

**МИНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**UKRAJNA OKTATÁSI ÉS TUDOMÁNYOS MINISZTERIUMA**

**Закарпатський угорський інститут імені Ференца Ракоці II**  
**II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola**

**Кафедра математики та інформатики**  
**Matematika és Informatika Tanszék**

dr. Kucsinka Katalin, Papp Gabriella

**Практика з виготовлення наукових текстів та засобів наочності з інформатики / Informatikai tudományos szövegek és szemléltető eszközök készítésének gyakorlata**

# МЕТОДИЧНІ ПОРАДИ ТА ЗАВДАННЯ ЩОДО ПРОХОДЖЕННЯ ПРАКТИКИ

Перший (бакалаврський)  
(ступінь вищої освіти)

|                                                                               |
|-------------------------------------------------------------------------------|
| А Освіта/Педагогіка / A Oktatás/Pedagógia                                     |
| A4.09 Середня освіта (Інформатика) / A4.09 Középiskolai oktatás (Informatika) |
| Інформатика<br>(освітня програма)                                             |
| Informatika<br>(Képzési program)                                              |



Beregszász, 2025

Методичні поради та завдання щодо проходження практики з освітнього компонента « Практика з виготовлення наукових текстів та засобів наочності з інформатики » для здобувачів вищої освіти першого (бакалаврського) рівня, галузь знань: А Освіта/Педагогіка, А4.09 Середня освіта (Інформатика)б освітня програма: «Інформатика» / Укладач: Кучінка К.Й. Берегове: Закарпатський угорський інститут ім. Ференца Ракоці II, 2025. 53 с.

Затверджено до використання у навчальному процесі  
на засіданні кафедри математики та інформатики ЗУІ ім. Ф. Ракоці II  
(протокол № 10 від «23 » червня 2025 року)

Розглянуто та рекомендовано Радою із забезпечення якості освіти  
Закарпатського угорського інституту ім. Ф.Ракоці II  
(протокол № 10 від «23 » червня 2025 року)

Рекомендовано до видання в електронній формі (PDF)  
рішенням Вченої ради Закарпатського угорського інституту ім. Ф.Ракоці II  
(протокол № 8 від «28» серпня 2025 року)

Укладач:

*Каталін Кучінка* – кандидат фіз.-мат. наук, доцент, завідувач кафедри математики та інформатики Закарпатського угорського інституту ім. Ференца Ракоці II

Рецензенти:

*Ганна Сливка-Тилищак* – доктор фіз.-мат. наук, завідувач кафедри теорії ймовірностей та математичної статистики УжНУ

*Адам Доровці* – доктор філософії, старший викладач кафедри математики та інформатики Закарпатського угорського інституту ім. Ференца Ракоці II

повідальні за випуск:

*Каталін КУЧІНКА* – завідувач кафедри математики та інформатики Закарпатського угорського інституту імені Ференца Ракоці II, доцент, кандидат фізико-математичних наук

*Олександр ДОБОШ* – начальник Видавничого відділу ЗУІ ім. Ф.Ракоці II

За зміст методичних вказівок відповідальність несе розробник.

**Видавництво:** Закарпатський угорський інститут імені Ференца Ракоці II (адреса: пл. Кошута 6, м. Берегове, 90202. Електронна пошта: [foiskola@kmf.uz.ua](mailto:foiskola@kmf.uz.ua))

© Каталін Кучінка, 2025  
© Кафедра математики та інформатики ЗУІ ім. Ф.Ракоці II, 2025

Módszertani *ajánlások* és feladatok az „Informatikai tudományos szövegek és szemléltető eszközök készítésének gyakorlata” című oktatási komponenshez az első (alapképzés) szintű felsőoktatási hallgatók számára, tudományterület: 01 Neveléstudomány/Pedagógia, 014.09 Középfokú oktatás (Informatika), képzési program: „Informatika” / Összeállította: Kucsinka K. J. – Beregszász: II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola, 2025. – 53 p.

Az oktatási folyamatban történő felhasználását jóváhagyta a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola Matematika és Informatika Tanszéke  
(2025.06.23. , 10 számú jegyzőkönyv).

Megjelentetésre javasolta a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola  
Oktatási és Módszertani Tanácsa  
(2025.augusztus 26., 7 . számú jegyzőkönyv).

Elektronikus formában (PDF fájlformátumban) történő kiadásra javasolta  
a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola Tudományos Tanácsa  
(2025. augusztus 28., 8. számú jegyzőkönyv).

Kiadásra előkészítette a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola  
Matematika és Informatika Tanszéke, valamint Kiadói Részlege.

A módszertani útmutató kidolgozója:

*KUCSINKA Katalin* – PhD, a fizikai és matematikai tudományok kandidátusa, docens, a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola Matematika és Informatika Tanszékének tanszékvezetője és docense

Szakmai lektorok:

*Slyvka-Tylyshchak Ganna* – fizika-matematika tudományok doktora, a Valószínűességszámítás és Matematikai Statisztika Tanszék vezetője, Ungvári Nemzeti Egyetem

*Daróci Ádám*– PhD, a Matematika és Informatika Tanszék vezető tanára, II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola

A kiadásért felelnek:

*KUCSINKA Katalin* – PhD, a fizikai és matematikai tudományok kandidátusa, a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola Matematika és Informatika Tanszékének tanszékvezetője és docense

*DOBOS Sándor* – a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola Kiadói Részlegének vezetője

A segédlet tartalmáért kizárólag a módszertani útmutató kidolgozója felel.

**Kiadó:** II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola (cím: 90 202, Beregszász, Kossuth tér 6. E-mail: [foiskola@kmf.uz.ua](mailto:foiskola@kmf.uz.ua))

© Kucsinka Katalin, 2025

© A II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola Matematika és Informatika Tanszéke, 2025

© Кафедра математики та інформатики ЗУІ ім. Ф.Ракоці II, 2024

# Tartalomjegyzék

|                                                                                                              |           |
|--------------------------------------------------------------------------------------------------------------|-----------|
| <b>Bevezetés</b>                                                                                             | <b>7</b>  |
| <b>A tantrágy leírása</b>                                                                                    | <b>8</b>  |
| <b>1. óra – Bevezetés a LaTeX-be</b>                                                                         | <b>9</b>  |
| <b>2. óra – Alapstruktúra és preambulum (bővített jegyzet)</b>                                               | <b>10</b> |
| 2.1. Dokumentumosztályok részletes ismertetése . . . . .                                                     | 10        |
| 2.2. A preambulum részletes ismertetése . . . . .                                                            | 11        |
| 2.3. További példák . . . . .                                                                                | 11        |
| 2.4. Feladatok . . . . .                                                                                     | 12        |
| <b>3. óra – Szövegformázás, listák és bekezdések</b>                                                         | <b>13</b> |
| 3.1. Bekezdések és sortörések . . . . .                                                                      | 13        |
| 3.2. Szövegstílusok . . . . .                                                                                | 13        |
| 3.3. Listák . . . . .                                                                                        | 13        |
| 3.4. Idézetek és lábjegyzetek . . . . .                                                                      | 14        |
| 3.5. További példák . . . . .                                                                                | 14        |
| 3.6. Feladatok . . . . .                                                                                     | 14        |
| <b>4. óra – Matematikai környezetek és képletek</b>                                                          | <b>15</b> |
| 4.1. Inline és display mód . . . . .                                                                         | 15        |
| 4.2. Egyenletek számozása és hivatkozások . . . . .                                                          | 15        |
| 4.3. Gyakori struktúrák . . . . .                                                                            | 15        |
| 4.4. Egyenletrendszerek és igazítás . . . . .                                                                | 16        |
| 4.5. További példák és tippek . . . . .                                                                      | 16        |
| 4.6. Feladatok . . . . .                                                                                     | 16        |
| <b>5. óra – Táblázatok készítése és testreszabása</b>                                                        | <b>17</b> |
| 5.1. A tabular környezet . . . . .                                                                           | 17        |
| 5.2. Hosszú táblázatok . . . . .                                                                             | 17        |
| 5.3. Táblázatok igazítása és szélességek . . . . .                                                           | 17        |
| 5.4. További példák . . . . .                                                                                | 17        |
| 5.5. Feladatok . . . . .                                                                                     | 18        |
| <b>6. óra – Képek beillesztése és rajzolás TikZ-zel</b>                                                      | <b>19</b> |
| 6.1. Képek beillesztése <code>graphicx</code> -szel . . . . .                                                | 19        |
| 6.2. Több kép egy ábrában – <code>subfigure</code> . . . . .                                                 | 19        |
| 6.3. Rajzolás TikZ-zel . . . . .                                                                             | 19        |
| 6.4. Feladatok . . . . .                                                                                     | 20        |
| <b>7. óra – Kereszthivatkozások és bibliográfia kezelése</b>                                                 | <b>21</b> |
| 7.1. Címkék és hivatkozások . . . . .                                                                        | 21        |
| 7.2. Hivatkozási stílusok testreszabása . . . . .                                                            | 21        |
| 7.3. BibTeX adatbázis felépítése . . . . .                                                                   | 21        |
| 7.4. Feladatok . . . . .                                                                                     | 22        |
| <b>8. óra – Kódbeillesztés, algoritmusok és hiperhivatkozások</b>                                            | <b>23</b> |
| 8.1. Forráskód beillesztése a <code>listings</code> csomaggal . . . . .                                      | 23        |
| 8.2. Algoritmusok dokumentálása <code>algorithmicx</code> és <code>algpseudocode</code> segítségével . . . . | 23        |
| 8.3. Hiperhivatkozások testreszabása <code>hyperref</code> -fel . . . . .                                    | 24        |

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| 8.4. Feladatok . . . . .                                               | 24        |
| <b>9. óra – Szakdolgozat és jelentés felépítése</b>                    | <b>25</b> |
| 9.1. A dolgozat fő részei . . . . .                                    | 25        |
| 9.2. Sorközök és betűtípusok beállítása . . . . .                      | 26        |
| 9.3. Fejezetcímek és tartalomjegyzék testreszabása . . . . .           | 26        |
| 9.4. Feladatok . . . . .                                               | 26        |
| <b>10.óra – Beamer prezentációk</b>                                    | <b>27</b> |
| 10.1. Prezentációk felépítése . . . . .                                | 27        |
| 10.2. Animációk és overlayek . . . . .                                 | 27        |
| 10.3. Vizualizációk . . . . .                                          | 27        |
| 10.4. Feladatok . . . . .                                              | 28        |
| <b>11.óra – Oldalbeállítások és tipográfiai stílusok</b>               | <b>29</b> |
| 11.1. Margók és oldalméretetek . . . . .                               | 29        |
| 11.2. Fejléc és lábléc testreszabása . . . . .                         | 29        |
| 11.3. Sorköz és bekezdésköz . . . . .                                  | 29        |
| 11.4. Fejezetcímek formázása . . . . .                                 | 29        |
| 11.5. Betűtípusok és tipográfia . . . . .                              | 30        |
| 11.6. Feladatok . . . . .                                              | 30        |
| <b>12. óra – Tételek, definíciók és bizonyítások</b>                   | <b>31</b> |
| 12.1. Új környezetek létrehozása . . . . .                             | 31        |
| 12.2. Theorem-stílusok . . . . .                                       | 31        |
| 12.3. Bizonyítások környezete . . . . .                                | 31        |
| 12.4. Feladatok . . . . .                                              | 31        |
| <b>13.óra – Kódlisák testreszabása és használata</b>                   | <b>33</b> |
| 13.1. Globális beállítások a <code>listings</code> csomagnál . . . . . | 33        |
| 13.2. Sorok kiemelése és háttérszínek . . . . .                        | 33        |
| 13.3. Alternatíva: <code>minted</code> csomag . . . . .                | 33        |
| 13.4. Kód mellékletekben . . . . .                                     | 34        |
| 13.5. Feladatok . . . . .                                              | 34        |
| <b>14.óra – Algoritmusleírások és pszeudokód</b>                       | <b>35</b> |
| 14.1. Az <code>algpseudocode</code> alapjai . . . . .                  | 35        |
| 14.2. Algoritmusok listázása . . . . .                                 | 35        |
| 14.3. Stílusok és testreszabás . . . . .                               | 35        |
| 14.4. Feladatok . . . . .                                              | 35        |
| <b>15.óra – Diagramok és vizualizációk TikZ-zel</b>                    | <b>37</b> |
| 15.1. TikZ alapjai . . . . .                                           | 37        |
| 15.2. Koordináta-rendszerek és görbék . . . . .                        | 37        |
| 15.3. PGFPlots – grafikonok készítése . . . . .                        | 37        |
| 15.4. Feladatok . . . . .                                              | 38        |
| <b>16.óra – Tárgymutatók és szószedetek készítése</b>                  | <b>39</b> |
| 16.1. Indexkészítés lépésről lépésre . . . . .                         | 39        |
| 16.2. Szószedetek és rövidítések . . . . .                             | 39        |
| 16.3. Haladó beállítások . . . . .                                     | 40        |
| 16.4. Feladatok . . . . .                                              | 40        |
| 16.5. óra – Haladó irodalomkezelés és hivatkozási stílusok . . . . .   | 41        |

|                                                                |           |
|----------------------------------------------------------------|-----------|
| 16.6. Több bibliográfia egy dokumentumban . . . . .            | 41        |
| 16.7. Hivatkozási parancsok . . . . .                          | 41        |
| 16.8. Bibliográfiai rekordok szerkesztése . . . . .            | 41        |
| 16.9. Makrók és formátumok testreszabása . . . . .             | 42        |
| 16.10.Feladatok . . . . .                                      | 42        |
| <b>17.óra – Haladó irodalomkezelés és hivatkozási stílusok</b> | <b>43</b> |
| 17.1. Több bibliográfia egy dokumentumban . . . . .            | 43        |
| 17.2. Hivatkozási parancsok . . . . .                          | 43        |
| 17.3. Bibliográfiai rekordok szerkesztése . . . . .            | 43        |
| 17.4. Makrók és formátumok testreszabása . . . . .             | 44        |
| 17.5. Feladatok . . . . .                                      | 44        |
| <b>18.óra – Git, Overleaf és kollaboráció</b>                  | <b>45</b> |
| 18.1. Git alapfogalmak . . . . .                               | 45        |
| 18.2. Gyakorlati munkamenet . . . . .                          | 45        |
| 18.3. .gitignore és LaTeX projektek . . . . .                  | 45        |
| 18.4. Overleaf és Git integráció . . . . .                     | 46        |
| 18.5. Feladatok . . . . .                                      | 46        |
| <b>19. óra – Haladó Beamer funkciók és testreszabás</b>        | <b>47</b> |
| 19.1. Téma testreszabása . . . . .                             | 47        |
| 19.2. Összetett overlayek . . . . .                            | 47        |
| 19.3. Interaktív elemek . . . . .                              | 47        |
| 19.4. PGFPlots integráció . . . . .                            | 47        |
| 19.5. Feladatok . . . . .                                      | 48        |
| <b>20.óra – Végző projekt összeállítása és csomagolása</b>     | <b>49</b> |
| <b>Irodalomjegyzék</b>                                         | <b>51</b> |

## Bevezetés

A kurzus célja, hogy megismertesse a hallgatókat a LaTeX szövegszerkesztő rendszer használatával, amely napjainkban a tudományos és oktatási szövegek készítésének egyik legfontosabb eszköze. A LaTeX különösen alkalmas matematikai képletek, táblázatok, grafikonok és ábrák kezelésére, ezért a természettudományos és informatikai területeken gyakorlatilag szabvánnyá vált. A kurzus során a hallgatók megtapasztalhatják, hogyan lehet pontos, áttekinthető és tipográfailag igényes dokumentumokat létrehozni.

A képzés az informatika tanár szak második évfolyamos hallgatói számára készült, és alapvető gyakorlati ismereteket nyújt a LaTeX rendszer használatához. A tananyag lépésről lépésre vezeti végig a hallgatókat a dokumentumkészítés folyamatán: az egyszerű szövegformázástól eljutnak a bonyolultabb matematikai környezetek, táblázatok és szemléltető ábrák összeállításáig. A hangsúly a gyakorlati munkán van, így a hallgatók aktívan sajátítják el a tudományos szövegek előállításának módszereit.

A feladatok az Overleaf online felületén történnek, amely lehetőséget biztosít a dokumentumok azonnali kipróbálására és szerkesztésére. A kurzus a tavaszi félévben kerül megrendezésre, a tanulási folyamatot rendszeres gyakorlati feladatok kísérik. A félév végén a megszerzett ismeretekről és készségekről zárthelyi jellegű vizsga formájában kell számot adni.

Ez a segédlet a kurzushoz kapcsolódó módszertani háttértámogatást nyújtja, és hozzájárul ahhoz, hogy a hallgatók biztos alapokra építve sajátítsák el a tudományos szövegalkotás korszerű technikáit.

## A tantrágy leírása

**A tantárgy célja:** A LaTeX-2e kiadói rendszer használatának alapjainak elsajátítása, amely később alkalmazható tudományos publikációk, valamint tudományos-módszertani és oktatási kiadványok előkészítésében.

**Feladatok:** A hallgatókban olyan készségek és igények kialakítása, amelyek lehetővé teszik a korszerű számítógépes technológiák alkalmazását tanulmányi, tudományos és üzleti publikációk szerkesztésében, továbbá az információk weboldalakon és prezentációkban való megjelenítésére való előkészítésben. A kurzus során a hallgató által elsajátítandó általános és szakmai kompetenciák: ZK8 – Tájékozódás az információs térben; információk keresése, elemzése és feldolgozása különböző forrásokból; digitális erőforrások és technológiák hatékony alkalmazása az oktatási folyamatban; IKT-használat etikai-jogi normák szerint.

ZK11 – Absztrakt gondolkodás, elemzés és szintézis; új ötletek generálása; problémamegoldás; tudás alkalmazása gyakorlati helyzetekben.

ZK13 – Tanulásra való képesség és a korszerű ismeretek elsajátítása.

FK2 – Tanítás államnyelven; a tanulók szaknyelvi kommunikációs készségeinek fejlesztése informatikából; az oktatás biztosítása az intézmény nyelvi környezetének sajátosságaihoz, illetve a nemzeti kisebbségekhez igazodva.

FK18 – Szöveges, numerikus, grafikus, hang- és videóinformáció digitális előállítása és feldolgozása.

A program eredményei:

PRN22 – A matematika és az informatika alapvető témaköreinek ismeretének bemutatása.

PRN23 – Informatikai feladatok megoldása különböző nehézségi szinteken, racionális módszerek alkalmazásával.

PRN29 – Az egyes tantárgyak és integrált kurzusok tartalmának felhasználása a tanulók kulcskompetenciáinak és átfogó készségeinek fejlesztésére, valamint a tanulók gyakorlati alkalmazásra való tanítása.

## 1. óra – Bevezetés a LaTeX-be

### Cél

Ez a jegyzet informatikatanár-szakos hallgatóknak készült, akik a következő tanévben LaTeX-ben írják évfolyam- vagy szakdolgozatukat. A LaTeX egy magas minőségű szedőrendszer, amelyet kifejezetten tudományos és műszaki dokumentumok készítésére terveztek. A projekt hivatalos honlapja szerint a LaTeX a tudományos publikációk de facto szabványa<sup>1</sup>. Főbb jellemzői közé tartozik a nagy dokumentumok kezelése, kereszthivatkozások, táblázatok és ábrák, komplex matematikai képletek, valamint automatikus bibliográfia- és indexgenerálás.

### Miért LaTeX?

- A LaTeX nem WYSIWYG szerkesztő, hanem egyszerű szövegfájlban tárolt forrás, parancsokkal. A forrást egy TeX motor fordítja PDF-ré, így a szerző a tartalomra koncentrálhat.
- Előnyös a bonyolult matematikai képletek, táblázatok és hivatkozások kezelésében. A dokumentum tartalma és stílusa jól elválasztható.
- Többnyelvű szedéshez speciális csomagok (pl. `magyar.ldf`) állnak rendelkezésre.

### Telepítés és eszközök

- **TeX Live** / **MiKTeX**: teljes TeX-disztribúciók.
- **Overleaf**: online szerkesztő, nem igényel telepítést.
- **Texmaker** / **TeXstudio**: integrált szerkesztők.

### A magyar nyelv beállítása

- A magyar szövegek helyes elválasztásához a MagyarLaTeX csomag részeként elérhető `magyar.ldf` modul (1.5-ös verzió) szükséges. A fájlt a forrás mappájába célszerű tenni.
- A `uni8.sty` kényelmi stílus automatikusan beállítja a betűtípusokat és elválasztást pdf<sub>l</sub>atex és lua<sub>l</sub>atex használatakor.

### Gyakorló feladatok

1. Telepíts egy TeX-disztribúciót (vagy használj Overleafet), hozz létre `main.tex` állományt.

2. Írd meg az első dokumentumot:

```
\documentclass{article}
\usepackage[magyar]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\title{Első LaTeX dokumentum}
\author{Neved}
\date{\today}
\begin{document}
\maketitle
Ez az első LaTeX dokumentumom.
\end{document}
```

3. Gyakorold a magyar ékezetek használatát (ő, ű).

---

<sup>1</sup>LaTeX Project: <https://www.latex-project.org/>

## 2. óra – Alapstruktúra és preambulum (bővített jegyzet)

### Cél

A második órán részletesen áttekintjük, hogyan épül fel egy  $\text{\LaTeX}$  dokumentum. Elsajátítjuk a különböző dokumentumosztályok használatát, megtanuljuk, hogyan kell a preambulomot személyre szabni, és áttekintjük, milyen csomagokat érdemes betölteni a magyar nyelvű szedéshez. A részletes ismertetés célja, hogy képesek legyünk tudatosan megválasztani a megfelelő osztályt és beállítani az alapvető paramétereket.

### 2.1. Dokumentumosztályok részletes ismertetése

A  $\text{\LaTeX}$  dokumentum felépítésének kiindulópontja a dokumentumosztály. A dokumentumosztály határozza meg a dokumentum alapvető struktúráját, a fejezetek elrendezését, az alapértelmezett betűméretet és a számozási rendszert. Három alapvető osztályt különböztetünk meg, azonban ezekhez rengeteg opció tartozik.

#### article

Az **article** osztály a leggyakrabban használt. Ide tartoznak a rövid publikációk, jegyzetek, házi feladatok, laboratóriumi beszámolók. Ebben az osztályban a dokumentum legfelső hierarchikus szintje a `\section`, és nincs lehetőség `\chapter` használatára. Az **twocolumn** opció segítségével két hasábos tördelést érhetünk el, ami tudományos cikkekben gyakori. Az **10pt**, **11pt** és **12pt** betűméret opciók a sor- és bekezdésköz változtatására szolgálnak. Ha kétszeres sortávolságot szeretnénk, használhatjuk a **setspace** csomagot.

#### report

A **report** osztályt hosszabb dokumentumokhoz, például szakdolgozatokhoz, projektbeszámolókhöz használjuk. Ez az osztály már támogatja a `\chapter` parancsot, amely lehetővé teszi fejezetek létrehozását. Az opciók között találjuk az **openright** és **openany** beállítást: előbbi esetén minden fejezet új páratlan oldalon kezdődik (ez könyveknél kötelező), utóbbi rugalmasan kezeli a fejezetek elhelyezkedését. A **report** osztály jól együttműködik a **geometry** csomaggal, amellyel pontosan beállíthatjuk a margókat.

#### book

A **book** osztály a legösszetettebb. Nemcsak fejezeteket, hanem részeket is kezel (`\part`), és külön mechanizmusa van az előszó (front matter), a fő rész (main matter) és a mellékletek (back matter) elkülönítésére. A bal és jobb oldali lapok margói különbözhetnek, és a könyvszerkezet automatikusan kezeli a címlapokat. Olyan opciókkal rendelkezik, mint a **twoside** (kétoldalas nyomtatás) és a **draft**, utóbbi kijelzi a túl hosszú sorokat és jelzi az oldalak végén lévő kitöltetlen helyet.

### Speciális osztályok

A fenti alaposztályokon kívül vannak speciális, adott kiadókhöz vagy célokhoz igazított osztályok. Például az **memoir** osztály egyesíti a **book**, **report** és **article** tulajdonságait, és rengeteg testreszabási lehetőséget kínál. Az **IEEEtran** osztály az IEEE konferenciák és folyóiratok formátumát valósítja meg, míg az **beamer** osztály prezentációkhoz alkalmas. Mindig a kiírás vagy a kiadó által megkövetelt osztályt használjuk.

## 2.2. A preambulum részletes ismertetése

A preambulum a dokumentum azon része, amely a `\documentclass` deklarációt követően, de a `\begin{document}` előtt helyezkedik el. Itt adhatjuk meg a betöltendő csomagokat, beállíthatjuk a dokumentum globális jellemzőit, és definiálhatunk saját parancsokat.

### Csomagok sorrendje és szerepe

- **Nyelvi csomagok:** a `babel` csomag gondoskodik az elválasztásról és a lokalizált feliratról. Magyar nyelv esetén a `\usepackage[magyar]{babel}` beállítást használjuk. Ez a csomag cseréli a "Chapter" szót "Fejezet"-re, a dátumot magyar formátumra állítja, és megfelelő elválasztási algoritmust alkalmaz.
- **Betűtípusok:** a `fontenc` és `inputenc` csomagok a kódolást állítják be. A T1 kódolás jobban kezeli a közép-európai karaktereket, mint az alapértelmezett OT1.
- **Margók beállítása:** a `geometry` csomaggal nemcsak a margókat (bal, jobb, felső, alsó), hanem a fejléc és lábléc térközeit is konfigurálhatjuk. A `bindingoffset` opcióval könyvkötéshez szükséges plusz teret adhatunk.
- **Fejlécek és láblécek:** a `fancyhdr` csomag rugalmasan beállítható fejléceket és lábléceket biztosít. Definiálhatunk külön fejléceket a páros és páratlan oldalakra, megjeleníthetjük a fejezet címét vagy saját logót illeszthetünk be.
- **Hyperhivatkozások:** a `hyperref` csomag automatikusan kattinthatóvá teszi a belső hivatkozásokat (fejezet-, ábra- és táblázatszámok), és PDF-ben könyvjelzőket generál. Az opciókkal megadható a linkek színe, keretezése, és metaadatokat is beállíthatunk (szerző, cím, kulcsszavak).

### Saját parancsok és környezetek

A preambulumban definiálhatunk rövidítéseket és új környezeteket, amelyek megkönnyítik a dokumentum szerkesztését. Például a hosszú képletek gyakran használt részeit `\newcommand` segítségével rövidíthetjük, a gyakran használt stílusokat pedig `\newenvironment` segítségével egyszerűsíthetjük.

## 2.3. További példák

Az alábbiakban néhány példa forrásrészletet láthatunk, amelyek bemutatják az előző szakaszban tárgyalt beállításokat.

### Címlap és metaadatok

```
\documentclass[12pt,a4paper]{report}
\usepackage[magyar]{babel}
\usepackage[OT2,T1]{fontenc}
\usepackage[utf8]{inputenc}
\title{Szakdolgozat címe}
\author{Hallgató Neve \and Konzulens Neve}
\date{2025. május 30.}
\begin{document}
\maketitle
\end{document}
```

A fenti példában több szerzőt a `\and` kulcsszóval választunk el. A `\date` parancs segítségével tetszőleges szöveget adhatunk meg dátum helyett.

## Fejezetek és margók kezelése

```
\documentclass[11pt]{report}
\usepackage{geometry}
\geometry{left=3cm,right=3cm,top=2.5cm,bottom=2.5cm}
\begin{document}
\chapter{Első fejezet}
Ebben a fejezetben röviden ismertetjük a témát. A megadott margóbeállítások
    biztosítják, hogy a kötésnél elegendő hely maradjon.
\section{Alfejezet}
További részletek.
\end{document}
```

### 2.4. Feladatok

1. Hozz létre különböző dokumentumosztályú fájlokat (`article`, `report`, `book`), és próbáld ki a 10pt, 11pt és 12pt betűméret opciókat. Figyeld meg, hogyan változik a sortávolság és az oldalbeosztás.
2. Készíts el egy saját preambulomot, amely betölti a `babel`, `geometry`, `hyperref` és `fancyhdr` csomagokat. Állíts be egyedi margókat, fejléceket (oldalszám a jobb felső sarokban, fejezetcím a bal felső sarokban), és hozz létre egy címlapot.
3. Definiálj a preambulumban egy `\newcommand{\R}{\mathbb{R}}` parancsot, majd használj matematikai környezetet az új parancs kipróbálására.
4. Írj egy rövid dokumentumot, amelyben saját környezetet definiálsz (pl. `\begin{note} ... \end{note}`) figyelemfelhívó stílushoz. A környezet használjon keretet vagy szürke háttérszínt.

### 3. óra – Szövegformázás, listák és bekezdések

#### Cél

A harmadik órán a szövegformázás finomságait vizsgáljuk meg. Megtanuljuk, hogyan hozhatunk létre bekezdéseket, hogyan használhatjuk a tipográfiai stílusokat (félkövér, dőlt, aláhúzott), hogyan szervezhetünk listákat és hogyan kezelhetjük az idézeteket. Ezek a technikák elengedhetetlenek ahhoz, hogy igényes, jól struktúrált dokumentumokat hozzunk létre.

#### 3.1. Bekezdések és sortörések

A  $\text{\LaTeX}$  a bekezdéseket üres sorral választja el. Ha egy sor végén szeretnénk sort törni új bekezdés nélkül, használhatjuk a `\` parancsot. A sorok végén elhelyezett százalékek (%) segítségével megakadályozhatjuk a felesleges szóközök beszúrását. A bekezdések első sorának behúzását az `\parindent` hossz változtatásával állíthatjuk, a bekezdések közötti függőleges térközt pedig az `\parskip` hosszal szabályozhatjuk.

**Kézi elválasztás és kötőjelek.** Ha egy szó elválasztása problémás, a `\-` parancs beillesztésével jelezhetjük, hogy hol engedélyezzünk elválasztást. Például **rend-szer** a "rend-szer" szót két lehetséges elválasztási hellyel látja el. A nem törhető kötőjelet (`\sim`) használjuk akkor, amikor a kötőjel mögött nem szeretnénk sortörést (pl. 20 %-os).

**Gyakori hibák.** Ne használjunk több szóközt a sorokban a sorköz növelésére; a  $\text{\LaTeX}$  maga szabályozza a szóközök méretét. A felsorolások előtt és után is célszerű üres sort hagyni a megfelelő bekezdéskezelés érdekében.

#### 3.2. Szövegstílusok

A tipográfiai hangsúlyok kiemelik a fontos fogalmakat. A `\textbf{...}` parancs félkövérré, a `\textit{...}` dőltre, míg a `\underline{...}` aláhúzásra formázza a szöveget. Kombinálhatjuk is őket: `\textbf{\textit{kiemelt}}` félkövér és dőlt lesz egyszerre. A betűtípus családját a `\textsf` (talpatlan), `\textrm` (talpas) parancsokkal módosíthatjuk. Ha színes szöveget szeretnénk, a `color` csomag `\textcolor{red}{szöveg}` parancsával határozhatunk meg RGB vagy nevű színeket.

#### Láthatatlan szótagelválasztás

Magyar szövegben előfordul, hogy bizonyos összetett szavakat a  $\text{\LaTeX}$  nem választ el helyesen. A `\allowbreak` paranccsal kijelölhetünk olyan helyeket, ahol a sor törhető, de a szóban nem jelenik meg kötőjel, például "programozás".

#### 3.3. Listák

A felsorolások három alapvető környezettel készíthetők:

- `itemize` – szimbólumos felsorolás. A szimbólumokat a `\item[--]` opcionális paraméterrel testre szabhatjuk.
- `enumerate` – számozott lista. A számozás stílusát a `\label` kulcs és a `\alph`, `\roman` számlálók kombinálásával módosíthatjuk.
- `description` – definíciós lista, ahol az elem címét paraméterként adjuk meg.

**Beágyazott listák.** A listák egymásba ágyazhatók, de minden szint saját behúzással jelenik meg. Ügyeljünk arra, hogy a beágyazott `\item` parancsokat megfelelően egymás alá illesszük, különben nehézkes lesz az olvashatóság.

**Egyéni listajelölések.** Ha saját szimbólumot szeretnénk használni az `itemize` listában, a `\item[\textbullet]` formát alkalmazzuk. Numerikus listák esetén a `\renewcommand{\labelenumi}{\alph{enumi}}` parancs a külső szint elemeit "a)", "b)" formátumra cseréli.

### 3.4. Idézetek és lábjegyzetek

Hosszabb szövegeket idézetkörnyezetbe helyezhetünk a `\begin{quote} ... \end{quote}` paranccsal, amely automatikusan behúzza a szöveget mindkét oldalon. Többszörös idézethez a `quotation` környezetet használjuk, amely új bekezdéseket is enged. A lábjegyzetek a `\footnote{...}` paranccsal hozhatók létre: a lábjegyzet sorszámát a  $\text{\LaTeX}$  automatikusan generálja, és a megfelelő helyre illeszti a szöveg alján.

### 3.5. További példák

Az alábbi kódrészlet bemutatja, hogyan definiálhatunk egy egyedi számozott listát, amelyben az első szint latin betűkkel, a második szint római számlákkal van jelölve:

```
\documentclass{article}
\usepackage{enumitem}
\begin{document}
\renewcommand{\labelenumi}{\alph{enumi}}}
\begin{enumerate}
\item Első elem
\begin{enumerate}[label=\roman*.]
\item Alsó szint első eleme
\item Alsó szint második eleme
\end{enumerate}
\item Második elem
\end{enumerate}
\end{document}
```

Az `enumitem` csomag számos paramétert kínál a listaelemek közötti térköz csökkentésére vagy növelésére.

### 3.6. Feladatok

1. Írj egy rövid esszét három bekezdésben, és próbáld ki a `\parindent` és `\parskip` paraméterek módosítását. Figyeld meg, hogyan változik a bekezdések be- és kihúzása.
2. Készíts egy szöveget, amelyben különböző szövegstílusokat kombinálsz (félkövér, dőlt, aláhúzott, színes). Definiáld saját makrókat a pirossal kiemelt, vastagon szedett szavakhoz.
3. Hozz létre háromszintű listát úgy, hogy az első szint számokkal (1., 2., 3.), a második betűkkel (a), b), c)), a harmadik pedig római számokkal (i., ii., iii.) legyen jelölve. Használd az `enumitem` csomagot a testreszabáshoz.
4. Írj egy bekezdést, amely tartalmaz egy idézetet (`quote` környezet), valamint legalább két lábjegyzetet, amelyek különböző információkat közölnek. Fordítsd le a dokumentumot és ellenőrizd az automatikus sorszámozást.

## 4. óra – Matematikai környezetek és képletek

### Cél

A negyedik órán a  $\text{\LaTeX}$  matematikai környezetének részleteibe merülünk el. Megtanuljuk, hogyan írjuk be és tördeljük a képleteket, hogyan kezeljük az indexeket, felső- és alsó indexeket, törteket, gyököket, mátrixokat, összegeket, integrálokat, valamint hogyan hivatkozzunk az egyenletekre. A matematikai jelölések pontos használata elengedhetetlen a mérnöki és tudományos munkában.

### 4.1. Inline és display mód

A matematikai képleteket két módon helyezhetjük el:

- **Inline mód:** a szövegközi képleteket  $\$ \dots \$$  vagy  $\backslash( \dots \backslash)$  jelek közé írjuk. Ezek kisebb méretűek, és a sor magassága igazodik hozzájuk. Példa:  $a^2 + b^2 = c^2$ .
- **Display mód:** a nagyobb, kiemelt képleteket külön sorba tesszük  $\$ \dots \$$  vagy  $\backslash[ \dots \backslash]$  környezettel. Ezeket a  $\text{\LaTeX}$  középre igazítja, és nagyobb betűmérettel jeleníti meg. Fontos, hogy a kettős dollárjelet preferáltan kerüljük és helyette a  $\backslash[ \dots \backslash]$  formát használjuk.

### 4.2. Egyenletek számozása és hivatkozások

Az `equation` környezet automatikusan sorszámozza a képleteket. A `\label` paranccsal címkét adhatunk egy képletnek, amelyre később `\ref` vagy `\eqref` paranccsal hivatkozhatunk. Az `\eqref` zárójeleket is tesz a szám köré. Példa:

```
\begin{equation}\label{eq:pitagorasz}
a^2 + b^2 = c^2
\end{equation}
```

A Pitagorasz-tétel alakja a `\eqref{eq:pitagorasz}` egyenletben látható.

### 4.3. Gyakori struktúrák

#### Indexek és hatványok

A felső indexet a  $\wedge$  jel után kapcsos zárójelbe írjuk, az alsó indexet  $_$  jel után. Több szintű index is lehetséges:  $a_{i,j}^{(n)}$ .

#### Törtek és gyökök

A `\frac{a}{b}` parancs a törteket formázza. A `\dfrac` nagyobb betűméretet használ inline módban is. Gyököt a `\sqrt[n]{kifejezés}` paranccsal írhatunk. Példák:  $\frac{1}{2}$ ,  $\sqrt{2}$ ,  $\sqrt[3]{x+y}$ .

#### Szumma és integrál

Sokszor használunk összegzést és integrált. A `\sum` és `\int` jelek alsó és felső határai a  $_$  és  $\wedge$  jelek után kapcsos zárójelbe kerülnek. A `display` mód automatikusan a jel alatt-fölött jeleníti meg a határokat:  $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ ,  $\int_0^1 x^2 dx = \frac{1}{3}$ .

## Mátrixok és rendszerek

Mátrixokat a `\begin{matrix}` vagy `pmatrix` környezettel írhatunk. A sorokat választja el, az elemeket pedig `&` tagolja. Példa:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Rendszereket a `cases` környezettel formázhatunk:

$$f(x) = \begin{cases} x^2, & x \geq 0, \\ -x, & x < 0. \end{cases}$$

### 4.4. Egyenletrendszerek és igazítás

Az `align` és `align*` környezetek lehetővé teszik több sorból álló képletek igazítását. Az `align` jel határozza meg, hogy hol legyen az igazítási pont. Az `align*` nem számoz, míg az `align` soronként sorszámoz; ha egy sorhoz nem szeretnénk sorszámot, használjuk az `\nonumber` parancsot. Példa:

```
\begin{align}
f'(x) &= 2x + 3 \\
\int f(x)dx &= x^2 + 3x + C
\end{align}
```

### 4.5. További példák és tippek

**Egyéni makrók.** Ha gyakran használunk bonyolult képletelemeket, érdemes rájuk makrót írni. Például: `\newcommand{\diff}{\mathrm{d}}`, majd integráláskor `\int f(x)\diff x` formában következetesen írjuk a differenciált.

**Különleges karakterek.** A  $\text{\LaTeX}$  számos görög betűt és matematikai szimbólumot ismer. A görög kisbetűk `\alpha`, `\beta`, ..., a nagybetűk `\Gamma`, `\Delta`, stb. A logikai jeleket (`\land`, `\lor`, `\neg`) és halmazműveleti szimbólumokat (`\cup`, `\cap`, `\setminus`) is a képletben használjuk.

**Hárompontos jel és nyilak.** A definícióhoz tartozó jel a `\triangleq`, ekvivalencia jele `\equiv`. Nyilakat a `\rightarrow`, `\mapsto`, `\Rrightarrow` parancsokkal kapunk.

### 4.6. Feladatok

1. Írj egy dokumentumot, amelyben bemutatsz a binomiális tétel képletét általános és konkrét alakban. Használj `\sum` jelet és magyarázd el a jelöléseket.
2. Hozz létre egy mátrixot és egy determináns-kifejezést. Számítsd ki a  $2 \times 2$ -es mátrix determinánsát képlettel és szöveges magyarázattal.
3. Írj egy függvénydefiníciót a `cases` környezetben, majd készíts grafikus ábrázolást (akár későbbi órán tanult `TikZ`-cel), de itt elegendő a formula.
4. Készíts egy `align` környezetet, amelyben egy másodfokú egyenlet gyökeit deriválsz és integrálsz soronként. Használj hivatkozást az egyenletre a szövegben.

## 5. óra – Táblázatok készítése és testreszabása

### Cél

Az ötödik órán részletesen foglalkozunk a táblázatok szerkesztésével. Áttekintjük a `tabular` és `longtable` környezeteket, megtanuljuk a sorok és oszlopok összevonását, a táblázat szegélyezését, a szélességek és igazítások beállítását, valamint megismerkedünk a `booktabs` csomaggal, amely professzionális vonalstílusokat nyújt. A táblázatok fontos részei a műszaki dokumentumoknak, így a precíz szerkesztés alapvető készség.

### 5.1. A tabular környezet

A `tabular` környezet alapja a következő szintaxis: `\begin{tabular}{oszlop_specifikáció}`. Az oszlopokat betűk jelölik (`l` – balra zárt, `c` – középre zárt, `r` – jobbra zárt, `pszélesség` – több soros bekezdés). A vízszintes vonalakat a `\hline`, függőlegeseket a `\vline` szimbólum adja.

### Sorok és oszlopok összevonása

A `multicolumn` parancs egy cellát több oszlopon át terjeszt. Szintaxisa: `\multicolumn{n}{igazítás}{tartalom}`, ahol `n` az összefoglalt oszlopok száma, az igazítás pedig `l`, `c` vagy `r`. A `multirow` csomag `\multirow{n}{szélesség}{tartalom}` parancsa több sort fog össze egy cellában. A cella szélessége lehet `*` (automatikus) is.

### Professzionális vonalak

A `booktabs` csomag `\toprule`, `\midrule` és `\bottomrule` parancsai esztétikusabb vonalakat hoznak létre. A `\cmidrule` parancs részleges vonalat rajzol két oszlop között. Ajánlott kerülni a függőleges vonalakat; helyettük nagyobb térköz és vastagabb felső és alsó vonalak használata javasolt.

### 5.2. Hosszú táblázatok

Amikor a táblázat több oldalra nyúlik, a `longtable` környezetet használjuk. Ekkor a táblázat automatikusan törlik az oldalak között, és megadhatjuk a fejlécek és láblécek szövegét az egyes oldalak tetejére és aljára. A szintaxis a `tabular`-hoz hasonló, de a `\endhead`, `\endfoot`, `\endlastfoot` parancsokat is használhatjuk a fejléc és lábléc megadására.

### 5.3. Táblázatok igazítása és szélességek

Az oszlopok szélességét a `p{...}` oszloptípussal állíthatjuk be, ahol a kapcsos zárójelek között adott mértékben (`5cm`, `0.3\textwidth`) határozzuk meg a cella szélességét. Igazíthatjuk a szöveget fentről `\raggedright`, középre `\centering` vagy alulról `\raggedleft`. Az `array` csomag kiterjeszti a `tabular` környezetet, és lehetővé teszi új oszloptípusok definiálását.

### 5.4. További példák

Az alábbi példában egy összetett táblázatot hozunk létre, amelyben több oszlopot és sort összevonunk, és `booktabs` vonalakat használunk:

```
\begin{table}[ht]
\centering
\begin{tabular}{lccr}
\toprule
\multirow{2}{*}{Termék} & \multicolumn{2}{c}{rtékesítés} & \\
\multirow{2}{*}{sszbevétel} & & & \\
\end{tabular}
\end{table}
```

```

\cmidrule(r){2-3}
& 2024 & 2025 & \\
\midrule
Termék A & 100 & 120 & 220 \\
Termék B & 80 & 90 & 170 \\
Termék C & 60 & 70 & 130 \\
\bottomrule
\end{tabular}
\caption{Értékesítési adatok 20242025.}
\end{table}

```

A `\multirow` parancs alkalmazása megspórolja az ismételt cellák begépelését, a `\multicolumn` pedig lehetővé teszi az "Értékesítés" cím összevont elhelyezését.

## 5.5. Feladatok

1. Készíts egy `tabular` környezetű táblázatot, amelyben legalább négy oszlop és öt sor található. Használj `\multicolumn` és `\multirow` parancsokat a fejléc összevonására. Formázd a táblázatot a `booktabs` stílus szerint.
2. Alakítsd át az előző táblázatot `longtable` környezetté, hogy több oldalra törhessen. Állítsd be a fejléceket és lábléceket úgy, hogy minden oldalon megjelenjen a táblázat címe és az oldalszámozás.
3. Kísérletezz az oszlopok szélességével: hozz létre olyan táblázatot, ahol a harmadik oszlop szélessége 4 cm, a többi pedig automatikus. Igazítsd a szöveget középre és jobbra is különböző sorokban.
4. Írj olyan táblázatot, amelyben a cellákban több soros szöveg található. Használd a `p{...}` oszloptípust a tördeléshez, és állítsd be a sorok közötti függőleges távolságot a `\arraystretch` parancs segítségével.

## 6. óra – Képek beillesztése és rajzolás TikZ-zel

### Cél

A hatodik órán a képek beillesztésével és a diagramok rajzolásával foglalkozunk. A dokumentumok gyakori elemei az illusztrációk, grafikonok és folyamatábrák, amelyek a szöveges tartalmat vizuálisan támogatják. Megtanuljuk a `graphicx` csomag használatát, a kép méretezését és elhelyezését, a több ábrát tartalmazó `subfigure` környezet kezelését, valamint betekintést nyerünk a TikZ grafikus csomag alapjaiba.

### 6.1. Képek beillesztése graphicx-szel

A `\includegraphics` parancs használatával egyszerűen beilleszthetünk képfájlokat (PNG, JPG, PDF). A parancs opcionális paraméterei a kép méretezését szabályozzák:

- `width=` – megadja a kívánt szélességet (pl. `0.8\textwidth`).
- `height=` – beállítja a képmagasságot.
- `scale=` – a kép arányos nagyítását vagy kicsinyítését adja meg.

A képek általában a `figure` környezetben szerepelnek, amely lehetővé teszi felirat megadását (`\caption`) és címkézését (`\label`). A `\begin{figure}[htbp]` opciók (here, top, bottom, page) a lebegő objektum elhelyezésére szolgálnak.

### Képek vágása és forgatása

A `trim` opcióval levághatjuk a kép széleit:

`\includegraphics[trim=bal alsó jobb felső,clip]{fajl}`. A mértékegység általában pont (pt). A `angle` opció a kép elforgatását teszi lehetővé fokban.

### 6.2. Több kép egy ábrában – subfigure

A `subcaption` csomag `subfigure` környezete alkalmas arra, hogy egy ábrán belül több képet helyezzünk el külön felirattal. A `subfigure` környezetbe írt `\includegraphics` parancsok után a `\caption` megadja az alfeliratot, míg a fő `figure` környezet `\caption` parancsa az egész ábrára vonatkozik.

### 6.3. Rajzolás TikZ-zel

A TikZ csomag egy rendkívül erős eszköz vektoros grafikák készítéséhez. A `\begin{tikzpicture}` környezetben rajzolhatunk vonalakat, alakzatokat, nyilakat, szövegeket. A koordináták megadhatók derékszögű vagy poláris formában. Néhány alapvető parancs:

- `\draw (0,0) -- (2,0);` – szakasz rajzolása.
- `\draw (0,0) rectangle (2,1);` – téglalap rajzolása.
- `\draw[->] (0,0) -- (1,1);` – nyíl.
- `\node at (1,0.5) {Szöveg};` – szöveg elhelyezése.

A TikZ stílusok paraméterezésével (szín, vastagság, kitöltés) látványos diagramokat hozhatunk létre.

## Diagram példák

Lássunk egy egyszerű folyamatábrát, amely szemlélteti egy algoritmus lépéseit:

```
\begin{figure}[h]
\centering
\begin{tikzpicture}[node distance=2cm]
\node (start) [draw, rounded corners] {Start};
\node (cond) [draw, below of=start] {Feltétel?};
\node (yes) [draw, below left of=cond, xshift=-1cm] {Igen ág};
\node (no) [draw, below right of=cond, xshift=1cm] {Nem ág};
\draw[->] (start) -- (cond);
\draw[->] (cond) -- node[left] {igen} (yes);
\draw[->] (cond) -- node[right] {nem} (no);
\end{tikzpicture}
\caption{Egyszerű döntési fa TikZ-szel.}
\end{figure}
```

A fenti példában a csomópontok (node) elhelyezése a `node distance` paraméterrel történik. Az `rounded corners` opció lekerekített sarkokat eredményez.

## 6.4. Feladatok

1. Keresd egy tetszőleges képet (például egy ábrát a kedvenc könyvedből) és illeszd be a dokumentumba. Állítsd be a szélességét az oldal szélességének 60 %-ára, és készíts hozzá feliratot és címkét. Próbáld ki a `trim` opciót is, hogy levágj a széleiből.
2. Készíts egy ábra, amelyben két képet helyezel el egymás mellett `subfigure` környezetben. Adj mindkettőnek külön feliratot, és az egész ábrának közös címkét és feliratot.
3. Rajzolj TikZ segítségével egy koordináta-rendszert, és ábrázolj benne egy lineáris függvényt ( $y = 2x + 1$ ) egyenes vonallal és szöveggel jelöld a képletét. Használj színeket és tengelycímkéket.
4. Ábrázolj TikZ-ben egy négyszöget, amelynek csúcsaiba szöveg van írva (pl. "Kezdés", "Folyamat", "Választás", "Befejezés"), és nyilakkal kösd össze őket úgy, mintha folyamatábra lenne. Kísérletezz a `node distance` és az `align` paraméterekkel.

## 7. óra – Kereszthivatkozások és bibliográfia kezelése

### Cél

A hetedik órán a kereszthivatkozások és az irodalomjegyzék kezelésének mélyebb aspektusait tárgyaljuk. Megtanuljuk, hogyan adjunk címkéket a fejezeteknek, ábráknak, táblázatoknak és egyenleteknek, hogyan hivatkozunk ezekre a szövegben, milyen lehetőségei vannak a `\ref`, `\autoref` és `\cref` parancsoknak. Megismerkedünk a `biblatex` csomag által nyújtott haladó bibliográfiai beállításokkal, és áttekintjük a leggyakoribb hivatkozási stílusokat.

### 7.1. Címkék és hivatkozások

A  $\text{\LaTeX}$  címkerendszerével bármely lebegő objektumra vagy szakaszra hivatkozhatunk.

A `\label{kulcs}` parancsot közvetlenül a cím vagy környezet után kell elhelyezni. A hivatkozásokat három fő parancs szolgálja:

- `\ref` – csak a számot jeleníti meg (pl. 3. fejezet).
- `\eqref` – egyenletek számozására zárőjelezett hivatkozást ad (pl. (2.1)).
- `\autoref` vagy a `hyperref` csomag opciója esetén `\cref` – automatikusan hozzáteszi a típus nevét (pl. ábra 1.2).

A `\pageref` a címkézett elem oldalszámát adja vissza.

### 7.2. Hivatkozási stílusok testreszabása

A `biblatex` csomag a hivatkozások megjelenítését és sorrendjét finomhangolhatóvá teszi. Néhány fontos opció:

- `style`: meghatározza az irodalomjegyzék stílusát (`authoryear`, `numeric`, `alphabetic`).
- `citestyle`: a szövegek közötti hivatkozások formátuma. Az `authoryear` stílusnál a `\textcite` parancs szerző és év formátumot ad, míg az `\parencite` zárójelben jeleníti meg a hivatkozást.
- `sorting`: az irodalomjegyzék tételeinek sorrendjét határozza meg (pl. `ynt` – év, név, cím szerint).
- `backend`: `biber` (modern) vagy `bibtex` (hagyományos) feldolgozó motor.

A stílusok testreszabása a csomag dokumentációjában részletesen ismertetett

`\DeclareFieldFormat` és `\renewbibmacro` parancsokkal lehetséges, de az alapbeállítások a legtöbb esetben elegendőek.

### 7.3. BibTeX adatbázis felépítése

Az irodalmi hivatkozásokat külön `.bib` fájlban tároljuk. Minden bejegyzés egy egyedi címkével kezdődik, majd kulcs-érték párok formájában soroljuk fel az információkat. Például egy könyv bejegyzése:

```
@book{knuth1984,
  author = {Donald E. Knuth},
  title = {The \TeX{}book},
  year = {1984},
  publisher = {Addison--Wesley},
  address = {Reading, MA}
}
```

A mezők közül a `author`, `title` és `year` kötelező. Cikkekénél a `journal`, konferenciaanyagoknál a `booktitle` mezőt használjuk.

## 7.4. Feladatok

1. Hozz létre három különböző bejegyzést egy `referenciak.bib` fájlban: egy könyvet, egy folyóiratcikket és egy konferenciaticket. Ügyelj arra, hogy a szükséges mezőket kitöltsd, és adj minden tételnek egyedi kulcsot.
2. Készíts egy rövid dokumentumot, amelyben hivatkozol mindhárom forrásra. Használd a `\textcite`, `\parencite` és `\autocite` parancsokat különböző mondatszerkezetekben, és írd le a tapasztalt különbségeket.
3. Módosítsd a bibliográfia stílusát numerikusra (`style=numeric`) és a sorbarendezést név szerinti növekvő sorrendre. Fordítsd le a dokumentumot, és figyeld meg, hogyan változik az irodalomjegyzék.
4. Gyakorold a `\label` és `\ref` használatát: hozz létre két ábrát és egy táblázatot, mind-egyiknek adj címkét, majd írd róluk szöveget, amelyben a hivatkozások automatikusan a megfelelő sorszámot jelenítik meg.

## 8. óra – Kódbeillesztés, algoritmusok és hiperhivatkozások

### Cél

A nyolcadik órán olyan haladó csomagokkal ismerkedünk meg, amelyek jelentősen megkönnyítik a forráskód, algoritmusok és hiperhivatkozások kezelését. A `listings` csomag segítségével különféle programozási nyelvek kódjait emelhetjük be a dokumentumba szintaxiskiemeléssel. Az `algorithmicx` (és annak kiterjesztése, az `algpseudocode`) csomag olvasható álprogramok írását teszi lehetővé. A `hyperref` csomaggal pedig a belső és külső hivatkozásokat tehetjük kényelmesebbé és látványosabbá.

### 8.1. Forráskód beillesztése a listings csomaggal

A programkód dokumentálása során fontos, hogy a kód olvasható és jól formázott legyen. A `listings` csomag nyelvspecifikus szintaxiskiemelést, sorszámozást, keretezést biztosít. A környezet használatának alapja:

```
\begin{lstlisting}[language=Python, caption={sszeg függvény}, label={lst:sum}]
def osszeg(lista):
    """Visszaadja a lista elemeinek összegét."""
    s = 0
    for elem in lista:
        s += elem
    return s
```

A `language` kulcs jelöli a programozási nyelvet; a `caption` a magyarázó feliratot, a `label` pedig a hivatkozási kulcsot. A kód külseje a `\lstset` paranccsal globálisan konfigurálható (színek, betűtípus, sorszámozás).

### Példa: C program kódrészlet

Az alábbi példa bemutatja, hogyan jeleníthetünk meg C nyelvű kódot különböző beállításokkal:

1. Listing. C program példa

```
1 #include <stdio.h>
2 int main() {
3     // Helló világ program
4     printf("Hello, world!\n");
5     return 0;
6 }
```

A keret (`frame=single`) és a sorszámozás (`numbers=left`) segít a kód áttekintésében. A kulcsszavakat kék, a kommenteket zöld színnel emeltük ki.

### 8.2. Algoritmusok dokumentálása algorithmicx és algpseudocode segítségével

A pszeudokód írásához a `algorithm` környezet és az `algpseudocode` csomag nyújt könnyen olvasható struktúrát. Például egy keresési algoritmus:

```
\begin{algorithm}
\caption{Lineáris keresés}
\begin{algorithmic}[1]
\Procedure{LinearisKereses}{$tomb$, $x$}
\For{$i$ \gets 1 \textbf{to} $n$}
\If{$tomb[i] = x$}
\State \textbf{return} $i$
\EndIf
\EndFor
\EndProcedure
\end{algorithmic}
\end{algorithm}
```

```
\EndFor
\State \textbf{return} $-1$
\EndProcedure
\end{algorithmic}
\end{algorithm}
```

A [1] opció a sorok számozását aktiválja. A kulcsszavak (`\For`, `\If`) automatikusan félkövérek. Az algoritmus könnyen hivatkozható a `\label` paranccsal, és a `\listofalgorithms` parancs külön listát is generálhat róluk.

### 8.3. Hiperhivatkozások testreszabása `hyperref`-fel

A `hyperref` csomag automatikusan kattinthatóvá teszi a belső hivatkozásokat, de a megjelenésük testreszabható. Például:

```
\usepackage[colorlinks=true, linkcolor=blue, citecolor=green, urlcolor=red]{hyperref}
```

A `colorlinks` opció bekapcsolja a színes linkeket; a `linkcolor`, `citecolor` és `urlcolor` paraméterek beállítják a hivatkozások színét. A csomag metaadatokat is ad a PDF-nek, például a szerzőt és a címet (`pdfauthor`, `pdftitle`).

### 8.4. Feladatok

1. Írj egy Python függvényt, amely egy tömb maximumát keresi meg. Illeszd be a kódot a dokumentumba a `listings` csomaggal, állíts be saját színpalettát és sorszámozást.
2. Dokumentálj egy pszeudokódot az `algpseudocode` környezettel: például egy buborékrendezést. Adj neki címet és címkét, majd készíts listát az algoritmusokról a `\listofalgorithms` paranccsal.
3. Állítsd be a `hyperref` csomag opcióit úgy, hogy a belső hivatkozások pirosak, az irodalomjegyzék hivatkozásai zöldek, az URL-ek pedig lila színűek legyenek. Hozz létre egy kattintható URL-t, amely a kedvenc programozási dokumentációdra mutat.
4. Kombináld az előző feladatokat: írd egy dokumentumot, amelyben forráskódot és algoritmust is bemutatsz, majd belső hivatkozással utalj a kódlistára és az algoritmusra. Ellenőrizd, hogy a PDF-ben minden hivatkozás kattintható-e.

## 9. óra – Szakdolgozat és jelentés felépítése

### Cél

Ezen az órán megismerkedünk az évfolyam- és szakdolgozatok szerkezeti elemeivel. Áttekintjük a dolgozat részeit a címlaptól a mellékletekig, és megtanuljuk, hogyan hozhatunk létre egy egységes sablont. Megismerjük, hogyan állíthatjuk be a tartalomjegyzéket, az ábra- és táblázatjegyzéket, a sorközöket és a betűtípusokat az intézményi követelményeknek megfelelően.

### 9.1. A dolgozat fő részei

A dolgozat felépítése általában három nagy egységre bontható: elülső anyag, törzs és hátulsó anyag.

#### Elülső anyag

- **Címlap:** tartalmazza az intézmény nevét, a dolgozat címét, a hallgató nevét, a konzulens nevét és tudományos fokozatát, az elkészítés helyét és évét. A címlapot célszerű külön oldalra helyezni, és a cím nagy betűmérettel kiemelni.
- **Köszönetnyilvánítás és nyilatkozatok:** a hallgató itt köszönetet mondhat a konzulensének és támogatóinak, valamint bejelentheti, hogy a dolgozat saját munkája és nem sért szerzői jogot.
- **Tartalomjegyzék:** a `\tableofcontents` parancs segítségével generálható automatikusan. A `tocloft` csomaggal testre szabhatjuk a sorszámozást, betűméretet és behúzást.
- **Ábra- és táblázatjegyzék:** a `\listoffigures` és `\listoftables` parancsokkal készíthetők. Ezeket érdemes a tartalomjegyzék után elhelyezni.

#### Törzs

A törzs fejezetei a dolgozat szakmai tartalmát hordozzák. Az általános struktúra:

1. **Bevezetés:** bemutatja a témát, célkitűzéseket és a dolgozat felépítését.
2. **Elméleti háttér:** összefoglalja a releváns szakirodalmat, definiálja a fogalmakat.
3. **Módszertan:** ismerteti az alkalmazott módszereket, technológiákat, kísérleti beállításokat.
4. **Eredmények:** bemutatja az elért eredményeket, mérési adatokat, ábrákat, táblázatokat.
5. **Következtetések és javaslatok:** összegzi a tanulságokat, és javaslatot tesz további kutatásokra.

#### Hátulsó anyag

- **Irodalomjegyzék:** a `biblatex` csomaggal készítve, az intézményi előírások szerinti stílusban.
- **Mellékletek:** tartalmazhatják a kódlistákat, táblázatokat, további ábrákat vagy háttéranyagokat. A mellékleteket általában `\appendix` parancssal különítjük el, ami A, B, C betűkkel jelölt fejezetcímeket generál.

## 9.2. Sorközök és betűtípusok beállítása

A legtöbb intézmény megköveteli az 1,5-es vagy dupla sorközt. Ezt a `setspace` csomag `\onehalfspacing` vagy `\doublespacing` parancsával állíthatjuk be. Az egész dokumentumra érvényes, ha a `\begin{document}` után helyezzük el. A betűtípus meghatározása a `\usepackage{times}` vagy `\usepackage{libertine}` csomaggal történhet. Fontos, hogy a magyar karaktereket támogató fontokat válasszunk.

## 9.3. Fejezetcímek és tartalomjegyzék testreszabása

A címek megjelenését az `titlesec` csomaggal szabályozhatjuk. Például a `\titleformat{\chapter}[hang]{\huge\bfseries}{\thechapter.}{1em}{} parancs módosítja a fejezetcímek stílusát. A tocloft csomaggal a tartalomjegyzékben szereplő címek behúzását és betűméretét állíthatjuk. Példa:`

```
\usepackage{tocloft}
\renewcommand{\cftchapfont}{\bfseries}
\renewcommand{\cftchappagefont}{\bfseries}
\setlength{\cftbeforechapskip}{1.5ex}
```

## 9.4. Feladatok

1. Készíts sablont a saját dolgozatodhoz: hozd létre a címlapot, a nyilatkozatot, a tartalomjegyzéket és az ábra- és táblázatjegyzéket. Töltsd ki a címadatokat valós információkkal.
2. Írj három rövid fejezetet (Bevezetés, Elméleti háttér, Módszertan) néhány bekezdéssel, és győződj meg róla, hogy a címek megjelennek a tartalomjegyzékben. Állíts be 1,5-es sorközt és 12 pt betűméretet.
3. Készítsd el az irodalomjegyzéket a `biblatex` csomaggal. Adj hozzá legalább öt forrást, és hivatkozz rájuk a szövegben.
4. Adj hozzá egy mellékletet a dolgozatod végén, amelyben hosszabb kódrészlet található. A melléklet legyen a "Melléklet A" fejezetben, és jelenjen meg a tartalomjegyzékben.

## 10. óra – Beamer prezentációk

### Cél

**Célkitűzés** Ezen az órán a Beamer prezentációk készítésének részleteit vesszük át. A Beamer class olyan eszközöket nyújt, amelyekkel professzionális prezentációkat hozhatunk létre: különböző témák, színpaletták, átmenetek, animációk és tartalmi struktúrák állnak rendelkezésre. Megtanuljuk a keretek felépítését, a navigációs panelek használatát és a tartalom logikus tagolását.

### 10.1. Prezentációk felépítése

**Szekciók és al-szekciók** A Beamerben a `\section` és `\subsection` parancsok nem csupán tartalomjegyzéket generálnak, hanem a navigációs sávon is megjelennek. Ez segíti a hallgatóságot az előadás követésében. A keretek (`frame`) a legkisebb egységek, melyek alapesetben egy diát jelentenek.

**Témák és színek** A Beamer különböző témákat kínál (Madrid, Warsaw, CambridgeUS stb.), amelyek meghatározzák a háttér, a betűtípusokat és a navigációs eszközöket. A színpaletta módosításával a prezentáció illeszkedhet az intézmény arculatához. Például:

- `\usetheme{Warsaw}`: kiemelt navigációs sávok.
- `\usecolortheme{dolphin}`: visszafogott kék árnyalatok.
- Saját színek: `\setbeamercolor{structure}{fg=darkred}`.

Kísérletezzünk, hogy megtaláljuk a számunkra tetsző kombinációt.

### 10.2. Animációk és overlayek

**Felsorolások animálása** A diák tartalmának lépcsőről lépésre történő felfedése fókuszban tartja a hallgatóság figyelmét. A `\pause` parancs minden használata új overlayt hoz létre. Az `<n>` szögletes zárójelek segítségével pontosan megadhatjuk, mikor jelenjen meg egy elem. Például:

- `<1->` Első pont.
- `<2->` Második pont.
- `<3->` Harmadik pont.

A diagramoknál is használhatjuk az `\onslide` parancsot.

**Átmenetek és speciális effektek** A Beamer támogatja az áttűnéseket (transition). Például `\transfade` lassú elhalványulást, a `\transwipe` függőleges csúszást eredményez. Ezeket a keretek elején kell megadni. Ügyeljünk rá, hogy ne vigyük túlzásba: a túl sok effekt elvonja a figyelmet.

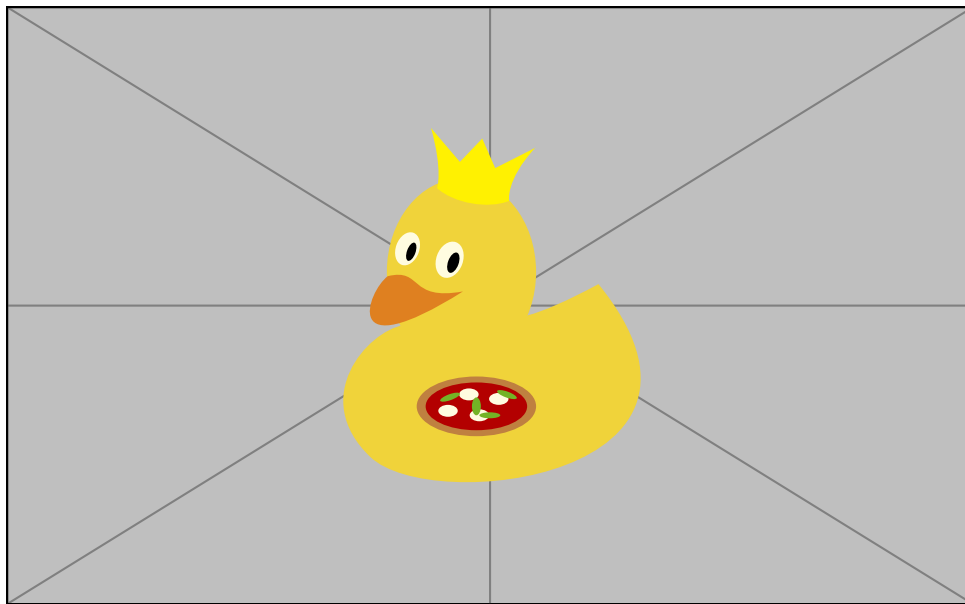
### 10.3. Vizualizációk

**Ábrák, diagramok és TikZ** Beépíthetünk képeket, ábrákat a `\includegraphics` paranccsal. A TikZ segítségével vektoros rajzokat vagy áramlási diagramokat készíthetünk. Például rajzolhatunk folyamatábrát vagy gráfot. A TikZ kódot közvetlenül a `frame` környezetbe írjuk, de nagyobb rajzokat célszerű külön file-ba szervezni.

**Gyakorlati példa** 0.5 Első oszlop Tartalom az első oszlopban. Használhatunk listákat:

- Kiemelt pont
- További információ

0.5 Második oszlop



1. ábra. Példa beillesztett kép

#### 10.4. Feladatok

1. Készíts egy 5–6 diából álló Beamer prezentációt a választott témádról. Használj legalább két különböző overlay technikát (`\pause` és `\onslide`), és válassz egy egyedi színsémát.
2. Hozz létre egy prezentációt, amely tartalmaz egy folyamatábrát TikZ segítségével és egy oszlopos diagramot (PGFPlots vagy beillesztett kép). Adj az ábráknak feliratot és hivatkozz rájuk a diákon.
3. Kísérletezz különböző Beamer témákkal és színtémákkal. Készíts egy táblázatot, amely összehasonlítja, hogy melyik téma milyen navigációs sávot és lábléceket használ. Mutasd be a preferált választásodat röviden.
4. Ha van intézményi Beamer sablon, próbáld meg azt testre szabni: cseréld le a színpalettát a sajátodéra, és adj hozzá logót a láblécbe. Ügyelj rá, hogy a logó mérete ne lógjon bele a tartalomba.

## 11. óra – Oldalbeállítások és tipográfiai stílusok

### Cél

Az oldalelrendezés és a tipográfiai stílus nem csupán esztétikai kérdés, hanem befolyásolja az olvashatóságot és a professzionális megjelenést is. A tizenegyedik órán megtanuljuk, hogyan állíthatjuk be pontosan a margókat, a sor- és bekezdésközöket, a fejléceket és lábléceket, hogyan szabhatjuk testre a fejezetcímek kinézetét a `titlesec` csomaggal, valamint hogyan választhatunk megfelelő betűtípus-családot.

### 11.1. Margók és oldalméretek

A `geometry` csomag segítségével tetszőlegesen állíthatjuk be a dokumentum margóit. Példa:

```
\usepackage{geometry}
\geometry{left=3cm,right=2.5cm,top=3cm,bottom=2.5cm}
```

A margók módosítása nemcsak esztétikai szempontból fontos; a könyvkötéshez szükséges kötés-margót a `bindingoffset` paraméterrel adhatjuk meg. Ha a dolgozatot kétoldalasra szánjuk, használjuk a `twoside` opciót, amely automatikusan tükrözi a margókat a páros és páratlan oldalakon.

### 11.2. Fejléc és lábléc testreszabása

A `fancyhdr` csomag lehetőséget ad arra, hogy a fejlécben és láblécben különféle információkat jelenítsünk meg: fejezetcím, dokumentum címe, oldalszám, dátum stb. A `\lhead`, `\chead` és `\rhead` parancsok a bal, középső és jobb oldali fejléc mezőket állítják be; hasonlóan `\lfoot`, `\cfoot`, `\rfoot` a lábléceket. Példa kétoldalas dokumentumra:

```
\fancyhead[LE,R0]{\thepage}
\fancyhead[L0]{\leftmark}
\fancyhead[RE]{\rightmark}
\fancyfoot[C]{\textit{Confidential}}
```

Itt a `\leftmark` a fejezetcím, a `\rightmark` az alfejezetcím, és a `\thepage` az oldalszámot jeleníti meg.

### 11.3. Sorköz és bekezdésköz

A sorköz beállítására a `setspace` csomag szolgál. Az `\doublespacing` dupla, az `\onehalfspacing` másfélszeres sorközt ad. A bekezdések közötti függőleges térközt a `\parskip` hosszal szabályozhatjuk. A bekezdés első sorának behúzását a `\parindent` állítja. Például:

```
\setlength{\parskip}{0.5\baselineskip}
\setlength{\parindent}{15pt}
```

### 11.4. Fejezetcímek formázása

A `titlesec` csomag segítségével tetszőleges stílusú címeket készíthetünk. A `\titleformat` parancs első argumentuma a cím típusa (`\section`, `\chapter`), a második a stílus (`display`, `hang`, `runin`), a harmadik a cím előtagja (pl. sorszám), a negyedik a cím előtti távolság, az ötödik pedig maga a cím formátuma. Példa alfejezetcím testreszabására:

```
\usepackage{titlesec}
\titleformat{\subsection}[runin]
  {\bfseries}{\thesubsection}{1em}{ }
\titlespacing*{\subsection}{0pt}{1ex}{1em}
```

Ez a beállítás félkövérré teszi az alfejezetcímet, közvetlenül a szöveg elé helyezi, és szabályozza a köz előtti és utáni térközt.

### 11.5. Betűtípusok és tipográfia

A betűtípus-család jelentősen befolyásolja a dokumentum hangulatát. A `\rmfamily` a talpas (serif), a `\sffamily` a talpatlan (sans-serif), a `\ttfamily` a rögzített szélességű (monospace) betűket aktiválja. Globálisan megváltoztathatjuk a betűtípust a `\renewcommand{\familydefault}{\sfdefault}` paranccsal. Léteznek modern betűkészletek is (pl. `lmodern`, `libertine`), amelyeket külön csomaggal tölthetünk be.

### 11.6. Feladatok

1. Készíts egy dokumentumot, amelyben a bal és a jobb margó eltérő méretű (pl. bal 4 cm, jobb 2 cm), és a felső margót nagyobbra állítod a fejléc miatt. Figyeld meg, hogyan változik az oldalszerkezet.
2. Állíts be egy egyedi fejléct és lábléct: a páros oldalakon jelenjen meg a dokumentum címe, a páratlanokon pedig a fejezet címe; a lábléc közepén tüntesd fel az oldalszámot.
3. Kísérletezz a sorközökkel: írd egy oldalnyi szöveget normál, másfélszeres és dupla sorközzel, majd hasonlítsd össze az olvashatóságot.
4. Formázd át a `\section` és `\subsection` címeket: legyenek talpas betűk helyett talpatlanok, a `\section` címek nagyobb betűmérettel, balról beljebb igazítva, a `\subsection` címek pedig félkövéren futószövegben jelenjenek meg. Használd a `titlesec` csomagot.

## 12. óra – Tételek, definíciók és bizonyítások

### Cél

A tizenkettedik órán a matematikai állítások, tételek, definíciók és bizonyítások  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -ben történő szakszerű megjelenítésével foglalkozunk. A `amsthm` csomag támogatja az elméleti munkákban gyakran használt strukturált környezeteket, és rugalmas lehetőségeket nyújt a környezetek sorszámozására és stílusára. Megismerkedünk a különböző theorem-stílusokkal, a bizonyítási környezet használatával és a QED szimbólum elhelyezésével.

### 12.1. Új környezetek létrehozása

Az `amsthm` csomag `\newtheorem` paranccsal hozhatunk létre új környezeteket. A szintaxis: `\newtheorem{név}{Megjelenő név}[számláló]`, ahol a név a parancs neve, a "Megjelenő név" a dokumentumban látható címke, a fakultatív számláló pedig azt jelzi, hogy melyik számláléhoz kötődjön a sorszám.

```
\newtheorem{tetel}{Tétel}[section]
\newtheorem{lemma}[tetel]{Lemma}
\newtheorem{definicio}{Definíció}[section]
```

Ebben a példában a `tetel` és `lemma` ugyanazt a számlálót használja, így egy fejezetben a lemmák és tételek közös sorszámozást kapnak (pl. 2.3. Tétel, 2.4. Lemma).

### 12.2. Theorem-stílusok

A `amsthm` különböző stílusokat enged meg a teorema típusú környezetekhez. A `plain` stílus döntött betűvel jeleníti meg a törzsszöveget; a `definition` stílus normál betűtípussal, a `remark` stílus pedig dőlt cím, de normál törzs betűkkel. Példa testreszabásra:

```
\theoremstyle{definition}
\newtheorem{pelda}{Példa}[section]
\theoremstyle{remark}
\newtheorem*{megjegyzes}{Megjegyzés}
```

A csillaggal jelölt környezetek nem sorszámozódnak.

### 12.3. Bizonyítások környezete

A `proof` környezet félkövér "Bizonyítás." felirattal kezdődik, majd a bizonyítás végén automatikusan elhelyezi a QED szimbólumot ( $\square$ ). Ha a bizonyítás végén egyenlet szerepel, a QED jelet a `\qedhere` paranccsal helyezhetjük el a sor végére. Példa:

```
\begin{lemma}
  Legyen  $n$  természetes szám. Ekkor  $n(n+1)$  páros.
\end{lemma}
\begin{proof}
  Ha  $n$  páros, akkor  $n(n+1)$  is páros, mert egy páros és egy páratlan szám szorzata páros. Ha  $n$  páratlan, akkor  $n+1$  páros, így a szorzat ismét páros. \qedhere
\end{proof}
```

### 12.4. Feladatok

1. Hozz létre saját tétel-, definíció- és megjegyzés-környezetet, amelyek sorszámozása a fejezethez kötődik. Definíálj néhány matematikai fogalmat (pl. gyűrű, test) definícióként, majd adj meg egy egyszerű tételt ezekre vonatkozóan.

2. Készíts egy rövid bizonyítást, amely két esetet vizsgál (pl. egy állítás igazodását páratlan és páros számokra), és a végén helyezd el a QED jelet manuálisan a `\qedhere` segítségével.
3. Változtasd meg az új tétel-környezet stílusát: legyen félkövér cím és normál törzsszöveg, majd hozz létre egy csillaggal ellátott változatot, amely nem sorszámozódik.
4. Írj egy példát, amelyben definíció, tétel, lemma és megjegyzés követik egymást. Használd a címkék és hivatkozások rendszerét, hogy a szövegben visszautalhass a korábban megadott definícióra vagy tételre.

## 13. óra – Kódlísták testreszabása és használata

### Cél

A tizenharmadik órán a forráskód dokumentumba illesztésének professzionális módjait tárgyaljuk. Megismerkedünk a `listings` csomag testreszabási lehetőségeivel, az alternatív `minted` csomag rövid bemutatásával, és megtanuljuk, hogyan emelhetünk ki bizonyos sorokat, hogyan adhatunk meg egyéni stílusokat, valamint hogyan illeszthetünk kódot a mellékletekbe.

### 13.1. Globális beállítások a `listings` csomagnál

A `\lstset` parancs segítségével globálisan beállíthatjuk a kódrészletek megjelenését. Például:

```
\lstset{
  numbers=left,
  numberstyle=\tiny\color{gray},
  stepnumber=1,
  numbersep=5pt,
  keywordstyle=\color{blue}\bfseries,
  commentstyle=\color{green!60!black}\itshape,
  stringstyle=\color{red},
  frame=single,
  breaklines=true,
  postbreak=\mbox{\textcolor{red}{${\hookrightarrow}$}}\space,
  backgroundcolor=\color{yellow!5}
}
```

Az ilyen beállítások biztosítják a konzisztens megjelenést: a sorszámok szürkék, a kulcsszavak kékek, a kommentek zöldek, az idézőjeles karakterláncok pirosak. A `breaklines` opció hosszú sorok tördelését teszi lehetővé, a `postbreak` pedig jelzi a sorok folytatását.

### 13.2. Sorok kiemelése és háttérszínek

Az `\lstset` vagy környezetszintű opciókkal megadhatjuk, hogy bizonyos sorok háttérszíne eltérjen. A `\lstdefinestyle` paranccsal elnevezhetünk stílusokat, majd alkalmazhatjuk őket:

```
\lstdefinestyle{kiemelt}{
  backgroundcolor=\color{orange!20},
  emph={main,return},
  emphstyle=\color{red}\bfseries
}
\begin{lstlisting}[style=kiemelt]
int main() {
    // kezdőpont
    return 0;
}
```

A `emph` kulcs a kiemelendő szavakat jelöli; a `emphstyle` pedig a hozzájuk tartozó stílust.

### 13.3. Alternatíva: `minted` csomag

A `minted` csomag a Python alapú Pygments szintaxiskiemelőt használja. Előnye, hogy több nyelvet támogat és szebb kiemelést biztosít. Használata során az `-shell-escape` opcióval kell fordítanunk a dokumentumot, mivel a Pygments külső program. Példa:

```
\usepackage{minted}
\begin{minted}[frame=lines,linenos]{python}
def fibonacci(n):
    a, b = 0, 1
```

```
for _ in range(n):  
    a, b = b, a + b  
return a  
\end{minted}
```

A `minted` környezet opcióival (`linenos`, `frame`, `fontsize`) finoman hangolható a megjelenés.

### 13.4. Kód mellékletekben

Nagy kódrészleteket célszerű a mellékletekbe szervezni, és a fő szövegben csak utalni rájuk. A mellékletben használhatjuk a `\lstinputlisting` parancsot, amely külső fájlból emeli be a kódot:

```
\appendix  
\chapter{Forráskód}  
\lstinputlisting[language=C++, caption={Főprogram}, label={lst:mainprog}]{main.cpp}
```

Ez a megoldás különösen hasznos hosszú programoknál, mivel így a kód karbantartása könnyebb.

### 13.5. Feladatok

1. Határozz meg egy globális `listings` stílust, amely megfelel az intézményed arculatának (válassz betűtípust, színeket, keretet). Illessz be két kódrészletet (különböző nyelveken), és ellenőrizd a következetes megjelenést.
2. Készíts egy kódrészletet, ahol kiemelsz néhány fontos kulcsszót piros félkövérrel, és beállítasz külön háttérszínt a teljes listának. Dokumentáld a használt beállításokat.
3. Próbáld ki a `minted` csomagot: illessz be egy Python kódrészletet, és állítsd be, hogy a sorok számozva legyenek és a keret árnyékolt legyen. Ne felejtsd el a dokumentum fordításakor az `-shell-escape` opciót.
4. Írj külön fájlba egy C programot (pl. `prog.c`), majd emeld be a dokumentumba a `\lstinputlisting` paranccsal. Adj hozzá mellékletet a dolgozatodhoz, és hivatkozz rá a főszövegben.

## 14. óra – Algoritmusleírások és pszeudokód

### Cél

A tizennegyedik órán a különböző algoritmusok dokumentálási módszereit tárgyaljuk. Megtanuljuk, hogyan írjuk le a pszeudokódot olvasható formában, miként strukturáljuk a ciklusokat, elágazásokat, alprogramokat az `algpseudocode` csomag segítségével, és hogyan hivatkozunk az algoritmusokra. Emellett bemutatjuk az `algorithm` környezet testreszabását és azt, hogy miként építhetők listákba az algoritmusok.

### 14.1. Az `algpseudocode` alapjai

A csomag kulcselemei a procedúrák, ciklusok és elágazások leírása. A `\Procedure` parancs egy eljárást határoz meg; a `\If–\Else–\EndIf`, `\For–\EndFor` szerkezetek pedig a vezérlési szerkezeteket jelölik. A paraméterek nyilaival (`\gets`) értékadást jelzünk.

```
\begin{algorithm}
  \caption{Buborékrendezés}
  \begin{algorithmic}[1]
    \Procedure{BubbleSort}{$A$}
      \State $n \gets \text{length}(A)$
      \For{$i \gets 1$ \textbf{to} $n$}
        \For{$j \gets 1$ \textbf{to} $n-i$}
          \If{$A[j] > A[j+1]$}
            \State csere$(A[j], A[j+1])$
          \EndIf
        \EndFor
      \EndFor
    \EndProcedure
  \end{algorithmic}
\end{algorithm}
```

A `[1]` opció soronként számozza a pszeudokódot, ami hivatkozáskor hasznos. A `\State` parancs egy utasítást jelöl.

### 14.2. Algoritmusok listázása

Ha több algoritmust is készítünk, a `\listofalgorithms` parancs automatikusan listát generál az algoritmusokról. A lista megjelenését a `\floatname` paranccsal magyarázhatjuk (pl. `\floatname{algorithm}{Algoritmus}`).

### 14.3. Stílusok és testreszabás

A `algorithm` környezet alapvetően lebegő objektum, amelynek megjelenítését a `\caption` és `\label` parancsokkal szabályozhatjuk. A csomag opciói között szerepel a pl. `[H]`, amely fix helyre rögzíti az algoritmust. A sorok közötti távolságot a `\setstretch` paranccsal növelhetjük. Az egyes utasításokhoz nyelvspecifikus formákat is rendelhetünk, például a `"to"` vagy `"do"` kulcsszavak magyarra cserélésével.

### 14.4. Feladatok

1. Írj egy pszeudokódot egy bináris keresés algoritmusára az `algpseudocode` csomaggal. Használj soronkénti számozást, és hivatkozz az algoritmusra a szövegben.
2. Dokumentálj három különböző algoritmust (pl. kiválasztásos rendezés, összefésülési rendezés, faktoriális kiszámítása rekurzívan). Hozz létre `\listofalgorithms` listát, és állítsd magyarra az `"Algorithm"` szó helyett az `"Algoritmus"` kifejezést.

3. Testreszabott kulcsszavakkal (pl. "**ciklus**", "**amíg**") írd le az algoritmus vezérlési szerkezeit. Használd a `\algnewcommand` parancsot a módosításhoz.
4. Mutass be egy példát arra, hogyan lehet egy algoritmus leírását lábjegyzetbe tenni: készíts egy rövidebb függvény leírását, és helyezd el a `\footnote` parancs segítségével.

## 15. óra – Diagramok és vizualizációk TikZ-zel

### Cél

A tizenötödik órán a TikZ és PGFPlots csomagok segítségével készítünk professzionális diagramokat. Megtanuljuk az alaprajzelemek, grafikonok, hálózatok megjelenítését, megtárgyaljuk a koordináta-rendszerek használatát, görbék rajzolását és a diagramok testreszabását. A vizualizáció kulcsfontosságú a tudományos munkában, hiszen az adatok szemléltetése sokszor többet mond minden szónál.

### 15.1. TikZ alapjai

A TikZ-ben a rajzot a `\begin{tikzpicture}` és `\end{tikzpicture}` környezet határolja. Az alakzatok rajzolását a `\draw` parancs végzi. Használhatunk színt, vonalvastagságot, kitöltést és gömbölyített sarkokat. Például:

```
\begin{tikzpicture}
  \draw[fill=blue!20] (0,0) rectangle (3,1.5);
  \draw[thick,->] (0,0) -- (3,0);
  \node at (1.5,0.75) {Címke};
\end{tikzpicture}
```

A `fill` paraméter kitölti az alakzatot, a `thick` növeli a vonalvastagságot. A `\node` parancs segítségével szöveget helyezhetünk el.

### 15.2. Koordináta-rendszerek és görbék

A TikZ nemcsak egyszerű alakzatok rajzolására alkalmas, hanem függvények ábrázolására is. A `\draw` parancs parabolát, szinuszt vagy bármilyen egyenletet is rajzolhat, ha a `plot` funkciót használjuk:

```
\begin{tikzpicture}
  \draw[->] (-2,0) -- (2,0) node[right] {$x$};
  \draw[->] (0,-1) -- (0,4) node[above] {$y$};
  \draw[domain=-2:2,samples=100,blue] plot ({x},{x*x});
  \node[blue] at (1.5,2.25) {$y=x^2$};
\end{tikzpicture}
```

A `domain` és `samples` opciók a rajzolandó szakasz hosszát és a pontok számát határozzák meg.

### 15.3. PGFPlots – grafikonok készítése

A `pgfplots` csomag a TikZ-re épül, és diagramok készítésére specializálódott. Könnyedén rajzolhatunk oszlopdiagramot, vonaldiagramot vagy scatter plotot. Egy egyszerű példában egy matematikai függvényt ábrázolunk:

```
\begin{tikzpicture}
  \begin{axis}[
    xlabel=$x$,
    ylabel={$f(x)$},
    title={Az $\sin(x)$ függvény},
    grid=both
  ]
    \addplot[red, thick] {sin(deg(x))};
  \end{axis}
\end{tikzpicture}
```

Az `axis` környezet beállításai között szerepel a cím, tengelyfeliratok, rácsvonalak. Az `\addplot` parancs után a függvényt megadhatjuk.

## Adatfájlok beolvasása

A PGFPlots képes külső CSV vagy TSV fájlok beolvasására is. Például:

```
\begin{tikzpicture}
  \begin{axis}[
    xlabel=Nap,
    ylabel=Hőmérséklet (C),
    grid=major
  ]
    \addplot table [x=nap, y=hom, col sep=comma] {adatok.csv};
  \end{axis}
\end{tikzpicture}
```

Itt az `adatok.csv` fájl "nap" és "hom" oszlopait használjuk a diagramhoz.

### 15.4. Feladatok

1. Rajzolj egy háromszöget TikZ-ben, jelöld a csúcsokat A, B, C betűkkel, és tüntesd fel a szögeket ívelt ívekkel. Színezd ki a háromszög belsejét halvány sárgával.
2. Ábrázolj két különböző függvényt ( $y = x^3$  és  $y = \sqrt{x}$ ) ugyanazon koordinátarendszerben. Használd a `plot` funkciót és különböző színeket, jelmagyarázatot (legend) is adj a görbékhez.
3. Készíts PGFPlots segítségével oszlopdiagramot, amely három termék eladási adatait hasonlítja össze két évben. Add meg az adatsorokat táblázatként a forráskódban.
4. Kísérletezz a rácsvonalak, tengelycímkék és tengelyskálák beállításával: hozz létre logaritmikus skálát és másodlagos tengelyt. Írj hozzá magyarázó szöveget, amely bemutatja, mikor célszerű logaritmikus ábrázolást használni.

## 16. óra – Tárgymutatók és szószedetek készítése

### Cél

Az indexek és szószedetek segítik az olvasót abban, hogy gyorsan megtalálja a dokumentumban előforduló fogalmakat és speciális kifejezéseket. A tizenhatodik órán megtanuljuk, hogyan jelölhetjük meg a fontos kulcsszavakat, hogyan strukturálhatjuk a tárgymutatót hierarchikusan, és hogyan állíthatunk elő részletes glosszáriumot több típusra bontva (fogalmak, rövidítések). A jól szerkesztett index és glosszárium nagyban növeli a hosszabb dokumentumok használhatóságát.

### 16.1. Indexkészítés lépésről lépésre

#### Index parancsok

Az index generálását az `imakeidx` csomaggal végezzük. A preambulumban a `\makeindex` parancs inicializálja az indexet. A dokumentumban a `\index{kulcsszó}` hívással jelöljük meg a szócikket. Alcsoportokat `!"` jellel hozunk létre: `\index{programozás!fájlkezelés}`. Az index előállításához a fordítás során extra lépésre van szükség: `pdflatex`, majd `makeindex` futtatása, végül újra `pdflatex` a frissített index beillesztéséhez.

#### Index formázása

Az index elrendezése a `\printindex` parancs paramétereivel módosítható. Például az `title` opcióval megadhatjuk a tárgymutató címét ("Tárgymutató"), a `columns` opcióval pedig többhasábos tördelést állíthatunk be. Egyéni stílus definiálásához használhatjuk a `\setindexpagestyle`, `\indexsetup` parancsokat, amelyekkel a betűméretet, sorközt, behúzást szabhatjuk.

#### Példák

Tegyük fel, hogy szeretnénk indexet készíteni egy matematikai jegyzethez. A szövegben a `\index` parancs így néz ki:

```
A sorozat fogalmát a \textit{sorozatok}\index{sorozat} fejezetben ismertetjük.
\newline
A \textit{határérték}\index{határérték!sorozat} definíciója fontos.
```

Az indexbe kerülő szócikkek hierarchikusan jelennek meg: a "határérték" a "sorozat" alcíme alatt. A `\printindex[columns=2]` parancs két hasábban rendezi a tárgymutatót.

### 16.2. Szószedetek és rövidítések

A `glossaries` csomag segítségével többféle glosszáriumot készíthetünk: definíciók, rövidítések, szimbólumok jegyzéke. A preambulumban a `\makeglossaries` parancsot hívjuk meg. Ezután új bejegyzést a `\newglossaryentry` paranccsal hozunk létre:

```
\newglossaryentry{latex}{
  name={\LaTeX},
  description={Magas színvonalú szedőrendszer tudományos dokumentumokhoz},
  sort=latex
}
```

A `sort` mező segítségével megadhatjuk a rendezési kulcsot. A szövegben a `\gls{latex}` parancs jeleníti meg a szócikk nevét, első előforduláskor automatikusan kiírja a teljes definíciót is (ha `\glsaddall` aktív).

## Rövidítések jegyzéke

A gyakori rövidítéseket a `\newacronym` paranccsal definiálhatjuk:

```
\newacronym{os}{OS}{operációs rendszer}
```

A `\gls{os}` parancs először kiírja "operációs rendszer (OS)", később csak a rövidítést. A rövidítések listája a `\printglossary[type=acronym]` parancs segítségével generálható.

### 16.3. Haladó beállítások

A glosszárium formázását a `\setglossarystyle` parancs határozza meg. A beépített stílusok (`list`, `altlist`, `super`) mellett saját stílust is készíthetünk. A többnyelvű dokumentumoknál a definíciók fordítása külön glosszáriumba kerülhet. Az index és glosszárium szinkronizálható is: hivatkozhatunk index bejegyzésre glosszáriumból.

### 16.4. Feladatok

1. Hozz létre egy rövid tanulmányt a kedvenc tantárgyadról, és készíts hozzá tárgymutatót legalább tíz bejegyzéssel, köztük alcsoportokkal (pl. "programozás!változók"). Állítsd be, hogy az index két hasábjában jelenjen meg.
2. Készíts glosszáriumot öt fogalommal és három rövidítéssel. Használd a `\gls` és `\Gls` (nagy kezdőbetűs forma) parancsokat a szövegben, és a dokumentum végén generáld le a glosszáriumot és az akronimok listáját.
3. Testreszabott indexstílust készíts: változtasd meg a betűméretet 10 pt-re, növelj meg a sorok közötti távolságot, és állítsd be, hogy a bejegyzések neve félkövérrel jelenjen meg.
4. Készíts egy dokumentumot, amelyben az index és a glosszárium együtt szerepel. Hivatkozz a glosszárium definícióira az indexben: például `\index{operációs rendszerseeOS|jelleggel}`.

## 16.5. óra – Haladó irodalomkezelés és hivatkozási stílusok

### Cél

A tizenhetedik órán elmélyítjük ismereteinket az irodalomjegyzék kezelésében. Megtanuljuk, hogyan hozhatunk létre több bibliográfiát ugyanazon dokumentumban, hogyan rendezhetjük a hivatkozásokat kategóriák szerint, hogyan módosíthatjuk a megjelenést a `biblatex` csomag makróival, és áttekintjük az idézési parancsok gazdag repertoárját. A tudományos munkában az egységes és pontos hivatkozási stílus elengedhetetlen.

### 16.6. Több bibliográfia egy dokumentumban

Nagyobb dolgozatoknál előfordulhat, hogy külön bibliográfiára van szükség a könyvek, a cikkek és az internetes források számára. Ezt a `\printbibliography` parancs `type` és `keyword` opcióival oldhatjuk meg. Például a forrásfájlon belül minden bejegyzéshez hozzárendelhetünk egy kulcsszót (`keywords={könyv}`), majd külön bibliográfiát generálunk:

```
\printbibliography[keyword=könyv, title=Könyvek]
\printbibliography[keyword=cikk, title=Cikkek]
\printbibliography[keyword=honlap, title=Weboldalak]
```

Alternatív megoldásként több `.bib` fájlt is megadhatunk a `\addbibresource` parancs segítségével, majd `\printbibliography` hívással külön-külön listázhatjuk őket.

### 16.7. Hivatkozási parancsok

A `biblatex` számos citációs parancsot kínál, amelyek különböző kontextusokban használhatók:

- `\textcite{kulcs}` – szerző neve a szövegben jelenik meg, év zárójelben.
- `\parencite{kulcs}` – szerző és év zárójelben.
- `\footcite{kulcs}` – lábjegyzetben jelenik meg a hivatkozás.
- `\citeauthor{kulcs}`, `\citeyear{kulcs}` – csak a szerző vagy az év beillesztése.
- Több forrás együttes hivatkozása vesszővel elválasztva: `\parencite{kulcs1,kulcs2}`.

Az `\autocite` parancs a stílustól függően választja ki a megfelelő hivatkozási formát (lábjegyzetes vagy zárójeles).

### 16.8. Bibliográfiai rekordok szerkesztése

A BibTeX bejegyzéseknél lehetőség van további mezők megadására: `url`, `urldate` (lekérés dátuma), `isbn`, `doi`, `note`. A DOI és URL mezők esetén a `hyperref` csomag automatikusan kattintható linkeket hoz létre. Példa konferencia bejegyzésre:

```
@inproceedings{minta2023,
  author = {Kiss, János and Nagy, Péter},
  title = {Egy új algoritmus bemutatása},
  booktitle = {XXI. Magyar Informatikai Konferencia},
  year = {2023},
  pages = {12--18},
  doi = {10.1234/konf.2023.001},
  note = {Előadás}
}
```

## 16.9. Makrók és formátumok testreszabása

Haladó felhasználók a `biblatex` makróinak újradefiniálásával változtathatják meg a hivatkozások megjelenését. Például a szerzők nevének sorrendjét, a köztes kötőszavakat (in:, et al.), vagy azt, hogy a szerzők és év között legyen-e vessző. Ezeket a `\DeclareFieldFormat`, `\DeclareNameAlias` parancsokkal tehetjük meg. A kód részletes ismertetése a csomag dokumentációjában található.

## 16.10. Feladatok

1. Hozz létre egy `forrasok.bib` fájlt, amely legalább hat forrást tartalmaz: két könyvet, két cikket és két internetes forrást. Adj mindegyikhez kulcsszót (`book`, `article`, `online`).
2. Készíts egy dokumentumot, amely külön irodalomjegyzéket generál a könyvekre, cikkekre és online forrásokra. Használj eltérő címet és felvezető szöveget mindegyikhez. Hivatkozz mindegyik típusból legalább egyszer a szövegben.
3. Gyakorold a különböző citációs parancsok használatát: írd egy bekezdést, amelyben `\textcite`, `\parencite` és `\footcite` parancsokat használj ugyanarra a forrásra. Jegyezd meg, hogyan változik a formátum.
4. Próbáld ki a lábjegyzetes hivatkozási stílust a `biblatex verbose-note` stílusával: állítsd át a stílust, majd nézd meg, hogyan jelennek meg a hivatkozások és az irodalomjegyzék. Hasonlítsd össze az `authoryear` stílussal.

## 17. óra – Haladó irodalomkezelés és hivatkozási stílusok

### Cél

A tizenhetedik órán elmélyítjük ismereteinket az irodalomjegyzék kezelésében. Megtanuljuk, hogyan hozhatunk létre több bibliográfiát ugyanazon dokumentumban, hogyan rendezhetjük a hivatkozásokat kategóriák szerint, hogyan módosíthatjuk a megjelenést a **biblatex** csomag makróival, és áttekintjük az idézési parancsok gazdag repertoárját. A tudományos munkában az egységes és pontos hivatkozási stílus elengedhetetlen.

### 17.1. Több bibliográfia egy dokumentumban

Nagyobb dolgozatoknál előfordulhat, hogy külön bibliográfiára van szükség a könyvek, a cikkek és az internetes források számára. Ezt a `\printbibliography` parancs `type` és `keyword` opcióival oldhatjuk meg. Például a forrásfájlon belül minden bejegyzéshez hozzárendelhetünk egy kulcsszót (`keywords={könyv}`), majd külön bibliográfiát generálunk:

```
\printbibliography[keyword=könyv, title=Könyvek]
\printbibliography[keyword=cikk, title=Cikkek]
\printbibliography[keyword=honlap, title=Weboldalak]
```

Alternatív megoldásként több `.bib` fájlt is megadhatunk a `\addbibresource` parancs segítségével, majd `\printbibliography` hívással külön-külön listázhatjuk őket.

### 17.2. Hivatkozási parancsok

A **biblatex** számos citációs parancsot kínál, amelyek különböző kontextusokban használhatók:

- `\textcite{kulcs}` – szerző neve a szövegben jelenik meg, év zárójelben.
- `\parencite{kulcs}` – szerző és év zárójelben.
- `\footcite{kulcs}` – lábjegyzetben jelenik meg a hivatkozás.
- `\citeauthor{kulcs}`, `\citeyear{kulcs}` – csak a szerző vagy az év beillesztése.
- Több forrás együttes hivatkozása vesszővel elválasztva: `\parencite{kulcs1,kulcs2}`.

Az `\autocite` parancs a stílustól függően választja ki a megfelelő hivatkozási formát (lábjegyzetes vagy zárójeles).

### 17.3. Bibliográfiai rekordok szerkesztése

A BibTeX bejegyzéseknél lehetőség van további mezők megadására: `url`, `urldate` (lekérés dátuma), `isbn`, `doi`, `note`. A DOI és URL mezők esetén a **hyperref** csomag automatikusan kattintható linkeket hoz létre. Példa konferencia bejegyzésre:

```
@inproceedings{minta2023,
  author = {Kiss, János and Nagy, Péter},
  title = {Egy új algoritmus bemutatása},
  booktitle = {XXI. Magyar Informatikai Konferencia},
  year = {2023},
  pages = {12--18},
  doi = {10.1234/konf.2023.001},
  note = {Előadás}
}
```

## 17.4. Makrók és formátumok testreszabása

Haladó felhasználók a `biblatex` makróinak újradefiniálásával változtathatják meg a hivatkozások megjelenését. Például a szerzők nevének sorrendjét, a köztes kötőszavakat (in:, et al.), vagy azt, hogy a szerzők és év között legyen-e vessző. Ezeket a `\DeclareFieldFormat`, `\DeclareNameAlias` parancsokkal tehetjük meg. A kód részletes ismertetése a csomag dokumentációjában található.

## 17.5. Feladatok

1. Hozz létre egy `forrasok.bib` fájlt, amely legalább hat forrást tartalmaz: két könyvet, két cikket és két internetes forrást. Adj mindegyikhez kulcsszót (`book`, `article`, `online`).
2. Készíts egy dokumentumot, amely külön irodalomjegyzéket generál a könyvekre, cikkekre és online forrásokra. Használj eltérő címet és felvezető szöveget mindegyikhez. Hivatkozz mindegyik típusból legalább egyszer a szövegben.
3. Gyakorold a különböző citációs parancsok használatát: írd egy bekezdést, amelyben `\textcite`, `\parencite` és `\footcite` parancsokat használsz ugyanarra a forrásra. Jegyezd meg, hogyan változik a formátum.
4. Próbáld ki a lábjegyzetes hivatkozási stílust a `biblatex verbose-note` stílusával: állítsd át a stílust, majd nézd meg, hogyan jelennek meg a hivatkozások és az irodalomjegyzék. Hasonlítsd össze az `authoryear` stílussal.

## 18. óra – Git, Overleaf és kollaboráció

### Cél

A tizennyolcadik órán a dokumentumok közös szerkesztésének kérdésével foglalkozunk. A Git verziókezelő rendszer és az Overleaf online szerkesztő használatával megtanuljuk, hogyan kezeljük a változtatásokat, hogyan egyesítjük a párhuzamosan dolgozó csapattagok munkáját, és hogyan oldjuk fel az ütközéseket. A verziókezelés nemcsak biztonsági mentést nyújt, hanem visszakereshetővé teszi a korábbi állapotokat is.

### 18.1. Git alapfogalmak

#### Repository és commit

A Git repository a projekt teljes történetét tárolja. Az `git init` parancs inicializálja az új repositoryt. A változtatásokat staging area-ba helyezzük (`git add fájl`), majd `git commit -m "Üzenet"` parancsal rögzítjük őket. Minden commit egyedi azonosítóval rendelkezik és tárolja a módosításokat, az időpontot és a szerző nevét.

#### Branch-ek és merge

A branch (ág) segítségével párhuzamos fejlesztési szálakat hozhatunk létre. Például a `main` ágról leágazva létrehozhatunk egy `draft` ágot, amelyen új fejezetet írunk. Az `git merge` paranccsal egyesíthetjük a változtatásokat, de ütközés esetén kézzel kell eldöntenünk, melyik verziót tartjuk meg.

#### Távoli repository-k

A GitHub, GitLab és Bitbucket távoli tárolók, ahol a repositoryk online elérhetők.

A `git remote add origin url` parancs regisztrálja a távoli címet, majd a `git push -u origin main` feltölti az első commitot. A `git pull` parancs letölti és egyesíti a távoli változásokat.

### 18.2. Gyakorlati munkamenet

Egy tipikus LaTeX projekt során így nézhet ki a munkamenet:

1. A hallgatók klónozzák a központi repositoryt: `git clone url`.
2. Létrehoznak egy új ágot a saját munkájuknak: `git checkout -b sajátág`.
3. Szerkesztik a `.tex` fájlokat, majd rendszeresen commitolnak: `git add` és `git commit`.
4. A munkájuk végén a `git push` paranccsal feltöltik az ágot a távoli szerverre.
5. A team lead összevonja a változtatásokat a fő ágba (pull request), majd megoldja az esetleges ütközéseket.

### 18.3. .gitignore és LaTeX projektek

A Git repositoryba nem célszerű minden generált fájlt feltölteni. A `.gitignore` fájl felsorolja, hogy milyen mintájú fájlokat hagyjon figyelmen kívül a Git. LaTeX projektnél érdemes elrejtetni a következőket: `*.aux`, `*.log`, `*.toc`, `*.out`, `*.synctex.gz`, `*.pdf`. A generált PDF-et elég a kész verzió közzétételénél tárolni.

## 18.4. Overleaf és Git integráció

Az Overleaf lehetővé teszi az online szerkesztést és a csapatmunkát. Főbb funkciói:

- **Valós idejű szerkesztés:** a szerzők egyszerre dolgozhatnak, az Overleaf pedig egy "track changes" funkcióval követi a módosításokat.
- **Git hozzáférés:** minden Overleaf projekt rendelkezik egy Git végponttal, amelyen keresztül klónozhatjuk a projektet. Így offline is dolgozhatunk, majd a `git push` paranccsal visszaküldhetjük a változtatásokat.
- **Megosztás és jogosultságkezelés:** meghívhatunk szerkesztőket, megadhatunk csak olvasási jogosultságot.

## 18.5. Feladatok

1. Inicializálj egy LaTeX projektre Git repositoryt. Készíts `.gitignore` fájlt a generált LaTeX fájlok kizárására. Végezz el legalább három commitot, amelyek új fejezeteket vagy táblázatokat adnak hozzá.
2. Hozz létre két branch-et (**fejlesztés** és **kiserlet**), és mindkettőn végezz különböző változtatásokat (pl. új környezetek, sorköz beállítások). Próbáld meg összevonni őket a main ágra; ha ütközés van, oldd meg manuálisan.
3. Regisztrálj egy GitHub vagy GitLab fiókot, és töltsd fel a repositoryt. Kérj meg egy csoporttársat, hogy klónozza a repositoryt, hozzon létre egy pull requestet, és végezzétek el a code review-t.
4. Hozz létre egy Overleaf projektet, és kapcsolj hozzá Git hozzáférést. Klónozd a projektet a saját gépedre, szerkeszd a dokumentumot offline, majd pushold vissza a módosításokat. Ellenőrizd, hogy az Overleaf felületén is megjelennek a változások.

## 19. óra – Haladó Beamer funkciók és testreszabás

### Cél

**Célkitűzés** A tizenkilencedik órán a Beamer előrehaladott funkcióit tekintjük át. Megtanuljuk, hogyan készíthetünk saját témát vagy testreszabhatunk meglévő témákat, hogyan használhatunk összetett overlay technikákat (dinamikus táblázatok, felépülő ábrák), hogyan illeszthetünk be interaktív elemeket (kvíz kérdések, animációk), és hogyan integráljuk a PGFPlots grafikonjait Beamer környezetbe. Az igényes prezentáció a tanár számára eszköz a szemléletes oktatáshoz.

### 19.1. Téma testreszabása

**Saját téma készítése** A Beamer témák a `beamercolortheme`, `beamerfonttheme` és `beamertemplatename` parancsokkal állnak össze. Egy egyszerű saját téma definiálása:

- Hozz létre egy `mytheme.sty` állományt, amelyben definiáld a színeket:  
`\definecolor{myblue}{RGB}{30,85,165}`.
- Állítsd be a struktúra színeit: `\setbeamercolor*{section in toc}{fg=myblue}`.
- Változtasd meg a betűtípust:  
`\setbeamerfont{title}{family=\sfdefault, series=\bfseries}`.
- Töltsd be a témát: `\usepackage{mytheme}`.

A saját téma kialakítása időigényes, de teljes kontrollt ad a megjelenés felett.

### 19.2. Összetett overlayek

**Táblázatok és diagramok animálása** Az `\only` és `\uncover` parancsok segítségével tetszőleges elemeket jeleníthetünk meg bizonyos overlayeken. Például egy táblázat sorainak fokozatos felfedése:

| Év       | Eladás  | Nyereség |
|----------|---------|----------|
| <1->2023 | <1->120 | <1->30   |
| <2->2024 | <2->150 | <2->50   |
| <3->2025 | <3->200 | <3->80   |

Minden új overlay felfed egy új sort. Hasonló technikát alkalmazhatunk diagramoknál is az `\addplot` parancsok overlay specifikációjával (pl. `\addplot<2->`).

### 19.3. Interaktív elemek

**Kvíz kérdések beépítése** A Beamerhez tartozó `buttons` és `hyperlink` funkciók segítségével interaktív prezentációkat készíthetünk. Például kvízkérdések navigációja:

Mutasd a választ

Kérdés Mi a különbség a `\pause` és az `\onslide` parancs között? Válasz A `\pause` egymás után következő overlayeket hoz létre, míg az `\onslide` explicit módon határozza meg, hogy az elem melyik overlayen jelenjen meg.

### 19.4. PGFPlots integráció

A PGFPlots csomag diagramjai könnyen illeszthetők a Beamer keretekbe. Az `axis` környezet automatikusan igazodik a diák méreteihez. Például egy lineáris és egy exponenciális függvény összehasonlítása:

$$xf(x)f(x) = x^x;^x$$

A legend csak akkor jelenik meg, amikor a kapcsolódó görbe látható.

## 19.5. Feladatok

1. Készíts saját Beamer témát: válassz színeket, betűtípusokat, és definiálj egy egyedi lábléceket, amely tartalmazza a diaszámot és az előadás címét. Töltsd be a témát a prezentációdba.
2. Hozz létre egy táblázatot, amelynek sorait overlay segítségével fokozatosan feded fel. Adj hozzá ehhez tartozó diagramokat PGFPlots-szal, amelyek szintén overlayen jelennek meg.
3. Valósíts meg egy interaktív kvízkeretet: tegyél fel három kérdést, a válaszokat rejtve, és gombokkal navigálj a válaszokhoz. Használd a `\hyperlink` és `\beamertobutton` parancsokat.
4. Illessz be két különböző skálázású grafikon ugyanabban a keretben: az egyik lineáris, a másik logaritmikus y-tengellyel. Állítsd be a tengelycímkéket, és magyarázd el a különbség jelentőségét.

## 20. óra – Végző projekt összeállítása és csomagolása

Ez az óra összefoglalja, hogyan állíts össze egy nagyobb terjedelmű LaTeX-projektet – például szakdolgozatot vagy évfolyamunkát – a már elkészített részdokumentumokból. Az előző foglalkozásokon elsajátítottad a dokumentumstruktúra, szövegformázás, matematikai képletek, táblázatok, ábrák, kereszthivatkozások és bibliográfia kezelésének alapjait, valamint megismerted a különféle kiegészítő csomagokat. Most bemutatjuk, hogyan illeszkedik mindez egyetlen koherens dokumentumba, és mit kell tenned a munka archiválásához.

### Fejezetek összeillesztése

A nagy dokumentumok kezelése könnyebb, ha a tartalmat logikusan részfájlokra bontod. Például a bevezetést, elméletet, módszertant, eredményeket és következtetéseket külön `.tex` fájlokban tárolhatod. Az összeállításhoz hozz létre egy fő fájlt (`main.tex`), amelyben megadod a dokumentumosztályt, betöltöd a szükséges csomagokat, majd a `\include` vagy `\input` parancsokkal beilleszted a fejezetfájlokat. A következő vázlat mutatja a fő dokumentum felépítését:

```
% main.tex
\documentclass[12pt,a4paper]{report}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[magyar]{babel}
% ... (további csomagok) ...

\begin{document}
\tableofcontents

\include{chapters/bevezetes}
\include{chapters/elmelet}
\include{chapters/modszertan}
\include{chapters/eredmenyek}
\include{chapters/kovetkeztetesek}

\bibliography{bibliography}
\bibliographystyle{plain}

\end{document}
```

### Bibliográfia kezelése

Az irodalomjegyzék összeállításához BibTeX vagy `biblatex` használható. Ha egyszerű listát szeretnél (a stílus és a nyelv megadása nélkül), a következő minimális beállítás is elegendő:

```
\bibliographystyle{plain}
\bibliography{bibliography}
```

Itt a `bibliography.bib` fájl tartalmazza a forrásaidat. A `'biblatex'` csomag fejlettebb megoldásokat kínál (magyar nyelvű hivatkozási stílus, szerző-év jelölés, stb.), de a használatához `'biber'` nevű segédprogram szükséges. A `'plain'` stílus könnyen kompilálható, még haladó eszközök nélkül is.

### Kódpéldák dokumentálása

A LaTeX forráskód bemutatásához a `listings` csomagot használjuk. Ehhez a fenti beállításokat már alkalmaztuk: a `listingsutf8` csomag engedélyezi az UTF-8 kódolást, a `literate` opció pedig

meghatározza, hogyan jelenjenek meg az ékezetes magyar betűk. Ha sortörést szeretnél a hosszú soroknál, a `'breaklines=true'` és `'breakatwhitespace=true'` opciók gondoskodnak róla. Az alábbi példa megmutatja, hogy miként dokumentálhatsz egy egyszerű LaTeX állományt kódként:

```
\documentclass[12pt,a4paper]{report}
\usepackage[magyar]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\title{Szakdolgozat címe}
\author{Hallgató Neve \and Konzulens Neve}
\date{2025. május 30.}

\begin{document}
\maketitle
\end{document}
```

Figyeld meg, hogy az ékezetes betűk („Hallgató”, „május”) az alapbeállítással is helyesen jelennek meg, mert gondoskodtunk a megfelelő kódtérképezésről.

## Projekt archíválása

Miután összeállítottad a végső dokumentumot, érdemes a forrásokat és a generált PDF-et archíválni. A legegyszerűbb módszer egy ZIP vagy `tar.gz` állomány létrehozása, amely tartalmazza a `.tex`, `.bib`, ábra- és adatfájlokat. A projekt könnyen reprodukálható marad, ha mellékelsz egy `README.md` fájlt is, ami leírja, milyen TeX-disztribúcióval és melyik parancsokkal kell fordítani (például `'pdflatex main.tex'`).

## Záró feladatok

1. Hozz létre egy saját projektet, amely három fejezetből áll, és `\include` parancsokkal illeszd őket egy fő dokumentumba.
2. Készíts egy rövid bibliográfiát legalább három forrással (könyv, cikk, weboldal), és jelenítsd meg az irodalomjegyzéket a dokumentum végén.
3. Írj egy kódrészletet a `listings` környezettel, amelyben magyar ékezetes betűket is használsz. Ellenőrizd, hogy a hosszú sorok sortöréssel jelennek meg.
4. Archiváld a projektet egy tömörített fájlba, és készíts hozzá `README.md` dokumentációt.

Ezeknek a lépéseknek a végrehajtásával sikeresen lezárhatod a LaTeX-kurzust, és magabiztosan vághatsz bele a szakdolgozat vagy évfolyammunka elkészítésébe. A LaTeX lehetővé teszi, hogy a tartalomra koncentrálj, miközben a formázás és a dokumentum struktúrája automatizáltnak, magas minőségben kerül előállításra.

## Hivatkozások

## Irodalomjegyzék

### Irodalomjegyzék

1. **Lamport, L.:** *LaTeX: A Document Preparation System*. Addison–Wesley, Reading (MA), 1994, 272 p.
2. **Mittelbach, F. – Goossens, M.:** *The LaTeX Companion*. Addison–Wesley, Boston, 2004, 1120 p.
3. **Oetiker, T. – Partl, H. – Hyna, I. – Schlegl, E.:** *The Not So Short Introduction to LaTeX2ε*. Online kiadás, 2021, — p.
4. **Horváth, P.:** *LaTeX a gyakorlatban – Bevezetés és példatár*. ELTE Eötvös Kiadó, Budapest, 2019, 210 p.
5. **Kovács, G.:** *Tudományos szövegek készítése LaTeX-ben*. Typotex, Budapest, 2022, 185 p.
6. **Tómacs, T.:** *LaTeX – kézikönyv kezdő és haladó felhasználóknak*. Online kiadás, Eszterházy Károly Egyetem, Eger, 2025, — p.
7. **Андрусишин, Л.:** *LaTeX для студентов та викладачів*. Львівський національний університет, Львів, 2020, 152 p.
8. **Бондаренко, О. – Мельник, С.:** *Практичний курс з LaTeX для підготовки наукових публікацій*. Київський університет, Київ, 2021, 198 p.
9. **Чумаченко, І.:** *Інформаційні технології в освіті та науці*. Харківський національний університет, Харків, 2018, 240 p.

### Jogszabályok és szabályzatok

10. *Закон України «Про освіту» від 05.09.2017 р.* Верховна Рада України, Київ, 2017.
11. *Закон України «Про вищу освіту» від 01.07.2014 р.* Верховна Рада України, Київ, 2014.
12. *Положення про проведення практики студентів ЗУІ*. Закарпатський угорський інститут, Берегове, — p.

### Online források

1. *MiKTeX hivatalos oldala*. <https://miktex.org/download>
2. *Overleaf – LaTeX Tutorials*. <https://www.overleaf.com/learn/latex/Tutorials>
3. *TeXnicCenter – hivatalos letöltés*. <https://www.texniccenter.org/download/>

*Виробничо-практичне видання*

# **«Практика з виготовлення наукових текстів та засобів наочності з інформатики»**

**МЕТОДИЧНІ ПОРАДИ ТА ЗАВДАННЯ  
ЩОДО ПРОХОДЖЕННЯ ПРАКТИКИ**

2025 р.

Затверджено до використання у навчальному процесі  
на засіданні кафедри математики та інформатики ЗУІ ім. Ф.Ракоці II  
(протокол № 10 від «23» червня 2025 року)

Розглянуто та рекомендовано Навчально-методичною радою  
Закарпатського угорського інституту імені Ференца Ракоці II  
(протокол № 7 від «26» серпня 2025 року)

Рекомендовано до видання в електронній формі (PDF)  
рішенням Вченої ради Закарпатського угорського інституту імені Ференца Ракоці II  
(протокол № 8 від «28» серпня 2025 року)

Підготовлено до видання в електронній формі (PDF) кафедрою математики та інформатики  
спільно з Видавничим відділом Закарпатського угорського інституту імені Ференца Ракоці II  
(53 с., угорською мовою)

Укладачі:

*Каталін Кучінка* – кандидат фіз.-мат. наук, доцент, завідувач кафедри математики та інформатики Закарпатського угорського інституту ім. Ференца Ракоці II;

*Габріела Пан* – старший викладач кафедри математики та інформатики Закарпатського угорського інституту ім. Ференца Ракоці II

Рецензенти:

*Ганна Сливка-Тилищак* – доктор фіз.-мат. наук, завідувач кафедри теорії ймовірностей та математичної статистики УжНУ

*Адам Доровці* – доктор філософії, старший викладач кафедри математики та інформатики Закарпатського угорського інституту ім. Ференца Ракоці II

Відповідальні за випуск:

*Каталін КУЧІНКА* – завідувач кафедри математики та інформатики Закарпатського угорського інституту імені Ференца Ракоці II, доцент, кандидат фізико-математичних наук

*Олександр ДОБОШ* – начальник Видавничого відділу ЗУІ ім. Ф.Ракоці II

За зміст методичних вказівок відповідальність несуть розробники.

**Видавництво: Закарпатський угорський інститут імені Ференца Ракоці II** (адреса:  
пл. Кошута 6, м. Берегове, 90202. Електронна пошта: foiskola@kmf.uz.ua) *Свідоцтво про  
внесення суб'єкта видавничої справи до Державного реєстру видавців, виготовлювачів і  
розповсюджувачів видавничої продукції Серія ДК \_\_\_\_ від \_\_\_\_ 2025 року*

Розмір сторінок методичних вказівок: А4 (210x297мм)