



Selye János Egyetem
Gazdaságtudományi és Informatikai Kar

UDVAROS JÓZSEF

Bevezetés a LATEXbe

Komárom, 2023



Selye János Egyetem
Gazdaságtudományi és Informatikai Kar

UDVAROS JÓZSEF
Bevezetés a LATEXbe

Komárom, 2023

© Udvaros József

Rezenzenti/recenzensek:

Constantin Mihai Avornicului, PhD.

Ákos Gubán, Professor Emeritus

Jazyková korektúra/nyelvi korrektúra: Stefánia, Bódi, Dr. habil.

ISBN 978-808122-459-1

Bevezetés a LATEXbe

1 BEVEZETÉS	6
1.1 A LATEX BEMUTATÁSA.....	6
1.2 JELÖLŐNYELVEK	7
1.2.1 Nyomdai jelölés	7
1.2.2 Logikai jelölés	8
1.3 TEX ÉS UTÓDAI.....	9
1.3.1 A TEX program.....	9
1.3.2 Plain TEX.....	10
1.3.3 LATEX 2 ϵ	10
1.3.4 TEX betűtípusok	10
1.3.5 LATEX felhasználói hozzájárulások.....	11
1.3.6 LATEX és az elektronikus közzététel	11
1.4 A LATEX FÁJL ALAPJAI	12
1.4.1 Szöveg és parancsok	12
1.4.2 LATEX forrásfájl tartalma	12
1.4.3 A LATEX kiterjesztése csomagokkal.....	13
1.5 TEX FELDOLGOZÁSI ELJÁRÁS.....	14
1.5.1 Régebben.....	14
1.5.2 Napjainkban	16
2 SZÖVEG, SZIMBÓLUMOK ÉS PARANCSONK	16
2.1 PARANCSONNEVEK ÉS ARGUMENTUMOK.....	16
2.2 KÖRNYEZETEK	19
2.3 DEKLARÁCIÓK.....	21
2.4 HOSSZÚSÁGOK	22
2.4.2 Változó hosszúságok	24
2.5 SPECIÁLIS KARAKTEREK	25
2.5.1 Szóközök	25
2.5.2 Idézőjelek.....	25
2.5.3 Kötőjelek.....	26
2.5.4 Parancskarakterek nyomtatása	26
2.5.5 Ékezetek.....	26
2.5.6 Különleges karakterek	26
2.5.7 A dátum	27
2.6 SZÖVEG FINOMHANGOLÁSA	28
2.6.1 Szóközök és karakterek.....	28
2.6.2 Sortörés	32
2.6.3 Függőleges térköz.....	34
2.6.4 Oldaltörés	36
2.7 A SZAVAK ELVÁLASZTÁSA	39
2.7.1 Kézi elválasztás.....	39
2.7.2 Elválasztási lista.....	40
2.7.3 Elválasztás elnyomása.....	40
2.7.4 Szóelválasztás többnyelvű szövegben	40
3 DOKUMENTUM ELRENDEZÉSE ÉS FELÉPÍTÉSE	41
3.1 A DOKUMENTUM OSZTÁLY	41

3.1.1 Standard osztályok opciói.....	42
3.1.2 Csomagok használata.....	45
3.1.3 Globális és helyi lehetőségek.....	45
3.1.4 Osztály- és csomagverziók.....	46
3.2 OLDALAK STÍLUSA	46
3.2.1 Címnyilatkozatok	47
3.2.2 Egyedi fej- és lábléc	47
3.2.3 Oldalszámozás.....	48
3.2.4 Bekezdés formázása	49
3.2.5 Oldalformátum	50
3.2.6 Egyszerűsített oldalformázás	53
3.2.7 Egy- és kétszlopos oldalak	55
3.2.8 Többoszlopos szöveg	55
3.3 A DOKUMENTUM RÉSZEI.....	57
3.3.1 Címlap.....	57
3.3.2 Absztrakt	61
3.3.3 Szakaszok.....	62
3.3.4 Függelék	65
3.3.5 A könyv szerkezete	65
3.4 TARTALOMJEGYZÉK	65
3.4.1 Automatikus bevitel.....	65
3.4.2 A tartalomjegyzék kinyomtatása	66
3.4.3 További bejegyzések.....	66
3.4.4 Egyéb listák.....	68
4 SZÖVEG MEGJELENÍTÉSE	68
4.1 BETŰTÍPUS MÓDOSÍTÁSA	68
4.1.1 Kiemelés	68
4.1.2 A betűméret kiválasztása	70
4.1.3 Betűtípus-attribútumok.....	72
4.1.4 Betűtípus-parancsok.....	73
4.1.5 További betűtípusok	74
4.1.6 Karakterkészletek és szimbólumok.....	74
4.2 KÖZPONTOZÁS ÉS BEHÚZÁS.....	74
4.2.1 Középre igazított szöveg.....	74
4.2.2 Egyoldalú igazítás.....	75
4.2.3 Kétoldali behúzás.....	75
4.2.4 Versek behúzása	77
4.3 LISTÁK	79
4.3.1 Számozatlan lista.....	79
4.3.2 Számozott lista	79
4.3.3 Leíró lista	79
4.3.4 Beágyazott listák	82
4.4 DOBOZOK.....	85
4.4.1 LR dobozok	85
4.4.2 Meghatározott magasságú bekezdésdobozok.....	87
4.4.3 Szabálydobozok	88
4.4.4 Beágyazott dobozok	89
4.5 TÁBLÁZATOK	90
4.5.1 Táblázatok készítése.....	90
4.5.2 Táblázatstílus-paraméterek.....	94
4.6 SZÓ SZERINTI SZÖVEG NYOMTATÁSA (VERBATIM).....	94
4.6.1 E-mail és internet címek	96

4.7 LÁBJEGYZETEK ÉS SZÉLJEGYZETEK	97
4.7.1 Szabványos lábjegyzetek	97
4.7.2 Széljegyzetek	99
4.8 MEGJEGYZÉSEK A SZÖVEGBEN.....	99
5 MATEMATIKAI KÉPLETEK.....	100
5.1 MATEMATIKAI KÖRNYEZETEK.....	100
5.2 A MATEMATIKAI MÓD FŐ ELEMEI	103
5.2.1 Állandók és változók	103
5.2.2 Hatványkitevők és indexek	103
5.2.3 Törtek	104
5.2.4 Gyökök.....	105
5.2.5 Összegek és integrálok	107
5.2.6 Folytatási pontok megadása	108
5.3 MATEMATIKAI SZIMBÓLUMOK	108
5.3.1 Görög betűk.....	109
5.3.2 Kalligrafikus betűk.....	111
5.3.3 Bináris operátorok.....	111
5.3.4 Relációk és tagadásai.....	113
5.3.5 Nyilak és mutatók.....	115
5.3.6 Egyéb szimbólumok.....	115
5.3.7 Kétméretű szimbólumok.....	115
5.3.8 Függvények.....	117
5.3.9 Matematikai ékezetek.....	119
5.4 TOVÁBBI ELEMEK	121
5.4.1 A zárójelek automatikus méretezése.....	123
5.4.2 Mátrixok és tömbök.....	125
5.4.3 Vonalak a képletek felett és alatt.....	127
5.4.4 Többsoros egyenletek.....	129
5.4.5 A zárójeles szimbólumok kézi méretezése	131
6 KÉPEK	131
6.1 A GRAFIKUS CSOMAGOK	131
6.1.1 Külső grafikák importálása.....	132
6.1.2 Kép importálása a graphic csomaggal	132
7 FELHASZNÁLT IRODALOM	135

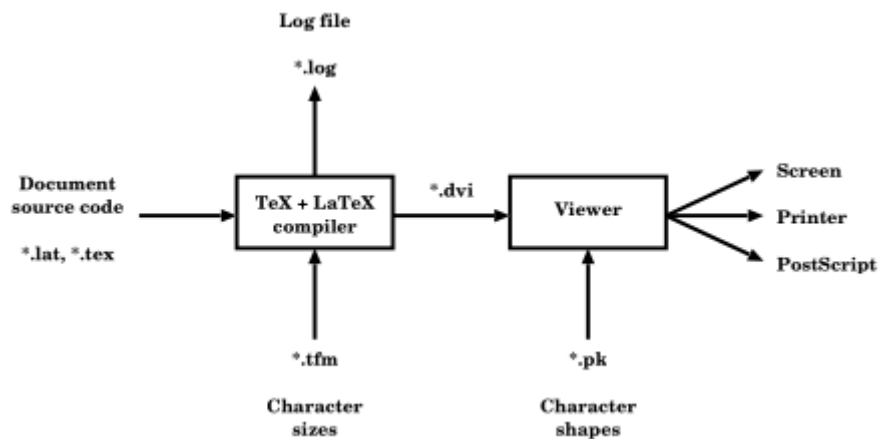
1 Bevezetés

A tankönyv ezen fejezetében bemutatásra kerül a LATEX kialakulása és fejlődése. Szó lesz a jelölőnyelvek jelentőségéről, a TEX feldolgozási eljárásáról, valamint a LATEX felépítéséről.

1.1 A LATEX bemutatása

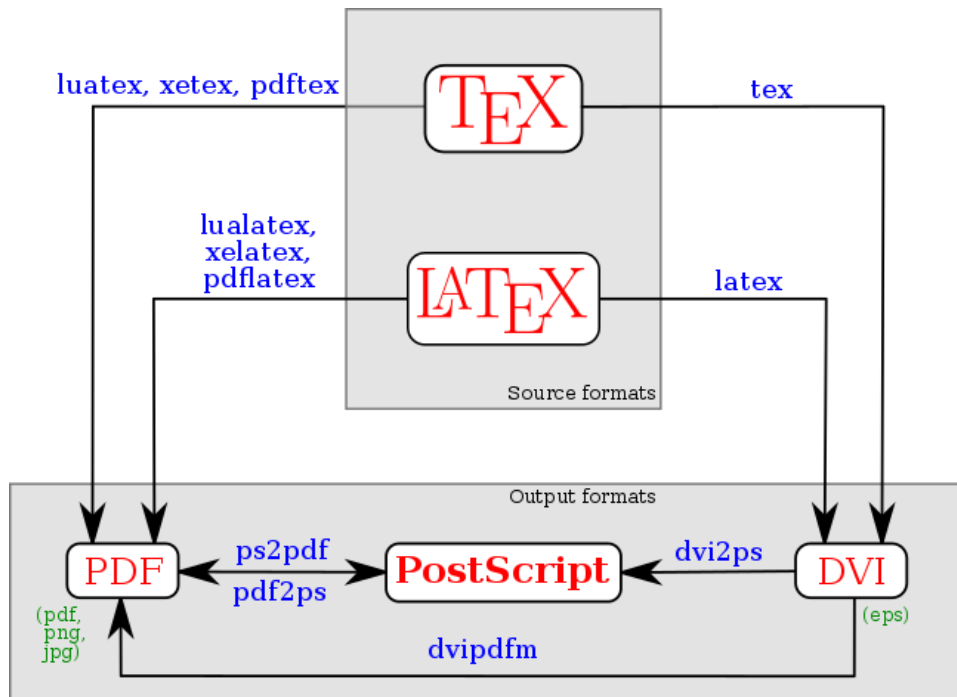
Nagyon röviden összefoglalva:

- A LATEX a jelölőparancsok átfogó készlete, amelyet a hatékony TEX szedőprogrammal használnak a dokumentumok széles skálájának elkészítéséhez, a tudományos cikkektől, jelentésektől egészen az összetett könyvekig.
- A LATEX, hasonlóan a TEX-hez, egy nyílt szoftverrendszer, amely ingyenesen elérhető.
- A LATEX dokumentum egy vagy több forrásfájlból áll, amelyek egyszerű szöveges karaktereket, a tényleges szöveges tartalmat és a jelölőparancsokat tartalmazzák. Ide tartoznak az olyan utasítások, amelyek más programok által készített grafikus anyagokat illeszthetnek be.
- A TEX program feldolgozza, hogy DVI (eszközfüggetlen) formátumú bináris fájlt állítson elő, amely pontos utasításokat tartalmaz az egyes karakterek szedéséről. Ez megtekinthető a monitoron, vagy átalakítható nyomtató-utasításokká, vagy más elektronikus formában, például PostScript, HTML, XML vagy PDF formátumban.



1. ábra. Forrásfájl feldolgozása.

- A TEX program egy pdfTEX nevű változata közvetlenül a forrásfájlból állítja elő a PDF kimenetet anélkül, hogy átmenne a DVI közvetítőn. Ezzel a LATEX automatikusan beépíthet belső hivatkozásokat és könyvjelzőket, valamint PDF gombokat és külső hivatkozásokat, valamint számos általános formátumú grafikát.



2. ábra. Forrásfájl típusok feldolgozása.

1.2 Jelölőnyelvek

1.2.1 Nyomdai jelölés

A számítógép előtti időkben a szerző kézzel vagy írógéppel készített egy kéziratot, amelyet benyújtott egy kiadóhoz. Miután elfogadták közzétételre (és több javítási és módosítási kör után, amelyek mindegyike a papír kézirat átírását igényelte), elküldték egy másolatszerkesztőnek, aki a kéziratot jelölésekkel, széljegyzetekkel bővítette, amelyek tájékoztatják a szedőt, milyen betűtípusokat, szóközöket és egyéb tipográfiai jellemzőket kell használnia ahhoz, hogy azt a könyvektől és cikkektől elvárható végső nyomtatott formára alakítsák.

A szövegek elektronikus feldolgozása manapság hasonló eljárást követ, kivéve, hogy az embert számítógépes programok váltották fel. A jelölést általában közvetlenül a kézirat tartalmazza oly módon, hogy az azonnal kimeneti formájába kerül, és megjelenik a számítógép képernyőjén.

Amit azonban látunk, nem mindig az, amit kapunk. A nagy kiadók által egyre gyakrabban használt alternatíva a jelölőnyelv, amelyben a nyers szöveget a „szedő számára” jelzőkkel bővítik. Ez már nem rövidített széljegyzet, hanem egy rejtélyes kód a tényleges szövegen belül. Ezt az egyszerű szövegszerkesztő programmal elkészíthető forrásszöveget egy külön program alakítja át tipográfiaileg beállított kimenetűvé. Például a következő sor esetében:

Az **oroszlán** az állatok királya.

A HTML a WWW klasszikus jelölőnyelve esetén a forrásszövegbe be kell írni:

Az `oroszlán` az állatok királya.

A PlainTEX-ben ugyanaz a mondat következőképpen volna kódolva:

Az `{\bf oroszlán}` az állatok királya.

Az első példát egy webböngésző programnak kell feldolgoznia (megjelenítenie), amely úgy dönt, hogy a `` és `` között mindent félkövérre állít. A második példa a TEX programhoz készült. A jelölés ebben a két példában különböző szabályokat, eltérő szintaxist követ, de a funkcionalitás ugyanaz.

1.2.2 Logikai jelölés

A fenti példák a tipográfiai jelölést szemléltetik, ahol a beillesztett parancsok vagy címkék közvetlen utasításokat adnak a kimenet megjelenésének megváltoztatására, itt a betűtípus megváltoztatására. Egy másik lehetőség a szöveg céljának feltüntetése. A HTML több szintű címsort alkalmaz, mint például a cím legmagasabb szintre helyezéséhez a következőket kell megadni:

```
<h1>Logikai jelölés</h1>
```

Ezzel egyenértékű LATEX bejegyzés a következő lenne:

```
\section{Logikai jelölés}
```

Ezzel a logikus jelöléssel a szerző teljes mértékben a tartalomra koncentrál, a tipográfiai szempontokat a szakértőkre bízva. Az ember csupán a dokumentum szerkezetét jelöli meg, és nincs mód arra, hogy a logikai elemeket, például a szakaszcímeket hogyan jelenítse meg tipográfiaiilag. Ez az információ HTML stíluslapokba vagy LATEX osztályokba és csomagokba kerül, amelyek kívül esnek a tényleges forrásfájlon. Ez azt jelenti, hogy a dokumentum teljes elrendezése felülvizsgálható a forrásfájl minimális vagy akár semmilyen módosítása nélkül.

Manapság sok erőfeszítést tesznek az XML-re, az Extensible Markup Language-re, mint a végső jelölőrendszerre, mivel lehetővé teszi a jelölések vagy címkék szükség szerinti definiálását anélkül, hogy jeleznék a megvalósítás módját. Ez az XSL-re, az Extensible Stylesheet Language-re maradt. Hangsúlyozni kell, hogy sem az XML, sem az XSL nem program, ezek a dokumentumok és adatbázisok jelölésének módjára vonatkozó előírások, és a jelölőcímkék valós kimenetté való fordításának módja. A tényleges feladat elvégzéséhez még programokat kell találni.

A jelölőnyelvek mögött meghúzódó alapgondolat a következő: a forrásszöveg jelzi tartalmának logikai szerkezetét. Az ilyen egyszerű ASCII-szövegben írt forrásfájlok rendkívül robusztusak, és nem kötődnek semmilyen konkrét szoftvercsomaghoz vagy számítógéptípushoz.

Mi köze ennek az egésznek a LATEX-hez? A következő részben bemutatjuk a TEX és a LATEX fejlődését, hogy a LATEX, az 1980-as évek közepének terméke, egy programozható jelölőnyelv, amely ideálisan megfelel az elektronikus publikálás modern világának.

1.3 TEX és utódai

A tudományos és műszaki művek könyvminőségű szövegének előállítására a legerősebb formázási programot 1969-ben Donald E. Knuth készítette. A program neve TEX, ami a görög $\tau\epsilon\chi$ betűk nagybetűs leképezése. Emiatt az utolsó betűt nem x-nek, hanem ch-nek ejtik. Az elnevezés azt hivatott hangsúlyozni, hogy a matematikai szövegek nyomtatása a program szerves része, nem pedig nehézkes kiegészítő. A TEX mellett ugyanez a szerző kifejlesztett egy további programot METAFONT néven karakterbetűkészletek előállítására. A standard TEX programcsomag 75 betűtípust tartalmaz különböző tervezési méretekből, amelyek mindegyike akár nyolc nagyítási lépésben is elérhető. Mindezek a betűtípusok a METAFONT programmal készültek. További alkalmazásokkal további karakterbetűtípusok készültek, például cirill, kínai és japán nyelvre, amelyekkel ezekben az ábécékben lévő szövegek könyvminőségben nyomtathatók. A TEX program ingyenes, a forráskód pedig könnyen elérhető. Bárki elérheti és tetszés szerint módosíthatja, feltéve, hogy az eredményt nem TEX-nek nevezi. Számos TEX-változat létezik, beleértve a pdfTEX-et is, amellyel később foglalkozunk. Csak Knuth módosíthatja magát a TEX-et, amit csak azért tesz, hogy kijavítsa a nyilvánvaló hibákat. Ellenkező esetben a TEX-et befejezettnek tekinti; a jelenlegi verziószám 3.141592653, halálával a kód örökre lefagy, a verziószám pedig pontosan π lesz.

1.3.1 A TEX program

Az alap TEX program csak nagyon primitív parancsokat ért meg, amelyek a legegyszerűbb szedési műveletekhez és programozási funkciókhoz is megfelelőek. Ez azonban a primitívek szempontjából lehetővé teszi bonyolultabb, magasabb szintű parancsok definiálását. Így az alacsony szintű építőelemekből felhasználóbarátabb környezet építhető fel.

A feldolgozás során a program először egy úgynevezett formátumfájlt olvas be, amely tartalmazza a primitívek szempontjából a magasabb szintű parancsok definícióit, és tartalmazza a szóosztás elválasztási mintáit is. Csak ezután olvassa be a szerző forrásfájlját, amely tartalmazza a ténylegesen feldolgozandó szöveget, beleértve a formátumfájlban előre meghatározott formázási parancsokat is.

Az új formátumok létrehozását a hozzáértő programozókra kell bízni. A definíciók egy forrásfájlba íródnak, amelyet azután a TEX program egy speciális változata, az initex dolgoz fel. Az új formátumú fájlt kompakt módon tárolja, hogy a szokásos TEX program gyorsan be tudja olvasni.

1.3.2 Plain TEX

Knuth a legegyszerűbb szinten egy Plain TEX nevű alapformátumot biztosított a TEX-szel való interakcióhoz. Ez annyira alapvető része a TEX feldolgozásnak, hogy az ember hajlamos elfelejteni a különbséget a tényleges TEX feldolgozó program és az adott formátum között. A legtöbb ember, aki azt állítja, hogy „csak TEX-szel dolgozik”, valójában csak a Plain TEX-szel dolgozik.

A sima TEX minden más formátum alapja is, ami csak erősíti azt a benyomást, hogy a TEX és a Plain TEX egy és ugyanaz.

1.3.3 LATEX 2 ϵ

A LATEX óriási népszerűsége és olyan területekre való terjeszkedése, amelyekre eredetileg nem szánták, valamint a számítástechnikai fejlesztések, különösen az olcsó, de nagy teljesítményű lézernyomtatók terén, sokféle formátumot hozott létre a LATEX címkével. A valódi, továbbfejlesztett szabvány visszaállítása érdekében a LATEX3 Projectet 1989-ben Leslie Lamport, Frank Mittelbach, Chris Rowley és Rainer Schopf hozta létre. Céljuk az volt, hogy optimalizált és hatékony alapparancsokat hozzanak létre, amelyeket különféle csomagok egészítenek ki, hogy szükség szerint speciális funkciókat adhassanak hozzá.

Ahogy a projekt neve is sugallja, LATEX számára a cél a 3. verzió elérése volt. A hosszú távú cél elérése érdekében, 1994 közepén a LATEX 2 ϵ kiadásra került, valamint a Lamport alapkézikönyvének második kiadása is.

Kezdetben a LATEX 2 ϵ frissítéseit évente kétszer, júniusban és decemberben adták ki, de mára annyira stabillá vált, hogy 2000 óta a változtatásokat csak évente egyszer, névlegesen júniusban teszik közzé.

A LATEX 2 ϵ ma már a szabványos verzió, a LATEX 2.09 pedig elavultnak számít, bár a régebbi verzióhoz szánt forrásfájlok továbbra is feldolgozhatók az újabb verzióval. Továbbiakban, ha csak másképp nem jelezzük, a „LATEX” mindig a LATEX 2 ϵ -t jelenti.

1.3.4 TEX betűtípusok

A TEX kezdetben saját, Computer Modern nevű betűkészletét használta, amelyet Knuth METAFONT programja generált. Ennek oka az volt, hogy a nyomtatók akkoriban (és még ma is) tartalmazhatnak saját előre betöltött betűtípusokat, de ezek gyakran nyomtatónként kissé eltérnek. Továbbá hiányoztak belőlük azok a matematikai karakterkészletek, amelyek elengedhetetlenek a TEX fő ismertetőjegyéhez, a matematikai szedéshez. Így Knuth pixel-betűkészletet készített, amelyeket minden nyomtatóra el lehetett küldeni, így mindenhol ugyanazt az eredményt biztosította.

Mára a betűtípusok helyzete drámaian megváltozott. Az Outline betűtípusok (más néven 1-es típusú betűtípusok) kompaktabbak és sokoldalúbbak, mint a pixel-betűtípusok (3-as típus). Ezenkívül sokkal jobb megjelenésűek, és sokkal gyorsabban rajzolódnak ki a PDF-fájlokban. Az eredeti Computer Modern betűtípusokat vázlatos betűtípusokká alakították át, de nincs miért ragaszkodni hozzájuk, kivéve esetleg a matematikai szimbólumok miatt. A LATEX 2 ϵ az új betűtípus-kiválasztási rendszerével szabadította meg a TEX-et a Computer Modernhez fűződő merev kapcsolódásához.

1.3.5 LATEX felhasználói hozzájárulások

A TEX programhoz hasonlóan, amelyre támaszkodik, a LATEX is ingyenes. Lehet, hogy van olyan előítélet, hogy ami ingyen van, az semmit sem ér, de a számítógépes világban más példák is megcáfolják ezt az állítást. És mivel a LATEX makrók egyszerű szöveget tartalmazó fájlokban találhatóak, nem okoz gondot a cseréjük, módosításuk és kiegészítésük. Más szóval, a felhasználó részt vehet az alap LATEX rendszer bővítésében.

Kihasználva a LATEX 2.09-ben található mechanizmust, amely lehetővé tette az alapértelmezett elrendezések opcióinak az úgynevezett stílusbeállítási fájlokban való szerepeltetését, sok felhasználó elkezdett saját „opciókat” írni, hogy további szolgáltatásokat biztosítson az alap LATEX-hez. Azután az interneten keresztül ezeket elérhetővé tették más felhasználók számára is. Sokukat nagyon specifikus problémákra szánták, de mivel több általánosan hasznosnak bizonyult, ezért a szabványos LATEX telepítés részévé vált. Ily módon maguk a felhasználók olyan rendszert építettek ki, amely megfelel az igényeiknek.

A LATEX 2 ϵ -vel ezek a felhasználói hozzájárulások hivatalos státuszt kaptak: csomagként váltak ismertté, közvetlenül a dokumentumba lehetett bevinni, megírásukhoz útmutatókat adtak ki, és további parancsokat vezettek be a csomagok programozásának segítésére. A csomagfájlok a LATEX 2.09 verziójától kezdve .sty kiterjesztést viselnek, így a régebbi stílusbeállítási fájlok ma is csomagként működhetnek.

Azokat a csomagokat, amelyek nélkülözhetetlenek a kifinomult LATEX-feldolgozáshoz, továbbiakban azokban a részekben ismertetjük, ahol a leginkább relevánsak.

1.3.6 LATEX és az elektronikus közzététel

Az elmúlt évtizedek legjelentősebb fejleménye a számítógép-használatban a világháló térnyerése, ezért a LATEX támogatja:

- olyan programokat, melyek a LATEX fájlok HTML formátumba konvertálását biztosítják

- a PDF kimenet létrehozásának eszközeit, hipertext jellemzőkkel, mint például hivatkozások, könyvjelzők, aktív gombok
- az interfészt az XML-hez úgy, hogy motorként működjön az XML dokumentumok megjelenítéséhez, valamint a LATEX-et XML-re konvertáló programokkal és fordítva.

Az elektronikus közzététel ezen formái a hagyományos papírkidadás alternatívái. Az elektronikus nyomtatványok, szükség esetén mindig képesek reprodukálni a papírt.

1.4 A LATEX fájl alapjai

1.4.1 Szöveg és parancsok

A LATEX feldolgozás forrásfájlja, vagy egyszerűen a LATEX fájl, tartalmazza a nyomtatott kimenet előállításához feldolgozandó forrásszöveget. A szöveg egyenlő szélességű sorokra való felosztása, bekezdésekre formázása, oldalszámozással és futófejekkel való oldalakra bontása mind a feldolgozó program feladata, nem pedig magának a bemeneti szövegnek.

Például a forrásszöveg szavai olyan betűsorok, amelyeket nem betűk zárnak le, például írásjelek, szóközök vagy sorvégek (kemény sorvégek, olyanok, amelyek valóban ott vannak, nem pedig lágyak, amelyek mozognak az ablak szélességével); míg az írásjelek átkerülnek a kimenetre, a szóközök és a sorvégek csupán a szavak közötti hézagot jelzik. A többszörös szóközök a bemenetben vagy a sor elejének nincs hatással a szóközökre a kimenetben.

Hasonlóképpen az új bekezdést a beviteli szövegben egy üres sor jelzi; több üres sor ugyanazt a hatást fejt ki, mint egyetlen. A kimenetben a bekezdés formázható vagy az első sor behúzásával, vagy extra sorközökkel, de ezt semmilyen módon nem befolyásolja a bemenetben lévő üres sorok vagy extra szóközök száma.

A forrásfájl azonban nem csak szöveget tartalmaz, a formázást vezérlő vagy a szerkezetet jelző jelölőparancsok is tartózzák. Ezért szükséges, hogy a szerző fel tudja ismerni, mi a szöveg és mi a parancs. A parancsok vagy bizonyos egyedi karakterekből állnak, amelyek nem használhatók szövegkarakterként, vagy olyan szavakból, amelyeket közvetlenül egy speciális karakter, a fordított perjel (\) előz meg.

1.4.2 LATEX forrásfájl tartalma

Minden LATEX fájl tartalmaz egy preambulomot és egy törzset. A preambulom olyan parancsok gyűjteménye, amelyek meghatározzák a következő szöveg globális feldolgozási paramétereit, például a papírformátumot, a szöveg magasságát és szélességét, a kimeneti oldal formáját az oldalszámozással és az automatikus fejlécekkel és láblécekkel. A preambulomnak a dokumentum általános feldolgozási típusának megadásához legalább a `\documentclass` parancsot kell tartalmaznia. Ez az első parancs a preambulumban.

Ha nincs más parancs a preambulumban, a LATEX szabványos értékeket választ ki a sorszámszámhoz, a margókhoz, a bekezdésközkhöz, az oldalmagassághoz és szélességhez és még sok másához. Alapértelmezés szerint ezek a specifikációk az amerikai szabványokhoz vannak igazítva. Az európai követelményeknek megfelelően beépített opciók állnak rendelkezésre a szöveg magasságának és szélességének az A4-es szabványnak megfelelő módosítására. Ezenkívül vannak nyelvspecifikus csomagok bizonyos címsorok, például a „Fejezet” és az „Absztrakt” fordítására.

A preambulumban a következővel végződik: `\begin{document}`. Minden, ami ezt a parancsot követi, testként értelmeződik. Ez a tényleges szövegből áll, jelölőparancsokkal keverve. A preambulumban szereplőkkel ellentétben ezeknek a parancsoknak csak helyi hatása van, vagyis csak a szöveg egy részére vonatkoznak, például behúzásra, egyenletekre, ideiglenes betűtípusváltásra stb. A törzs a `\end{document}` parancssal végződik. Ez általában a fájl vége is.

A LATEX fájl általános szintaxisa a következő:

```
\documentclass[options]{class}
```

További globális parancsok és specifikációk

```
\begin{document}
```

Szöveg keverve további helyi hatásparancsokkal

```
\end{document}
```

A `\documentclass` parancs lehetséges opcióit és osztályait a továbbiakban mutatjuk be.

A `jo_reggelt.tex` nevű minimális LATEX fájl csak a következő sorokat tartalmazza:

```
\documentclass{article}
```

```
\begin{document}
```

Jó reggelt!

```
\end{document}
```

1.4.3 A LATEX kiterjesztése csomagokkal

A csomagok a LATEX nagyon fontos jellemzői. Ezek az alapvető LATEX-parancsok kiterjesztései, amelyek `.sty` végű fájlba íródnak, és a preambulumban a `\usepackage` parancssal töltődnek be. A csomagokat származásuk szerint osztályozhatjuk:

core csomagok a LATEX alaptervezésének szerves részét képezik, ezért teljesen szabványosak;

tools csomagok a LATEX3 csapat tagjai által írt készletek, és mindig a telepítésben kell lenniük;

graphics csomagok szabványos készletek más programok által generált képek felvételére és a színek kezelésére; egy szinten vannak az eszközcsomagokkal;

AMS-LaTeX csomagok az American Mathematical Society által kiadott LATEX csomagoknak minden telepítésben szerepelniük kell;

contributed csomagokat tényleges felhasználók küldték be; ezek közül néhány „alapvetőnek” bizonyult a szabványos LATEX használathoz, de mindegyik hasznos.

Most csak korlátozott számú csomagot ismertetünk, azokat, amelyeket nélkülözhetetlenek tartunk. Azonban semmi sem akadályozza meg a felhasználót abban, hogy olyan egyéb eszközöket szerezzen be és építsen be, amelyek céljai szempontjából előnyösnek bizonyulnak. A beküldött csomagok dokumentálása némileg véletlenszerű, attól függően, hogy a szerző mennyit tett bele. A csomagok terjesztésének előnyben részesített módja az, hogy a kódot tartalmazó dokumentációt egyetlen, *.dtx* kiterjesztésű fájlba integrálja. Egy speciális DocStrip programot használnak a tényleges csomagfájl vagy fájlok kibontására, míg az eredeti *.dtx* fájl LATEX-ezése elkészíti a használati útmutatót. A legtöbb futtatásra kész telepítés már megtette mindezt a felhasználó helyett, az így kapott kézikönyveket DVI vagy PDF fájlként tárolja valahol a `\texmf\doc\latex\` mappában. Előfordulhat azonban, hogy nekünk kell létre hozni a *dokumentációs* kimenetet, melyet a *.dtx* fájl feldolgozásával végezhetjük el, amely a `\texmf\source\latex\` mappában található.

Egyes csomagszerzők *extra.tex* fájlként írják meg kézikönyveiket, amelyek kimenete előre tárolható DVI vagy PDF formátumban. Mások HTML fájlokat biztosítanak, vagy pedig egyszerűen hozzáfűzik az utasításokat megjegyzésként magában a csomagfájlban.

1.5 TEX feldolgozási eljárás

Mivel a LATEX a TEX program definícióinak halmaza, maga a LATEX feldolgozás valójában nem más, mint TEX feldolgozás LATEX formátummal. Az összes szedési munkát a TEX végzi, míg a LATEX kezeli a logikai jelölésből a szedőparancsokká történő átalakítást. Lehetővé teszi továbbá a kereszthivatkozásokat, a címsorok futtatását, a tartalomjegyzéket, a szakirodalmi hivatkozásokat és az irodalomjegyzéket, az indexelést és egyebeket. A forrásfájl feldolgozása a végső kimenetig azonban a TEX feladata, függetlenül a használt formátumtól.

1.5.1 Régebben

A TEX több mint 30 évvel ezelőtt jelent meg, mielőtt léteztek olyan eszközök, mint a PC-k, a grafikus kijelzők, és mielőtt a számítógépeket kiegészítették volna grafikus interfésszel. A TEX és a támogató programjai parancssorból lettek meghívva, nem egérekattintással. Lehet, hogy ez nagyon régmódinak hangzik, de biztosítja minden számítógéptípushoz a hordozhatóságot.

Az akkori feldolgozási lépések még ma is léteznek a mai grafikus felületeken, de ma már kényelmesebben végrehajthatók. Még mindig meg lehet nyitni egy "parancssor ablakot", és futtatni a parancssorból.

Az első lépés természetesen az, hogy egy szövegszerkesztő programmal megírjuk a tényleges szöveget és jelölést tartalmazó forrásfájlt. Ez egy szövegfájlba kerül, vagy amit gyakran „ASCII” fájlnak neveznek, amely csak szabványos írásjeleket, számokat, ékezetes betűket, nagy- és kisbetűket tartalmaz. Más szóval, a szöveg az, amit szabványos írógéppel lehet előállítani.

A forrásfájl neve általában *.tex* kiterjesztésű; akkor a TEX feldolgozza, hogy egy új eszközfüggetlen fájl hozzon létre azonos alapnévvel és *.dvi* kiterjesztéssel. Ez egy bináris fájl, amely pontos utasításokat tartalmaz az egyes szimbólumok kiválasztásához és elhelyezéséhez, valamint a végső nyomtatott oldal kódolt leírását. A TEX meghívására a parancs a *jo_reggelt.tex* forrásfájllal a következő:

```
tex &latex jo_reggelt
```

Ez azt jelenti, hogy futtassa a TEX programot latex formátumban. Általában a telepítés egy latex nevű parancsikont definiált ehhez, tehát:

```
latex jo_reggelt
```

Csak akkor szükséges megadni a forrásfájlnév kiterjesztését, ha az nem *.tex*.

A feldolgozás során a TEX információkat, figyelmeztetéseket, hibaüzeneteket ír a számítógép képernyőjére és egy *.log* kiterjesztésű átiratfájlba. Érdemes megnézni ezt a fájlt, ha váratlan eredmények jelennek meg.

Az utolsó lépés a DVI fájlból történő nyomtatott oldalak előállítása. Ehhez egy másik program, egy illesztőprogram szükséges, amely az adott nyomtatóra jellemző utasításokat generálja. Például egy PostScript fájl létrehozásához le kell futni

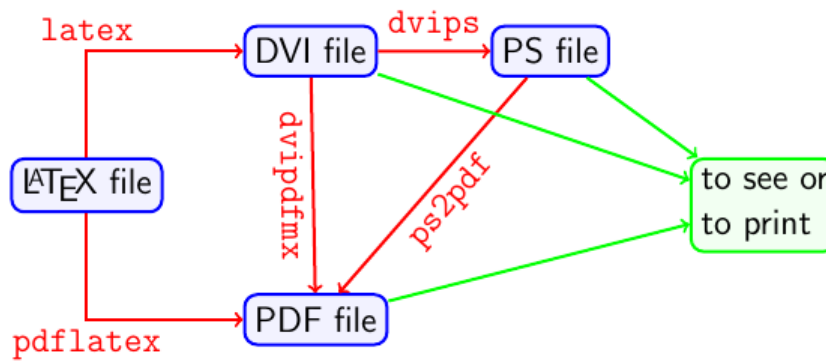
```
dvips jo_reggelt
```

parancsnak, hogy *jo_reggelt.ps*-t kapjunk a *jo_reggelt.dvi*-ből. Ezután elküldjük a *jo_reggelt.ps* fájlt a PostScript nyomtatónak a szokásos paranccsal az adott számítógépes rendszerben.

A DVI-fájl nyomtatás előtti előnézete számítógépképernyőn egy későbbi fejlesztés volt, amely kiváló minőségű grafikus megjelenítést igényel. Ezek a programok alapvetően speciális illesztőprogramok, amelyek a kimenetet közvetlenül a képernyőre küldik, nem pedig nyomtatóba vagy nyomtatófájlba. Az egyik nagyon népszerű előnézetet a következő paranccsal hívják meg:

```
xdvi jo_reggelt
```

hogy megtekinthessük a *jo_reggelt.dvi*-t, mielőtt papírra vetnénk.



3. ábra. A LATEX forrásfájl megjelenítési folyamata.

1.5.2 Napjainkban

A LATEX feldolgozás fent ismertetett lépései ma is szükségesek, és megnyithatunk egy parancssori ablakot, és ugyanúgy végrehajthatjuk azokat, mint korábban. Azonban ma már léteznek olyan intelligens, LATEX-tudással rendelkező szerkesztők, amelyek nem csak a forrásszöveg írását segítik, hanem egy egérkattintással meghívják a különböző programokat, TEX-et, előnézeti programot, nyomtató-illesztőprogramot, BIBTEX-et, MakeIndexet. Ilyen program például a TeXStudio.

2 Szöveg, szimbólumok és parancsok

A LATEX feldolgozási bemeneteként szolgáló szöveg egy *.tex* végződésű forrásfájlba kerül, a fájlnevkiterjesztéssel. Ez a fájl szövegszerkesztővel készült, amely vagy az egyszerű szöveget kezeli, vagy olyan, amely úgy van beállítva, hogy segítse a LATEX fájlok írását és feldolgozását. Mindkét esetben a fájl tartalma csak sima ASCII karakterekből áll, speciális szimbólumok és ékezetes betűk nélkül, lehetőleg rögzített szélességű írógép betűtípussal, olyan formázások nélkül, mint a félkövér vagy dőlt betű, mindez egy méretben. A valódi szedés mindezen szempontjait utólag a TEX feldolgozó program állítja elő a szövegbe jól láthatóan beillesztett jelölőparancsok segítségével. Ezért létfontosságú tudni, hogy a parancsok hogyan különböztethetők meg a nyomtatandó szövegtől, és természetesen hogyan működnek.

2.1 Parancsnevek és argumentumok

A parancs a LATEX-nek egy utasítás, hogy tegyen valami különlegeset, például nyomtasson olyan szimbólumot vagy szöveget, amely nem elérhető a bemeneti fájlban használt korlátozott karakterkészlet számára, vagy módosítsa az aktuális betűtípust vagy más formázási tulajdonságokat. Háromféle parancsnév létezik:

- speciális karakterek # \$ & ~ ^ % ~ { } mindegyiknek különleges jelentése van;

- a backslash karakter `\` plusz egyetlen nem betű karakter; például `\$` a `$` karakter kinyomtatásához; az összes fent felsorolt speciális karakterhez tartozik egy megfelelő kétkarakteres parancs a szó szerinti kinyomtatáshoz;

Ha ezeket a karaktereket megszeretnénk jeleníteni, akkor a következő parancsokat használjuk:

<code>\</code>	<code>\textbackslash</code>
<code>%</code>	<code>\%</code>
<code>{</code>	<code>\{</code>
<code>}</code>	<code>\}</code>
<code>\$</code>	<code>\\$</code>
<code>&</code>	<code>\&</code>
<code>#</code>	<code>\#</code>
<code>_</code>	<code>_</code>
<code>^</code>	<code>\textasciicircum</code>
<code>~</code>	<code>\textasciitilde</code>

- a backslash karakter `\` plusz egy betűsorozat, amely az első nem betűvel végződik; például a `\large` nagyobb betűtípusra váltáshoz. A parancsnevek megkülönböztetik a kis- és nagybetűket, `\large`, `\Large` és `\LARGE` mind külön parancsok.

Sok parancs néhány rövid szövegrészen működik, amely ezután argumentumként jelenik meg a parancs nevét követő kapcsos zárójelben. Például az `\emph{Einstein}` a kihangsúlyozott szót hangsúlyos betűtípussal (itt dőlt betűvel) nyomtatja ki. Az ilyen argumentumokat kötelezőnek mondják, mert mindig meg kell adni őket.

Egyes parancsok opcionális (*optional*) argumentumokat vesznek fel, amelyeket általában a parancs hatásainak módosítására használnak. Az opcionális argumentumok szögletes zárójelben jelennek meg.

A továbbiakban a parancsok általános szintaxisát mutatjuk be:

```
\name[optional]{mandatory}
```

ahol a karaktereket pontosan az ábrán látható módon kell beírni. A dőlt szöveg olyasmit jelez, amit helyettesíteni kell. Az opcionális argumentumok szögletes zárójelbe `[]`, a kötelezőek pedig kapcsos zárójelbe `{ }` kerülnek. Egy parancsnak több opcionális argumentuma is lehet, mindegyik a zárójelben, a megadott sorrendben kell megadni. Ha az opcionális argumentumok egyikét sem használja, a szögletes zárójelek elhagyhatók. Tetszőleges számú üres hely, vagy akár egyetlen új sor is megjelenhet a parancs neve és az argumentumok között az olvashatóság javítása érdekében.

Egyes parancsokhoz több kötelező argumentum is tartozik. Mindegyiket egy { } párba kell tenni, és a sorrendjüket a parancsleírás szerint kell fenntartani. Például,

```
\rule[lift]{width}{height}
```

egy fekete méretű szélességű és magasságú téglalapot hoz létre, amelyet az aktuális alapvonal fölé emel, megadott értékkel. Egy 20 mm széles és 5 mm magas téglalap készül a `\rule{20mm}{5mm}` parancs segítségével. Mivel az opcionális argumentumemelés kimarad, a téglalap az alapvonalra kerül emelés nélkül. Az argumentumoknak a szintaxis által meghatározott sorrendben kell megjeleníteniük, és nem cserélhetők fel.

Egyes parancsoknak a szokásos megjelenésük mellett van egy úgynevezett *-alakja is. A nevükhöz egy * jelet adunk, hogy valamilyen módon módosítsák a funkcionalitásukat. Például a `\section` parancsban van egy *, mint `\section*`, amely a normál úrlappal ellentétben nem nyomtat automatikus szakaszszámot. A parancsnevek csak betűkből állnak, az első nem betű a név végét jelzi. Ha opcionális vagy kötelező argumentumok vannak a parancs neve után, akkor a [vagy a { zárójel előtt végződik, mivel ezek a karakterek nem betűk. Sok parancs azonban nem tartalmaz argumentumot, és csak egy névből áll, mint például a `\LaTeX` parancs, amely a LATEX logót hozza létre. Ha egy ilyen parancsot írásjel követ, például vessző vagy pont, akkor nyilvánvaló, hogy hol végződik a parancs. Ha normál szó követi, akkor a parancs neve és a következő szó közötti üres hely a parancs lezárójaként értelmeződik.

Ha csak névből álló parancs után szóközt szeretne beszúrni, akkor vagy egy üres {} szerkezetet vagy egy szóköz parancsot (\ és szóköz) kell a parancs után elhelyezni. A „LATEX logó” előállításának megfelelő módja a `\LaTeX{}` logó vagy a `\LaTeX\` logó beírása.


Feladat:


Hozd létre a következő szövegrészt!

L^AT_EXlogó

A következő speciális karaktereket a mellettük levő parancssal írjuk ki:

```
\ \textbackslash
% \%
{ \{
} \}
$ \$
& \&
# \#
- \_
- \textasciicircum
- \textasciitilde
```

Fekete téglalap, melynek szélessége 2 cm és magassága: 1 cm 

Hasonló téglalap, csak piros színű:  Ebben az esetben szükséges az xcolor csomag használata.

Megoldás:

```
\documentclass{article}
\usepackage{xcolor}

\begin{document}

\begin{center}
\LaTeX logó\
\end{center}

\begin{flushleft}
A következő speciális karaktereket a mellettük levő paranccsal írjuk ki:
\verb |\ \textbackslash|
\verb |% \ %|
\verb |{ \ {|
\verb |} \ }|
\verb |$ \ $|
\verb |& \ &|
\verb |# \ #|
\verb |_ \ _|
\verb |^ \textasciicircum|
\verb |~ \textasciitilde|

Fekete téglalap, melynek szélessége 2 cm és magassága: 1 cm \rule{2cm}{1cm}
Hasonló téglalap, csak piros színű: {\color{red}\rule{2cm}{1cm}} Ebben az esetben
szükséges az xcolor csomag használata.
\end{flushleft}
\end{document}
```

2.2 Környezetek

A környezet a `\begin{name}` paranccsal indul, és a `\end{name}` paranccsal fejeződik be.

Egy környezetnek az a hatása, hogy a benne lévő szöveget a környezeti paraméterek szerint eltérően kezelik. Lehetőség van bizonyos feldolgozási jellemzők (ideiglenes) megváltoztatására, mint például a behúzás, a vonalszélesség, a betűtípus és még sok más. A változtatások csak a környezetben belül érvényesek. Például az idézetkörnyezettel (*quote*)

előző szöveg

```
\begin{quote}
```

```
szöveg1 \small szöveg2 \bfseries szöveg3
```

```
\end{quote}
```

további szöveg

a bal és a jobb margó megnőtt az *előző* és a *további* szövegekhez képest. A példában ez a *szöveg1*, *szöveg2* és *szöveg3* szövegre vonatkozik. A *szöveg1* után jön a `\small` parancs, aminek

hatására a következő *szöveg2* kisebb betűtípussal kerül beállításra. A *szöveg2* után van egy további `\bfseries` parancs a félkövér betűtípusra váltáshoz. Mindkét parancs csak az `\end{quote}`-ig marad érvényben.

Az idézetkörnyezeten belüli három szöveg mindkét oldalán behúzott az előző és a *további* szövegekhez képest. A *szöveg1* normál betűtípussal jelenik meg, ugyanazzal, mint a környezetén kívül. A *szöveg2* és *szöveg3* kisebb betűtípussal, a *szöveg3* pedig ezen felül félkövér betűtípussal jelenik meg.

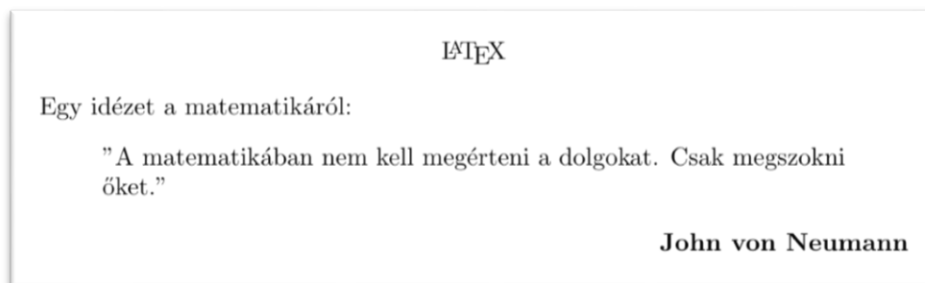
Az idézetkörnyezet vége után a következő szöveg ugyanazzal a betűtípussal jelenik meg, mint amilyen korábban volt.

Ha a környezet neve a `\begin{...}` `\end{...}` parancspárban nem egyezik, a feldolgozás során hibaüzenet jelenik meg.

A legtöbb deklarációs parancsnév környezetnévként is használható. Ebben az esetben a parancs nevét a megelőző `\` karakter nélkül használjuk. Például az `\em` parancs hangsúlyos, általában dőlt betűtípusra vált, és a megfelelő környezetre vált a `\begin{em}` az összes szöveget dőlt betűvel szedi, amíg el nem éri az `\end{em}` parancsot.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass{article}
\begin{document}

\begin{center}
\LaTeX \\
\end{center}

\begin{flushleft}
Egy idézet a matematikáról:
\begin{quote}
"A matematikában nem kell megérteni a dolgokat. Csak megszokni őket."
\end{quote}
\end{flushleft}
```

```
\begin{flushright}
\bfseries John von Neumann
\end{flushright}

\end{document}
```

2.3 Deklarációk

A deklaráció olyan parancs, amely szöveg nyomtatása nélkül módosítja bizonyos paraméterek vagy parancsok értékét vagy jelentését. A deklaráció hatása azonnal megkezdődik, és akkor ér véget, amikor egy másik, azonos típusú deklarációt találunk. Ha azonban a deklaráció egy környezetben vagy egy {...} páron belül történik, akkor hatóköre csak a megfelelő \end parancsra vagy a záró kapcsos zárójelre } terjed ki. Az előző részben említett \bfseries és \small parancsok példák olyan nem nyomtatható deklarációkra, amelyek megváltoztatják az aktuális betűtípust.

Példák:

```
{\bfseries Ez a szöveg félkövérrel van szedve}
```

A \bfseries deklaráció megváltoztatja a betűtípust: Ez a szöveg félkövérrel van szedve. Ennek a deklarációnak a hatása a záró kapcsos zárójellel ér véget.

```
\setlength{\parindent}{0,5cm}
```

A bekezdés behúzása 0,5 cm-re van állítva. Ennek a deklarációnak a hatása a \setlength{\parindent} parancs következő találkozásával ér véget, de legkésőbb az \end parancssal, amely lezárja az aktuális környezetet.

```
\pagenumbering{roman}
```

Az oldalszámozást római számokkal nyomtatja.

Egyes deklarációk globálisak (például az utolsó példa), azaz hatásaik nem korlátozódnak az aktuális környezetre. A következő deklarációk is hasonlóak:

```
\newcounter \pagenumbering \newlength \setcounter
\thispagestyle \newsavebox \addtocounter
```

Ezekkel a parancsokkal létrehozott deklarációk azonnal hatályba lépnek, és mindaddig érvényesek maradnak, amíg felül nem írják őket egy azonos típusú új deklarációval. A fenti utolsó példában az oldalszámozás római számokkal történik mindaddig, amíg egy új \pagenumbering{arabic} parancs felül nem bíráljuk.

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

Ez a szöveg félkövérrel van szedve.
Ez a szöveg *dólt betűstílusban* lesz megjelenítve.
A bekezdés behúzása 1 cm-re van állítva.

Megoldás:

```
\documentclass{article}
\begin{document}

\begin{center}
\LaTeX \\
\end{center}

\begin{flushleft}
{\bfseries Ez a szöveg félkövérrel van szedve.} \\
Ez a szöveg \textit{dólt betűstílusban} lesz megjelenítve. \\
\setlength{\parindent}{1cm}
A bekezdés behúzása 1 cm-re van állítva.
\end{flushleft}

\end{document}
```

2.4 Hosszúságok

A hosszúságok egy decimális számból állnak, amely előtt lehetséges egy előjel (+ vagy -), amelyet egy kötelező méretegység követ. A megengedett mértékegységek és rövidített neveik a következők:

- cm** centiméter,
- mm** milliméter,
- in** hüvelyk (1 hüvelyk = 2,54 cm),
- pt** pont (1 hüvelyk = 72,27 pt),
- bp** nagy pont (1 hüvelyk = 72 bp),
- em** egy betűtípus-specifikus méret, a nagybetű M szélessége,
- ex** egy másik betűméret, az x betű magassága.

A TEX-ben és a LATEX-ben a tizedes számokat angol vagy európai módon is megadhatjuk, ponttal vagy vesszővel: 12.5cm és 12,5cm is megengedett.

A 0 nem megfelelő hosszúság, mivel hiányzik az egység specifikációja. A nulla hosszúság megadásához hozzá kell adni néhány mértékegységet, például 0pt vagy 0cm.

Az értékeket a LATEX-ben a `\setlength` paranccsal hozzárendeljük egy hosszparaméterhez. A szintaxisa a következő:

`\setlength{\length name}{length spec}`

Például egy szövegsor szélességét a `\textwidth` paraméter határozza meg, amely általában az osztálytól, a dokumentumtípustól és a betűmérettől függően alapértelmezett értékre van állítva. A sor szélességének 10cm-re történő módosításához a következőket kell megadni:

`\setlength{\textwidth}{10cm}`

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

Ez a szöveg nincs behúzva, és van 10 pontnyi szünet az bekezdések között.

Ez a dokumentum 9 hüvelyk magas és 6 hüvelyk széles.

Az oldal bal margója 0,25 hüvelyk, és az oldal páros és páratlan oldalai között nincs eltérés a margókban.

A kódban a `\setlength` parancsot használjuk az alapértelmezett behúzás és sortávolság beállítására (`\parindent` és `\parskip` paraméterekkel), a dokumentum méretének beállítására (`\textheight` és `\textwidth` paraméterekkel), valamint az oldal margóinak beállítására (`\oddsidemargin` és `\evensidemargin` paraméterekkel).

Megoldás:

```
\documentclass{article}

\setlength{\parindent}{0pt}
\setlength{\parskip}{10pt}
\setlength{\textheight}{9in}
\setlength{\textwidth}{10cm}
\setlength{\oddsidemargin}{0.25in}
\setlength{\evensidemargin}{0.25in}

\begin{document}
\begin{center}
\LaTeX \\
\end{center}

Ez a szöveg nincs behúzva, és van 10 pontnyi szünet az bekezdések között.

Ez a dokumentum 9 hüvelyk magas és 6 hüvelyk széles.

Az oldal bal margója 0,25 hüvelyk, és az oldal páros és páratlan oldalai között nincs eltérés a margókban.

A kódban a \verb|\setlength| parancsot használjuk az alapértelmezett behúzás és sortávolság beállítására (\verb|\parindent és \verb|\parskip| paraméterekkel), a
```


dokumentum méretének beállítására (`\verb|\textheight` és `\textwidth|` paraméterekkel), valamint az oldal margóinak beállítására (`\verb|\oddsidemargin` és `\evensidemargin|` paraméterekkel).

```
\end{document}
```

2.4.2 Változó hosszúságok

Egyes paraméterek változó hosszúságot várnak el. Ezek egy bizonyos mértékben nyújtható vagy zsugorítható hosszúságok. A szintaxisa a következő:

nominal value plus *stretch value* minus *shrink value*

ahol a *nominal value*, a *stretch value* és a *shrink value* egy-egy hossz.

Például a `\setlength{\parskip}{1ex plus0.6ex minus0.3ex}` azt jelenti, hogy a bekezdések közötti extra sortávolság, az úgynevezett `\parskip`, az x magassága az aktuális betűtípusban, de növelhető 1,6-szorosára vagy csökkenthető ennek 0,7-szeresére.

Egy speciális változó hossz a `\fill`. Ennek megvan a természetes hossza a nulla, de tetszőleges méretre nyújtható.

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

A bekezdések közötti térköz alapértelmezett (nominal) értéke 12pt. A plusz érték (2pt) azt jelenti, hogy ha van szükség a térköz növelésére (pl. a dokumentum egészének jobb elrendezése miatt), akkor a Latex maximum 2 ponttal növeli a bekezdések közötti térközt.

A mínusz érték (2pt) azt jelenti, hogy ha a dokumentum soraiból kevesebb hely áll rendelkezésre (pl. a dokumentum oldalra való szűkítése miatt), akkor a latex minimum 2 ponttal csökkenti a bekezdések közötti térközt.

Megoldás:

```
\documentclass{article}

\setlength{\parskip}{12pt plus 2pt minus 2pt}

\begin{document}
\begin{center}
\LaTeX \\
\end{center}
```

A bekezdések közötti térköz alapértelmezett (nominal) értéke 12pt. A plusz érték (2pt) azt jelenti, hogy ha van szükség a térköz növelésére (pl. a dokumentum egészének

jobb elrendezése miatt), akkor a Latex maximum 2 ponttal növeli a bekezdések közötti térközt.

A mínusz érték (2pt) azt jelenti, hogy ha a dokumentum soraiból kevesebb hely áll rendelkezésre (pl. a dokumentum oldalra való szűkítése miatt), akkor a latex minimum 2 ponttal csökkenti a bekezdések közötti térközt.

```
\end{document}
```

2.5 Speciális karakterek

2.5.1 Szóközök

A szóköznek vagy üres karakternek van néhány tulajdonsága, amelyek eltérnek a normál karakterektől. A feldolgozás során a beviteli szövegben lévő üres helyeket változó hosszúságok, hogy a sor a teljes sorszélességig kitölthető legyen. Ennek eredményeként bizonyos hatások léphetnek fel, ha valaki nem ismeri a következő szabályokat:

- egy szóköz ugyanaz, mint ezer, csak az első számít;
- a beviteli sor elején lévő szóközöket figyelmen kívül hagyja;
- a parancsnevet lezáró üres karakterek (szóközök) eltávolításra kerülnek;
- a sor végét üresnek kell tekinteni.

Ezeknek a szabályoknak az a következménye, hogy tetszőleges számú szóköz lehet a szavak között vagy a sor elején (a bevitt szöveg olvashatóbbá tétele érdekében), és egy szó után közvetlenül a sor végén kerülhet szóköz és a következő szóközök eltűnnek. Ahhoz, hogy egy szóköz megjelenjen ott, ahol egyébként figyelmen kívül hagynák, ki kell adni a `\szóköz` parancsot.

Annak biztosítására, hogy bizonyos szavak együtt maradjanak ugyanabban a sorban, egy rögzített szóköz kerül közéjük a `~` karakterrel. A több rögzített szóközt a rendszer kinyomtatja, ellentétben a normál szóközökkel.

Néha el kell nyomni az új sor miatt megjelenő szóközt. Ebben az esetben a sor utolsó karakterének a `%` megjegyzés karakternek kell lennie.

A forrásszövegben a bekezdéseket üres sorok választják el. Ami az üres karaktereket illeti, egy üres sor annyi, mint ezer.

Üres sor helyett a `\par` parancs is használható a bekezdés végének jelzésére.

2.5.2 Idézőjelek

Az írógépen található idézőjelek nem használatosak a könyvnyomtatásban. Ehelyett különböző karakterek szerepelnek az elején és a végén, például „egyszeres idézőjelek” és „dupla

idézőjelek”. A megfelelő karakterek kétszeri begépelésével készülnek: " a "" és " helyett. Ezt azonban kerülni kell, mert zavart okozhat.

2.5.3 Kötőjelek

A könyvnyomtatásban az írógépen ’-’ ként megjelenő karakter, különböző hosszúságú: -, –, —. Ezek közül a legkisebbet, a kötőjelet az összetett szavakra, például a sorvégi szóosztásra használjuk; a közepes méretű, az ’-’ kötőjelet számtartományokban használják; a legnagyobbat, az ’—’ kötőjelet pedig írásjelként használják – amit általában kötőjelnek neveznek. Ezek a kötőjel karakter egyszeri, kétszeri vagy háromszori beírásával jönnek létre, így a - esetén - eredményt kapjuk, míg -- esetében – készít, és --- esetén — hoz létre. A kötőjel negyedik típusa a mínuszjel −, amelyet matematikai módban \$-\$-ként írunk be.

2.5.4 Parancskarakterek nyomtatása

Ahogy az előzőekben említettük, a # \$ ~ _ ^ % { } karakterek parancsként értelmeződnek. Szöveggént való kinyomtatáshoz \ plusz egy karakterből álló parancsot kell adni.

\$ = \ \$	& = \ &	% = \ %	# = \ #	_ = \ _
{ = \ {	} = \ }			

2.5.5 Ékezetek

A különböző nyelvekben számos diakritikus vagy ékezetes jel található, amelyek többsége TEX-szel nyomtatható:

ò = \ ‘ {o}	ó = \ ’ {o}	ô = \ ^ {o}	ö = \ " {o}	õ = \ ~ {o}
õ = \ = {o}	ô = \ . {o}	ö = \ u {o}	õ = \ v {o}	õ = \ H {o}
oo = \ t {oo}	o = \ c {o}	o = \ d {o}	o = \ b {o}	o = \ r {o}

A fenti „o” betű csak példaként szolgál, bármilyen betű használható.

2.5.6 Különleges karakterek

e	\euro	€	eurosym
Ł	\pounds		
˘	\textcent	€	textcomp
®	\textregistered	€	textcomp
©	\textcopyright	€	textcomp
†	\dag		
‡	\ddag		
*	\textasteriskcentered		
•	\textbullet		
◦	\textopenbullet	€	textcomp
‰	\textperthousand	€	textcomp

§ \S

№ \textnumero ∈ textcomp

※ \textreferencemark ∈ textcomp

A legtöbb különleges karakter megtalálható a textcomp, illetve az eurosym csomagban. A textcomp csomagot be kell tölteni a preambulumba `\usepackage{textcomp}` paranccsal, amely számos parancsot határoz meg, beleértve a `\texteuro`-t az € szimbólum kinyomtatására. Mivel az Európai Bizottság eredetileg azt diktálta, hogy csak *sans serif* betűtípussal nyomtatható, jobb, ha beállítjuk a `\textsf{\texteuro}`-t, hogy az €-t hozzon létre.

2.5.7 A dátum

Az aktuális dátum a `\today` paranccsal a szöveg bármely pontjára elhelyezhető. A dátum szabványos formája a hónap, nap, év amerikai stílusa (például január 12, 2022.). A brit formátumot (12. január 2022) vagy más nyelven a dátumot a `\day`, `\month` és `\year` TEX parancsokkal lehet előállítani, amelyek ezeknek a paramétereknek az aktuális értékeit adják vissza számként.

Valójában jobb explicit módon megadni a dátumot, mint a `\today`-ra hagyatkozni. Egy többéves LATEX-forrásfájl újrafeldolgozása a feldolgozás dátumával, nem pedig a szöveg írásának eredeti létrehozásának dátumát eredményezi.

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

A különböző nyelvekben számos diakritikus vagy ékezetes jel található, amelyek többsége TEX-szel nyomtatható:

ó \'{o} ő \H{o} ö \"{o} ò \' {o} ô \^ {o} õ \~ {o} ò \={o}
ó \. {o} ö \u {o} ö \v {o} ô \r {o} o \c {o} o \d {o} o \b {o}
ôo \t {oo}

A fenti "o" betű csak példaként szolgál, bármilyen betű használható.

Különleges karakterek:
€, £, ¢, ®, ©, †, ‡, *, •, °, ‰, §, №, ※

Aktuális dátum amerikai formátumban: April 25, 2023

Megoldás:

```
\documentclass{article}
\usepackage{textcomp}
\usepackage{eurosym}

\begin{document}
\begin{center}
```

```
\LaTeX \\
\end{center}
```

A különböző nyelvekben számos diakritikus vagy ékezetes jel található, amelyek többsége TEX-szel nyomtatható:\par

```
\{o} \verb| \{o} |
\H{o} \verb| \H{o} |
\"{o} \verb| \"{o} |
\`{o} \verb| \`{o} |
\^{o} \verb| \^{o} |
\~{o} \verb| \~{o} |
\={o} \verb| \={o} |\par
\.{o} \verb| \.{o} |
\u{o} \verb| \u{o} |
\v{o} \verb| \v{o} |
\r{o} \verb| \r{o} |
\c{o} \verb| \c{o} |
\d{o} \verb| \d{o} |
\b{o} \verb| \b{o} |\par
\t{oo} \verb| \t{oo} |
```

A fenti "o" betű csak példaként szolgál, bármilyen betű használható.

Különleges karakterek:

```
\euro, \pounds, \textcent, \textregistered, \textcopyright, \dag, \ddag,
\textasteriskcentered, \textbullet, \textopenbullet, \textperthousand, \S, \textnumero,
\textreferencemark
```

Aktuális dátum amerikai formátumban: \today

```
\end{document}
```

2.6 Szöveg finomhangolása

A szöveg finomhangolása tisztán tipográfiai jelölésekre vonatkozik, és semmi köze ahhoz a logikai jelöléshez, amelyet ebben a könyvben hangsúlyozni kívánunk. Sajnos előfordul, hogy a szerzőnek vagy a szerkesztőnek segítenie kell a szedőprogramot a jó megjelenés érdekében.

2.6.1 Szóközök és karakterek

A szavak és karakterek közötti térközt általában automatikusan a TEX állítja be, ami nemcsak a karakterek természetes szélességét használja ki, hanem bizonyos karakterkombinációk változásait is figyelembe veszi. Például egy A és egy V után nem AV-ként, hanem AV-ként jelenik meg; vagyis kissé össze vannak húzva a kellemesebb megjelenés érdekében. Az egy soron belüli szóköz egyenletes, és úgy van megválasztva, hogy a jobb és a bal vége pontosan

illeszkedjen az oldalmargókhoz. Ezt bal- és jobboldali igazításnak nevezzük. A TEX arra is törekszik, hogy a szóközt a különböző soroknál a lehető legnagyobb mértékben azonosan tartsa. Az írásjellel végződő szavak a karaktertől függően extra szóközt kapnak: a „” pont vagy a „!” felkiáltójel után több szóköz van, mint a „,” vessző után. Ez megfelel az angol szedés szabályának, miszerint extra szóközt kell hagyni a mondatok között. Bizonyos esetekben az automatikus eljárások nem működnek megfelelően, vagy kívánatos felülbírálni őket.

A mondat befejezése és szóközök

A TEX a kisbetűt követő pontot a mondat végeként értelmezi, ahol további szóközt kell beilleszteni. Ez összetéveszthető az olyan rövidítésekkel, mint az i. e., Dr. Sleep vagy Tud. Akad., ahol a normál térköz szükséges. Ezt úgy érhetjük el, hogy a normál üres hely helyett a ~ vagy \ karaktereket használjuk. (A karakter egyszerűen az üres karakter szimbóluma, amely egyébként nem látható.) Mindkét módszer beilleszti a normál szóközt; emellett a ~ egy rögzített mező, amely megakadályozza, hogy a sor ezen a ponton megszakadjon. A fenti példákat így kell beírni: i.~e., Dr.~Sleep és Tud.\Akad. Eredményül az i. e., Dr. Sleep és Tud. Akad. rövidítéseket kapjuk a megfelelő térközzel, és arra kényszerítve, hogy az első kettő mind egy sorban legyen. A harmadik esetben nem gond, ha a Tud. és Akad. különböző sorokban jelenik meg.

A nagybetűt követő pontot nem a mondat végeként értelmezzük, hanem rövidítésként. Ha valóban egy mondat vége, akkor az extra szóköz elérése érdekében a pont elé \@-t kell hozzáadni. Például ha a mondat a NASA-val végződik, akkor a megoldás: NASA\@.

Francia térköz

A mondatok közötti további szóközt a \frenchspacing parancsal lehet kikapcsolni, amely mindaddig érvényben marad, amíg a \nonfrenchspacing parancs ki nem kapcsolja. Ebben az esetben a \@ parancs figyelmen kívül marad, és elhagyható.

Tetszőleges térköz beszúrása

A következő parancsokkal tetszőleges méretű szóközt lehet beilleszteni a szövegbe:

```
\hspace{space}
```

```
\hspace*{space}
```

ahol a *space* a távolság mértékére vonatkozó hossz-specifikáció, például 1,5cm vagy 3em. (Az *em* az M betű szélessége az aktuális betűtípusban.)

Ez a parancs a szöveg azon pontjára helyezi az adott szélességű szóközt, ahol megjelenik. A szabványos alaknak (* nélkül) nincs hatása a dokumentumra, ha a kimeneti sor elején van, mint ahogy a normál üres helyeket is eltávolítják a sorok elejéről. A *-os alak viszont beszúrja a szóközt, függetlenül attól, hogy hol fordul elő.


```
Egy soron belüli szöveg igazítás:\\
Balra\hfill Jobbra\\
Balra\hfill Középre\hfill Jobbra\par
Folytonos pontok és vonalak sorozatának beszúrása:\\
Balra\dotfill Jobbra\\
Balra\hrulefill Középre\hrulefill Jobbra\\
Indulás \dotfill\dotfill\dotfill\ 8:30 \hfill Érkezés \hrulefill\ 11:45\\

\end{document}
```

2.6.2 Sortörés

A szöveg sorokra bontása automatikusan történik TEX és LATEX nyelven. Vannak azonban olyan esetek, amikor a sortörést kényszeríteni vagy ösztönözni kell, vagy amikor a sortörést el kell nyomni.

A `\par` parancs

Új sort további sorközzel vagy anélkül a `\par` paranccsal lehet létrehozni. A szintaxisa:

`\[space]`

`*[space]`

A *space* opcionális argumentum egy olyan hosszúság, amely megadja, hogy mekkora további sortávolságot kell helyezni a sorok közé. Ha új oldalt kell kezdeni, akkor a további sorköz nem kerül bele, és az új oldal a következő szövegsorral kezdődik. A *-os alak megakadályozza, hogy új oldal jöjjön létre a két sor között.

A `*[5cm]` esetén az aktuális sor véget ér, és a következő sor elé 5 cm-es függőleges térköz kerül beillesztésre, amely arra kényszerül, hogy az aktuális sorral azonos oldalon legyen. Ha oldaltörésre van szükség, akkor az az aktuális sor elé kerül, majd az új oldal tetejére kerül az 5 cm-es függőleges térközzel és a következő szövegsorral együtt.

A `\newline` parancs az opcióköz nélkül megegyezik a `\par` paranccsal. Ez azt jelenti, hogy új sor kezdődik további szóköz nélkül, és ezen a ponton oldaltörés lehetséges.

Mindkét parancs csak egy bekezdésen belül adható, és nem közöttük, ahol értelmetlen lenne.

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

Ez az első sor.

Ez a második sor, amely 1 cm-rel van elválasztva az előző sortól.
A harmadik és a második sor előtt nincs szóköz.

Megoldás:

```
\documentclass{article}

\begin{document}
\begin{center}
\LaTeX \\
\end{center}

\begin{flushleft}
Ez az első sor. \\[1cm]
Ez a második sor, amely 1 cm-rel van elválasztva az előző sortól.
\newline A harmadik
és a második sor előtt nincs szóköz.
\end{flushleft}

\end{document}
```

További sorbontó parancsok

A `\linebreak` parancs arra szolgál, hogy ösztönözze vagy kényszerítse a sortörést a szöveg egy bizonyos pontján. Szintaxisa:

```
\linebreak[num]
```

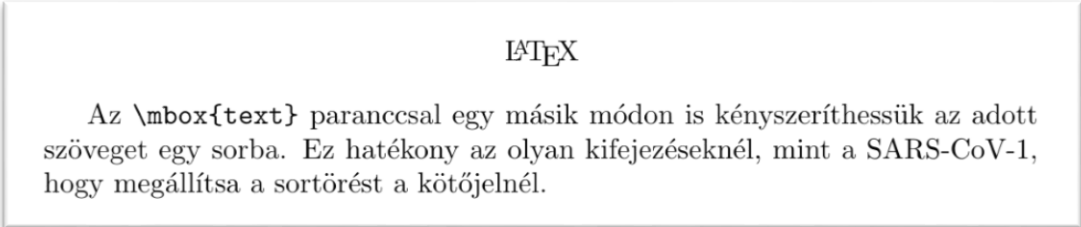
ahol a *num* egy opcionális argumentum, egy 0 és 4 közötti szám, amely meghatározza a sortörés fontosságát. A parancs sortörést javasol, és minél nagyobb a szám, annál erősebb az ajánlás. A 0 érték megengedi a törést ott, ahol egyébként nem fordulna elő (például egy szó közepén), míg a 4 sortörést kényszerít, csakúgy, mint a `\linebreak` szám nélkül. A különbség ezen parancs és a `\\` vagy a `\newline` között az, hogy az aktuális sor teljesen sorkizárt lesz, azaz szóközök kerülnek hozzáadásra, így a szöveg teljesen kitölti a sort. A `\\` és `\newline` esetén azonban a sor üres szóközzel lesz kitöltve az utolsó szó után.

Az előző parancs ellenkezője a `\nolinebreak[num]` parancs, amely letiltja a sortörést az adott helyen, a *num* pedig az ajánlás mértékét adja meg. A `\nolinebreak` *num* argumentum nélkül ugyanazt a hatást fejt ki, mint a `\nolinebreak[4]`, vagyis a sortörés itt teljesen lehetetlen.

Az `\mbox{text}` paranccsal egy másik módon is kényszeríthessük az adott szöveget egy sorba. Ez hatékony az olyan kifejezéseknél, mint a „SARS-CoV-1”, hogy megállítsa a sortörést a kötőjelnél.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass{article}

\begin{document}
\begin{center}
\LaTeX \\
\end{center}

Az \verb|\mbox{text}| paranccsal egy másik módon is kényszeríthessük az adott
szöveget egy sorba. Ez hatékony az olyan kifejezéseknél, mint a \mbox{SARS-CoV-1},
hogy megállítsa a sortörést a kötőjelnél.

\end{document}
```

2.6.3 Függőleges térköz

A következő parancsok segítségével további függőleges térközt lehet hozzáadni az egyes bekezdések közötti távolsághoz

`\vspace{space}`

`\vspace*{space}`

A *-os alak akkor is hozzáadja a szóközt, ha új oldal jelenik meg, vagy ha a parancs egy új oldal tetején jelenik meg. A szabványos alakja ezekben a helyzetekben figyelmen kívül hagyja az extra függőleges térközt.

Ha ezeket a parancsokat egy bekezdésen belül adjuk meg, akkor plusz szóköz az aktuális sor után kerül beszúrásra, amely a szokásos módon jobbra és balra igazodik.

A szóköz paraméter akár negatív is lehet, hogy a következő szöveget feljebb helyezze az oldalon, mint ahol általában nyomtatva lenne. A `\vfill` parancs a `\vspace{\fill}` rövidítése. Ez a függőleges térköz `\hfill` megfelelője, elegendő üres függőleges szóköz beszúrása ahhoz, hogy a

szöveg felső és alsó része pontosan illeszkedjen a felső és alsó margóhoz. A `\hfill` többszöri előfordulására vonatkozó megjegyzések a `\vfill`-re is vonatkoznak. Ha ezt a parancsot egy oldal elején adjuk meg, akkor a rendszer figyelmen kívül hagyja, akárcsak a `\vspace{\fill}` szabványos formája. Ha változó szóközt kell elhelyezni az oldal tetején, akkor a *-os alakot, `\vspace*{\fill}` kell használni.

További parancsok a bekezdések közötti térköz növelésére:

`\bigskip`

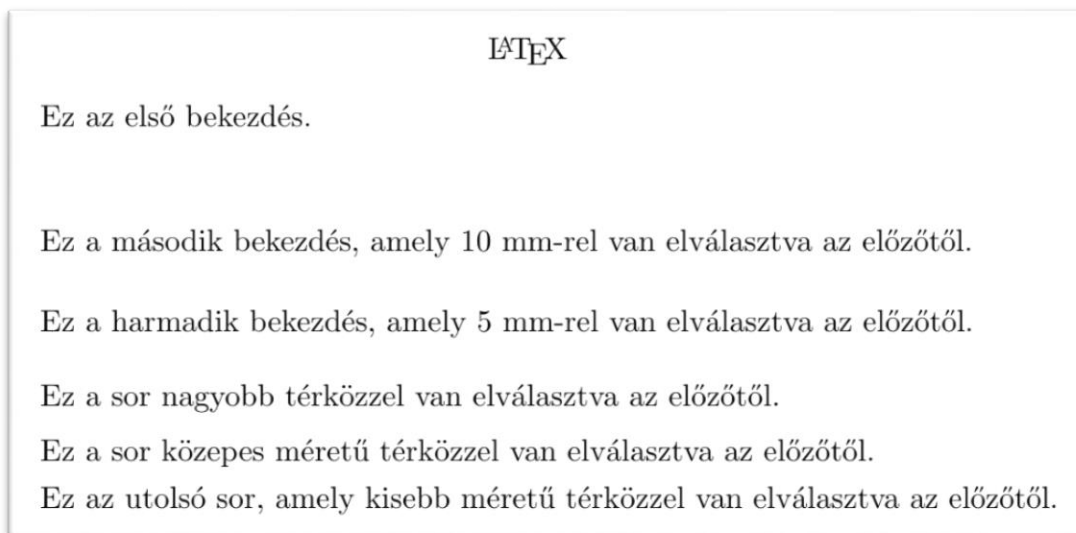
`\medskip`

`\smallskip`

amelyek a dokumentumosztályban deklarált betűmérettől függően függőleges térközt adnak hozzá.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass{article}

\begin{document}
\begin{center}
\LaTeX \\
\end{center}

Ez az első bekezdés.\par
\vspace{10mm}
Ez a második bekezdés, amely 10 mm-rel van elválasztva az előzőtől.\par
\vspace{5mm}
Ez a harmadik bekezdés, amely 5 mm-rel van elválasztva az előzőtől.\par
```

```
\bigskip
Ez a sor nagyobb térközzel van elválasztva az előzőtől.\par
\medskip
Ez a sor közepes méretű térközzel van elválasztva az előzőtől.\par
\smallskip
Ez az utolsó sor, amely kisebb méretű térközzel van elválasztva az előzőtől.

\end{document}
```

2.6.4 Oldaltörés

A szöveg oldalakra bontása automatikusan megtörténik a TEX és a LATEX nyelvekben, ugyanúgy, mint a sortörésnél. Itt is szükség lehet arra, hogy beleavatkozzunk a program azon elképzelésébe, hogy hol kell szünetet tartani.

Normál oldalak

A következő parancsok

```
\pagebreak[num]
```

```
\nolinebreak[num]
```

a `\linebreak` és a `\nolinebreak` sortöréshez kapcsolódó parancsok megfelelői az oldaltöréshez. Ha két bekezdés között megjelenik a `\pagebreak`, akkor a rendszer ezen a ponton új oldalt kényszerít. Ha egy bekezdésen belül van, az új oldal az aktuális sor befejezése után kerül megvalósításra. Ez a sor a szokásos módon jobbra és balra igazított lesz.

A `\nolinebreak` parancs az ellenkező hatást fejt ki. A bekezdések között megakadályozza az oldaltörést, egy bekezdésen belül pedig az aktuális sor végén előforduló oldaltörést.

A 0 és 4 közötti *num* értékét meghatározó opcionális számok az oldaltörésre való bátorítás vagy elkedvetlenítés mértékét fejezik ki. A `\linebreak` parancshoz hasonlóan tovább megy, ahogy a törés előtti sor balra és jobbra sorkizárt extra szóközzel, ugyanúgy a törés előtti oldal sorközi sorközökkel bővül, hogy felül és alul sorkizárt legyen.

Az oldal közepén történő befejezéséhez, üres szóközzel való kitöltéséhez és egy új oldalra lépéshez megfelelő parancs a következő:

```
\newpage
```

ami a `\newline`-nek felel meg az oldaltörés tekintetében.

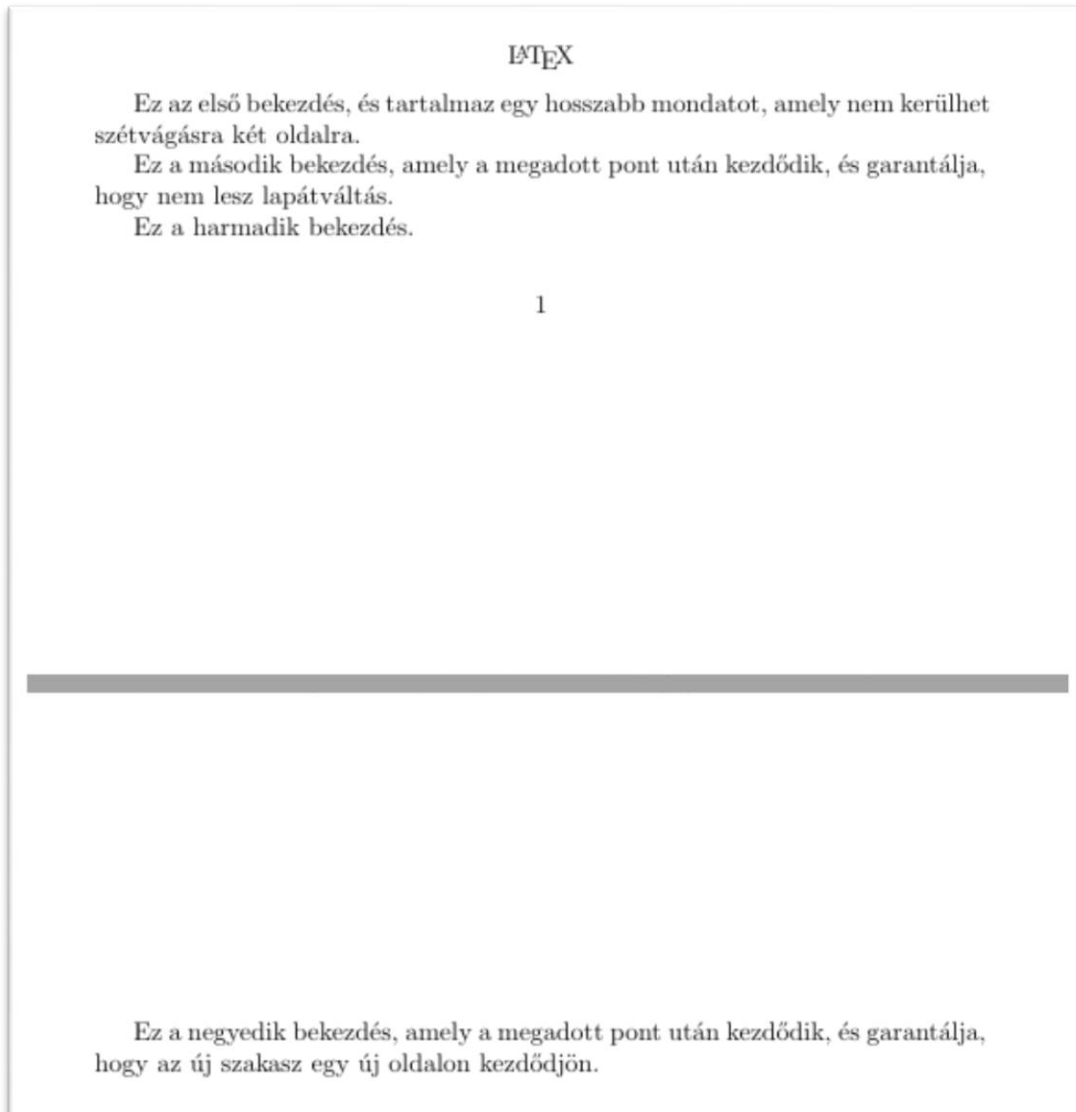
Oldalak ábrákkal és táblázatokkal

Ha a szöveg táblázatokat, képeket vagy ábráknak fenntartott helyet tartalmaz, akkor ezek a megfelelő parancs helyére kerülnek beszúrásra, feltéve, hogy az aktuális oldalon van számukra elegendő hely. Ha nincs elég hely, a szöveg folytatódik, és az ábra vagy táblázat eltárolódik, hogy egy következő oldalra kerüljön.

A `\clearpage` parancs bezárja az aktuális oldalt, mint a `\newpage`, ezen felül az összes függőben lévő ábrát és táblázatot egy vagy több extra oldalon jeleníti meg.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass{article}

\begin{document}
\vspace*{16cm}
\begin{center}
\LaTeX \\
\end{center}

Ez az első bekezdés, és tartalmaz egy hosszabb mondatot, amely nem kerülhet szétvágásra két oldalra.
```

```
\nopagebreak
```

Ez a második bekezdés, amely a megadott pont után kezdődik, és garantálja, hogy nem lesz lapátváltás.

Ez a harmadik bekezdés.

```
\pagebreak
```

Ez a negyedik bekezdés, amely a megadott pont után kezdődik, és garantálja, hogy az új szakasz egy új oldalon kezdődjön.

```
\end{document}
```

Kéthasábos oldalak

Ha a dokumentumosztályban kétoszlop opciót választotta, vagy a `\twocolumn` parancs érvényben van, akkor a `\pagebreak` és `\newpage` két parancs befejezi az aktuális oszlopot, és kezd egy újat, oldalként kezelve az oszlopokat. Másrészt a `\clearpage` és `\cleardoublepage` parancsok lezárják az aktuális oldalt, és szükség esetén beszúrnak egy üres jobb oldali oszlopot.

Feladat:

Hozd létre a következő szövegrészt!

Volt, hol nem volt, még hetedhét ország is túl, volt egy szegény asszony. Ennek a szegény asszonynak volt három fia: a legnagyobbik csizmadia-, a második szabómesterséget folytatott, hanem a legkisebbikkel semmire sem lehetett menni: mindig a hamuban. . .

Megoldás:

```
\documentclass{article}
```

```
\begin{document}
```

```
\twocolumn
```

Volt, hol nem volt, még hetedhét ország is túl, volt egy szegény asszony. Ennek a szegény asszonynak volt három fia: `\pagebreak` a legnagyobbik csizmadia-, a második szabómesterséget folytatott, hanem a legkisebbikkel semmire sem lehetett menni: mindig a hamuban`\dots`

```
\end{document}
```

Kétoldalas oldalak

Amennyiben a kétoldali dokumentumosztály opció ki van választva, egy további oldaltörő parancs is elérhető:

`\cleardoublepage`

amely pontosan ugyanúgy működik, mint a `\clearpage`, de emellett a következő szöveg egy páratlan számú oldalra kerül. Ha szükséges, ennek érdekében egy üres oldalt nyomtatunk páros számmal.

Az oldaltörések meghatározása

A LATEX a következő parancsokkal lehetőséget biztosít az aktuális oldal magasságának kismértékű növelésére:

`\nlargethispage{size}`

`\nlargethispage*{size}`

amelyek csak ennek az egy oldalnak adják hozzá a hosszmeretet a `\textheight`-hez. Néha csak néhány pont különbségre van szükség a rossz oldaltörés elkerüléséhez. A parancs *-os alakja szükség szerint csökkenti a sorközi távolságot is, hogy maximalizálja az oldalon található szöveg mennyiségét.

2.7 A szavak elválasztása

Amikor egy sort jobbra és balra kell igazítani, gyakran kiderül, hogy nem lehet törést tenni az egész szavak között anélkül, hogy a szöveget ne toljuk túl közel egymáshoz, vagy ne szúrjunk be hatalmas hézagokat a szavak közé. Ezután el kell választani egy szót. Ezt az alapvető feladatot a TEX, a LATEX alapja, egy szóelválasztó algoritmus segítségével hajtja végre. Ez (majdnem) tökéletesen működik angol szöveghez. Ennek ellenére időnként még ez is követ el hibákat, amelyeket emberi beavatkozással ki kell javítani.

Ha normál TEX/LATEX nyelvet használnak más nyelvekhez, vagy ha idegen szavak szerepelnek az angol szövegben, nagy valószínűséggel helytelen elválasztások jelennek meg.

2.7.1 Kézi elválasztás

A hibásan tagolt szó kijavításának legegyszerűbb módja, ha be írunk egy `\-` parancsot a szóba, a megfelelő helyre. A *kézirat* szó például egyáltalán nem lesz kötőjellel jelölve, ezért ha gondot okoz a sortöréssel, akkor *kéz\-\i\-\rat*ként írja be. Ez arra utasítja a TEX-et, hogy szükség szerint ossza fel a szót és hagyja figyelmen kívül a szokásos szabályait.

A `\-` parancs csak az elválasztást teszi lehetővé a jelzett helyeken, de nem kényszeríti rá. Ha a szerző határozottan ragaszkodik ahhoz, hogy egy szót egy adott ponton el kell választani, mondjuk az *i* és az *r* között a *kézirat*ban, akkor begépelheti a *kézi\-\linebreak rat* parancsot, hogy ezt elérje. Ez a *brute force* módszer azonban nem ajánlott, mert a sortörés itt mindig előfordul, még akkor is, ha később módosítjuk a szöveget.

A TEX lehetővé teszi az általános elválasztási szabálymeghatározását a

`\discretionary{before}{after}{without}`

paranccsal, ahol előtte és utána található a törés mindkét oldalán lévő betűk (kötőjellel), ha felosztásra kerül sor, a nélkül pedig a normál szöveg kötőjel nélkül. Így Boris Becker nevét így kell beírni

Boris Be`\discretionary{k}{k}{ck}`er

Egyébként a `\-` parancs a `\discretionary{-}{ }{ }` rövidítése.

2.7.2 Elválasztási lista

A következő paranccsal a helytelenül kötőjellel ellátott és a dokumentumban gyakran előforduló szavakat a preambulumban a kivételek listájába lehet tenni, hogy elkerüljük a `\-` minden alkalommal történő fáradságos beszúrását:

`\hyphenation {lista}`

A *lista* üres vagy új sorokkal elválasztott szavak halmazából áll, a megengedett osztási pontokat kötőjellel jelölve. Például

`\hyphenation{kéz-i-rat lap-top Bu-da-pest ... }`

A lista csak normál ('a'-'z') betűket tartalmazó szavakat tartalmazhat, speciális karakterek vagy ékezetek nélkül. Ha azonban alkalmazzuk az *inputenc* csomagot, akkor a közvetlenül begépelt betűk is bekerülnek az automatikus elválasztásba.

2.7.3 Elválasztás elnyomása

A szavak rossz elválasztásának elkerülésének másik módja az elválasztás kikapcsolása, legalább egy-két bekezdésre. Valójában a

`\begin{sloppypar} bekezdés szövege \end{sloppypar}`

parancs nem akadályozza meg a szóelválasztást, de engedélyezi a nagyobb szóközök használatát, figyelmeztető üzenet nélkül. Ez azt jelenti, hogy gyakorlatilag minden sor megszakad a szavak között. Lehetőség van arra is, hogy a `\sloppy` parancsot a preambulumba vagy az aktuális környezetbe helyezzük, hogy csökkentsük a teljes dokumentumban vagy a környezet hatókörén belül a szóelválasztások számát. Ez akkor javasolt, ha a sor szélessége meglehetősen szűk.

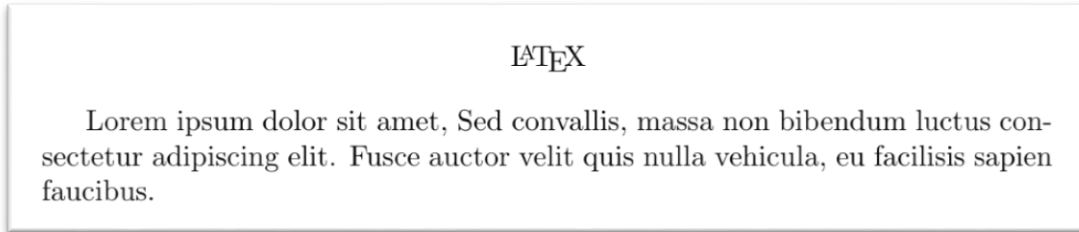
Ha a `\sloppy` parancs érvényben van, akkor ideiglenesen visszavonható, és a `\fussy` paranccsal vissza lehet kapcsolni az elválasztást.

2.7.4 Szóelválasztás többnyelvű szövegben

Több elválasztási lista is szerepelhet a TEX formátumban, lehetővé téve az elválasztási sémák váltását a `\language TEX` paranccsal az adott dokumentumon belül.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass{article}

\hyphenation{con-sec-te-tur a-dipis-cing}

\begin{document}
\begin{center}
\LaTeX \\
\end{center}

Lorem ipsum dolor sit amet, Sed convallis, massa non bibendum luctus consectetur
adipiscing elit. Fusce auctor velit quis nulla vehicula, eu facilisis sapien faucibus.

\end{document}
```

3 Dokumentum elrendezése és felépítése

3.1 A dokumentum osztály

A LATEX fájl preambulának első parancsa határozza meg a teljes dokumentum globális feldolgozási formátumát. A szintaxisa a következő:

```
\documentclass[options]{class}
```

ahol az *class* valamilyen értékét kötelezően meg kell adni, míg az *options* elhagyható, ha az alapértelmezett értékek számunkra elfogadhatók.

Az osztály standard értékei a következők lehetnek, amelyekből csak egy adható meg: *book*, *report*, *article*, vagy *letter*. Az alapvető különbségek ezen osztályok között nem csak az oldalelrendezésben, hanem a szervezésben is rejlenek. Egy cikk (*article*) tartalmazhat részeket, szakaszokat, alfejezeteket és így tovább, míg a jelentésnek (*report*) is lehetnek fejezetei. Egy könyvnek (*book*) is vannak fejezetei, de másképp kezeli a páros és páratlan oldalakat; emellett minden oldalra élőfejezet nyomtat a fejezetek és szakaszok címével együtt.

A szabványos osztályokon kívül más osztályok is léteznek, konkrét folyóiratokhoz vagy könyvprojektekhez való hozzájárulásként. Ezek saját opciókkal és további parancsokkal

rendelkeznek, amelyeket külön dokumentációban vagy utasításokban kell leírni. Mivel azonban ezek általában a szabványos osztályok valamelyikének módosításai, a következő opciók többsége rájuk is vonatkozik.

3.1.1 Standard osztályok opciói

A rendelkezésre álló opciók lehetővé teszik a formázás különféle módosításait. Az alábbiak szerint csoportosíthatók.

Betűméret kiválasztása

Az alap betűméretet az egyik opcióval lehet megadni, mint például

10p 11pt 12pt

Ez annak a betűtípusnak a mérete, amelyben a dokumentum normál szövege be lesz állítva. Ha nincs megadva méret opció, az alapértelmezett érték 10pt. Az összes többi betűméret-deklaráció ehhez a szabványos mérethez kapcsolódik, így a szakaszcímek, lábjegyzetek stb. mérete automatikusan megváltozik, ha más alapvető betűméretet választunk.

A papírméret megadása

A LATEX a kiválasztott betűméretnek és dokumentumosztálynak megfelelően kiszámítja a szövegsor szélességét és oldalankénti sorait. A margókat is úgy állítja be, hogy a szöveg vízszintesen és függőlegesen is középre kerüljön. A papírformátumot a következő lehetőségek egyike határozza meg:

letterpaper (11 × 8,5 hüvelyk)	a4paper (29,7 × 21 cm)
legalpaper (14 × 8,5 hüvelyk)	a5paper (21 × 14,8 cm)
executivepaper (10,5 × 7,25 hüvelyk)	b5paper (25 × 17,6 cm)

Az alapértelmezett papír méret a *letterpaper*, amelynek mérete 11 × 8,5 hüvelyk.

Általában a papírformátum az úgynevezett portré (*portrait*) módban van megadva, vagyis állított elhelyezkedésű. A tájkép (*landscape*) mód megadásával fekvő módra állíthatjuk.

Oldalformátumok

A következő opciókkal az oldalon található szöveg egy vagy két oszlopba formázható:

onecolumn twocolumn

Az alapértelmezett az *onecolumn*. A *twocolumn* opció esetén az oszlopok közötti távolságot, a `\columnsep` és `\columnseprule` paraméterekkel lehet megadni.

A páros és páratlan oldalak a következő opcióktól függően eltérően nyomtathatók:

oneside twoside

Az *oneside* esetében minden oldalt ugyanúgy nyomtat; *twoside* esetében azonban az élőfejek olyanok, hogy az oldalszám a páratlan oldalakon jobb oldalon jelenik meg, míg a páros oldalakon bal oldalon. Nyomtatáskor az oldalszámok mindig a külső oldalon vannak, ahol

könnyebben észrevehető. Ez a könyvosztály (*book*) alapértelmezett értéke. Az *article* és a *report* esetében az alapértelmezett az *oneside*.

A könyvosztályban a fejezetek általában egy jobb oldali, páratlan számú oldalon kezdődnek. Az *openright* és *openany* lehetőségekkel megadhatjuk, hogy a fejezetek mely oldalon jelenjenek meg. Az *openany* esetén a fejezet mindig a következő oldalon kezdődik, de az alapértelmezett *openright* esetén egy üres oldal is beszúrható.

Általában egy *book* vagy *report* címe külön oldalra kerül, míg egy *article* esetében ugyanazon az oldalon, mint a kezdődő szöveg. A *notitlepage*, *titlepage* lehetőségekkel a szokásos viselkedés felülbíráható.

További lehetőségek

A többi standard opciók a következők:

- | | |
|----------------|---|
| <i>leqno</i> | Az egyenletszámok a megjelenített képletekben a bal oldalon jelennek meg a normál jobb oldal helyett. |
| <i>fleqn</i> | A megjelenített képletek balra rendezve lesznek beállítva, nem pedig középre. A behúzás mértéke a <code>\mathindent</code> paraméterrel állítható be. |
| <i>openbib</i> | A bibliográfiák formátuma módosítható, így a szegmensek új sorokra kerülnek. Alapértelmezés szerint az egyes bejegyzések szövegei együtt futnak. |
| <i>draft</i> | Ha a LATEX sortörő mechanizmus (szóelválasztás) nem működik megfelelően, és a szövegnek ki kell lógnia a jobb margóba, akkor ezt vastag fekete sáv jelöli, hogy észrevehető legyen. |
| <i>final</i> | A piszkozat (<i>draft</i>) és az alapértelmezett ellentéte. A túl széles szövegsorok semmilyen módon nincsenek megjelölve. |

Ha több opciót kell megadni, akkor azokat vesszővel kell elválasztani, mint például: `\documentclass[11pt,twoside,fleqn]{article}`. Az opciók sorrendje lényegtelen. Ha két egymásnak ellentmondó lehetőséget adunk meg, mint pl. az *oneside* és a *twoside*, akkor nem egyértelmű, hogy melyik lesz a hatékony. Ez teljes mértékben az osztályfájlban lévő definícióktól függ, ezért a legjobb az ilyen helyzetek elkerülése.

Néhány opcióhoz tartozó paraméter

Egyes opciók olyan paramétereket használnak, amelyek bizonyos alapértelmezett értékeket kaptak:

- | | |
|--------------------------|--|
| <code>\mathindent</code> | megadja a behúzást a bal margótól az egyenletszámokhoz, ha a <i>fleqn</i> van kiválasztva; |
| <code>\columnsep</code> | megadja a két oszlop közötti helyet a két oszlop számára; |

`\columnseprule` meghatározza a két oszlop közötti függőleges vonal szélességét a *twocolumn* beállításnál. Az alapértelmezett szélesség nulla;

Ezen paraméterek szabványos értékei a `\setlength` LATEX paranccsal módosíthatók. Például a `\mathindent 3 cm` megváltoztatásához:

```
\setlength{\mathindent}{3cm}
```

Ezek a paraméterek értéket kaphatnak akár a preambulumban, akár a dokumentum bármely pontján. A preambulum paraméterei a teljes dokumentumra vonatkoznak, míg a szövegben lévők a következő változtatásig vagy annak a környezetnek a végéig érvényben vannak. Ez utóbbi esetben a korábbi értékek ismét érvényesek.

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis. Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris venenatis neque ut vestibulum facilisis. Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero. Proin mattis posuere neque, at sodales massa rhoncus eu.

Nulla id dolor vel nisl aliquet efficitur. Integer a feugiat sem. Sed venenatis, sem vel fermentum sodales, turpis justo eleifend arcu, lacinia imperdiet sem neque sit amet urna. Suspendisse dictum iaculis elementum. Nullam efficitur feugiat elit, ac facilisis nunc. In dolor justo, ullamcorper eget consequat at, rhoncus quis mi.

Megoldás:

```
\documentclass[12pt,a4paper]{report}

\begin{document}
\begin{center}
\LaTeX \\
\end{center}

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis. Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris venenatis neque ut vestibulum facilisis. Mauris felis massa,
```

```
scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero. Proin mattis posuere neque, at sodales massa rhoncus eu.
```

```
Nulla id dolor vel nisl aliquet efficitur. Integer a feugiat sem. Sed venenatis, sem vel fermentum sodales, turpis justo eleifend arcu, lacinia imperdiet sem neque sit amet urna. Suspendisse dictum iaculis elementum. Nullam efficitur feugiat elit, ac facilisis nunc. In dolor justo, ullamcorper eget consequat at, rhoncus quis mi.
```

```
\end{document}
```

3.1.2 Csomagok használata

A LATEX bővíthető olyan csomagokkal, amelyek vagy az alapterelés részét képezik, vagy a felhasználók által biztosítottak.

A csomag nem más, mint egy *.sty* kiterjesztésű fájlban tárolt LATEX (vagy TEX) parancsok halmaza, bár vannak speciális parancsok, amelyek csak bennük jelennek meg. Egy csomag meghívásához egyszerűen

```
\usepackage{package}
```

parancsot a preambulumban hívjuk, ahol a *package* a fájl gyökérneve. Egynél több csomag is betölthető a `\usepackage` egyetlen hívásával. Például két szabványos LATEX-szel ellátott csomag a *makeidx.sty* és az *ifthen.sty* fájlokban van tárolva. Együtt következő módon olvashatóak be:

```
\usepackage{makeidx,ifthen}
```

Egy csomaghoz opciók is rendelhetők, amelyeket ugyanúgy lehet kiválasztani, mint a dokumentumosztályoknál: az opciónevek szögletes kapcsos zárójelek közé helyezésével. Az általános szintaxis a következő:

```
\usepackage[opt1,opt2...]{package1,package2,...}
```

ahol az összes felsorolt opció az összes kiválasztott csomagra érvényes lesz. Ha valamelyik csomag nem érti az egyik opciót, figyelmeztető üzenet jelenik meg a monitoron.

3.1.3 Globális és helyi lehetőségek

A `\documentclass` paranccsal megadott opciók egyik érdekes tulajdonsága, hogy ezek a következő csomagokra is vonatkoznak. Ez azt jelenti, hogy ha több csomag ugyanazt az opciót használja, akkor azt csak egyszer kell deklarálni a `\documentclass`-ban. Például tervezhetünk egy csomagot az *article* módosítására egy helyi stílus generálására, amely különböző dolgokat hajthat végre egy- vagy kétoszlopos szöveg esetén; ez a csomag az *onecolumn* és a *twocolumn* osztálybeállításokat használhatja ennek eléréséhez. Alternatív megoldásként több csomag tartalmazhat nyelvfüggő funkciókat, amelyeket olyan opciókkal lehet aktiválni, mint a francia

vagy a német; elég csak az ilyen lehetőségeket felsorolni a `\documentclass`-ban, hogy alkalmazza őket minden csomagra. Az ilyen opciókat globálisnak nevezzük, mivel a rendszer automatikusan továbbítja őket az összes következő csomagnak.

A globális opcióknak nem kell korlátozódnuk szabványos osztályopciókra. Figyelmeztető üzenet csak akkor kerül kinyomtatásra, ha sem az osztály, sem a csomagok közül egyet vagy többet nem ért meg. Ezzel szemben a `\usepackage` paraméterrel megadott beállítások csak az adott parancsban felsorolt csomagokra vonatkoznak. Figyelmeztetés jelenik meg, ha a csomagok közül egy vagy több nem ismeri fel a helyi opciók egyikét sem.

3.1.4 Osztály- és csomagverziók

Az osztály- és csomagfájloknak általában belső verzió-specifikációjuk van a kiadási dátum formájában (`éééé/hh/nn`). Ha olyan szolgáltatást szeretnénk használni, amelyről tudjuk, hogy egy adott napon adták hozzá, akkor ezt a dátumot szögletes zárójelben adjuk meg az osztály vagy a csomag neve után. A verzió dátuma is hozzáadható a `\documentclass` parancshoz, hogy biztosítsa az osztályfájl megfelelő verziójának alkalmazását.

3.2 Oldalak stílusa

Az alapvető oldalformátumot az oldalstílus határozza meg. Egy kivétellel ez a parancs általában a preambulumban szerepel. A formája a következő:

```
\pagestyle{style}
```

A kötelező argumentumstílus a következő értékek egyike lehet:

- plain* A fejléc üres, a láblécben középre helyezett oldalszám található. Ez az alapértelmezett az *article* és *report* esetén, ha a preambulumban nincs megadva a `\pagestyle`.
- empty* Ebben az esetben a fejléc és a lábléc üres, valamint nincs oldalszám nyomtatva.
- headings* A fejléc tartalmazza az oldalszámot, valamint a címadatokat (fejezet- és szakaszcímeket); a lábléc üres. Ez az alapértelmezett *book* esetében.
- myheadings* Ugyanaz, mint a fejléceknél, kivéve, hogy a fejlécben lévő oldalcímek kiválasztása nem automatikusan történik, hanem kifejezetten a `\markright` vagy a `\markboth` parancsok adják őket.

A `\thispagestyle{style}` parancs pontosan úgy működik, mint a `\pagestyle`, kivéve, hogy csak az aktuális oldalt érinti. Például a `\thispagestyle{empty}` paranccsal le lehet tiltani az oldalszámozást az aktuális oldalon. Csak az oldalszám kinyomtatása kerül letiltásra; a következő oldal úgy lesz számozva, mintha soha nem adták volna ki a parancsot.

3.2.1. Címnyilatkozatok

Az oldalstílusok címsorainál és a saját címsoroknál a fejlécben megjelenő információ megadható a következő deklarációkkal:

```
\markright{right head}
```

```
\markboth{left head}{right head}
```

A `\markboth` deklarációt a kétoldali dokumentumosztály opcióval használjuk, a páros oldalakat bal oldalon, a páratlan oldalakat pedig a jobb oldalon. Ezenkívül az oldalszámot a fejléc bal oldalára nyomtatják bal oldal esetén, és a jobb oldalra jobb oldal esetén.

Egyoldalas nyomtatás esetén minden oldal jobbkezesnek számít. Ebben az esetben a `\markright` deklaráció a megfelelő. Kétoldalas kimenettel is használható a `\markboth`-ban megadott jobb oldali fejléc felülírására.

Az oldalstílus-fejléceknél az oldal fejlécében a szabványos címek a dokumentumtól és az oldalstílustól függően a fejezet, a szakasz vagy az alszakasz fejlécei a következő séma szerint:

Stílus		Bal oldal	Jobb oldal
book, report	egyoldalas	—	Fejezet
	kétoldalas	Fejezet	Szakasz
article	egyoldalas	—	Szakasz
	kétoldalas	Szakasz	Alszakasz

Ha egy oldalon több `\section` vagy `\subsection` van, akkor az utolsó fejléce jelenik meg az oldal fejlécében.

3.2.2 Egyedi fej- és lábléc

A szabványos oldalstílusok határozzák meg, hogyan jelenjenek meg a fejléc és a láblécek, és milyen információkat tartalmaznak. A `fancyhdr` csomag lényegesen nagyobb rugalmasságot kínál a felhasználónak. Ez a csomag egy további, `fancy` nevű oldalstílust tesz elérhetővé, amelyet a felhasználó könnyen újradefiniálhat. A fej- és lábléc három részből tevődik össze: bal, közép, jobb, amelyek mindegyike külön-külön definiálható

```
\lhead{Left head}    \chead{Center head} \rhead{Right head}
```

```
\lfoot{Left foot}    \cfoot{Center foot} \rfoot{Right foot}
```

ahol szöveget adhatunk meg, vagy egy parancsot, mint pl. a `\thepage` az aktuális oldalszám kinyomtatására. Mind a fejléc, mind a lábléc díszíthető, amelynek szélességét a `\headrulewidth` és `\footrulewidth` parancsok határozzák meg. Alapértelmezés szerint a díszes fejléc és lábvonal nagyjából ugyanaz, mint a címsoroldal stílusa, de a fejléc 0,4 pontra, a lábléc pedig 0-ra van állítva. Ezek újradefiniálhatók is, mint például a következő példában:

```
\renewcommand{\footrulewidth}{0.6pt}
```


A fenti definiáló parancsok valójában konkrét példák az általánosabb `\fancyhead` és `\fancyfoot` parancsokra, ahol

`\lhead{..}` megfelelője a `\fancyhead[L]{..}`

`\cfoot{..}` megfelelője a `\fancyfoot[C]{..}`

és így tovább, az L C R jelentése: „bal”, „közép”, „jobb”.

A kétoldalas opcióval rendelkező kétoldalas kimenet esetén általában a bal és a jobb oldali részeket az oldalszámmal kell felváltani. Ennek legegyszerűbb módja a

`\fancyhead[LE,RO]{szöveg1}` `\fancyhead[LO,RE]{szöveg2}`

hogy ugyanazt a *szöveg1*-et a páros oldalak bal oldalába, a páratlan oldalak jobb oldalába, a *szöveg2*-öt pedig fordítva helyezze el. A `\fancyhead{}` használatával a fejléc minden része üresre van állítva, amit meg kell tenni, mielőtt kifejezetten visszaállítaná őket. Hasonlóképpen a `\fancyfoot{}` minden láb bejegyzést üresre állít.

A díszes oldalstílus alapértelmezett (kétoldalas) definíciói a következők:

`\fancyhead[EL,OR]{\textsl{\rightmark}}`

`\fancyhead[ER,OL]{\textsl{\leftmark}}`

ahol a `\rightmark` és a `\leftmark` a `\chapter`, `\section`, `\subsection` parancsok által generált címodalstílusok automatikus szövegeit tartalmazza, míg a `\textsl` ferde betűtípussal állítja be argumentumát. Ezeket felhasználhatjuk a fejléc automatikus szövegekkel való újradefiniálására is.

Továbbá létezik a legáltalánosabb `\fancyhf` parancs, amely opcionális [H] és [F] argumentumokat tartalmaz a fej- vagy láblécekre. Így a `\fancyhf[HL]{..}` ugyanaz, mint a `\fancyhead[L]{..}`. Természetesen a `\fancyhf{}` mindent visszaállít.

Több osztályban a fejezet első oldala vagy a dokumentum legelső oldala automatikusan sima lapra vált. Ha a felhasználó ezt meg akarja változtatni, akkor újra kell definiálnia az oldalstílust.

A *fancyhdr* csomag leegyszerűsíti ezt a feladatot a

`\fancypagestyle{plain}{definitions}`

paranccsal, ahol a definíciókat a `\fancyhead`, `\fancyfoot` parancsok alkotják, amelyek a felülvizsgált egyszerű stílusra vonatkoznak. Valójában bármely meglévő oldalstílus újradefiniálható ilyen módon.

3.2.3 Oldalszámzás

Az oldalszámzás stílusát meghatározó deklaráció alakja a következő:

`\pagenumbering{num style}`

A számstílus (*num style*) megengedett értékei:

arabic a normál arab (*arabic*) számokhoz,

roman a kisbetűs római számokhoz,
Roman a nagybetűs római számokhoz,
alph a kisbetűkhöz,
Alph a nagybetűknek.

Az alapértelmezett érték az *arabic*. Ez a deklaráció visszaállítja az oldalszámlálót 1-re. Ahhoz, hogy egy dokumentum előszavát római számokkal, a többi pedig az 1. oldallal kezdődő arab számokkal lapozzuk, a `\pagenumbering{roman}` szót kell deklarálnia az 1. fejezet elején. Előszót, majd az első `\chapter` parancs után azonnal visszaállítjuk az oldalszámozást a `\pagenumbering{arabic}` paranccsal.

A parancs kiadásával az oldalak számozhatók 1-től eltérő értékkel is a következő paranccsal:

`\setcounter{page}{page num}`

ahol az oldalszám (*page num*) az aktuális oldalon megjelenő szám.

3.2.4 Bekezdés formázása

A bekezdés megjelenését a következő paraméterek befolyásolják, valamint új értékeket kaphatnak a `\setlength` paranccsal:

`\parskip` A bekezdések közötti távolság *ex* egységekben kifejezve, így automatikusan változik a karakter betűméretével. Ennek változó hosszúságúnak kell lennie.

`\parindent` A bekezdés első sorának behúzásának mértéke.

`\baselinestretch` Ez egy olyan szám, amely felnagyítja az alapvonalak közötti normál távolságot, azt a képzeletbeli vonalat, amelyen a betűk ülnek. Ez a szám kezdetben szabványos sorköz esetén 1. Megváltoztatható a `\renewcommand{\baselinestretch}{factor}` paranccsal, ahol a *factor* decimális szám, például 1,5 és ez 50%-os növekedést eredményez. Ez ezután minden betűméretre vonatkozik. Ha ezt a parancsot a preambulumon kívül adjuk meg, akkor csak akkor lép életbe, ha másik betűméretet választunk.

Ezek a paraméterek a preambulumban vagy a dokumentum szövegében bárhol beállíthatók. Utóbbi esetben a változtatások a következő változtatásig vagy annak a környezetnek a végéig maradnak érvényben.

A következő parancsok elnyomják a behúzást a bekezdés elején, vagy oda kényszerítik, ahol egyébként nem fordulna elő: `\noindent` vagy `\indent`.

Alapértelmezés szerint a LATEX az első sor behúzásával jelzi a bekezdéseket. Egy másik lehetőség behúzás nélküli, de extra szóközzel a bekezdések között. Újradefiniálhatjuk a `\parindent` és `\parskip` parancsokat ennek megfelelően, vagy használhatjuk a legegyszerűbb *parskip* csomagot. Az csomag definiálása: `\usepackage{parskip}`

A *parskip* csomag frissítése tartalmazza a *parfill* opciót is:

```
\usepackage[parfill]{parskip}
```

amely elkerüli a téglalap alakú bekezdéseket, mivel biztosítja, hogy az utolsó sor végén mindig legyen szóköz. Amennyiben szeretnénk használni a *parfill* opciót, érdemes megadni a frissítés időpontját:

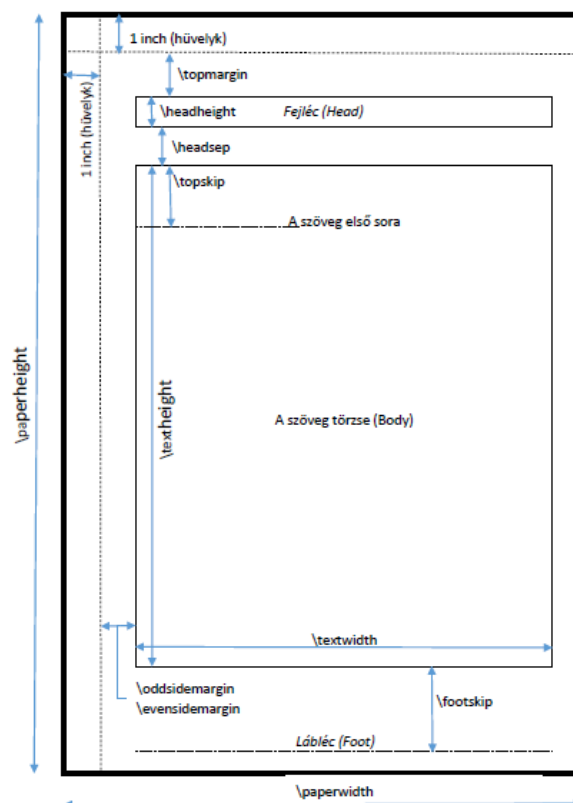
```
\usepackage[parfill]{parskip}[2001/04/09]
```

Amennyiben a *parskip* csomag aktuális verziója az adott dátumnál korábbi, figyelmeztetés jelenik meg a feldolgozás során, hogy ismeretlen a *parfill* opció.

3.2.5 Oldalformátum

Minden oldal egy fejlécből, a tényleges szöveget tartalmazó törzsből és egy láblécből áll. Az oldalstílus kiválasztása határozza meg, hogy milyen információk találhatóak a fejlécben és a láblécben.

A LATEX alapértelmezett értékeket használ a fejléc, a törzs és a lábléc közötti távolságra, a felső és a bal margóra, valamint a fejléc, a törzs és a lábléc szövegsorának szélességére és magasságára. Ezeket a formázási hosszokat a következő ábra mutatja:



4. ábra. Oldalbeállítások.

A `\setlength` paranccsal a preambulumban ezek az értékek a megváltoztathatók. Például:

```
\setlength{\textwidth}{14cm}
```

hogy a szövegsor szélessége 14 cm legyen.

Létezik a `\linewidth` parancs is, amely megegyezik a szövegsor szélességével abban a környezetben, amelyben éppen tartózkodunk. Ezt soha nem szabad megváltoztatni, akkor használjuk, ha ismerni szeretnénk ezt a szélességet.

A `layout` csomagban található elrendezési csomaggal megvizsgálhatjuk saját oldalelrendezésünket. A `\layout` parancs megadása után egy elrendezési ábrát kapunk az elrendezési paraméterek aktuális értékeinek listájával együtt. Természetesen ezt nem a végleges dokumentumban használjuk, hanem csak diagnosztikai ellenőrzésként kell megtenni.

<code>\oddsidemargin</code>	bal margó páratlan oldalakhoz,
<code>\evensidemargin</code>	bal margó páros oldalakhoz,
<code>\topmargin</code>	felső margó a fejtetőig,
<code>\headheight</code>	fejléc magassága,
<code>\headsep</code>	távolság a címsor aljától a törzs tetejéig,
<code>\topskip</code>	távolság a törzs tetejétől a szöveg első sorának alapvonaláig,
<code>\textheight</code>	a törzsszöveg magassága,
<code>\textwidth</code>	a törzsszöveg szélessége,
<code>\footskip</code>	távolság a test aljától a lábléc aljáig,

`\paperwidth`, `\paperheight` a lap teljes szélessége és magassága a lapméret opciótól függően, beleértve az összes margót.

Az oldalkiosztás pontos kiszámításához tudomásul kell vennünk, hogy a LATEX minden távolságot a papír tetejétől és a bal szélétől egy hüvelyknyire lévő ponttól mér. Így a teljes bal margó `\oddsidemargin` plusz egy hüvelyk. A `\paperwidth` és `\paperheight` LATEX paraméterek, amelyek ezt az extra hüvelyket is tartalmazzák, a `\documentclass` parancs papírméret opciójával kapják meg értéküket; belsőleg használják a margók kiszámításához, hogy a szöveg középre kerüljön. A felhasználó számításokhoz is felhasználhatja ezeket.

A `book` dokumentumosztály esetén vagy a kétoldalas opcióval a törzs alsó széle mindig pontosan ugyanazon a helyen jelenik meg minden oldalon. A többi osztályban vagy opcióban ez kissé eltérhet. Az első két esetben az állandó alsó él a belső `\flushbottom`, míg a változó alsó szegélyt a `\raggedbottom` parancs hozza létre. Ezeket a deklarációkat alkalmazva bármikor megváltoztathatjuk az alsó él viselkedését, függetlenül a dokumentum osztályától és beállításaitól.

Feladat:

Hozd létre a következő szövegrészt!

1 Első szakasz

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis. Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris venenatis neque ut vestibulum facilisis. Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero. Proin mattis posuere neque, at sodales massa rhoncus eu.

2 Második szakasz

Suspendisse condimentum fringilla diam, vel egestas urna. Morbi tincidunt, nisi euismod aliquet tincidunt, elit eros finibus sem, nec pharetra ex ipsum sed elit. Donec a dictum magna, quis consequat urna. Integer ut diam laoreet, vestibulum orci ut, pretium turpis. Nam vel felis in sem ultrices maximus. Quisque feugiat purus in libero dapibus gravida. Fusce eu odio et ex imperdiet tempus. Nulla eu sodales magna. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut ut enim ligula. Fusce vehicula lacinia tellus id tristique. Pellentesque nec pretium ex.

Megoldás:

```
\documentclass{article}
\usepackage{fancyhdr}
\setlength{\textwidth}{14cm}
```

```
\begin{document}
\pagenumbering{Alph}
\pagestyle{headings} % Az oldalszámozás és a fejléc bekapcsolása
```

```
\section{Első szakasz}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis. Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris venenatis neque ut vestibulum facilisis. Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero. Proin mattis posuere neque, at sodales massa rhoncus eu.

```
\section{Második szakasz}
```

Suspendisse condimentum fringilla diam, vel egestas urna. Morbi tincidunt, nisi euismod aliquet tincidunt, elit eros finibus sem, nec pharetra ex ipsum sed elit. Donec a dictum magna, quis consequat urna. Integer ut diam laoreet, vestibulum orci ut, pretium turpis. Nam vel felis in sem ultrices maximus. Quisque feugiat purus in libero dapibus gravida. Fusce eu odio et ex imperdiet tempus. Nulla eu sodales magna. Class

```
apient taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.
Ut ut enim ligula. Fusce vehicula lacinia tellus id tristique. Pellentesque nec pretium
ex.
```

```
\end{document}
```

3.2.6 Egyszerűsített oldalformázás

Nagyon bonyolult elérni, hogy az oldal elrendezése pontosan olyan legyen, ahogyan szeretnénk. A szövegnek az oldalon való középre állítása összetett beállításokat foglal magában, amelyek egyáltalán nem intuitívak. Ebben *geometry* csomag jelentős segítséget nyújt. Ezzel a csomaggal egyszerűen megadhatunk értékeket néhány elrendezési paraméterhez, a többit pedig a teljes papírméret figyelembevételével automatikusan beállítja a program. Például a következő képpen adjuk meg a beállításokat: a `\textwidth` paranccsal 14 cm-re és a `\textheight` paranccsal 24 cm-re állítsuk be A4-es papíron az értékeket.

```
\usepackage{geometry}
```

```
\geometry{a4paper,textwidth=15cm,textheight=25cm}
```

Az előbbi parancsok automatikusan beállítják a `\oddsidemargin` és `\topmargin` értékeket is, hogy a szöveg vízszintesen és függőlegesen középre kerüljön, beleértve a fej- és lábléceket is. Vagy beállíthatja az összes margót 1 inch-re az amerikai levélpapíron a következővel:

```
\geometry{letterpaper,margin=1in}
```

Ahelyett, hogy a `\geometry` parancsot használnánk, a paramétereket a `\usepackage` opciójaként is megadhatjuk, például a következő módon

```
\usepackage[a4paper,left=3cm,right=2cm]{geometry}
```

ahol a bal és a jobb margót meghatározott értékekre, a `\textwidth` pedig a maradékra állítja be. A papírméretet vagy a `\documentclass` parancsból kapjuk, vagy előre definiált opcióként adjuk meg, mint *a4paper*, vagy explicit módon adjuk meg, mint *paperwidth=pwidth* és *paperheight=pheight*, vagy mint *papersize={pwidth,pheight}*.

Alapértelmezés szerint a többi elrendezési paraméter úgy van beállítva, hogy a `\textwidth` a `\paperwidth` 80%-a, továbbá `\textheight`, `\headheight`, `\headsep`, `\footskip` a `\paperheight` 90%-a, vízszintesen és függőlegesen középre igazítva.

A szöveg szélessége és magassága explicit módon beállítható a *width=width* és *height=height*, vagy a `body={width,height}` paranccsal. A margókat ezután középre állítjuk. Itt a *height* a teljes szövegmagasságot jelenti, beleértve a fej- és lábléceket is.

A margók megadhatók explicit módon a *left=lmarg* és *right=rmarg*, vagy a *hmargin={lmarg,rmarg}* paraméterekkel. Hasonlóképpen a *top=tmarg* és *bottom=bmarg*, vagy *vmargin={tmarg,bmarg}*. Ha egy párból csak az egyik értéket adjuk meg, a másik értéket

ugyanarra az értékre állítjuk, kivéve, ha a szöveg megfelelő magasságát vagy szélességét konkrétan megadtuk. Az összes margó közös értékre állítható a `margin=marg` paranccsal.

A LATEX szabványos `\oddsidemargin` és `\topmargin` értékektől eltérően a `geometry` margóit a papír szélétől mérjük, nem pedig egy 1 hüvelykkel távolabbi ponttól.

A `nohead`, `nofoot` és `noheadfoot` parancsokkal a geometriát arra utasítjuk, hogy a megfelelő fej- vagy lábléceket ne vegye figyelembe a magasság kiszámításánál. Így a `noheadfoot` esetén a teljes magasság megegyezik a `\textheight` értékkel.

Az `includemp` esetén a széljegyzet `\marginparwidth` és `\marginparsep` paraméterek szerepelnek a teljes szélesség kiszámításában, amely így kisebb, mint a `\textwidth`. A `reversemp` használatával a széljegyzetek a bal oldali margón jelennek meg.

A szöveg szélessége és magassága a papírméret töredékére állítható a `hscale=h` és `vscale=v`, vagy a `scale={h,v}` értékekkel. A `scale=s` esetén a `h` és a `v` értéke is `s`. Például a `\geometry{scale=0.8}` a szélességet és a magasságot a `\paperwidth` és a `\paperheight` 80%-ára állítja be.

Kétoldalas kimenethez adjuk hozzá a `twoside` opciót. Ebben az esetben a bal és a jobb margó értékei páros oldalszámok esetén változnak. Ezenkívül a páratlan és páros oldalak bal oldali margójához `20pt` mennyiséget adunk hozzá, illetve vonunk le. Ez az érték a `twosideshift=shift` paranccsal módosítható, amely a `twoside` opciót is automatikusan beállítja.

A `verbose` opcióval az összes elrendezési paraméter kiszámított értékei a monitorra és az átiratfájlba kerülnek kiírásra.

Mint látható, a `geometry` csomag óriási segítséget nyújt az oldalkiosztás beállításához, meglehetősen intuitívan működik, és a meg nem adott paraméterek értékeit automatikusan úgy állítja be, ahogyan általában egyébként is szeretnénk.

Feladat:

Állítsuk be az előző feladatban az oldal paramétereit a `geometry` csomag segítségével a következőkre: A4 papírméret, szövegszélesség 15 cm és a szöveg magassága 25 cm!

Megoldás:

```
\documentclass{article}
\usepackage{geometry}
\geometry{a4paper,textwidth=15cm,textheight=25cm}

\begin{document}
\pagenumbering{Roman}
\pagestyle{headings}
```

```
\section{Első szakasz}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis. Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris venenatis neque ut vestibulum facilisis. Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero. Proin mattis posuere neque, at sodales massa rhoncus eu.

```
\end{document}
```

3.2.7 Egy- és kétszlopos oldalak

A dokumentumosztály két oszlop beállítása a teljes dokumentumot oldalanként két oszlopra állítja. Az alapértelmezett érték oldalanként egy oszlop. Az egyes oldalak egy vagy két oszlopban adhatók ki a következő deklarációkkal:

```
\twocolumn[fejléc szövege]
```

Befejezi az aktuális oldalt, újat indítva oldalanként két oszloppal. Az opcionális fejléc szövege az oldal tetejére kerül egy oszlopba a teljes oldal szélességével.

```
\onecolumn
```

Befejezi az aktuális kétszlopos oldalt, és oldalanként egy oszloppal folytatja.

Az egyoszlopos formátumhoz képest a *twocolumn* beállítás automatikusan módosít bizonyos oldalstílus-paramétereket, például a behúzást. Ez nem fordul elő a `\twocolumn` paranccsal. Ha szükséges, ezeket a további változtatásokat a megfelelő `\setlength` paranccsal kell elvégezni. Ha a dokumentum nagy része kétszlopos formátumú, akkor az osztály opciót kell előnyben részesíteni.

További oldalstílus-paraméter a `\columnwidth`, egy szövegoszlop szélessége. Egyoszlopos szöveg esetén ez ugyanaz, mint a `\textwidth`, de ha két oszlop lett kiválasztva, a LATEX a `\textwidth` és `\columnsep` értékeiből számítja ki. Ezt a paramétert soha nem szabad megváltoztatnunk.

A `\linewidth` hossz még általánosabb, az aktuális környezetben mindig tartalmazza a szöveg sorszélességét. Egy oszlopon belül megegyezik a `\columnwidth` értékkel.

3.2.8 Többoszlopos szöveg

A `\twocolumn` és `\onecolumn` parancsok mindig új oldalt kezdenek, és ha a kéthasábos szöveg befejeződik, a két oszlop nem egyenlő hosszúságú. Ezeket a problémákat a *multicol* csomag

oldja meg, amely akár 10 oszlopnyi szöveget is támogat. A csomag betöltése után az oldal közepén lévő oszlopok számát a gombbal lehet váltani

```
\begin{multicols}{num cols}[header text][pre space]
```

Szöveg beállítva *num cols* számú oszlopokban

```
\end{multicols}
```

ahol az opcionális fejléc szövegét az összes oszlopra írják, mielőtt hasábokra váltanak.

A többoszlopra váltás előtti és utáni oldaltörés bizonyos automatikus vezérlését a két hossz `\premulticols` és a `\postmulticols` parancsok végzik. Amennyiben az aktuális oldalon a fennmaradó hely kevesebb, mint a `\premulticols`, akkor egy új oldal indul, mielőtt átváltana több oszlopra. Hasonlóképpen, ha a környezet végén kevesebb van, mint a `\postmulticols` az oldalon, a folytatás előtt beszúr egy oldaltörést. Ezen hosszúságok standard értékeit a módosíthatjuk a `\setlength`, vagy `\premulticols` parancsokkal, a második opcionális *pre space* argumentum felülírásával.

A kétoszlopos szövegekre érvényes `\columnsep` és `\columnseprule` hosszúságok a *multicols* környezetre is érvényben vannak, hogy beállítsák az oszlopok közötti hézag szélességét, illetve egy lehetséges elválasztási szabályt.

Létezik csillagozott verzió is:

```
\begin{multicols*}{num cols} . . .
```

```
\end{multicols*}
```

, amelynél az utolsó oldalon lévő oszlopok nincsenek kiegyensúlyozva, és a maradék hely az utolsó oszlopba kerül.

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

A felosztott szöveg fejléce

A bal oldalon levő oszlopban jelenik meg ez a szöveg, amely átfolyik a jobb oldali oszlopba.	A jobb oldalon levő oszlopban jelenik meg ez a szöveg, amely szintén átfolyik a bal oldali oszlopba.
--	--

Ez a szöveg már csak egy oszlopban jelenik meg.

Megoldás:

```
\documentclass{article}
\usepackage{multicol}

\begin{document}
```

```
\begin{center}
\LaTeX \\
\end{center}
```

```
\begin{multicols}{2}[A felosztott szöveg fejléce]
```

A bal oldalon levő oszlopban jelenik meg ez a szöveg, amely átfolyik a jobb oldali oszlopba.

A jobb oldalon levő oszlopban jelenik meg ez a szöveg, amely szintén átfolyik a bal oldali oszlopba.

```
\end{multicols}
```

Ez a szöveg már csak egy oszlopban jelenik meg.

```
\end{document}
```

3.3 A dokumentum részei

Minden dokumentum fejezetekre, szakaszokra, alszakaszokra és így tovább van felosztva. Lehet egy függelék a végén, az elején címlap, tartalomjegyzék, kivonat, stb. A LATEX számos jelölőparancsot tartalmaz ezeknek a struktúráknak a meghatározására. Ezenkívül a címsorok sorszámozása és alszámozása automatikusan megtörténik. Egy tartalomjegyzék is létrehozható egyetlen paranccsal.

Egyes szakaszoló parancsok hatása a kiválasztott dokumentumosztálytól függ, és nem minden parancs érhető el minden osztályban.

3.3.1 Címlap

Címlap készíthető a környezettel formázatlan formában is

```
\begin{titlepage} Címoldal szövege \end{titlepage}
```

vagy a

```
\title{Címszöveg}
```

```
\author{Szerző neve és címe}
```

```
\date{Dátum szöveg}
```

```
\maketitle
```

parancsokkal, előre programozott LATEX stílusban.

A címlap szabványos LATEX elrendezésében minden bejegyzés azon sorok középpontjában van, amelyekben megjelennek. Ha a cím túl hosszú, akkor sortörés lesz.

```
\title {Szerkesszünk LATEXben }
```

```
\author {Fischer Benjámín\thanks{Tel. [+36] 20 1234 567}}\\
```

LATEX kutató központ

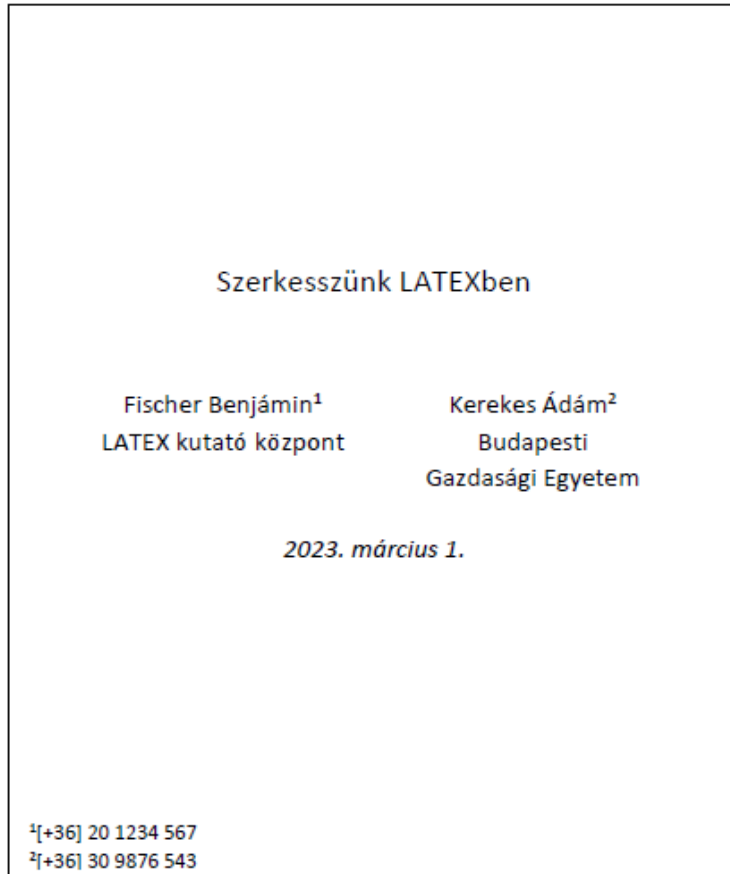
```
\and
```

Kerekes Ádám

\thanks {Tel. [+36] 30 9876 543}\

Budapesti\ Gazdasági Egyetem}

\maketitle



5. ábra. Címlap minta.

A töréspontokat a szerző maga is kiválaszthatja a `\` paranccsal, azaz a `\title{...\\...\\...}` megadásával.

Ha több szerző van, akkor a nevüket elválaszthatja `\and`-el egymástól, például `\author{ Fischer Benjámín \and Kerekes Ádám}`. Ezek a nevek egy sorba kerülnek egymás mellé. A következő szekvencia:

```
\author{Szerző1\Intézmény1\Cím1 \and Szerző2\Intézmény2\Cím2}
```

külön-külön középre helyezi a bejegyzéseket, soronként egyet a *Szerző1*, *Intézet1*, *Cím1* és *Szerző2*, *Intézet2*, *Cím2* készletek mindegyikében, és a címlapon egymás mellé helyezi a két központosított bejegyzésblokkot.

Ahelyett, hogy a szerzők neveit egymás mellé nyomtatnánk, egymásra helyezhetjük őket az `\and` vagy a `\\` paranccsal. Ebben az esetben a függőleges távolság a `\\` után egy opcionális hossz-meghatározással [*space*] állítható be.

Ha a `\date` parancsot kihagyjuk, az aktuális dátum automatikusan kinyomtatásra kerül a címlap szerzői bejegyzései alá. Másrészt a `\date{Date text}` parancs a *Date text* szöveget helyezi az aktuális dátum helyére. Bármilyen kívánt szöveg beszúrható ide, beleértve a sortörés parancsokat `\\` egynél több középre igazított szövegsorhoz.

A `\thanks{Lábjegyzet szövege}` parancs a cím, a szerző vagy a dátum szövegének bármely pontján megadható. Ez egy jelölőt helyez el arra a pontra, ahol a parancs megjelenik, és lábjegyzet szövegét lábjegyzetként írja a címlapra.

A címloldal a `\title`, `\author`, `\date` és `\thanks` parancsokkal jön létre, a `\maketitle` parancs megadása hatására. Maga a címlap nem rendelkezik oldalszámmal, a következő dokumentum első oldala pedig 1-el kezdődik. Külön címlap csak a *book* és *report* osztályú dokumentumokhoz készül. *Article* esetén a `\maketitle` parancs létrehoz egy címfejléct az első oldalon a `\title`, `\author` és ha van, akkor a `\date` és `\thanks` parancsok középre helyezett bejegyzéseivel. Ha a *titlepage* dokumentumosztály opció került megadásra, akkor a cím még az *article* esetében is külön oldalon jelenik meg.

Feladat:

Hozd létre a következő címlapot!

Észak-Amerikai Sólyomkutató Intézet



A sólyom ébredése

Sam Hawkins

2023

Megoldás:

```
\documentclass{article}
\usepackage{graphicx}

\title{
Észak-Amerikai Sólyomkutató Intézet
\vskip2cm
\includegraphics[width=6cm]{Hawk.png}
\vskip2cm
\LARGE{A sólyom ébredése}
\vskip2cm
}
\author{\Huge Sam Hawkins}

\date{2023}

\begin{document}

\maketitle
```

```
\end{document}
```

3.3.2 Absztrakt

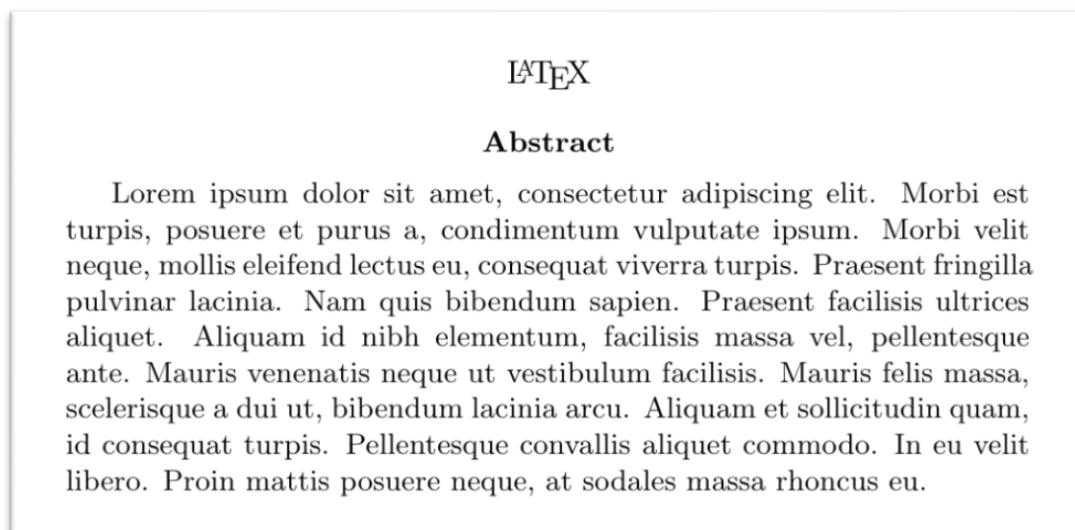
Az absztrakt a következő paranccsal készül:

```
\begin{abstract} Szöveg az absztrakthoz \end{abstract}
```

A *report* dokumentumosztályban az absztrakt külön oldalon jelenik meg oldalszám nélkül; *article* esetében az első oldalon a címfejléc után következik, kivéve, ha a *titlepage* dokumentumosztály opció van kiválasztva, ebben az esetben külön oldalra is nyomtatjuk. A *book* dokumentumosztály esetében absztrakt nem lehetséges.

Feladat:

Hozd létre a következő absztraktot!



Megoldás:

```
\documentclass{article}
```

```
\begin{document}
```

```
\begin{center}
```

```
\LaTeX\
```

```
\end{center}
```

```
\begin{abstract}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis. Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris venenatis neque ut vestibulum facilisis. Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin quam, id consequat

```
turpis. Pellentesque convallis aliquet commodo. In eu velit libero. Proin mattis posuere
neque, at sodales massa rhoncus eu.
\end{abstract}

\end{document}
```

3.3.3 Szakaszok

A következő parancsok állnak rendelkezésre az automatikus, szekvenciális szakaszolás létrehozásához:

```
\part      \chapter      \subsection      \paragraph
           \section      \subsubsection      \subparagraph
```

A `\part` kivételével ezek a parancsok szakaszolási hierarchiát alkotnak. A *book* és *report* dokumentumosztályokban a legmagasabb tagolási szint a `\section`. A fejezetek szakaszokra vannak osztva a `\section` paranccsal, amelyek további felosztása a `\subsection` segítségével történik. Az *article* dokumentumosztályban a hierarchia a `\section` karakterlánccal kezdődik, mivel a `\chapter` nem érhető el.

Mindezen parancsok szintaxisa:

```
\sec command[short title]{title} vagy
\sec command*{title}
```

Az első esetben a szakasz sorszámot kap a sorozatban, amely azután a szövegcímet használó címsorral együtt kerül kinyomtatásra. A *short title* a tartalomjegyzék bejegyzésévé és az oldal fejlécévé válik. Ha az opcionális *short title* kimarad, akkor a *title* beállítással lesz egyenlő; ez a normális helyzet, kivéve, ha a *title* túl hosszú a többi bejegyzéshez.

A második (*-os alak) esetben nem kerül kinyomtatásra a szakaszszám és nem történik bejegyzés a tartalomjegyzékbe.

A címfejléc mérete és a számozás mélysége a szakaszoló parancs hierarchián belüli pozíciójától függ. Az *article* dokumentumosztály esetében a `\section` parancs egyetlen számot, a `\subsection` parancs két számot generál, a két rész között egy ponttal (például 3.2).

A *book* és *report* dokumentumosztályokban a fejezetcímek egyetlen számot kapnak a `\chapter` paranccsal, a `\section` parancs két számot hoz létre, és így tovább. Továbbá a `\chapter` parancs mindig új oldalt kezd.

Minden szakaszolási parancshoz tartozik egy belső számláló, amely a parancs minden egyes meghívásakor eggyel nő, és a következő magasabb szakaszoló parancs minden egyes hívásánál nullára áll vissza. Ezeket a számlálókat nem változtatja meg a *-os alak, ami nehézségekhez vezethet, ha a parancsok szabványos és *-os alakjait úgy keverik össze, hogy a *-os alakok magasabban vannak a hierarchiában, mint a szabványos formák. Azonban nincs gond, ha a *-

os alakok mindig alacsonyabb szinten vannak, mint a szabványos formák. A `\section ... \subsection ... \subsubsection* ...` szekvencia számozza a `\section` és `\subsection` fejléceit, míg a `\subsubsection` fejléceit számozás nélkül hagyja.

A `\part` szakaszolási parancs speciális eset, és nem befolyásolja a többi parancs számozását. A szakaszok automatikus számozása azt jelenti, hogy a számok nem feltétlenül ismertek az írás időpontjában. Előfordulhat, hogy a szerző törli őket a végső sorrendből, vagy később új részeket vezet be, vagy akár eltávolít néhányat.

A `\label{name}` és `\ref{name}` parancsok, amelyek közül az első kulcsszót rendel a szakaszszámhoz, míg a második hivatkozásként használható a szövegben a szám nyomtatásához. A kulcsszó neve betűk, számok vagy szimbólumok tetszőleges kombinációja lehet.

A második hivatkozási parancs a `\pageref`, amely az oldalszámot nyomtatja ki, ahol a megfelelő `\label` meg van határozva.

A hivatkozási parancsok sok más helyzetben is használhatók automatikusan számozott elemek, például ábrák, táblázatok, egyenletek címkézésére.

Minden szakaszoló parancshoz egy szintszám van hozzárendelve, így a `\section` mindig 1. szintű, `\subsection` 2. szintű, . . . `\subparagraph` szintje pedig 5. Az *article* dokumentumosztályban a `\part` a 0, míg a *book* és a *report* osztályokban a `\part` az 1. szint és a `\chapter` 0. szint lesz. A szakaszok számozása a *secnumdepth* szám által megadott szintig történik. Ez a korlát *book* és *report* esetén 2, *article* esetén 3. Ez azt jelenti, hogy a *book* és a *report* esetében a szakaszok számozása csak az `\subsection` szintjéig terjed ki, az *article* esetében pedig a `\subsection` alszakasz szintjéig.

A szakaszszámozás szintjének növeléséhez (vagy csökkentéséhez) a *secnumdepth* értékét módosítani kell. Ez a `\setcounter{secnumdepth}{num}` paranccsal történik.

Az *article* *num* értéke 0-tól 5-ig, a *book* és *report* esetében pedig 1-től 5-ig terjedhet.

Egy dokumentumon belüli szakaszoló parancs kezdeti értékének megváltoztatására a `\setcounter{sec name}{num}` paranccsal van lehetőség, ahol a *sec name* a szakaszoló parancs neve a `\` megelőző karakter nélkül. Ez az eljárás akkor lehet hasznos, ha az egyes szakaszokat a LATEX egyetlen fájlként dolgozza fel. Például a `\setcounter{chapter}{2}`, ahol a `\chapter` számlálót 2-re állítja. A számláló a következő `\chapter` híváskor növekszik, amely a „3. Fejezet”-et eredményezi.

Feladat:

Hozd létre a következő szövegrészt!

1 Első szakasz

Ez az első szakasz szövege.

1.1 Első szakasz első alszakasza

Ez az első alszakasz szövege.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis.

1.2 Első szakasz második alszakasza

Ez a második alszakasz szövege.

Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris venenatis neque ut vestibulum facilisis.

2 Második szakasz

Ez a második szakasz szövege.

Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero. Proin mattis posuere neque, at sodales massa rhoncus eu.

Megoldás:

```
\documentclass{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\section{Első szakasz}
Ez az első szakasz szövege.

\subsection{Első szakasz első alszakasza}
Ez az első alszakasz szövege.\par
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et
purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu,
consequat viverra turpis.

\subsection{Első szakasz második alszakasza}
```

```

Ez a második alszakasz szövege.\par
Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilis ultrices
aliquet. Aliquam id nibh elementum, facilis massa vel, pellentesque ante. Mauris
venenatis neque ut vestibulum facilis.

\section{Második szakasz}
Ez a második szakasz szövege.\par
Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin
quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero.
Proin mattis posuere neque, at sodales massa rhoncus eu.

\end{document}

```

3.3.4 Függelék

A következő paranccsal a dokumentumunkba egy függelék kerül be:

```
\appendix
```

Ennek hatása az *article* szakaszszámlálója, a *book* és a *report* fejezetszámlálója nullázódik, és ezen szakaszoló parancsok számozásának formája számokból *A*, *B* nagybetűkre módosul. Továbbá a „Fejezet” szó lecserélődik. „Függelék” által úgy, hogy a következő fejezetcímek előtt az „A. Függelék”, „B. Függelék” stb. szerepel. Az alsó tagolási parancsok számozása a fejezetszám helyett a betűt tartalmazza.

3.3.5 A könyv szerkezete

A könyv szerkezetének egyszerűsítésére a

```
\frontmatter
```

előszó, tartalomjegyzék

```
\mainmatter
```

szövegtörzs

```
\backmatter
```

bibliográfia, tárgymutató, kolofon

parancsok szolgálnak (*book* dokumentumosztályban). A `\frontmatter` parancs átváltja az oldalszámozást római számokra, és megszünteti a fejezetek számozását. A `\mainmatter` 1-re visszaállítja az oldalszámozást arab számokkal, és újra aktiválja a fejezetszámozást, majd ez ismét ki lehet kapcsolni a `\backmatter`-rel.

3.4 Tartalomjegyzék

3.4.1 Automatikus bevitelek

A LATEX képes automatikusan elkészíteni és kinyomtatni a teljes dokumentum tartalomjegyzékét. Tartalmazza a szakaszszámokat és a megfelelő címsorokat a szakaszoló parancsok szabványos formájában, valamint az oldalszámokat, amelyeken ezek kezdődnek. A

következő paranccsal a preambulumban állítható be, hogy a tartalomjegyzékbe milyen vágási mélységig kerüljön bejegyzés:

```
\setcounter{tocdepth}{num}
```

A *num* értéknek pontosan ugyanaz a jelentése és hatása, mint a fent leírt számláló *secnumdepth* esetében, amellyel az automatikus alszekció maximális szintje van rögzítve. Alapértelmezés szerint a bejegyzések tartalomjegyzékben való szerepeltetésének mélysége megegyezik az automatikus szakaszolás szabványos szintjével: azaz a `\subsection` szintre a *book* és a *report*, és a `\subsubsection` szint az *article* esetében.

3.4.2 A tartalomjegyzék kinyomtatása

A `\tableofcontents` paranccsal létrejön és kinyomtatja a tartalomjegyzéket, amely a tartalomjegyzék megjelenési helyén van megadva, ami általában a címlap és az absztrakt után van.

3.4.3 További bejegyzések

A *-os alakú tagolási parancsok nem kerülnek be automatikusan a tartalomjegyzékbe. A beszúrásához vagy bármely más további bejegyzéshez

```
\addcontentsline{toc}{sec name}{entry text}
```

```
\addtocontents{toc}{entry text}
```

parancsok használhatók.

Az első paranccsal a bejegyzések megfelelnek a tartalomjegyzék formátumának, ahol a szakaszok fejlécei jobban behúzódnak, mint a fejezetek, de kisebbek, mint az alfejezetek esetében. Ezt a *sec name* argumentum értéke határozza meg, amely megegyezik a `\` karakter nélküli szakaszoló parancsok egyikével (például *section*). A bejegyzés szövege az oldalszámmal együtt bekerül a tartalomjegyzékbe. Ez a parancs akkor hasznos, ha számozatlan szakaszcímeket ír be a tartalomjegyzékbe. Például

```
\section*{Szerzői címek}
```

```
\addcontentsline{toc}{section}{Szerzői címek}
```

Az `\addtocontents` parancs a kívánt parancsot vagy szöveget behelyezi a *.toc* fájlba. Ez lehet egy formázási parancs, például a `\newpage`, amely a tartalomjegyzék kinyomtatásakor lép életbe.

Feladat:

Hozd létre az előző feladat tartalomjegyzékét!

Contents

1	Első szakasz	1
1.1	Első szakasz első alszakasza	1
1.2	Első szakasz második alszakasza	1
2	Második szakasz	1

1 Első szakasz

Ez az első szakasz szövege.

1.1 Első szakasz első alszakasza

Ez az első alszakasz szövege.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu, consequat viverra turpis.

1.2 Első szakasz második alszakasza

Megoldás:

```

\documentclass{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\tableofcontents

\section{Első szakasz}
Ez az első szakasz szövege.

\subsection{Első szakasz első alszakasza}
Ez az első alszakasz szövege.\par
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est turpis, posuere et
purus a, condimentum vulputate ipsum. Morbi velit neque, mollis eleifend lectus eu,
consequat viverra turpis.

\subsection{Első szakasz második alszakasza}
Ez a második alszakasz szövege.\par
Praesent fringilla pulvinar lacinia. Nam quis bibendum sapien. Praesent facilisis ultrices
aliquet. Aliquam id nibh elementum, facilisis massa vel, pellentesque ante. Mauris
venenatis neque ut vestibulum facilisis.

```

```
\section{Második szakasz}
Ez a második szakasz szövege.\par
Mauris felis massa, scelerisque a dui ut, bibendum lacinia arcu. Aliquam et sollicitudin
quam, id consequat turpis. Pellentesque convallis aliquet commodo. In eu velit libero.
Proin mattis posuere neque, at sodales massa rhoncus eu.

\end{document}
```

3.4.4 Egyéb listák

A tartalomjegyzéken kívül a LATEX automatikusan generálhat és nyomtathat ábrák és táblázatok jegyzékeit is (listákat). A listák létrehozásához szükséges parancsok a következők:

```
\listoffigures beolvassa és/vagy létrehozza a file.lof fájlt
```

```
\listoftables beolvassa és/vagy létrehozza a file.lot fájlt
```

Ezekben a listákban a bejegyzések automatikusan a `\caption` paranccsal jönnek létre az ábra- és táblázatkörnyezetben. A további bejegyzések ugyanazokkal a parancsokkal készülnek, mint a tartalomjegyzéknél, amelynek általános formája a következő

```
\addcontentsline{file}{format}{entry}
```

```
\addtocontents{file}{entry}
```

parancsok, ahol a *file* a *.toc* (tartalomjegyzék), *.lof* (ábrák listája) vagy *.lot* (táblázatok listája) típusának egyikét jelenti. A *format* argumentum a tartalomjegyzék szakaszoló parancsainak egyike, a fent leírtak szerint, vagy az ábra az ábrák listájához, vagy a táblázat a táblázatok listájához. Az *entry* argumentum azt a szöveget jelöli, amelyet a megfelelő fájlba kell beilleszteni.

4 Szöveg megjelenítése

A szöveg megjelenítésének vagy kiemelésének többféle módja van: a betűstílus vagy a betűméret megváltoztatása, a középre állítás, a behúzás, a bekezdések megjelölése, stb.

4.1 Betűtípus módosítása

Általában a tipográfiában egy bizonyos méretű és megjelenésű betűk, számok és karakterek halmazát betűtípusnak nevezzük. A dokumentum elején a `\documentclass` utasítással megadjuk a LATEX szabványos betűtípusát, amely a törzsszövegrészben függőleges, *Roman*, közepes magasságú. A három lehetséges alapméret a 10, 11 és 12 pt, a következő méretopcióktól függően: *10pt* (alapértelmezett), *11pt* és *12pt*.

4.1.1 Kiemelés

A gépirással írt kéziratban a szöveg kiemelésének szokásos módja az aláhúzás. A szedő az aláhúzott szöveget dőlt betűssé alakítja a nyomtatott változathoz. A szabványos szövegről a

hangsúlyos szövegre való váltás a LATEX-ben az `\emph` paranccsal vagy az `\em` deklarációval történik.

Az `\em` deklaráció ugyanúgy működik, mint a többi, alább leírt betűtípus-deklaráció: a betűtípus-változtatás addig marad érvényben, amíg egy másik megfelelő deklaráció meg nem változtatja, vagy az aktuális környezet végéig. Környezet létrehozható egy pár kapcsos zárójellel is `{...}`. Az `\emph` parancs viszont csak a következő argumentumban szereplő szövegen működik. Ez a legegyszerűbb módja a rövid szövegrészek *kiemelésének*, például:

Ez a legegyszerűbb módja a rövid szövegrészek `\emph{kiemelésének}`, ...

A `\em` deklaráció megfelelőbb olyan hosszabb szövegekhez, amelyek egy környezetbe vannak zárva, *névvel vagy név nélkül*.

... környezetbe vannak zárva, `{\em névvel vagy név nélkül.}`

Nézzük meg a különbséget a deklaráció és parancs között. A deklaráció addig marad érvényben, amíg a helyi környezet a záró kapcsos zárójellel véget nem ér, a parancs esetében csak a kapcsos zárójelek közé zárt argumentumra működik. Egy másik különbség, hogy az `\emph` parancs szükség esetén automatikusan extra szóközt szúr be a végére, az úgynevezett dőlt korrekciót, hogy javítsa a megjelenést a ferde és függőleges betűtípusok határfelületén.

A deklaráció és a parancs is hangsúlyozó betűtípusra vált. Ez azt jelenti, hogy ha az aktuális betűtípus függőleges, akkor dőlt betűsre vált, míg ha a szöveg már ferde, akkor függőleges betűtípus kerül kiválasztásra.

Egymásba ágyazott kiemelés lehetséges, és könnyen érthető:

Az `\emph{első}`, `második` és `\emph{harmadik betűtípus}` Az `{\em első}`, `{\em második}` és `{\em harmadik betűtípus}}`

mindkettő esetében a következő lesz az eredmény: „Az *első*, *második* és *harmadik betűtípus*”.

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

Kiemelés az `\emph{}` deklarációval:
Az *első*, *második* és *harmadik betűtípus*.

Kiemelés az `\em` deklarációval:
Az *első*, *második* és *harmadik betűtípus*.

Megoldás:

```
\documentclass{article}
```

```
\begin{document}
```

```
\begin{center}
```

```
\LaTeX\
```

```
\end{center}
```

Kiemelés az `\verb|\emph{}` deklarációval:

Az `\emph{első}`, `\emph{második}` és `\emph{harmadik betűtípus.}` \\

Kiemelés az `\verb|\em|` deklarációval:

Az `{\em első}`, `{\em második}` és `{\em harmadik betűtípus.}}`

```
\end{document}
```

4.1.2 A betűméret kiválasztása

A LATEX-ben a következő deklarációk állnak rendelkezésre a betűméret megváltoztatására:

<code>\tiny</code>	<small>smallest</small>	<code>\Large</code>	larger
<code>\scriptsize</code>	<small>very small</small>	<code>\LARGE</code>	even larger
<code>\footnotesize</code>	<small>smaller</small>	<code>\huge</code>	still larger
<code>\small</code>	<small>small</small>	<code>\Huge</code>	largest
<code>\normalsize</code>	normal		
<code>\large</code>	<small>large</small>		

amelyek mindegyike a dokumentumosztály opcióban kiválasztott szabványos méretéhez van viszonyítva. A szabványos méretet a `\normalsize` paranccsal választjuk ki. A betűméret-deklaráció ugyanúgy viselkedik, mint az összes többi deklaráció: azonnali módosítást hajtanak végre, amely mindaddig érvényben marad, amíg egy másik méretdeklaráció nem változtatja, vagy amíg az aktuális környezet véget nem ér. Ha a kaptos zárójelben adják ki `{..}`, a deklaráció hatása csak a záró kaptos zárójelre terjed ki:

szabványos `{\large nagy \Large nagyobb }` ismét szabványos

szabványos nagy **nagyobb** ismét szabványos

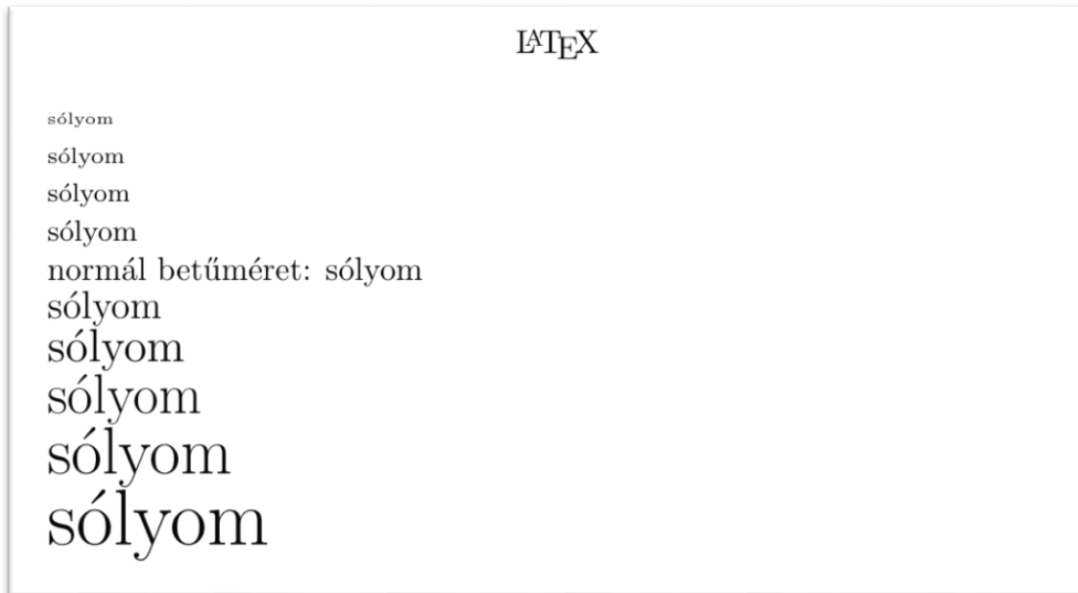
A betűméret módosítása a fenti parancsok egyikével automatikusan megváltoztatja a sorközi távolságot is. Minden betűmérethez tartozik egy megfelelő természetes sortávolság `\baselineskip`. Ez bármikor módosítható. Ha a természetes sortávolság 12 pont, akkor a `\setlength{\baselineskip}{16pt}` parancs 16 pontra növeli.

A bekezdés végén érvényes `\baselineskip` értéke a teljes bekezdés összeállítására szolgál. Ez azt jelenti, hogy ha egy bekezdésen belül több módosítás is történik a `\baselineskip`-ben, akkor csak az utoljára megadott értéket veszi figyelembe a rendszer.

A betűméret minden módosításával a `\baselineskip` visszaáll az adott mérethez tartozó természetes értékre. Minden korábbi `\setlength` beállítás érvénytelen lesz.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

{\tiny sólyom}

{\scriptsize sólyom}

{\footnotesize sólyom}

{\small sólyom}

normál betűméret: sólyom

{\large sólyom}

{\Large sólyom}

{\LARGE sólyom}
```



```
{\huge sólyom}
{\Huge sólyom}
\end{document}
```

4.1.3 Betűtípus-attribútumok

A LATEX 2 ϵ részeként bevezetett New Font Selection Scheme (NFSS) segítségével lehetséges a betűtípusok kiválasztása szigorúan ezen attribútumok alapján. Normál használat esetén azonban létezik néhány deklaráció és megfelelő parancs, amelyek egyszerűsítik ezt az eljárást. A TEX-szel és LATEX-szel biztosított Computer Modern betűtípusokhoz a következő attribútumok és értékek léteznek:

Family (Család): A hagyományos tipográfiai családoknak olyan neveik vannak, mint *Baskerville*, *Bodoni*, *Times Roman*, *Helvetica*, stb. A szabványos LATEX telepítés három családot biztosít deklarációkkal:

<code>\rmfamily</code>	visszaváltás <i>Roman</i> betűtípusra;
<code>\ttfamily</code>	írógép betűtípusra váltáshoz;
<code>\sffamily</code>	a <i>Sans Serif</i> betűtípus kiválasztásához.

Shape (Alak): A szabványos telepítéssel elérhető alaknyilatkozatok a következők:

<code>\upshape</code>	függőleges betűtípusra visszaváltás;
<code>\itshape</code>	a dőlt alak kiválasztásához;
<code>\slshape</code>	a ferde betűtípus kiválasztásához;
<code>\scshape</code>	a Caps és Small Caps használatához.

Series (Sorozat): A lehetséges nyilatkozatok a következők:

<code>\mdseries</code>	visszaváltás közepes szélességű betűtípusra;
<code>\bfseries</code>	félkövér betűtípus kiválasztásához.

Ezek azonban nem merítik ki az összes lehetséges attribútumbeállítást, de lefedik a legszokványosabbakat, különösen a Computer Modern betűtípusok esetében. Más betűtípusok, különösen a PostScript betűtípusok esetében további attribútumértékek léteznek. Ezeket a deklarációkat ugyanúgy használják, mint a többit. Általában egy kapcsos zárójelbe zárva {...}, mint például a `{\scshape Chuck Norris}`, amely a Chuck Norris eredményezi. Hosszabb szövegrészek esetén előnyösebb egy környezet használata:

```
\begin{font style} . . . szöveg új betűtípussal . . . \end{font style}
```

Ez jobban nyomon követi az átállás kezdetét és végét. Betűstílushoz a fenti betűtípus-parancsok bármelyike használható, elhagyva a kezdeti `\` karaktert.

Mivel az egyik attribútum megváltoztatása a többit úgy hagyja, ahogy volt, minden lehetséges kombináció elérhető. Ha először egy félkövér sorozatot választunk `\bfseries`-el, majd egy ferde alakzatot `\slshape`-val, akkor félkövér, ferde betűtípust kapunk. Például:

szabványos és `{\bfseries félkövér és {\slshape ferde}` és újra} vissza

eredmény: szabványos és **félkövér és ferde és újra** vissza

Végül a `\normalfont` deklaráció visszaállítja az összes attribútumot (a méret kivételével) az alapértelmezett értékekre: *Roman*, függőleges, közepes szélességű betűtípus.

4.1.4 Betűtípus-parancsok

Minden fent felsorolt betűtípus-deklarációhoz tartozik egy megfelelő parancs, amely az argumentumát a megadott attribútummal rendelkező betűtípusba állítja be.

Family: `\textrm{text}` `\texttt{text}` `\textsf{text}`

Shape: `\textup{text}` `\textit{text}` `\textsl{text}` `\textsc{text}`

Series: `\textmd{text}` `\textbf{text}`

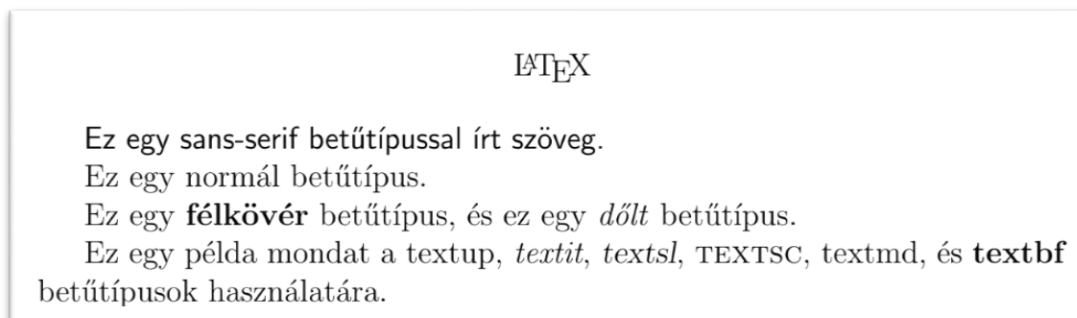
Alapértelmezett: `\textnormal{text}`

Kiemelt: `\emph{text}`

A `\textnormal` argumentuma a `\normalfont` paranccsal kiválasztott szabványos betűtípussal van beállítva.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt]{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\textsf{Ez egy sans-serif betűtípussal írt szöveg.}
```

Ez egy normál betűtípus.

Ez egy `\textbf{félkövér}` betűtípus, és ez egy `\textit{dőlt}` betűtípus.

Ez egy példa mondat a `\textup{textup}`, `\textit{textit}`, `\textsl{textsl}`, `\textsc{textsc}`, `\textmd{textmd}`, és `\textbf{textbf}` betűtípusok használatára.

`\end{document}`

4.1.5 További betűtípusok

LATEXben a felsoroltaknál is több betűtípus és méret létezik. Új betűtípus név szerinti betöltéséhez a következő parancsok használatosak:

`\newfont{\fnt}{name scaled factor }` vagy

`\newfont{\fnt}{name at size }`

ahol hozzárendeli a betűtípust az új `\fnt` nevű parancshoz. Az első esetben a *faktor* a méretezési tényező 1000-szeresének megfelelő szám, amelyet a betűtípus alap- vagy tervezési méretének nagyításához vagy kicsinyítéséhez kell használni. A második esetben a betűméret a megadott méretre (*size*) lesz méretezve. Ferde méretű, *Sans Serif* betűtípus telepítéséhez: 20,74 pt, mint `\sss`, a *cmssi17*-et az alábbi módon töltjük be:

`\newfont{\sss}{cmssi17, 20,74pt}`

Ekkor az `\sss` deklaráció közvetlenül erre a betűtípusra vált, de az alapvonal elválasztásának megváltoztatása nélkül.

4.1.6 Karakterkészletek és szimbólumok

A különböző karakterkészletek mindegyike a saját fájljaiban tárolódik. A karakterkészleten belül minden szimbólum 0 és 127 (vagy 255) közötti számmal van megcímezve. A `\symbol{szám}` parancs létrehozza azt a szimbólumot a belső azonosító számmal az aktuális betűtípussal. A jelen betűtípusban szereplő `@` szimbólum azonosítója a 64, és a `\symbol{64}` paranccsal nyomtatható.

A `\symbol` parancs olyan szimbólumok generálására is használható, amelyekhez nincs más parancs definiálva. Például: `{\ttfamily\symbol{40}}`

4.2 Központozás és behúzás

4.2.1 Középre igazított szöveg

A következő környezet

`\begin{center} 1. sor \\ 2. sor \\ ... \\ n. sor \end{center}`

középre helyezi a `\` paranccsal elválasztott szövegrészeket. Ha a szöveg túl hosszú egy sorhoz, akkor egyetlen szóközzel több sorra osztja, a lehető legjobban kitöltve a teljes sorszélességet, kivéve az utolsó sort. Ebben az esetben szóelválasztás nem fordul elő.

Egy adott környezetben a `\centering` paranccsal lehet középre helyezni a rákövetkező szöveget, ahol ismét a `\` parancsot alkalmazzuk sorrelválasztóként. Ennek a nyilatkozatnak a hatása a környezet végéig tart.

Egyetlen sort is középre lehet helyezni, ha beírjuk a szövegét a TEX parancs argumentumaként: `\centerline{szöveg}`.

4.2.2 Egyoldalú igazítás

A következő környezetek

```
\begin{flushleft} 1. sor \\ 2. sor \\ . . . \\ n. sor \end{flushleft}
```

```
\begin{flushright} 1. sor \\ 2. sor \\ . . . \\ n. sor \end{flushright}
```

balra (*flushleft*) vagy jobbra (*flushright*) igazított szöveget állítanak elő. Ha a szöveg egy része nem fér el egy sorba, akkor többre oszlik el fix szóközzel, ugyanúgy, mint a középre (*center*) esetében. A szóelválasztás ismét nem fordul elő.

Ugyanezek az eredmények hozhatók létre egy környezetben az alábbi deklarációkkal:

```
\raggedright a flushleft környezetet helyettesítve, és
```

```
\raggedleft a flushright környezetet helyettesítve.
```

4.2.3 Kétoldali behúzás

A szöveg egy része úgy jeleníthető meg, hogy mindkét oldalon azonos mértékben behúzza a környezettel együtt

```
\begin{quote} szöveg \end{quote}
```

```
\begin{quotation} szöveg \end{quotation}
```

További függőleges térköz szúr be a megjelenített szöveg fölé és alá, hogy vizuálisan elkülönüljön a normál szövegtől.

A megjelenítendő szöveg bármilyen hosszúságú lehet; lehet egy mondat része, egy egész bekezdés vagy több bekezdés is.

A bekezdések a szokásos módon üres sorral vannak elválasztva, nincs szükség üres sorokra a megjelenített szöveg elején és végén, mivel itt mindenesetre további függőleges térköz kerül beillesztésre.

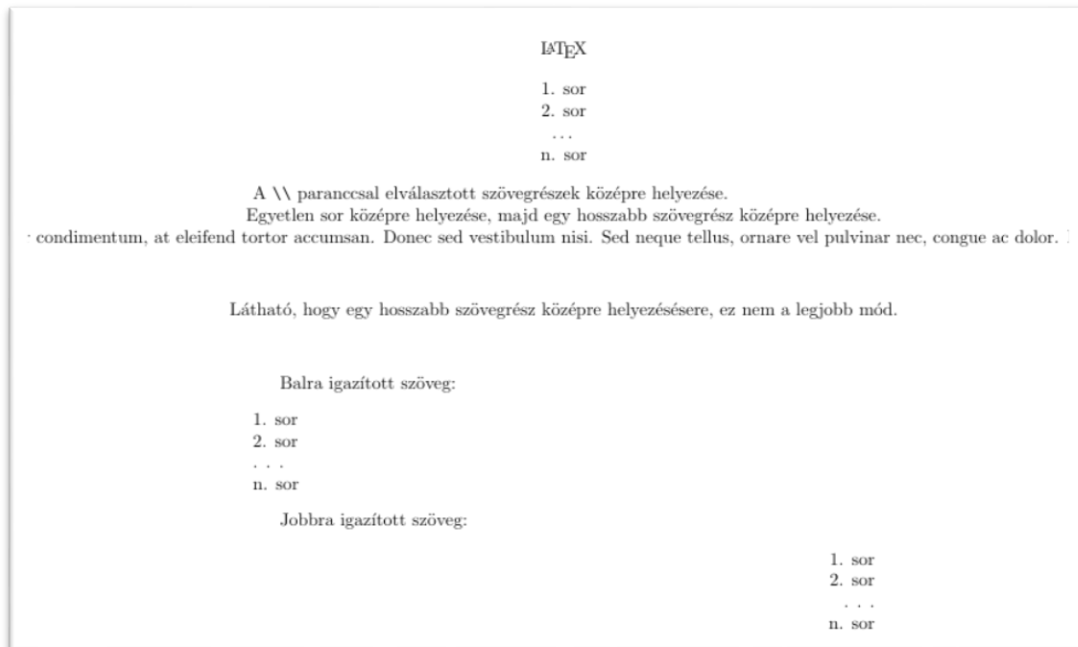
A fenti két forma közötti különbség a következő:

Az idézetkörnyezetben (*quotation*) a bekezdések az első sor extra behúzásával, míg az idézet (*quote*) környezetben nagyobb függőleges térközzel vannak jelölve.

Az idézet környezetnek csak akkor van igazán értelme, ha a normál szöveg az első sor behúzását használja az új bekezdések bemutatására.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass{article}
\begin{document}

\begin{center}
\LaTeX\
\end{center}

\begin{center} 1. sor \\ 2. sor \\ ... \\ n. sor \end{center}
A \verb|\\| parancssal elválasztott szövegrészek középre helyezése.

\centerline{Egyetlen sor középre helyezése, majd egy hosszabb szövegrész középre
helyezése.}

\centerline{Nam porttitor turpis id nisl efficitur lobortis eu non libero. In id sollicitudin
orci, ac iaculis enim. Pellentesque cursus tortor diam. Nunc ullamcorper mattis sapien
ac pharetra. Ut pretium sapien consectetur tortor condimentum, at eleifend tortor
accumsan. Donec sed vestibulum nisi. Sed neque tellus, ornare vel pulvinar nec,
congue ac dolor. Mauris magna leo, facilisis sed sapien vel, dictum aliquam sem.
Maecenas sed tellus gravida metus ornare tristique. Duis feugiat, eros et volutpat
fermentum, ante leo pretium eros, vitae rhoncus libero tortor at sapien.}
\vspace{1cm}
```

```

\centerline{Látható, hogy egy hosszabb szövegrész középre helyezésére, ez nem a
legjobb mód.}
\vspace{1cm}
Balra igazított szöveg:
\begin{flushleft} 1. sor \\ 2. sor \\ . . . \\n. sor \end{flushleft}

Jobbra igazított szöveg:

\begin{flushright} 1. sor \\ 2. sor \\ . . . \\ n. sor \end{flushright}

\end{document}

```

4.2.4 Versek behúzása

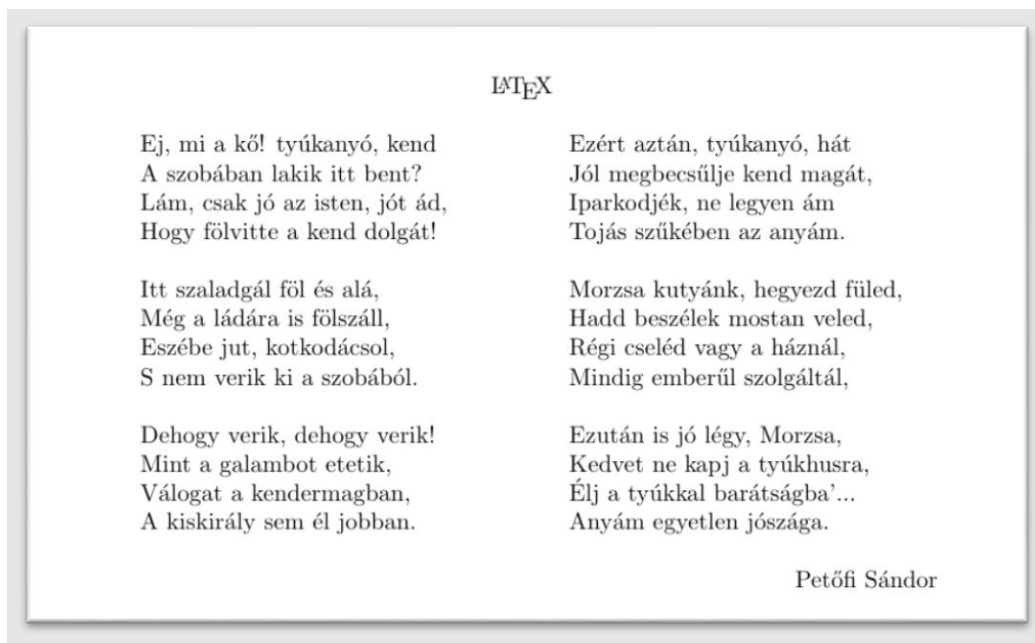
A rímek, versek, versszakok stb. kétoldalas behúzásához a következő környezetet alkalmazzuk:

```
\begin{verse} vers \end{verse}
```

Az előző behúzási sémák egymásba ágyazhatók. Az idézet környezetén belül lehet egy másik idézet, idézet vagy verskörnyezet. Minden alkalommal további behúzások jönnek létre a szöveg mindkét oldalán, és függőleges térközt adunk alá és fölé. Legfeljebb hat ilyen egymásba ágyazás engedett.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```

\documentclass[4pt,a4paper]{article}
\usepackage{multicol}
\usepackage{verse}

```

```

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\begin{multicols}{2}
\settowidth{\versewidth}{Morzsa kutyánk, hegyezd füled,}

\begin{verse}[\versewidth]
Ej, mi a kő! tyúkanyó, kend\
A szobában lakik itt bent?\
Lám, csak jó az isten, jót ad,\
Hogy fölvitte a kend dolgát!\
\space{12pt}
Itt szaladgál föl és alá,\
Még a ládára is fölszáll,\
Eszébe jut, kotkodácsol,\
S nem verik ki a szobából.\
\space{12pt}
Dehogyan verik, dehogyan verik!\
Mint a galambot etetik,\
Válogat a kendermagban,\
A kiskirály sem él jobban.\
\space{12pt}
Ezért aztán, tyúkanyó, hát\
Jól megbecsülje kend magát,\
Iparkodjék, ne legyen ám\
Tojás szűkében az anyám. \
\space{12pt}
Morzsa kutyánk, hegyezd füled,\
Hadd beszéljek mostan veled,\
Régi cseléd vagy a háznál,\
Mindig emberül szolgáltál,\
\space{12pt}
Ezután is jó légy, Morzsa,\
Kedvet ne kapj a tyúkhusra,\
Élj a tyúkkal barátságba'...\
Anyám egyetlen jószága. \
\end{verse}
\end{multicols}

\begin{flushright}
Petőfi Sándor
\end{flushright}

\end{document}

```

4.3 Listák

Három környezet áll rendelkezésre a formázott listák előállításához:

```
\begin{itemize} lista szövege \end{itemize}
```

```
\begin{enumerate} lista szövege \end{enumerate}
```

```
\begin{description} lista szövege \end{description}
```

Ezekben a környezetekben a lista szövege behúzódik a bal margótól, és egy címke vagy jelölő is szerepel benne. A használt címke típusa a kiválasztott listakörnyezettől függ. A címke létrehozásának parancsa az `\item`.

4.3.1 Számozatlan lista

- Az egyes bejegyzéseket fekete ponttal jelölik, ún. címkével.
- A bejegyzések szövege bármilyen hosszúságú lehet. A címke a szöveg első sorának elején jelenik meg.
- Az egymást követő bejegyzéseket további függőleges térköz választja el egymástól. Négy szintig ágyazhatók egymásba.

A fenti szöveg a következőképpen készült:

```
\begin{itemize}
```

```
\item Az egyes bejegyzéseket fekete ponttal jelölik, ún. címkével.
```

```
\item A bejegyzések szövege bármilyen hosszúságú lehet. A címke a szöveg első sorának elején jelenik meg.
```

```
\item Az egymást követő bejegyzéseket további függőleges térköz választja el egymástól. Négy szintig ágyazhatók egymásba.
```

```
\end{itemize}
```

4.3.2 Számozott lista

1. A címkék sorszámokból állnak. Négy szintig ágyazhatók egymásba.
2. A számozás 1-gyel kezdődik a felsorolási környezet minden hívásánál.

A fenti példa a következő szöveggel készült:

```
\begin{enumerate}
```

```
\item A címkék sorszámokból állnak. Négy szintig ágyazhatók egymásba.
```

```
\item A számozás 1-gyel kezdődik a felsorolási környezet minden hívásánál.
```

```
\end{enumerate}
```

4.3.3 Leíró lista

`mintal` Ez a környezet akkor megfelelő, ha több szót vagy kifejezést kell meghatározni.

minta2 Egy kulcsszót használunk címkeként, és a bejegyzés pontosítást vagy magyarázatot tartalmaz.

minta3 Az irodalomjegyzékben szerzőlistaként is használható.

A fenti minta a következő módszerrel készült:

```
\begin{description}
```

```
\item[minta1] Ez a környezet akkor megfelelő, ha több szót vagy kifejezést kell meghatározni.
```

```
\item[minta2] Egy kulcsszót használunk címkeként, és a bejegyzés pontosítást vagy magyarázatot tartalmaz.
```

```
\item[minta3] Az irodalomjegyzékben szerzőlistaként is használható.
```

```
\end{description}
```

Az `\item[option]` parancs tartalmaz egy opcionális argumentumot, amely címkeként félkövéren jelenik meg.

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

Listatípusok

Egyszerű pontozott lista:

- Első elem
- Második elem
- Harmadik elem

Egyszerű számozott lista:

1. Első elem
2. Második elem
3. Harmadik elem

Számozott bekezdések leírással:

1. Első bekezdés.
Ez a bekezdés a "1." elemhez tartozik.
2. Második bekezdés.
Ez a bekezdés a "2." elemhez tartozik.
3. Harmadik bekezdés.
Ez a bekezdés a "3." elemhez tartozik.

Egyszerű leíró lista:

alfa Első bekezdés.

béta Második bekezdés.

gamma Harmadik bekezdés.

Megoldás:

```
\documentclass[4pt,a4paper]{article}
```

```
\begin{document}
```

```
\begin{center}
```

```
\LaTeX\
```

```
\end{center}
```

```
\begin{center}
```

```
{\Huge Listatípusok}
```

```
\end{center}
```

Egyszerű pontozott lista:

```
\begin{itemize}
```

```
\item Első elem
```

```
\item Második elem
```

```
\item Harmadik elem
```

```
\end{itemize}
```

Egyszerű számozott lista:

```
\begin{enumerate}
```

```
\item Első elem
```

```
\item Második elem
```

```
\item Harmadik elem
```

```
\end{enumerate}
```

Számozott bekezdések leírással:

```
\begin{enumerate}
```

```
\item Első bekezdés.
```

Ez a bekezdés a "1." elemhez tartozik.

```
\item Második bekezdés.
```

Ez a bekezdés a "2." elemhez tartozik.

```
\item Harmadik bekezdés.
```

Ez a bekezdés a "3." elemhez tartozik.

```
\end{enumerate}
```

Egyszerű leíró lista:

```

\begin{description}
\item[alfa] Első bekezdés.
\item[béta] Második bekezdés.
\item[gamma] Harmadik bekezdés.
\end{description}

\end{document}

```

4.3.4 Beágyazott listák

A fenti listák egymásba ágyazhatóak, akár vegyesen, akár egy típusban, négy szint mélységig. A használt címke típusa a beágyazás mélységétől függ. A behúzás mindig az adott lista bal margójához viszonyítva történik. Az *itemize* környezet négyszeres egymásba ágyazása a következőképpen jelenik meg:

- Az első szint címkéje egy fekete pont, egy golyó.
 - A második szint egy hosszú kötőjel.
 - * A harmadik szint egy csillag.
 - A negyedik szint címkéje pedig egy egyszerű pont.
 - Ugyanakkor a függőleges távolság a mélység növekedésével csökken.
 - * Vissza a harmadik szintre.
 - Vissza a második szintre.
- És itt vagyunk ismét a tételezés első szintjén.

Hasonlóan a felsorolási környezethez, ahol a számozás stílusa a beágyazási szinttel együtt változik:

1. Az első szinten a számozás arab számokkal, majd ponttal történik.
 - (a) A második szinten kisbetűkkel zárójelben.
 - i. A harmadik szint kisbetűs római számokkal van számozva ponttal.
 - A. A negyedik szinten nagybetűket használnak.
 - B. A címke stílusa a következő részben leírtak szerint módosítható.
 - ii. Vissza a harmadik szintre.
 - (b) Vissza a második szintre.

2. És ismét az *enumerate* első szintje.

Példa egy beágyazott listára vegyes típusokkal:

- Az első szinten lévő tételes címke egy golyó.
 1. A számozás arab számokkal történik, mivel ez a felsorolási környezet első szintje.
 - Ez a beágyazás harmadik szintje, de a második tételes szint.

(a) És ez a teljes beágyazás negyedik szintje, de csak a második szintje a felsorolt környezetnek.

(b) Így a számozás kisbetűkkel történik zárójelben.

– A címke ezen a szinten egy hosszú kötőjel.

2. Minden listának legalább két pontot kell tartalmaznia.

• Az `\item` parancs előtti üres soroknak nincs hatása. A fenti vegyes lista a következő szöveggel készült:

```
\begin{itemize}
\item Az \texttt{itemize} első szinten lévő tételes...
\begin{enumerate}
\item A számozás arab számokkal történik, mivel ez...
\begin{itemize}
\item Ez a beágyazás harmadik szintje, de a ...
\begin{enumerate}
\item És ez a teljes beágyazás negyedik szintje, de ...
\item Így a számozás kisbetűkkel történik zárójelben.
\end{enumerate}
\end{itemize}
\item A címke ezen a szinten egy hosszú kötőjel.
\end{itemize}
\item Minden listának legalább két ...
\end{enumerate}
\item Az \verb+\item+ parancs előtt üres soroknak ...
\end{itemize}
```

Feladat:

Hozd létre a következő beágyazott listát!

LaTeX

- Gyümölcsök
 - Alma
 - Banán
 - Narancs
 - * Spanyol narancs
 - * Portugál narancs
 - Körte
- Zöldségek
 - Sárgarépa
 - Brokkoli
 - Paradicsom
 - * Cseresznye paradicsom
 - * Húsos paradicsom
 - Paprika

Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\begin{itemize}
\item Gyümölcsök
\begin{itemize}
\item Alma
\item Banán
\item Narancs
\begin{itemize}
\item Spanyol narancs
\item Portugál narancs
\end{itemize}
\item Körte
\end{itemize}
\item Zöldségek
\begin{itemize}
\item Sárgarépa
```

```

\item Brokkoli
\item Paradicsom
\begin{itemize}
\item Cseresznye paradicsom
\item Húsos paradicsom
\end{itemize}
\item Paprika
\end{itemize}
\end{document}

```

4.4 Dobozok

A doboz egy szövegrész, amelyet a TEX egységként kezel, mint egyetlen karaktert. Egy doboz (a benne lévő szöveggel együtt) mozgatható balra, jobbra, felfelé vagy lefelé. Mivel a doboz egy egység, a TEX nem tudja újra felbontani, még akkor sem, ha eredetileg kisebb, különálló dobozokból állt. A kisebb dobozokat azonban tetszés szerint összeállíthatjuk a teljes doboz elkészítésekor.

A LATEX három doboztípus közül választhat: LR dobozok, bekezdésdobo­zok és szabálydobo­zok. Az LR (bal-jobb) doboz olyan anyagokat tartalmaz, amelyek vízszintesen, balról jobbra vannak rendezve, egyetlen sorban. A bekezdésdobo­z tartalma függőlegesen halmozott sorokba kerül. A szabálydobo­z egy feketével tömören kitöltött téglalap, általában vízszintes és függőleges vonalak rajzolására.

4.4.1 LR dobozok

Egysoros szöveget tartalmazó LR mezők létrehozásához a következő parancsokat használjuk:

`\mbox{text}` és `\makebox[width][pos]{text}`

`\fbox{text}` és `\framebox[width][pos]{text}`

A fenti két parancs egy LR-dobo­zt hoz létre, amelynek szélessége pontosan megegyezik a kapcsos zárójelek között megadott { } szöveg szélességével. Az `\fbox` parancs ugyanaz, mint az `\mbox`, azzal a különbséggel, hogy a *text* is keretben van.

A jobb oldalon található két parancs­sal a szélességet az opcionális hossz argumentum *width* határozza meg. A másik opcionális *pos* argumentum azt határozza meg, hogy a szöveg hogyan helyezkedjen el a dobozon belül. Érték megadása nélkül a szöveg középre kerül. A *pos* lehetséges értékei:

- l balra igazítani a szöveget,
- r jobbra igazítani a szöveget,
- s nyújtani, hogy a teljes szélességben kitöltse.

Így a `\makebox[4cm]{középre igazítva}` egy 4 cm szélességű mezőt hoz létre, amelyben a szöveg középre kerül, `\framebox[4cm][r]{jobbra igazítva}` szöveg `\framebox[4cm][l]{balra igazítva}` egy 4 cm széles keretes dobozon belül.

Lehetőség van egy LR doboz szélességének megadására is a természetes méretekhez viszonyítva (azok, amelyeket az egyszerű `\mbox` paranccsal állítanak elő):

`\width` a doboz természetes szélessége,
`\height` az alapvonaltól a csúcsig terjedő távolság,
`\depth` az alapvonal és a mélység közötti távolság,
A `\totalheight` értéke `\height` plusz `\depth`.

A következő doboz olyan keretezett dobozt hoz létre, amelynek szélessége a teljes magasság hatszorosa, és középen szöveget tartalmaz:

`\framebox[6\totalheight]{Szöveg}` Szöveg

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\framebox[4cm][r]{Hello, World!} \quad \framebox[4cm][l]{Hello, World!} \quad
\framebox[4cm][c]{Hello, World!}

\vspace{1cm}
\makebox[4cm][r]{Hello, World!} \quad \makebox[4cm][l]{Hello, World!} \quad
\makebox[4cm][c]{Hello, World!}

\end{document}
```

4.4.2 Meghatározott magasságú bekezdésdobozok

A `\parbox` parancs és a `minipage` környezet teljes szintaxisa még további két opcionális argumentumot tartalmaz:

```
\parbox[pos][height][inner pos]{width}{text}
\begin{minipage}[pos][height][inner pos]{width}
szöveg
\end{minipage}
```

Mindkét esetben a *height* a doboz magasságát meghatározó hosszúság: a `\height`, `\width`, `\depth` és `\totalheight` paraméterek ugyanúgy használhatók a magasság argumentumban, mint a *width* a `\makebox` és `\framebox` argumentumában.

Az opcionális *inner pos* argumentum azt mondja ki, hogyan kell a szöveget belül elhelyezni, aminek csak akkor van értelme, ha magasságot adtunk meg. Lehetséges értékei a következők:

- t a szöveget a doboz tetejére tolja,
- b a szöveget a doboz aljára tolja,
- c függőlegesen középre állítsa,
- s nyújtani, hogy kitöltse az egész dobozt.

Példa:

```
\begin{minipage}[t][2cm][t]{3cm}
```

Ez egy 2 cm magas *minipage* szöveggel a tetején.

```
\end{minipage}\hrulefill
```

```
\parbox[t][2cm][c]{3cm}{Ebben a parboxban a szöveg ugyanabban a magasságban van középen.}\hrulefill
```

```
\begin{minipage}[t][2cm][b]{3cm}
```

Ebben a harmadik bekezdésdobozban a szöveg alul található.

```
\end{minipage}
```

A dobozok közötti `\hrulefill` parancsok megmutatják, hol vannak az alapvonalak. Mindhárom doboz azonos méretű, és csak a belső pozíció értékében tér el egymástól.

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Az alábbiakban dobozt láthatunk, amely tartalmaz egy egyszerű matematika feladatot:

Feladat: Adj meg egy példát egy páros és egy páratlan szám összegére!
Megoldás: Például, $2 + 3 = 5$ egy páros és egy páratlan szám összege.

Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\vspace{1cm}
\begin{center}
Az alábbiakban dobozt láthatunk, amely tartalmaz egy egyszerű matematika feladatot:
\\
\vspace{1.5cm}
\fbbox{
\begin{minipage}{0.8\textwidth}
\textbf{Feladat:} Adj meg egy példát egy páros és egy páratlan szám összegére! \\
\textbf{Megoldás:} Például,  $2 + 3 = 5$  egy páros és egy páratlan szám összege.
\end{minipage}
}
\end{center}

\end{document}
```

4.4.3 Szabálydobozok

A szabálydoboz alapvetően egy kitöltött fekete téglalap. Az általános parancs szintaxisa a következő:

```
\rule[lift]{width}{height}
```

amely egy tömör téglalapot hoz létre, amelynek *width* szélessége és *height* magassága van, és az alapvonal fölé emelkedik. Így a `\rule{8mm}{3mm}` egy fekete téglalapot generál. Az opcionális argumentumemelés nélkül a téglalap az aktuális szövegsor alapvonalára kerül.

Ha a *lift* értéke negatív, a téglalap az alapvonal alá van állítva.

4.4.4 Beágyazott dobozok

A fent leírt dobozparancsok bármilyen kívánt szintre beágyazhatók. Az LR doboz beépítése egy *parbox*ba vagy egy *minipage*-be nem okoz nyilvánvaló fogalmi nehézségeket. Az ellenkezője, egy LR dobozon belüli *parbox* is lehetséges, és könnyen elképzelhető, ha szem előtt tartjuk, hogy minden doboz egy egység, amelyet a LATEX egyetlen, megfelelő méretű karakterként kezel.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\fbbox{
\begin{minipage}{\textwidth}
\begin{minipage}{0.4\linewidth}
\fbbox{\parbox{\linewidth}{Ez egy bal oldali doboz.}}
\end{minipage}
\hfill
\begin{minipage}{0.4\linewidth}
\fbbox{\parbox{\linewidth}{Ez egy jobb oldali doboz.}}
\end{minipage}
\end{minipage}
}

\end{document}
```

4.5 Táblázatok

Az előző szakaszok dobozelemeivel és tabulátoros környezetével mindenféle keretezett és keret nélküli táblázatok készíthetők. A LATEX azonban sokkal kényelmesebb módokat kínál a felhasználónak ilyen bonyolult szerkezetek felépítésére.

4.5.1 Táblázatok készítése

A *tabular*, *tabular** és *array* környezetek azok az alapvető eszközök, amelyekkel táblákat és mátrixokat lehet létrehozni. Ezeknek a környezeteknek a szintaxisa a következő:

```
\begin{array}[pos]{cols} sor \end{array}
```

```
\begin{tabular}[poz]{cols} sor \end{tabular}
```

```
\begin{tabular*}{width}[pos]{cols} sor \end{tabular*}
```

A tömbkörnyezet csak matematikai módban alkalmazható. A szintaxisa és argumentumainak jelentése pontosan megegyezik a táblázatos környezetével. Mindhárom környezet valójában egy *minipage*-t hoz létre. A paraméterek jelentése a következő:

pos Függőleges pozicionálási argumentum. A következő értékeket veheti fel:

t a táblázat felső sora az aktuális külső szövegsor alapvonalához igazodik;

b a táblázat alsó sora a külső alapvonalhoz igazodik; pozicionálási argumentum megadása nélkül a táblázat a külső alapvonalra igazodik.

width Ez az argumentum csak a *tabular** környezetre vonatkozik, és meghatározza annak teljes szélességét. Ebben az esetben a *cols* argumentumnak valahol az első bejegyzés után tartalmaznia kell a @-kifejezést a `@{\extracolsep{\fill}}` parancs esetében. A másik két környezet esetében a teljes szélességet a szöveges tartalom határozza meg.

cols Az oszlop formázási argumentuma. Minden oszlophoz tartalmaznia kell egy bejegyzést, valamint a táblázat bal és jobb oldali szegélyéhez, illetve az oszlopközökhöz lehetséges extra bejegyzéseket.

A lehetséges oszlopformázási szimbólumok a következők:

l az oszlop tartalma balra igazított;

r az oszlop tartalma jobbra van igazítva;

c az oszlop tartalma középre van állítva;

p{wth} ebben az oszlopban a szöveg *wth* szélességű sorokra van beállítva, a felső sor pedig a többi oszlophoz igazodik. Valójában a szöveget parboxban állítjuk be a `\parbox[t]{wth}{oszlopszöveg}` paranccsal;

{num}{cols} a oszlopokban található oszlopformátum reprodukálódik *num*-szor, így `{5}{|c|}` megegyezik a `|c|c|c|c|c|`-vel.

A rendelkezésre álló formázási szimbólumok a bal és jobb oldali szegélyekhez, valamint az oszlopok közötti távolsághoz:

| függőleges vonalat húz;

|| két függőleges vonalat húz egymás mellé;

`@{text}` ezt a bejegyzést @-kifejezésnek nevezzük, és szöveget szúr be a táblázat minden sorába a két oszlop közé, ahol megjelenik.

A @-kifejezés eltávolítja az oszlopok közötti távolságot, amely automatikusan az egyes oszloppárok közé kerül. Ha szóközre van szükség a beszúrt szöveg és a következő oszlop között, akkor ezt kifejezetten szerepeltetni kell a `\hspace{ }` karakterlánccal a @-kifejezés szövegében. Ha az oszlopok közötti távolság két adott oszlop között a szabványtól eltérő, akkor ezt könnyen elérhetjük úgy, hogy a formázási argumentumban a megfelelő oszlopok közé `@{\hspace{wth}}`-t helyezünk. Ez felváltja a szabványos oszlopközi távolságot a *wth* szélességre.

Egy @-kifejezésen belüli `\extracolsep{wth}` extra *wth* szóközt helyez el az összes következő oszlop között, amíg egy másik `\extracolsep` parancs nem ellenzi. A szabványos térközzel ellentétben ezt a további térközt a későbbi @-kifejezések nem távolítják el. A *tabular** környezetben valahol az oszlopformátumban kell lennie egy `@{\extracolsep\fill}` parancsnak, hogy az összes következő oszlopközi távolság megnyúlhasson, hogy kitöltse az előre meghatározott táblázatszélességet.

Ha a táblázat bal vagy jobb oldali szegélye nem függőleges vonalból áll, akkor a rendszer a normál oszlopközi távolság felével egyenlő távolságot ad hozzá. Ha ezt a szóközt nem kívánja, akkor letilthatja egy üres @-kifejezés `@{ }` beillesztésével az oszlopformátum elejére vagy végére.

`rows` tartalmazzák a táblázat tényleges bejegyzéseit, és minden vízszintes sor egy `\\` paranccsal zárul. Ezek a sorok oszlopbejegyzések sorozatából állnak, amelyeket & szimbólum választ el egymástól. Így a táblázat minden sora ugyanannyi oszlopbejegyzést tartalmaz, mint az oszlopdefiníciós oszlopokban. Egyes bejegyzések üresek lehetnek. Az egyes oszlopbejegyzéseket a LATEX úgy kezeli, mintha kapcsos zárójelek közé `{ }` lennének, így a típusstílusban vagy méretben bekövetkező bármilyen változás erre az egyetlen oszlopra korlátozódik.

`\hline` Ez a parancs csak az első sor előtt vagy közvetlenül a `\\` sor lezárása után jelenhet meg. Vízszintes vonalat húz a táblázat teljes szélességében az éppen befejezett sor alá, vagy a táblázat tetejére, ha az elején van.

Két `\hline` parancs együttesen két vízszintes vonalat rajzol, köztük egy kis térközzel.

`\cline{m n}` Ez a parancs vízszintes vonalat húz az m oszlop bal oldalától az n oszlop jobb oldaláig. Hasonlóan, mint `\hline` esetében, csak közvetlenül a `\\` sorlezárás után adható meg, és több is lehet egymás után. A `\cline{1-3}` és `\cline{5-7}` parancs két vízszintes vonalat húz az 1. oszlopból a 3. oszlopba és az 5. oszlopból a 7. oszlopba, az éppen befejezett sor alá. A teljes oszlopszélesség minden esetben alá van húzva.

`\multicolumn{num}{col}{text}` Ez a parancs a következő num számú oszlopot egyetlen oszlopba egyesíti teljes szélességükkel, beleértve az oszlopok közötti távolságokat is. A col argumentum pontosan tartalmazza az l , r vagy c pozicionáló szimbólumok egyikét, lehetséges $@$ -kifejezésekkel és függőleges $|$ vonalakkal. A num értéke 1 lehet, ha a pozicionálási argumentumot egy adott sorban az adott oszlopra vonatkozóan kell megváltoztatni.

Ebben az összefüggésben egy col egy l , r vagy c pozicionáló szimbólummal kezdődik, és mindent magában foglal a következő oszlopig, de a következő kivételével. Az első oszlop az első pozicionáló szimbólum előtt mindent tartalmaz. Így a `|c@{}rl|` három oszlopot tartalmaz: az első a `|c@{}|`, a második az r , a harmadik pedig az l . A `\multicolumn` parancs csak egy sor elején vagy közvetlenül az $\&$ oszlopelválasztó szimbólum után jöhet.

`\vline` Ez a parancs függőleges vonalat húz a sor magasságával azon a helyen, ahol megjelenik. Ily módon olyan függőleges vonalak is beszúrhatók egy oszlopba, amelyek nem nyúlnak le a táblázat teljes magasságában.

Mivel a táblázat egy függőleges doboz, amely ugyanolyan típusú, mint a *parbox* és a *minipage*, vízszintesen elhelyezhető más dobozokkal vagy szöveggel. A következő paranccsal a táblázat az oldal közepére helyezhető:

```
\begin{center} táblázat \end{center}
```

Feladat:

Hozd létre a következő táblázatokat!

L^AT_EX

Egyszerű táblázat:

Sorszám	Név	Életkor
1	Anna	25
2	Béla	33
3	Csaba	42

Bonyolultabb táblázat:

Táblázat címe		
Sorszám	Név	Pontszám
1	Anna	25
2	Béla	33
<i>Átlag: 29</i>		
Összpontszám:		58

Megoldás:

```

\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

Egyszerű táblázat:
\vspace{12pt}

\begin{tabular}{|c|c|c|}
\hline
\textbf{Sorszám} & \textbf{Név} & \textbf{Életkor} \\
\hline
1 & Anna & 25 \\
\hline
2 & Béla & 33 \\
\hline
3 & Csaba & 42 \\
\hline
\end{tabular}

\vspace{12pt}
Bonyolultabb táblázat:

```

```

\vspace{12pt}

\begin{tabular}{|c||r|}
\hline
\multicolumn{3}{|c|}{\textbf{Táblázat címe}} \\
\hline
\textbf{Sorszám} & \textbf{Név} & \textbf{Pontszám} \\
\hline
1 & Anna & 25 \\
\hline
2 & Béla & 33 \\
\hline
\multicolumn{3}{|c|}{\textit{Átlag: 29}} \\
\cline{1-2}
\multicolumn{2}{|r|}{\textbf{Összpontszám:}} & 58 \\
\hline
\end{tabular}

\end{document}

```

4.5.2 Táblázatstílus-paraméterek

Számos stílusparamétert használnak a táblázatok generálásához, amelyeket a LATEX szabványos értékekre állít be. Ezeket a felhasználó módosíthatja, akár globálisan a preambulumban, akár lokálisan egy környezetben. Ezeket nem szabad a táblázat környezetében megváltoztatni.

A `\tabcolsep` a `tabular` és a `tabular*` környezetben az oszlopok közé beszúrt térköz fele;

az `\arraycolsep` a megfelelő fél oszlopközi távolság a tömbkörnyezetben;

`\arrayrulewidth` a függőleges és vízszintes vonalak vastagsága a táblázaton belül;

A `\doublerulesep` a kettős szabály sorainak elválasztása.

Ezek a paraméterek a szokásos módon módosíthatók a `\setlength` paranccsal. Például, hogy a vonal vastagsága 0,5 mm legyen, adjuk meg a `\setlength` értéket az alábbi módon:

```
{\arrayrulewidth}{0,5mm}.
```

Az `\arraystretch` paraméter segítségével módosítható a táblázat sorai közötti távolság. Ez egy szorzóérték, alapértéke 1. Az 1,5 érték azt jelenti, hogy a sorköz 50%-kal nő. A következő paranccsal történő újradefiniálásával új érték adható meg:

```
\renewcommand{\arraystretch}{faktor}
```

4.6 Szó szerinti szöveg nyomtatása (*verbatim*)

Időnként pontosan úgy kell kinyomtatni a szöveget, ahogyan begévelték, és minden speciális karakter, üres hely és sortörés szó szerint, formázatlanul és írógépes betűtípussal jelenik meg.

A számítógépes kód sorai vagy a LATEX bemeneti szöveg mintái példák az ilyen szó szerinti szövegekre. Ez a környezettel valósul meg

```
\begin{verbatim} text \end{verbatim}
```

```
\begin{verbatim*} text \end{verbatim*}
```

Egy új sor kerül beszúrásra ezen környezetek elé és utána is.

A *-os alak az üres helyekre a „szököz” szimbólum látható, hogy láthatóvá váljanak.

Feladat:

Hozd létre a következő szövegrészt!

```

                                LATEX

\begin{tabular}{|c|l|r|}
\hline
\multicolumn{3}{|c|}{\textbf{Táblázat címe}} \\
\hline
\textbf{Sorszám} & \textbf{Név} & \textbf{Pontszám} \\
\hline
1 & Anna & 25 \\
\hline
2 & Béla & 33 \\
\hline
\multicolumn{3}{|c|}{\textit{Átlag: 29}} \\
\cline{1-2}
\multicolumn{2}{|r|}{\textbf{Összpontszám:}} & 98 \\
\hline
\end{tabular}

Szöközök megjelenítésével:

\begin{tabular}{|c|l|r|}
\hline
\multicolumn{3}{|c|}{\textbf{Táblázat címe}} \\
\hline
\textbf{Sorszám} & \textbf{Név} & \textbf{Pontszám} \\
\hline
1 & Anna & 25 \\
\hline
2 & Béla & 33 \\
\hline
\multicolumn{3}{|c|}{\textit{Átlag: 29}} \\
\cline{1-2}
\multicolumn{2}{|r|}{\textbf{Összpontszám:}} & 98 \\
\hline
\end{tabular}
```

Megoldás:

```
\documentclass[12pt,a4paper]{article}
```

```
\begin{document}
```

```
\begin{center}
```



```

\LaTeX\
\end{center}

\begin{verbatim}
\begin{tabular}{|c||r|}
\hline
\multicolumn{3}{|c|}{\textbf{Táblázat címe}} \\
\hline
\textbf{Sorszám} & \textbf{Név} & \textbf{Pontszám} \\
\hline
1 & Anna & 25 \\
\hline
2 & Béla & 33 \\
\hline
\multicolumn{3}{|c|}{\textit{Átlag: 29}} \\
\cline{1-2}
\multicolumn{2}{|r|}{\textbf{Összpontszám:}} & 98 \\
\hline
\end{tabular}
\end{verbatim}

```

Szöközők megjelenítésével:

```

\begin{verbatim*}
\begin{tabular}{|c||r|}
\hline
\multicolumn{3}{|c|}{\textbf{Táblázat címe}} \\
\hline
\textbf{Sorszám} & \textbf{Név} & \textbf{Pontszám} \\
\hline
1 & Anna & 25 \\
\hline
2 & Béla & 33 \\
\hline
\multicolumn{3}{|c|}{\textit{Átlag: 29}} \\
\cline{1-2}
\multicolumn{2}{|r|}{\textbf{Összpontszám:}} & 98 \\
\hline
\end{tabular}
\end{verbatim*}

\end{document}

```

4.6.1 E-mail és internet címek

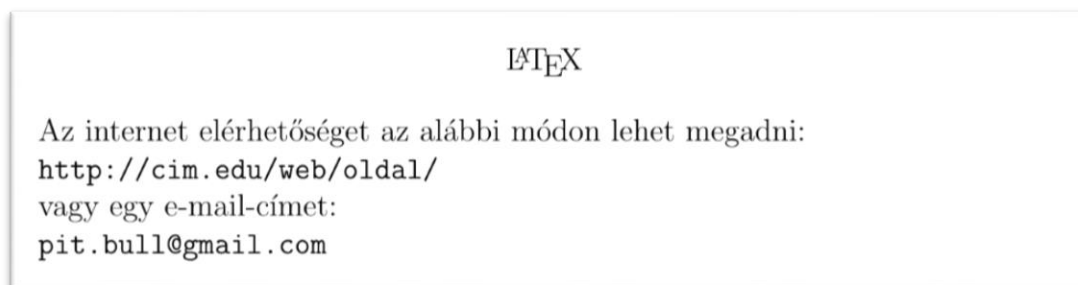
Az e-mail és internetcímek speciális problémát jelentenek, amelyeket az *url* csomaggal oldunk meg. Ezeket a címeket legjobban írógépes betűtípussal lehet felsorolni, gyakran tartalmaznak speciális szimbólumokat, és soha nem szabad kötőjelet elhelyezni, mivel a kötőjel a cím

részeként értelmezhető. A kézenfekvő megoldás az lenne, ha a `\verb` parancsot használnánk, amely teljesíti ezeket a feltételeket. Ha azonban a cím nem fér el az aktuális sorban, akkor a jobb margóra fog kilógni. Ehelyett a `\texttt` használata lehetővé teszi, hogy a címben szereplő szavak két sorra törjenek, de kétértelmű kötőjel beszúrása árán.

A cím beszúrása az `\url` paranccsal lehetővé teszi, hogy a szavak között a nem betűknél kötőjel nélkül válassza el. Sőt, argumentumát szó szerint kezelik, így minden speciális karakter adottnak van kinyomtatva, ugyanúgy, mint a `\verb`. Valójában az `\url` parancs argumentuma a szokásos módon kapcsos zárójelek között lehet, vagy tetszőleges karakterrel elválasztható, akár csak a `\verb` esetében.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt,a4paper]{article}
\usepackage{url}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

\noindent
Az internet elérhetőséget az alábbi módon lehet megadni:\\
\url{http://cim.edu/web/oldal/} \\
vagy egy e-mail-címet:\\
\url=pit.bull@gmail.com=

\end{document}
```

4.7 Lábjegyzetek és széljegyzetek

4.7.1 Szabványos lábjegyzetek

A lábjegyzetek az alábbi paranccsal generálódnak:

```
\footnote{lábjegyzet szövege}
```

amely lábjegyzetben közvetlenül a magyarázatot igénylő szó után következik. A szöveges lábjegyzet szövege lábjegyzetként, kisebb betűtípussal jelenik meg az oldal alján. A lábjegyzet első sora behúzott, és ugyanazt a lábjegyzetjelzőt kapja, mint a törzsszövegben. Az oldal első lábjegyzetét egy rövid vízszintes vonal választja el az oldal szövegének többi részétől.

A szabványos lábjegyzetjelző egy kicsi, emelt szám¹, amely sorban van számozva. Ez a lábjegyzet a következőkkel készült:

... emelt szám\ footnote {A szokásos módszer a lábjegyzetek megjelölésére egy géppel írt ... ugyanazon az oldalon.}, ami ...

A lábjegyzetek számozása az *article* dokumentumosztály teljes dokumentumában növekszik, míg a *report* és *book* osztályok minden új fejezeténél 1-re áll vissza.

A \footnote parancs csak normál bekezdésmódban adható meg, matematikai vagy LR módban nem.

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Mi a számítógép? A legfigyelemreméltóbb eszköz, amit valaha feltaláltunk: olyan, mint egy bicikli az agyunk számára.¹

¹Steve Jobs idézete

1

Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}
\vspace*{16cm}
\begin{center}
\LaTeX\
\end{center}

Mi a számítógép? A legfigyelemreméltóbb eszköz, amit valaha feltaláltunk: olyan, mint
egy bicikli az agyunk számára.\footnote{Steve Jobs idézete}

\end{document}
```

4.7.2 Széljegyzetek

Az oldalmargóban lévő megjegyzések a következő paranccsal jönnek létre:

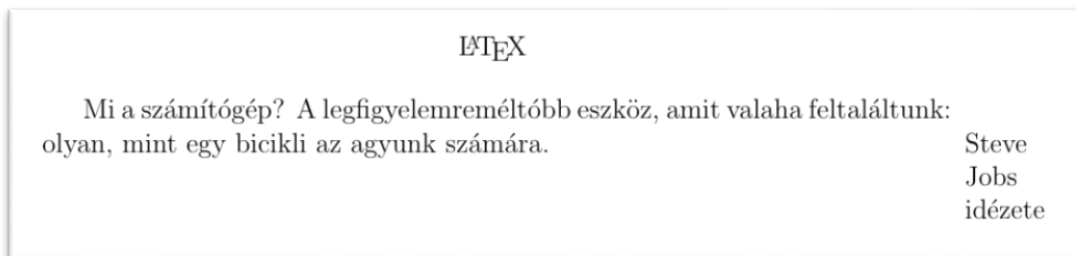
```
\marginpar{jegyzet_szövege}
```

amely a szöveges *jegyzet_szövegét* a margóra helyezi az aktuális sor szintjétől kezdve, ahol a parancsot adják.

Ha a jegyzet egy oldal alján jelenik meg, akkor lefelé nyúlik a normál szöveg utolsó sora alá. Emiatt, valamint a keskeny oszlopok sortörésének nehézségei miatt a széljegyzeteket röviden kell írni, néhány szóra vagy szimbólumra korlátozva.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\
\end{center}

Mi a számítógép? A legfigyelemreméltóbb eszköz, amit valaha feltaláltunk: olyan, mint
egy bicikli az agyunk számára.\marginpar{Steve Jobs idézete}

\end{document}
```

4.8 Megjegyzések a szövegben

Valamennyi számítógépes programozási nyelv lehetőséget biztosít megjegyzések beillesztésére a kódba. Ezek magyarázó megjegyzések, dokumentációk, fejlesztéstörténetek vagy alternatív szövegek vagy kódok, amelyeket ideiglenesen deaktiváltak. A megjegyzéssorokat a rendszer teljesen figyelmen kívül hagyja a feldolgozás során. Csak a forrásszövegbe betekintő olvasók számára készültek.

A LATEX-ben a megjegyzés karakter a százalékjel %. Amikor ez a karakter megjelenik a szövegben, a rendszer figyelmen kívül hagyja azt és a sor többi részét. Ha egy megjegyzés több soros, minden sor előtt % előtagot kell feltüntetni.

Más egykarakteres parancsokhoz hasonlóan magát a százalékjelet a \% paranccsal nyomtatja ki.

A % megjegyzés karakter akkor is hasznos, ha szövegekkel vagy felhasználói parancsok vagy formázási paraméterek definícióival kísérletezünk, hogy alternatívákat próbálunk ki a régi verziók törlése nélkül.

Végül a % karakternek fontos szerepe van a sorvégi szóközök elnyomásában. Ez különösen kívánatos olyan felhasználói definíciókban, ahol váratlan üres helyek kúszhatnak be az egyébként láthatatlan argumentumokkal ellátott deklarációk közé.

5 Matematikai képletek

Donald Knuth azért találta ki a szövegformázási rendszerét, hogy a matematikai képletek beállítását és bonyolult a nyomtatását megkönnyítse.

Ebben a fejezetben a matematikai szövegszerkesztésnek a szabványos LATEX-ben elérhető elemeire szorítkozunk. A matematikai képletek előállítására speciális leíró jellegű szöveg beírásával történik. Ez azt jelenti, hogy a LATEX-et tájékoztatni kell arról, hogy a következő szöveget matematikai képletként kell értelmezni, és azt is közölni kell vele, hogy mikor ér véget a matematikai szöveg, és mikor kezdődik újra a normál szöveg. A matematikai szöveg feldolgozása a matematikai üzemmódba való átváltással történik. Ezt a célt szolgálják a matematikai környezetek.

5.1 Matematikai *környezetek*

Matematikai képletek előfordulhatnak egy szöveg sorában, mint például $c^2 = a^2 + b^2$, vagy a főszövegtől elválasztva, mint:

$$(x + a)^n = \sum_{k=0}^n \binom{n}{k} x^k a^{n-k}$$

Ezt a két típust úgy különböztetjük meg, hogy szöveges, illetve megjelenített képletként hivatkozunk rájuk.

A szöveges képleteket, vagy egyenleteket a környezet segítségével generáljuk.

`\begin{math} szöveges képlet\end{math}`

Mivel a *szöveges képletek* gyakran nagyon rövidek, néha csak egyetlen karakterből állnak, ezért egy rövidített változata is rendelkezésünkre áll `\(szöveges képlet\)` alakban.

A legtöbb felhasználó azonban a nagyon rövid $\$szöveges\ képlet\$$ formulát részesíti előnyben, ami valójában a TEX módszere.

A formula tartalma, a *szöveges képlet*, amely matematikai konstrukciókból áll, amelyeket a következőkben ismertetünk.

A megjelenített képleteket, vagyis egyenleteket a következő környezetekben állítjuk elő:

```
\begin{displaymath} képlet \end{displaymath}
```

```
\begin{equation} képlet \end{equation}
```

A különbség a kettő között az, hogy az egyenletkörnyezet (*equation*) automatikusan hozzáad egy sorszámot az egyenlethez. A *displaymath* környezetet a `\[. . . . \]` vagy a `$$ $$` rövidítéssel is megadhatjuk.

Alapértelmezés szerint a megjelenített képletek vízszintesen középre vannak igazítva, az egyenletszám pedig, ha van, a jobb oldali margóval egy szintbe lesz igazítva. A dokumentumosztály *fleqn* opciójának kiválasztásával a képletek balra igazított, állítható behúzással kerülnek beállításra. Ez az opció az egész dokumentumra érvényes marad, míg a behúzás mértéke tetszőlegesen módosítható a `\setlength{\mathindent}{indent}` paranccsal, ahol a behúzás egy hosszmeghatározás. Továbbá a dokumentumosztály *leqno* opciója az egész dokumentumban a bal margóval egy síkba állítja az egyenletszámokat.

Többsoros képletek hozhatók létre a következő környezetekkel:

```
\begin{eqnarray} képlet \end{eqnarray}
```

```
\begin{eqnarray*} képlet \end{eqnarray*}
```

ahol a standard forma minden sorhoz egy sorszámot ad hozzá, a *-os forma pedig egyenletszámok nélkül lesz.

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

Szöveges képletek:

A Pitagorasz-tétel azt mondja ki, hogy a derékszögű háromszög oldalai között van egy fontos összefüggés: a leghosszabb oldalának a négyzete egyenlő a másik két oldal négyzetének összegével.

Azaz a négyzet meg b négyzet egyenlő c négyzettel.

A megjelenített képletek:

$$a^2 + b^2 = c^2$$

$$a^2 + b^2 = c^2$$

Az egyenletkörnyezet (equation) automatikusan hozzáad egy sorszámot az egyenlethez:

$$a^2 + b^2 = c^2 \tag{1}$$

Megoldás:

```
\documentclass[12pt,a4paper]{article}
```

```
\begin{document}
```

```
\begin{center}
```

```
\LaTeX\
```

```
\end{center}
```

```
Szöveges képletek:\\
```

```
A Pitagorasz-tétel azt mondja ki, hogy a derékszögű háromszög oldalai között van egy fontos összefüggés: a leghosszabb oldalának a négyzete egyenlő a másik két oldal négyzetének összegével.\\
```

```
Azaz  $a$  négyzet meg  $b$  négyzet egyenlő  $c$  négyzettel. \\
```

```
A megjelenített képletek:
```

```
 $a^2 + b^2 = c^2$ 
```

```
\begin{displaymath} a^2 + b^2 = c^2 \end{displaymath}
```

```
Az egyenletkörnyezet (equation) automatikusan hozzáad egy sorszámot az egyenlethez:
```

```
\begin{equation} a^2 + b^2 = c^2 \end{equation}
```

```
\end{document}
```

5.2 A matematikai mód fő elemei

5.2.1 Állandók és változók

A képletekben megjelenő számokat konstansoknak nevezzük, míg az egyszerű változókat egyszerű betűkkel ábrázoljuk. A matematikai szedés általános gyakorlata, hogy a konstansokat *Roman* betűtípussal, a változókat pedig dőlt betűvel írjuk. Matematikai módban a LATEX ezt a szabályt automatikusan betartja. Az üres betűket (szóközöket) teljesen figyelmen kívül hagyjuk, és csak azért szerepelnek a beviteli szövegben, hogy az felhasználó számára a megjelenést javítsuk. A konstansok, változók és az olyan operátorok, mint a +, -, = közötti távolságot a LATEX automatikusan beállítja. Például

$c=3a+5b$, $c = 3 a + 5 b$ mindkettő $c=3a+5b$ -t eredményez.

A billentyűzeten elérhető matematikai szimbólumok a következők:

+ - = < > / : ! ' | [] ()

amelyek mindegyike közvetlenül használható képletekben. A kapcsos zárójelek { } a képlet részeinek logikai összekapcsolását szolgálják, ezért nem működhetnek nyomtatható karakterekként. A zárójelek képletbe való beillesztéséhez ugyanazokat a \{ és \} parancsokat kell használni, mint a normál szövegben.

5.2.2 Hatványkitevők és indexek

A matematikai képletek gyakran tartalmaznak hatványkitevőket és indexeket, olyan karaktereket, amelyek a képlet fő sorához képest megemelt vagy lesüllyesztettek, és kisebb betűtípussal vannak nyomtatva. Bár matematikai jelentésük eltérő, a felső és az alsó indexek tipográfiailag ugyanazok, mint az hatványkitevők és az indexek. A LATEX és a TEX lehetővé teszi a hatványkitevők bármilyen kombinációjának létrehozását és indexek megfelelő típusméretű kombinációjának egyszerű módon történő létrehozását: a ^ karakter parancs a következő karaktert hatványkitevőként (emelt), míg a _ karakter parancs indexként (csökkentett) állítja be.

x^3 x^3 a_n a_n x_n^3 x^3_n

Ha a hatványkitevők és az indexek együtt fordulnak elő, a sorrendjük nem fontos.

Ha a hatványkitevő vagy az index egynél több karaktert tartalmaz, a karaktercsoportot zárójelbe { } kell zárni:

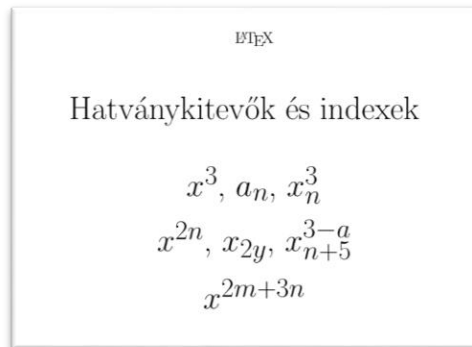
x^{2n} $x^{\{2n\}}$ x_{2y} $x_{\{2y\}}$ x_{n+5}^3 $x^{\{3-a\}}_{\{n+5\}}$

A többszörös emelések és csökkentések (hatványozások és indexelések) úgy jönnek létre, hogy az hatványkitevőkre és indexekre újból ^ és _ karakter parancsokat alkalmazunk.

A ^ és _ karakter parancsok csak matematikai módban engedélyezettek.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\

\vspace{1cm}
\Huge{Hatványkitevők és indexek }
\vspace{1cm}

$x^3$, $a_n$, $x_n^3$

\vspace{12pt}
$x^{2n}$, $x_{2y}$, $x_{n+5}^{3-a}$

\vspace{12pt}
$x^{2m+3n}$
\end{center}

\end{document}
```

5.2.3 Törtek

A rövid törtek, különösen a szöveges képleteken belül, legjobban a / perjellel jeleníthetők meg, mint a $\$(a+b)/2\$$ az $(a+b)/2$ esetén. Bonyolultabb törtek esetén a következő paranccsal adjuk meg:$

```
\frac{számláló}{nevező}
```

A számlálót a nevező tetejére írják úgy, hogy közöttük egy megfelelő szélességű vízszintes törtvonal legyen.

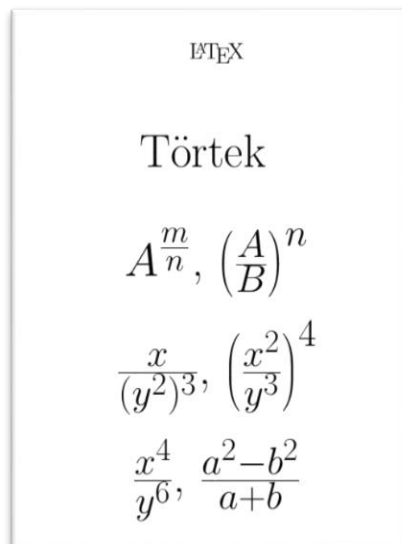
$$\frac{1}{a+b} \quad \backslash [\frac{1}{a+b}] \backslash$$

$$\frac{a^2-b^2}{a+b} \quad \backslash [\frac{a^2 - b^2}{a+b} \backslash]$$

A törtek bármilyen mélységig egymásba ágyazhatók.

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\

\vspace{1cm}
\Huge{Törtek }
\vspace{1cm}

 $A^{\frac{m}{n}}$ ,  $\left(\frac{A}{B}\right)^n$ 

\vspace{12pt}
 $\frac{x}{(y^2)^3}$ ,  $\left(\frac{x^2}{y^3}\right)^4$ 

\vspace{12pt}
 $\frac{x^4}{y^6}$ ,  $\frac{a^2 - b^2}{a+b}$ 
\end{center}

\end{document}
```

5.2.4 Gyökök

A gyököket a következő paranccsal adjuk meg:

$\sqrt[n]{arg}$

Példa: $\sqrt[3]{125} = 5$ a következőt határozza meg: $\sqrt[3]{125} = 5$. Ha az opcionális n argumentumot kihagyjuk, a négyzetgyök jön létre: \sqrt{a} vagyis \sqrt{a} .

A gyökjel mérete és hossza automatikusan illeszkedik az argumentumhoz:

$$\sqrt{a^2 + b^2 + 2ab} = a+b \qquad \sqrt{a^2 + b^2 + 2ab} = a + b$$

$$\sqrt[n]{\frac{a^n - b^n}{1 + a^{2n-1}}} \qquad \sqrt[n]{\frac{a^n - b^n}{1 + a^{2n-1}}}$$

A gyökök bármilyen mélységig egymásba ágyazhatók:

$$\sqrt[3]{-q + \sqrt{q^2 + p^3}} \qquad \sqrt[3]{-q + \sqrt{q^2 + p^3}}$$

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Gyökök

$$\sqrt{2}, \sqrt[3]{8}$$

$$(\sqrt{2})^3, \sqrt{\frac{x^2}{y^3}}$$

$$\sqrt{x^2 + y^2}$$

Megoldás:

```
\documentclass[12pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\

\vspace{1cm}
\Huge{Gyökök }
\vspace{1cm}

 $\sqrt{2}$ ,  $\sqrt[3]{8}$ 

\vspace{12pt}
 $\left(\sqrt{2}\right)^3$ ,  $\sqrt[5]{\frac{x^2}{y^3}}$ 

```

```
\vspace{12pt}
 $\sqrt{x^2+y^2}$ 
\end{center}

\end{document}
```

5.2.5 Összegek és integrálok

Az összegzés és az integrál jelek a `\sum` és `\int` paranccsal készülnek, amelyek két különböző méretben jelenhetnek meg attól függően, hogy szövegben vagy megjelenített képletben fordulnak elő.

Az összegeknek és integráloknak nagyon gyakran van felső és alsó határa. Ezeket a `^` és `_` karakter paranccsokkal adjuk meg. A határértékek elhelyezése attól is függ, hogy a képlet szövegben vagy megjelenített képletben fordulnak elő.

Egy szöveges képletben a `\sum_{i=0}^n` és `\int_a^b` eredményezi. Amennyiben ha az integrál korlátait az integráljel felett és alatt elhelyezve szeretnénk megjeleníteni, akkor írjunk egy `\limits` parancsot közvetlenül az integráljel után. Hasonlóan az összegzés esetében is.

```
\int\limits_{x=0}^{x=1}
```

$$\int_{x=0}^{x=1}$$

Feladat:

Hozd létre a következő szövegrészt!

ℒ_{TEX}

Összegek és integrálok

$$\sum_{i=1}^4 i^3, \sum_{i=1}^5 i^3, \sum_{i=1}^6 \frac{i}{i+2}$$

$$\sum_{i=1}^2 \sum_{j=1}^3 ij, \sum_{i=1}^7 \frac{x^i}{i!}$$

$$\int x^2 dx + C, \int_0^1 x^2 dx$$

$$\int \cos(x) dx, \int \frac{x^2+1}{x+1} dx, \int \frac{x^2+1}{x+1} dx$$

Megoldás:

```
\documentclass[12pt,a4paper]{article}
\usepackage{amsmath}
```

```

\begin{document}

\begin{center}
\LaTeX\

\vspace{1cm}
\Huge{Összegek és integrálok }
\vspace{1cm}

 $\sum_{i=1}^4 i^3$ ,  $\sum_{i=1}^5 i^3$ ,  $\sum_{i=1}^6 \frac{i}{i+2}$ 

\vspace{12pt}
 $\sum_{i=1}^2 \sum_{j=1}^3 ij$ ,  $\sum_{i=1}^7 \frac{x^i}{i!}$ 

\vspace{12pt}
 $\int x^2 dx + C$ ,  $\int_0^1 x^2 dx$ 

\vspace{12pt}
 $\int \cos(x) dx$ ,  $\int \frac{x^2+1}{x+1} dx$ ,  $\int \frac{x^2+1}{x+1} dx$ 
\end{center}

\end{document}

```

5.2.6 Folytatási pontok megadása

A képletek esetenként pontsört tartalmaznak. . . , jelentése „és így tovább”. Egy pont háromszor egymás utáni beírása nemkívánatos eredményt eredményez: ..., vagyis a pontok túl közel vannak egymáshoz. Ezért a LATEX számos parancsot ad a pontok helyes megadásához.

`\ldots` . . . lefelé igazított pontok `\cdots` . . . középre igazított pontok

`\vdots` . függőlegesen igazított pontok `\ddots` . . . átlósan igazított pontok

Az első két parancs közötti különbséget legjobban az a_0, a_1, \dots, a_n és az $a_0 + a_1 + \dots + a_n$ példák szemléltetik, ahol az elsőt $\$a_0, a_1, \dots, a_n\$$ és a másodikat a $\$a_0 + a_1 + \dots + a_n\$$ módon hoztuk létre.

A `\ldots` parancs normál szöveges módban is alkalmazható, míg a másik három csak matematikai módban engedélyezett. Szöveges módban a parancs a `\dots` a `\ldots` helyett használható, ugyanolyan eredménnyel.

5.3 Matematikai szimbólumok

A matematikai szövegben nagyon sokféle szimbólumot használnak, amelyek közül csak néhány érhető el közvetlenül a billentyűzetről. A LATEX számos általánosan használt matematikai szimbólumot biztosít. Megjelenítésük a szimbólumnévvel és a `\` karakterrel történik. Maguk a nevek matematikai jelentésükből származnak.

5.3.1 Görög betűk

Kisbetűk

α	<code>\alpha</code>	θ	<code>\theta</code>	\omicron	<code>\o</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	e	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>				

Nagybetűk

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

A görög betűket egyszerűen úgy állítják elő, hogy a `\` parancskaraktert a betű neve elé helyezik. A nagybetűket a név első betűjének nagybetűivel lehet megkülönböztetni. A fenti listában nem szereplő görög betűk azonosak valamelyik megfelelő latin betűvel. Például a ρ nagybetűje megegyezik a latin P-vel, ezért nincs szükség külön szimbólumra.

A LATEX általában egy matematikai képletben a nagybetűs görög betűket állítja be *Roman* betűtípussal. Ha dőlt betűvel kell szedni, akkor ez a matematikai ábécé `\mathnormal` paranccsal érhető el:

`\mathnormal{\Gamma\Pi\Phi}` $\Gamma \Pi \Phi$ formában jelenik meg.

A görög betűk csak matematikai módban használhatók. Ha normál szövegben szükség van rájuk, akkor a parancsot `\mathnormal` jelek közé kell zárni.

Feladat:

Hozd létre a következő szövegrészt!

TeX

Görög betűk használata

$$\alpha + \beta = \gamma$$

$$\Delta x = x_2 - x_1$$

$$\epsilon = \frac{F}{A}$$

$$\kappa = \frac{\Delta x}{\Delta y}$$

$$\lambda = \frac{c}{f}$$

$$\mu = \rho \cdot v$$

$$\xi = \sum_{i=1}^n x_i$$

$$\Phi = \int_{-\infty}^{\infty} \phi(x) dx$$

$$\omega = 2\pi f$$

Megoldás:

```
\documentclass[12pt,a4paper]{article}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{1cm}
\Huge{Görög betűk használata }
\vspace{1cm}

\begin{align*}
\alpha + \beta &= \gamma \\
\Delta x &= x_2 - x_1 \\
\epsilon &= \frac{F}{A} \\
\kappa &= \frac{\Delta x}{\Delta y} \\
\lambda &= \frac{c}{f} \\
\mu &= \rho \cdot v \\
\xi &= \sum_{i=1}^n x_i \\
\Phi &= \int_{-\infty}^{\infty} \phi(x) dx \\
\omega &= 2 \pi f
\end{align*}
\end{center}

\end{document}
```

5.3.2 Kalligrafikus betűk

A következő 26 kalligrafikus betű is használható matematikai képletekben:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

Ezeket a `\mathcal` matematikai ábécé paranccsal hívják meg:

`\mathcal{A, B, C,...,Z}`

ABCDEF GHIJK LMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxy z
ABCDEF GHIJK LMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxy z

6. ábra. Matematikai ábécé

5.3.3 Bináris operátorok

A bináris operátorok két matematikai mennyiséget egymással kombinálva egy új mennyiséget állítanak össze bináris művelettel. A bináris operátorként használható szimbólumok a következők

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>	\bigcirc	<code>\bigcirc</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\bullet	<code>\bullet</code>	\square	<code>\Box</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	\diamond	<code>\diamond</code>	\diamond	<code>\Diamond</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\lhd</code>	\triangle	<code>\bigtriangleup</code>
\cdot	<code>\cdot</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\rhd</code>	∇	<code>\bigtriangledown</code>
$*$	<code>\ast</code>	\vee	<code>\vee</code>	\triangleleft	<code>\unlhd</code>	\triangleleft	<code>\triangleleft</code>
\star	<code>\star</code>	\wedge	<code>\wedge</code>	\triangleright	<code>\unrhd</code>	\triangleright	<code>\triangleright</code>
\dagger	<code>\dagger</code>	\oplus	<code>\oplus</code>	\oslash	<code>\oslash</code>	\setminus	<code>\setminus</code>
\ddagger	<code>\ddagger</code>	\ominus	<code>\ominus</code>	\odot	<code>\odot</code>	\wr	<code>\wr</code>
\amalg	<code>\amalg</code>	\otimes	<code>\otimes</code>				

Az aláhúzott szimbólumnevek a fenti és a következő táblázatokban csak akkor érhetők el, ha a *latexsym* vagy az *amsfonts* csomagok valamelyike be van töltve.

Feladat:

Hozd létre a következő szövegrészt!

Halmaz operátorok használata

$$A = \{1, 2, 3, 4, 5\}$$

$$B = \{2, 4, 6\}$$

$$C = \{1, 3, 5, 7\}$$

Metszet:

$$A \cap B \cap C = \emptyset$$

$$A \cap (B \cup C) = \{2, 4\}$$

$$A \cap (B \setminus C) = \{2, 4\}$$

Unió:

$$(A \cup B) \cap (A \cup C) = \{1, 2, 3, 4, 5, 7\}$$

$$(A \cap B) \cup (A \cap C) = \{1, 2, 3, 4, 5\}$$

$$(B \cup C) \setminus A = \{6, 7\}$$

Megoldás:

```
\documentclass[10pt,a4paper]{article}
\usepackage{amssymb}
```

```
\begin{document}
```

```
\begin{center}
```

```
\LaTeX\
```

```
\end{center}
```

```
\vspace{0.5cm}
```

```
\Huge{Halmaz operátorok használata }
```

```
\vspace{0.5cm}
```

```
$A = \{1, 2, 3, 4, 5\}$
```

```
$B = \{2, 4, 6\}$
```

```
$C = \{1, 3, 5, 7\}$
```

Metszet:

```
$A \cap B \cap C = \varnothing$
```

```
$A \cap (B \cup C) = \{2, 4\}$
```

```
$A \cap (B \setminus C) = \{2, 4\}$
```

Unió:

$$(A \cup B) \cap (A \cup C) = \{1, 2, 3, 4, 5, 7\}$$

$$(A \cap B) \cup (A \cap C) = \{1, 2, 3, 4, 5\}$$

$$(B \cup C) \setminus A = \{6, 7\}$$

`\end{document}`

5.3.4 Relációk és tagadásaik

Ha két matematikai mennyiséget hasonlítunk össze, akkor reláció köti össze őket. A különböző típusú relációs szimbólumok a különböző összehasonlításokhoz a következők:

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

A fenti szimbólumok közül néhányat több néven is lehet nevezni. Például a \leq előállítható `\le` vagy `\leq` parancsokkal.

A reláció ellentétes vagy tagadott jelentését a matematika perjellel / jelöljük (pl. $=$ és \neq jelentései egyenlő és nem egyenlő). A fenti szimbólumok bármelyikén perjel áthúzható, ha a neve elé `\not` kerül. Így a `\notin` megegyező \notin -vel. Ugyanez igaz a billentyűzet karaktereire is: `\not=`, `\not>` és `\not<` jelentése \neq , $>$ és $<$. A `\not=` esetén létezik a `\neq` \neq -t létrehozó speciális parancs is, amely önmagában egy szimbólum, és nem a kombinálható `\neq`-vel.

A következő szimbólumok ily módon tagadhatók. Vegye figyelembe, hogy az utolsó kettő, a `\notin` és a `\notin`, nem teljesen ugyanaz: \notin és \notin . Ez utóbbi alakot részesítjük előnyben.

$\not<$	<code>\not<</code>	$\not>$	<code>\not></code>	\neq	<code>\neq</code>
$\not\le$	<code>\not\le</code>	$\not\ge$	<code>\not\ge</code>	$\not\equiv$	<code>\not\equiv</code>
$\not\prec$	<code>\not\prec</code>	$\not\succ$	<code>\not\succ</code>	$\not\sim$	<code>\not\sim</code>
$\not\preceq$	<code>\not\preceq</code>	$\not\succeq$	<code>\not\succeq</code>	$\not\simeq$	<code>\not\simeq</code>
$\not\subset$	<code>\not\subset</code>	$\not\supset$	<code>\not\supset</code>	$\not\approx$	<code>\not\approx</code>
$\not\subseteq$	<code>\not\subseteq</code>	$\not\supseteq$	<code>\not\supseteq</code>	$\not\cong$	<code>\not\cong</code>
$\not\sqsubset$	<code>\not\sqsubset</code>	$\not\sqsupseteq$	<code>\not\sqsupseteq</code>	$\not\asymp$	<code>\not\asymp</code>
\notin	<code>\notin</code>	\notin	<code>\notin</code>		

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Operátorok használata

$$a = b, a \neq b, a \geq b, a < b, a \leq b,$$

$$a \in A, a \notin A,$$

$$\Sigma \subset \Gamma, \Delta \in \Gamma, \Delta \notin \Sigma$$

$$\delta : (Q \setminus \{q_F\}) \times \Gamma \rightarrow 2^{Q \times (\Gamma \cup \{L, R\})}$$

ahol $L, R \notin \Gamma$, *parciális átmeneti függvény*

$q_0 \in Q$ a *kezdőállapot* és $q_F \in Q$
a *végző állapot*

Megoldás:

```
\documentclass[8pt,a4paper]{article}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{Operátorok használata }

\vspace{0.5cm}

$a=b$, $a\neq b$, $a\geq b$, $a<b$, $a\leq b$, $a \in A$, $a \notin A$,

$\Sigma \subset \Gamma$, $\Delta \in \Gamma$, $\Delta \notin \Sigma$

$$\delta : (Q \setminus \{q_F\}) \times \Gamma \rightarrow 2^{Q \times (\Gamma \cup \{L, R\})}$$ ahol $L, R \notin \Gamma$, parciális
\emph{átmeneti függvény}

\vspace{0.5cm}
$q_0 \in Q$ a \emph{kezdőállapot} és $q_F \in Q$ a \emph{végző állapot}
\end{center}

\end{document}
```

5.3.5 Nyilak és mutatók

A matematikai leírások gyakran tartalmaznak nyíl szimbólumokat, amelyeket mutatóknak is neveznek. A következő nyíl szimbólumok állnak rendelkezésünkre:

\leftarrow	<code>\leftarrow</code>	<code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Lleftarrow	<code>\Lleftarrow</code>		\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	<code>\to</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>		\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>		\longleftrightarrow	<code>\longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>		\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>		\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>		\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>		\leftharpoonup	<code>\leftharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>		\leftharpoondown	<code>\leftharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>		\rightleftharpoons	<code>\rightleftharpoons</code>		
\rightsquigarrow	<code>\rightsquigarrow</code>		\rightsquigarrow	<code>\rightsquigarrow</code>		

A \leftarrow és \rightarrow szimbólumokra a `\to` és `\gets` néven is hivatkozhatunk. Továbbá a `\Longleftrightarrow` parancs helyettesíthető `\iff`-el, bár az utóbbi (\iff) mindkét oldalon valamivel nagyobb szóközzel rendelkezik, mint az előbbi (\Leftrightarrow).

5.3.6 Egyéb szimbólumok

A fenti listák semmiképpen sem merítik ki a matematikai szimbólumok teljes repertoárját. A következő további karaktereket a szabványos LATEX elérhetővé teszi számunkra:

\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>	<code>\forall</code>	<code>\forall</code>	\square	<code>\Box</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\exists	<code>\exists</code>	\diamond	<code>\Diamond</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\neg	<code>\neg</code>	\triangle	<code>\triangle</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\flat	<code>\flat</code>	\clubsuit	<code>\clubsuit</code>
ℓ	<code>\ell</code>	∂	<code>\partial</code>	\natural	<code>\natural</code>	\natural	<code>\natural</code>	\blacklozenge	<code>\blacklozenge</code>
\wp	<code>\wp</code>	\top	<code>\top</code>	\sharp	<code>\sharp</code>	\sharp	<code>\sharp</code>	\heartsuit	<code>\heartsuit</code>
\Re	<code>\Re</code>	\perp	<code>\perp</code>	\parallel	<code>\parallel</code>	\parallel	<code>\parallel</code>	\spadesuit	<code>\spadesuit</code>
\Im	<code>\Im</code>	\vdash	<code>\vdash</code>	\angle	<code>\angle</code>	\angle	<code>\angle</code>	\Join	<code>\Join</code>
\mho	<code>\mho</code>	\dashv	<code>\dashv</code>	\backslash	<code>\backslash</code>	\backslash	<code>\backslash</code>	∞	<code>\infty</code>

5.3.7 Kétméretű szimbólumok

A következő szimbólumok különböző méretben vannak megjelenítve attól függően, hogy szövegben vagy megjelenített képletben jelennek meg:

Σ	<code>\Sigma</code>	\sum	<code>\sum</code>	\cap	<code>\cap</code>	\bigcap	<code>\bigcap</code>	\odot	<code>\odot</code>	\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\int	<code>\int</code>	\cup	<code>\cup</code>	\bigcup	<code>\bigcup</code>	\otimes	<code>\otimes</code>	\bigotimes	<code>\bigotimes</code>
\oint	<code>\oint</code>	\oint	<code>\oint</code>	\sqcup	<code>\sqcup</code>	\bigsqcup	<code>\bigsqcup</code>	\oplus	<code>\oplus</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\prod	<code>\prod</code>	\vee	<code>\vee</code>	\bigvee	<code>\bigvee</code>	\uplus	<code>\uplus</code>	\biguplus	<code>\biguplus</code>
\coprod	<code>\coprod</code>	\wedge	<code>\wedge</code>	\bigwedge	<code>\bigwedge</code>						

Az `\int` és `\sum` szimbólumokat már előzőekben bemutattuk. Megmutattuk, hogy ezek a szimbólumok hogyan vehetnek fel felső és alsó határt; ugyanígy az összes fenti szimbólumhoz

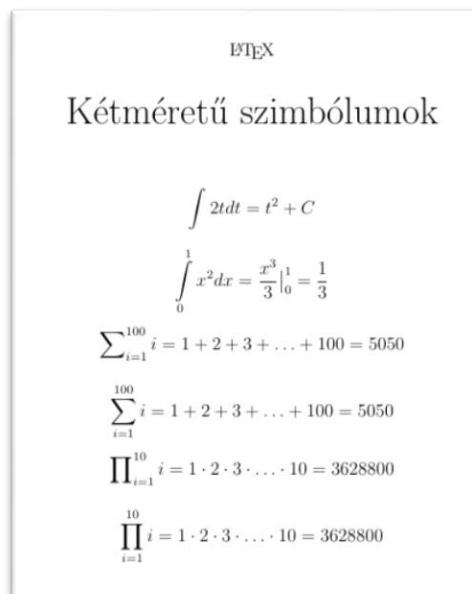
is hozzárendelhető felső és alsó határérték a `^` és `_` karakter parancsokkal. A határértékek elhelyezése egyes szimbólumok esetén attól függően változik, hogy szövegben vagy megjelenített képletben fordulnak-e elő. A `\limits` arra kényszeríti, hogy a határértékeket a szimbólum fölé és alá írják, ahol egyébként mellette lennének. Hasonlóképpen a `\nolimits` kiegészítő parancs a szimbólum mellé állítja be őket, ha a szabványos elhelyezés a szimbólum felett és alatt van.

$$\int_0^{\infty} \quad \int_0^{\infty} \quad \backslash \left[\ \ooint^{\infty}\limits_0 \ \ooint\nolimits^{\infty}_0 \ \backslash \right]$$

$$\prod_{\nu=0}^n \quad \prod_{\nu=0}^n \quad \backslash \left[\ \prod^{\nu}_n \ \prod\nolimits^{\nu}_n \ \backslash \right]$$

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```
\documentclass[12pt,a4paper]{article}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{Kétméretű szimbólumok }
\end{center}
\vspace{0.5cm}
```

```


$$\int 2t \, dt = t^2 + C$$


$$\int \lim_{x \rightarrow 0} x^3 \, dx = \frac{x^4}{4} \Big|_0^1 = \frac{1}{4}$$


$$\sum_{i=1}^{100} i = 1 + 2 + 3 + \dots + 100 = 5050$$


$$\prod_{i=1}^{10} i = 1 \cdot 2 \cdot 3 \cdot \dots \cdot 10 = 3628800$$


```

5.3.8 Függvények

A matematikai képletek univerzális szabálya az, hogy a változóneveket dőlt betűvel, a függvényneveket pedig *Roman* betűstílussal kell megadni. Ha egyszerűen csak matematikai módban íránk a *sin* vagy *log* függvényneveket, a LATEX ezeket *s i n* és *l o g* változókként értelmezné, és *sin* és *log*ként írná ki. Ahhoz, hogy a LATEX függvénynévként kezelje, meg kell adni egy parancsot, amely a fordított perjelből `\` és a függvény nevéből áll. A következő neveket ismeri fel a LATEX:

<code>\arccos</code>	<code>\cosh</code>	<code>\det</code>	<code>\inf</code>	<code>\limsup</code>	<code>\Pr</code>	<code>\tan</code>
<code>\arcsin</code>	<code>\cot</code>	<code>\di</code>	<code>\ker</code>	<code>\ln</code>	<code>\sec</code>	
		<code>m</code>				<code>\tan</code>
					<code>h</code>	
<code>\arctan</code>	<code>\coth</code>	<code>\ex</code>	<code>\lg</code>	<code>\log</code>	<code>\sin</code>	
		<code>p</code>				
<code>\arg</code>	<code>\csc</code>	<code>\gc</code>	<code>\lim</code>	<code>\max</code>	<code>\sinh</code>	
		<code>d</code>				
<code>\cos</code>	<code>\deg</code>	<code>\ho</code>	<code>\liminf</code>	<code>\min</code>	<code>\sup</code>	
		<code>m</code>				

Ezen funkciók némelyike korlátozásokkal is megjelenhet. Ez könnyen elérhető a függvény neve utáni `_` karakter paranccsal. Pl. a `\lim_{x \rightarrow \infty}` a szöveges képletekben $\lim_{x \rightarrow \infty}$ értéket adja, míg a megjelenített képletekben a következőt:

$$\lim_{x \rightarrow \infty}$$

A következő függvénynevekhez a `_` karakter paranccsal egy korlátot adhatunk:

```
\det \gcd \inf \lim \liminf \limsup \max \min \Pr \sup
```

A `\bmod` és `\pmod{arg}` függvényparancsok, amelyek a *mod* függvényt kétféle formában állítják elő:

$$a \bmod b \Rightarrow a \text{ mod } b$$

$\$ y \pmod{a+b} \$ \Rightarrow y \text{ (mod } a + b \text{)}.$

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Függvények

Ebben a mondatban bemutatjuk, hogy lehet egy határértéket tartalmazó képletet írni szöveges képletként, mint egy bekezdés szövege: $\lim_{n \rightarrow \infty} f(n)$. Hasonlóan a további képletek esetében is: $\bigcup_{N \in \mathcal{M}}^N$ valamint $\sum_{j=0}^n x_j^2$. A `\limits` arra kényszeríti, hogy a határértékeket a szimbólum fölé és alá írják, ahol egyébként mellette lennének: $\lim_{n \rightarrow \infty} f(n)$ és $\sum_{j=0}^n x_j^2$

$$\binom{n}{k}$$
$$f(x) = \sin(x) + \log(x) + \arcsin(x)$$
$$f(x) = \cos(x) + \arccos(x) + e^x + |x| + \sum_{i=1}^n a_i + \prod_{i=1}^n a_i$$

Megoldás:

```
\documentclass[12pt,a4paper]{article}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{Függvények }
\end{center}
\vspace{0.5cm}

Ebben a mondatban bemutatjuk, hogy lehet egy határértéket tartalmazó képletet írni szöveges képletként, mint egy bekezdés szövege:  $\lim_{n \to \infty} f(n)$ . Hasonlóan a további képletek esetében is:
 $\bigcup_{N \in \mathcal{M}}^N$  valamint  $\sum_{j=0}^n x_j^2$ .
A \verb|\limits| arra kényszeríti, hogy a határértékeket a szimbólum fölé és alá írják, ahol egyébként mellette lennének:  $\lim\limits_{n \to \infty} f(n)$  és  $\sum\limits_{j=0}^n x_j^2$ 

 $\binom{n}{k}$ 
```

```


$$f(x) = \sin(x) + \log(x) + \arcsin(x)$$


$$f(x) = \cos(x) + \arccos(x) + e^x + |x| + \sum_{i=1}^n a_i + \prod_{i=1}^n a_i$$


```

5.3.9 Matematikai ékezetek

A következő matematikai ékezetek érhetőek el a matematikai módban:

\hat{a} <code>\hat{a}</code>	\breve{a} <code>\breve{a}</code>	\grave{a} <code>\grave{a}</code>	\bar{a} <code>\bar{a}</code>
\check{a} <code>\check{a}</code>	\acute{a} <code>\acute{a}</code>	\tilde{a} <code>\tilde{a}</code>	\vec{a} <code>\vec{a}</code>
\dot{a} <code>\dot{a}</code>	\ddot{a} <code>\ddot{a}</code>	\mathring{a} <code>\mathring{a}</code>	

Az *i* és *j* betűket pontok nélkül kell nyomtatni, ha ékezetet szeretnénk adni nekik. Ennek eléréséhez írjuk be a `\imath` és a `\jmath` szimbólumokat a betűk helyett:

`\vec{\imath} + \tilde{\jmath}` $\vec{i} + \tilde{j}$

A `\hat` és `\tilde` szélesített változatai is elérhetőek `\widehat` és `\widetilde` néven. Ily módon ezek az ékezetek a képlet részei fölé helyezhetőek:

$\widehat{1-x} = \widehat{-y}$ `\widehat{1-x}=\widehat{-y}`
 \widetilde{xyz} `\widetilde{xyz}`

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Matematikai ékezetek

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|}$$

$$\vec{u} \cdot \vec{v} = |\vec{u}| |\vec{v}| \cos(\theta)$$

\mathbf{R}^3 vagy \mathbb{R}^3

$$\vec{v} \times \vec{w} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{vmatrix}$$

$$\ddot{\theta} + \frac{g}{L} \sin \theta = 0$$

$$f(x) = e^{\hat{x}}$$

$$f(\hat{a}) = \frac{1}{2\pi i} \oint_{\Gamma} \frac{f(z)}{z - \hat{a}} dz$$

$$|\ddot{x}|^n = \left(\sum_{i=1}^n |x_i|^n \right)^{\frac{1}{n}}$$

$$\hat{f}, \bar{f}, \tilde{f}, \dot{f}$$

A \hat{O} jelölés az affin térnek egy adott pontját jelöli.

Megoldás:

```
\documentclass[12pt,a4paper]{article}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{Matematikai ékezetek }
\end{center}
\vspace{0.5cm}


$$\hat{v} = \frac{\vec{v}}{|\vec{v}|}$$


$$\vec{u} \cdot \vec{v} = |\vec{u}| |\vec{v}| \cos(\theta)$$


 $\mathbf{R}^3$  vagy  $\mathbb{R}^3$ 


$$\vec{v} \times \vec{w} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{vmatrix}$$



$$\ddot{\theta} + \frac{g}{L} \sin \theta = 0$$


$$f(x) = e^{\hat{x}}$$


$$f(\hat{a}) = \frac{1}{2\pi i} \oint_{\Gamma} \frac{f(z)}{z - \hat{a}} dz$$


$$|\ddot{x}|^n = \left( \sum_{i=1}^n |x_i|^n \right)^{\frac{1}{n}}$$



$$\hat{f}, \bar{f}, \tilde{f}, \dot{f}$$


A  $\hat{O}$  jelölés az affin térnek egy adott pontját jelöli.
```

```
\end{vmatrix}$$
```

```
$$\ddot{\theta} + \frac{g}{L} \sin{\theta} = 0$$
```

```
$$f(x) = e^{\grave{x}}$$
```

```
$$f(\check{a}) = \frac{1}{2\pi i} \oint_{\Gamma} \frac{f(z)}{z-\check{a}} dz$$
```

```
$$|\breve{x}|_n = \left(\sum_{i=1}^n |x_i|^n\right)^{\frac{1}{n}}$$
```

```
$$\hat{f}, \tilde{f}, \bar{f}, \dot{f}$$
```

A \mathring{O} jelölés az affin térnek egy adott pontját jelöli.

```
\end{document}
```

5.4 További elemek

Az előző részekben ismertetett matematikai elemek már lehetővé teszik nagyon összetett képletek készítését is.

$$I(z) = \sin\left(\frac{\pi}{2} z^2\right) \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n}}{1 \cdot 3 \cdots (4n+1)} z^{4n+1} - \cos\left(\frac{\pi}{2} z^2\right) \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n+1}}{1 \cdot 3 \cdots (4n+3)} z^{4n+3}$$

Az előző egyenlet a következővel jön létre:

```
\begin{equation}
```

```
I(z) = \sin\left(\frac{\pi}{2} z^2\right) \sum_{n=0}^{\infty}
```

```
\frac{(-1)^n \pi^{2n}}{1 \cdot 3 \cdots (4n+1)} z^{4n+1}
```

```
-\cos\left(\frac{\pi}{2} z^2\right) \sum_{n=0}^{\infty}
```

```
\frac{(-1)^n \pi^{2n+1}}{1 \cdot 3 \cdots (4n+3)} z^{4n+3}
```

```
\end{equation}
```

A példák az egyenletkörnyezet (*equation*) felhasználásával készültek a *displaymath* környezet vagy annak rövidített alakja $\left[\dots \right]$ helyett, amelynek hatására az egyenletszámok automatikusan hozzáadódnak.

Alapértelmezés szerint az egyenlet száma jobbra igazítva és függőlegesen középre igazítva jelenik meg az egyenlettel. Ha nincs elég hely ugyanazon a sorban, akkor az egyenlet alá jobbra igazítva kerül kinyomtatásra. Ha a *leqno* dokumentumosztály opciót választjuk, az egyenletszámok balra igazítva lesznek beállítva a teljes dokumentumban.

Az egyenletek automatikus számozása azt jelenti, hogy a szerző az írás pillanatában nem tudja, hogy pontosan mi az egyenletszám. Ha a $\label{\text{name}}$ parancsot beépítjük az egyenletkörnyezetbe, az ismeretlen egyenletszámot a $\ref{\text{name}}$ paranccsal nyomtathatjuk ki a szövegbe, ahol a *name* egy kulcsszó, amely betűk, számok vagy szimbólumok tetszőleges kombinációjából áll.

Feladat:

Hozd létre a következő szövegrészt!

ℒ_TE_X

További elemek

$$x^2 - 4x + 4 = 0 \tag{1}$$

$$a_n = \frac{1}{n^2} \tag{2}$$

$$f(x) = x^3 - 2x^2 + 5x - 3 \tag{3}$$

Egyenlet számozása nélkül:

$$\int_0^\infty \frac{x^3}{e^x - 1} dx$$

$$I(z) = \sin\left(\frac{\pi}{2}z^2\right) \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n}}{1 \cdot 3 \cdots (4n+1)} z^{4n+1} - \cos\left(\frac{\pi}{2}z^2\right) \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n+1}}{1 \cdot 3 \cdots (4n+3)} z^{4n+3}$$

Megoldás:

```

\documentclass[12pt,a4paper]{article}
\usepackage{amsthm}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{További elemek }
\end{center}
\vspace{0.5cm}

\begin{equation}
x^2 - 4x + 4 = 0
\end{equation}

\begin{equation}
a_n = \frac{1}{n^2}
\end{equation}

\begin{equation}
f(x) = x^3 - 2x^2 + 5x - 3
\end{equation}

```

Egyenlet számozása nélkül:

```

\begin{equation*}
\int_0^{\infty} \frac{x^3}{e^x-1} dx
\end{equation*}

\begin{equation*}
I(z) = \sin\left(\frac{\pi}{2} z^2\right) \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n}}{1 \cdot 3 \cdot \dots \cdot (4n+1)} z^{4n+1} - \cos\left(\frac{\pi}{2} z^2\right) \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n+1}}{1 \cdot 3 \cdot \dots \cdot (4n+3)} z^{4n+3}
\end{equation*}

\end{document}

```

5.4.1 A zárójelek automatikus méretezése

A matematikában gyakran alkalmazunk zárójeles szimbólumokat, általában párokban, amelyek a képlet egy részét zárják be. Nyomtatáskor ezeknek a zárójeles szimbólumoknak meg kell egyeznie a mellékelt részleges képlet méretével. Ennek megvalósításához a LATEX egy parancspárt biztosít:

```
\left lbrack képletek \right rbrack
```

A `\left` parancs közvetlenül a nyitó (bal oldali) zárójeles `lbrack` szimbólum elé kerül, míg a `\right` közvetlenül a záró (jobb oldali) szimbólum `rbrack` elé kerül.

A `\left[\int + \int \right]_{x=0}^{x=1}` eredménye:

$$\left[\int + \int \right]_{x=0}^{x=1}$$

A `\left` és `\right` parancsoknak párban kell megjeleníteniük. A párok egymásba ágyazhatók. Az első `\left` párosul az utolsó `\right` paranccsal; a következő `\left` az utolsó előtti `\right` paranccsal, és így tovább. A beágyazásban ugyanannyi `\right`, mint `\left` parancsnak kell lennie.

A megfelelő `lbrack` és zárójelek teljesen tetszőlegesek lehetnek, és nem kell párban lenniük.

Néha egy képlet csak egyetlen nyitó vagy záró zárójelet tartalmaz megfelelő párja nélkül. Azonban a `\left . . . \right` parancsokat továbbra is párként kell megadni, de egy „,” ponttal láthatatlan zárójelként:

$$y = \begin{cases} -1 & : x < 0 \\ 0 & : x = 0 \\ +1 & : x > 0 \end{cases} \quad \left[y = \left\{ \begin{array}{l} \\ \\ \\ \end{array} \right. \right. \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right. \right]$$

A `\left. . . . \right` parancsok a következő szimbólumra alkalmazhatók:

(())	[\lfloor]	\rfloor
[[]]	[\lceil]	\rceil
{	\{	}	\}	<	\langle	>	\rangle
			\	↑	\uparrow	↑	\Uparrow
/	/	\	\backslash	↓	\downarrow	↓	\Downarrow
				↕	\updownarrow	↕	\Updownarrow

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

A zárójelek automatikus méretezése

$$\binom{n}{k} = \begin{cases} \frac{n!}{k!(n-k)!} & \text{ha } 0 \leq k \leq n \\ 0 & \text{ha } k > n \text{ vagy } k < 0 \end{cases}$$

$$F(x) = \begin{cases} 0 & \text{ha } x \leq 0 \\ \sin x & \text{ha } 0 < x < \pi/2 \\ 1 & \text{ha } \pi/2 \leq x \end{cases}$$

$$\text{abs}(x) = \begin{cases} x & \text{ha } x \geq 0 \\ -x & \text{máskülönben} \end{cases}$$

Megoldás:

```

\documentclass[12pt,a4paper]{article}
\usepackage{amsthm}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{A zárójelek automatikus méretezése }
\end{center}
\vspace{0.5cm}

\[ \left( \begin{array}{c} n \\ k \end{array} \right) = \left\{ \begin{array}{l} \frac{n!}{k!(n-k)!} \\ 0 \end{array} \right. \text{ ha } 0 \leq k \leq n \\ \text{ ha } k > n \text{ vagy } k < 0 \\ \end{array} \right. \]

\[ F(x) = \left\{ \begin{array}{l} 0 \\ \sin x \\ 1 \end{array} \right. \text{ ha } x \leq 0 \\ \text{ ha } 0 < x < \pi/2 \\ \text{ ha } \pi/2 \leq x \\ \end{array} \]

\sin x &

```

```
\textrm{ha }$0 < x < \pi/2$ \\  
1 & \textrm{ha }$\pi/2 \leq x$\  
\end{array} \right. \]
```

```
$$\mathrm{abs}(x) = \left\{\  
\begin{array}{c} l \\ x & \text{ha } x \geq 0 \\ -x & \text{máskülönben } \\ \end{array} \right. $$
```

```
\end{document}
```

5.4.2 Mátrixok és tömbök

Az alábbihoz hasonló struktúrák alapjai a mátrixok, determinánsok, egyenletrendszerek. Itt mindegyikre tömbként fogunk hivatkozni.

```
a11 a12 ... a1n  
⋮ ⋮ ⋱ ⋮  
an1 an2 ... ann
```

A tömbök a tömbkörnyezet (*array*) segítségével jönnek létre. A tömbkörnyezet matematikai módban generál táblázatot, vagyis az oszlopbejegyzéseket képletszöveggé értelmezi. Például:

```
a11x1 + a12x2 + ... + a1nxn = b1  
a22x1 + a22x2 + ... + a2nxn = b2  
.....  
an1x1 + an2x2 + ... + annxn = bn
```

```
\[ \begin{array}{*{3}{c@{\:+\:}}c@{\:=\:}c}\  
a_{11}x_1 & a_{12}x_2 & \cdots & a_{1n}x_n & b_1 \\ a_{22}x_1 & a_{22}x_2 & \cdots & a_{2n}x_n & b_2 \\ \multicolumn{5}{c}{\dotfill} \\ a_{n1}x_1 & a_{n2}x_2 & \cdots & a_{nn}x_n & b_n \\ \end{array} \]
```

A `@{t}` beszúrja a *t* tartalmát a szomszédos oszlopok közé. A fenti példában ez `\:+\:` és `\:=\:`. A `\:` és `\;` kis vízszintes térközt állítanak elő matematikai módban. `*{3}{c@{\:+\:}}` a `c@{\:+\:}` oszlopdefiníció három ismétlésének rövidítése. A *c* az oszlopot középre állított szöveggé határozza meg.

A `\multicolumn{5}{c}` azt mondja, hogy a következő öt oszlopot össze kell vonni, és egy középre helyezett szöveggel kell helyettesíteni. A `\dotfill` parancs kitölti az oszlopot pontokkal.

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Mátrixok és tömbök

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{vmatrix} = 5$$

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix} = \begin{vmatrix} t & u \\ v & w \end{vmatrix} = tw - uv$$

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \Rightarrow \begin{array}{cc} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{array}$$

Megoldás:

```
\documentclass[12pt,a4paper]{article}
\usepackage{amsthm}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{Mátrixok és tömbök }
\end{center}
\vspace{0.5cm}

\[
\left[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
9 & 10 & 11 & 12 \\
13 & 14 & 15 & 16
\end{array}
\right]
= 5
\]
```

```

\[ A =
\left|\begin{array}{cccc}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn}
\end{array}\right|
=
\left|\begin{array}{cc}
t & u \\
v & w
\end{array}\right|
= tw - uv\]

\[
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6
\end{array}
\quad \rightarrow \quad
\begin{array}{cc}
1 & 4 \\
2 & 5 \\
3 & 6
\end{array}
\]
\end{document}

```

5.4.3 Vonalak a képletek felett és alatt

Az alábbi parancsok használható vonalak rajzolására képletek fölé vagy alá.

`\overline{képlet}` és `\underline{képlet}`

Bármilyen szintre beágyazhatók:

```
\[ \overline{\overline{a}^2 + \underline{xy} + \overline{\overline{z}}} \]
```

$$\overline{\overline{a^2 + \underline{xy} + \overline{\overline{z}}}}$$

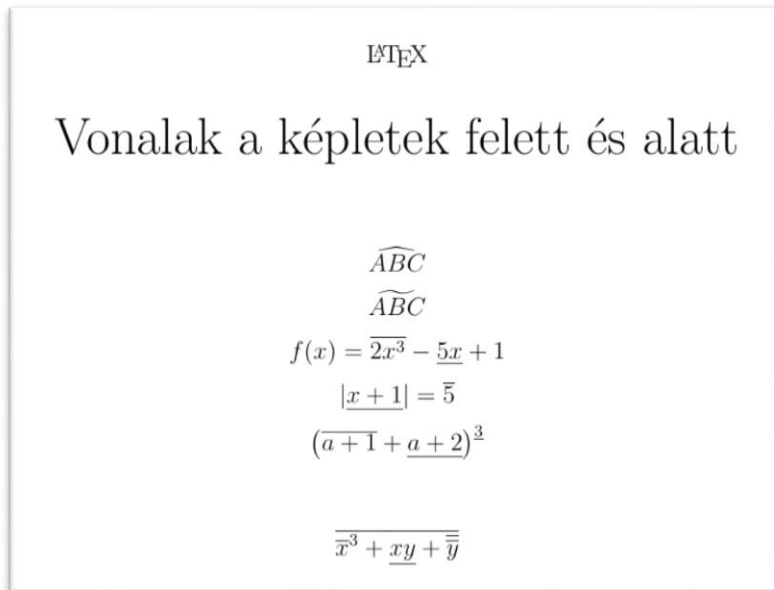
Az `\underline` parancs normál szöveges módban is használható szöveg aláhúzására, míg az `\overline` csak matematikai módban engedélyezett.

A `\overbrace{képlet}` és `\underbrace{képlet}` parancsok segítségével vízszintes kapcsos zárójeleket helyezhetünk a képletek fölé vagy alá.

$$\overbrace{a + b + c + d} \quad \overbrace{a + \underbrace{b+c} + d}$$

Feladat:

Hozd létre a következő szövegrészt!



Megoldás:

```

\documentclass[12pt,a4paper]{article}
\usepackage{amsthm}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\

\vspace{0.5cm}
\Huge{Vonalak a képletek felett és alatt }
\end{center}
\vspace{0.5cm}

$$\widehat{ABC}$$
$$\widetilde{ABC}$$
$$f(x) = \overline{2x^3} - \underline{5x} + 1$$
$$\left|\underline{x+1}\right| = \overline{5}$$
$$\left(\overline{a+1} + \underline{a+2}\right)^{\underline{3}}$$

\vspace{0.5cm}
\left[\overline{\overline{x}^3} + \underline{xy} + \overline{\overline{y}}\right]

\end{document}

```

5.4.4 Többsoros egyenletek

A többsoros egyenletek esetében az egyes sorokban lévő relációs szimbólumok függőlegesen vannak egymáshoz igazítva. Erre a célra a megjelenített matematikai módban a környezetek több képletsor vagy egyenlet beállítására szolgál.

```
\begin{eqnarray} sor_1 \\ ... \\ sor_n \end{eqnarray}
```

```
\begin{eqnarray*} sor_1 \\ ... \\ sor_n \end{eqnarray*}
```

Az egyenlet vagy képlet egyes sorait `\\` választja el egymástól.

Az `eqnarray` környezet szabványos formája egy automatikus szekvenciális egyenletszámot ad hozzá, amely hiányzik a *-os alak esetében. Ha valamelyik sornál nem szeretnénk az egyenletszámot megjeleníteni, akkor közvetlenül a sorlezárás `\\` elé hozzáírjuk a `\nonumber` parancsot.

Az egyenletszámokra hivatkozhatunk a szövegben a paranccsal

```
\ref{name}
```

, miután a kulcsszó neve hozzá lett rendelve az egyenletszámhoz a

```
\label{name}
```

 paranccsal valahol az egyenlet sorában.

Feladat:

Hozd létre a következő szövegrészt!

L^AT_EX

Többsoros egyenletek

$$\begin{aligned} 3x &= 6(x-9) + 15 & (1) \\ 3x &= 6x - 54 + 15 & (2) \\ 3x - 6x &= -39 & (3) \\ 3x &= 39 & (4) \\ x &= 13 & (5) \end{aligned}$$
$$x = \left\{ a \oplus \left[b \cdot (c \ominus d) \right] \right\}^{4/2} \quad (6)$$
$$y = \lim_{\beta \rightarrow \infty} \frac{\tan^2 \beta - \sin^3 \beta}{\frac{1}{\cos 4\beta + 7}} \quad (7)$$
$$\begin{aligned} 2x + 3y - z &= 1 & (1) \\ x - 4y + 3z &= -2 \\ 3x - y - z &= 4 & (2) \end{aligned}$$
$$\begin{aligned} x_1 - x_2 - 2x_3 - x_4 + 3x_5 &= 4 & (3) \\ 2x_1 + 3x_2 - 2x_3 + x_4 - 3x_5 &= 2 & (4) \\ -3x_1 - 2x_2 + 4x_3 - x_4 - 5x_5 &= -6 & (5) \\ x_1 + 2x_2 - 2x_3 + 3x_4 + 4x_5 &= 3 & (6) \\ 4x_1 + 3x_2 + 3x_3 - 5x_4 - x_5 &= -1 & (7) \end{aligned}$$

Megoldás:

```
\documentclass[12pt,a4paper]{article}
```

```

\usepackage{amsthm}
\usepackage{amsmath}

\begin{document}

\begin{center}
\LaTeX\
\vspace{0.5cm}
\Huge{Többsoros egyenletek }
\end{center}
\vspace{0.5cm}

\setcounter{equation}{0}
\begin{eqnarray}
3x & = & 6(x-9)+15 \\
3x & = & 6x-54+15 \\
3x-6x & = & -39 \\
3x & = & 39 \\
x & = & 13
\end{eqnarray}

\begin{eqnarray}
\label{egyenlet:1}
x & = & \bigg\{a \oplus \Big[b \cdot \big(c \ominus d\big)\Big]\bigg\}^{4/2} \\
\label{egyenlet:2}
y & = & \lim_{\beta \to \infty} \frac{\tan^2 \beta}{\sin^3 \beta} \left\{ \frac{1}{\log_2 x} + \frac{1}{2} \right\}
\end{eqnarray}

\setcounter{equation}{0}
\begin{eqnarray}
2x + 3y - z & = & 1 \\
x - 4y + 3z & = & -2 \text{ \nonumber } \\
3x - y - z & = & 4
\end{eqnarray}

\begin{eqnarray}
x_1 - x_2 - 2x_3 - x_4 + 3x_5 & = & 4 \\
2x_1 + 3x_2 - 2x_3 + x_4 - 3x_5 & = & 2 \\
-3x_1 - 2x_2 + 4x_3 - x_4 - 5x_5 & = & -6 \\
x_1 + 2x_2 - 2x_3 + 3x_4 + 4x_5 & = & 3 \\
4x_1 + 3x_2 + 3x_3 - 5x_4 - x_5 & = & -1
\end{eqnarray}

\end{document}

```


A dokumentumunk elején a *graphics* és *color* csomagok betöltésekor meg kell adni az illesztőprogram-opciót:

```
\usepackage[dvips]{graphics,color}
```

Elképzelhető, hogy telepítéskor már ezt automatikusan megtette, ezért érdemes megnézni létezik-e már alapértelmezett illesztőprogram-beállítás.

6.1.1 Külső grafikák importálása

Két csomag áll rendelkezésre a külső grafikus fájlok importálására és manipulálására: az alap *graphic* csomag és a bővebb *graphicx* csomag. Mindkettő azonos funkcionalitást kínál, csak a szintaxisukban különböznek.

Az illesztőprogramok nevein kívül van néhány egyéb opció is, amelyek a *graphics* vagy *graphicx* csomagok betöltésekor meghívhatók:

- draft* nem importál, hanem egy keretezett keretet helyez el a grafika helyén, amelybe a fájl nevét nyomtatja; ez jelentősen felgyorsítja a feldolgozást, ha csak a szöveggel dolgozunk;
- final* ellensúlyozza a draftot; akkor szükséges, ha a *draft* opciót globálisan adtuk ki a `\documentclass`-ban;
- hidescale* üres helyet hagy a méretezett szöveg helyén;
- hiderotate* üres helyet hagy ott, ahol elforgatott szövegnek kellene lennie; ez és a *hidescale* opció hasznos, ha az előnézeti program nem tudja kezelni a méretezést vagy az elforgatást.

6.1.2 Kép importálása a *graphic* csomaggal

A képet a *graphic* csomaggal a következő paranccsal importáljuk:

```
\includegraphics[llx,lly][urx,ury]{fájlnév}
```

ahol *llx*, *lly* a bal alsó sarok, *urx*, *ury* pedig a jobb felső sarok koordinátái a beillesztendő képnek.

Azt adják meg, hogy hol vágjuk a képet méretre. Az egységek megadhatók például [2cm, 4in] módon. Ha csak egy opcionális argumentumot adunk meg, akkor az a jobb felső sarok, a bal alsó pedig [0,0].

További importálási lehetőség:

```
\includegraphics{fájlnév}
```

Fájlnév megadásakor nem szükséges megadni a grafikus fájl kiterjesztését.

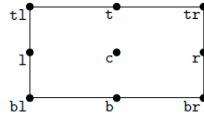
Az `\includegraphics` parancs fontosabb opciói:

- width* A kép szélességét határozza meg (*width=4cm*).

height A kép magasságát határozza meg ($height=3cm$). A *width* és a *height* együttes megadásával a képet torzíthatjuk is.

angle Kép forgatásának szögét adhatjuk megfokokban. A pozitív érték az óra járásával ellentétes irány.

origin Megadásával a forgatás középpontját határozhatjuk meg. A középpont értékei a következők lehetnek: *tl*, *t*, *tr*, *l*, *c*, *r*, *bl*, *b*, *br* (alapértelmezettértéke *bl*). Ezek magyarázata a következő ábrán látható:



Az alábbiakban a kép szélességét 4 cm-re, magasságát 3 cm-re állítjuk és elforgatjuk 90 fokkal az óra járásával megegyező irányban a kép középpontja körül:

```
\includegraphics[width=4cm, height=3cm, angle=-90, origin=c]{fajlnév}
```



7. ábra. Kép méretei beállítva és elforgatva.

Feladat:

Hozd létre a következő szövegrészt!

LaTeX

Az alábbi ábrán látható egy példa a `graphicx` csomaggal történő ábrakezelésre. Az ábra 50% szélességben jelenik meg a dokumentumban.



Figure 1: Ez egy példa ábra.

A következő ábrán a pontosan beállítjuk a kép szélességét (4 cm) és magasságát (3 cm), elforgatjuk jobbra a kép középpontja szerint 45 fokkal.



Figure 2: Ez másik példa ábra.

Megoldás:

```
\documentclass{article}
\usepackage{graphicx}
```

```
\begin{document}
\begin{center}
\LaTeX \
```

Az alábbi ábrán látható egy példa a `\texttt{graphicx}` csomaggal történő ábrakezelésre. Az ábra 50\% szélességben jelenik meg a dokumentumban.

```
\end{center}
```

```
\begin{figure}[ht]
\centering
\includegraphics[width=0.5\textwidth]{Hawk.png}
\caption{Ez egy példa ábra.}
\label{fig:pelda}
\end{figure}
```

A következő ábrán a pontosan beállítjuk a kép szélességét (4 cm) és magasságát (3 cm), elforgatjuk jobbra a kép középpontja szerint 45 fokkal.

```
\begin{figure}[ht]
\centering
\includegraphics[width=4cm, height=3cm,angle=45,origin=c]{Hawk.png}
\caption{Ez másik példa ábra.}
\label{fig:pelda}
\end{figure}

\end{document}
```

7 Felhasznált irodalom

Helmut Kopka, Patrick W. Daly: Guide to LATEX, 4th Edition. Addison -Wesley 2004, pp. 624, ISBN 0-321-17385-6.

Wettl Ferenc, Mayer Gyula, Szabó Péter: LaTeX kézikönyv, Panem Kiadó, 2004, pp. 780, ISBN 963545398-1.

Wettl Ferenc, Mayer Gyula, Sudár Csaba: LaTeX kezdőknek és haladóknak, Budapest, 1998, Panem Kiadó, 1998, pp. 409, ISBN 9635451415

Donald E. Knuth: The TeXbook, Reading, Massachusetts: Addison-Wesley, 1984, pp. 483, ISBN 0-201-13448-9

Tómacs Tibor: LaTeX

URL: <https://tomacstibor.uni-eszterhazy.hu/tananyagok/LaTeX.pdf>

Tobias Oetiker, Hubert Partl, Irene Hyna, Elisabeth Schlegl: The Not So Short Introduction to LATEX 2 ϵ

URL: <https://gking.harvard.edu/files/lshort2.pdf>



Selye János Egyetem Gazdaságtudományi és Informatikai Kar
Hradná ul. 21.
SK-945 01 Komárno
fei.ujs.sk

Autor/Szerző
RNDr. Udvaros József, PhD.

Názov/ Cím
Bevezetés a LATEXbe

Recenzenti/recenzensek:
Avornicului Constantin Mihai, PhD.
Gubán Ákos, Professor Emeritus

Jazyková korektura/Nyelvi ellenőrzés:
Dr. habil. Bódi Stefánia

Vydavateľ/Kiadó:
Univerzita J. Selyeho
Selye János Egyetem

Rozsah/Terjedelem:
4,6 AH/4,6 szerzői ív

Rok vydania/Megjelenés éve
2023

Prvé vydanie/Első kiadás

ISBN 978-808122-459-1