

ΛT_EX

Móra Péter

Informatika 1 előadás, 2008. október 27.

TEX

Donald Knuth jelenleg *A számítógép-programozás művészete* című könyvének 4. kötetén dolgozik.

$\text{\TeX}$

Donald Knuth jelenleg *A számítógép-programozás művészete* című könyvének 4. kötetén dolgozik.

A 2. kötet szedési módja elavult, ezért 1976-ban a 2. kiadásban újra kellett szedni a kötetet. A matematikai képletek nem úgy jelentek meg, ahogyan azt ő szeretne volna, ezért Knuth elhatározta, hogy pár hónap alatt ír egy dokumentum szedő rendszert, a $\text{\TeX}$ -et, amiben kiadhatja művét.

$\text{\TeX}$

Donald Knuth jelenleg *A számítógép-programozás művészete* című könyvének 4. kötetén dolgozik.

A 2. kötet szedési módja elavult, ezért 1976-ban a 2. kiadásban újra kellett szedni a kötetet. A matematikai képletek nem úgy jelentek meg, ahogyan azt ő szeretne volna, ezért Knuth elhatározta, hogy pár hónap alatt ír egy dokumentum szedő rendszert, a $\text{\TeX}$ -et, amiben kiadhatja művét.

8 évvel később készült el.

$\text{\TeX}$

Donald Knuth jelenleg *A számítógép-programozás művészete* című könyvének 4. kötetén dolgozik.

A 2. kötet szedési módja elavult, ezért 1976-ban a 2. kiadásban újra kellett szedni a kötetet. A matematikai képletek nem úgy jelentek meg, ahogyan azt ő szeretne volna, ezért Knuth elhatározta, hogy pár hónap alatt ír egy dokumentum szedő rendszert, a $\text{\TeX}$ -et, amiben kiadhatja művét.

8 évvel később készült el.

A $\text{\TeX}$ verziószáma 3.1415926, és konvergál a π -hez. Alapelv, hogy a régen megírt dokumentumok is működjenek az újabb rendszerekkel, ezért a 3-as verziószám óta csak hibajavítást tartalmaz.

$\text{\TeX}$

Donald Knuth jelenleg *A számítógép-programozás művészete* című könyvének 4. kötetén dolgozik.

A 2. kötet szedési módja elavult, ezért 1976-ban a 2. kiadásban újra kellett szedni a kötetet. A matematikai képletek nem úgy jelentek meg, ahogyan azt ő szeretne volna, ezért Knuth elhatározta, hogy pár hónap alatt ír egy dokumentum szedő rendszert, a $\text{\TeX}$ -et, amiben kiadhatja művét.

8 évvel később készült el.

A $\text{\TeX}$ verziószáma 3.1415926, és konvergál a π -hez. Alapelv, hogy a régen megírt dokumentumok is működjenek az újabb rendszerekkel, ezért a 3-as verziószám óta csak hibajavítást tartalmaz.

A $\text{\TeX}$ valójában programozási nyelv, lényegében bármit meg lehet benne valósítani.

$\text{\TeX}$

Donald Knuth jelenleg *A számítógép-programozás művészete* című könyvének 4. kötetén dolgozik.

A 2. kötet szedési módja elavult, ezért 1976-ban a 2. kiadásban újra kellett szedni a kötetet. A matematikai képletek nem úgy jelentek meg, ahogyan azt ő szeretne volna, ezért Knuth elhatározta, hogy pár hónap alatt ír egy dokumentum szedő rendszert, a $\text{\TeX}$ -et, amiben kiadhatja művét.

8 évvel később készült el.

A $\text{\TeX}$ verziószáma 3.1415926, és konvergál a π -hez. Alapelv, hogy a régen megírt dokumentumok is működjenek az újabb rendszerekkel, ezért a 3-as verziószám óta csak hibajavítást tartalmaz.

A $\text{\TeX}$ valójában programozási nyelv, lényegében bármit meg lehet benne valósítani.

Megalkotta a METAFONT fontleíró rendszert, amellyel görbékkel lehet felépíteni betűket. Jelenleg 2.718281 verziószámú.

$\text{\TeX}$

Donald Knuth jelenleg *A számítógép-programozás művészete* című könyvének 4. kötetén dolgozik.

A 2. kötet szedési módja elavult, ezért 1976-ban a 2. kiadásban újra kellett szedni a kötetet. A matematikai képletek nem úgy jelentek meg, ahogyan azt ő szeretne volna, ezért Knuth elhatározta, hogy pár hónap alatt ír egy dokumentum szedő rendszert, a $\text{\TeX}$ -et, amiben kiadhatja művét.

8 évvel később készült el.

A $\text{\TeX}$ verziószáma 3.1415926, és konvergál a π -hez. Alapelv, hogy a régen megírt dokumentumok is működjenek az újabb rendszerekkel, ezért a 3-as verziószám óta csak hibajavítást tartalmaz.

A $\text{\TeX}$ valójában programozási nyelv, lényegében bármit meg lehet benne valósítani.

Megalkotta a METAFONT fontleíró rendszert, amellyel görbékkel lehet felépíteni betűket. Jelenleg 2.718281 verziószámú.

A fenti rendszerek ingyenesek, és apróbb megkötésekkel szabadon módosíthatóak.

$\LaTeX$

Az 1980-as évek elején *Leslie Lamport* írt egy $\LaTeX$ nevű makrócsomagot $\TeX$ -hez. A $\LaTeX$ segítségével komolyabb tipográfiai tudás nélkül lehet dokumentumot létrehozni.

$\LaTeX$

Az 1980-as évek elején *Leslie Lamport* írt egy $\LaTeX$ nevű makrócsomagot $\TeX$ -hez. A $\LaTeX$ segítségével komolyabb tipográfiai tudás nélkül lehet dokumentumot létrehozni.

1989-ben létrehozták a $\LaTeX 3$ fejlesztői csoportot, amely egy újabb funkciókat tartalmazó $\LaTeX 3$ rendszert hoz létre.

L^AT_EX

Az 1980-as évek elején *Leslie Lamport* írt egy L^AT_EX nevű makrócsomagot T_EX-hez. A L^AT_EX segítségével komolyabb tipográfiai tudás nélkül lehet dokumentumot létrehozni.

1989-ben létrehozták a L^AT_EX3 fejlesztői csoportot, amely egy újabb funkciókat tartalmazó L^AT_EX3 rendszert hoz létre.

Miután világossá vált, hogy a L^AT_EX3 elkészítése még sok ideig tart, 1994-ben kiadták a L^AT_EX 2_ε dokumentumszedő rendszert. Ez több T_EX alapú rendszer ($\mathcal{A}\mathcal{M}\mathcal{S}$, $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX) előnyeit ötvözi, és általánosan elterjedt.

L^AT_EX

Az 1980-as évek elején *Leslie Lamport* írt egy L^AT_EX nevű makrócsomagot T_EX-hez. A L^AT_EX segítségével komolyabb tipográfiai tudás nélkül lehet dokumentumot létrehozni.

1989-ben létrehozták a L^AT_EX3 fejlesztői csoportot, amely egy újabb funkciókat tartalmazó L^AT_EX3 rendszert hoz létre.

Miután világossá vált, hogy a L^AT_EX3 elkészítése még sok ideig tart, 1994-ben kiadták a L^AT_EX 2_ε dokumentumszedő rendszert. Ez több T_EX alapú rendszer ($\mathcal{A}\mathcal{M}\mathcal{S}$, $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX) előnyeit ötvözi, és általánosan elterjedt.

A félév során a L^AT_EX 2_ε programot ismerjük meg részletesebben.

Tipográfia

Egy könyv kiadásának lépései:

- a szerző megírja a művet, amelyet lektorálnak,
- a tipográfus elkészíti az oldalak kinézeti tervét, esetleg grafikus bevonásával megtervezi a borítót, stb.
- a szedő a tipográfus által megadott módon kieszdi a könyvet,
- a folyamatot a szerkesztő felügyeli.

Tipográfia

Egy könyv kiadásának lépései:

- a szerző megírja a művet, amelyet lektorálnak,
- a tipográfus elkészíti az oldalak kinézetét, esetleg grafikus bevonásával megtervezi a borítót, stb.
- a szedő a tipográfus által megadott módon kizsedi a könyvet,
- a folyamatot a szerkesztő felügyeli.

A mai szövegszerkesztő programokban a felhasználó elvégezheti az összes feladatot egyszerre, még akkor is, ha nem ért a tipográfiához. A szerkesztés közben azt látjuk, ami a végeredmény lesz (WYSIWYG).

Tipográfia

Egy könyv kiadásának lépései:

- a szerző megírja a művet, amelyet lektorálnak,
- a tipográfus elkészíti az oldalak kinézeti tervét, esetleg grafikus bevonásával megtervezi a borítót, stb.
- a szedő a tipográfus által megadott módon kiszedi a könyvet,
- a folyamatot a szerkesztő felügyeli.

A mai szövegszerkesztő programokban a felhasználó elvégezheti az összes feladatot egyszerre, még akkor is, ha nem ért a tipográfiához. A szerkesztés közben azt látjuk, ami a végeredmény lesz (WYSIWYG).

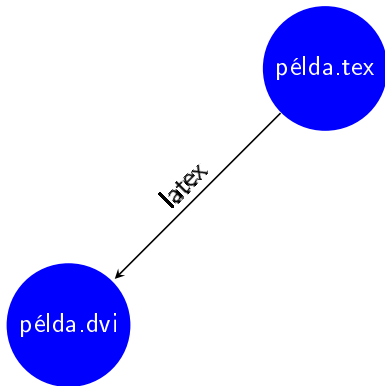
A $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ esetében megmondjuk, hogy mi legyen a cím, fejezet, alfejezet, melyek a kiemelendő szavak, és a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ elvégzi a tipográfiai munkát (WYSIWYM). Természetesen lehetőség van módosítani a kinézeten utólag.

AT_EXfordítás

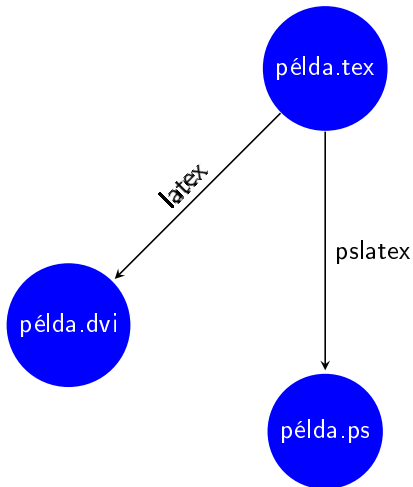


pelda.tex

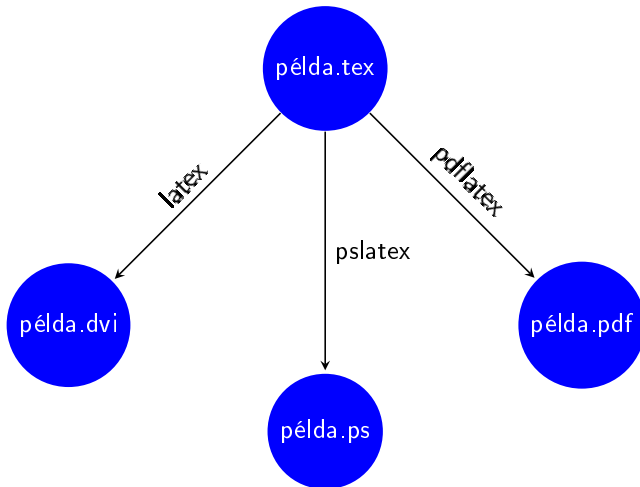
LaTeXfordítás



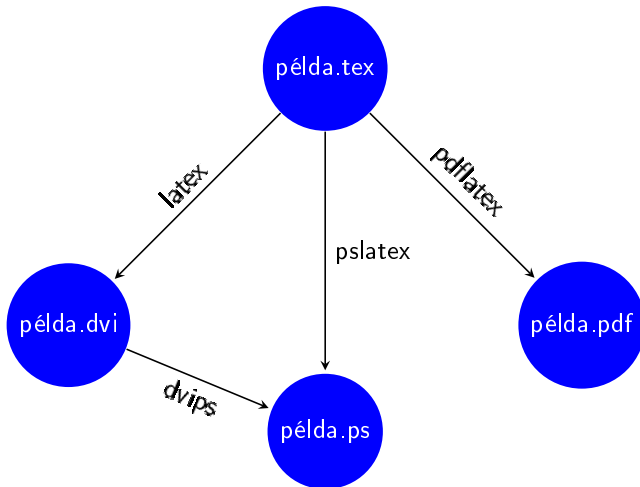
LaTeX fordítás



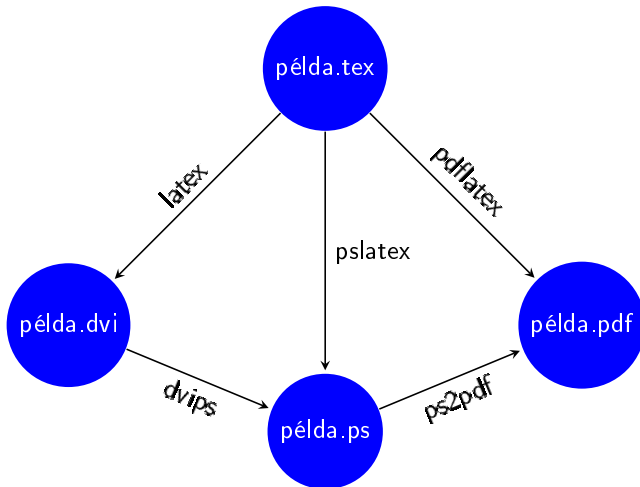
LaTeXfordítás



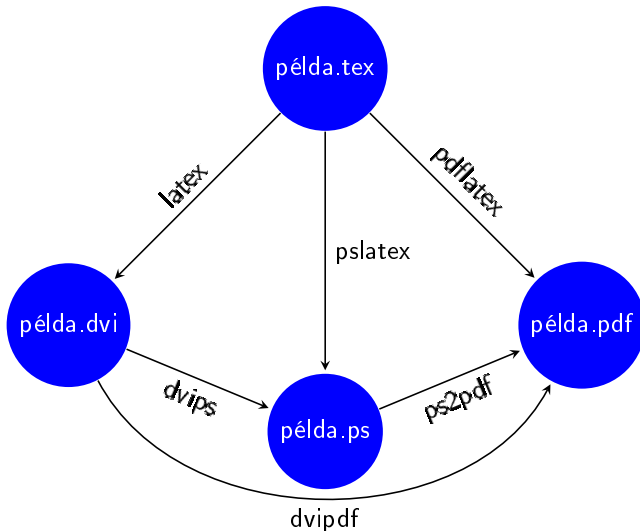
$\text{\textit{L}A\text{\textit{T}}_E\text{\textit{X}}$ fordítás



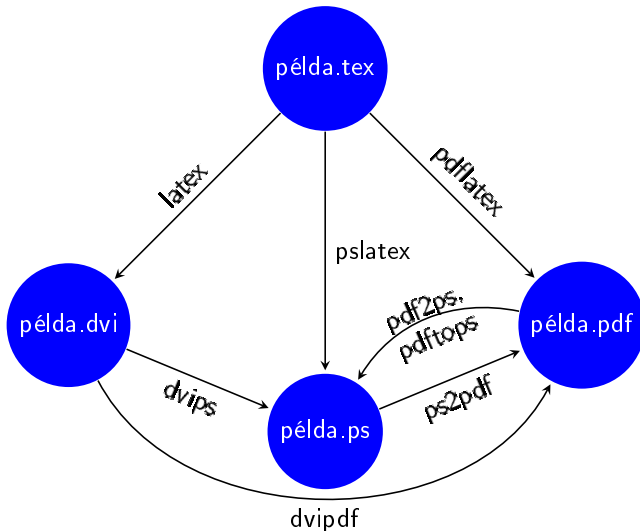
LaTeX fordítás



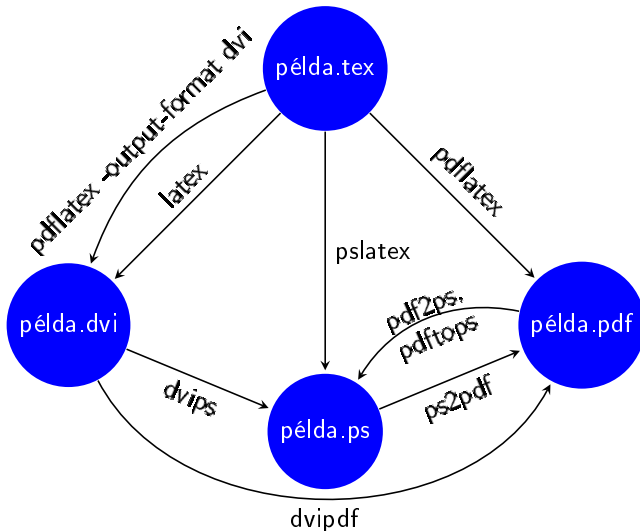
LaTeX fordítás



LaTeXfordítás



LaTeX fordítás



Dokumentum formátumok

DVI

- Device independent file format (általános, nem eszköztől függő fájl formátum)
- Nem tartalmazza a $\text{\TeX}$ fájlban beillesztett képeket és a felhasznált fontokat. A megjelenítésnél rendelkezésre kell állnia az adott gépen mindennek.
- Házit nem fogadunk el ebben.
- Gyorsan fordít, támogatja az *inverse search* technikát.
- Megjelenítők: YAP (Windowsra), KDVI, evince, xdvi (Linuxra)

Dokumentum formátumok

PS

- PostScript (Adobe[©])
- Tartalmaz mindent, ami a megjelenítéshez, nyomtatáshoz kell.
- Programozási nyelv.
- Nyomtatók egy része ismeri, a többinél a nyomtató drivere dekódolja.
- Elterjedt szabvány.
- Megjelenítő: Ghostscriptre épülő GSView/gv (Windowsra, Linuxra), evince (Linuxra)

Dokumentum formátumok

PDF

- Portable Document Format (Adobe[©])
- Tartalmaz mindent, ami a megjelenítéshez, nyomtatáshoz kell. Hordozható.
- Többet tud a PostScriptnél.
- Tartalmazhat nyomtatási útmutatást, kulcsszavakat a kereséshez, indexeléshez, interaktív hiperlinkeket, animációt, videót, stb.
- Elterjedt szabvány.
- Beilleszthetünk *jpeg*, *png*, *pdf* képeket.
- Ebben kérjük a házi feladatot.
- Megjelenítők: Acrobat Adobe Reader/acroread (Windowsra, Linuxra), evince, kpdf, xpdf (Linuxra)

Fordítás

pdflatex proba.tex

Fordítás

```
pdflatex proba.tex
```

Hiba nélküli fordítás:

...

```
fb></usr/share/texmf-texlive/fonts/type1/bluesky/ams/msbm10.pfb>
```

Output written on proba.pdf (3 pages, 98140 bytes).

Transcript written on proba.log.

Fordítás

```
pdflatex proba.tex
```

Hiba nélküli fordítás:

```
...
```

```
fb></usr/share/texmf-texlive/fonts/type1/bluesky/ams/msbm10.pfb>
```

```
Output written on proba.pdf (3 pages, 98140 bytes).
```

```
Transcript written on proba.log.
```

Hibás fordítás:

```
! Undefined control sequence.
```

```
l.31 Li\gat
```

```
úrak: Puff! Oda a maffia mafla fia!
```

```
?
```

Fordítás

```
pdflatex proba.tex
```

Hiba nélküli fordítás:

```
...
```

```
fb></usr/share/texmf-texlive/fonts/type1/bluesky/ams/msbm10.pfb>
```

```
Output written on proba.pdf (3 pages, 98140 bytes).
```

```
Transcript written on proba.log.
```

Hibás fordítás:

```
! Undefined control sequence.
```

```
l.31 Li\gat
```

```
úrák: Puff! Oda a maffia mafla fia!
```

```
?
```

$q + ENTER \Rightarrow$ hibákat figyelmen kívül hagyva adjon egy kimenetet,

$x + ENTER \Rightarrow$ fordítás leállítása

Szövegszerkesztők

Windows alatt a MikTeX programcsomag telepítése után (www.miktex.org) tudunk $\text{\LaTeX}$ fájlokat fordítani. Az alábbi programokkal könnyebbé válik a források szerkesztése:

- **TeXnicCenter**: ingyenes, elterjedt
- **WinEdt**: 30 napig ingyenes, elterjedt
- **WinShell**: ingyenes
- **TexMaker**: ingyenes

Szövegszerkesztők

Windows alatt a MikTeX programcsomag telepítése után (www.miktex.org) tudunk $\text{\LaTeX}$ fájlokat fordítani. Az alábbi programokkal könnyebbé válik a források szerkesztése:

- **TeXnicCenter**: ingyenes, elterjedt
- **WinEdt**: 30 napig ingyenes, elterjedt
- **WinShell**: ingyenes
- **TexMaker**: ingyenes

Linux alatt TeXlive kezdetű csomagok telepítése után működik a fordítás. $\text{\LaTeX}$ szövegszerkesztők:

- **Kile**: elterjedt, ezt használjuk gyakorlaton
- **Lyx**: a kimenethez hasonló formában szerkeszthetünk
- **Winefish**: végtelenül egyszerű
- **TexMaker**: Kile-hoz hasonló elrendezésű

Szövegszerkesztők

Windows alatt a MikTeX programcsomag telepítése után (www.miktex.org) tudunk $\text{\LaTeX}$ fájlokat fordítani. Az alábbi programokkal könnyebbé válik a források szerkesztése:

- **TeXnicCenter**: ingyenes, elterjedt
- **WinEdt**: 30 napig ingyenes, elterjedt
- **WinShell**: ingyenes
- **TexMaker**: ingyenes

Linux alatt TeXlive kezdetű csomagok telepítése után működik a fordítás. $\text{\LaTeX}$ szövegszerkesztők:

- **Kile**: elterjedt, ezt használjuk gyakorlaton
- **Lyx**: a kimenethez hasonló formában szerkeszthetünk
- **Winefish**: végtelenül egyszerű
- **TexMaker**: Kile-hoz hasonló elrendezésű

Természetesen a „nagy” szövegszerkesztőknek, mint például a **Vim**nek és az **Emacs**-nek is van $\text{\LaTeX}$ et támogató kiegészítése.

Sortörés, új paragrafus

egy space	⇒	egy üres hely
több space	⇒	egy üres hely
_□	⇒	egy új üres hely
tilda (~)	⇒	törhetetlen szóköz, például VIII.~Henrik

Sortörés, új paragrafus

egy space	⇒	egy üres hely
több space	⇒	egy üres hely
_□	⇒	egy új üres hely
tilda (~)	⇒	törhetetlen szóköz, például VIII.~Henrik
egy enter	⇒	egy space
több enter	⇒	új paragrafus

Sortörés, új paragrafus

egy space	⇒	egy üres hely
több space	⇒	egy üres hely
_	⇒	egy új üres hely
tilda (~)	⇒	törhetetlen szóköz, például VIII.~Henrik
egy enter	⇒	egy space
több enter	⇒	új paragrafus
%	⇒	komment a sor végéig

Sortörés, új paragrafus

egy space	⇒	egy üres hely
több space	⇒	egy üres hely
_	⇒	egy új üres hely
tilda (~)	⇒	törhetetlen szóköz, például VIII.~Henrik
egy enter	⇒	egy space
több enter	⇒	új paragrafus
%	⇒	komment a sor végéig

Sortörést kényszeríthetünk ki \\ paranccsal, ám ha lehet, akkor inkább kezdjünk új bekezdést.

Parancsok szerkezete

A parancsok \ karakterrel kezdődnek, a csoportosításhoz a { } kapcsos zárójeleket használjuk.

Parancsok szerkezete

A parancsok `\` karakterrel kezdődnek, a csoportosításhoz a `{ }` kapcsos zárójeleket használjuk.

Például `\sqrt{2}` eredménye

$$\sqrt{2},$$

Parancsok szerkezete

A parancsok `\` karakterrel kezdődnek, a csoportosításhoz a `{ }` kapcsos zárójeleket használjuk.

Például `\sqrt{2}` eredménye

$$\sqrt{2},$$

`\sqrt{\sqrt{2}+1}` eredménye

$$\sqrt{\sqrt{2} + 1}.$$

Parancsok szerkezete

A parancsok `\` karakterrel kezdődnek, a csoportosításhoz a `{ }` kapcsos zárójeleket használjuk.

Például `\sqrt{2}` eredménye

$$\sqrt{2},$$

`\sqrt{\sqrt{2}+1}` eredménye

$$\sqrt{\sqrt{2} + 1}.$$

Egy parancsnaK lehetnek opciói is, például `\sqrt[3]{2}` eredménye

$$\sqrt[3]{2}.$$

Parancsok szerkezete

A parancsok `\` karakterrel kezdődnek, a csoportosításhoz a `{ }` kapcsos zárójeleket használjuk.

Például `\sqrt{2}` eredménye

$$\sqrt{2},$$

`\sqrt{\sqrt{2}+1}` eredménye

$$\sqrt{\sqrt{2} + 1}.$$

Egy parancsnaK lehetnek opciói is, például `\sqrt[3]{2}` eredménye

$$\sqrt[3]{2}.$$

Elképzehető több paraméter is, például `\frac{12}{23}` eredménye

$$\frac{12}{23}.$$

Parancsok szerkezete

A kapcsos zárójel lehangyása esetén 1 karaktert vesz csak figyelembe.
Például teljes pompában így néz ki: $\{23\}^{\{34\}}$ eredménye

$$23^{34},$$

Parancsok szerkezete

A kapcsos zárójel lehangyása esetén 1 karaktert vesz csak figyelembe.
Például teljes pompában így néz ki: {23}^{34} eredménye

$$23^{34},$$

ugyanaz hibásan: 23^34 eredménye

$$23^34,$$

Parancsok szerkezete

A kapcsos zárójel lehangyása esetén 1 karaktert vesz csak figyelembe.
Például teljes pompában így néz ki: $\{23\}^{\{34\}}$ eredménye

$$23^{34},$$

ugyanaz hibásan: 23^34 eredménye

$$23^34,$$

illetve jól, de rövidebben: $23^{\{34\}}$ eredménye

$$23^{34}.$$

Parancsok szerkezete

A kapcsos zárójel lehangyása esetén 1 karaktert vesz csak figyelembe.
Például teljes pompában így néz ki: $\{23\}^{\{34\}}$ eredménye

$$23^{34},$$

ugyanaz hibásan: 23^34 eredménye

$$23^34,$$

illetve jól, de rövidebben: $23^{\{34\}}$ eredménye

$$23^{34}.$$

A $\frac{a}{b} + \frac{1}{2}$ eredménye:

$$\frac{a}{b} + \frac{1}{2}.$$

Parancsok szerkezete

A kapcsos zárójel lehangyása esetén 1 karaktert vesz csak figyelembe.
Például teljes pompában így néz ki: $\{23\}^{\{34\}}$ eredménye

$$23^{34},$$

ugyanaz hibásan: 23^34 eredménye

$$23^34,$$

illetve jól, de rövidebben: $23^{\{34\}}$ eredménye

$$23^{34}.$$

A $\frac{a}{b} + \frac{1}{2}$ eredménye:

$$\frac{a}{b} + \frac{1}{2}.$$

Általánosn: $\backslash\text{parancsnév}[\text{esetleges opciók}]\{\text{esetleges paraméterek}\}\dots$

L^AT_EX fájlok szerkezete

```
\documentclass{a dokumentumosztály típusa, pl. article}  
  \usepackage[T1]{fontenc}  
  \usepackage[utf8]{inputenc}  
  \usepackage[magyar]{babel}  
  ...  
\begin{document}  
  Szöveg, lényegi rész, képletek.  
\end{document}
```

} preambulum

Speciális jelentésű karakterek

/
{
}
%
\$
&

_
>
~

Speciális jelentésű karakterek

<code>\</code>	<code>\$\backslash\$</code> , <code>\textbackslash</code>
<code>{</code>	<code>\{</code>
<code>}</code>	<code>\}</code>
<code>%</code>	<code>\%</code>
<code>\$</code>	<code>\\$</code>
<code>&</code>	<code>\&</code>
<code>#</code>	<code>\#</code>
<code>^</code>	<code>\^{}{}</code> , <code>\textasciicircum (^)</code>
<code>~</code>	<code>\$\sim\$</code> , <code>\textasciitilde (~)</code> , <code>\~{}{}</code> (<code>~</code>)