

# Számítógépes alkalmazások Dokumentum-kezelés

Soós Sándor, egyetemi adjunktus

Nyugat-magyarországi Egyetem  
Simonyi Károly Műszaki, Faanyagtudományi és Művészeti Kar  
Informatikai és Gazdasági Intézet

E-mail: [soossandor@inf.nyme.hu](mailto:soossandor@inf.nyme.hu)

Sopron, 2015.



# Tartalomjegyzék

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



# Miről lesz szó a mai órán?

- Szöveg-feldolgozás
- Szövegszerkesztés – Kiadványszerkesztés
- Szövegszerkesztők speciális funkciói
- Tipográfiai tanácsok, ajánlások
- Hogyan írjunk szakdolgozatot?
- Bemutató-készítés, prezentációk
- Táblázatkezelés
- Számítógépes grafika
- Pixelgrafika – Vektorgrafika
- Színek kezelése
- Konverziók
- Verziókezelés



# Dokumentum-feldolgozás

- Napjaink korszerű kommunikációs lehetőségei mellett is az írásos dokumentáció maradt az elsődleges információtárolási és továbbítási forma
- A legtöbb írott dokumentumot azonban ma már számítógéppel állítjuk elő
- Ennek lehetőségeiről fogunk most beszélni
- A dokumentumokkal kapcsolatos tevékenységek összessége a **dokumentum-feldolgozás**. Ez három fő részre bontható:
  - 1 dokumentum készítés
  - 2 dokumentum tárolás
  - 3 dokumentum visszakeresés
- A dokumentum-feldolgozáson belül először foglalkozunk a **szöveg-feldolgozással**, ami a dokumentum szöveges részeinek létrehozásához, tárolásához és visszakereséséhez kapcsolódó tevékenységek összességét jelenti



# Outline

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



# Szöveg-feldolgozás

- Az előállított elektronikus dokumentumok összetettségétől függően három szintet különböztetünk meg:
  - 1 text-editor  
formázás nélküli, text fájlok előállítása és kezelése, pl. programozói editorok, html szerkesztők
  - 2 szövegszerkesztés  
formázott, nyomtatásra is alkalmas dokumentumok előállítása és kezelése, pl. Microsoft Office, OpenOffice
  - 3 kiadványszerkesztés, desktop publishing (DTP)  
igényes, nyomdai minőségű dokumentumok előállítása és kezelése, pl. T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, MS Publisher, Adobe InDesign, PageMaker, QuarkXPress, Corel Ventura
- Nincs éles határ az egyes kategóriák között, de nagyon fontos, hogy tisztában legyünk a különbségekkel



# Szöveg-feldolgozás

- A text-editorokkal most nem foglalkozunk
- A hangsúlyt a szövegszerkesztés és a kiadványszerkesztés vizsgálatára helyezzük
- Hol a határ a kettő között?
  - a megbízhatóan kezelhető dokumentumok mérete szerint:
    - szövegszerkesztő: 10-20 oldal
    - kiadványszerkesztő: nincs korlát
  - a kezelhető dokumentumok összetettsége szerint:
    - szövegszerkesztő: 1 szövegfolyam beszúrt objektumokkal
    - kiadványszerkesztő: több keret (frame) kezelése, átfolyó dokumentumokkal (újság), tartalomjegyzék, tárgymutató, irodalomjegyzék, stb.
  - az alkalmazás eredeti célja szerint:
    - szövegszerkesztő: alapvetően szövegszerkesztő kiadványszerkesztő funkciókkal
    - kiadványszerkesztő: kiadványszerkesztőnek tervezve



# Szövegszerkesztők kevésbé ismert funkciói I

## 1 A szöveg tartalmi egységei:

- betű, karakter – letter, character
  - szó – word: szóközökkel (white space) határolt egység
  - bekezdés – paragraph: Enter karakterekkel határolt egység
  - szakasz (fejezet) – section: szakaszvége jelekkel határolt egység
  - teljes dokumentum – document
- 
- Figyelem! A sor és az oldal nem tartalmi egység, automatikusan változhat

## 2 A szöveg formázása a tartalmi egységekhez igazodik:

- karakterjellemzők: betűtípus, betűméret, betűstílus, betűszín, effektusok
- külön szó jellemzők nincsenek, esetleg az elválasztást sorolhatjuk ide





## Szövegszerkesztők kevésbé ismert funkciói II

- bekezdésjellemzők: igazítás, térköz, sorköz, számozás, behúzás, tabulátorok, tördelés szabályozása
- szakaszjellemzők: ide tartoznak a lap beállításai, azaz egy szakasz mindig azonos lapokból áll, lapméret, margók, tájolás, hasábok, fejléc, lábléc, oldalszámozás

### 3 Stílusok

- különböző szoftverek eltérő módon teszik elérhetővé a stílusok használatát, pl. MS-Office és OpenOffice: stílusok, HTML: CSS
- nem egyedileg formázzuk a betűket, bekezdéseket, helyette betű-, bekezdés-típusokat alakíthatunk ki
- egyszer meghatározzuk az egyes típusok jellemzőit
- az egyes szövegrészekről azt mondjuk meg, hogy milyen típusba tartozzon
- a konkrét formázást a szoftvert végzi el dinamikusan



## Szövegszerkesztők kevésbé ismert funkciói III

- ha módosítjuk egy típus jellemzőit, akkor automatikusan megváltozik az összes ilyen típusú szövegrészlet formázása
- segíti, hogy mindig egységes maradjon a szöveg(ek) formázása

### 4 Sablonok

- mintafájl hasonló dokumentumok létrehozására
- szövegelemek és stílusok egyaránt lehetnek benne
- például egyszer kell elkészíteni egy szakdolgozat sablont, és mindenki használhatja ugyanazt

### 5 Nyelv

- napjainkban már egyértelműen elvárás, hogy minden értelemben hibátlan dokumentumokat hozzunk létre bármilyen nyelven
- ez nem csak a begépelte szövegek nyelvét érinti
- helyesírási könyvtárak



## Szövegszerkesztők kevésbé ismert funkciói IV

- elválasztás
- írásjelek, tizedespont-vessző, idézőjelek
- standard szövegek: fejezet, ábra, kép, oldalszám, ...
- dátum, idő, pénznem formátumok

### 6 Automatikus javítás

- segíti és gyorsítja a munkát, de bizonyos funkciókat érdemes kikapcsolni
- kikapcsolandó funkciók: KÉt KEzdő NAgybetű, Cellák első betűje nagybetű

### 7 Gyorsszöveg, Kész szöveg, Autotext, makrók

- gyakran begépelendő szövegrészek gyorsítása

### 8 Helyesírás-ellenőrzés

- nagy segítség, de ne higgyünk neki vakon



# Szövegszerkesztők kevésbé ismert funkciói V

- legyen kéznél egy megbízható helyesírási szótár, pl.  
<http://magyarhelyesiras.hu/>

## 9 Korrektúra

- nagyon hasznos lehetőség, ha egy nagyobb dokumentumon hosszabb ideig dolgozunk, több változat készül belőle, vagy többen dolgozunk rajta
- amikor egy már lezárt változaton módosítunk, vagy a szerzőn kívül valaki más szerkeszti a dokumentumot, akkor kapcsoljuk be a *Korrektúra / Változatok követése* funkciót!
- így a módosításokat javaslatként rögzíti a rendszer, később visszanezethetjük ezeket, és eldönthetjük, hogy elfogadjuk, vagy visszavonjuk ezeket
- legkésőbb a dokumentum lezárásakor ne felejtsünk el véglegesíteni minden javaslatot



# Szövegszerkesztők kevésbé ismert funkciói VI

- ha valaki átnézi a munkánkat, kérjük meg, hogy ő is így jelezze a javaslatait

## 10 Tabulátorok:

- a szóközzel történő szövegigazítás helyett
- különböző igazítású tabulátorok: balra, jobbra, középre, tizedesvessző, ...
- a bekezdéshez tartozó jellemző
- professzionális kinézetű eredményt ad kis munkával

## 11 Táblázatok:

- a tabulátorok mellett egy másik lehetőség táblázatok létrehozására
- nem feltétlenül kell behúzni a cellák közötti vonalakat
- használhatunk különböző vonaltípusokat is



## Szövegszerkesztők kevésbé ismert funkciói VII

- kombinálhatjuk a tabulátorokkal, de egy cellában Ctrl-Tab-bal tudunk tabulátorjelet beszúrni

### 12 Fejléc – Lábléc – (Oldalszámozás)

- szakaszjellemző, tehát egy szakaszon belül egyformának kell lennie
- ha különbözőt szeretnénk, akkor új szakaszt kell kezdenünk
- egy szakaszon belül is háromféle lehet: első oldal, páros, páratlan oldal
- beszúrhatunk speciális mezőket is: dátum, oldalszám, oldalak száma, fejezetcímek, ...

### 13 Címsorszintek

- különböző szintű alpontjaink lehetnek: 1-9
- MS-Office, OpenOffice: stílusok segítségével állíthatjuk be: Címsor1, Címsor2, ...



## Szövegszerkesztők kevésbé ismert funkciói VIII

- $\text{\LaTeX}$ : külön parancsok segítségével állíthatjuk be:  
`\part`, `\section`, `\subsection`, ...
- ezen alapszik a tartalomjegyzék és a vázlatnézet

### 14 Vázlatnézet

- segíti a hosszabb dokumentumok áttekintését
- kiemeli a címsorokat
- szabályozhatjuk, hogy milyen mélységig akarjuk látni a címeket és az alattuk lévő normál szövegeket

### 15 Körlevél készítés, címkék, borítékcímzés

- hasonló tartalmú, de eltérő adatokat tartalmazó dokumentum sorozatok készítése
- különválasztjuk az adatokat és a sémát
- készítünk egy adatbázist, egy-egy rekord tartozik egy példányhoz



## Szövegszerkesztők kevésbé ismert funkciói IX

- készítünk egy sémát, amiben az állandó szövegrészek mellett megjelöljük az adatbázisból kiolvasandó adatelemeket
- összefuttatjuk a sémát és az adatforrást, így megkapjuk a dokumentumköteget
- a végeredmény lehet: különálló lapok, borítékok, különböző típusú etikettek (egy oldalon több címke)

### 16 Matematikai formulák kezelése

- tudományos dolgozatok esetében (a szakdolgozat is ilyen) kötelező elvárás, hogy megfelelő minőségben tartalmazzon matematikai kifejezéseket
- a használandó eszköz kiválasztásakor vegyük ezt figyelembe
- több lehetőség közül választhatunk:
  - 1 integrált képletszerkesztő (MS-Office, OpenOffice)
  - 2 integrált matematikai leíró nyelv (L<sup>A</sup>T<sub>E</sub>X)





# Szövegszerkesztők kevésbé ismert funkciói X

- 3 külső eszközökkel előállított képletek képként beszúrva (Maple, Mathematica, WolframAlpha, ...)
- bármelyik megoldást választjuk, szánjunk időt az eszköz használatának megtanulására
- alaposan teszteljük le, hogy milyen módon fogjuk illeszteni a képleteket a szöveg többi részéhez, felbontás, oldalszámozás, képletsorszámozás, nyomtatási minőség, ...
- tartsuk be a matematikai szövegek szedésére vonatkozó szabályokat, és megállapodásokat:
  - speciális szimbólumok:  $\sum$ ,  $\prod$ ,  $\int$ ,  $\pm\infty$ , ...
  - speciális műveleti jelek:  $\times$ ,  $\forall$ ,  $\exists$ ,  $\notin$ ,  $\cap$ ,  $\subseteq$ , ...
  - speciális ábécék:  $\alpha$ ,  $\beta$ ,  $\Delta$ ,  $\Omega$ ,  $\Re$ , ...
  - különböző fajta és méretű zárójelek:  $(, [, \{, |, |, ), ], \}$ , ...



# Szövegszerkesztők kevésbé ismert funkciói XI

- Például

$$f(x) = \sum_{i=0}^{\infty} \lambda_i x^i \left( \sqrt{\frac{x}{i!}} + \pi^2 \right)$$



# Tipográfiai tanácsok, ajánlások I

- 1 A dokumentumkészítés javasolt sorrendje: gépelés → formázás, a gépelés közben ne foglalkozzunk a formázással
- 2 ENTER csak a bekezdés végén! Esetleg új sor karakter (Shift-Enter)
- 3 Szóközből egyszerre csak egyet használunk, azaz szóközzel nem csinálunk helyet! Helyette tabulátorokat használjunk
- 4 Nyitózárájel előtt van szóköz (1 darab), utána nincsen
- 5 Végzárájel előtt nincs szóköz, utána van (1 darab)
- 6 Többféle „kötőjel” létezik: -, –, —
- 7 Többféle szóköz létezik: törhető, nem törhető, nyújtható, nem nyújtható



## Tipográfiai tanácsok, ajánlások II

- 8 Egyéb írásjelek előtt nincsen szóköz, utána 1 darab (.,;;!?)
- 9 A magyar idézőjel nem azonos az aposztróffal: „magyar idézet”, "aposztróf"
- 10 Ügyeljünk a hosszú szavak elválasztására, különösen akkor, ha sorkizárást használunk, használjunk automatikus, vagy opcionális elválasztást (Ctrl+kötőjel)
- 11 Ügyeljünk a laptörésekre! Bizonyos pontokon nem lenne jó, ha odakerülne a laptörés. Ne üres sorokkal oldjuk meg a problémát! Miért?  
A bekezdésjellemzők között megadhatunk erre vonatkozó szabályokat: fattyú- és árvasorok, együtt a következővel, egy oldalra, új oldalra



# Tipográfiai tanácsok, ajánlások III

Például:

- a címsort tartsuk együtt a következő bekezdéssel, ehhez az is kell, hogy a címsor után ne üres sor következzen, hanem a címsor térközét növeljük meg
  - hasonlóan általában együtt tartandók egy táblázat sorai
- 12 Ne vigyük túlzásba a formázási eszközök használatát!
- 13 Milyen dokumentum formátumot használjunk?
- Válasszuk szét a munka során használt és a végleges fájlformátumot!
  - A munkafájl formátumát majdnem egyértelműen meghatározza a használt szoftver:
    - 1 Winword: DOC bináris
    - 2 Winword 2007-től: Office Open XML (DOCX)
    - 3 L<sup>A</sup>T<sub>E</sub>X: TEX (text)



## Tipográfiai tanácsok, ajánlások IV

- 4 OpenOffice Writer: OpenDocument Format (ODF)
- 5 Ne felejtskezzünk el a csatolt fájlokról, fontokról!
- Gondoljuk meg, hogy kinek kell tudni hozzáférni a munkafájlokhoz (munkatársak, korrektor, konzulens, ... )!
- Végleges dokumentum formátumként olyat válasszunk,
  - amit bárki kényelmesen el tud olvasni, lehetőleg ingyen
  - változatlanul jelenik meg minden körülmények között
- Mire kell gondolni?
  - különböző nyelvű felhasználók (kódrendszerek)
  - különböző operációs rendszerek
  - olvasható-e online, le kell tölteni, vagy online is olvasható?
  - az olvasónak kell-e tudnia módosítani a fájlt?



# Tipográfiai tanácsok, ajánlások V

- Ezeknek a feltételeknek többé-kevésbé megfelelő fájlformátumok:
  - TXT
  - HTML
  - PDF
  - PS (Postscript)

## 14 Zárt és nyílt szabványok jelentősége!



# Hogyan írjuk a szakdolgozatot és a beadandó feladatokat?

- 1 Gondoljuk végig az előbbi szempontokat!
- 2 Ne ragadjunk le az első, legegyszerűbb megoldásnál (Winword)!
- 3 Ne akarjunk mindent egy programmal megoldani!
- 4 Keressük meg a legjobb eszközöket minden feladatra!
- 5 Készüljünk fel az illesztési, és konverziós feladatokra!
- 6 Törekedjünk a hordozható, szabványos, időtálló megoldásokra!  
Ezeket valószínűleg évek múlva is használhatjuk majd.
- 7 Bármelyik megoldást választjuk, a munka megkezdése előtt tervezzük meg az egész folyamatot! Ne a bekötés előtt egy héttel derüljön ki, hogy nem sikerült elkészíteni az ábrákat!
- 8 Hagyjunk időt az utolsó simításokra és a nyomtatásra, bekötésre!





# Outline

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



# Bemutatók készítése – Prezentációk

- A dokumentumok speciális formája a prezentáció
- Mi a különbség az írott dokumentum és a prezentáció között?
- Mikor választjuk az írott dokumentumot, mikor a prezentációt?
- A prezentációk típusai:
  - Információközlő prezentáció (pl. egyetemi előadás)
    - tárgyyszerű tudást kell átadnia
    - tanulni kell tudni belőle
    - a fő hangsúlynak a tartalomnak kell lennie
    - persze a forma sem közömbös
  - Figyelemfelkeltő prezentáció
    - meg kell győzni a hallgatóságot valamiről
    - nagyobb szerepet játszik a forma
- Prezentáció-készítő  $\neq$  Powerpoint!!!



# A prezentáció-készítés folyamata

- 1 a prezentáció megtervezése
- 2 forrásanyagok előkészítése
- 3 a prezentáció létrehozása
- 4 az egyes diák létrehozása
- 5 a diák sorrendjének és megjelenítési módjának meghatározása
- 6 a végleges formátum előállítása, esetleg nyomtatás
- 7 a prezentáció lejátszása, előadás



# A prezentáció-készítés folyamata részletesen I

## 1 a prezentáció megtervezése:

- mi a prezentáció célja?
- melyik típusba tartozik?
- kinek fog szólni a prezentáció?
- mennyi idő áll rendelkezésre az előadásra, és az elkészítésre?
- hogyan kell majd lejátszani a prezentációt: projektorral kis teremben, nagy teremben, weben, ...

## 2 forrásanyagok előkészítése:

- milyen forrás anyagok állnak rendelkezésre
- milyen anyagokat kell-tudunk elkészíteni
- képek, ábrák, grafikák, táblázatok, animációk, videók
- anyagok konvertálása
- vizsgáljuk meg, hogyan fog megjelenni az előadáson: méret, felbontás, színek, hang, nyelv



# A prezentáció-készítés folyamata részletesen II

## 3 a prezentáció létrehozása:

- milyen formát, stílust, sablont, témát választunk?
- szemben az írott dokumentumnál javasolttal, itt előbb válasszuk ki a formát, sablont, utána kezdjük el elkészíteni a diákat! Miért?
- tervezzük meg a diák állandó elemeit: diacím, fejléc, lábléc, dátum, diaszám, összes diák száma (legyen, vagy ne legyen?)

## 4 az egyes diák létrehozása:

- a tartalom létrehozása
- különböző objektumtípusokat használhatunk:
  - normál szöveg
  - pontozott lista
  - számozott lista
  - többszintű listák, különböző kombinációkban
  - grafikus objektumok, ábrák, képek, fotók,



## A prezentáció-készítés folyamata részletesen III

- esetleg animációk, videók, óvatosan használjuk!
  - külső linkek weblapokra, külső programokra  
ügyeljünk a hordozhatóságra! A lejátszáskor is elérhetők lesznek-e a külső kapcsolatok?
  - elemek formázása, színek, betűtípusok, méretek
  - Ne vigyük túlzásba az elemeket és a formázásokat!
  - Ne váljon öncélúvá a díszítés!
  - Mindig tartsuk szem előtt a prezentáció célját és a hallgatóságot! Másképp formázzuk a tartalmat, ha a barátunkat akarjuk vele felköszönteni a születésnapján, vagy az államvizsgán akarjuk bemutatni a szakdolgozatunkat
- 5 a diák sorrendjének és megjelenítési módjának meghatározása:
- Gondoljuk át a bemutatónk gondolatmenetét, és ennek megfelelően rendezzük sorba a diákat!



## A prezentáció-készítés folyamata részletesen IV

- Tervezzük meg, hogyan fogjuk átkötni az egyes diákat tartalmilag!
- Lehetőleg ne kelljen oda-vissza ugrálni a diák között! Néha ez nem elkerülhető!
- Ilyenkor gondoljuk meg a dia megismétlésének lehetőségét!
- Nem minden prezentációs eszköz lineáris, lásd Prezi!
- Ott is célszerű beállítani egy lineáris útvonalat is!
- Gondoljunk arra, hogy az olvasó utólag is megnézné a prezentációnkát (ha adunk rá lehetőséget)!
- Látványos effektekkel színesíthetjük a diaátmeneteket
- Animálhatjuk az egyes diákat is
- Ebben is legyünk mértéktartóak!

### 6 a végleges formátum előállítás:

- Milyen módon adhatjuk közre a prezentációt?



# A prezentáció-készítés folyamata részletesen V

- Szerkeszthető, módosítható, javítható: PPT, PPS (csak egy átnevezés), PPTX, OpenDocument ODP
- Nem szerkeszthető (nehezebben): PDF, Flash, képsorozat (JPG, BMP), HTML slideshow

## 7 a prezentáció lejátszása:

- Készüljünk fel alaposan!
- Próbáljuk ki a helyszínt!
- Teszteljük a projektort!
- Működnek-e a beágyazott elemek? Látszanak-e a projektoron is?





# Prezi.com

- magyar fejlesztés: [www.prezi.com](http://www.prezi.com)
- új megközelítés
- nincsenek diák, mint a megszokott prezentációkészítő programokban
- van egy nagy vászon, ezen helyezzük el a megjeleníteni kívánt elemeket: szöveg, kép, videó
- lejátszáskor egy képzeletbeli kamerát irányítunk a vászon felett
- előre megtervezhetjük a kamera mozgását, így ugyanolyan egyszerűen lépegethetünk diáról-diára mint a hagyományos programok esetében
- érdemes kipróbálni az ingyenes regisztrációt



# Speciális prezentációs eszközök

- ❶ képernyőfotó
- ❷ képernyővideó
- ❸ presenter
- ❹ mutatópálca
- ❺ lézerpointer
- ❻ intelligens tábla



# Outline

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



# Táblázatkezelés

- Táblázatkezelés  $\neq$  Microsoft Excel
- Alternatívák:
  - OpenOffice Calc
  - Cloud Computing megoldások: Google Docs, Zoho, ...
- Az alapötlet:
  - Vegyünk egy nagy négyzetrácsos „papírt”
  - Minden cellába írhatunk egy számot, egy tetszőleges szöveget, vagy egy képletet
  - Minden cellához tartozik egy képlet, egy érték, és egy formátum
  - Ezek kombinációival lehet kialakítani a kívánt kalkulációkat
- További funkciók:
  - nyilvántartások vezetése
  - adatok prezentálása
  - kisebb adattáblák kezelése



# A táblázatkezelő programok legfontosabb eszközei

- ❶ formázás ugyanúgy, mint a szövegszerkesztőben
- ❷ minden cellába írhatunk egy teljes dokumentumot
- ❸ cellaformátumok használata, automatikus, kézi beállítás
- ❹ ablaktábla rögzítése, felosztás
- ❺ sorozatok kezelése
- ❻ automatikus kitöltés
- ❼ számítások, kifejezések használata
- ❽ abszolút és relatív címzés, szorzótábla
- ❾ adatok kezelése
- ❿ adatok rendezése több szempont szerint  
Hogyan lehet rendezni háromnál több szempont szerint?
- ⓫ adatok szűrése, Autoszűrő
- ⓬ adatok ábrázolása, diagram rajzolás 2D/3D



# Függvények

- A számítások egyszerűsítésére több száz függvény áll rendelkezésre különböző kategóriákban:
  - matematikai
  - logikai
  - trigonometriai
  - statisztikai
  - pénzügyi
  - mátrix
  - dátum és idő
- A Függvényvarázsló segíti a függvények használatát



# Adatbáziskezelés

- Mikor nem elegendők a táblázatkezelő programok adatkezelő képességei?
  - túl nagy rekordszám
  - túl nagy adatbázisméret
  - a táblázatkezelő az egész adattáblát a memóriában tartja, az adatbáziskezelő csak az éppen szükséges rekordokat
  - több adattábla összekapcsolására van szükség
- Mit értünk adatbázis alatt?
  - adattáblák
  - adattáblák közötti kapcsolatok



# Számítógépes grafika

- 1 Pixelgrafika (rasztergrafika)
- 2 Vektorgrafika
- 3 Mi a különbség?
- 4 Melyik a jobb?





# Outline

- 1 Dokumentum-feldolgozás
  - Szöveg-feldolgozás
  - Bemutató-készítés
  - Táblázatkezelés
- 2 Számítógépes grafika
  - Pixelgrafika
  - Vektorgrafika
  - Színkezelés
  - Konverziók
- 3 Verziókezelés



# Pixelgrafika – Rasztergrafika

- Olyan digitális kép, ábra, melyen minden egyes képpontot (pixelt) önállóan definiálunk
- Előnyei:
  - egyszerű adatszerkezet
  - egyszerű algoritmus
  - gyors feldolgozás
  - fotótechnikai trükköknél jól alkalmazható
- Hátrányai:
  - az adatállomány nagy méretű
  - rögzített felbontás
  - nagyításnál a minőség romlik



# Pixelgrafika – Rasztergrafika

- Felbontás

- a külön kezelhető pixelek száma (sorok száma  $\times$  oszlopok száma)
- minél nagyobb a felbontás, annál jobb a kép minősége
- minél nagyobb a felbontás, annál nagyobb a fájl mérete
- a pixel tovább nem bontható (egyszínű), a legkisebb megkülönböztethető részlet 1 pixel
- ha ismerjük a felbontást és tudjuk, hogy a valóságban mekkora méretű területet ábrázol a kép, akkor meghatározhatjuk, hogy mekkora a legkisebb megkülönböztethető részlet

- Bitfelbontás – színmélység

- 1 pixelt hány biten ábrázolunk
- ettől függ, hogy hány színt, vagy hány szürkeárnyalatot tudunk megkülönböztetni
- a színek kódolására különböző módszereket használhatunk



# Outline

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



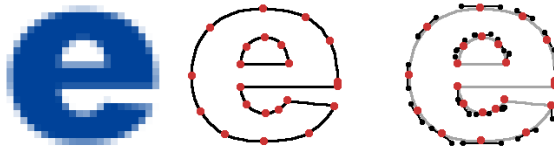
# Vektorgrafika

- Olyan számítógépes grafikai eljárás, melynek során geometriai primitíveket (rajzelemeket), mint például pontokat, egyeneseket, görbéket és sokszögeket használunk képek leírására
- Előnyei:
  - kisebb fájlméret
  - független a felbontástól
  - korlátlanul nagyítható minőségromlás nélkül
  - kiterjeszthető három dimenzióra
- Hátrányai:
  - bonyolultabb adatszerkezet
  - fényképfeldolgozásra nem használható
  - a legtöbb mai megjelenítő eszközünk rastergrafikus elven működik, ezért konvertálni kell! Kivétel?

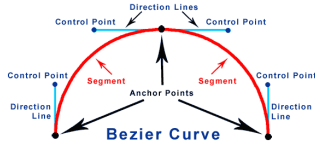


# Vektorgrafika

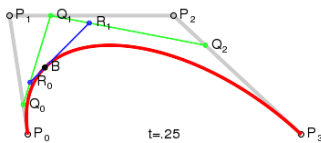
- A leggyakrabban használt primitívek:
  - vonalak és vonalláncok
  - sokszögek
  - körök és ellipszisek
  - Bezier-görbék és spline-ok
  - szöveg (A számítógépes betűket, például a TrueType fontokat Bezier görbék írják le.)



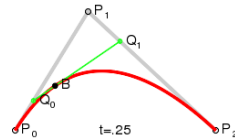
# Bezier-görbe



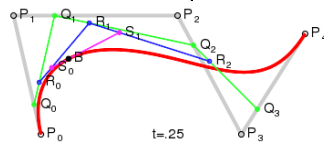
elsőfokú – lineáris



harmadfokú – köbös



másodfokú – quadratikus



negyedfokú

- Animáció GIF fájlokban: (Bezier\_1\_big.gif, Bezier\_2\_big.gif, Bezier\_3\_big.gif, Bezier\_4\_big.gif)



# Miről is van szó matematikailag?

- 1 Elsőfokú, lineáris Bezier-görbe:

$$B(t) = P_0 + t(P_1 - P_0) = (1 - t)P_0 + tP_1, t \in [0, 1]$$

- 2 Másodfokú, quadratikus Bezier-görbe:

$$B(t) = (1 - t)^2 P_0 + 2(1 - t)tP_1 + t^2 P_2, t \in [0, 1]$$

- 3 Harmadfokú, köbös Bezier-görbe:

$$B(t) = (1 - t)^3 P_0 + 3(1 - t)^2 tP_1 + 3(1 - t)t^2 P_2 + t^3 P_3, t \in [0, 1]$$

- 4 n-edfokú Bezier-görbe:

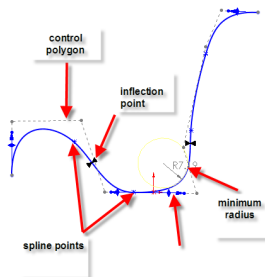
$$B(t) = \sum_{i=0}^n \binom{n}{i} (1 - t)^{n-i} t^i P_i$$





# Spline-ok

- szakaszosan parametrikus polinomokkal leírt görbe
- az  $n$ -edfokú spline-ban legfeljebb  $n$ -edfokú polinomszakaszok csatlakoznak egymáshoz úgy, hogy nemcsak a folytonosságot, hanem az  $n-1$ -szeri differenciálhatóságot is biztosítják
- lásd Spline01.gif!



# Outline

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

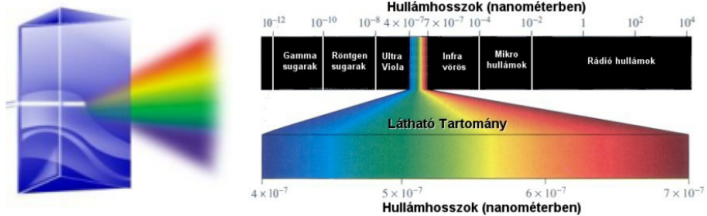
- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



# Hogyan kezeljük a színeket?

## 1 Fizika:



## 2 Festészet:

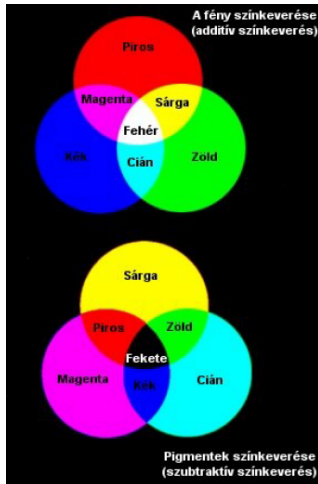
- Három alapszín: piros, sárga, kék

## 3 Világítástechnika:

- Három alapszín: piros, zöld, kék



# Additív és szubtraktív színkeverés



# Additív és szubtraktív színkeverés

## 1 Additív (összeadó) színkeverés

- alapszínek: Red-piros, Green-zöld, Blue-kék (RGB)
- a színek összeadódnak
- egy újabb szín hozzáadása növeli a színerőt
- a három alapszín egyenlő arányú keveréke fehéret eredményez
- a színek hiánya fekete színt ad
- mintha újabb és újabb színes lámpákat kapcsolnánk be

## 2 Szubtraktív (kivonó) színkeverés

- alapszínek: Cyan-ciánkék, Magenta-bíborvörös, Yellow-sárga (CMY)
- az elnyelt színek szerint keverednek
- újabb szín bekeverése csökkenti a színerőt
- az alapszínek egyenlő arányú keveréke feketét eredményez
- a színek hiánya fehéret ad
- mintha újabb és újabb átlátszó színes fóliákat helyeznénk egymásra



# Színek kódolása I

Különböző módszerekkel kódolhatjuk a színeket:

- ❶ Black and White (fekete-fehér):
  - egy képpontnak két állapota van: fekete vagy fehér
  - minden pixel 1 biten kódolható
- ❷ 16 Color (16 szín, vagy szürkeárnyalat)
  - minden pixel 4 biten kódolható
- ❸ 256 Color (256 szín, vagy szürkeárnyalat)
  - minden pixel 8 biten (1 bájt) kódolható

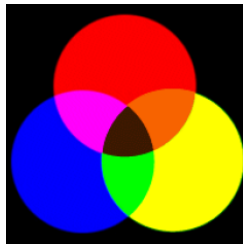
Az eddigi módszerek nem színkeverést kódolnak, hanem egy előre definiált palettáról választják ki az egyik színt, a paletta kicserélésével automatikusan megváltoznak a kép színei



# Színek kódolása II

## 4 RYB színrendszer

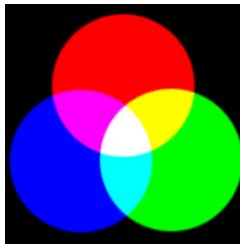
- alapszínek: piros, sárga, kék, a tiszta színek
- másodlagos színek: narancs, zöld, lila
- szubtraktív rendszer
- fehér: színek hiánya
- a három alapszín összege: barna
- ezzel a rendszerrel tiszta fekete nem keverhető ki



## Színek kódolása III

### 5 RGB színrendszer (24 bit Color), True Color

- alapszínek: piros, zöld, kék
- minden színkomponenst 256 fokozatban adhatunk meg
- 24 bit, 16 millió színárnyalat, 1 pixel: 3 bájt
- additív rendszer  $\Rightarrow$  minden: fehér, semmi: fekete
- ezt használják a monitorok és a projektorok





## Színek kódolása IV

### 6 CMYK színrendszer (32 bit Color)

- alapszínek: türkiz (cyan), bíbor (magenta), sárga (yellow), fekete (black)
- minden színkomponenst 256 fokozatban adhatunk meg
- 32 bit, 4,3 milliárd (billió) színárnyalat, 1 pixel: 4 bájt
- szubtraktív rendszer  $\Rightarrow$  minden: fekete, semmi: fehér
- ezt használják a nyomtatók
- egyes képszerkesztő programok csak 0-100 közötti értéket engednek meg komponensenként, ez 100 millió színárnyalatot eredményez



# Színek kódolása V

## 7 CMY színrendszer

- ugyanaz, mint a CMYK, csak fekete nélkül
- a fekete nem keverhető ki, csak sötétbarba
- szubtraktív rendszer  $\Rightarrow$  minden: fekete, semmi: fehér
- ezt használják egyes tintasugaras nyomtatók, a fekete külön patron
- az RGB–CMY átalakítás egy vektorművelettel egyszerűen megfogalmazható:

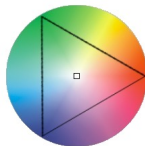
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



# Színek kódolása VI

## 8 HSB színrendszer

- három komponens: Hue – színárnyalat, Saturation – telítettség, Brightness – fényesség
- a színárnyalat egy színt választ ki a színceréken 0 és 359 között
- a telítettség és a fényesség megadása százalékban történik
- a telítettség jelentése: mennyire keskeny sávot határoz meg a színcerékből. Nagyobb érték esetén a megadott szín távolabbi szomszédai is részt vesznek a szín kikeverésében, a szín pasztell, majd szürkés árnyalatú lesz. A szín a telítettség minimális értéke esetén „tisztá” lesz.



# Színek kódolása VII

## 9 Színkerék modell tiszta színekkel

- a kerék peremén szerepelnek a színek
- egy egyenlő oldalú háromszög csúcaiban a tiszta színek: (kék, sárga, piros)
- velük átellenben a másodlagos színek (narancs, zöld, lila)
- két tetszőleges szín között a két színnel való keverés árnyalatai szerepelnek, félúton a két adott szín egyenlő arányú keverésével kapott szín
- a színek sorrendje adja a szivárvány színsorrendjét



## Színek kódolása VIII

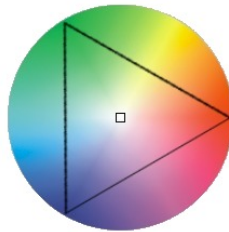
### 10 Színkerék modell alap színekkel (RGB)

- a színkerék egy sugarán haladva az adott szín telítettsége változik
- a színkerék területén a tiszta színek találhatók, a belsejében a pasztell színek
- ahogy ezen a sugáron a színkerék területétől a középpont felé haladunk, az eredeti (területen lévő) szín egyre nagyobb környezetéből keverednek össze a színek és ez a keveredés adja a sugár adott helyén lévő színárnyalatot. Maga a szín nem változik a sugár vonalában (mivel a területen lévő szín jobb és bal oldaláról azonos mértékben adódnak a bekeveredő színek és ezek a színek párosíthatók egymással, a párok összege viszont az eredeti színt adja)
- a sugár belseje felé haladva a szín veszít a telítettségéből



## Színek kódolása IX

- minden ilyen sugár esetében a színkerék középpontjához érve a fehér színt kapjuk, mivel ekkor már az eredeti szín bővülő környezete eléri a színkerék teljes kerületét, az additív színek összessége viszont fehér színt eredményez.



# Színkezelés a dokumentumkezelő programokban I

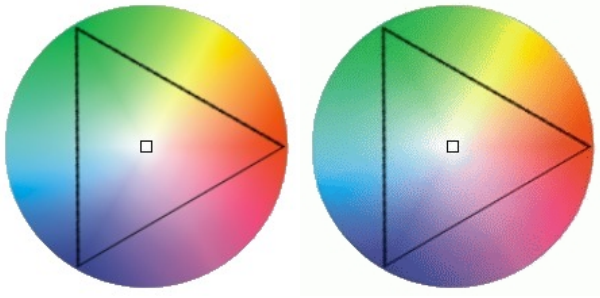
- A különböző dokumentumkezelő programok (nem csak a grafikus alkalmazások) az előbbi módszerek közül használnak egyet, vagy többet a színek kezelésére
- Ezeken az alapelveken alapszanak a programok színkezelő űrlapjai
- Ezért hasznos ismerni ezeket az elveket
- A felhasználó által beállított színeket azután a kiválasztott fájlformátum szabályai szerint tárolja el, szükség szerint konvertálva azokat
- Figyeljünk ezekre a konverziókra, és nem csak a színek tekintetében!



# Színkezelés a dokumentumkezelő programokban II

- Például

- ❶ kiválasztunk egy színt a programban az RGB színterékről
- ❷ elmentjük a képet GIF formátumban
- ❸ a GIF formátum csak 256 színt enged meg





# HTML színpaletták

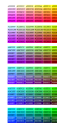
- Példaképpen nézzük meg a böngészőprogramok által támogatott színpalettákat:

Single Internet HTML Color Table  
web HTML Codes - 81 color set

	000	001	002	003	004	005	006	007	008
Row 1	FFFFFF	000000	000000	000000	000000	000000	000000	000000	000000
Row 2	000000	000000	000000	000000	000000	000000	000000	000000	000000
Row 3	000000	000000	000000	000000	000000	000000	000000	000000	000000
Row 4	000000	000000	000000	000000	000000	000000	000000	000000	000000
Row 5	000000	000000	000000	000000	000000	000000	000000	000000	000000
Row 6	000000	000000	000000	000000	000000	000000	000000	000000	000000
Row 7	000000	000000	000000	000000	000000	000000	000000	000000	000000
Row 8	000000	000000	000000	000000	000000	000000	000000	000000	000000
Row 9	000000	000000	000000	000000	000000	000000	000000	000000	000000

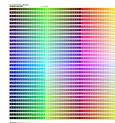
81 elemű paletta

html-color-palette-81.png



216 elemű paletta

html-color-palette-216.png



1536 elemű paletta

html-color-palette-1536.png



# Outline

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



# Konverziók a raszter- és a vektorgrafika között

## 1 Vektorgrafika → Rasztergrafika

- ez a gyakoribb és egyszerűbb eset
- majdnem mindig megtörténik, amikor egy program megjelenít egy vektorgrafikus ábrát
- a megadott sorrendben és szabályok szerint sorban ki kell rajzolni az objektumokat

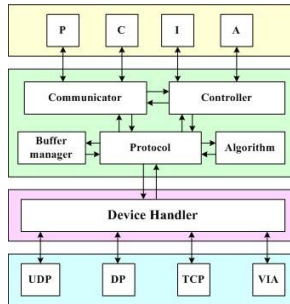
## 2 Rasztergrafika → Vektorgrafika

- ez sokkal ritkább és nehezebb
- mesterséges intelligenciát igényel, hogy a pixelgrafikus ábrában felismerjük az összetartozó pixeleket
- speciális eset az optikai karakterfelismerés (OCR), a felismerendő objektumok egy előre meghatározott készletből kerülnek ki, az ábécé betűi lehetnek



## Visio házi feladat

- Egy raszter → vektor konverzió elvégzése kézzel



téglalapok, vonalak, nyílak, szövegek, . . . , kapcsolatok!!!



# Speciális eszközök a vektorgrafikus programokban (Pl. Visio)

- egymástól független objektumok
- objektumok kombinációja
- csoportbafoglalás, rekurzívan
- rétegtechnika, layer (ez megvan a jobb pixelgrafikus programokban is)
- segédháló, rács
- segédvonal, vonalzó
- objektumok illesztése egymáshoz, rácshoz, segédvonalhoz
- kész, módosítható objektumok, clipart
- illesztő pontok, vonalak, nyílak illesztésére
- geometriai szerkesztő eszközök (CAD programok)



# Konverziók a fájlformátumok között

- Miért fontos a konverzió?
- A különböző fájlformátumok eltérő lehetőségekkel, szabályokkal, és korlátokkal rendelkeznek
- Ezek között vannak áthidalható, áthidalhatatlan és részben áthidalható ellentétek
- A konverzió speciális esete a tömörítés és a kódolás
- Különböző konverziók lehetőségek:
  - megfordítható  $\Leftrightarrow$  nem megfordítható
  - veszteséges  $\Leftrightarrow$  veszteségmentes
  - minőségartó, minőségcsökkentő, minőségjavító
- Fájlformátumok adatbázisa: <http://www.wotsit.org/>



# Verziókezelés

Miről van szó?

- különböző verziók, változatok születnek egyes dokumentumokból
- egy ember hosszabb ideig dolgozik rajta
- több ember dolgozik rajta egy időben, vagy egymás után
- sokszor hasznos lenne, ha
  - elő tudnánk venni egy korábbi verziót
  - elő tudnánk venni valamelyik munkatárs által készített verziót
  - össze tudnánk hasonlítani különböző verziókat
- ezeket az igényeket tudják kielégíteni a verziókezelő programok (Revision vagy Version Control System, RCS, VCS)



# A munkafolyamat

- elkészítem a dokumentum első változatát
- átadom a verziókezelő programnak (Check-in)
- a rendszer elteszi az adatbázisába (1.0 verzió)
- Ha legközelebb módosítani akarom a dokumentumot:
  - a verziókezelő programtól elkérem a dokumentum aktuális, vagy bármelyik korábbi verzióját (Check-out)
  - módosítom a kapott dokumentumot
  - az adatbázisban foglalttá, módosítottá (Modified) válik
  - visszaadom a dokumentumkezelő programnak (Check-in)
  - a rendszer elteszi az adatbázisba, növeli a verziószámot, de megőrzi az előző verziót is
  - minden változathoz megjegyzések és további információk fűzhetők
  - mindez működik több felhasználó esetén is
- Például: CS-RCS, Subversion, Git, Visual SourceSafe, GNU Arch, ...





# Összefoglalás

## 1 Dokumentum-feldolgozás

- Szöveg-feldolgozás
- Bemutató-készítés
- Táblázatkezelés

## 2 Számítógépes grafika

- Pixelgrafika
- Vektorgrafika
- Színkezelés
- Konverziók

## 3 Verziókezelés



## Befejezés

Köszönöm a figyelmet!

