LaTeX
Slike, novi ukazi, nova okolja

Matjaž Željko

Fakulteta za matematiko in fiziko

10. november 2012

Plavajoči objekti

Besedilo lahko vsebuje veliko slik in tabel, ki jih ni možno deliti med stranmi. Sliko oziroma tabelo, ki je prevelika za tekočo stran, pustimo za naslednjo stran, prostor na tekoči strani pa zapolnimo z besedilom, ki sledi.

LaTeX ponuja dve okolji za plavajoče objekte: `table` za tabele in `figure` za slike.

Vse znotraj okolja `figure` oziroma `table` se obravnava kot en plavajoči objekt. Obe okolji imata še opcijske parametre v obliki

`\begin{figure}[`*lega*`]` ali `\begin{table}[`*lega*`]`

ki določajo lego. Parameter *lega* je sestavljen iz ene ali več črk, ki označujejo dovoljene položaje.

Lega plavajočega objekta

Oznaka	Objekt lahko stoji . . .
`h`	*tukaj* na mestu v tekstu, kjer je vstavljen.
`t`	na *vrhu* strani.
`b`	na *dnu* strani.
`p`	na posebni *strani*, ki vsebuje le plavajoče objekte.
`!`	ignoriraj ostale parametre pri pozicioniranju.

Tak npr. `\begin{table}[!hbp]` pomeni, da lahko LaTeX tabelo postavi točno na to mesto (`h`), na dno strani (`b`) ali pa na posebno stran (`p`), to vse pa lahko naredi tudi, če rezultat ni najlepši (`!`). Če ne podamo opcijskega argumenta z lego, potem standardni razredi privzamejo lego `[tbp]`.

Plavajoči objekti

Z ukazom `\caption{`*pojasnilo*`}` definiramo pojasnilo za objekt. Za oštevilčenje in niz »Figure« oziroma »Table« (v slovenščini »Slika« in »Tabela«) poskrbi LaTeX.
Ukaza `\listoffigures` in `\listoftables` delujeta podobno kot `\tableofcontents`, izpišeta pa kazalo slik oziroma tabel. V teh kazalih se še enkrat ponovijo celotna pojasnila. Če uporabljamo dolga pojasnila, lahko podobno kot pri logičnih enotah kot dodatni parameter v oglatih oklepajih zapišemo kratko pojasnilo za kazalo.

```
\caption[Kratko]{Dolgo pojasnilo.}
```

S pomočjo ukazov `\label` in `\ref` se lahko v besedilu sklicujemo na plavajoči objekt.

Vključevanje zunanjih slik

Zapletenejše slike narišemo v zunanjih programih (Mathematica, GeoGebra, Corel Draw, ...) in jih nato vključimo v LaTeX. Slike nastopajo v različnih formatih:

- vektorski: PDF, EPS, WMF, ...
- rastrski: PNG, BMP, TIF, JPG/JPEG, ...

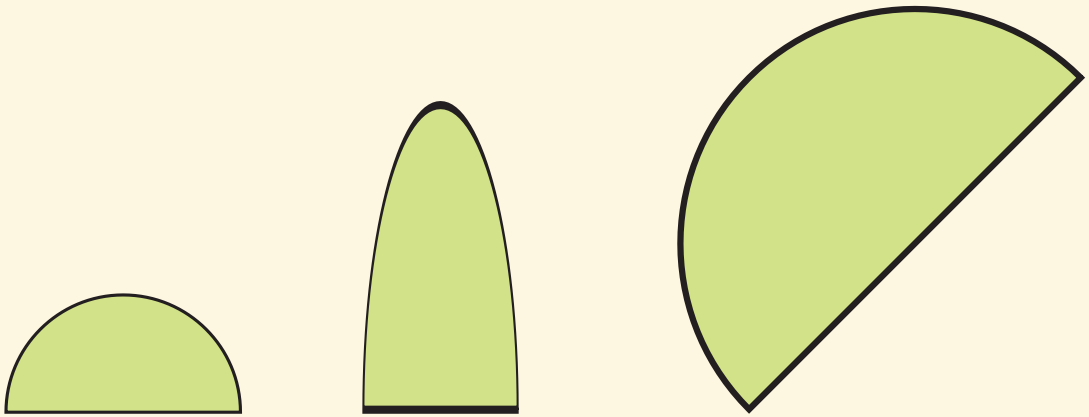
Če je le možno, uporabimo vektorski format, pri rastrskem pa se izogibamo formatu JPG/JPEG, saj zaradi načina kompresije slika ni nujno dovolj ostra. V LaTeX in PDFLaTeX lahko vključimo:

format	DVI	PS	PDF
TIF	•		
BMP	•		
JPG			•
PNG			•
EPS		•	
PDF			•

Vektorska grafika

Elementi na sliki v **vektorskem formatu** so podana v obliki geometrijskih objektov. Sliko lahko transformiramo (povečamo, vrtimo, raztegnemo itd.) brez poslabšanja kakovosti slike. Z ustreznimi programi lahko vektorsko sliko enostavno spreminjamo.

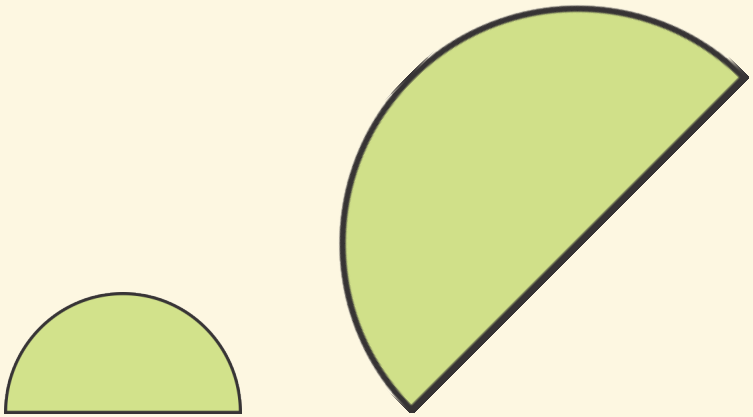
```
\includegraphics{polkrog.pdf} % original: 2 cm x 1cm
\includegraphics[width=1cm, height=2cm]{polkrog.pdf}
\includegraphics[scale=2, angle=45]{polkrog.pdf}
```



Rastrska grafika

Slika v **rastrskem formatu** je podana kot matrika točk z določenimi lastnostmi (npr. barva, intenziteto, ...). **Če sliko transformiramo, se lahko njena kakovost bistveno spremeni.** Ker so v sliki v rastrskem formatu vsi elemeti zliti z ozadjem (npr. napis na pisanem ozadju), je spreminjanje takih slik praktično nemogoče.

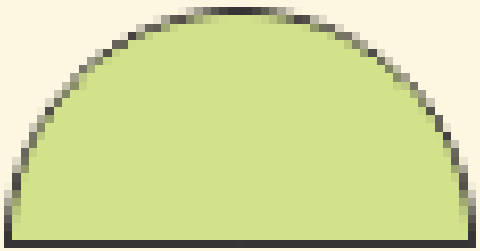
```
\includegraphics{polkrog.png} % 600 DPI
\includegraphics[scale=2, angle=45]{polkrog.png}
```



Rastrska grafika

Pomembna informacija vsake rastrske slike je njena **ločljivost**, podana v enotah DPI (dots per inch). Za kakovosten natis je priporočljivo uporabljati rastrske slike z ločljivostjo 600 DPI. Zaslonske slike ali slike, ki jih narišemo v enostavnih grafičnih programih, imajo običajno bistveno premajhno ločljivost (npr. 96 DPI).

```
\includegraphics[scale=2]{polkrog96.png} % 96 DPI
\includegraphics[scale=2]{polkrog.jpg}   % 300 DPI
```



Vključevanje zunanjih slik

Uporabljamo paket `graphicx`, ki ga naložimo z

- `\usepackage[dvips]{graphicx}` za LaTeX,
- `\usepackage[pdftex]{graphicx}` za PDFLaTeX.

Posamezno sliko z *datoteke* vključimo z ukazom `\includegraphics[`*velikost*`]{`*datoteka*`}`. Pri tem v parametru *velikost* navedemo velikost slike v obliki

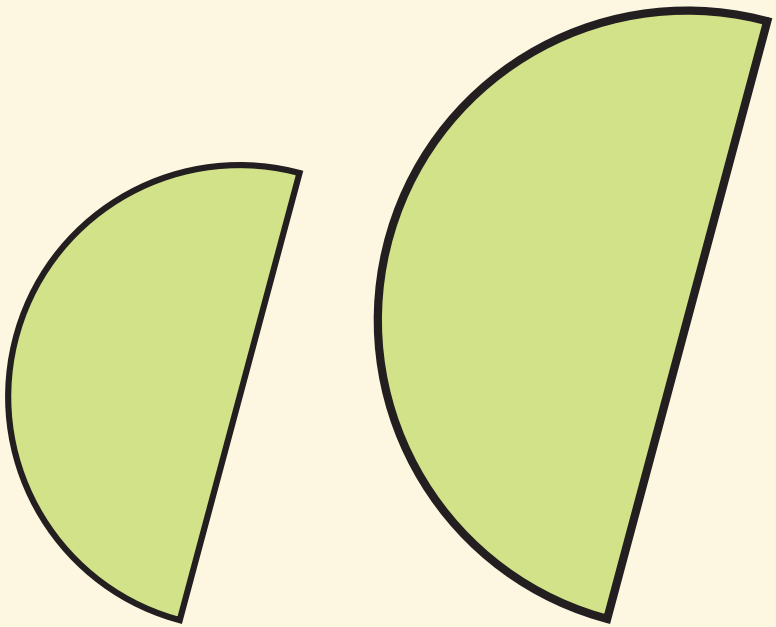
`\includegraphics[bb=`*llx lly urx ury*`, angle=`*kot*`, width=`*širina*`, height=`*višina*`, scale=`*razmerje*`]{`*datoteka*`}`.

Parametri v `bb` so koordinate (z enotami; privzeto pike) levega spodnjega kota in desnega zgornjega kota. Parameter *kot* je kot rotacije v stopinjah, *širina* in *višina* pa predstavljata dimenzije slike. Če je podan le en podatek, se drugega določi avtomatično, da se ohrani razmerje. **Vsi parametri ne delujejo za vse formate in za vse vrste izhodnih datotek.**

Vključevanje zunanjih slik

Parametri ukaza `\includegraphics` se upoštevajo po vrsti.

```
\includegraphics[width=3cm, angle=75]{polkrog.pdf}
\includegraphics[angle=75, width=3cm]{polkrog.pdf}
```



Vključevanje slike v formatu EPS

To sliko lahko vključimo v PS datoteko. V DVI datoteki bomo na zaslonu videli sliko, a se lahko zgodi, da se ne bo izpisala. Za izpis moramo datoteko prevesti z `dvips` v PostScript, potem pa jo lahko izpišemo.
Datoteke, z vključeno EPS sliko, ne moremo prevesti z PDFLaTeX-om, lahko pa dobljeno PS datoteko naknadno spremenimo v PDF z ukazom `ps2pdf`.

```
\documentclass[a4paper]{article}
\usepackage[dvips]{graphicx}
\begin{document}
\includegraphics[width=6cm]{pink.eps}
\end{document}
```

Vključevanje slik v PDFLaTeX

V PDFLaTeXu slike vključimo z rahlimi spremembami. Preberemo lahko formate JPEG, PNG in PDF, vse pa vključimo na isti način. V PDFLaTeXu parameter `bb` ne deluje, zato pa delujejo vsi ostali.

```
\documentclass[a4paper]{article}
\usepackage[pdftex]{graphicx}
\begin{document}
\includegraphics[width=5cm, angle=90]{graf.png}
\end{document}
```

Kako isti dokument uporabimo za LaTeX in PDFLaTeX?

Če nimamo slik, potem v ukazni vrstici dokument namesto z `latex` prevedemo s `pdflatex` in morali bi dobiti datoteko PDF z enako vsebino kot DVI.

Če pa dokument vsebuje slike, jih pripravimo v različnih grafičnih formatih vključujemo na različne načine. Možno pa je v datoteki povedati, da določene vrstice pridejo v poštev le za `latex`, določena pa le za `pdflatex`.

```
\newif\ifpdf
\ifx\pdfoutput\undefined
\pdffalse
\else
\pdfoutput=1
\pdftrue
\fi

\documentclass[11pt,a4paper]{article}
\ifpdf \usepackage[pdftex]{graphicx}
\else \usepackage[dvips]{graphicx} \fi
\begin{document}
\ifpdf
   \includegraphics[width=4cm]{graf.png}
\else
   \includegraphics[width=4cm]{graf.eps}
\fi
\end{document}
```

Definicije novih ukazov

Ze definiranje novega ukaza ali za spremembo že obstoječega so na voljo ukazi

`\newcommand{`*ime ukaza*`}[`*narg*`][`*opcija*`]{`*definicija*`}`
`\renewcommand{`*ime ukaza*`}[`*narg*`][`*opcija*`]{`*definicija*`}`
`\providecommand{`*ime ukaza*`}[`*narg*`][`*opcija*`]{`*definicija*`}`

- `\newcommand` definira nov ukaz in bo javil napako, če poskušamo definirati že obstoječi ukaz.
- `\renewcommand` že obstoječi ukaz spremeni, bo pa protestiral, če ukaz še ne obstaja.
- `\providecommand` če ukaz že obstaja, ne naredi ničesar, sicer pa definira nov ukaz.

Število argumentov podamo v *narg* (med 1 in 9), če tega argumenta ni, potem ukaz nima nobenega parametra. V *opcija* podamo privzeto vrednost za opcijski parameter, ki ga sprejema ta ukaz.

Primeri novih ukazov s parametri

```
\newcommand{\azap}{a_1\cdots a_n}
\newcommand{\bzap}[1]{#1_1\cdots #1_n}
\newcommand{\czap}[2]{#1_1\cdots #1_#2}
\newcommand{\dzap}[2][n]{#2_1\cdots #2_#1}

$\azap=\bzap{a}=\czap{a}{n}=\dzap{a}\ne\dzap[m]{a}$.
```

$a_1 \cdots a_n = a_1 \cdots a_n = a_1 \cdots a_n = a_1 \cdots a_n \neq a_1 \cdots a_m$.

Ukaz `\dzap` je primer ukaza z neobveznim argumentom. Neobvezni argument je vedno prvi med vsemi argumenti, vnašamo pa ga med oglatimi oklepaji. Tako moramo tudi navesti v definiciji, kjer ima neobvezni argument vedno oznako `#1`. Kot število parametrov moramo v definiciji navesti vsoto neobveznih in obveznih parametrov.

Števci

LaTeX samostojno številči strani, poglavja, enačbe, ... s števci:

`part`	`chapter`	`paragraph`	`figure`	`enumi`
	`section`	`subparagraph`	`table`	`enumii`
	`subsection`	`page`	`footnote`	`enuiii`
	`subsubsection`	`equation`	`mpfootnote`	`enumiv`

Števci `enumi`, ..., `enumiv` štejejo indeks v okolju `enumerate`, od zunanjega okolja do morebitnega na globini 4. Števec `mpfootnote` šteje opombe pod črto v okolju `minipage`.

Poleg teh števcev imamo za matematične trditve še števce, definirane z ukazom `\newtheorem`.

Vrednost števca je celo in ponavadi nenegativno število.
Ko uporabimo npr. ukaz `\subsection` in se izpiše npr. 7.2, to pomeni, da ima števec `subsection` vrednost 2, števec `section` pa 7. Ukaz `\subsection` povzroči, da se izpiše oznaka podpodrazdelka, hkrati pa se ustrezni števec poveča za ena.

Novi števci

Z ukazom `\newcounter{`*ime števca*`}[`*nadrejeni števec*`]` definiramo nov števec z izbranim *imenom*, njegova začetna vrednost pa je 0. Če definiramo še *nadrejeni števec*, se bo vrednost novega števca postavila na 0 vsakič, ko se bo povečala vrednost *nadrejenega števca*.

Vrednost števca spreminjamo z naslednjimi ukazi:

- `\setcounter{`*števec*`}{`*vrednost*`}` : nastavimo *vrednost števca*,
- `\addtocounter{`*števec*`}{`*dodatek*`}`: vrednosti *števca* prištejemo *dodatek*,
- `\stepcounter{`*števec*`}`: vrednost *števca* se poveča za 1,
- `\refstepcounter{`*števec*`}`: vrednost *števca* se poveča za 1, poleg tega pa *števec* postane tisti števec, na katerega vrednost se postavi referenca `\ref` na ustrezni `\label`.

Vrednost števca kot celo število dobimo z ukazom `\value{`*števec*`}`.

Izpisovanje števcev

Vrednost števca lahko izpišemo v različnih oblikah:

- `\arabic{`*števec*`}`: arabske številke: 1, 2, 3, 4, ...,
- `\Roman{`*števec*`}`: rimske številke z velikimi črkami: I, II, III, ...,
- `\roman{`*števec*`}`: rimske številke z malimi črkami: i, ii, iii, ...,
- `\alph{`*števec*`}`: male črke: a, b, c, ...,
- `\Alph{`*števec*`}`: velike črke: A, B, C, ...,
- `\fnsymbol{`*števec*`}`: simbol za opombe pod črto: ∗, †, ‡, §, ¶, ‖, ∗∗, ††, ‡‡.

Nekateri števci poznajo ukaz `\the`*števec*, ki izpiše vrednost *števca*, kot npr. `\thepage`, `\thesection`, ...

```
To je stran \thepage.
```

To je stran 19.

Spremenjen izpis števcev

Če nismo zadovoljni z videzov števcev pri enačbah, razdelkih, okolju `enumerate`, ..., lahko spremenimo ustrezno definicijo. Običajno je

```
\newcommand{\thechapter}{\arabic{chapter}}
\newcommand{\thesection}{\thechapter.\arabic{section}}
\newcommand{\thesubsubsection}{\thesection.\arabic{su
```

Primer spremembe številčenja enačb:

```
\renewcommand{\theequation}
{\thesubsubsection/\Roman{equation}}
\begin{equation}
a=b+c
\end{equation}
```

$$a = b + c \qquad (0/I)$$

Spremenjen izpis števcev

Spremenjen izpis je npr.:

```
\renewcommand{\theenumi}{\alph{enumi}}
\renewcommand{\theenumii}{\theenumi.\Roman{enumii}}
```

a. Ena
   a.i Dva
   a.ii Tri
b. Štiri

Na ta način smo spremenili le vrednost števca, ne pa tudi način izpisa. Če želimo spremeniti še to, moramo spremeniti ukaze `\labelenumi`, . . . , ki v resnici izpišejo oznako pred poljem.

```
\newcommand{\labelenumii}{\theenumii.}
```

a. Ena
   a.i. Dva
   a.ii. Tri
b. Štiri

Nova okolja

Podobno kot nove ukaze lahko definiramo tudi nova okolja. Ukaz `\newenvironment` ima naslednjo obliko:

`\newenvironment{`*ime*`}[`*num*`]{`*preden*`}{`*potem*`}`

Podobno kot pri ukazu `\newcommand`, lahko `\newenvironment` uporabljamo z neobveznim argumentom ali pa brez njega. Vsebina *preden* se izvede pred procesiranjem teksta v okolju, vsebina *potem* pa se procesira takrat, ko srečamo ukaz `\end{`*ime*`}`.

```
\newenvironment{misel}
{\rule{1ex}{1ex}\hfill}{\hfill\rule{1ex}{1ex}}

\begin{misel}
Kdor visoko leta je pilot.
\end{misel}
```

■ Kdor visoko leta je pilot. ■

Nova okolja

Okolja, ki že obstaja, ne moremo ponovno definirati okolja. Če želimo spremeniti kakšno že obstoječe okolje, uporabimo ukaz `\renewenvironment`, ki ga uporabljamo enako kot `\newenvironment`.

V stavkih, ki se izvedejo na koncu okolja, ne smemo uporabljati parametrov.

```
\newenvironment{misel}[1]
{(#1)\hfill}{\hfill\rule{1ex}{1ex}}

\begin{misel}{pilot}
Kdor visoko leta, je pilot.
\end{misel}
```

(pilot) Kdor visoko leta, je . . . . ■

Nova okolja

V novih okoljih lahko uporabljamo tudi števce, ki jih prej definiramo.

```
\newcounter{stm}
\newenvironment{misel}[1]
{\stepcounter{stm}\textbf{\arabic{stm}}: #1\hfill}
{\hfill\rule{1ex}{1ex}}

\begin{misel}{pilot}
Kdor visoko leta je pilot.
\end{misel}

\begin{misel}{razgled}
Kdor visoko leta, ima lep razgled.
\end{misel}
```

1: pilot Kdor visoko leta, je pilot. ■

2: razgled Kdor visoko leta, ima lep razgled. ■