polština), pt (portugalština), ro (rumunština), ru (ruština), sk (slovenština), sl (slovinština), sv (švédština), tr (turečtina), ua (ukrajinština), uk (britská angličtina), us (americká angličtina) a vn (vietnamština).

K dočasné změně jednoho jazyka na jiný použijeme kratší variantu:

```
\language [en]
```

nebo

```
\en
```

Například:

Pokud se chcete dozvědět více o Hasseltu, asi nejlepší knihou, kterou si můžete přečíst, je `{\nl \em Uit de geschiedenis van Hasselt}` od F.~Peerebooma.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Pokud se chcete dozvědět více o Hasseltu, asi nejlepší knihou, kterou si můžete přečíst, je *Uit de geschiedenis van Hasselt* od F. Peerebooma.

Pokud je slovo rozděleno nesprávně, můžeme si dodefinovat místo rozdělení sami. Uděláme to v úvodní části dokumentu takto:

```
\hyphenation{his-to-rie}
```

37.13 Komentář ve zdrojovém souboru

Veškerý text mezi `\starttext` a `\stoptext` bude `CONTEXT`em zpracován. Někdy se však vyskytne fragment textu, který zpracovávat nechceme, nebo chceme okomentovat příkazy `CONTEXT`u.

Pokud předradíme našemu textu znak procenta `%`, nebude zpracován.

```
% Ve velmi velkém dokumentu můžeme zdrojový text vložit
% z~externích souborů.
%
% Například:
%
% \input hass01.tex % kapitola 1 o Hasseltu
% \input hass02.tex % kapitola 2 o Hasseltu
% \input hass03.tex % kapitola 3 o Hasseltu
```

Pokud před každým příkazem `\input` smažeme znak `%`, soubory budou zpracovány. Komentář popisující obsah souborů zpracován nebude.

37.14 Vstup jiného zdrojového tex souboru

V mnoha situacích chceme do svého zdrojového souboru vložit jiný soubor `TEX`. Někdy je například efektivnější rozdělit text psaný v `CONTEXT`u do více souborů, aby mohly být zpracovány samostatně.

Soubor se jménem `jiny.tex` může být vložen pomocí:

```
\input jiny.tex
```

Přípona `tex` je nepovinná, takže bude fungovat i

```
\input jiny
```

Příkaz `\input` je primitivum `TEX`u.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

38 | Externí moduly

Kvůli efektivnosti se autor rozhodl realizovat některé funkce `CONTEXT`u prostřednictvím vnějších modulů. Můžeme například zavést následující moduly:¹¹

- `bib` sazba bibliografických referencí programem `BIBTEX`;
- `chemic` sazba chemických struktur;
- `units` použití jednotek `SI`;
- `pictex` kreslení obrázků (používá se společně s modulem `chemic`);
- `chart` náčrt vývojových diagramů a organizačních schémat;
- `steps` náčrt stavových diagramů;
- `narrowtt` použití užšího strojopisu `Computer Modern` pro verbatim prostředí;
- `amsl` implementace `AMSMATH` příkazů;
- `nath` přirozená sazba matematiky, obsahuje modul `amsl`;
- `streams` synchronizace několika vstupních textů (dvojjazyčná sazba);
- `gnuplot` přímý zápis `GNU PLOT` grafů;
- `lilypond` sazba notových osnov programem `LilyPond`.

Zavedení se provede v deklarační oblasti příkazem:

```
\usemodule [...]*[...]
```

```
* IDENTIFIER
```

Již jsme uvedli množství příkladů použití modulu `units`. Níže bez dalšího vysvětlení uvádíme dva příklady použití modulů `chemic` a `chart`. Tyto moduly jsou popsány ve dvou samostatných příručkách ([mp-ch-en.pdf](#), [mcharts.pdf](#)).

¹¹ Více viz seznam modulů na [Wiki Modules](#).

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

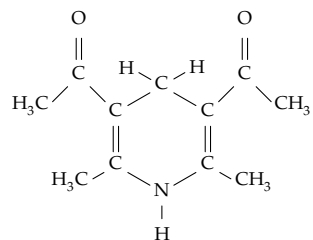
+

krok zpět

-

+

Chemické struktury mohou vypadat velice působivě:



Sloučenina A

Při kreslení chemických struktur tohoto typu se `CONTEXT` spoléhá na `METAPOST`. Třebaže jsou tyto chemické struktury definovány jenom pomocí dvou nebo tří příkazů, na získání správných výsledků je potřebná jistá praxe. Vstup může vypadat takto:

```
\placeformula[-]
\startformula
\startchemical[scale=small,width=fit,top=3000,bottom=3000]
  \chemical[SIX,SB2356,DB14,Z2346,SR3,RZ3,-SR6,+SR6,-RZ6,+RZ6]
    [C,N,C,C,H,H,H]
  \chemical[PB:Z1,ONE,Z0,DIR8,Z0,SB24,DB7,Z27,PE][C,C,CH_3,0]
  \chemical[PB:Z5,ONE,Z0,DIR6,Z0,SB24,DB7,Z47,PE][C,C,H_3C,0]
  \chemical[SR24,RZ24][CH_3,H_3C]
  \bottext{Sloučenina A}
\stopchemical
\stopformula
```

Modul `chart` si představíme ukázkou definice organizačního schématu. Může vypadat následovně:

```
\startFLOWchart[organogram]
  \startFLOWcell
    \shape {action}
    \name {01}
    \location {2,1}
    \text {Hasselt}
    \connect [bt]{02}
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```

\connect [bt]{03}
\connect [bt]{04}
\stopFLOWcell
\startFLOWcell
\shape {action}
\name {02}
\location {1,2}
\text {Mastenbroek}
\stopFLOWcell
\startFLOWcell
\shape {action}
\name {03}
\location {2,2}
\text {Genne}
\stopFLOWcell
\startFLOWcell
\shape {action}
\name {04}
\location {3,2}
\text {Zwartewaterklooster}
\stopFLOWcell
\stopFLOWchart
\FLOWchart [organogram]

```

Výsledkem bude:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

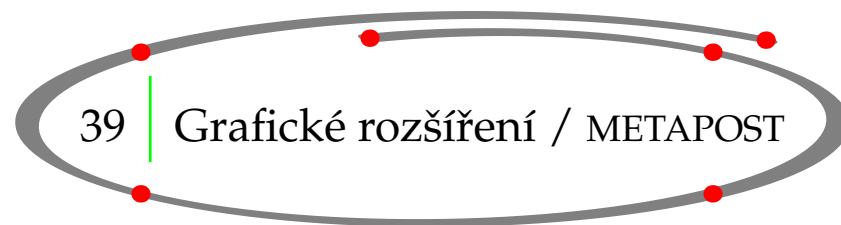
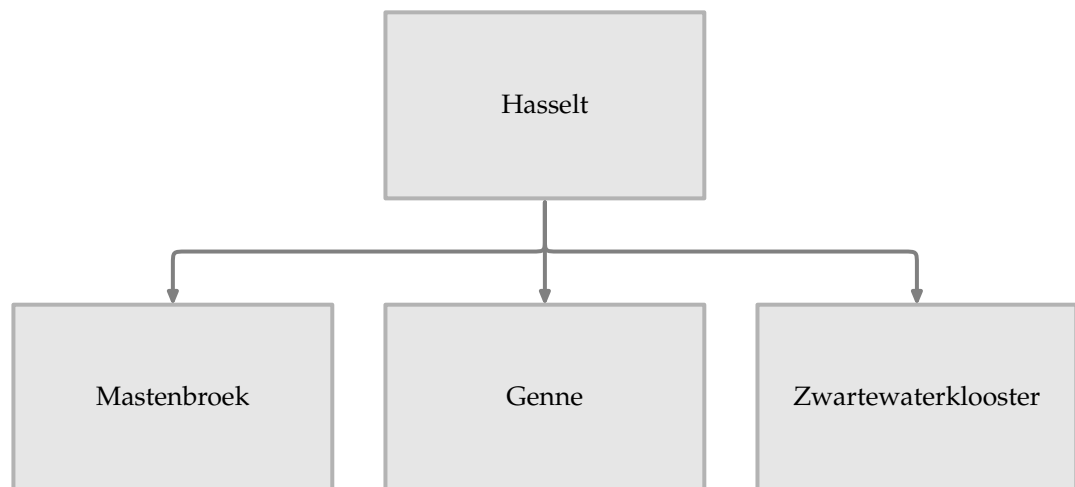
vyhledej

+

krok zpět

-

+



Grafické možnosti makrojazyka \TeX jsou dosti omezené. Použijeme-li však grafický systém METAPOST od Johna Hobbyho, získáme úplnou množinu grafických prvků, které mohou zlepšit vzhled našeho dokumentu.

\CONTEXT přímo spolupracuje s METAPOSTem, takže uživatel může využít všechny vlastnosti METAPOSTu přímo ve svém dokumentu. Nadpisy kapitol a čísla stránek tohoto manuálu jsou zvýrazněny grafikou přímo generovanou METAPOSTem.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Použití METAPOSTu v CONTEXTu je popsáno v rozsáhlém manuálu `metafun-s.pdf`.



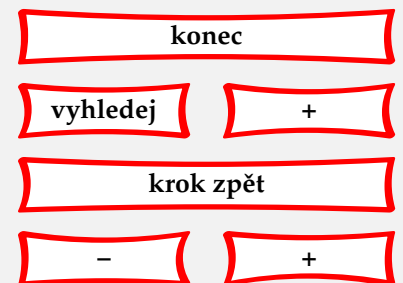
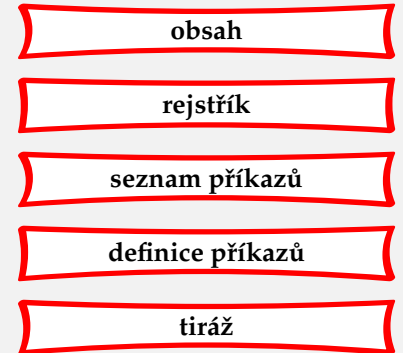
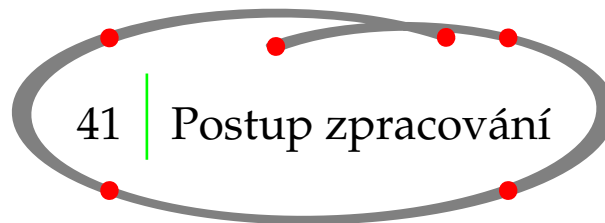
Během provozu CONTEXTu se nastavuje množství implicitních parametrů. Tyto parametry jsou zadefinovány v souboru `cont-sys.tex`. Uživatelé mohou tento soubor vytvořit přejmenováním souboru `cont-sys.rme` (nejlépe ve stejném adresáři, aby jej CONTEXT našel) a v tomto souboru definovat své vlastní preference (firemní styl). Nejdůležitější část tohoto souboru se pravděpodobně týká výstupu:

```
\setupoutput [pdf tex]
```

Sdělí CONTEXTu, že má vytvářet výstup PDF namísto DVI, zatímco

```
\setupoutput [dvipsone, dviwindo]
```

nastaví podmínky pro tyto programy. Předvolený je DVIPS výstup.



Během zpracování zapisuje `CONTEXT` informace do souboru `muj_soubor.tui`. Tyto informace se využívají v následujícím průchodu. Část těchto informací je zpracována programem `TEXUTIL`. Informace o registrech a seznamech se jím převede do souboru `muj_soubor.tuo`. `CONTEXT` filtruje informace z tohoto souboru a (v případě potřeby) je využívá.

Program `TEXUTIL` umí další užitečné činnosti. Obecný zápis pro použití tohoto programu je:¹²

```
texmfstart texutil --specifikace jmeno_souboru
```

Velikosti všech vektorových i bitmapových obrázků (boundingbox) v zadaném adresáři vypíšeme do souboru `texutil.tuf` zadáním:

```
texmfstart texutil --figures *.*
```

Pokud potřebujeme konvertovat EPS-ilustrace do formátu PDF, můžeme zadat:

```
texmfstart texutil --figures --epspage --epspdf
```

Dočasné soubory (`.tui`, `.tuo`, `*-mpgraph.*`, `ap.`) smažeme

```
texmfstart texutil --purge
```

nebo novější

```
texmfstart ctxtools --purgefiles
```

a všechny generované soubory v aktuálním adresáři smažeme pomocí

```
texmfstart texutil --purgeall
```

nebo

```
texmfstart ctxtools --purgefiles --all
```

Informace o přetečených nebo podtečených boxech a neznámých referencích odfiltrujeme ze záznamového souboru o běhu `CONTEXTu` (`.log`) příkazem

```
texmfstart texutil --logfile soubor.log
```

Na spuštění `CONTEXTu` používáme povel `TEXEXEC`:

```
texmfstart texexec jmeno_souboru
```

Tento program se postará o to, aby se `CONTEXT` spustil tolikrát, kolikrát je potřeba pro získání správných odkazů. Zvláštní požadavky můžeme zadat také přímo v příkazové řádce. Například, když je požadován výstup ve formátu pdf, napíšeme:



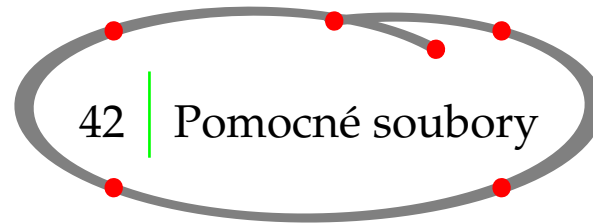
¹² Některé funkce v době překladu tohoto manuálu mohl převzít program `ctxtools`.

obsah
rejstřík
seznam příkazů
definice příkazů
tiráž

konec	
vyhledej	+
krok zpět	
-	+

```
texmfstart texexec --pdf jmeno_souboru
```

Když jsme na pochybách, můžeme se na parametry programů dotázat přepínačem `--help`. K dispozici je také dokumentace **mtexexec.pdf** a **mtexutil.pdf**.



V průběhu zpracování vytváří `CONTEXT` množství pomocných souborů. Pokud název vstupního souboru je `muj_soubor.tex`, v pracovním adresáři se objeví následující soubory:

Soubor	Obsah	Stav
<code>muj_soubor.tex</code>	vstupní text	neodstraňujeme
<code>muj_soubor.tui</code>	vstupní informace	může být odstraněn
<code>muj_soubor.tuo</code>	výstupní informace	neodstraňujeme
<code>muj_soubor.tub</code>	informace o blocích	neodstraňujeme
<code>muj_soubor.tmp</code>	dočasné informace	může být odstraněn
<code>texutil.tuf</code>	informace o obrázcích	vytvoříme v případě potřeby
<code>mpgraph.mp</code>	informace <code>METAPOSTu</code>	může být odstraněn
<code>muj_soubor.dvi</code>	vysázený text	může být odstraněn
<code>muj_soubor.ps</code>	text k tisku	může být odstraněn
<code>muj_soubor.pdf</code>	text k tisku/prohlížení	může být odstraněn

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

43 | Sazba českých/slovenských textů

43.1 Minimální český/slovenský dokument

Pro sazbu českých nebo slovenských textů nám stačí základní formát `cont-en`¹³. Ten již obsahuje vzory dělení pro češtinu/slovenštinu v náležitých kódováních, o čemž se lze přesvědčit příkazem `\showpatterns`, viz **tabulku 31.1**.

Formát `cont-en` se všemi vzory dělení vygenerujeme příkazem

```
texmfstart texexec --make --all
```

V samotném dokumentu musíme nastavit češtinu nebo slovenštinu jako hlavní jazyk: `\mainlanguage[cz]` / `\mainlanguage[sk]` a zvolit námi používané vstupní kódování. Minimální český dokument `min-cz.tex` v kódování ISO-8859-2 pak bude vypadat takto:

```
\mainlanguage[cz]
\enableregime[latin2]

\starttext
  Nějaký český text, třeba o těch příliš žluťoučkých koních,
  co pěli ďábelské ódy.
\stoptext
```

Pro minimální slovenský dokument pouze nahradíme `cz` → `sk`. Soubor přeložíme do PDF (i s jeho automatickým zobrazením) příkazem

```
texmfstart texexec --pdf --autopdf min-cz
```

¹³ Alternativou je lokalizovaný formát `cont-cz` s počeštěnými názvy uživatelských příkazů. Tuto lokalizaci ConTeXtu vytvořil Tomáš Hudec v rámci své diplomové práce v roce 2001 pod vedením Petra Sojky. Protože je však dosud česká CONTEXTOVÁ komunita malá, česká dokumentace téměř žádná a terminologie příkazů neustálená a neúplná, používá se hlavně z důvodů kompatibility téměř výhradně anglické rozhraní `cont-en`. Toto anglické rozhraní popisuje i tento manuál.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

43.2 Písma

CONTEXT implicitně používá písmo Latin Modern v kódování ec.¹⁴ Toto kódování obsahuje všechny znaky potřebné pro sazbu českých i slovenských textů. Výchozí nastavení písma je provedeno příkazy

```
\usetypescript [modern] [ec]
\setupbodyfont [modern, 10pt, rm]
```

Pokud chceme tisknout kombinací

Times New Roman – Helvetica – Courier, nastavíme

```
\usetypescript [postscript] [ec]
\setupbodyfont [postscript, 10pt, rm]
Times New Roman -- \ss Helvetica -- \tt Courier%
```

Množství dalších písem je definováno v souborech type-enc.tex a type-buy.tex.

Pro některá písma ze Štormovy písmolijny je též vytvořena podpora, viz adresu <http://modules.contextgarden.net/stormfontsupport> s katalogem **cont-storm-catalog.pdf**. Pomocí ní je pak například užití písma Lido jednoduché:

```
\enableregime [latin2]
\useencoding [st2] % or st1 for ec encoding
\useencoding [st3]

\usetypescriptfile [t-type-slido]
\usetypescript [Lido] [st2] % or st1 for ec encoding
\setupbodyfont [Lido, 10pt, rm]
```

43.3 Kódování

Náš vstupní text vytváříme v určitém kódování, nejčastěji v ISO-8859-2 (Latin 2), CP1250 (Windows), či UTF-8. Protože tato tři kódování se většinou liší od kódování fontu (a vzorů dělení slov), je třeba CONTEXTu toto vstupní kódování sdělit. Potřebnou konverzi pak zařídí za nás.

Vstupní kódování se v CONTEXTu nazývá *režim* (regime). Jeho nastavení se provede pomocí `\enableregime`, tedy pro výše zmíněné kódování `\enableregime [Latin2]`, `\enableregime [cp1250]` nebo `\enableregime [utf8]`.

¹⁴ Pokud bychom potřebovali CSfonty, můžeme využít zakomentovaný kód souboru type-old.tex.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

43.4 Uvozovky, pomlčky a spol

Text „v uvozovkách“ a ‚jednoduchých uvozovkách‘ se napíše pomocí `\quotation{...}` a `\quote{...}`. Pomlčka – ta se napíše pomocí `|<|` a spojovník je-li na vstupu `|-|`. `CONTEXT` zařídí případné opakování na konci řádky. Francouzská mezera za tečkou `\frenchspacing` je v `ConTeXtu` implicitní.

43.5 Datum

Máme-li hlavní jazyk `cz`, pak příkazem `\currentdate` vysázíme 6. prosince 2006. Pokud lokálně přepneme do jiného jazyka

```
\start
  \language[en]%
  \currentdate
  \ nebo
  \language[sk]%
  \currentdate
\stop
```

dostaneme December 6, 2006 nebo 6. decembra 2006. Z číselné informace získáme název měsíce `\month{2}` (února) a název dne počínaje nedělí `\weekday{5}` (čtvrtek).

43.6 Popisky a názvy

Nastavením hlavního jazyka dostaneme patričné názvy pro *obrázky*, *tabulky*, *přílohy* ap., viz soubor `lang-sla.tex`. Změnit název lze pomocí `\setupheadtext`, např.

```
\setupheadtext [cz] [content=Obsah průvodce po Hasseltu]
```

43.7 Řazení

`CONTEXT` obsahuje interní řadící a částečně konfigurovatelný algoritmus (soubor `sort-lan.tex`)¹⁵. Vychází se z normy ČSN 97 6030 z roku 1994, ale zatím nejde o její plnou implementaci; hlavní rozdílnost je v pořadí číslic, znaku apostrofu a spojovníku. Řazení využívají rejstříky (`index`) a řazené seznamy (`sorting`).

¹⁵ Slovenská sekce pravidel v tomto souboru zatím není napsaná.

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

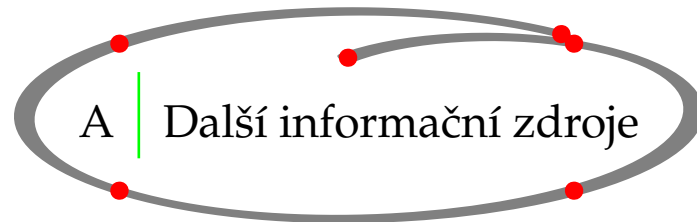
vyhledej

+

krok zpět

-

+



Už vám tento začátečnický manuál nedostačuje? Potřebujete poradit? Zajímá vás mnoho dalších možností CONTeXTu? Tato kapitola může pomoci v orientaci po dostupných informačních zdrojích. Bohužel – v češtině existuje zatím pouze tento manuál. Všechna ostatní zde doporučovaná literatura je v angličtině.

A.1 CONTeXTová wikipedie

CONTeXT vznikl v Holandsku a tak nepřekvapí, že jeho symbolem jsou tulipány. Internetové stránky **ConTeXt garden** jsou zahradou, kde uživatelé doslova pěstují CONTeXT: sdílí své poznatky na principu wikipedie. Najdeme zde návody a řešení, referenční manuál (**Show-web**), odkazy na užitečné informace, errata manuálů, „živý“ CONTeXT pro testování bez nutnosti domácí instalace, archiv velice živé konference – a to vše plně prohledávatelné.

A.2 Manuály

Uvádíme zde výběr nejzajímavějších manuálů z pera Hanse Hageny. Každý je originální ukázkou možností CONTeXTu. Úplný seznam čtenář najde na stránkách **Pragma ADE** nebo přímo v adresáři <http://www.pragma-ade.com/dir/general>.

CONTeXT the manual (cont-eni.pdf)

Základní manuál CONTeXTu. Všeobecný a hluboký zdroj informací, byť poměrně staršího data (2001).

METAFUN (metafun-s.pdf)

Zevrubný manuál o grafických možnostech CONTeXTu založený na integraci METAPOSTu. (2002)

Fonts in CONTeXT (mfonts.pdf)

Instalace nových fontů. (2001)

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Widgets uncovered (mwidget.pdf)

Použití interaktivních prvků v PDF: hyperlinky, videa, formuláře, Java script. (2001)

XML in CONTEXT (xmlcontext.pdf)

Přímá sazba z XML zdrojových textů. (2001)

BIBTEXový modul (bibmod-doc.pdf)

Popis použití modulu pro BIBTEX.

Columns (columns.pdf)

Vícesloupcová sazba včetně plavajících objektů, sazba do řádkového rejstříku, sazba obrázků přes několik stránek. (2004)

Natural Tables in CONTEXT (enattab.pdf)

Alternativní sazba tabulek s podobnou syntaxí jako v HTML. Vhodné pro sazbu tabulek z XML, s barevným pozadím, nepravidelnou velikostí buněk a dlouhých tabulek přes několik stránek. (2004)

Charts uncovered (mcharts.pdf)

Vývojové diagramy v pravidelné mřížce buněk. (2001)

PPCHT_{EX}, a macropackage for typesetting chemical structure formulas

(mp-ch-en.pdf) Sazba chemických struktur a diagramů. (2001)

Adding Text to Graphics (mlabels.pdf)

Přidání popisků do obrázků. (2002)

It's in the Details (details.pdf)

Popis různých triků: dodržení řádkového rejstříku, pseudosloupce, barevné pozadí či rámy i přes několik stránek, usazení matematických rovnic, umístění plavajících objektů, popisky obrázků. (2004)

Typographic Programming (style.pdf)

Rozpracovaný manuál o typografii v CONTEXTu. (2005)

A.3 Magazíny

Magazínem se v této komunitě myslí ukázkový dokument s jednotnou úpravou zaměřených na nějaké téma. Existují dva druhy. První ‚This way‘ je psán samotným tvůrcem CONTEXTu Hansem Hagemem, druhý ‚My way‘ je psán jednotlivými uživateli. Jako ukázky uveďme:

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Good looking shapes (mag-0010.pdf)

Sazba do definovaného tvaru odstavce.

Using \startalign and friends

(Aditya Mahajan, **mathalign.pdf**) Sazba matematických rovnic a jejich zarovnávání.

Natural Tables

(Willi Egger, **NaturalTables.pdf**) Ukázka použití ‚přirozených tabulek‘.

A.4 Instalace

Čerstvý `CONTEX`T čtenář nalezne na stránkách firmy Hanse Hagen **Pragma ADE** (Advanced Document Engineering). Většinou se vyplatí instalovat nejnovější beta verzi. K dispozici jsou samostatné instalace pro různé operační systémy nebo balíky do stávající distribuce `TEX`. V takovém případě instalace spočívá v rozbalení `cont-tmf.zip` do patřičného `TEX` stromu a v přegenerování formátu (viz **kapitolu 43**).

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

B | Definice příkazů

Následuje souhrn příkazů, jež jsme uvedli v předchozích kapitolách. Je to jen výběr z celého repertoáru příkazů `CONTEXT`. Kdo je chce uvidět všechny, může nahlédnout do rozsáhlejší příručky *CONTEXT the manual* (**cont-eni.pdf**) nebo do online referenčního manuálu (**Show-web**), poskytujících úplný přehled uživatelských příkazů `CONTEXT`. Vysvětlení použité syntaxe je podáno v kapitole 35.

```
\at {.1.} {.2.} [.3.]
1 TEXT
2 TEXT
3 REFERENCE
```

```
\blank [...,*...]
      OPTIONAL
* small medium big nowhite back white disable
  force reset line halfline FORMULA fixed flexible
  none
```

```
\cap {...}
* TEXT
```

```
\chapter [...1...] {.2.}
      OPTIONAL
1 REFERENCE
2 TEXT
```

```
\crlf
```

```
\define [.1.] \2. {.3.}
      OPTIONAL
1 NUMBER
2 NAME
3 TEXT
```

```
\defineblock [.*]
* IDENTIFIER
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\definebodyfont [...1,...] [2.]
[...3,,...]
```

```
1 5pt ... 12pt default
2 rm ss tt mm hw cg
3
tf = FILE
bf = FILE
sl = FILE
it = FILE
bs = FILE
bi = FILE
sc = FILE
ex = FILE
mi = FILE
sy = FILE
ma = FILE
mb = FILE
mc = FILE
```

```
\definecolor [1.] [...,2,,...]
```

```
1 IDENTIFIER
2 r = TEXT
g = TEXT
b = TEXT
c = TEXT
m = TEXT
y = TEXT
k = TEXT
s = TEXT
h = TEXT
t = TEXT
a = TEXT
p = TEXT
e = TEXT
```

```
\definecombinedlist [1.] [...2,...]
[...3,,...]
OPTIONAL
```

```
1 IDENTIFIER
2 LIST
3 inherits from \setupcombinedlist
```

```
\definedescription [1.] [...,2,,...]
OPTIONAL
```

```
1 IDENTIFIER
2 inherits from \setupdescriptions
```

```
\defineenumeration [...1,...] [2.]
[...3,,...] OPTIONAL
```

```
1 IDENTIFIER
2 IDENTIFIER
3 inherits from \setupenumerations
```

```
\definefloat [1.] [2.]
```

```
1 SINGULAR NAME
2 PLURAL NAME
```

```
\definehead [1.] [2.]
```

```
1 IDENTIFIER
2 SECTION
```

```
\defineinteractionmenu [1.] [...2,...]
[...3,,...]
OPTIONAL
```

```
1 IDENTIFIER
2 left right top bottom IDENTIFIER
3
before = COMMAND
after = COMMAND
inbetween = COMMAND
left = COMMAND
right = COMMAND
middle = COMMAND
state = start stop none local
style = normal bold slanted
boldslanted type cap small...
COMMAND
color = IDENTIFIER
distance = overlay DIMENSION
samepage = yes empty no none
unknownreference = yes empty no none
leftoffset = DIMENSION
rightoffset = DIMENSION
topoffset = DIMENSION
bottomoffset = DIMENSION
position = yes no
inherits from \setupframed
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\definemakeup [1.] [...2, ...]`

- 1 IDENTIFIER
- 2 inherits from \setupmakeup

`\defineregister [1.] [2.]`

- 1 SINGULAR NAME
- 2 PLURAL NAME

`\definestartstop [1.] [...2, ...]`

- 1 IDENTIFIER
- 2 before = COMMAND
- after = COMMAND
- style = normal bold slanted boldslanted type
- cap small... COMMAND
- commands = COMMAND

`\definesynonyms [1.] [2.] [3.] [4.]`
OPTIONAL

- 1 SINGULAR NAME
- 2 PLURAL NAME
- 3 COMMAND
- 4 COMMAND

`\definetabulate [1.] [2.] [3.]`
OPTIONAL OPTIONAL

- 1 IDENTIFIER
- 2 IDENTIFIER
- 3 TEXT

`\enableregime [...]`

- * Latin1 Latin2 Latin7 Latin15 iso-8859-1 iso-8859-2 iso-8859-7 iso-8859-15 cp1250 cp1251 cp1252 cp1253 cp1254 cp1257 viscii cyr grk mac utf8 unicode

`\externalfigure [1.] [...2, ...]`

- 1 FILE
- 2 inherits from \useexternalfigure

`\footnote [1.] {2.}`
OPTIONAL

- 1 REFERENCE
- 2 TEXT

`\framed [...1, ...] {2.}`
OPTIONAL

- 1 inherits from \setupframed
- 2 TEXT

`\from [...]`

- * REFERENCE

`\godown [...]`

- * DIMENSION

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\goto {1.} {2.} [...3,...]`

- 1 TEXT
- 2 TEXT
- 3 REFERENCE

`\hairline`

`\hideblocks [...1,...] [...2,...]`
OPTIONAL

- 1 IDENTIFIER
- 2 IDENTIFIER

`\in {1.} {2.} [3.]`

- 1 TEXT
- 2 TEXT
- 3 REFERENCE

`\indenting [...,*...]`

- * never none not no yes always first next small
medium big normal odd even DIMENSION

`\inmargin [1.] [2.] {3.}`
OPTIONAL OPTIONAL

- 1 + - low
- 2 REFERENCE
- 3 TEXT

`\keepblocks [...1,...] [...2,...]`
OPTIONAL

- 1 IDENTIFIER
- 2 all IDENTIFIER

`\mainlanguage [.*.]`

- * nl fr en uk de es cz ..

`\note [.*.]`

- * REFERENCE

`\page [...,*...]`

- * yes makeup no preference bigpreference left
right disable last quadruple even odd blank
empty reset start stop

`\pagereference [.*.]`

- * REFERENCE

`\placefloat [1.] [...2,...] {3.} {4.}`
OPTIONAL OPTIONAL

- 1 left right here top bottom inleft inright
inmargin margin leftmargin rightmargin leftedge
rightedge innermargin outermargin inneredge
outeredge inner outer line high low fit page
leftpage rightpage opposite always auto force
tall reset line height depth split
- 2 REFERENCE
- 3 TEXT
- 4 TEXT

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\placefootnotes [...,*,...]`
OPTIONAL

* inherits from \setupfootnotes

`\placeformula [...1,...] {2.} $$$3..$$`
OPTIONAL OPTIONAL

- 1 REFERENCE
- 2 TEXT
- 3

`\placelist [...1,...] [...2,...]`
OPTIONAL

- 1 IDENTIFIER
- 2 inherits from \setuplist

`\placelocalfootnotes [...,*,...]`
OPTIONAL

* inherits from \setupfootnotes

`\position (...1,...) {2.}`

- 1 POSITION
- 2 TEXT

`\rotate [...1,...] {2.}`
OPTIONAL

- 1 inherits from \setuprotate
- 2 TEXT

`\section [...1,...] {2.}`
OPTIONAL

- 1 REFERENCE
- 2 TEXT

`\selectblocks [...1,...] [...2,...] [...3.]`
OPTIONAL OPTIONAL

- 1 IDENTIFIER
- 2 IDENTIFIER
- 3 criterium = all SECTION

`\setupalign [...]`

* width left right middle inner outer wide broad
 height bottom line reset hanging nothanging
 hyphenated nothyphenated

`\setupbackgrounds [1.] [...2,...]`
OPTIONAL OPTIONAL
[...³,...]

- 1 top header TEXT footer bottom page paper
 leftpage rightpage
- 2 leftedge leftmargin TEXT rightmargin rightedge
- 3 state = start stop cd:repeat
 inherits from \setupframed

`\setupblank [...]`
OPTIONAL

* normal standard line DIMENSION big medium small
 fixed flexible

`\setupbodyfont [...,*,...]`

* IDENTIFIER serif regular roman sans support
 sansserif mono type teletype handwritten
 calligraphic 5pt ... 12pt

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\setupbuffer` [¹.] [².,.,.]
OPTIONAL

1 IDENTIFIER
2 paragraph = NUMBER
before = COMMAND
after = COMMAND

`\setupcaption` [¹.] [².,.,.]

1 IDENTIFIER
2 inherits from \setupcaptions

`\setupcaptions` [.,.,*.,.]

* location = top bottom none high low middle
width = fit broad max DIMENSION
minwidth = fit DIMENSION
headstyle = normal bold slanted boldslanted
type cap small... COMMAND
style = normal bold slanted boldslanted
type cap small... COMMAND
number = yes no
inbetween = COMMAND
align = inner outer left right middle
normal no yes
conversion = numbers characters Characters
romannumerals Romannumerals
way = bytext bycd:section
separator = TEXT
stopper = TEXT
command = COMMAND
distance = DIMENSION

`\setupcolors` [.,.,*.,.]

* state = start stop global local
conversion = yes no always
reduction = yes no
rgb = yes no
cmyk = yes no
mpcmyk = yes no
mpspot = yes no
textcolor = IDENTIFIER
split = c m y k p s no IDENTIFIER
criterion = all none

`\setupcolumns` [.,.,*.,.]

* n = NUMBER
ntop = NUMBER
rule = on off
height = DIMENSION
tolerance = verystRICT strict tolerant
verytolerant stretch
distance = DIMENSION
balance = yes no
align = inner outer left right middle
normal no yes
blank = fixed halfline line flexible big
medium small
option = background
direction = left right
inherits from \setupframed

`\setupcombinedlist` [¹.] [².,.,.]

1 IDENTIFIER
2 level = 1 2 3 4 SECTION current
inherits from \setuplist

`\setupdescriptions` [¹.,.] [².,.,.]
OPTIONAL

1 IDENTIFIER
2 headstyle = normal bold slanted boldslanted
type cap small... COMMAND
style = normal bold slanted boldslanted
type cap small... COMMAND
color = IDENTIFIER
width = fit broad DIMENSION
distance = DIMENSION
sample = TEXT
text = TEXT
align = inner outer left right middle
normal no yes
margin = standard yes no DIMENSION
location = left right top serried inmargin
inleft inright hanging
headcommand = COMMAND
hang = fit broad NUMBER
before = COMMAND
inbetween = COMMAND
after = COMMAND
indentnext = yes no
indenting = never none not no yes always
first next small medium big
normal odd even DIMENSION

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+


```
\setupenumerations [...,1,...] [...,2,...]
                                OPTIONAL
```

- 1 IDENTIFIER
- 2 inherits from \definedescription

```
\setupfillinlines [...,*,...]
```

- * width = DIMENSION
- margin = DIMENSION
- distance = DIMENSION
- before = COMMAND
- after = COMMAND

```
\setupfillinrules [...,*,...]
```

- * width = fit broad DIMENSION
- distance = DIMENSION
- before = COMMAND
- after = COMMAND
- style = normal bold slanted boldslanted
- type cap small... COMMAND
- n = NUMBER
- interlinespace = small medium big
- separator = TEXT

```
\setupfloat [.1.] [...,2,...]
```

- 1 IDENTIFIER
- 2 height = DIMENSION
- width = DIMENSION
- maxheight = DIMENSION
- maxwidth = DIMENSION
- minwidth = DIMENSION
- default = IDENTIFIER
- pageboundaries = LIST
- leftmargindistance = DIMENSION
- rightmargindistance = DIMENSION
- location = left middle right
- inherits from \setupframed

```
\setupfloats [...,*,...]
```

- * location = left right middle
- width = fit DIMENSION
- before = COMMAND
- after = COMMAND
- margin = DIMENSION
- spacebefore = small medium big none
- spaceafter = small medium big none
- sidespacebefore = small medium big none
- sidespaceafter = small medium big none
- indentnext = yes no
- ntop = NUMBER
- nbottom = NUMBER
- nlines = NUMBER
- default = IDENTIFIER
- tolerance = 0 1 2
- leftmargindistance = DIMENSION
- rightmargindistance = DIMENSION
- sidealign = normal line
- numbering = yes nocheck
- inherits from \setupframed

```
\setupfooter [.1.] [...,2,...]
                                OPTIONAL
```

- 1 inherits from \setupheader
- 2 inherits from \setupheader

```
\setupfootertexts [.1.] [.2.] [.3.]
                                OPTIONAL OPTIONAL OPTIONAL
```

- 1 text margin edge
- 2 TEXT SECTION date MARK pagenumber
- 3 TEXT SECTION date MARK pagenumber

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\setupfootnotes [..,.*,..]`

```
* conversion = numbers characters Characters
              romannumerals Romannumerals
way           = bytext bycd:section
location      = page TEXT columns firstcolumn
              lastcolumn high none
rule         = on off
before       = COMMAND
after        = COMMAND
width        = DIMENSION
height       = DIMENSION
bodyfont     = 5pt ... 12pt small big
style        = normal bold slanted boldslanted
              type cap small... COMMAND
distance     = DIMENSION
columnwidth  = DIMENSION
marginwidth  = DIMENSION
n            = NUMBER
numbercommand = \..#1
textcommand  = \..#1
split        = tolerant strict verystRICT
              NUMBER
textstyle    = normal bold slanted boldslanted
              type cap small... COMMAND
textcolor    = IDENTIFIER
interaction  = yes no
factor       = NUMBER
inherits from \framed
```

`\setupformulae [..,.*,..]`

```
* location    = left right
left         = TEXT
right        = TEXT
align        = inner outer left right middle
              normal no yes
option       = middle
strut        = yes no
distance     = DIMENSION
margin       = DIMENSION standard yes no
leftmargin   = DIMENSION
rightmargin  = DIMENSION
indentnext   = yes no
alternative  = IDENTIFIER
spacebefore  = DIMENSION
after        = DIMENSION
separator    = TEXT
conversion   = numbers characters Characters
              romannumerals Romannumerals TEXT
```

`\setupframed [..,.*,..]`

```
* height      = fit broad DIMENSION
width         = fit broad fixed local
              DIMENSION
autowidth     = yes no force
offset        = none overlay default DIMENSION
location      = depth hanging high lohi low
              top middle bottom keep
option        = none empty
strut         = yes no global local
align         = inner outer left right
              middle normal no yes
bottom        = COMMAND
top           = COMMAND
frame         = on off none overlay
topframe      = on off
bottomframe   = on off
leftframe     = on off
rightframe    = on off
frameoffset   = DIMENSION
framedepth   = DIMENSION
framecorner   = round rectangular
framradius    = DIMENSION
framecolor    = IDENTIFIER
rulethickness = DIMENSION
background    = screen color none foreground
              IDENTIFIER
backgroundscreen = NUMBER
backgroundcolor = IDENTIFIER
backgroundoffset = frame DIMENSION
backgrounddepth = DIMENSION
backgroundcorner = round rectangular
backgroundradius = DIMENSION
depth         = DIMENSION
corner        = round rectangular
radius        = DIMENSION
empty         = yes no
foregroundcolor = IDENTIFIER
foregroundstyle = normal bold slanted
              boldslanted type cap small...
              COMMAND
```

`\setupframedtexts [..,.*,..]`

```
* bodyfont    = 5pt ... 12pt small big
style         = normal bold slanted
              boldslanted type cap small...
              COMMAND
left          = COMMAND
right         = COMMAND
before        = COMMAND
after         = COMMAND
inner         = COMMAND
linecorrection = on off
depthcorrection = on off
margin        = standard yes no
location      = left right middle none
indenting     = never none not no yes always
              first next small medium big
              normal odd even DIMENSION
inherits from \setupframed
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\setuphead [...1,...] [...2,...]`

```

1 SECTION
2 style = normal bold slanted
        boldslanted type cap small...
        COMMAND
textstyle = normal bold slanted
        boldslanted type cap small...
        COMMAND
numberstyle = normal bold slanted
        boldslanted type cap small...
        COMMAND
color = IDENTIFIER
textcolor = IDENTIFIER
numbercolor = IDENTIFIER
number = yes no
ownnumber = yes no
page = left right yes
continue = yes no
header = none empty high nomarking
text = none empty high nomarking
footer = none empty high nomarking
before = COMMAND
inbetween = COMMAND
after = COMMAND
alternative = normal inmargin middle TEXT
hang = none broad fit line NUMBER
command = \..#1#2
numbercommand = \..#1
textcommand = \..#1
deepnumbercommand = \..#1
deeptextcommand = \..#1
prefix = + - TEXT
placehead = yes no empty
incrementnumber = yes no LIST FILE
resetnumber = yes no
file = IDENTIFIER
expansion = yes no command
margintext = yes no
inherits from \setupheads
    
```

`\setupheada`

`\setupheader [...1,...2,...]`
OPTIONAL

```

1 TEXT margin edge
2 state = normal stop start empty high none
        nomarking IDENTIFIER
strut = yes no
style = normal bold slanted boldslanted
        type cap small... COMMAND
leftstyle = normal bold slanted boldslanted
        type cap small... COMMAND
rightstyle = normal bold slanted boldslanted
        type cap small... COMMAND
leftwidth = DIMENSION
rightwidth = DIMENSION
before = COMMAND
after = COMMAND
    
```

`\setupheadertexts [...1] [...2] [...3]`
OPTIONAL OPTIONAL OPTIONAL

```

1 text margin edge
2 TEXT SECTION date MARK pagenumber
3 TEXT SECTION date MARK pagenumber
    
```

`\setupheads [...,*,...]`

```

* sectionnumber = yes NUMBER no
  alternative = normal margin middle TEXT
                paragraph
separator = TEXT
stopper = TEXT
align = inner outer left right middle
        normal no yes
alightitle = yes float no
tolerance = verystRICT strict tolerant
            verytolerant stretch
indentnext = yes no
command = \..#1#2
margin = DIMENSION
    
```

`\setupindenting [...,*,...]`

```

* never none not no yes always first next small
  medium big normal odd even DIMENSION
    
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\setupinmargin [.1.] [...,2,...]
OPTIONAL
```

```
1 left right NUMBER
2 location = left right both
   style = normal bold slanted boldslanted
         type cap small... COMMAND
   before = COMMAND
   after = COMMAND
   align = inner outer left right middle
         normal no yes
   line = NUMBER
   distance = DIMENSION
   separator = TEXT
   width = DIMENSION
   distance = DIMENSION
   stack = yes no
   inherits from \setupframed
```

```
\setupinteraction [...,*...]
```

```
* state = start stop
  menu = on off
  page = yes no
  click = yes no
  split = yes no
  display = new
  openaction = REFERENCE
  closeaction = REFERENCE
  openpageaction = REFERENCE
  closepageaction = REFERENCE
  calculate = IDENTIFIER
  strut = yes no
  width = DIMENSION
  height = DIMENSION
  depth = DIMENSION
  style = normal bold slanted
         boldslanted type cap small...
         COMMAND
  color = IDENTIFIER
  contrastcolor = IDENTIFIER
  symbolset = IDENTIFIER
  title = TEXT
  subtitle = TEXT
  author = TEXT
  date = TEXT
  keyword = TEXT
  fieldlayer = auto IDENTIFIER
```

```
\setupitemize [.1.] [...,2,...] [...,3,...]
OPTIONAL OPTIONAL OPTIONAL
```

```
1 NUMBER each
2 standard broad serried packed unpacked stopper
   joinedup atmargin inmargin autointro loose repeat
   SECTION paragraph intext random
3 margin = no standard DIMENSION
   leftmargin = no standard DIMENSION
   rightmargin = no standard DIMENSION
   width = DIMENSION
   distance = DIMENSION
   factor = NUMBER
   items = NUMBER
   start = NUMBER
   before = COMMAND
   inbetween = COMMAND
   after = COMMAND
   left = TEXT
   right = TEXT
   beforehead = COMMAND
   afterhead = COMMAND
   headstyle = normal bold slanted boldslanted
               type cap small... COMMAND
   marstyle = normal bold slanted boldslanted
               type cap small... COMMAND
   symstyle = normal bold slanted boldslanted
               type cap small... COMMAND
   stopper = TEXT
   n = NUMBER
   symbol = NUMBER
   align = inner outer left right middle
         normal no yes
   indentnext = yes no
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\setuplayout [...,*,...]`

```
* width = DIMENSION fit middle
height = DIMENSION fit middle
backspace = DIMENSION
topspace = DIMENSION
margin = DIMENSION
leftmargin = DIMENSION
rightmargin = DIMENSION
header = DIMENSION
footer = DIMENSION
top = DIMENSION
bottom = DIMENSION
leftedge = DIMENSION
rightedge = DIMENSION
headerdistance = DIMENSION
footerdistance = DIMENSION
topdistance = DIMENSION
bottomdistance = DIMENSION
leftmargindistance = DIMENSION
rightmargindistance = DIMENSION
leftedgedistance = DIMENSION
rightedgedistance = DIMENSION
horoffset = DIMENSION
veroffset = DIMENSION
style = normal bold slanted
boldslanted type cap
small... COMMAND

color = IDENTIFIER
marking = on off color screen TEXT
location = left middle right bottom
top singlesided doublesided

scale = DIMENSION
nx = NUMBER
ny = NUMBER
dx = DIMENSION
dy = DIMENSION
lines = NUMBER
columns = NUMBER
columnndistance = DIMENSION
grid = yes no
bottomspace = DIMENSION
cutspace = DIMENSION
textdistance = DIMENSION
textwidth = NUMBER
textmargin = DIMENSION
clipoffset = DIMENSION
page = IDENTIFIER
paper = IDENTIFIER
```

`\setuplist [...,1...] [...,2,...]`

```
1 IDENTIFIER
2 state = start stop
alternative = a b c ... none command
coupling = on off
criterium = SECTION local previous current
pageboundaries = LIST
style = normal bold slanted
boldslanted type cap small...
COMMAND

numberstyle = normal bold slanted
boldslanted type cap small...
COMMAND

textstyle = normal bold slanted
boldslanted type cap small...
COMMAND

pagestyle = normal bold slanted
boldslanted type cap small...
COMMAND

color = IDENTIFIER
command = \...#1#2#3
numbercommand = \...#1
textcommand = \...#1
pagecommand = \...#1
interaction = cd:sectionnumber TEXT
pagenumber all

before = COMMAND
after = COMMAND
inbetween = COMMAND
left = TEXT
right = TEXT
label = yes no
prefix = yes no none
pagenumber = yes no
headnumber = yes no
cd:sectionnumber = yes no
alighttitle = yes no
margin = DIMENSION
width = DIMENSION fit
height = DIMENSION fit broad
depth = DIMENSION fit broad
distance = DIMENSION
separator = TEXT
stopper = TEXT
symbol = none 1 2 3 ...
expansion = yes no command
maxwidth = DIMENSION
inherits from \framed
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\setupmakeup [1.] [...,2,...]`

```

1 IDENTIFIER
2 width      = DIMENSION
  height     = DIMENSION
  voffset    = DIMENSION
  hoffset    = DIMENSION
  page       = left yes right
  commands   = COMMAND
  doublesided = yes no empty
  headerstate = normal stop start empty none
               nomarking
  footerstate = normal stop start empty none
               nomarking
  textstate  = normal stop start empty none
               nomarking
  topstate   = stop start
  bottomstate = stop start
  pagestate  = stop start
  color      = IDENTIFIER
    
```

`\setuppagenumbering [...,*,...]`

```

* alternative = singlesided doublesided
  location    = header footer left right
               middle margin marginedge
               inleft inright
  conversion  = numbers characters Characters
               romannumerals Romannumerals
  style       = normal bold slanted
               boldslanted type cap small...
               COMMAND
  left        = TEXT
  right       = TEXT
  way         = bytext bycd:section
  text        = TEXT
  numberseparator = TEXT
  textseparator = TEXT
  cd:sectionnumber = yes no
  separator   = TEXT
  strut       = yes no
  state       = start stop
  width       = DIMENSION
  command     = \...#1
    
```

`\setupparagraphs [1.] [2.] [...,3,...]`
OPTIONAL

```

1 IDENTIFIER
2 NUMBER each
3 style     = normal bold slanted boldslanted
               type cap small... COMMAND
  width     = DIMENSION
  height    = DIMENSION
  align     = inner outer left right middle
               normal no yes
  tolerance = verystRICT strict tolerant
               verytolerant stretch
  distance  = DIMENSION
  before    = COMMAND
  after     = COMMAND
  inner     = COMMAND
  command   = COMMAND
  rule      = on off
    
```

`\setuppositioning [...,*,...]`

```

* state = start overlay
  unit  = cm pt em mm ex es in
  factor = NUMBER
  scale = NUMBER
  offset = yes no
  xstep  = absolute relative
  ystep  = absolute relative
    
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\setupregister [1.] [2.] [...,3,...]`
OPTIONAL

```

1 SINGULAR NAME
2 IDENTIFIER
3 n = NUMBER
balance = yes no
align = inner outer left right
      middle normal no yes
style = normal bold slanted
      boldslanted type cap small...
      COMMAND
pagestyle = normal bold slanted
          boldslanted type cap small...
          COMMAND
textstyle = normal bold slanted
          boldslanted type cap small...
          COMMAND
indicator = yes no
coupling = yes no
cd:sectionnumber = yes no
criterium = SECTION local all
distance = DIMENSION
symbol = 1 2 ... n a ... none
interaction = pagenummer TEXT
expansion = yes no command
referencing = on off
command = \...#1
location = left middle right
maxwidth = DIMENSION
unknownreference = empty none
alternative = a b A B
prefix = both first none
compress = no yes
deeptextcommand = \...#1
    
```

`\setuprotate [...,*,...]`

```

* rotation = NUMBER
location = normal high fit broad depth
inherits from \setupframed
    
```

`\setupsectionblock [1.] [...,2,...]`

```

1 IDENTIFIER
2 number = yes no
page = yes right
before = COMMAND
after = COMMAND
    
```

`\setupsynonyms [1.] [...,2,...]`

```

1 IDENTIFIER
2 textstyle = normal bold slanted boldslanted
            type cap small... COMMAND
synonymstyle = normal bold slanted boldslanted
              type cap small... COMMAND
location = left right top serried inmargin
          inleft inright
width = DIMENSION
state = start stop
criterium = all used
conversion = yes no
expansion = yes no command
command = \...#1#2#3
    
```

`\setuptables [...,*,...]`

```

* distance = small medium big
bodyfont = 5pt ... 12pt small big
HL = NUMBER small medium big none
VL = NUMBER small medium big none
depth = NUMBER strut
height = NUMBER strut
rulethickness = DIMENSION
rulecolor = IDENTIFIER
align = inner outer left right
       middle normal no yes
commands = COMMAND
align = inner outer left right
       middle normal no yes
background = screen color none
backgroundscreen = NUMBER
backgroundcolor = IDENTIFIER
bodyfont = 5pt ... 12pt small big
    
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\setuptabulate` [`1.`] [`2.`,`...`]

OPTIONAL

1 IDENTIFIER

2 unit = DIMENSION
 indenting = never none not no yes always
 first next small medium big
 normal odd even DIMENSION

before = COMMAND
 after = COMMAND
 inner = COMMAND
 EQ = TEXT
 rulecolor = IDENTIFIER
 align = inner outer left right middle
 normal no yes

rulethickness = DIMENSION
 distance = blank grid depth DIMENSION small
 medium big none

bodyfont = Spt ... 12pt small big
 rule = normal line
 split = yes no

`\setuptextbackground` [`1.`,`2.`,`...`]

* leftoffset = DIMENSION
 rightoffset = DIMENSION
 topoffset = DIMENSION
 bottomoffset = DIMENSION
 before = COMMAND
 after = COMMAND
 state = start stop
 location = text paragraph none
 alternative = 0 1 2
 inherits from \setupframed

`\setupthinrules` [`1.`,`2.`,`...`]

* interlinespace = small medium big
 n = NUMBER
 before = COMMAND
 inbetween = COMMAND
 after = COMMAND
 color = IDENTIFIER
 backgroundcolor = IDENTIFIER
 height = DIMENSION max
 depth = DIMENSION max
 alternative = a c
 rulethickness = DIMENSION
 color = IDENTIFIER
 background = color none
 backgroundcolor = IDENTIFIER

`\setuptype` [`1.`,`2.`,`...`]

* space = on off
 option = slanted normal none
 style = normal bold slanted boldslanted type
 cap small... COMMAND
 color = IDENTIFIER

`\setuptyping` [`1.`] [`2.`,`...`]

OPTIONAL

1 file typing IDENTIFIER

2 space = on off
 page = yes no
 option = slanted normal commands color none
 text = yes no
 icommand = COMMAND
 vcommand = COMMAND
 ccommand = COMMAND
 before = COMMAND
 after = COMMAND
 margin = DIMENSION standard yes no
 evenmargin = DIMENSION
 oddmargin = DIMENSION
 blank = DIMENSION small medium big standard
 halflines line

escape =
 space = on off
 tab = NUMBER yes no
 page = yes no
 indentnext = yes no
 style = normal bold slanted boldslanted
 type cap small... COMMAND

color = IDENTIFIER
 palet = IDENTIFIER
 lines = yes no hyphenated
 empty = yes all no
 numbering = line file no

`\setupwhitespace` [`1.`,`2.`,`...`]

OPTIONAL

* none small medium big line fixed fix DIMENSION

`\showsymbolset` [`1.`,`2.`,`...`]

* IDENTIFIER

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+


```
\startalignment [...] ...
\stopalignment OPTIONAL
```

* inherits from \setupalign

```
\startbuffer [...] ... \stopbuffer
OPTIONAL
```

* IDENTIFIER

```
\startcolumns [...,*,...] ...
\stopcolumns OPTIONAL
```

* inherits from \setupcolumns

```
\startcombination [...] ...
\stopcombination
```

* N*M

```
\startformula ... \stopformula
```

```
\startframedtext [1.] [...,2,...] {3.}
{4.} ... \stopframedtext OPTIONAL
```

- 1 left right middle none
- 2 inherits from \setupframedtexts
- 3 TEXT
- 4 TEXT

```
\starthiding ... \stophiding
```

```
\startinteractionmenu [...] ...
\stopinteractionmenu
```

* IDENTIFIER

```
\startitemize [...1,...] [...,2,...] ...
\stopitemize OPTIONAL OPTIONAL
```

- 1 a A KA n N m r R KR NUMBER continue
standard broad serried packed stopper joinedup
atmargin inmargin intro columns text paragraph
repeat
- 2 inherits from \setupitemize

```
\startlinecorrection ...
\stoplinecorrection
```

```
\startlines ... \stoplines
```

```
\startlocal ... \stoplocal
```

```
\startlocalfootnotes ...
\stoplocalfootnotes
```

```
\startnamemakeup [...,*,...] ...
\stopnamemakeup OPTIONAL
```

* inherits from \startmakeup

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\startpacked [...] ... \stoppacked
      OPTIONAL
```

* blank

```
\startpostponing ... \stoppostponing
```

```
\starttable [...] ... \stoptable
```

* TEXT IDENTIFIER

```
\starttabulate [.1.] [.2.] ...
\stoptabulate  OPTIONAL  OPTIONAL
```

1 TEXT
2 TEXT

```
\starttextbackground ...
\stoptextbackground
```

```
\starttyping ... \stoptyping
```

```
\startunpacked ... \stopunpacked
```

```
\subject [...] [...] {.2.}
      OPTIONAL
```

1 REFERENCE
2 TEXT

```
\subsection [...] [...] {.2.}
      OPTIONAL
```

1 REFERENCE
2 TEXT

```
\subsubject [...] [...] {.2.}
      OPTIONAL
```

1 REFERENCE
2 TEXT

```
\switchtobodyfont [...] [...] ...]
```

* 5pt ... 12pt small big global

```
\textreference [.1.] {.2.}
```

1 REFERENCE
2 TEXT

```
\thinrule
```

```
\thinrules [...] [...]
      OPTIONAL
```

* inherits from \setupthinrules

```
\title [...] [...] [...] [.2.}
      OPTIONAL
```

1 REFERENCE
2 TEXT

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

```
\type {...}
```

```
* TEXT
```

```
\useblocks [...1;...] [...2;...]  
OPTIONAL
```

```
1 IDENTIFIER  
2 IDENTIFIER
```

```
\useexternaldocument [.1.] [.2.] [.3.]
```

```
1 IDENTIFIER  
2 file  
3 TEXT
```

```
\usemodule [...;*...]
```

```
* IDENTIFIER
```

```
\usesymbols [...]
```

```
* IDENTIFIER
```

```
\writebetweenlist [.1.] {.2.}
```

```
1 SECTION IDENTIFIER  
2 TEXT
```

```
\writetolist [.1.] {.2.} {.3.}
```

```
1 SECTION IDENTIFIER  
2 TEXT  
3 TEXT
```

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

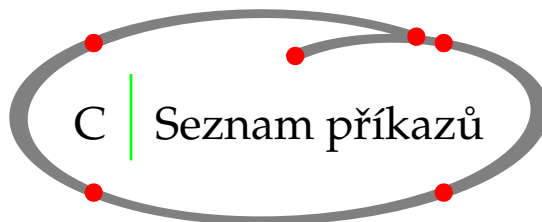
vyhledej

+

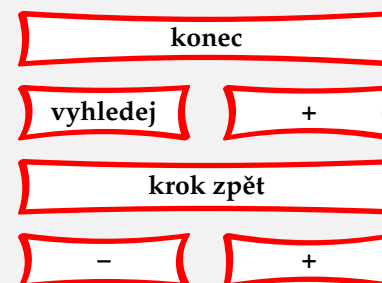
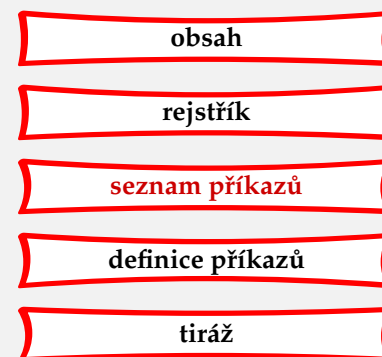
krok zpět

-

+



— 164	\de 153
- 164	\defineblock 141
\\ 152	\definebodyfont 114
	\definecolor 97
\abbreviation 93	\definecombinedlist 85
\adaptdisplay 121	\definedescription 68
\af 153	\defineenumeration 70
\at 95, 106	\definefloat 138
	\definelist 85
\bf 113	\definemakeup 136
\blank 127	\defineregister 90
	\definesymbol 14
\ca 153	\definesynonyms 93
\cap 115	\definetabulate 57
\chapter 9	\DL 42
\cn 153	\DR 42
\color 97	
\column 61	\em 115
\completecontent 85	\en 153
\completeindex 90	\enableregime 117, 162, 163
\completelistofabbreviations 93	\es 153
\completeregister 90	\externalfigure 34
\crlf 152	
\currentdate 149, 164	\fi 153
\cz 153	\footnote 64
	\FR 42
\da 153	\fr 153
\DC 42	\framed 73



<code>\frenchspacing</code> 164	<code>\mainlanguage</code> 153,162
<code>\from</code> 106	<code>\margintitle</code> 78
	<code>\midaligned</code> 102
<code>\getbuffer</code> 144	<code>\month</code> 164
<code>\goto</code> 106	<code>\MR</code> 42
<code>\gr</code> 153	
	<code>\NC</code> 42,57
<code>\hairline</code> 145	<code>\nl</code> 153
<code>\head</code> 14	<code>\no</code> 153
<code>\hideblocks</code> 141	<code>\noheaderandfooterlines</code> 84
<code>\high</code> 148	<code>\noindenting</code> 130
<code>\HL</code> 42	<code>\nowhitespace</code> 127
<code>\hr</code> 153	<code>\NR</code> 42,57
<code>\hu</code> 153	
	<code>\overstrikes</code> 145
<code>\in</code> 95,106	
<code>\indenting</code> 130	<code>\page</code> 80
<code>\index</code> 90	<code>\pagereference</code> 95
<code>\inframed</code> 73	<code>\par</code> 126
<code>\infull</code> 93	<code>\paragraph</code> 9,126
<code>\inleft</code> 78	<code>\percent</code> 33
<code>\inmargin</code> 78,152	<code>\periods</code> 145
<code>\input</code> 154	<code>\permille</code> 33
<code>\inright</code> 78	<code>\pl</code> 153
<code>\it</code> 153	<code>\placecontent</code> 85
<code>\item</code> 14	<code>\placefigure</code> 34
	<code>\placeformula</code> 27
<code>\ja</code> 153	<code>\placeindex</code> 90
	<code>\placeintermezzo</code> 138
<code>\language</code> 153	<code>\placelistofabbreviations</code> 93
<code>\leftlines</code> 102	<code>\placeregister</code> 90
<code>\loadabbreviations</code> 93	<code>\placetable</code> 42
<code>\lohi</code> 148	<code>\position</code> 149
<code>\LOW</code> 42	<code>\pt</code> 153
<code>\low</code> 148	
<code>\LR</code> 42	<code>\quotation</code> 164

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

`\quote` 164
`\rightaligned` 102
`\rm` 113
`\ro` 153
`\rotate` 150
`\ru` 153

`\setupalign` 102
`\setupbackgrounds` 101
`\setupblank` 127
`\setupblock` 141
`\setupbodyfont` 111, 117, 163
`\setupbuffer` 144
`\setupcaption` 138
`\setupcaptions` 34, 42, 138
`\setupcolors` 97
`\setupcolumns` 61
`\setupcombinedlist` 85
`\setupdescriptions` 68
`\setupenumerations` 70
`\setupfigures` 34
`\setupfillinlines` 145
`\setupfillinrules` 145
`\setupfloat` 138
`\setupfloats` 34, 42, 138
`\setupfooter` 84
`\setupfootertexts` 84
`\setupfootnotes` 64
`\setupformulae` 27
`\setupframed` 73
`\setupframedtext` 76
`\setuphead` 9
`\setupheader` 84
`\setupheadertexts` 84
`\setupheads` 9

`\setupheadtext` 164
`\setupindenting` 130
`\setupinteraction` 105
`\setupitemize` 14
`\setuplayout` 121
`\setuplist` 85
`\setupmakeup` 136
`\setupmargin` 78
`\setuppositioning` 149
`\setuppagenumbering` 80
`\setupparagraphs` 131
`\setupregister` 90
`\setupsynonyms` 93
`\setuptables` 42
`\setuptabulate` 57
`\setuptextbackground` 100
`\setupthinrules` 145
`\setuptolerance` 102
`\setuptype` 116
`\setuptyping` 116
`\setupwhitespace` 127
`\showframe` 121
`\showlayout` 121
`\showmakeup` 121
`\showpatterns` 117, 162
`\showsetups` 121
`\showsymbolset` 120
`\sk` 153
`\sl` 113, 153
`\SR` 42
`\ss` 113
`\startbuffer` 144
`\startchemical` 57
`\startcolumns` 61
`\startcombination` 34, 42
`\startfiguretext` 34

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

<code>\startformula</code> 27	<code>\thinrule</code> 145
<code>\startframedtext</code> 76	<code>\thinrules</code> 145
<code>\starthiding</code> 145	<code>\THREE</code> 42
<code>\startitemize</code> 14	<code>\title</code> 9
<code>\startlinecorrection</code> 127	<code>\tr</code> 153
<code>\startlines</code> 152	<code>\tt</code> 113
<code>\startlocal</code> 121	<code>\TWO</code> 42
<code>\startnamemakeup</code> 136	<code>\type</code> 116
<code>\startpacked</code> 127	<code>\typebuffer</code> 144
<code>\startpostponing</code> 80	
<code>\startstandardmakeup</code> 136	<code>\ua</code> 153
<code>\starttable</code> 42	<code>\uk</code> 153
<code>\starttabulate</code> 57	<code>\underbar</code> 145
<code>\starttext</code> 162	<code>\unit</code> 33, 93
<code>\starttextbackground</code> 100	<code>\us</code> 153
<code>\starttextrule</code> 145	<code>\useblocks</code> 141
<code>\starttyping</code> 116	<code>\useencoding</code> 163
<code>\startunpacked</code> 127	<code>\useexternaldocument</code> 106
<code>\stoptext</code> 162	<code>\usemodule</code> 155
<code>\subject</code> 9	<code>\usesymbols</code> 120
<code>\subparagraph</code> 9	<code>\usetypescript</code> 117, 163
<code>\subsubject</code> 9	
<code>\sv</code> 153	<code>\VL</code> 42
<code>\switchtobodyfont</code> 111	<code>\vn</code> 153
<code>\texteuro</code> 120	<code>\weekday</code> 164
<code>\tfa</code> 113	<code>\whitespace</code> 127
<code>\tfb</code> 113	<code>\writebetweenlist</code> 85
<code>\tfc</code> 113	<code>\writetolist</code> 85
<code>\tfd</code> 113	

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

D | Rejstřík

a

akcenty 119
 AMS 111
 AMSmath 155

b

barva 97
 barva pozadí 100
 bibliografie 155
 blok textu 141

c

Computer Modern 111
 cont-sys.tex 159
 Courier 163

č

čeština 162
 číslované poznámky 70
 číslování stránek 80

d

datum 149, 164

definice
 pojmu 68
 příkazů a maker 133
 den v týdnu 164
 dělení slov 153
 display mód 21
 Acrobat Distiller 104
 dolní index 148
 dvi-soubor 5

e

euro 120
 Acrobat Exchange 104

f

font
 bezpatkový 113
 Computer Modern 111
 Courier 163, 113
 Helvetica 163
 kurzivní 113
 Latin Modern 163
 neproporcionální 116
 řez 111
 skloněný 113
 Times 163

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej +

krok zpět

- +

tučný 113
velikost 111, 113
formátování 121, 131

g

GNUplot 155
grafika 158

h

Helvetica 163
hlavička
 viz záhlaví
horní index 148
hrana textu 121

ch

chemické struktury 155

i

index
 dolní 148
 horní 148
interakce
 vnější 106
 vnitřní 106
interaktivní mód 105
italika
 viz kurziva

j

jazyk 153

jednotka SI 33
jednotky 33, 155

k

kapitálky 115
komentář kódu 145, 154
kódování
 dělení slov 117, 163
 fontu 117, 163
 vstupní 117, 163
křížové reference 95
kurziva 113

l

Latin Modern 163
LilyPond 155
linky 145
Lucida Bright 111

m

marginálie 78, 121
matematické symboly 30
matematický mód 21
matematika 21
METAPOST 158
mezera
 meziodstavcová 127
měsíce 164
minimální dokument 162
modul
 amsl 155
 bib 155
 gnuplot 155

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

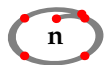
+

krok zpět

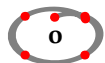
-

+

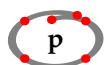
chart 155
 chemic 155
 lilypond 155
 narrowtt 155
 nath 155
 pictex 155
 steps 155
 streams 155
 units 155



nadpisy 9
 nastavení 131
 názvy 164

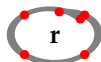


obrázek 34
 obsah 85
 odkazování 95
 odrážky 14
 ve sloupcích 14
 odsazení 130
 odstavec
 odsazení 130, 126
 ukončení 126
 vyznačení 76
 orámování textu 73
 otočení 150

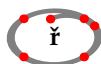


patička
 viz úpatí
 pdf-soubor 5

pictex 155
 písmo
 viz font
 plocha stránky 101
 plovoucí objekt 34, 42, 138
 polohování 149
 pomlčka 164
 pomocné soubory 161
 popisky 164
 pozadí
 stránky 101
 text 100
 poznámky pod čarou 64
 % ve vstupním souboru 154
 procento 33
 promile 33
 PS-fonty 111



Acrobat Reader 104
 reference 95
 referenční jméno 95
 rejstřík 90
 režim
 viz kódování vstupní
 rotace 150



řazení 164
 řádkový rejstřík 64
 řecké znaky 30

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

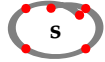
vyhledej

+

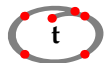
krok zpět

-

+

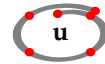


seznamy 85
 skrytý text 145
 sloupce 57, 61
 slovenština 162
 složené znaky 119
 spodek textu 121
 spojovník 164
 stavové diagramy 155
 stránkové zrcadlo 121
 strojopis 116
 úzký 155
 symboly
 matematické 30
 řecké 30, 120
 synchronizace textů 155
 synonyma 93

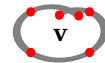


tabelace 57
 tabulky
 odstavcový text 57, 42
 text
 skrytý 145
 uložení 144
 textový blok 141
 textový mód 21
 T_EXUTIL 159
 Times 163
 titulní strana 136
 tub-soubor 161
 tuf-soubor 161
 tui-soubor 161
 tuo-soubor 159, 161

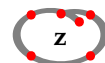
typescript 117, 163



ukončení řádku 152
 uložení textu 144
 umístění do předepsané polohy 149
 uvozovky
 dvojitě 164
 jednoduché 164
 uživatelská nastavení 159
 úpatí 84, 121
 úprava stránky 121



vektorová grafika 158
 velikost fontu 114
 verbatim 116
 vršek textu 121
 vstup jiného souboru 154
 vstupní soubor 3
 zpracování 5, 159
 vyznačení odstavce 76
 výčty 14
 vývojové diagramy 155
 vzorce 27



zalamování stránek 80
 zarovnání 102
 záhlaví 84, 121
 zvláštní znaky 7
 zvýraznění 115

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+

Autor základního textu
Ton Otten

Překlad
Vít Zýka, Ján Buša, Jiří Hrbek, Martina Plachá a Petr Tesařík

Design a typografie
Hans Hagen

Ilustrace
Johan Jonker

PRAGMA ADE
Ridderstraat 27
8061GH Hasselt NL
+31 (0)38 477 53 69
www.pragma-ade.com

6. prosince 2006

obsah

rejstřík

seznam příkazů

definice příkazů

tiráž

konec

vyhledej

+

krok zpět

-

+