

Digitális kultúra emelt

(A) szóbeli mintatételek

köszö Dani :)

Tartalom:

Digitális kultúra emelt	1
I. Dokumentumszerkesztes – Szövegszerkesztes	3
1. tétel: Betű és bekezdésformázás	3
2. tétel: Táblázatok és tabulátorok	6
3. tétel: Objektumok beszúrása	8
4. tétel: Oldalbeállítások, fájlműveletek	10
5. tétel: Kördokumentum	12
6. tétel: Nagy dokumentumok formázása	13
II. Dokumentumkészítés – Publikálás a világhálón	15
7. tétel: Betű és bekezdésformázások	15
8. tétel: Objektumok beszúrása	17
9. tétel: Stíluslapok használata	20
10. tétel: Dobozmodell.....	23
III. Dokumentumkészítés – Bemutatókészítés.....	25
11. tétel: A bemutatókészítés alapjai.....	25
12. tétel: A bemutatókészítéshez használható eszközök.....	27
IV. Dokumentumkészítés – Pixelgrafika.....	29
13. tétel: A pixelgrafikus ábrázolás alapjai	29
14. tétel: Pixelgrafikus szerkesztés.....	32
V. Dokumentumkészítés – Vektorgrafika	33
15. tétel: A vektorgrafikus ábrázolás alapjai	33
16. tétel: Vektorgrafikus szerkesztés	35
VI. Táblázatkezelés	37
17. tétel: A táblázatkezelés alapjai.....	37
18. tétel: Függvények	39
VII. Adatbázis-kezeles.....	41
19. tétel: Az adatbázis-kezelés alapjai.....	41
20. tétel: SQL utasítások és függvények.....	43

I. Dokumentumszerkesztes – Szövegszerkesztes

1. tétel: Betű és bekezdésformázás

Milyen betűtípusokat különböztetünk meg? Milyen betűstílusokat ismersz? Milyen speciális betűformázási beállításokat ismersz? Mit tudsz a szimbólum karakterekről? Milyen mértékegységeket használunk szövegszerkesztőben? Mondj példákat bekezdésformázásokra! Mit tudsz a listákról, felsorolásokról? Mit tudsz az iniciáléről?

A **betűtípusok** olyan karakterkészletek, amik meghatározzák a szöveg megjelenését. Több fő kategória van, ezek a továbbiak:

- talpas (pl. Times New Roman)
 - díszítő, valamint olvasást segítő céllal a betűk végén kis vonalak vannak
 - pl. Árvíztűrő tükörfúrógép
- talpatlan (pl. Arial)
 - egyszerűbb vonalvezetésű, egyszerű betűk
 - pl. Árvíztűrő tükörfúrógép
- monospace (rögzített szélességű) (pl. courier new)
 - minden betű azonos szélességgel rendelkezik, az írógépek használták eredetileg ezeket a betűstílusokat
 - pl. Árvíztűrő tükörfúrógép
- díszbetűk
 - különleges megjelenésű betűk, címsorokban vagy kreatív szövegekben használtak
 - pl. Árvíztűrő tükörfúrógép

A **betűstílusok** a betűk megjelenésének módosítására szolgál pl. más jellegű információ kiemelése céljából

- **félkövér (bold)**
- *dőlt (italic)*
- aláhúzott (underline)
 - az aláhúzás típusát a speciális beállítások között lehet állítani

a **speciális betűstílusok** külön felugró ablakból érhetőek el, további formázási lehetőséget adnak. Ezek a következők:

- KISKAPITÁLIS
- NAGYBETŰS
- felső index (superscript)
- alsó index (subscript)
- Szín és kiemelés (pl. **kiemelt háttérszín**, **szövegszín**)

A **szimbólum karakterek** olyan nem alfanumerikus karakterek, melyek segítik a dokumentumok szerkesztését.

numerikus billentyűzettel rendelkező számítógépnél az alt + a szimbólumhoz tartozó számsor leütésével, egyéb esetben a beszúrás menüpontból érhetőek el ezek a szimbólumok

- matematikai jelek +, -, ×, ÷, √
- pénz jelek: €, \$, £
- nyilak, grafikus elemek: ←, →, ■
- különleges karakterek: @, #, ©, ®

A szövegszerkesztők különböző beállításokhoz különböző **mértékegységeket** használnak. Ezek a következők:

- **pont (pt)**
- **Milliméter (mm), vagy centiméter (cm)**
- **Pixel (px)**
- **Em** (a betű névleges magassága alapján számolt egység)

Bekezdések formázásához a lehetőségek a következők:

- **sorköz**
 - (pl. szimpla, többszörös, 1,5 sor)
- **behúzások**
 - (pl. első sor behúzása, balra/jobbra/középre igazítása)

- **igazítás**
 - balra
 - középre
 - jobbra
 - sorkizárt (Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.)
- **távolság**
 - bekezdések közötti távolság növelése, vagy csökkentése

a listák és felsorolások:

- **felsorolás** a listaelemek elejére egy szimbólum, piktogram kerül (pl. pont
⇒ nyíl
* csillag)
- 1. **számozott lista**
- 2. a listaelemek sorszámozást kapnak
- 1. **az allistákban** a belső elemek behúzással és eltérő sorszámozással rendelkeznek
 - a. így

A

z iniciálé egy nagyméretű, díszített kezdőbetű, amit nagyrészt nyomtatott kiadványokban vagy díszes dokumentumokban használnak. Az iniciálé több sort is elfoglalhat.

2. tétel: Táblázatok és tabulátorok

A táblázatokon milyen elrendezésbeállításokat tudunk eszközölni? Milyen lehetőségeink vannak táblázatok formázására? Mi a funkciója a tabulátor gombnak? Mit tudsz a tabulátorpozíciókról? Milyen egyéni tabulátorokat tudunk felvenni?

A táblázatok elrendezési lehetőségei:

- **oszlopok és sorok méretének módosítása**
 - az egyes cellák mérete állítható manuálisan, vagy automatikusan a tartalomhoz igazítva
- **cellaegyesítés és -felosztás**
 - lehetőségünk van több cella egyesítésére (mint excelben), illetve egy cella több kisebb cellára bontására
- **igazítás beállítása**
 - a tartalom elhelyezhető balra, középre vagy jobbra igazítva, illetve a függőleges igazítás is testre szabható (felső, középső, alsó)
- **táblázat pozicionálása a dokumentumban**
 - a táblázat elhelyezhető az oldalon belül különböző módokon (pl. körbefuttatással, vagy önálló elemként kezelve)

A táblázat formázási lehetőségei

- **szegélyek és vonalstílusok módosítása**
 - a cellák köré kerülő, különböző vastagságú és típusú (pl. folytonos, szaggatott, dupla vonal) szegélyek állíthatók
- **háttérszín és kitöltés beállítása**
 - a cellák vagy a sorok színezhetők a jobb, értelmezhetőbb vizuális elkülönítés érdekében
- **betűtípus és szövegformázás**
 - a táblázaton belüli szöveg formázható a szövegszerkesztőben alapból elérhető betűformázási opciókkal (pl. betűméret, betűtípus, betűstílus)

A **tabulátor** funkciója és beállításai:

A **tabulátor gomb** (TAB, vagy ␣) segítségével gyorsan mozgathatjuk a kurzort előre meghatározott helyekre a szövegsorban. Ez listák, felsorolások és táblázatszerű adatok bevitelkor a leghasznosabb. Alapbeállításként egy tabulátor nagysága 4 db space karakternek felel meg.

A tabulátorpozíciók típusai

a tabulátorpozíciók beállításai a vonalzóra kattintással érhető el egy felugró ablakban.

Típusai a következők:

- **Balra igazított tabulátor** (a szöveg a beállított ponttól balra kezdődik)
- **jobbra igazított tabulátor** (a szöveg a beállított ponttól jobbra helyezkedik el)
- **középre igazított tabulátor** (a szöveg a tabulátorpozícióhoz viszonyítva középre helyezkedik el)
- **tizedes tabulátor** (a számok tizedespontja egymás alá kerül, így könnyen azonosíthatók, átláthatók)

Egyéni tabulátorok felvétele

a szövegszerkesztők lehetőséget biztosítanak egyéni tabulátorpozíciók beállítására, ezek Wordben a vonalzóra kattintással érhető el egy felugró ablakban.

- **tabulátorpozíciók beállítása a vonalzon:** a vonalzon kattintással, vagy manuális beállítással meghatározzuk, hogy a szöveg hova kerüljön
- **kitöltő karakterek használata:** a tabulátorhelyek közötti részt pontokkal, vonalakkal vagy egyéb karakterekkel jelölhetjük ki.
 - pl. név..... pontszám
- **különböző tabulátorok kombinálása:** egy soron belül több(féle) tabulátor is használható a komplexebb elrendezés érdekében.

3. tétel: Objektumok beszúrása

Milyen objektumokat lehet szövegszerkesztőbe beszúrni? Alakzatok esetében mit nevezünk csomópontnak? Milyen lépései vannak egy kép beillesztésének? Milyen hasonló, és milyen egyedi beállításokat lehet egy beszúrt képen, illetve egy beszúrt alakzaton eszközölni? Mit tudsz a szövegdobozokról? Mit tudsz a hivatkozások beszúrásáról?

Szövegszerkesztőben különböző objektumokat különböztetünk meg, melyek a dokumentum esztétikai javítása és az információk vizuális kiemelése érdekében fontosak. A leggyakoribb objektumtípusok:

- **Képek** (pl. JPG, PNG, GIF formátumban)
- **Egyéb média** (videók, hangfájlok pl.)
- **Alakzatok** (pl. vonalak, nyilak, téglalapok, körök, egyéni alakzatok) – ezek vektorgrafikus ábrák, tehát bármilyen skálán nagyíthatóak minőségvesztés nélkül
- **Diagramok és grafikonok**
- **Táblázatok**
- **Szövegdobozok**
- **Hivatkozások** (hiperhivatkozások)
- **Matematikai képletek és egyenletek**
- **Szimbólumok és különleges karakterek**

Alakzatok esetében **csomópontnak** nevezzük azok szerkesztésekor azokat a pontokat, amiket módosítva megváltozik az alakzat formája. Ezek az alakzatok szerkezetét meghatározó kulcspontok, amiket a felhasználó áthelyezhet, átalakíthat, vagy törölhet. Csomópontokat az egyéni alakzatok és szabadkézi rajzok módosítására használnak.

Egy kép beillesztésének lépései a következők:

1. **A beszúrási lehetőség kiválasztása:** A *beszúrás* menüben a *kép* opció kiválasztása (vagy a vágólapról történő beszúrás, amennyiben előzetesen oda kimásoltuk, ebben az esetben a következő lépés kihagyandó)
2. **A forrás kiválasztása:** kép beszúrása számítógépről, internetről, vagy online (stock) forrásból.
3. **A kép elhelyezése:** A beszúrt kép a dokumentumban megjelenik szükség esetén a következőképp módosítható:
4. **Méret és pozíció módosítása:** A kép átméretezése és áthelyezése az oldal megfelelő részére
5. **Formázási beállítások alkalmazása:** Igazítás, körbefuttatás, szegélyek és különböző effektusok beállítása

Egy képen vagy egy alakzaton lehet **hasonló és egyedi** beállításokat eszközölni:

Hasonló beállítások:

1. méret módosítása
2. elhelyezés és igazítás
3. körbefuttatás a szöveghez viszonyítva (négyzetes, szoros, keresztül, alatta és fölött)
4. szegélyek és árnyékok alkalmazása
5. forgatás és tükrözés

Egyedi beállítások

kép esetén: színmódosítás, fényerő és kontraszt állítása, átlátszóság, képeffektusok alkalmazása

alakzatok esetén: kitöltési szín, körvonalstílus és -vastagság beállítása, alakzat módosítása a **csomópontok** segítségével

A szövegdobozok olyan speciális objektumok, amelyekbe szöveget lehet írni és tetszőlegesen elhelyezhetők a dokumentumban. főbb jellemzőik:

- függetlenek a szövegtől, mozgathatók és formázhatók
- általános szövegformázási beállítások alkalmazhatók rajta (betűtípus- és bekezdésformázás)
- összekapcsolhatók más szövegdobozokkal a „folytonos szövegáramlás” érdekében

Hiperhivatkozásokat lehet beszúrni a dokumentumba, amelyek kattintásra (újabb szövegszerkesztőkben shift + kattintásra) egy adott weboldalra, dokumentum más részére, vagy külső fájlra irányíthatnak.

a beszúrás lépései:

1. a hiperhivatkozásként használni kívánt szöveg kijelölése
2. a *hivatkozás* opció kiválasztása a *beszúrás* menüben, VAGY a *Ctrl + K* billentyűkombináció megnyomása
3. a hivatkozás típusának meghatározása (lehet weboldal, fájl, dokumentum egy pontja, vagy e-mail cím)
4. a cél megadása és egyéb beállítások elvégzése

4. tétel: Oldalbeállítások, fájlműveletek

Miért fontosak a nyelvi beállítások (nyelvi ellenőrzés, elválasztás)? Ismertesd a keresés és csere funkció használatát! Milyen oldalbeállításokat tudunk szövegszerkesztőben végrehajtani? Hogyan készíthetjük elő dokumentumunkat nyomtatásra, PDF-be exportálásra? Mik az előnyei a PDF fájlformátumnak?

A szövegszerkesztők lehetőséget biztosítanak különböző nyelvi beállításokra, amik jelentősen megkönnyíthetik a helyesírási- és stilisztikai hibák elkerülését, valamint a dokumentum esztétikai formázását és az akadálymentesség minél célszerűbb elérését.

A nyelvi ellenőrzés:

- Az automatikus helyesírás-ellenőrző kiemeli a hibásan írt szavakat (piros, hullámos aláhúzással) és javaslatokat kínál a helyes alakra
- A nyelvtani ellenőrzés segít az esetleges szerkezeti hibák, például egyeztetési, vagy központozási hibák felismerésében (duplán, kékkel aláhúzva a problémás részt)
- Egyes szövegszerkesztők bizonyos nyelveken rendelkeznek szinonima szótárral, ami segítséget jelent egy választékosabb szókincsű dokumentum megírásában

Elválasztás:

- Két féle elválasztásra van lehetőség: automatikus és manuális. Ezek segítenek a szöveg szebb tördelésében, különösen hatásos többhasábos vagy keskeny hasábokkal rendelkező dokumentumok esetében.
- Az elválasztás csökkenti a sorvégi szóközöket (amennyiben szükséges) és egyenletesebb szövegmejlénést biztosít.

A **keresés és csere** funkció lehetővé teszi, hogy a dokumentumban bizonyos szavakat gyorsan megtaláljunk és szükség szerint egyszerre módosítsuk valamennyit.

- **keresés:** egy adott szó, vagy kifejezés megkeresése a dokumentumban (Ctrl + F, vagy az ablak jobb felső sarkában található keresőmező használatával érhető el.)
- **csere:** az előzetesen megkeresett szó/kifejezés cseréje egy másikra. Hasznos lehet egy hibás név vagy a tizedesvessző pontra módosításakor.
- **speciális keresési beállítások:** például kis- és nagybetűk figyelembe- vagy nem vétele, teljes szavak keresése

A dokumentum megfelelő **oldalbeállításai** segítenek az esztétikus és nyomtatásra alkalmas formátum elérésében.

- **papírméret:** A4, A3, vagy pl. US letter méretű oldalméretek beállítása
- **tájolás:** álló vagy fekvő mód választása
- **margók beállítása:** a dokumentum szélein hagyott térközök átméretezése
- **hasábok létrehozása:** többhasábos tördelés a jobb olvashatóság érdekében (versknél, vagy újságcikkeknel használatos például).
- **sortávolság és behúzás:** Az olvashatóság és a dokumentum struktúrájának, könnyebb értelmezhetőségének érdekében.

Egy dokumentum **nyomtatása előtt** fontos ellenőrizni:

- a margók és az elrendezés helyességét
- a fejlécek, élőfejek és élőlábak, valamint az oldalszámozás megfelelő beállítását
- a dokumentum színes vagy monokróm nyomtatási beállításait
- a nyomtatási előnézet segítségével a végleges kinézetet

A legtöbb szövegszerkesztő lehetőséget ad a dokumentum **közvetlen exportálására PDF formátumba**. Ennek a formátumnak az előnyei:

- **platformfüggetlenség:** minden operációs rendszeren és eszközön azonos formában jelenik meg
- **formázás megőrzése:** a dokumentum kinézete nem változik meg a megnyitás során
- **védelem:** lehetőség van a jelszavas védelemre az illetéktelen megtekintés elkerülése érdekében, valamint a PDF-ek általánosságban nem szerkeszthetők
- **nyomtathatóság:** nyomtatáskor az eredeti formázási beállítások sértetlenek maradnak
- **kompakt méret:** a PDF dokumentumok tömöríthetőek, így könnyebb őket tárolni és megosztani.

5. tétel: Kördokumentum

Mit nevezünk kördokumentumnak? Milyen formáit ismered? Ismertesd egy kördokumentum összetevőit! Hogyan készítünk el egy kördokumentumot? Készíts el egy kördokumentumot! (Kapsz laptopot és forrást hozzá.)

Kördokumentumnak azokat a dokumentumokat nevezzük, amelyek tartalma részben állandó, részben pedig változó adatokból épül fel. Főként tömeges levélküldéshez, címkék vagy egyedi űrlapok létrehozásához használják. Például egy iskolai értesítő levélben minden címzett neve, címe és egyéb személyes adatai eltérnek, míg a levél tartalma azonos marad.

Egy kördokumentumnak többféle formája lehet, pl.:

- **levelek** (pl. értesítések, meghívók, pályázati értesítések)
- **borítékok** (a címzettek adatai automatikusan rákerülnek)
- **címkék** (pl. postai címkék, névjegyek, vagy akár árucikkek címkéi)
- **adatlapok és jelentések** (pl. számlák, egyedi ügyfél-adatlapok)

egy kördokumentum **összetevői** a következők:

- **(fő)törzs-dokumentum:** ez tartalmazza a szöveg állandó részét (pl. egy levél sablonja)
- **adatforrás:** Ez tartalmazza a változó adatokat (ez lehet egy excel tábla, adatbázis, vagy rögzített szélességű adatokat tartalmazó szövegfájl)
- **összekapcsolás és mezők:** a változó adatok helyét a főtörzs-dokumentumban ún. mezők jelölik, pl. {címzett neve}

egy kördokumentum létrehozásának **főbb lépései** MS Word használatával:

1. **Dokumentum előkészítése**
 - a. egy új, vagy meglévő dokumentumot hozunk létre / nyitunk meg, amely a törzsszöveget fogja adni
2. **adatforrás létrehozása vagy kiválasztása**
 - a. használhatunk excel-táblázatot, access-adatbázist, vagy rögzített szélességű adatokat tartalmazó szövegfájlt
3. **mezők beszúrása**
 - a. a levelezés fülön a *körlevél* funkció használatával a kívánt helyen a dokumentumban válasszuk a *mező beszúrása* funkciót
4. **ellenőrzés**
 - a. a *körlevél előnézete* segítségével meg lehet bizonyosodni az adatok helyességéről, valamint, hogy a megfelelő helyen jelennek-e meg megfelelő formában az adatok
5. **a dokumentum létrehozása**
 - a. a *befejezés és egyesítés* opcióval elkészítjük és kinyomtathatjuk, vagy elküldhetjük e-mailben a dokumentumokat

Példa egy kördokumentumra itt érhető el

ehhez természetesen szükséges az akár krétából lekérhető adatbázis / táblázat.

6. tétel: Nagy dokumentumok formázása

Mit nevezünk stílusnak? Hogyan használjuk őket? Hogyan kapcsolódnak a tartalomjegyzékhez? Mit tudsz az élőfejről, élőlábról? Mondj gyakorlati példákat alkalmazásukra! Milyen speciális beállításokat lehet beállítani élőfejre élőlábra vonatkozóan? Mi a különbség a lábjegyzet és a végjegyzet között? Mit nevezünk szakasznak?

A stílus a szövegszerkesztésben egy előre beállított formázási beállítások összessége, amely lehetővé teszi (nagyobb) dokumentumok egységes megjelenését. Egy stílus tartalmazhat:

- Betűtípust, betűméretet
- Betűszínt, kiemeléseket (félkövér, aláhúzott, dőlt stb.)
- Behúzásokat, térközöket
- Bekezdésformázásokat (sorköz, igazítás, tabulátorok stb.)
- Számozást, felsorolást

A stílusokat előre beállíthatjuk a szövegszerkesztőben, amelyeknek használatának előnyei az **egységes formázás, a gyorsabb szerkesztés, vagy a tartalomjegyzék generálása**

A szövegszerkesztőben beépített stílusokat is találunk, de egyéni beállításokat is létre tudunk hozni, például:

- címekhez
- törzsszövegekhez
- kiemelésekhez

Amennyiben a dokumentumban a címekhez megfelelő stílusokat használunk, tartalomjegyzés automatikusan generálhatunk, mert a szövegszerkesztő ezek alapján hozza létre a **tartalomjegyzéket** ez

- automatikusan frissíthető
- kattintható hivatkozásokat is tartalmazhat
- segíti a dokumentum szerkezetének áttekintését

Az élőfej és az élőláb a dokumentumok speciális területei, amely minden oldalon megjelennek és általában állandó információkat tartalmaznak. az **élőfej** az oldal felső részén, amíg az **élőláb** az oldal alsó részén található

az **élőfej** tartalmazhatja pl. a dokumentum címét, a fejezetcímet, vagy a szerző nevét

az **élőlábban** hasznos lehet elhelyezni az oldalszámot, dátumot, vagy az adott cég logóját

speciális beállítások az élőfejre és élőlábra vonatkozóan:

- **eltérő első oldal:** témamegjelölő; pl. egy dolgozat első oldalán nem szerepel az oldalszám
- **eltérő páros és páratlan oldalak:** könyveknél gyakori beállítás, hogy a kétoldalas formátumnál pl. az oldalszám az oldal külső részén legyen látható
- **szakaszonként eltérő élőfej/élőláb**

Mind a **lábjegyzet** és a **végjegyzet** magyarázó, vagy hivatkozási célt szolgál, de eltérő helyen jelennek meg. A **lábjegyzet** az adott oldal alján, a **végjegyzet** pedig a dokumentum vagy az adott fejezet végén szerepel.

Tudományos dolgozatokban a forrásokat a lábjegyzetben tüntetünk fel, amíg könyvekben pl. pedig a főszöveg gondolatmenetét követhetőbbé, töretlenebbé teszi a végjegyzet. Tartalmaz mindazt, ami a főszöveg gondolatmenetéhez kapcsolható és nincs okvetlenül szükség elhelyezni a főszövegben.

Egy **szakasz** a dokumentum egy olyan része, amelyben eltérő formázási beállítások lehetnek. Minden új szakasz egy szakasztöréssel hozható létre (az elrendezés fülön a szakasztörésre kattintva).

szakaszokat használhatunk

- **különböző élőfejek és élőlábak beállítása** egyes dokumentumrészekhez
- **eltérő oldaltájéolás** beállításához
- **eltérő oszlopbeállításokhoz** (pl. újságszerű elrendezéshez)

II. Dokumentumkészítés – Publikálás a világhálón

7. tétel: Betű és bekezdésformázások

Milyen hasznos funkciói vannak egy kódszerkesztőnek? Ismertesd a HTML5 alapstruktúráját! Milyen betűformázáshoz kapcsolódó tageket ismersz? Mutasd be a <p>, <h1>, <h2>... tagek használatát, sajátosságait! Milyen listákat szűrhatunk be?

Html (HyperText Markup Language), vagy más egyéb kódnyelv használatánál sok hasznos funkciót kínálnak a kódszerkesztők (IDE, vagy a legszéleskörűbben használható VSCode), amelyek megkönnyítik a kódolást és a hibakeresést. Néhány fontos funkció:

- **szintaxiskiemelés:** a különböző nyelvi elemek (parancsok, változók, tagek stb.) különböző színű megjelenítése nagyban megkönnyíti a kód olvasását
- **automatikus kiegészítés:** segít a parancsok és tagek gyorsabb beírásában, valamint, hogy a bezárandó tagek be is legyenek zárva
- **hibajelzés:** felismeri a szintaktikai hibákat és figyelmeztet rájuk (pl. aláhúzással, vagy szöveges figyelmeztetéssel)
- **verziókezelés támogatása:** Git és más rendszerek integrálása
- **beépített terminál:** parancssoros műveletek elvégzésére
- **böngésző-előnézet:** bizonyos kódszerkesztők képesek a HTML-kódok azonnali megtekintésére a böngészőben

a HTML5 alapstruktúrája a következő:

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document Name</title>
</head>
<body>
  <h1>Main Title</h1>
  <p>paragraph</p>
</body>
</html>
```

<!DOCTYPE html>: meghatározza, hogy a dokumentum HTML5 szabványt használ

<html>: a teljes dokumentum gyökere

<head>: metaadatokat és hivatkozásokat tartalmaz

<meta charset="UTF-8">: meghatározza a karakterkódolást

<title>: a böngésző címsorában megjelenő szöveg

<body>: a megjelenítendő tartalom helye

a betűformázáshoz használt HTML tagek a következők:

- ****: félkövér szöveg
- ****: kiemelten fontos szöveg, általában félkövéren jelenik meg
- **<i>**: dőlt betűs szöveg
- ****: kiemelt szöveg, általában dőlt betűvel jelenik meg
- **<u>**: aláhúzott szöveg
- **<mark>**: kiemelt szöveg, általában sárga háttérrel jelenik meg
- **<small>**: kisebb betűméret
- ****: áthúzott, törölt szövegre utal
- **<ins>**: beszúrásként jelölt szöveg

a **<p>** tag egy bekezdést jelöl, ami új sorban jelenik meg, példa:

```
<p>Ez egy bekezdés</p>
```

a **<h1>** - **<h6>** tagek címsorokat jelölnek (heading), melyek hierarchikus rendben Vannak, értelemszerűen a **<h1>** a legnagyobb és a **<h6>** pedig a legkisebb.

ezek a tartalom strukturálása mellett a keresőoptimalizálást is segítik

HTML-ben háromféle listatípust használhatunk:

- rendezetlen lista (****) felsorolás szimbólumokkal
- rendezett lista (****), amely számozott, vagy betűzött felsorolás
- definíciós lista (**<dl>**), ami fogalmak és meghatározások párosítására szolgál, például:

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
</dl>
```

a felsorolás szimbólum tag-je : ****

8. tétel: Objektumok beszúrása

Milyen paraméterek adhatók meg képek beszúrásánál? Hogyan szűrhatunk be hivatkozásokat? Mit tudsz az oldalon belüli hivatkozásokról? Miben különbözik a képek beszúrása a videók és hangok beszúrásától? Mutasd be röviden a táblázatok struktúráját!

Képek beszúrásához az tag használható, amely több paramétert is elfogad, ezek a következők:

- **src:** a kép forrását (elérési útját) adja meg, ami lehet helyi link, helyi tárhely, vagy külső internetes link
- **alt:** alternatív szöveg a kép leírására, ami keresésoptimalizálás és akadálymentesség szempontjából fontos, valamint, ha a kép valamilyen okból nem tud betölteni
- **width és height:** a kép szélessége és magassága pixelben, vagy százalékban megadva
- **title:** lebegő szöveg, amely az egérkurzor a kép fölé helyezésekor jelenik meg
- **loading:** meghatározza a kép betöltését (lazy a késleltetett betöltéshez)

Hivatkozásokat a <a> tag segítségével tudunk beszúrni, például:

```
<a href="https://www.pelda.hu" target="_blank"><Kattints ide</a>
```

href: a cél URL vagy fájl elérési útja

target="_blank": Új ablakban vagy fülön nyitja meg a hivatkozást

title: magyarázó szöveg, ami az egér föléhúzásakor jelenik meg

Az oldalon belüli hivatkozáshoz az ankerek használhatók

pl:

```
<a href="#szakasz">Ugras a szakaszhoz</a>
```

```
<h2 id="szakasz"> Ez a kivant szakasz</h2>
```

id: egyedi azonosítót ad az elemnek

href="#id_neve": az adott elemhez ugrik az oldalon belül

A képek beszúrására a , míg a **videókra a <video>, hangokra az <audio>** tagek használhatóak

```
<video width="400" height="240" autoplay muted>  
    <source src="video.mp4" type="video/mp4">  
    <source src="video.ogg" type="video/ogg">
```

A böngésző nem támogatja a videólejátszást

```
</video>
```

Ebben a két esetben több forrást is megadhatunk (<source>). Ez abban az esetben hasznos, ha egy böngésző nem támogat egy adott fájltypust, az azonos médiatartalmat lehet meg tudja jeleníteni egy másik formátumban.

Videóknál és hangoknál használhatók vezérlőelemek (controls) is

videóknál beállítható automatikus és néma lejátszás (autoplay, muted) (a chromium alapú böngészők az automata lejátszást csak némított módban engedélyezik)

A HTML táblázatok a <table> taggel hozhatók létre

```
<table border="1">  
    <thead>  
        <tr>  
            <th>Fejlec 1</th>  
            <th>Fejlec2</th>  
        </tr>  
    </thead>  
    <tbody>  
        <tr>  
            <td>Adat 1</td>  
            <td>Adat 2</td>  
        </tr>  
    </tbody>  
</table>
```

<table>: a táblázat maga

<thead>: table head, vagyis a táblázat fejléc része

<tbody>: a táblázat törzse

<tr>: a táblázat egy sora

<th>: fejléc cella

<td>: normál cella

9. tétel: Stíluslapok használata

Mit nevezünk CSS-szabálynak? Mondj példákat a használatára! Hogyan adhatunk meg CSS szabályokat több deklarációval? Milyen betűformázó CSS deklarációkat ismersz? Milyen bekezdésformázó CSS deklarációt ismersz? Mit tudsz az osztály, illetve azonosító kijelölőkről? Hogyan csatolhatunk stíluslapot HTML dokumentumunkhoz?

A **CSS-szabály** egy olyan előírás, amely meghatározza, hogy egy adott HTML elem milyen módon jelenjen meg a böngészőben. Egy CSS-szabály általában két részből áll:

- **szelektor**: meghatározza, melyik elem(ek)re vonatkozik a szabály
- **tulajdonság**: meghatározzák az adott elem megjelenését

példa egy CSS-szabályra:

```
body {  
    color: cyan;  
}  
h1 {  
    font-size: 25px;  
}  
p {  
    text-align: center;  
    color: black;  
    line-height: 1.5;  
}
```

Egy CSS-szabály egyszerre több tulajdonságot is tartalmazhat, az alsó, az összes bekezdésre vonatkozó deklarációhoz hasonló módon.

Ennek megfelelően az összes <p> taggel rendelkező bekezdés középre zárt, fekete betűtípusú szöveggel és 1,5-ös sorközzel fog megjelenni.

a (leg)hasznosabb betűtípusformázó CSS deklarációk a következők:

- color: betűszín megadása
- font-size: betűméret
- font-family: betűtípus kiválasztása
- font-weight: betűvastagság (lehet normal, bold, light, bolder)
- text-decoration: szövegformázás (pl. underline, overline, line-through)

- letter-spacing: betűközök beállítása
- word-spacing: szavak közötti térköz beállítása

bekezdésformáló CSS deklarációk:

- text-align: szöveg igazítása (left, center, right, justify)
- line-height: sorköz
- text-indent: bekezdés behúzása
- margin: külső térköz
- padding: belső térköz

osztály (class) kijelölővel több elemhez is hozzárendelhetünk különböző deklarációkat, ezeket ponttal jelöljük

pl.

```
.kiemelt {  
    color: red;  
    font-weight: bold;  
}
```

a HTML kódban így lehet rá hivatkozni:

```
<p class="kiemelt">Ez itt egy kiemelt bekezdés volna.</p>
```

Az azonosítók (id) egyedi elemek formázásához használatosak és kettőskeresztrel kell őket jelölni.

pl.

```
#focim {  
    color: red;  
    font-weight: bold;  
}
```

HTML kódban:

```
<h1 id="focim">Ez az oldal focime></h1>
```

fontos, hogy osztályokat többször is lehet használni egy oldalon, de azonosítókat csak egyszer

Stíluslapot háromféleképpen csatolhatunk egy HTML dokumentumhoz:

- Külső CSS fájl használata

head tagen belül

```
<link rel="stylesheet" href="patkostilus.css">
```

- belső CSS a <style> tagben

```
<style>
```

```
body{
```

```
    background-color: black;
```

```
}
```

```
</style>
```

- inline egy elemhez

```
<p style="color: blue; font-size: 18px; font-family: serif;">Ez egy bekezdés szövege</p>
```

ezek közül az első az ajánlott, mert ez könnyebben kezelhető, valamint újrahasználható, ha egy összetett weblapnak egységes megjelenést akarunk biztosítani

10. tétel: Dobozmodell

Ismertesd a CSS dobozmodelljét! Milyen szegélybeállításokat ismersz? Hogyan, milyen módokon adhatjuk meg objektumok méreteit? Hogyan igazíthatunk blokkokon belül, illetve blokkokat? Hogyan állíthatunk be „dobozunknak” háttérképet? Mit tudsz egy weboldal szabványosságáról, illetve publikálásáról

A **CSS dobozmodell (box model)** a HTML elemek megjelenésének alapelve

egy ilyen „doboz” részei a következők:

- **tartalom (content)**
 - az elem tényleges tartalma (szöveg, kép, link, videó stb.)
- **belső térköz (padding)**
 - a tartalom és a szegély közötti tér
- **szegély (border)**
 - az elem határvonala, ez lehet látható és láthatatlan is
- **külső térköz (margin)**
 - jelenlegi és a többi elem közötti távolság

példa a dobozmodell beállítására:

```
div{  
    width: 200px;  
    height: 100px;  
    padding: 10px;  
    border: 2px solid black;  
    margin: 20px;  
}
```

egy **objektum méretét** a következő módokon adhatjuk meg:

- Fix méret (pl. px, cm)
- relatív méret (pl. %)
- automatikus méret (auto)

blokkokon belül (szöveget) a text-align és a vertical-align tulajdonságokkal igazíthatunk

blokkokat vízszintesen a margin tulajdonsággal (középre igazításkor: margin: 0 auto;)

A flexboxot a pontos igazításhoz tudjuk használni
pl.

display: flex;

justify-content: center; (vízszintesen középre igazít)

align-items: center; (függőlegesen középre igazít)

A **background-image** segítségével beállíthatunk háttérképet is:

```
div{  
    background-image: url('fajlnev.jpg')  
    background-size: cover;  
    background-repeat: no-repeat;  
    background-position: center;  
}
```

Egy weboldalnak meg kell felelnie a W3C (World Wide Web Consortium) ajánlásainak, annak érdekében, hogy megfelelően jelenjen meg minden böngészőben. Ezt könnyen ellenőrizhetjük a W3C validator segítségével.

III. Dokumentumkészítés – Bemutatókészítés

11. tétel: A bemutatókészítés alapjai

Mit nevezünk bemutatónak? Milyen érveket tudsz felhozni a bemutató használata mellett, illetve ellen? Mik egy előadáshoz készült bemutató elkészítésének lépései? Miben különböznek az előadásra készült bemutatók az egyéni feldolgozásra készült bemutatóktól?

A **bemutató** egy vizuális segédanyag, amelyet előadások, oktatás, vagy prezentációk támogatására használnak. Ez leggyakrabban diavetítés formájában történik, ahol a diákon különböző médiaelemek (szöveg, képek, videók, grafikonok, animációk) jelennek meg.

A bemutató használata mellett számos érv szól, például:

- vizuális támogatás: segíti a közönség figyelmének fenntartását, valamint a mondanivalónk könnyebb megértését
- strukturált információ: könnyebb követni (az előadónak és a nézőnek egyaránt) az előadás főbb részeit
- multimédiás lehetőségek: képek, grafikák és videók használatával gördülékenyen és érthetően lehet bemutatni egy bonyolultabb témát is

Hátrányként felhozható a technikai háttér, ami nem minden esetben adott, vagy megbízható; valamint egy rosszul összeállított bemutató (pl. túl sok szöveg, átláthatatlan struktúra) pont a fentebb említettek ellentétjét tudják kiváltani.

egy bemutató készítésének **főbb lépései** a következők:

1. a téma és a cél meghatározása (miről szeretnénk beszélni, milyen részletesen, **kiknek szól a bemutató**, utóbbi alapján milyen nyelvezetet érdemes használni)
2. vázlat készítése (a tartalom főbb pontjainak, azoknak a struktúrájának meghatározása)
3. dizájn és elrendezés kiválasztása (egységes színséma és betűtípusok használata, amelyhez a MS PowerPoint beépített témái nagy segítséget nyújthatnak)
4. diák létrehozása (szövegek, képek, videók, grafikonok elhelyezése)
5. animációk és áttűnések beállítása (, amelynek segítségével az előadásunkat dinamikusabbá, követhetőbbé tudjuk tenni)
6. próba és ellenőrzés (az előadás levetítése az elejétől a végéig, az esetleges hibák kiküszöbölésének érdekében)

7. az előadás megtartása

Az **előadásra készült bemutatók** az előadó támogatására szolgálnak, míg az egyéni feldolgozásra készültekre jellemzőbb a részletesebb információ, az önálló tanulás elősegítésének érdekében

(inkább) jellemző	Előadásra készült bemutató	Egyéni feldolgozásra készült bemutató
Szövegmeny-nyiség	keves szöveg, inkább kulcsszavak és vázlatpontok a könnyebb követhetőségért	Részletesebb, kifejtettebb, jegyzetszerű magyarázatok
vizualitás	Nagyobb betűméret, letisztult és egyszerű(bb) megjelenés	Több információ, részletes diák
interaktivitás	Az előadó magyarázatát kíséri	Több önálló, interaktív elem, például hivatkozások
célja	a hallgatóság figyelmének irányítása, fenntartása	részletes információátadás

12. tétel: A bemutatókészítéshez használható eszközök

Milyen vizuális eszközöket használhatunk a bemutatónk szerkezetének kialakítására? Milyen hasonlóságok, illetve különbségek vannak szövegszerkesztőbe beszúrható objektumok és bemutatókészítőben beszúrható objektumok között? Sorolj fel példákat! Mit tudsz az áttűnésekről, illetve az animációkról?

A bemutató vizuális szerkezetének kialakítására számos eszköz a rendelkezésünkre áll:

- dizájn és sablonok (a Microsoft és más felhasználók által kiadott sablonok, melyek egységes megjelenést biztosítanak)
- diacímek és alcímek (segítenek a tartalom tagolásában, egy hosszabb téma felosztásának követésében)
- színek és betűtípusok (a szövegszerkesztőkhöz hasonló formázási beállításokkal, pl. kontrasztos színekkel és könnyen olvasható betűtípusokkal (NEM comic sans) nagyban javítható a bemutató érthetősége)
- képek és diagramok (akár saját eszközről, akár az internetről beszúrható vizuális információk kísérhetik, alátámaszthatják, illusztrálhatják mondanivalóinkat)
- felsorolások és számozások (a lényeges pontokat kiemelik, egy rendezett és könnyen érthető képet tudnak adni a nézőnek)
- táblázatok
- áttűnések és animációk (elősegítik a figyelem irányítását)

a szövegszerkesztőben és a bemutatókészítőben beszúrható objektumok összehasonlítása:

Jellemző	Szövegszerkesztő	Bemutatókészítő
Szövegdoboz	igen	igen
Képek	igen	igen
Táblázatok	igen	igen
Diagramok	igen	igen
Hiperhivatkozások	igen	igen
Videók és hangok	ritkán	gyakran
Animációk	nem	igen
Áttűnések	nem	igen

Az **áttűnések** és az **animációk** használata segíti a közönség figyelmének fenntartását és irányítását, de túlzott és nem megfelelő stílusú alkalmazásuk zavaró lehet, ezért alaposan meg kell választani az alkalmazásuk módját.

Az **áttűnések** a következő diára való átmenet módját határozzák meg, segítik az előadás folyamatosságát

például: elhalványulás, tolás, forgatás, origami

Az **animációk** egyes objektumok mozgását vagy megjelenési módját határozzák meg. Ezeket lehet időzíteni is, de kattintásra is jöhetnek egymás után, valamint csoportonként egyszerre több objektumot is meg lehet jeleníteni.

például: szöveg beúsztatása, kép nagyítása, elem eltüntetése

fajtái:

- belépő animációk (fade in, zoom in)
- kiemelő animációk (villogás, színváltás)
- kilépő animációk (fade out, eltűnés, kiforgás)
- mozgásvonalak (egy adott objektum mozgatása egy előre meghatározott úton)

IV. Dokumentumkészítés – Pixelgrafika

13. tétel: A pixelgrafikus ábrázolás alapjai

Mit nevezünk pixelgrafikus képnek? Mit tudsz a színkódokról, illetve a színmélységről? Milyen pixelgrafikus szerkesztőket ismersz? Milyen pixelgrafikus fájlformátumokat ismersz? Mutasd is be őket! Milyen esetekben használhatunk pixelgrafikus szerkesztőprogramot?

Képszerkesztés során két fő grafikus módot különböztetünk meg, a **pixelgrafikus (rastergrafikus)** és a **vektorgrafikus** (objektumalapú) képek. A vektorgrafikáról most nincs szó, a pixelgrafikus képek pixelekből, vagyis kis (egyszínű) pontokból állnak, amelyek együttesen alkotnak egy egész képet. Általános felhasználásnál ez az elterjedt, a digitális fényképezés is ilyen módon működik. Ennek hátránya, hogy bizonyos szintű nagyítást követően láthatóvá válnak a pixelek, így „szétesik” a kép, valamint a magasabb felbontású képek nagy tárhelyigénnyel bírnak.

A digitális képek színeit különböző módokon lehet kódolni, ezeknek mind megvannak az előnyeik, hátrányaik és az ezekből adódó felhasználási területeik. A leggyakoribb színkódolási rendszerek:

- **RGB** (Red, Green, Blue) – a képernyőkön használt additív színmodell. A három „alapszín” mennyiségének megadásával lehet szinte bármilyen színt „kikeverni”, a három szín egymástól függetlenül intenzitásukat növelve lehet 0 és 255 között. Így ezzel a színkódolással összesen 255^3 , vagyis kb. 16,5 millió különböző szín adható meg
- **CMYK** (Cyan, Magenta, Yellow, Key/Black) – ez a nyomdai színkeverés alapja, a legtöbb színes nyomtató is ezekkel dolgozik, sokszor ezek a patronok külön-külön megvásárolandók
- **HEX kódok** – Ez a webdesignban használatos leginkább, hatjegyű karaktersorozattal le lehet megadni a színeket. Az RGB alapszínek mindegyikéhez két karakter tartozik, melyek intenzitásukat szabályozzák, ez 00-tól FF-ig terjed, így egy sokkal részletesebb színskálán tudunk mozogni. Például egy szimpatikus szín: **#f2c8b3**
- **HSL/HSV** (Hue, Saturation, Lightness) – a színeket árnyalat, telítettség és világosság alapján határoz meg. Ez egy intuitív, felhasználóbarát formázási lehetőség

A képszerkesztő programokat számos célra használhatjuk, ezek közül például:

- **fotók restaurálása** – képek javítása, színek módosítása, nemkívánatos elemek eltűntetése
- **grafikai tervezés** – logók, bannerek, illusztrációk készítése
- **kollázsok és montázsok létrehozása** – több kép kombinálása, összeállítása
- **digitális festészet** – kézi rajtolás és festés digitális eszközökkel, amely számos előnyt ad a „hagyományos” képzőművészettel szemben
- **webdesign** – gombok, ikonok és más vizuális elemek megtervezése és elkészítése weboldalakhoz

A legelterjedtebb képszerkesztő program évtizedek óta az **Adobe Photoshop**, viszont ez kifejezetten magas árcédulával rendelkezik. Szerencsére sok ingyenes opció is rendelkezésre áll, ezek közül a legtöbbet használt a **GIMP**

A **GIMP** (GNU Image Manipulation Program) egy nyílt forráskódú képszerkesztő program, amely rengeteg (Photoshopot megközelítő) funkciót kínál képek szerkesztésére és manipulálására. Amatőr grafikusok és fotósok körében kifejezetten elterjedt, valamint közoktatásban is legtöbbször ennek segítségével oktatnak.

főbb jellemzői:

- rétegek kezelése – több réteg egyidejű szerkesztése, maszkolási lehetőségek
- ecsetek és szűrők – számos beépített eszköz áll rendelkezésre a képek módosítására
- színeskorrekció – kontraszt, fényerő, színtelítettség (szaturáltság) módosítása
- sok elérhető, közösségi fejlesztésű plugin – további funkciók tölthetők le és építhetők be
- széleskörű fájlformátum támogatás – pl. JPG, GIF, PSD (photoshop formátum)

A gimp alapvető eszközei és funkciói:

- kijelölő eszközök – téglalap, ellipszis, szabadkézi, valamint „varázspálca”, amely egybefüggő elemeket jelöl ki
- festőeszközök – ceruza, ecset, festékvödör, radír
- szöveges eszköz – betűtípusok és szövegformázás beállítása
- rétegkezelés – rétegek átlátszóságának és sorrendjének módosítása
- szűrők és effektek – elmosás, élesítése, torzítás stb.

A gimp számos fájlformátumot támogat, például:

- PNG – átlátszóságot támogató fájlformátum (weboldal elemeinél hasznos pl.)
- JPG – tömörített formátum, amely főként webes és fotós célokra hasznos
- GIF – animált és egyszerűbb grafikákhoz ideális
- XCF – a GIMP saját formátuma, amely megőrzi a beállított rétegeket és újbóli, akár más eszközön való megnyitáskor is szerkeszthető marad
- PSD – Az Adobe Photoshoppal kompatibilis fájlformátum

14. tétel: Pixelgrafikus szerkesztés

Mutasd be egy pixelgrafikus szerkesztőprogram eszközeit! Milyen csoportba sorolhatod őket? Szemléltesd néhány eszköz használatát! Mit tudsz a rétegekről? Mire jó az alfa-csatorna? Ismertesd egy szöveg elhelyezésének módjait, lehetséges beállításait!

Pixelgrafikus szerkesztőprogramok széleskörű eszköztárral rendelkeznek, amelyek a képek módosítására és szerkesztésére nyújtanak lehetőséget.

Ezeket az eszközöket általában az **alábbi fő csoportokba soroljuk:**

- **kijelölő eszközök**
 - téglalap és ellipszis kijelölés
 - szabadkézi és „varázspálca” kijelölés
 - gyorsmaszk mód a pontosabb kijelölésekhez
- **festő- és rajzeszközök**
 - ecset és ceruza (kézi rajzeszközök)
 - festékszóró (airbrush)
 - kitöltő eszköz (festékvödör)
 - színátmenet eszköz (gradients)
- **szövegkezelő eszközök**
 - szöveg(doboz) elhelyezése és szerkesztése
 - betűtípus, méret, szín és igazítás beállítása
- **transzformációs eszközök**
 - áthelyezés
 - forgatás, tükrözés és méretezés
 - perspektíva és torzítás
- **restauráló eszközök**
 - korrektor
 - klozett (másolás egy adott területről)
 - elmosás és élesítés
- **rétegkezelő eszközök**
 - új rétegek létrehozása
 - átlátszóság és keverési módok beállítása
 - rétegek csoportosítása és maszkolása

V. Dokumentumkészítés – Vektorgrafika

15. tétel: A vektorgrafikus ábrázolás alapjai

Mi a vektorgrafikus ábrázolás központi fogalma? Jellemezd a tulajdonságait! Miben különbözik a vektorgrafikus tárolás a rastergrafikus tárolástól? Milyen vektorgrafikus szerkesztőprogramokat ismersz? Mutasd be az előnyét, illetve hátrányát az önálló vektorgrafikai szerkesztőprogramoknak az irodai programokhoz képest!

A **vektorgrafika** az a számítógépes-grafikai eljárás, melynek során *geometriai primitíveket*, vagyis rajzelemeket, pl. pontokat, egyeneseket, görbéket és sokszögeket, röviden **objektumokat** használnak képek leírására, megjelenítésére.

Ezek az objektumok az alábbi tulajdonságok alapján vannak meghatározva:

- koordináták
- színek
- vonalvastagság

a **vektorgrafika** alapvető tulajdonságai a következők:

- **felbontásfüggetlen** (tehát bármekkora nagyítás mellett sem lesz pixeles a kép)
- **könnyen módosítható**
- **nem használható fotókhoz**, hiszen nem tud természetes, részletgazdag képeket kezelni
- **logókhoz, ikonokhoz, műszaki rajzokhoz, vagy térképekhez** cserébe tökéletes

Tulajdonság	Vektorgrafika	Raster/pixelgrafika
tárolás módja	matematikai alakzatok	képpontok (pixelek)
nagyítás	minőségromlás nélkül	minőségromlással
fájlméret	általában kisebb	általában nagyobb
használat	logók, diagramok, illusztrációk	fotók, digitális rajzok, részletgazdag képek
fájlformátum	.SVG	.JPG .PNG

Az egyik **legismertebb, ingyenes** vektorgrafikus szerkesztőprogram az **Inkscape**, amely nyílt forráskódú és kifejezetten erős eszköztárral rendelkezik. A professzionális szcénában a leggyakrabban használt, de kifejezetten drága opció az **Adobe Illustrator**.

Az **önálló vektorgrafikai szerkesztők**, mint pl. a fentebb említett Inkscape sokkal precízebb munkát tesz lehetővé az irodai programokhoz képest, amelyek rendelkeznek valamilyen szintű vektorgrafikai lehetőségekkel, mint pl. a MS Word, vagy PowerPoint. Amíg

utóbbiak sokkal elérhetőbbek és általánosságban egyszerűbb a használatuk, funkcionálisuk lényegesen korlátozottabb, nem tesznek lehetővé pontos grafikai munkát.

16. tétel: Vektorgrafikus szerkesztés

Mutasd be egy vektorgrafikus szerkesztőprogram felhasználói felületét! Hogyan kategorizálnád a szerkesztőprogram eszközeit? Szemléltesd néhány eszköz használatát! Mit tudsz a klónok használatáról? Ismertesd az unió, metszet, különbség műveleteket! Mit tudsz az útvonalakról?

Az **Inkscape** UI-nak a legfőbb részei a következők:

- **eszköztár** (bal oldalon)
- **tulajdonságsáv** (felül)
- **színpaletta** (lenn)
- **réteg- és objektumkezelő** (jobb oldalt, vagy külön ablakban)
- **munkaterület** (középen)

Különböző rajzeszközök:

- **rajzoló eszközök**
 - egyenes vonal
 - toll, ceruza, görbék
 - alakzatok: négyzet, kör, sokszög
- **szerkesztő eszközök**
 - kiválasztó (select tool) – F1 billentyű
 - útvonal csomópont-szerkesztő – F2 billentyű
 - szöveg, kitöltés, színátmenet
- **transzformáló eszközök**
 - mozgatás, elforítás, tükrözés, méretezés
- **speciális/egyéb eszközök**
 - klónozás
 - útvonalműveletek (unió, metszet, különbség)
 - effektek, szűrők

néhány eszköz használatának bemutatása:

négyzet eszköz - kattintás – húzás -> négyzet (ha Shift + húzást alkalmazunk, a középpontból nő)

színtöltés – a színpalettán kiválasztott színre változik a kijelölt objektum színe

forgatás – az objektum kijelölését követően forgatási fogók jelennek meg

node tool (útvonal csomópont-szerkesztő) – útvonal csomópontjait lehet vele mozgatni, görbíteni

A **klón** egy olyan másolat, amely **folyamatosan követi az eredeti módosításait**. Tehát az eredeti objektum bármilyen attribútumának módosítását követően a klón is automatikusan frissül.

használni a következőképpen lehet: objektum kijelölése → klónozás → klón létrehozása

klónokat lehet külön elhelyezni, forgatni, de a formájuk egyesével nem módosítható

mintákhoz és ismétlődő elemekhez (pl. virágszirom, mandala, ikonrács) hasznos

„Bool”műveletek – unió, metszet, különbség

Ezek a **logikai műveletek** több alakzat összevonására használhatók, az **útvonalak** → **bools műveletek** menüpontban érhetők el.

példák:

- **unió** – két vagy több alakzathoz egy új, egybefüggő formát csinál.
- **metszet** – csak azt a részt hagyja meg, ahol az alakzatok fedik egymást
- **különbség** – az elsőből kivágja a másodikat

útvonalak:

az útvonal az objektumok pontjaiból (**csomópontokból**) és az azok közti **szakaszokból** áll.

A legegyszerűbb útvonal **egy egyenes/vonal**, a legbonyolultabb egy görbe, amely bármilyen formát felvehet.

Az **alakzatok is átalakíthatók útvonallá**, így szabadon szerkeszthetőkké válnak.

VI. Táblázatkezelés

17. tétel: A táblázatkezelés alapjai

Mit nevezünk adatnak? Milyen adattípusokat ismersz? Mit tudsz a dátum és idő adatokról? Mit nevezünk cellahivatkozásnak, milyen fajtái vannak? Hogyan tudunk kifejezéseket összeállítani? Milyen formázási beállításokat ismersz táblázatkezelőben? Mire jók a diagramok, illetve milyen diagramfajtákat ismersz? Jellemezd őket röviden!

A számítástechnikában **adatnak** nevezzük azokat a számokkal leírható dolgokat, melyek számítástechnikai eszközökkel rögzíthetők, feldolgozhatóak és megjeleníthetők. **Táblázatkezelés** esetében az adat minden olyan információ, amit egy **cellában** tárolunk. Ez lehet szöveg, szám, dátum, logikai érték, függvény vagy képlet. A táblázatkezelő az adatokkal műveleteket végezhet, pl. számolás, szűrés, rendezés stb.

Az egyik legnépszerűbb táblázatkezelő, az **Excel** a következő adatokat ismeri fel:

- szám
- szöveg
- dátum – valójában egy szám, a napok száma 1900.01.01-től
- idő – valójában egy törtszám, pl. 12:00 = 0.5 (fél nap)
- logikai érték (boolean)
- képlet

a **dátum és idő** adattípusok számbeli mivoltának köszönhetően formázhatók és viszonylag könnyedén számolni is lehet velük

A **cellahivatkozás** az az eset, amikor egy képlet egy másik cellára hivatkozik

Típus	példa	jelentés
Relatív	=A1	mindig az aktuális pozícióhoz viszonyít
Abszolút	=\$A\$1	mindig pontosan arra a cellára mutat
Vegyes	=\$A1 vagy =A\$1	csak az oszlop, vagy a sor rögzített

tehát ha az =A1*B1 képletet lemásoljuk jobbra, abból a következő cellában =B1*C1 lesz, de ha =\$A\$1*B1, akkor az A1 mindig fix marad, tehát a jobbra levő cellában a következő képlet lesz: =\$A\$1*C1

Kifejezések/képletek összeállítása

a képleteket az Excel onnan ismeri fel, hogy mindig = jellel kezdődnek

lehetnek benne számok, hivatkozások, műveleti jelek és függvények

Pl: =INDEX(\$T\$1:\$T\$5;HOL.VAN(D3;\$S\$1:\$S\$5;1))

néhány gyakori **függvény**:

- **SZUM** – összeg
- **ÁTLAG** – átlag
- **HA(logikai vizsgálat; érték, ha igaz; érték, ha hamis)** – feltételes elágazás
- **DARAB** – nem üres cellák száma az adott tartományban

táblázatkezelőben a következő formai beállításokat lehet eszközölni:

- **betűtípus, méret, szín** – hagyományos szövegformázás
- **szegélyek** – táblázatszerű kinézet
- **cellakitöltés** – kiemelésre
- **igazítás** – balra/középre/jobbra
- **számformátum** – pl. pénznem, dátum, százalék, de egyéni formátumot is be lehet állítani, pl. Km
- **feltételes formázás** – automatikus formázási szabály, amely azokban a cellákban lép érvénybe, ahol a megadott feltétel eredménye IGAZ

A **diagramok** segítségével **vizuálisan ábrázolhatjuk** az adatokat, így nagyban elősegítve az értelmezésüket, összehasonlíthatóságukat és átláthatóságukat. A leggyakrabban használt (leghasznosabb) diagramok a következők:

Diagramtípus	használatuk
oszlopdiaagram	adatok összehasonlítása
vonaldiaagram	időbeli változás szemléltetése
kördiaagram	arányok szemléltetése
sávdiaagram	mint az oszlopdiaagram, de vízszintes elrendezésű

18. tétel: Függvények

Mit nevezünk függvénynek? Mutasd be egy függvény használatát általánosan! Milyen függvénytípusokat ismersz? Jellemezd az egyes típusokat! Magyarázd el az alábbi függvények használatát: HATVÁNY(), SZUMHATÖBB(), BAL(), INDEX(), ÉV(). Milyen eszközeink vannak nagy adathalmazok átláthatóvá tételére? Milyen kapcsolat van a képletek és a feltételes formázás között?

A **függvény** egy előre beépített **képlet**, amely valamilyen **számítást, vagy műveletet** hajt végre egy vagy több érték alapján.

egy függvény **általános alakja** a következő:

=FÜGGVÉNYNÉV(argumentum1; argumentum2; ...)

a függvényeket az alábbi **altípusokra** lehet osztani:

- **matematikai** – számolás, összeadás, kivonás, hatványozás stb.
- **statisztikai** – átlag, minimum maximum
- **logikai** – feltételes vizsgálatok
- **szövegkezelő** – szövegdarabolás, szöveggé alakítás, egyesítés
- **dátum, idő** – dátum felbontása, számolása
- **keresési, hivatkozási** – adatok kikeresése
- **összegzés (feltételekkel)** – kijelölt adatok – feltételes – összegzése, számolása

a HATVÁNY függvény működése:

visszaadja egy szám hatványát

=HATVÁNY(2; 3) → 8 ($2^3 = 8$)

SZUMHATÖBB:

több feltétel alapján számol össze értékeket

=SZUMHATÖBB(B2:B10; A2:A10; „=piros”; C2:C10; „>10”)

azokat az értékeket számolja össze a B oszlopban, ahol A értéke „piros” és a C értéke nagyobb, mint 10

BAL:

a szöveg bal oldaláról ad vissza megadott számú karaktert

=BAL("Excel"; 2)

eredménye „Ex” lesz

INDEX:

egy adott sor- és oszlopszám alapján visszaadja a tartalom értékét

=INDEX(A2:C5; 2; 3)

a kijelölt tartományon belül a második sor harmadik oszlopának értékét adja vissza

ÉV:

csak az évszámot adja vissza egész számként egy dátum típusú adatból

=ÉV("2025.05.20")

2025

A nagy adathalmazok kezelésére, átláthatóvá tételére számos eszköz áll rendelkezésre:

- **szűrő** – csak a kívánt adatok jelennek meg
- **rendezés** – növekvő/csökkenő vagy ABC/fordított ABC sorrendben
- **csoportosítás**
- **táblázatként formázás**
- **keresési függvények** – FKERES(), INDEX() + HOL.VAN()

A **feltételes formázás** egy vizuális kiemelés, ami egy képlet, feltétel alapján történik, a saját képlettel történő feltételes formázáskor **gyakorlatilag egy HA() függvényt írunk**, amely a kijelölt tartomány minden egyes celláján végig megy.

VII. Adatbázis-kezeles

19. tétel: Az adatbázis-kezelés alapjai

Mit nevezünk adatbázisnak? Mit nevezünk adattáblának? Hogyan kapcsolódhatnak egymáshoz? Mit nevezünk mezőnek, rekordnak, adatnak? Milyen kulcsokat ismeresz? Ismertesd őket! Mit nevezünk kapcsolótáblának? Milyen adattípusokat használunk adatbázis-kezelőben?

Egy **adatbázis (database)** egy olyan rendszerezett adatgyűjtemény, amit egy **adatbázis-kezelő rendszer** (pl. Access, MySQL) segítségével **tárolunk, kezelünk és lekérdezzük**

Az adatbázis célja jól **strukturált, kereshető és összekapcsolható** adatok tárolása.

Az **adattábla** az adatbázis **alapegysége**

megjelenésben és információtartalmában nagyon hasonlít egy Excel táblára, **sorok és oszlopok** vannak benne

except for a sorokat **rekordoknak**, az oszlopokat **mezőknek** nevezik

több tábla abban az esetben **kapcsolható össze**, ha van bennük valamilyen közös mező (pl. azonosító)

a kapcsolatok rendszerezése típusuk szerint a következő:

1:1 pl. személy – személyigazolvány

1:N pl. tanár – diák

N:M pl. diák - tantárgy (ehhez a kapcsoláshoz viszont szükséges **kapcsolótábla**)

egy **mező** egyféle adatot tárol (pl. Név, Születési év)

egy **rekord** egy sor, tehát egy teljes „adatcsomag” (lekérdezésben a SELECT * használatakor ez jelenik meg) pl. egy ember adatai

adat: egy konkrét érték (Pl. „Kiss Anna”, vagy 2005)

háromféle kulcsot különböztetünk meg:

- **elsődleges kulcs (primary key)**
 - egyedi azonosító minden rekordhoz
- **idegen kulcs (foreign key)**
 - egy másik tábla elsődleges kulcsára hivatkozik
- **összetett kulcs**
 - több mező együtt ad egyedi azonosítót, pl. kapcsolótáblában

A kapcsolótábla egy külön tábla, ami két másik tábla közötti kapcsolatot biztosítja abban az esetben szükséges, ha **N:M** kapcsolat van mindkét tábla **idegen kulcsait** tartalmazza

Adatbázis-kezeléskor a következő adattípusokat használhatjuk:

- szám (INT, FLOAT)
- szöveg (TEXT, VARCHAR)
- dátum/idő (DATE, TIME, DATETIME)
- logikai (BOOLEAN)
- AutoNumber (SERIAL) – automatikusan növekvő azonosító (ID mezők)

20. tétel: SQL utasítások és függvények

Ismertesd az adatbázis létrehozásához szükséges utasítást! Ismertesd a táblák importálásához szükséges SQL utasításokat! Mutasd be a táblák bővítéséhez, frissítéséhez, törléséhez szükséges utasításokat! Ismertesd a lekérdezések szintaxisát! Milyen aggregáló függvényeket ismersz? Milyen dátum- és időfüggvényeket ismersz? Mit tudsz a szövegek kezeléséről SQL-ben? Mit tudsz az allekérdezésekről?

Az adatbázis létrehozásához szükséges utasítás a következő:

```
CREATE DATABASE <adatbazis_nev>;
```

tábla létrehozása:

```
CREATE TABLE <tabla_nev> (  
id INT PRIMARY KEY,  
nev VARCHAR(50),  
szuletési_ev INT  
);
```

importáláskor pedig:

```
LOAD DATA INFILE 'pelda.csv'  
INTO TABLE <table_nev>  
FIELDS TERMINATED BY  
'<adatelvalaszto>'  
LINES TERMINATED BY '<sorelvalaszto>'  
IGNORE n ROWS;
```

táblák bővítése:

új record beszúrása:

```
INSERT INTO <table_nev> (id, nev, szu-  
letési_ev)  
VALUES (1, 'Milyen Feri', 2004);
```

új mező hozzáadása:

```
ALTER TABLE <table_nev>  
ADD email VARCHAR(100);
```

adatok frissítése:

```
UPDATE <table_nev>  
SET nev = 'Olyan Feri'  
WHERE id = 1;
```

adatok törlése:

egy adott rekord törlése:

```
DELETE FROM <table_nev>  
WHERE id = 1;
```

teljes tábla kiürítése:

```
DELETE FROM <table_nev>
```

tábla törlése:

```
DROP TABLE <table_nev>
```

Az SQL lekérdezések alap szintaxisa:

```
SELECT mezok  
FROM tabla  
WHERE feltetel  
ORDER BY sorrend  
(GROUP BY mezo  
HAVING feltetel – aggregáló függvények-  
nél);
```

Az aggregáló (összegző) függvények:

Függvény	mit csin
COUNT()	megszámolja a sorokat
SUM()	összeadja az értékeket
AVG()	átlagot számol
MIN()	minimum érték
MAX()	maximum érték

szövegkezeléshez használt függvények:

Függvény	mit csin
CONCAT()	összefűz szövegeket
LENGHT()	megadja a szöveg hosszát
LOWER(), UPPER()	kis- vagy nagybetűssé alakít
LEFT(), RIGHT()	első, utolsó karaktereket jelenít meg

az allekérdezések (subquery) lényegében egy **SELECT a SELECT-ben**, tehát a (pl.) WHERE clause-ban egy másik lekérdezés eredményét használja

pl:

```
SELECT nev, szuletesi_ev
FROM <table_nev>
WHERE szuletesi_ev = (SELECT MAX(szuletesi_ev)
                      FROM <table_nev>);
```

ez pl. visszaadja a legfiatalabb diáko(ka)t