

Praktičen uvod v L^AT_EX

Jure Slak

Kazalo

1	Osnovno o L^AT_EXu	2
2	Struktura in oblikovanje dokumenta	3
2.1	Velikost in oblika pisave	3
2.2	Prazen prostor	3
2.3	Seznami	4
2.4	Ločila in opombe	5
3	Matematična okolja	5
3.1	Presledki	6
3.2	Poravnava in večji izrazi	6
3.3	Različni tipi črk	7
3.4	Lastni ukazi	7
4	Slike in tabele	8
4.1	Vstavljanje slik	8
4.2	Vstavljanje tabel	8
4.3	Slike in tabele med besedilom	8
5	Sklicevanje in kazala	9
5.1	Sklicevanje	9
5.2	Kazala	10
6	Navajanje literature	10
7	Razno (algoritmi, izvorna koda, več stolpcev)	10

1 Osnovno o L^AT_EXu

L^AT_EX (izgovorjava “lateh”) je sistem za pisanje dokumentov, ki ga je razvil Leslie Lamport kot izboljšavo predhodnega sistema Donalda Knutha, imenovanega T_EX.

L^AT_EX dokumente se ponavadi piše v .tex datoteke. Najbolj osnovna struktura datoteke izgleda tako:

```
1 \documentclass{article}
2 % preambula
3 \begin{document}
4 vsebina
5 \end{document}
```

Na vrhu je definiran tip dokumenta. Običajno se za manjše dokumente uporablja article, obstajajo pa tudi drugi tipi, ponavadi tudi dani s strani inštitucije, za katero se dokument piše. Temu sledi **preambula**, del dokumenta, ki ni vsebinski, ampak je namenjen različnim nastavitvam. Nato sledi značka za začetek dokumenta, sledi vsebina in značka za konec dokumenta. Vse kar sledi tej znački, prevajalnik ignorira.

Tekom prevajanja program pdf_latex producira različne datoteke .log. .toc, .aux itd., ki jih lahko po koncu pobrišemo. Tudi če gre med prevajanjem kaj narobe, jih lahko pobrišemo, da začnemo na novo.

Latex pozna ukaze in okolja:

- **ukazi** so oblike `\ukaz`, `\ukaz{arg1}`, `\ukaz{arg1}{arg2}`, itd. ali `\ukaz[arg]`, `\ukaz[arg]{arg}`, `\ukaz[arg]{arg1}{arg2}`, itd., kjer so argumenti `arg1`, `arg2` v zavutih oklepajih obvezni, argument `arg` v oglatih pa neobvezen in ga lahko izpustimo.
- **okolja** so oblike `\begin{okolje} ... \end{okolje}`, `\begin{okolje}{arg1} ... \end{okolje}`, itd. ali `\begin{okolje}[arg] ... \end{okolje}`, `\begin{okolje}[arg]{arg1} ... \end{okolje}`, itd., kjer kot prej v zavutih oklepajih navajamo obvezne argumente, v oglatih pa neobvezne. Kodo znotraj okolji ponavadi zamikamo za 2 presledka ali pa sploh ne.
- poznamo tudi **stare T_EX ukaze**, ki so oblike `{\ukaz}` in veljajo od ukaza do konca trenutnih zavutih oklepajev ali okolja.

Komentarje pišemo z znakom %. Prevajalnik ignorira vse od znaka % do konca vrstice.

Drugi **posebni znaki** so # \$ % ^ & _ { } ~ \. Če jih želite uporabiti v dokumentu dobesedno, je potrebno napisati `\#` `\$` `\%` `\^` `\&` `_` `\{` `\}` `\~` `\.` `\textbackslash`.

V preambuli dokumenta definiramo z ukazom `\usepackage[options]{package-name}` vse pakete, ki jih bomo uporabljali. Standardna preambula vsebuje:

```
1 \documentclass[a4paper,oneside,12pt]{article}
2
3 \usepackage[utf8]{inputenc}      % make čšž work on input
4 \usepackage[T1]{fontenc}        % make čšž work on output
5 \usepackage[slovene]{babel}     % slovenian language and hyphenation
6 \usepackage[reqno]{amsmath}     % basic math package
7 \usepackage{amssymb,amsthm}     % symbols and theorem environments
8 \usepackage{graphicx}           % images
9
10 \usepackage[
11   top=2.5cm,
12   bottom=2.5cm,
13   left=2.5cm,
14   right=2.5cm
15 ]{geometry} % page geomerty
16
17 % vstavi svoje pakete tukaj
18
19 % clickable references, pdf table of contents
20 \usepackage[bookmarks,colorlinks=true,linkcolor=black,anchorcolor=black,
21   citecolor=black,filecolor=black,menucolor=black,runcolor=black,
22   urlcolor=black,pdfencoding=unicode]{hyperref}
```

Ukazu `\documentclass` smo podali možnosti za velikost pisave in papirja. Vključimo pakete za delo z UTF-8 znaki, podporo slovenščini, osnovne matematične pakete in pakete za podporo slik. S paketom `geometry` nastavimo odmike na robu. Paket `hyperref` poskrbi za to, da lahko kliknemo na sklice in citate v PDF datoteki in da ima ta ob strani strukturo dokumenta. To lahko vidite tudi v tem dokumentu.

2 Struktura in oblikovanje dokumenta

Osnovno logično strukturo razdelkov definiramo z ukazi `\section`, `\subsection` in `\subsubsection`. Obstajajo tudi verzije ukazov z zvezdico, `\section*` itd., ki poglavij ne številčijo.

Besedilo pišemo prosto, pri čemer L^AT_EX ignorira prelome vrstic, zamike in presledke v besedilu. Sam tudi deli besede na koncu vrstice. Če katere besede ne zna deliti, mu lahko pomagamo z ukazom `\hyphenation{ra-ču-nal-nik}` v preambuli. **Nov odstavek** začnemo tako, da spustimo eno ali več praznih vrstic. Z ukazom `\\` ali `\newline` lahko na silo začnemo novo vrstico, z ukazom `\newpage` pa novo stran. Ta ukaza se uporablja sorazmerno redko.

```
1 To je odstavček z vrstico, \\ ki smo jo namenoma zlomili.
2 To je še vedno isti odstavek. LaTeX dodatne      presledke
3     ali zamike ignorira.
4
5 To je nov odstavek.
6
7 \newpage
8
9 To je nov odstavek na novi strani.
```

Stil strani lahko nastavljammo z ukazom `\pagestyle`, ki je najbolj pogosto poklican z argumentoma `empty` ali `plain`. Če želimo stil spremeniti samo za eno stran, uporabimo ukaz `\thispagestyle`. Za naslovnice se pogosto uporabi `\thispagestyle{empty}`.

Besedilo lahko poravnamo levo, desno ali sredinsko z okolji `flushleft`, `flushright` in `center`.

2.1 Velikost in oblika pisave

Za velikost pisave imamo na voljo stare T_EX ukaze:

<code>{\Huge</code>	Primer besedila}	Primer besedila
<code>{\huge</code>	Primer besedila}	Primer besedila
<code>{\LARGE</code>	Primer besedila}	Primer besedila
<code>{\Large</code>	Primer besedila}	Primer besedila
<code>{\large</code>	Primer besedila}	Primer besedila
<code>{\normalsize</code>	Primer besedila}	Primer besedila
<code>{\small</code>	Primer besedila}	Primer besedila
<code>{\footnotesize</code>	Primer besedila}	Primer besedila
<code>{\scriptsize</code>	Primer besedila}	Primer besedila
<code>{\tiny</code>	Primer besedila}	Primer besedila

Za obliko pisave imamo na voljo naslednje L^AT_EX ukaze:

Roman (serif):	<code>\textrm{Primer besedila}</code>	Primer besedila
Sans-serif:	<code>\textsf{Primer besedila}</code>	Primer besedila
Monospaced (typewriter-type):	<code>\texttt{Primer besedila}</code>	Primer besedila
Bold face:	<code>\textbf{Primer besedila}</code>	Primer besedila
Italic:	<code>\textit{Primer besedila}</code>	<i>Primer besedila</i>
Small caps:	<code>\textsc{Primer besedila}</code>	PRIMER BESEDILA
Upright:	<code>\textup{Primer besedila}</code>	Primer besedila
Slanted:	<code>\textsl{Primer besedila}</code>	<i>Primer besedila</i>
Medium weight:	<code>\textmd{Primer besedila}</code>	Primer besedila

Ukaze lahko seveda gnezdimo. Za uporabo znotraj matematičnih okolij imamo verzije `\mathrm`, ... Za **poudarjene izraze** se ponavadi uporablja ukaz `\emph{ }`, ki privzeto tekst napiše ležeče, dvojno poudarjen tekst pa zopet pokončno.

2.2 Prazen prostor

Praviloma se L^AT_EXu pusti, da se sam določa, kam postavi stvari in koliko prostora pusti vmes, pri čemer imamo na voljo veliko nastavitev.

Presledki: Za piko \LaTeX običajno vstavi daljši presledek kot ponavadi, kar je na koncu stavka zaželeno, na koncu okrajšav pa ne. Zato je potrebno za okrajšavami vstaviti trdi presledek `_`. Poleg tega imamo na voljo tudi nedeljivi presledek `~` (zapisan z vijugico), na katerem \LaTeX ne bo dovolil preloma vrstice. Ta se lahko uporablja za okrajšavami, vedno pa se ga uporablja tudi pred sklici. Obratno, če imamo piko za veliko črko, \LaTeX ne bo vstavil večjega presledka, vendar mu lahko z ukazom `\@` povemo, da ta pika končuje stavek. Primer napačne in pravilne rabe `\@` in `_` vidimo spodaj:

“Rad imam BASIC. Verjetno sem starejši od npr. 50 let.”

“Rad imam BASIC. Verjetno sem starejši od npr. 50 let.”

Pravilno pisanje okrajšav je torej `npr._tako` ali `npr.~tako`. Enako velja za datume. Nedeljivi presledek se piše tudi `npr. pri prof. Golob: prof.~Golob` in njegova uporaba je na splošno precej pogosta.

Daljšje presledke za piko lahko izklopimo z ukazom `\frenchspacing`.

Če uporabimo nek ukaz, npr. `\LaTeX`, po njem \LaTeX ne bo avtomatsko vstavil presledka, da lahko napišemo npr. “v `\LaTeX` u lahko”, kar se prevede v “v \LaTeX u lahko”. Če želimo presledek, je potrebno napisati `\LaTeX{}` ali `\LaTeX_`.

Odstavki: Nov odstavek privzeto ne preskoči nič prostora, naslednji je le zamaknjen noter. To lahko kontroliramo z dolžinama `\parskip` in `\parindent`. S kombinacijo ukazov

```
1 \setlength{\parskip}{12pt}
2 \setlength{\parindent}{0pt}
```

lahko \LaTeX pripravimo do tega, da pušča pod ostavki 12pt prostora in jih ne zamika, podobno kot Word. Če zgolj enega ostavka ne želimo zamakniti, uporabimo `\noindent`. Pri prelomu vrstice lahko kot opcijski argument povemo, koliko prostora naj spusti, npr. `\\[12pt]`

Preostale dolžine, ki se uporabljajo so še `\lineskip`, `\linewidth`, `\textwidth`, `\pagewidth` (in `height` verzije).

Poljuben prazen prostor: Za vstavljanje praznega prostora v horizontalni ali vertikalni smeti lahko uporabimo ukaza `\hspace` in `\vspace`, kot npr. `\hspace{10pt}` ali `\vspace{-2cm}`. Prazen prostor se ne vstavi če smo na robu strani. Če želimo vstavljanje praznega prostora tudi v tem primeru, uporabimo `\hspace*`. Če se zalotite, da uporabljate veliko ukazov `\hspace` ali `\vspace`, najbrž nekaaj počnete narobe.

Raztegljiv prazen prostor: vstavimo lahko tudi prostor, ki se raztegne, kolikor ima prostora na strani. To naredimo z `\hfill` in `\vfill` ali `\hspace*{\fill}` in `\vspace*{\fill}`. Zvezdica ima enak pomen kot zgoraj. To je uporabno, če želimo nekaaj poravnati na sredino strani vertikalno, recimo na naslovnici:

```
1 Univerza v Ljubljani
2 \vfill
3 Naslov diplome \\
4 Avtor diplome
5 \vfill
6 13.\ 4.\ 2018
```

ali pa v vrstici desno pri glavi uradnega pisma:

```
1 Fakulteta za matematiko in fiziko \\
2 Jadranska 19 \\
3 1000 Ljubljana \hfill 20.\ 3.\ 2018 \\
```

2.3 Sezname

Za urejene sezname imamo okolje `enumerate`, za neurejene pa `itemize`. Za opisne sezname imamo `description`. Nov element seznama začnemo z ukazom `\item`, ki za opcijski argument sprejme besedilo oz. znak za začetek te točke seznama.

<pre> 1 \begin{enumerate} 2 \item Prvi element. 3 \item Drugi element. Seznam ima lahko 4 več vrstic. 5 6 In več odstavkov. 7 8 \item Lahko ima tudi gnezdene sezname: 9 \begin{itemize} 10 \item Gnezned element 1 11 \item[\text{\diamond}] Gnezdene element 2 12 \end{itemize} 13 \end{enumerate} </pre>	<p>1. Prvi element.</p> <p>2. Drugi element. Seznam ima lahko več vrstic. In več odstavkov.</p> <p>3. Lahko ima tudi gnezdene sezname:</p> <ul style="list-style-type: none"> • Gnezned element 1 ◊ Gnezdene element 2
--	--

Več opcij glede seznamov ponuja paket `enumerate`.

2.4 Ločila in opombe

Večaj, **pomišljaj** in dolgi pomišljaj naredimo z ukazi `-`, `--` in `---`.

Narekovaje naredimo s pari ```text'`, `„text'`, `>>text<<`, ki izgledajo kot `“text”`, `„text”`, `»text«`. Pozor: noben od znakov v kodi ni znak za dvojni narekovaj. Večina urejevalnikov ima na voljo možnost, da se nastavi `“smart quotes”`, ki ob znaku za dvojni narekovaj vstavijo primerne narekovaje. Podobno naredimo enojne narekovaje ``text'` za `'text'`.

Tri pike naredimo z ukazom `\dots`: `...in` ne tako, da napišemo `... ali . . .`

Znak $\text{\textcircled{R}}$ vstavimo z `\textregistered`.

Naglase naredimo z ukazi `\{o}`, `\`{o}`, `\~{o}`, `\^{o}`, `\~{o}` za `ó`, `ò`, `ö`, `ô`, `õ`. Lahko jih napišemo tudi direktno v besedilo. Za `ö` je uporabljen znak za dvojni narekovaj. Pišemo jih lahko tudi na kratko kot `\'o`.

Opombe pod črto naredimo z ukazom `\footnote{besedilo}`, kot tukaj.¹

Ligature so skupki več znakov, ki so povezani, za lažje branje. \LaTeX jih naredi avtomatsko, kot npr.

tukaj: **fiz** Lahko jih prekinemo, tako da med črke vstavimo prazno škatlico: `f\mbox{}iz`: **fiz**.

3 Matematična okolja

Za pisanje matematičnih formul imamo dva načina: tekstovnega in vrstičnega (med njima lahko tudi ročno preklapljam z `\textstyle` in `\displaystyle`).

Tekstovni način začnemo in končamo z znakom `$` in vse vmes se izpiše v matematičnem načinu. Naj bo `n` naravno število izpiše `“Naj bo n naravno število”`. Vsaka spremenljivka mora biti napisana v matematičnem načinu, sicer izgleda narobe. Primerjava:

`“V i -ti vrstici algoritem obravnava število a .”` in `“V i -ti vrstici algoritem obravnava število a .”`

Enačbe v svoji vrstici naredimo s parom `\[\]`.² Npr. `\[a+b = c \]` izpiše

$$a + b = c$$

Za številčene enačbe uporabljamo okolje `equation`, kot `\begin{equation} a+b = c \end{equation}`:

$$a + b = c \tag{1}$$

V enačbah ne sme biti praznih vrstic, ker bi začele nov odstavek.

Podpisane in **nadpisane** znake dobimo kot `a^b` in `a_b`. Za več kot en nadpisan znak je potrebno uporabiti zavite oklepaje: `2^{10}`, sicer dobimo `210`.

Neenakosti pišemo z `$a < b$`, `$a > b$`, `$a \leq b$`, `$a \geq b$`, `$a \approx b$`: `a < b`, `a > b`, `a ≤ b`, `a ≥ b`, `a ≈ b`.

Elementarne funkcije so vgrajene in jih pišemo kot

`$\sin x`, `\log(x+y)`, `\sqrt{3}`, `\sqrt[7]{x+\frac{2}{y}}`: `\sin x`, `\log(x+y)`, `\sqrt{3}`, `\sqrt[7]{x+\frac{2}{y}}`.

Primerjava med uporabo ukaza `\sin` in kar `sin`: `\sin x`, `sinx`, `\sin(x)`, `sin(x)`.

¹Več ukazov za naglase lahko najdete na https://en.wikibooks.org/wiki/LaTeX/Special_Characters#Escaped_codes.

²Včasih se je uporabljalo `$$ $$`, vendar ima ta kombinacija lahko težave. Več: <https://tex.stackexchange.com/questions/503/why-is-preferable-to>

Če naša izbrana funkcija, npr. sinc, ni definirana, jo lahko za enkratno uporabo pišemo z ukazom `\operatorname{sinc}`, kar zagotovi pravilno računanje presledkov okoli nje, če pa jo uporabljamo večkrat, se jo splača v preambuli definirati z `\DeclareMathOperator{\sinc}{sinc}`, da jo lahko uporabljamo kot `\sinc x`.

Oklepaje napišemo kot `(x)`, `[x]`, `\{x\}`, `|x|`, `\|x\|`, `\langle x \rangle` za (x) , $[x]$, $\{x\}$, $|x|$, $\|x\|$, $\langle x \rangle$. Uporaba `||x||` ali `<x>` da napačne rezultate, primerjava $\|x\|$ in $\|x\|$ ter $\langle x \rangle$ in $\langle x \rangle$. Oklepaje lahko tudi skaliramo z `$(\frac{a}{b})$` za

$$\left(\frac{a}{b}\right) \quad \text{namesto} \quad \frac{a}{b}.$$

Velike operatorje, kot so

$$\sum_{i=1}^n \frac{1}{i} \quad \int_a^b f(x) dx \quad \max_{x \in [a,b]} f(x) \quad \lim_{n \rightarrow \infty} \sqrt[n]{n} \quad \bigcup_{i=1}^{\infty} [i, i+1] \times \emptyset$$

pišemo z

```
1 \sum_{i=1}^n \frac{1}{i}
2 \int_a^b f(x)\,dx
3 \max_{x \in [a, b]} f(x)
4 \lim_{n \to \infty} \sqrt[n]{n}
5 \bigcup_{i = 1}^{\infty} [i, i+1] \times \emptyset
```

Grške črke pišemo s primernimi ukazi:

αA	<code>\alpha A</code>	νN	<code>\nu N</code>
βB	<code>\beta B</code>	$\xi \Xi$	<code>\xi \Xi</code>
$\gamma \Gamma$	<code>\gamma \Gamma</code>	$o O$	<code>o O</code>
$\delta \Delta$	<code>\delta \Delta</code>	$\pi \varpi \Pi$	<code>\pi \varpi \Pi</code>
$\varepsilon \epsilon E$	<code>\epsilon \varepsilon E</code>	$\rho \varrho P$	<code>\rho \varrho P</code>
ζZ	<code>\zeta Z</code>	$\sigma \varsigma \Sigma$	<code>\sigma \varsigma \Sigma</code>
ηH	<code>\eta H</code>	τT	<code>\tau T</code>
$\theta \vartheta \Theta$	<code>\theta \vartheta \Theta</code>	$\upsilon \Upsilon$	<code>\upsilon \Upsilon</code>
ιI	<code>\iota I</code>	$\phi \varphi \Phi$	<code>\phi \varphi \Phi</code>
κK	<code>\kappa K</code>	χX	<code>\chi X</code>
$\lambda \Lambda$	<code>\lambda \Lambda</code>	$\psi \Psi$	<code>\psi \Psi</code>
μM	<code>\mu M</code>	$\omega \Omega$	<code>\omega \Omega</code>

Oznake nad črkami dobimo z ukazi `\bar{z}`, `\vec{z}`, `\hat{z}`, `\dot{z}`, `\ddot{z}`, `\tilde{z}`: \bar{z} , \vec{u} , \hat{x} , \dot{y} , \ddot{k} . Ukaza `\imath` in `\jmath` naredita i in j brez pike, da lahko nanju damo oznake. Pozor: če dajemo oznako nad več kot eno črko, se ta postavi na sredino, kar ni nujno zaželeno: primerjaj `\vec{p}_j` in `\vec{p}_j`: \vec{p}_j , \vec{p}_j . Pogoste je treba oznako dati samo na črko, dodatne okraske pa se napiše kasneje.

Obstaja tudi ogromno drugih simbolov, ki se jih ponavadi najde iskanjem po spletu oblike “latex what is the correct symbol for [insert here]” ali uporabo celotne tabele simbolov [2] ali orodja za prepoznavo simbolov [5]. Ne poskušajte simbolov napisati s pomočjo znakov na tipkovnici, ki izgledajo podobno, ampak poiščite pravi simbol.

3.1 Presledki

Presledki v matematičnem načinu delujejo drugače kot v običajnem tekstovnem načinu. \LaTeX ignorira vse presledke, in se sam odloči, koliko presledka dati okoli črk in operatorjev.

Za lastno vstavljanje presledkov imamo na voljo čedalje večje presledke `\!`, `\.`, `\:`, `\;`, `\quad`, `\qquad`. Prvi je negativen, `\.` je majhen presledek, primeren za npr. razdaljo med $f(x)$ in dx v integralu $\int f(x) dx$, zadnji pa je precej dolg (dvakrat `fontsize`). Enak presledek kot med besedami dobimo z `_`.

3.2 Poravnava in večji izrazi

Poravnane enačbe dobimo z okoljema `align` in `align*`, pri čemer zvezdica kot ponavadi označuje verzijo brez številčenja. Uporaba:

```
1 \begin{align*}
2 \nabla \cdot \vec{B} &= 0 \\
3 \nabla \times \vec{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
4 \end{align*}
```

Rezultat:

$$\begin{aligned}\nabla \cdot \vec{B} &= 0 \\ \nabla \times \vec{E} &= -\frac{1}{c} \frac{\partial B}{\partial t}\end{aligned}$$

Enačbe so poravnane na znak `&`, znak `\\` pa označuje novo vrstico.

Opomba: v zadnji vrstici `align` ne smemo imeti `\\`, sicer bo po enačbah dodaten prazen prostor. Prav tako pred in po okolju ne smemo imeti praznih vrstic, če ne želimo dodatnega praznega prostora ali nezaželenih zamikov.

Matrike pišemo z okolji tipa `matrix` pri čemer so definirana okolja z začetnimi črkami `pbBvV` za različne vrste oklepajev.

$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{100} \begin{vmatrix} 1 & \cdots & n \\ \vdots & \ddots & \vdots \\ n & \cdots & 1 \end{vmatrix}$	<pre> 1 \[2 \begin{bmatrix} 3 a & b \\ 4 c & d \\ 5 \end{bmatrix}^{100} 6 \begin{vmatrix} 7 1 & \cdots & n \\ 8 \vdots & \ddots & \vdots \\ 9 n & \cdots & 1 \\ 10 \end{vmatrix} 11 \]</pre>
---	--

Funkcije z več možnostmi definiramo s pomočjo okolja `cases`.

$f: \mathbb{R} \rightarrow \mathbb{R}$ $f(x) = \begin{cases} x, & x > 0 \\ -x & x \leq 0 \end{cases}$	<pre> 1 \begin{align*} 2 f & \colon \mathbb{R} \to \mathbb{R} \\ 3 f(x) & = \\ 4 \begin{cases} 5 x, & x > 0 \\ 6 -x & x \leq 0 \\ 7 \end{cases} \\ 8 \end{align*}</pre>
---	--

3.3 Različni tipi črk

V matematičnem načinu imamo za pisanje besedila na voljo ukaz `\text`. Znotraj ukaza `\text` lahko ponovno uporabljamo `$` in znotraj zopet `\text`, kolikorkrat želimo. Poleg slogov naštetih v razdelku 2.1, imamo na voljo še nekaj posebnih slogov, samo za matematični način.

Glavni so `\mathcal`, ki ponuja pisane (calligraphic) črke, kot so $\mathcal{L}\mathcal{P}\mathcal{Q}\mathcal{H}\mathcal{U}\mathcal{V}\mathcal{I}$, `\mathbb`, ki ponuja dvojne (blackboard bold) črke, kot $\mathbb{N}\mathbb{R}\mathbb{C}\mathbb{Z}\mathbb{A}\mathbb{H}$ in `\mathfrak`, ki ponuja “fraktur” verzijo: $\mathfrak{D}\mathfrak{F}\mathfrak{O}\mathfrak{S}\mathfrak{R}\mathfrak{L}$ (DFOSRL). Paket `mathrsfs` ponuja “script” verzijo črk, kot so $\mathcal{I}\mathcal{M}\mathcal{J}\mathcal{S}\mathcal{L}\mathcal{Y}$ (IMJSLY).

Krepke črke v matematičnem načinu dobimo z ukazom `\boldsymbol` (ki ga je mogoče, kljub imenu, uporabiti na več kot enen simbolu hkrati).

Primer:

```

$a \mathbf{a} \textbf{a} \boldsymbol{a}$:  $aa\mathbf{a}$ 
 $\alpha \mathbf{\alpha} \boldsymbol{\alpha}$ :  $\alpha\mathbf{\alpha}\boldsymbol{\alpha}$ 
```

3.4 Lastni ukazi

V \LaTeX u lahko definiramo tudi lastne ukaze, z ukazom `\newcommand{\ukaz}{nekaj kar želimo narediti}`. S tem lahko okrajšamo nekatere dolge ponavljajoče skupine ukazov, če npr. zelo pogosto uporabljamo `\mathsf{T}` za transponiranje, lahko to okrajšamo v `\T` z ukazom `\newcommand{\T}{\mathsf{T}}`. V tem primeru pišemo `$A^{\T}`, da dobimo A^{T} .

Definiramo lahko tudi ukaze, ki sprejmejo argumente. Poglejmo si to kar na primeru:

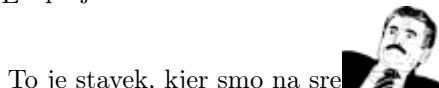
`\newcommand{\dpar}[2]{\frac{\partial #1}{\partial #2}}` definira ukaz `\dpar`, ki sprejme 2 argumenta, na katera se sklicujemo z `#1` in `#2`. Uporabimo ga kot `\dpar{x}{t}`, kar se med prevajanjem spremeni v `\frac{\partial x}{\partial t}`, kar se prikaže kot $\frac{\partial x}{\partial t}$.

Ne bojite se uporabljati novih ukazov, saj lahko precej izboljšajo berljivost kode. Na prvo vrsto ukazov glejte kot na neke vrste spremenljivke, na drugo vrsto pa kot na funkcije in jih uporabljajte podobno kot bi jih v drugih programskih jeziki: če se neka stvar večkrat ponavlja, jo shrani v spremenljivko ali naredi funkcijo.

4 Slike in tabele

4.1 Vstavljanje slik

Za delo s slikami potrebujemo paket `graphicx`. Slike vključimo z ukazom `\includegraphics{slika.png}`, \LaTeX pa jih obravnava kot črke enake velikosti, kot je velikost vključene slike:



To je stavek, kjer smo na sredini vstavili sliko.

Velikost slike lahko kontroliramo z neobveznimi argumenti, npr.:

```
\includegraphics[width=0.5\linewidth]{slike/surprise.png}.
```

Pogosto se uporablja relativne velikosti, kot zgoraj, lahko pa se predpiše tudi velikost v pikslih ali centimetrih. Višino lahko kontroliramo s parametrom `height`, na voljo pa imamo tudi veliko drugih možnosti za zrcaljenje, obrezovanje ipd.

Če je možno, naj bodo vključene slike **vektorske**, (format `pdf`, `eps`), saj se bodo tako skalirale brez izgub. Vključite pa lahko tudi najpogostejše formate bitnih slik, kot npr. `png` in `jpg`.

4.2 Vstavljanje tabel

Table naredimo z okoljem `tabular` in jih podobno kot slike \LaTeX obravnava kot velike črke. Primer tabele in njene izvorne kode:

```
1 \begin{tabular}{l|l|r|c} \hline
2 1.\ naloga & 2.\ naloga & 3.\ naloga
3 & Skupaj \\ \hline \hline
4 1 & 0.75 & 0.25 & 2 \\
5 0.5 & 0.25 & 0.25 & 1 \\
6 0.5 & 0 & 0.25 & 0.75 \\
7 0.25 & 1 & 2 & 3.25 \\
8 1 & 2 & 3 & 6 \\ \hline
9 \end{tabular}
```

1. naloga	2. naloga	3. naloga	Skupaj
1	0.75	0.25	2
0.5	0.25	0.25	1
0.5	0	0.25	0.75
0.25	1	2	3.25
1	2	3	6

Kot pri matrikah znaki `&` ločujejo stolpce, znaki `\\` pa označujejo nove vrstice. Ukaz `\hline` naredi horizontalno črto, vertikalne črte in poravnave stolpcev pa so podane kot prvi argument okolja `tabular`.

Napisati je možno tudi mnogo bolj zapletene tabele, vendar je verjetno lažje uporabiti kakšen generator \LaTeX kode, kot npr. [6].

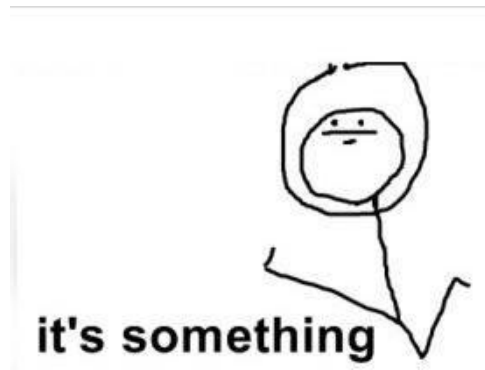
4.3 Slike in tabele med besedilom

Običajno v dokumentih slike in tabele plavajo med besedilom. Zato imamo na voljo okolji `figure` in `table`. Običajna koda za vstavljanje slike v dokument je:

```
1 \begin{figure}[h]
2 \centering
3 \includegraphics[width=0.4\linewidth]{slike/its_something.jpg}
4 \caption{Primer vstavljanja slike.}
5 \label{fig:its-something}
6 \end{figure}
```

To naredi plavajočo sliko, ki jo bo \LaTeX poskušal postaviti v dokumentu tja, kjer je definirana (to pomeni neobvezni parameter `h` – here; druge možnosti so `t` – top (na vrhu strani), `b` – bottom, `p` – posebna stran). Ukaz `\centering` vse znotraj okolja `figure` postavi na sredino. Nato se vključi slika, pod sliko pa je njen opis. Temu sledi oznaka slike, uporabljena zgolj interno za sklicevanje. Pozor: ukazov `\caption` in `\label` ne smemo zamenjati med seboj, sicer sklicevanje ne bo delovalo.

Podobno velja za tabele, kjer plavajočo tabelo naredimo z okoljem `table`. Pomen vseh ukazov je enak kot pri okolju `figure`. Običajen primer uporabe:



Slika 1: Primer vstavljanja slike.

1. naloga	2. naloga	3. naloga	Skupaj
1	0.75	0.25	2
0.5	0.25	0.25	1
0.5	0	0.25	0.75
0.25	1	2	3.25
1	2	3	6

Tabela 1: Primer vstavljanja tabele.

```

1 \begin{table}[t]
2   \centering
3   \begin{tabular}{l|l|l|r||c}
4     % podatki od zgoraj
5   \end{tabular}
6   \caption{Primer vstavljanja tabele.}
7   \label{tab:primer}
8 \end{table}

```

\LaTeX pogosto ne postavi slik in tabel tja, kamor bi si jih želeli, ker tam ni dovolj prostora. V praksi se med pisanjem dokumenta s tem ne smemo obremenjevati, saj se lahko veliko še spremeni, preden končamo. **Ko zaključimo s pisanjem** lahko popravimo lokacije s spreminjanem velikosti slik ali tabel, s spreminjanem položaja v kodi in, v skrajnem primeru, z morebitnim kontroliranjem prostora z `\vspace`. Če je slik res veliko, lahko tudi vstavimo prelom strani, ki pred začetkom nove strani vstavi še vse do zdaj sprocesirane slike in tabele. To naredimo z ukazom `\clearpage` (z razliko od `\newpage`, ki samo gre na novo stran in ne pomaga pri tem, da bi vstavili kakšno sliko več).

Če želimo, da je slika res vstavljena točno na tisti točki, kjer je v kodi, lahko uporabimo opcijo H iz paketa `float`. Toda preden se na veliko poslužujete je opcije velja opozorilo: če je slika le malo predolga za prejšnjo stran, bo skočila na novo, vsebina prejšnje strani pa se bo raztegnila in enakomerno razporedila preostali prazen prostor, s potencialno zelo čudno izgledajočim rezultatom. V praksi je bolje uporabljati `[h]` in morda na koncu malo popraviti situacijo na roke.

Paket `caption` omogoča več nadzora nad izgledom in pozicijo slik/tabel in pripadajočih napisov.

5 Sklicevanje in kazala

5.1 Sklicevanje

Avtomatko **sklicevanje** v \LaTeX u predstavlja zelo močno orodje. Skličemo se lahko na vsak števec – števec pa je zelo veliko. \LaTeX ima števece za strani, razdelke, podrazdelke, enačbe, slike, tabele, oštevilčene sezname, ...

Oznako števca naredimo z ukazom `\label`, na oznako pa se skličemo z ukazom `\ref`. Ukaz `\label` si zapomni vrednost zadnjega števca, ki je bil spremenjen, za to običajno `\label` napišemo tik za ukazom, ki poveča kakšen števec. Za slike in tabele tako `\label` vstavimo tik za `\caption`, za razdelke tik za `\section`, pri seznamih tik za `\item`. Dodatno velja nenapisan dogovor, da se oznake poimenuje s predpono, ki pove, na kaj se oznaka nanaša: oznake za slike se začnejo s `\label{fig:oznaka}`, oznake na tabele s `tab:`, oznake

za razdelke s `sec:`, oznake za enačbe z `eq:`, oznake za elemente seznamov z `itm:`, oznake za algoritme z `alg:` in oznake za izseke izvorne kode z `lst:`. Uporaba teh oblik ni obvezna, je pa zelo uporabna pri velikih dokumentih, da mi (ali pa naš urejevalnik) lažje iščemo med obstoječimi oznakami.

Na številčene elemente se vedno sklicujemo z `\ref` ali podobnimi ukazi, nikoli s konkretnimi številkami. Pred ukazom `\ref` moramo tudi običajno vstaviti nedeljiv presledek, da se sklic ne prelomi v novo vrstico. Pravilno sklicevanje na razdelek o obliki pisave, bi bilo tako:

O obliki pisave smo govorili v razdelku `\ref{sec:font-style}`, kjer [se nadaljuje], kar v dokumentu izgleda kot “O obliki pisave smo govorili v razdelku 2.1, kjer [se nadaljuje]”. Pri tem smo tik po ukazom `\subsection{Velikost in oblika pisave}` napisali `\label{sec:font-style}`.

Sklicevanje na stran, kjer se nahaja oznaka, se naredi z ukazom `\pageref`. Sklicali bi se kot: Slika `\ref{fig:its-something}` na strani `\pageref{fig:its-something}` je lepa., kar v dokumentu izgleda kot: “Slika 1 na strani 9 je lepa.”

Pri sklicevanju na enačbe običajno uporabljamo okrogle oklepaje, kot npr. pri enačbi (1). Sklic z avtomatskimi oklepaji okrog reference dobimo z ukazom `\eqref`. Sklic malo prej smo dobili z ukazom `\eqref{eq:sample}`, pri čemer smo pred takoj po začetku okolja `equation` vstavili `\label{eq:sample}`.

5.2 Kazala

Kazalo vsebine vstavimo z ukazom `\tableofcontents`. Uporabimo lahko še ukaza `\listoffigures` in `\listoftables` za kazala slik in tabel.

Vnosi v kazalu slik in tabel so opisi pod slikami in tabelami. Če so ti predolgi, lahko uporabimo pri ukazom `caption`, neobvezen argument: `\caption[Kratek opis]{Daljši opis.}`, pri čemer se kratek opis pojavi v kazalu, dolg opis pa pod sliko/tabelo.

Vsi izrazi so že prevedeni v slovenščino s paketom `babel`. Če nam kakšen prevod ni všeč, ga lahko spremenimo npr. iz privzetega “Slike” na “Kazalo slik” z

```
\addto\captionsslovene{\renewcommand\listfigurename{Kazalo slik}}.
```

Več možnosti za oblikovanje kazala ponuja paket `tocloft`.

6 Navajanje literature

Za literaturo se v večjih dokumentih uporablja sistem \LaTeX , ki omogoča avtomatsko generiranje seznamov referenc s predpisanim stilom (pogosto inštitucije ponudijo svojega, tudi FMF). Za navajanje referenc na roko pa imamo na voljo okolje `thebibliography`.

Okolje se uporablja takole:

```
1 \begin{thebibliography}{99}
2   \bibitem{oznaka} Poljubno besedilo.
3   \bibitem{latex} Lamport, Leslie. \textit{LaTeX: A Document
4     Preparation System, 2/e}. Pearson Education India, 1994. URL:
5     \url{http://cds.cern.ch/record/270275/files/9780201529838_TOC.pdf}
6 \end{thebibliography}
```

Številka 99 pomeni, da bomo imeli največ dvomestne številke referenc. Primer uporabe lahko vidite na koncu tega dokumenta. Na literaturo se sklicujemo z ukazom `\cite{oznaka}`. Kot pri referencah, je tudi tukaj prej pravilno vstaviti nedeljiv presledek `~`. Sklicujemo se lahko na več virov naenkrat, kot npr. `\cite{a4symb,detexify}`, kar izgleda kot [2, 5]. Citiramo lahko tudi bolj natančno, tako da v oglatih oklepajih podamo dodatno vsebino: `\cite[str.\ 12]{a4symb}` izpiše [2, str. 12].

7 Razno (algoritmi, izvorna koda, več stolpcev)

Za pisanje algoritmov priporočam paketa `algorithm` in `algorithmicx` (potrebujete oba). Za vključevanje in barvanje izvorne kode (kot v tem dokumentu) je uporabljen paket `minted`. Za več stolpcev se lahko uporabi paket `multicols`. Za risanje se uporablja paket `tikz`, za diagrame pa `xy`.

Literatura

- [1] Lamport, Leslie. *LaTeX: A Document Preparation System, 2/e*. Pearson Education India, 1994. URL: http://cds.cern.ch/record/270275/files/9780201529838_TOC.pdf
- [2] Scott Pakin. *The Comprehensive L^AT_EX Symbol List*. URL: <http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>
- [3] Tobias Oetiker. *The Not So Short Introduction to L^AT_EX 2 ϵ* . URL: <https://tobi.oetiker.ch/lshort/lshort.pdf>
- [4] Bor Plestenjak (prevod in priredba). *Ne najkrajši uvod v L^AT_EX 2 ϵ* . URL: <http://www-lp.fmf.uni-lj.si/plestenjak/vaje/latex/lshort.pdf>
- [5] Daniel Kirsch. *Detexify L^AT_EX handwritten symbol recognition*. URL: <http://detexify.kirelabs.org/classify.html>
- [6] *Tables Generator*. URL: <https://www.tablesgenerator.com/>