

Digitális Kultúra Kidolgozott Tételek

| | |
|--|----|
| Dokumentumkészítés - Szövegszerkesztés | 3 |
| 1. téTEL: Betű és bekezdésformázás | 3 |
| 2. téTEL: Táblázatok és tabulátorok | 5 |
| 3. téTEL: Objektumok beszúrása..... | 7 |
| 4. téTEL: Oldalbeállítások, fájlműveletek..... | 9 |
| 5. téTEL: Kőrdokumentum | 11 |
| 6. téTEL: Nagy dokumentumok formázása | 13 |
| Dokumentumkészítés – Publikálás a világhálón..... | 15 |
| 7. téTEL: Betű és bekezdésformázások | 15 |
| 8. téTEL: Objektumok beszúrása..... | 17 |
| 9. téTEL: Stíluslapok használata | 19 |
| 10. téTEL: Dobozmodell..... | 23 |
| Dokumentumkészítés – Bemutatókészítés | 26 |
| 11. téTEL: A bemutatókészítés alapjai | 26 |
| 12. téTEL: A bemutatókészítéshez használható eszközök..... | 28 |
| Dokumentumkészítés – Pixelgrafika | 30 |
| 13. téTEL: A pixelgrafikus ábrázolás alapjai..... | 30 |
| 14. téTEL: Pixelgrafikus szerkesztés | 32 |
| Dokumentumkészítés – Vektorgrafika | 34 |
| 15. téTEL: A vektorgrafikus ábrázolás alapjai | 34 |
| 16. téTEL: Vektorgrafikus szerkesztés | 36 |
| Táblázatkezelés..... | 38 |
| 17. téTEL: A táblázatkezelés alapjai..... | 38 |
| 18. téTEL: Függvények | 40 |
| Adatbázis-kezelés..... | 42 |
| 19. téTEL: Az adatbázis-kezelés alapjai | 42 |
| 20. téTEL: SQL utasítások és függvények | 44 |

Dokumentumkészítés - Szövegszerkesztés

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?

Betűtípusok (Font Types)

A betűtípusok különböző kategóriákba sorolhatók:

Talpas betűtípusok (Serif): Ezek a betűtípusok talpakkal (kis vonalakkal vagy díszítésekkel rendelkeznek a betűk végein). Példák erre a típusra a **Times New Roman, Georgia és Garamond**.

Talpatlan betűtípusok (Sans-serif): Ezek a betűtípusok nem rendelkeznek a betűk végein talpakkal (kis vonalakkal vagy díszítésekkel). Ilyen betűtípusok például az **Arial, Helvetica és Calibri**.

Script: Ezek a betűtípusok kézzel írott vagy kalligrafikus stílusúak, például a *Brush Script* és *Pacifico*.

Monospace: minden karakter ugyanannyi helyet foglal el. Példák erre a típusra a **Courier New és Consolas**.

Betűstílusok (Font Styles)

A betűstílusok a következők lehetnek:

Félkövér (Bold): A betűk vastagabbak, hangsúlyosabbak, hogy kiemelkedjenek a szöveg többi részéből.

Dőlt (Italic): A betűk jobbra döntötték, gyakran használt a hangsúlyozáshoz vagy idézetekhez.

Aláhúzott (Underline): A betűk alatt vonal található, ami gyakran használatos címek és linkek kiemelésére.

Áthúzott (Strikethrough): A betűkön keresztben vonal húzódik, ami a törlés jelzésére szolgál.

KISKAPITÁLIS (Small Caps): A szöveg kisbetűi nagybetűs formában jelennek meg, de kisebb méretben, mint a valódi nagybetűk.

Speciális betűformázási beállítások

Számos speciális betűformázási lehetőség van:

Szín: A betűk színének megváltoztatása.

Betűméret: A betűk nagyságának módosítása (pl. 12 pt, 14 pt).

Betűterköz (Kerning): A betűközösségi távolság módosítása.

Sor közötti távolság (Leading): A sorok közötti távolság módosítása.

Árnyékolás (Shadow): Árnyék hozzáadása a betűk mögé.

Szimbólum karakterek

A szimbólum karakterek speciális karakterek, amelyek nem tartoznak az alaphangkészlethez. Példák: @, #, \$, %, &, *, ©, ™, §. Emellett ide tartoznak az emojik (😊, 👍, 🌟) és a matematikai szimbólumok (±, ≠, ∞, √) is.

Mértékegységek szövegszerkesztőben

A szövegszerkesztőkben különböző mértékegységeket használunk:

Pont (pt): A leggyakrabban használt mértékegység a betűméret meghatározásához.

Inch (in): Szélesség és magasság meghatározásához.

Centiméter (cm): Szélesség és magasság meghatározásához.

Milliméter (mm): Szélesség és magasság meghatározásához.

Pica: Nyomdai mértékegység, egy pica **12** pontnak felel meg.

Bekezdésformázások

A bekezdésformázás lehetőségei a következők:

Igazítás: A szöveg igazítható balra, jobbra, középre vagy sorkizárással.

Behúzás: Lehet első sor behúzása, függő behúzás vagy a teljes bekezdés behúzása.

Sorköz (sortávolság): Lehet szimpla, másfeles vagy dupla sortávolság.

Térköz: Bekezdések közötti térköz, valamint bekezdés előtti és utáni térköz beállítása.

Listák és felsorolások

A szövegszerkesztőben használható listák:

Felsorolás (Bulleted list): Különféle szimbólumokkal (pl.

- pontokkal
 - körökkel
-) ellátott listák.

Számozott lista (Numbered list):

1. Sorszámozott
2. elemekkel
3. ellátott
4. listák.

❖ **Többszintű lista:**

- Hierarchikus listák,
- ahol az elemek
 - különböző szinteken
 - ◆ jelennek meg.

Iniciálé

Az iniciálé a bekezdés első betűje, amely általában nagyobb méretű és díszesebb formázású, mint a többi betű. Kiemeli a bekezdés kezdetét, és gyakran használt díszítő elem a könyvekben és újságokban.

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?

Táblázatok elrendezésbeállításai

A táblázatokban különféle elrendezésbeállításokat alkalmazhatunk, mint például:

| | | |
|---|--|--------------------|
| Cellaméret | Módosíthatjuk a cellák magasságát és szélességét. | |
| Sorok és oszlopok hozzáadása/törlése | Új sorokat vagy oszlopokat adhatunk hozzá, illetve törölhetünk meglévőket. | |
| Cellák egyesítése és szétválasztása | Két vagy több cellát egyesíthetünk, illetve egyesített cellákat | szétválaszthatunk. |
| Cellabeállítások | Cellák igazítása (balra, jobbra, középre, felső, középső, alsó), cellamargók beállítása. | |

Táblázatok formázási lehetőségei

A táblázatok formázása során többféle lehetőséggel élhetünk:

Cellszegélyek és kitöltés:

| | | |
|-----------------------|-----------|---|
| | Szegélyek | Megváltoztathatjuk a szegélyek vastagságát, stílusát és színét. |
| Kitöltés (arnyekolas) | | A cellák háttérszínét módosíthatjuk. |

Szövegformázás:

Betűtípus, betűméret, betűszín beállítása.

Szöveg igazítása a cellákon belül (balra, jobbra, középre, sorkizáras).

Szöveg elforgatása a cellákon belül.

Táblázatstílusok:

Előre definiált táblázatstílusok alkalmazása.

Fejlécsor, összegző sor és szalagcím beállítása.

Tartalomigazítás:

| | |
|--|--|
| Szöveg vízszintes és függőleges igazítása. | |
| Cellákon belüli margók beállítása. | |
| | |

Tabulátor gomb funkciója

A tabulátor gomb (Tab) funkciója, hogy a kurzort előre mozgassa egy előre beállított távolságra a szövegben. Használatával gyorsan létrehozhatunk oszlopokat vagy sorokat, anélkül hogy manuálisan kellene szóközöket beszúrnunk.

Tabulátorpozíciók

A tabulátorpozíciók beállítása lehetővé teszi, hogy a kurzort a szövegben meghatározott helyekre mozgassuk. A következő tabulátorpozíciók állíthatók be:

Balra igazított tabulátor: A szöveg a tabulátorpozíciótól balra igazodik.

Középre igazított tabulátor: A szöveg középen igazodik a tabulátorpozícióhoz képest.

Jobbra igazított tabulátor: A szöveg a tabulátorpozíciótól jobbra igazodik.

Decimális tabulátor: A szöveg a tizedesvesszőhöz igazodik, a számok rendezésekor hasznos.

Sávos tabulátor: Egy függőleges sávot helyez el a szövegbe, vizuálisan elkülönítve a tartalmakat.

Egyéni tabulátorok

A szövegszerkesztők lehetőséget biztosítanak egyéni tabulátorok beállítására is. Ezeket az alábbi módon lehetjük meg:

Tabulátorpozíció hozzáadása: A vonalzó segítségével, vagy a tabulátorbeállítások menüben megadhatjuk a kívánt pozíciókat.

Tabulátorok módosítása: Megváltoztathatjuk a meglévő tabulátorok pozícióját és típusát.

Tabulátorok törlése: Eltávolíthatjuk a felesleges tabulátorpozíciókat.

Kitöltés karakterek: A tabulátorok közötti részt kitölthetjük pontokkal, vonalakkal vagy egyéb karakterekkel, például aláírás sorok vagy tartalomjegyzékek készítéséhez.

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 szövegdobozkról? Mit tudsz a hivatkozások beszúrásáról?

Objektumok szövegszerkesztőbe való beszúrása

Szövegszerkesztőkben különféle objektumokat lehet beszúrni, mint például:

Képek: Fájlok (JPEG, PNG, GIF stb.) beszúrása.

Alakzatok: Különböző geometriai formák, például körök, négyzetek, nyilak.

Táblázatok: Adatok rendszerezett formában történő megjelenítése.

Diagramok: Adatok vizuális ábrázolása grafikonok, diagramok formájában.

Szövegdobozok: Különálló szövegmezők a szöveges tartalmak elhelyezésére.

Hivatkozások: Kattintható linkek weboldalakra, dokumentumokra vagy más helyekre a dokumentumon belül.

Videók és hangfájlok: Multimédiás tartalmak beszúrása.

SmartArt: Előre elkészített grafikus elemek hierarchiák, folyamatok ábrázolására.

Alakzatok esetében mit nevezünk csomópontnak?

Az alakzatok csomópontjai (vagy csomópontok) olyan pontok, ahol az alakzat vonalai találkoznak vagy elágaznak. Ezek a pontok lehetővé teszik az alakzatok formájának módosítását azáltal, hogy az egyes pontokat mozgatva átrendezhetjük az alakzatot.



Kép beillesztésének lépései

Egy kép beillesztésének lépései a következők:

Hely kiválasztása: Helyezzük a kurzort arra a helyre a dokumentumban, ahová a képet be szeretnénk illeszteni.

Beszúrás menü: Nyissuk meg a "Beszúrás" menüt.

Kép beszúrása: Válasszuk a "Kép" vagy "Képek" opciót, majd a "Fájlból" lehetőséget.

Kép kiválasztása: Tallózzuk ki a beszúrni kívánt képet a számítógépről.

Kép beillesztése: Kattintsunk a "Beszúrás" gombra, hogy a képet beillesztjük a dokumentumba.

Beszúrt képek és alakzatok beállításai

A beszúrt képeken és alakzatokon hasonló, illetve egyedi beállításokat is eszközölhetünk:

Hasonló beállítások:

Méret módosítása: Mind a képek, mind az alakzatok méretét módosíthatjuk.

Elhelyezés: Mindkettőt áthelyezhetjük a dokumentum különböző részeire.

Forgatás: Mind a képeket, mind az alakzatokat elforgathatjuk.

Szöveg körbefuttatása: Mindkét (szoros, négyzetes, mögött, előtt stb.).



objektumnál beállíthatjuk a szöveg körbefuttatását

Egyedi beállítások:

Képek: Színek és szűrők módosítása, fényerő és kontraszt beállítása, vágás, képstílusok alkalmazása.

Alakzatok: Kitöltési szín, vonalszín és vonalvastagság módosítása, alakzatstílusok alkalmazása, csomópontok módosítása.

Szövegdobozok



A szövegdobozok olyan objektumok, amelyekben különálló szövegrészket helyezhetünk el a dokumentum más szövegeitől függetlenül. Ezeket a dobozokat szabadon mozgathatjuk, méretezhetjük és formázhatjuk. Alkalmazhatjuk rájuk ugyanazokat a szövegformázási beállításokat, mint a dokumentum többi részén, például betűtípus, betűméret és szín módosítása.

Hivatkozások beszúrása

A hivatkozások (linkek) beszúrásával interaktív elemeket hozhatunk létre a dokumentumban. A hivatkozások beszúrásának lépései:

Szöveg vagy objektum kiválasztása: Jelöljük ki azt a [szöveget](#) vagy objektumot, amelyet hivatkozássá szeretnénk alakítani.

Beszúrás menü: Nyissuk meg a "Beszúrás" menüt.

Hivatkozás beszúrása: Válasszuk a "Hivatkozás" vagy "Link" opciót.

Cím vagy URL megadása: Írjuk be a hivatkozás célját (webcím, e-mail cím, [vagy dokumentumon](#) belüli hely).

Hivatkozás beszúrása: Kattintsunk az "OK" vagy "Beszúrás" gombra.

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?

Nyelvi beállítások fontossága

A nyelvi beállítások kritikus szerepet játszanak a dokumentumok pontosságában és olvashatóságában. Két fő aspektusa van:

Nyelvi ellenőrzés: Ez biztosítja, hogy a helyesírási és nyelvtan hibakat azonnal felismerjük és javítsuk. Segíti a dokumentum professzionális megjelenését és érthetőségét.

Elválasztás: Az automatikus elválasztás segít a szöveg olvashatóságának javításában azáltal, hogy a szavak tördelésekkel elkerüli a nagy szóközöket és egyenletesebb szövegeszlopokat hoz létre.

Keresés és csere funkció használata

A keresés és csere funkció hasznos eszköz a dokumentumok gyors és hatékony szerkesztésére:

Keresés:

A "Keresés" funkcióval gyorsan megtalálhatjuk a szövegben előforduló szavakat vagy kifejezéseket.

Használhatjuk a Ctrl+F billentyűkombinációt a keresési ablak megnyitásához.

Csere:

A "Csere" funkcióval egy adott szót vagy kifejezést másikra cserélhetünk a dokumentumban.

Használhatjuk a Ctrl+H billentyűkombinációt a csere ablak megnyitásához.

Lehetőség van az összes előfordulás egyszerre történő cseréjére, vagy egyenkénti jóváhagyására.

Oldalbeállítások szövegszerkesztőben

A szövegszerkesztők különféle oldalbeállításokat kínálnak, amelyek segítségével testre szabhatjuk a dokumentum megjelenését:

Oldal mérete: A dokumentum méretének beállítása (pl. A4, Letter).

Tájolás: Az oldal tájolásának beállítása (álló vagy fekvő).

Margók: Az oldalak margóinak beállítása (felső, alsó, bal, jobb margók).

Hasábok: Az oldalak hasábokra bontása (egy vagy több hasábos elrendezés).

Fejléc és lábléc: Fejléc és lábléc hozzáadása, szerkesztése.

Oldalszámok: Oldalszámok beszúrása és formázása.

Dokumentum előkészítése nyomtatásra és PDF-be exportálásra

Nyomtatás előkészítése:

Ellenőrizzük a nyomtatási beállításokat, például az oldalméretet, margókat, tájolást.

Nézzük meg a "Nyomtatási kép" (Print Preview) nézetet, hogy ellenőrizzük a dokumentum kinézetét nyomtatás előtt.

Válasszuk ki a megfelelő nyomtatót és beállításokat (színes vagy fekete-fehér nyomtatás, oldalankénti nyomtatási opciók).

PDF-be exportálás:

Válasszuk a "Fájl" menüt, majd az "Exportálás" vagy "Mentés másként" opciót.

Válasszuk a PDF formátumot a mentési lehetőségek közül.

Adjuk meg a szükséges beállításokat, például a PDF fájl minőségét, jelszóval való védelmét.

A PDF fájlformátum előnyei

A PDF (Portable Document Format) számos előnytel jár:

Platformfüggetlenség: A PDF fájlok minden eszközön és operációs rendszeren ugyanúgy jelennek meg.

Biztonság: A PDF fájlokat jelszóval védhetjük és korlátozhatjuk a nyomtatási, másolási és szerkesztési jogokat.

Kompakt méret: A PDF fájlok tömörítettek, így kevesebb helyet foglalnak el, mint az eredeti dokumentumok.

Formázás megőrzése: A PDF megőrzi az eredeti dokumentum formázását, beleértve a betűtípusokat, képeket és elrendezést.

Elektronikus aláírások: A PDF támogatja az elektronikus aláírásokat, így hivatalos dokumentumok aláírására is használható.

Nyomtatási minőség: A PDF fájlok nyomtatási minőségeben menthetők el, így professzionális nyomtatáshoz is ideálisak.

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.)

Mi az a kördokumentum?

A kördokumentum egy olyan speciális dokumentumtípus, amelyet személyre szabott tömeges levelezéshez használnak. Ez lehetővé teszi, hogy egyetlen dokumentumból több, egyedileg testreszabott példányt hozunk létre, ahol minden példány tartalmazhat egyedi információkat (pl. név, cím).

A kördokumentum formái

A kördokumentumok különböző formái lehetnek:

Levél: Személyre szabott levelek tömeges létrehozása.

Címke: Címkék vagy borítékok nyomtatása különböző címzettek részére.

E-mail: Személyre szabott e-mailed küldése több címzett részére.

Brossúra vagy hírlevél: Egyedi tartalommal rendelkező nyomtatványok készítése.

A kördokumentum összetevői

Egy kördokumentum három fő összetevőből áll:

Fő dokumentum: Az alap dokumentum, amely tartalmazza a rögzített szöveget és a helyőrzőket (mezők) a változó adatok számára.

Adatforrás: Az a fájl vagy adatbázis, amely tartalmazza az egyedi adatokat (pl. Excel, CSV, Access).

Mezők: A fő dokumentumban elhelyezett helyőrzők, amelyek az adatforrásból származó egyedi adatokat jelenítik meg (pl. {Név}, {Cím}).

Kördokumentum elkészítésének lépései

Fő dokumentum elkészítése:

Nyissuk meg a szövegszerkesztőt (pl. Microsoft Word).

Készítsük el a levelet vagy dokumentumot, amely tartalmazza a rögzített szöveget és a helyőrzőket.

Adatforrás kiválasztása:

Az adatforrást tartalmazó fájl (pl. Excel) létrehozása vagy megnyitása.

Győződjünk meg arról, hogy az adatforrás oszlopai helyesen vannak elnevezve (pl. Név, Cím).

Körlevél indítása:

A szövegszerkesztőben válasszuk a "Levelezés" fület.

Kattintsunk a "Körlevél indítása" lehetőségre, majd válasszuk a dokumentum típusát (pl. levél).

Adatforrás csatolása:

Kattintsunk az "Címzettek kiválasztása" gombra.

Válasszuk az adatforrás fájlt, majd nyissuk meg.

Mezők beillesztése:

Helyezzük el a kurzort arra a helyre a fő dokumentumban, ahol a változó adatnak meg kell jelennie.

Kattintsunk a "Mező beszúrása" gombra, és válasszuk ki a megfelelő mezőt (pl. {Név}, {Cím}).

Körlevél befejezése:

Kattintsunk a "Körlevél befejezése és egyesítése" gombra.

Válasszuk az egyesített dokumentumok előnézetét, nyomtatását vagy új dokumentumba mentését.

Példa kördokumentum létrehozására **GYAKOROLNI**

Most bemutatom egy egyszerű kördokumentum elkészítését, például egy levél formájában.

Nyissuk meg a Microsoft Word-öt.

Készítsük el a következő levelet:

mathematica

Kód másolása

Tisztelt {Név}! Szeretettel meghívjuk Önt az eseményünkre, amely {Dátum}-án/én kerül megrendezésre a {Helyszín}.

Üdvözettel, [Cégnév]

Mentsük el a fenti szöveget "Körlevél.docx" néven.

Nyissuk meg az adatforrást, például egy Excel fájlt "Címzettek.xlsx" néven, amely a következő adatokat tartalmazza:

| Név | Dátum | Helyszín |
|--------------|------------|----------|
| Kovács János | 2024.06.15 | Budapest |
| Nagy Anna | 2024.06.16 | Szeged |

A Word-ben válasszuk a "Levelezés" fület, majd a "Címzettek kiválasztása" lehetőséget, és válasszuk a "Címzettek.xlsx" fájlt.

A "Mező beszúrása" gombra kattintva illesszük be a mezőket a megfelelő helyekre a levélben.

Kattintsunk a "Körlevél befejezése és egyesítése" gombra, majd válasszuk a "Dokumentumok szerkesztése" lehetőséget.

6. tétel: Nagy dokumentumok formázása

Mit nevezünk stílusnak? Hogyan használjuk őket? Hogyan kapcsolódnak a tartalomjegyzékhez? Mit tudsz az előfejről, előlábról? Mondj gyakorlati példákat alkalmazásukra! Milyen speciális beállításokat lehet beállítani előfejre előlábra vonatkozóan? Mi a különbség a lábjegyzet és a végjegyzet között? Mit nevezünk szakasznak?

Stílusok és használatuk

Stílus: A stílusok **előre meghatározott formázási beállítások**, amelyek segítségével egységes megjelenést biztosíthatunk a dokumentum különböző részein. Ezek a beállítások tartalmazhatnak betűtípust, betűméretet, színeket, térközöket és egyéb formázási jellemzőket.

Használatuk:

Stílusok alkalmazása: A Word szalagján a "Stílusok" panelen válasszunk ki egy stílust (pl. Címsor 1, Címsor 2, Normál).

Új stílus létrehozása: A "Stílusok" panelen kattintsunk a "Új stílus létrehozása" gombra, majd állítsuk be a kívánt formázási jellemzőket.

Stílusok módosítása: A "Stílusok" panelen kattintsunk a módosítani kívánt stílusra, majd a "Módosítás" lehetőséget választva szerkeszthetjük a beállításokat.

Stílusok és a tartalomjegyzék kapcsolata

A stílusok segítenek a tartalomjegyzék (TOC) automatikus generálásában. A Word a címsorstílusokat (Címsor 1, Címsor 2, stb.) használja a tartalomjegyzék felépítéséhez.

Lépések:

Címsorstílusok alkalmazása: Alkalmazzuk a megfelelő címsorstílusokat a dokumentum főbb részeire.

Tartalomjegyzék beszúrása: A "Hivatkozások" fülön válasszuk a "Tartalomjegyzék" lehetőséget, és válasszunk egy stílust.

Frissítés: Ha a dokumentum változik, kattintsunk a tartalomjegyzékre, majd válasszuk a "Tartalomjegyzék frissítése" opciót.

Előfej és előláb

Előfej (Header): A dokumentum felső részén megjelenő tartalom, amely ismétlődik minden oldalon.

Előláb (Footer): A dokumentum alsó részén megjelenő tartalom, amely ismétlődik minden oldalon.

Gyakorlati példák alkalmazásukra:

Előfej: Dokumentum címe, fejezetcímek, oldalszámok.

Előláb: Oldalszámok, dátum, szerző neve, lábjegyzetek.

Speciális beállítások előfejre és előlábra vonatkozóan

Első oldal eltérő beállítása: Az első oldal előfejének és előlábának eltérő formázása (pl. borítólap).

Páros és páratlan oldalak különböző beállításai: Különböző előfej és előláb beállítások alkalmazása a páros és páratlan oldalakra.

¹Szakaszokra bontás: Különböző előfej és előláb beállítások alkalmazása különböző dokumentum szakaszokban.

¹ ez egy labjegyzet

Lábjegyzet és végjegyzet

Lábjegyzet (Footnote): Az oldalak alján elhelyezett megjegyzések, hivatkozások vagy magyarázatok.

Végjegyzet (Endnote): A dokumentum végén elhelyezett megjegyzések, hivatkozások vagy magyarázatok.

Különbségek:

Elhelyezkedés: A lábjegyzetek az oldalak alján, míg a végjegyzetek a **dokumentum végén** jelennek meg.

Használat: A lábjegyzetek azonnali magyarázatot nyújtanak, míg a végjegyzetek összefoglaló magyarázatokat biztosítanak a dokumentum végén.

Szakaszok

Szakasz (Section): A dokumentum egy része, amely eltérő formázási beállításokat tartalmazhat. minden szakaszban külön beállításokat alkalmazhatunk, például oldalméretet, margókat, előfejet és előlábat.

Lépések a szakaszok létrehozásához:

Szakasztörés beszúrása: Az "Elrendezés" menüben válasszuk a "Töréspont" opciót, majd a "Szakasztörés" lehetőséget.

Formázás: minden szakaszban külön beállításokat alkalmazhatunk, például külön előfejet, előlábat, margókat és tájolást.

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 szűrhetünk be?

Hasznos funkciók egy kódszerkesztőben

Egy kódszerkesztő számos hasznos funkcióval rendelkezik, amelyek megkönnyítik a kódírást és a hibakeresést:

Szintaxis kiemelés: Különböző színek használata a kód különböző elemeinek megjelenítésére (pl. kulcsszavak, változók, stringek).

Automatikus kiegészítés: Javaslatok a kód beírása közben, hogy gyorsabban írhassunk kódot.

Hibajelzés és hibajavítás: Azonnali hibajelzés és javítási javaslatok.

Kódfeldarabolás és navigáció: A kód gyors keresése és navigációja a fájlok között.

Beépített terminál: Parancsok futtatása a szerkesztőből.

Bővítmények és integrációk: Kiegészítők és pluginek használata a funkcionalitás bővítésére.

Verziókezelés integrációja: Git vagy más verziókezelő rendszerek támogatása.

HTML5 alapstruktúrája

A HTML5 dokumentum alapstruktúrája a következőképpen néz ki:

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Dokumentum címe</title>
</head>
<body>
    <!-- A tartalom itt helyezkedik el -->
</body>
</html>
```

<!DOCTYPE html>: Jelzi a böngészőnek, hogy HTML5 dokumentumról van szó.

<html lang="hu">: A HTML dokumentum kezdő tagja, a **lang** attribútum a nyelvet jelöli.

<head>: Metaadatok és linkek a stíluslapokhoz, szkriptekhez.

<meta charset="UTF-8">: A karakterkódolást határozza meg.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: A nézetablak beállításai mobil eszközökön.

<title>: Az oldal címe, amely a böngésző fülén jelenik meg.

<body>: Az oldal tartalma.

Betűformázáshoz kapcsolódó HTML tagek

: (bold) Félkövér szöveg.

<i>: (italic) Dőlt szöveg.

<u>: (underlined) Aláhúzott szöveg.

<mark>: Kiemelt szöveg sárga háttérrel.

<small>: Kisebb szöveg.

 (strikethrough): Törölt szöveg (áthúzott).

<sub>: Alsó index.

<sup>: Felső index.

A <p>, <h1>, <h2>... tagek használata és sajátosságai

<p>: Bekezdés tag. minden új bekezdést ezzel a taggal jelölünk.

<p>Ez egy bekezdés.</p>

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>: Címsor tagek. A címsorok hierarchiáját jelzik, ahol a <h1> a legfontosabb címsor, és a <h6> a legkevésbé fontos.

<h1>Főcím</h1>
<h2>Alcím</h2>
<h3> me</h3>

Ezek a tagok strukturálják a dokumentumot, és segítenek a keresőmotoroknak és az olvasóknak megérteni a tartalom hierarchiáját.

Beszúrható listák

HTML-ben két fő típusú lista létezik:

Rendezetlen lista (): Listapontok, amelyek általában körökkel vannak megjelölve.

 Első elem
 Második elem
 Harmadik elem

Rendezett lista (): Számozott listapontok.

 Első elem
 Második elem
 Harmadik elem

Egyéb listák:

Definíciós lista (<dl>): Definíciók listája, ahol a <dt> a definíció kifejezést, a <dd> pedig a definíció leírását jelöli.

<dl>
 <dt>HTML</dt>
 <dd>HyperText Markup Language</dd>
 <dt>CSS</dt>
 <dd>Cascading Style Sheets</dd>
</dl>

8. tétel: Objektumok beszúrása

Milyen paraméterek adhatók meg képek beszúrásánál? Hogyan szúrhatunk 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 beszúrásánál megadható paraméterek

Amikor képeket szúrunk be egy szövegszerkesztőbe vagy HTML dokumentumba, számos paramétert és beállítást adhatunk meg:

Méret: A kép szélessége és magassága. Például:

```

```

Alt szöveg: Az alternatív szöveg, amely akkor jelenik meg, ha a kép nem töltődik be.

```

```

Hiperhivatkozás: A kép linkké alakítása.

```
<a href="https://www.pelda.com"></a>
```

Hivatkozások beszúrása

```
<a href="https://www.pelda.com">Kattints ide</a>
```

Oldalon belüli hivatkozások

Az oldalon belüli hivatkozások segítségével a dokumentum egy másik pontjára ugorhatunk:

Hivatkozás a célpontra:

```
<a href="#celpont">Ugrás a célpontra</a>
```

Célpont meghatározása:

```
<h2 id="celpont">Célpont</h2>
```

Képek beszúrása vs. videók és hangok beszúrása

Képek beszúrása:

Egyszerű HTML taggal megoldható (****).

Kevés paraméter (src, alt, méret).

Videók és hangok beszúrása:

Több paraméter és beállítás szükséges.

HTML5 példák:

Videó:

```
<video width="320" height="240" controls>
    <source src="video.mp4" type="video/mp4">
        Az Ön böngészője nem támogatja a videó elemet.
</video>
```

Hang:

```
<audio controls>
    <source src="hang.mp3" type="audio/mpeg">
        Az Ön böngészője nem támogatja az audio elemet.
</audio>
```

Táblázatok struktúrája

HTML-ben egy táblázat struktúrája a következő elemekből áll:

<table>: A táblázatot definiálja.

<th>: Fejléccella, amely általában félkövér és középre igazított.

<tr>: Egy sor a táblázatban.

<td>: Egy cella a táblázat sorában.

Példa egy egyszerű táblázatra:

```
<table border="1">
    <tr>
        <th>Név</th>
        <th>Kor</th>
    </tr>
    <tr>
        <td>János</td>
        <td>25</td>
    </tr>
    <tr>
        <td>Anna</td>
        <td>30</td>
    </tr>
</table>
```

9. tétel: Stíluslapok használata

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

Mit nevezünk CSS-szabálynak?

A CSS-szabály (Cascading Style Sheets) egy olyan utasítás, amely meghatározza, hogyan jelenjen meg egy adott HTML elem. Egy CSS-szabály **egy kijelölőből** és **egy deklarációs** blokkból áll.

Példa egy egyszerű CSS-szabályra:

```
p {  
    color: blue;  
    font-size: 16px;  
}
```

Kijelölő (selector): Meghatározza, mely HTML elemekre vonatkozik a szabály (pl. **p** jelöli az összes **<p>** elemet).

Deklarációs blokk (declaration block): A stílusokat tartalmazza, amelyek meghatározzák az elemek megjelenését. A deklarációk tulajdonság-érték párokból állnak (pl. **color: blue;** és **font-size: 16px;**).

CSS-szabályok több deklarációjával

Több deklaráció megadásához használunk pontosvesszővel elválasztott tulajdonság-érték párokat egy deklarációs blokkban.

Példa:

```
p {  
    color: red;  
    font-size: 14px;  
    line-height: 1.5;  
}
```

Betűformázó CSS deklarációk

Néhány gyakori betűformázó CSS deklaráció:

color: Szöveg színe.

font-size: Betűméret.

font-family: Betűtípus.

font-weight: Betű vastagsága (pl. **bold**).

font-style: Betű stílusa (pl. **italic**).

text-align: Szöveg igazítása (pl. **center, left, right**).

text-decoration: Szöveg dekorációja (pl. **underline, none**).

Példa:

```
h2 {  
    color: darkblue;  
    font-size: 20px;
```

```
    font-family: Arial, sans-serif;  
    font-weight: bold;  
}
```

Bekezdésformázó CSS deklarációk

Néhány gyakori bekezdésformázó CSS deklaráció:

text-align: Szöveg igazítása.

line-height: Sortávolság.

text-indent: Bekezdés első sorának behúzása.

margin: Külső margó.

padding: Belső margó.

Példa:

```
p {  
    text-align: justify;  
    line-height: 1.6;  
    text-indent: 20px;  
    margin: 10px 0;  
}
```

Osztály (class) és azonosító (id) kijelölők

Osztály kijelölő (class selector):

Több elemhez is rendelhető.

. (pont) szimbólummal jelöljük.

```
.button {  
    background-color: blue;  
    color: white;  
}
```

HTML:

```
<button class="button">Kattints ide</button>
```

Azonosító kijelölő (id selector):

Egy elemhez rendelhető, egy dokumentumban csak egyszer használható.

(kettős kereszt) szimbólummal jelöljük.

```
#main {  
    width: 80%;  
    margin: 0 auto;  
}
```

HTML:

```
<div id="main">Fő tartalom</div>
```

Stíluslap csatolása HTML dokumentumhoz

Három módon csatolhatunk stíluslapot egy HTML dokumentumhoz:

Külső stíluslap: Egy külön CSS fájlt csatolunk az alábbi módon.

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <title>Dokumentum címe</title>
        <link rel="stylesheet" href="styles.css">
</head>
<body>
    <!-- Tartalom itt -->
</body>
</html>
```

Belső stíluslap: A CSS kódot a HTML dokumentum **<head>** részében helyezzük el.

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <title>Dokumentum címe</title>
        <style>
            body {
                font-family: Arial, sans-serif;
            }
            p {
                color: blue;
            }
        </style>
</head>
<body>
    <p>Példaszöveg</p>
</body>
</html>
```

Inline stílus: A CSS kódot közvetlenül az adott HTML elem stílus attribútumában adjuk meg.

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <title>Dokumentum címe</title>
```

```
</head>
<body>
    <p style="color: blue; font-size: 14px;">Példaszöveg</p>
</body>
</html>
```

10. téTEL: Dobozmodell

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

A CSS dobozmodell ismertetése

A CSS dobozmodell leírja, hogyan helyezkednek el a HTML elemek és azok tartalma a weboldalon. minden elem egy téglaalap alakú dobozként jelenik meg, amely a következő részeiből áll:

Tartalom (Content): A doboz belső része, amely a szöveget, képeket vagy egyéb elemeket tartalmazza.

Belső margó (Padding): A tartalom körülbelül belső tér, amely megnöveli a doboz méretét.

Külső margó (Margin): A doboz körülbelül külső tér, amely elválasztja az elemet más elemektől.

Szegély (Border): A belső margó körülbelül keret, amely megjelenhet vagy láthatatlan is lehet.

Példa a dobozmodellre:

```
.box {  
    width: 300px;  
    padding: 10px;  
    border: 5px solid black;  
    margin: 20px;  
}
```

Szegélybeállítások

A szegélyeket (border) különféle módon állíthatjuk be:

Szegély vastagsága: Meghatározza a szegély vastagságát.

`border-width: 5px;`

Szegély stílusa: Meghatározza a szegély típusát (pl. szilárd, pontozott, szaggatott).

`border-style: solid; /* vagy dotted, dashed, double, groove, ridge,
inset, outset, none, hidden */`

Szegély színe: Meghatározza a szegély színét.

`border-color: red;`

Rövidített szintaxis: Az összes szegély tulajdonság egy sorban történő megadása.

`border: 5px solid red;`

Egyéni szegélyek: Egyes oldalakon külön-külön szegélyt beállítani.

`border-top: 5px solid red;
border-right: 5px solid blue;
border-bottom: 5px solid green;
border-left: 5px solid yellow;`

Objektumok méreteinek megadása

Az objektumok méreteit különböző módokon adhatjuk meg:

Szélesség és magasság: Az elem szélességének és magasságának meghatározása.

```
width: 300px;  
height: 200px;
```

Maximális és minimális méretek: Az elem maximális és minimális szélességének és magasságának meghatározása.

```
max-width: 100%;  
min-width: 150px;  
max-height: 500px;  
min-height: 100px;
```

Blokkok igazítása

Blokkokon belüli igazítás:

Szöveg igazítása: A szöveg igazítása az elem belsejében.

```
text-align: center; /* vagy left, right, justify */
```

Vertikális igazítás: Az elem tartalmának függőleges igazítása.

```
vertical-align: middle; /* vagy top, bottom */
```

Blokkok igazítása:

Külső margók használata: A blokkok közötti tér növelése vagy csökkentése.

```
margin: 20px auto; /* középre igazítás */
```

Flexbox: Rugalmas elrendezés a blokkok igazításához.

```
.container {  
    display: flex;  
    justify-content: center; /* középre igazítás vízszintesen */  
    align-items: center; /* középre igazítás függőlegesen */  
}
```

Doboz háttérkép beállítása

Egy doboz háttérképének beállítása a **background** tulajdonsággal történik.

Példa:

```
.box {  
    width: 300px;  
    height: 200px;  
    background-image: url('background.jpg');  
    background-size: cover; /* vagy contain, auto */  
    background-position: center; /* vagy top, bottom, left, right */  
    background-repeat: no-repeat; /* vagy repeat, repeat-x, repeat-y */}
```

Weboldal szabványossága és publikálása

Weboldal szabványossága:

W3C szabványok: A World Wide Web Consortium (W3C) által meghatározott szabványok követése.

HTML és CSS validáció: Az oldalak ellenőrzése a W3C validátor eszközeivel, hogy megfeleljenek a szabványoknak.

Hozzáférhetőség: Az oldal biztosítása, hogy minden felhasználó számára hozzáférhető legyen, beleértve a fogyatékkal élőket is (WCAG szabványok).

Weboldal publikálása:

Domain és tárhely beszerzése: Regisztrálunk egy domaint és vásároljunk tárhelyet.

Fájlok feltöltése: Az oldal fájljainak feltöltése a tárhelyre FTP vagy webes fájlkezelő segítségével.

Tesztelés: Az oldal működésének ellenőrzése különböző böngészőkben és eszközökön.

SEO és analitika: A keresőoptimalizálás (SEO) beállítása és analitikai eszközök (pl. Google Analytics) integrálása a weboldal teljesítményének nyomon követéséhez.

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?

Mi az a bemutató?

A bemutató egy **vizuális segédanyag**, amelyet előadások, prezentációk vagy oktatási anyagok támogatására használnak. Általában diák sorozatából áll, amelyek szöveget, képeket, diagramokat, grafikákat, videókat és egyéb multimédiás elemeket tartalmazhatnak.

Érvek a bemutató használata mellett és ellen

Érvek a bemutató használata mellett:

Vizualizáció: Segít a komplex információk vizuális megjelenítésében, ami könnyebben érthetővé teszi azokat.

Strukturálás: Lehetővé teszi az előadás szerkezetének egyértelmű bemutatását.

Memorálás: A vizuális elemek és pontok segítenek a hallgatóságnak könnyebben megjegyezni az elhangzottakat.

Interaktivitás: Multimédiás elemekkel interaktívabbá és érdekesebbé tehető az előadás.

Professzionális megjelenés: Jól elkészített bemutatók növelik az előadás professzionális színvonalát.

Érvek a bemutató használata ellen:

Technikai problémák: Technikai hibák előfordulhatnak, például meghibásodott projektor vagy nem működő fájlok.

Túlzott vizualizáció: A túl sok vizuális elem elvonhatja a figyelmet az előadás tartalmáról.

Időigény: A bemutató elkészítése időigényes lehet.

Függőség: Az előadó túlzottan függhet a bemutatótól, ami ronthatja az előadás spontaneitását és rugalmasságát.

Egyeszerűsítés: A bemutatók hajlamosak az információk túlzott leegyszerűsítésére.

Az előadáshoz készült bemutató elkészítésének lépései

Cél meghatározása: Határozzuk meg a bemutató célját és a célközönséget.

Tartalom tervezése: Gyűjtsük össze a bemutatóhoz szükséges információkat és készítsük el a vázlatot.

Diák szerkesztése:

Címdia: Az előadás címe, előadó neve és dátum.

Bevezető diák: Áttekintés a témáról és a főbb pontokról.

Tartalmi diák: Részletes információk képekkel, diagramokkal, grafikákkal és más vizuális elemekkel.

Összefoglaló diák: A főbb pontok összegzése és következtetések.

Záródia: Kérdésekre való felhívás és elérhetőségek és **FORRÁSOK**.

Dizájn és vizualizáció: Alkalmazzunk egységes dizájnt és vizuális stílust, válasszunk megfelelő színeket, betűtípusokat és grafikus elemeket.

Átmenetek és animációk: Adjunk hozzá átmeneteket és animációkat, de ne vigyük túlzásba.

Gyakorlás: Gyakoroljuk a bemutatót, hogy biztosak legyünk az időzítésben és az előadás folyamatosságában.

Technikai ellenőrzés: Ellenőrizzük a technikai eszközöket (projektor, laptop, távirányító stb.) a zökkenőmentes előadás érdekében.

Az előadásra készült bemutatók és az egyéni feldolgozásra készült bemutatók közötti különbségek

Előadásra készült bemutatók:

Képi anyag dominanciája: Több képi elem, kevesebb szöveg.

Egyeszerűség: Egyszerű, jól érthető diák, amelyek támogatják az előadást.

Interaktivitás: Több interaktív elem, amelyek bevonják a hallgatóságot.

Szóbeli kiegészítés: Az előadó szóbeli magyarázata kiegészíti a bemutatót.

Egyéni feldolgozásra készült bemutatók:

Részletes szöveg: Több szöveg, részletesebb magyarázatok, amelyek önállóan is érhetővé teszik az anyagot.

Önálló használat: Úgy vannak kialakítva, hogy az olvasó egyedül is megértse az információkat.

Kevesebb animáció: Kevesebb vagy semmilyen animáció, mivel nincs élő előadás.

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 áttűnésekkről, illetve az animációkról?

Vizuális eszközök a bemutató szerkezetének kialakítására

Diák: Az alapegységek, amelyekre a bemutató tartalma kerül.

Sablonok és téma: Előre elkészített dizájnok, amelyek egységes megjelenést biztosítanak a bemutatónak.

Címdia: A bemutató címe és az előadó neve, valamint az esemény dátuma szerepel rajta.

Tartalmi diák: Képek, diagramok, grafikák, szövegek és egyéb vizuális elemek elhelyezésére szolgálnak.

Átmenetek és animációk: Diák közötti áttűnésök és az elemek animációi, amelyek segítenek a figyelem fenntartásában és a bemutató folyamatosságának biztosításában.

Szövegdobozok: Szövegek elhelyezésére szolgáló dobozok.

Diagramok és grafikonok: Az adatok vizuális megjelenítésére szolgálnak.

Képek és videók: A vizuális tartalom gazdagítására szolgálnak.

Alakzatok: Különféle geometriai formák, amelyekkel kiemelhetünk fontos információkat vagy vizuális elrendezést adhatunk a diáknak.

SmartArt: Előre elkészített grafikus elemek, amelyek segítenek folyamatok, hierarchiák és kapcsolatok vizuális bemutatásában.

Hasonlóságok és különbségek a szövegszerkesztőbe és bemutatókészítőbe beszúrható objektumok között

Hasonlóságok:

Szövegdobozok: Mindkét eszközben használhatók szöveg elhelyezésére.

Képek: Mindkét eszközben beilleszthetők és szerkeszthetők képek.

Táblázatok: Mindkét eszközben létrehozhatunk és formázhatsunk táblázatokat.

Diagramok és grafikonok: Adatok vizuális megjelenítésére szolgálnak minden eszközben.

Hivatkozások: Linkek beszúrása minden eszközben lehetséges.

Különbségek:

Animációk és áttűnésök: A bemutatókészítők (pl. PowerPoint) lehetőséget biztosítanak animációk és áttűnésök alkalmazására, míg a szövegszerkesztőkben (pl. Word) ezek a funkciók általában nem elérhetők.

Interaktív elemek: A bemutatókészítők lehetővé teszik interaktív elemek, például kattintható gombok vagy navigációs eszközök beillesztését.

Elrendezési sablonok: A bemutatókészítők különböző elrendezési sablonokat és téma-kínálatot, amelyek segítségével egységes megjelenést érhetünk el, míg a szövegszerkesztőkben a sablonok inkább dokumentumformátumokra korlátozódnak.

Multimédiás tartalom: A bemutatókészítők jobban támogatják a videók és hangfájlok beillesztését és lejátszását, míg a szövegszerkesztőkben ezek a lehetőségek korlátozottabbak.

Példák:

Szövegszerkesztő (pl. Word): Szövegdobozok, képek, táblázatok, diagramok, hivatkozások.

Bemutatókészítő (pl. PowerPoint): Szövegdobozok, képek, táblázatok, diagramok, hivatkozások, animációk, áttűnésök, videók, hangfájlok, SmartArt elemek.

Áttűnések és animációk

Áttűnések:

Definíció: Az áttűnések a diákok közötti váltásokat határozzák meg.

Példák: Elhalványulás, áttolás, áttűnés, csúszás, forgatás.

Cél: Az áttűnések segítenek a bemutató folyamatosságának biztosításában és a figyelem fenntartásában.

Animációk:

Definíció: Az animációk az egyes elemek (szöveg, képek, alakzatok) megjelenését, eltűnését vagy mozgását határozzák meg egy dián belül.

Példák: Beúszás, elhalványulás, forgatás, nagyítás, sorban megjelenítés.

Cél: Az animációk segítenek a fontos információk kiemelésében és a figyelem irányításában.

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?

Mi az a pixelgrafikus kép?

A pixelgrafikus kép (vagy **raszteres kép**) olyan kép, amely apró négyzetekből, pixelekből áll össze. minden pixel egyetlen szín információt tárol, és a sok pixel együtt alkotja a teljes képet. A felbontás a kép méretét és részletességét határozza meg, amelyet a szélesség és magasság pixelekben kifejezve ad meg (pl. 1920x1080).

Színkódok és színmélység

Színkódok:

Hexadecimális színkód: Hatjegyű kód, amely a vörös, zöld és kék (RGB) komponenseket tartalmazza. Például: **#FF5733** (vörös: FF, zöld: 57, kék: 33).

RGB színkód: Hárrom számérték, amely az egyes színtartalom intenzitását adja meg. Például: **rgb(255, 87, 51)**.

Színmélység:

A színmélység a képernyőn tárolható színinformáció mennyisége, és általában bitben méri.

1-bit: Fekete-fehér kép (2 szín).

8-bit: 256 szín.

24-bit: True color, amely 16,7 millió színt képes ábrázolni (8 bit vörös, 8 bit zöld, 8 bit kék).

32-bit: 24-bit színinformáció és 8-bit alfa csatorna az átlátszóság kezelésére.

Pixelgrafikus szerkesztők

Néhány népszerű pixelgrafikus szerkesztő:

Adobe Photoshop: Professzionális képszerkesztő program, amely számos eszközt kínál képek szerkesztésére és manipulálására.

GIMP (GNU Image Manipulation Program): Ingyenes, nyílt forráskódú képszerkesztő program, amely sok hasonló funkcióval rendelkezik, mint a Photoshop.

Paint.NET: Ingyenes képszerkesztő program, amely egyszerűbb, de mégis hatékony eszközöket kínál.

Corel Painter: Digitális festőprogram, amely valósághű ecsetekkel és festési technikákkal rendelkezik.

Pixelgrafikus fájlformátumok

Néhány gyakran használt pixelgrafikus fájlformátum:

JPEG (Joint Photographic Experts Group):

Előnyök: Nagyon jó tömörítési arány, kis fájlméret.

Hátrányok: Veszeséges tömörítés, ami minőségiromlást okozhat.

Használat: Fényképek és webes képek esetében.

Fájl kiterjesztés: .jpg, .jpeg

PNG (Portable Network Graphics):

Előnyök: Veszeségmenes tömörítés, átlátszóság támogatása.

Hátrányok: Nagyobb fájlméret, mint a JPEG.

Használat: Webes grafikák, logók, átlátszó hátterű képek.

Fájl kiterjesztés: .png

GIF (Graphics Interchange Format):

Előnyök: Támogatja az animációkat és az átlátszóságot.

Hátrányok: Korlátozott színpaletta (256 szín).

Használat: Egyszerű animációk, webes grafikák.

Fájl kiterjesztés: .gif

BMP (Bitmap):

Előnyök: Egyszerű formátum, veszeségmenes.

Hátrányok: Nagy fájlméret, nem támogatja a tömörítést.

Használat: Képszerkesztés, nyomtatás.

Fájl kiterjesztés: .bmp

TIFF (Tagged Image File Format):

Előnyök: Nagyon jó minőségű, veszeségmenes, több réteget és átlátszóságot támogat.

Hátrányok: Nagy fájlméret.

Használat: Nyomtatás, professzionális képszerkesztés.

Fájl kiterjesztés: .tiff

Pixelgrafikus szerkesztőprogramok használata

Pixelgrafikus szerkesztőprogramokat különböző esetekben használhatunk:

Fényképek szerkesztése: Képjavítás, retusálás, színkorrekció.

Grafikai tervezés: Logók, ikonok, webes grafikák készítése.

Digitális festészet: Digitális művészet és festmények készítése.

Webdesign: Weboldal elemeinek, bannerek, gombok tervezése.

Nyomdal előkészítés: Plakátok, szórólapok, brosúrák tervezése.

14. téTEL: Pixelgrafikus szerkesztés

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

Pixelgrafikus szerkesztőprogram eszközei

Bemutatom a GIMP (GNU Image Manipulation Program) eszközeit, mivel ez az egyik legnépszerűbb és legátfogóbb ingyenes pixelgrafikus szerkesztőprogram. Az eszközöket csoportokba sorolhatjuk a funkcióik szerint.

Kiválasztó eszközök (Selection Tools)

Téglalap és ellipszis kiválasztó (Rectangle and Ellipse Select Tool): Téglalap vagy ellipszis alakú területek kiválasztására szolgál.

Lasso eszköz (Free Select Tool): Szabadkézi alakzatok kiválasztására szolgál.

Gyors kiválasztó eszköz (Fuzzy Select Tool): Hasonló színű területek kiválasztására szolgál.

Retusáló eszközök (Retouching Tools)

Ecset eszköz (Paintbrush Tool): Szabadkézi festés különböző ecsetekkel.

Klonozó bélyegző eszköz (Clone Tool): Kép egy részének másolása egy másik részre.

Gyógyító ecset (Healing Tool): Kisebb hibák, bőrhibák eltávolítására szolgál.

Javító ecset (Heal Selection Tool): Kisebb területek gyors javítására szolgál.

Képszerkesztő eszközök (Image Editing Tools)

Áthelyező eszköz (Move Tool): Rétegek és elemek áthelyezésére szolgál.

Radír eszköz (Eraser Tool): Képrészletek törlésére szolgál.

Árnyék és fény eszközök (Dodge, Burn, and Sponge Tools): Kép fényerősségeinek és színtelítettségének módosítására szolgál.

Szöveg eszközök (Text Tools)

Vízszintes szöveg eszköz (Text Tool): Szöveg bevitelére és szerkesztésére szolgál.

Néhány eszköz használatának szemléltetése

Ecset eszköz (Paintbrush Tool):

Használata: Válasszuk ki az ecset eszközt, majd a méretet és a keménységet állítsuk be. Az egérrel festhetünk a képen.

Példa: Egy portré finom részleteinek kidolgozása, árnyékolás hozzáadása.

Klonozó bélyegző eszköz (Clone Tool):

Használata: Az Alt billentyű lenyomásával jelöljük ki a forrásterületet, majd az egérrel fessünk a célterületen.

Példa: Nem kívánt objektumok eltávolítása a képből, például vezetékek eltüntetése egy tájképben.

Radír eszköz (Eraser Tool):

Használata: Válasszuk ki az eszközt, állítsuk be a méretet és a keménységet, majd töröljük a nem kívánt részeket.

Példa: Háttér eltávolítása egy objektum körül.

Rétegek (Layers)

A rétegek olyan különálló szintek, amelyek egymásra helyezve alkotják a végleges képet. Mindegyik réteg tartalmazhat különálló képi elemeket, szöveget vagy effekteket.

Előnyök:

Függetlenség: minden réteg külön-külön szerkeszthető anélkül, hogy a többi réteg befolyásolná.

Rugalmasság: Rétegek sorrendjének módosítása, átlátszóság beállítása, maszkolás alkalmazása.

Kompozíció: Különböző elemek kombinálása egy képen belül.

Alfa-csatorna

Az alfa-csatorna az átlátszóság információt tárolja egy képen belül. Ez lehetővé teszi, hogy a kép egyes részei átlátszóak legyenek, ami különösen hasznos összetett kompozíciók készítésekor.

Használata:

Maszkolás: Az alfa-csatorna segítségével pontosan meghatározhatjuk, mely részek legyenek láthatók és melyek átlátszóak.

Átlátszóság: Különböző elemek átlátszósági szintjének beállítása a képen.

Szöveg elhelyezése és beállításai

Vízszintes szöveg: Válasszuk ki a vízszintes szöveg eszközt, kattintsunk a kívánt helyre a képen, és írjuk be a szöveget.

Függőleges szöveg: Válasszuk ki a függőleges szöveg eszközt, kattintsunk a kívánt helyre, és írjuk be a szöveget.

Lehetséges beállítások:

Betűtípus: A szöveg betűtípusának kiválasztása.

Betűméret: A szöveg méretének módosítása.

Szín: A szöveg színének kiválasztása.

Stílus: Félkövér, dőlt, aláhúzott stb.

Igazítás: Balra, középre, jobbra igazítás.

Elrendezés: Szöveg pozíciójának és elrendezésének finomhangolása.

Effektek: Árnyék, kontúr, ragyogás hozzáadása.

Dokumentumkészítés - Vektorgrafika

15. téTEL: A vektorgrafikus ábrázolás alapjai

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

A vektorgrafikus ábrázolás központi fogalma

A vektorgrafikus ábrázolás központi fogalma a **vektor**. A vektorgrafikában az ábrák matematikai egyenletek és geometriai formák segítségével kerülnek leírásra, például pontok, vonalak, görbék és sokszögek. minden elem vektorokból épül fel, amelyek meghatározzák a formákat és azok elhelyezkedését a térben.

A vektorgrafika tulajdonságai

Méretezhetőség: A vektorgrafikus képek minőségveszteség nélkül méretezhetők. Akár nagyításról, akár kicsinyítésről van szó, a kép éles és tiszta marad.

Felbontásfüggetlenség: A vektorgrafikus képek nem függnek a felbontástól, így bármilyen kijelzőn vagy nyomtatási méretben jól mutatnak.

Kis fájlméret: A vektorgrafikus képek általában kisebb fájlmérettel rendelkeznek, mint a rasztergrafikus képek, mivel csak a matematikai leírásokat tárolják.

Szerkeszthetőség: A vektoros elemek könnyen módosíthatók, áthelyezhetők és átalakíthatók.

Pontosság: A vektoros elemek pontosan meghatározhatók és manipulálhatók.

A vektorgrafikus tárolás és a rasztergrafikus tárolás közötti különbségek

Adattárolás módja:

Vektorgrafika: Matematikai egyenletek és geometriai formák tárolása.

Rasztergrafika: Pixelek mátrixa, ahol minden pixel színinformációt tartalmaz.

Méretezhetőség:

Vektorgrafika: Méretezés minőségveszteség nélkül.

Rasztergrafika: Méretezés során minőségveszteség léphet fel (**pixelesedés**).

Fájlméret:

Vektorgrafika: Általában kisebb fájlméret, kivéve nagyon összetett ábrák esetén.

Rasztergrafika: Általában nagyobb fájlméret, különösen nagy felbontású képek esetén.

Felhasználási területek:

Vektorgrafika: Logók, ikonok, technikai rajzok, illusztrációk.

Rasztergrafika: Fotók, digitális művészet, részletgazdag képek.

Vektorgrafikus szerkesztőprogramok

Adobe Illustrator: Professzionális vektorgrafikai szoftver, széleskörű eszközökkel és funkcionálitással.

CorelDRAW: Nagy teljesítményű vektorgrafikai program, amely különösen népszerű a nyomdai és tervezési iparban.

Inkscape: Ingyenes és nyílt forráskódú vektorgrafikai szerkesztő, amely sok funkcióval rendelkezik.

Affinity Designer: Közkedvelt alternatíva az Adobe Illustrator számára, erőteljes funkciókkal és megfizethető áron.

Az önálló vektorgrafikai szerkesztőprogramok előnyei és hátrányai az irodai programokhoz képest

Előnyök:

Speciális eszközök: Az önálló vektorgrafikai programok fejlettebb és specifikusabb eszközöket kínálnak a vektoros elemek létrehozására és szerkesztésére.

Professzionális minőség: Kiváló minőségű vektorgrafikus tartalmak készítése, amelyek professzionális nyomdai és digitális felhasználásra is alkalmasak.

Komplexitás kezelése: Nagyon részletes és összetett ábrák kezelése, beleértve a rétegek és maszkok használatát.

Hátrányok:

Tanulási görbe: Az önálló vektorgrafikai programok gyakran bonyolultabbak, és több időt igényelnek a megtanulásukhoz.

Költség: A professzionális vektorgrafikai szoftverek (pl. Adobe Illustrator, CorelDRAW) drágák lehetnek.

Erőforrás igény: Az ilyen programok nagyobb rendszererőforrásokat igényelnek, ami lassíthatja a számítógépet.

Irodai programok:

Korlátozott funkcionalitás: Az irodai programok (pl. Microsoft Word, PowerPoint) csak alapvető vektoros eszközöket kínálnak, amelyek elegendők egyszerű grafikákhoz, de nem megfelelők komplex vektorgrafikákhoz.

Egyszerűség: Az irodai programok könnyebben használhatók, gyorsabb tanulási görbével.

Költséghatékonyság: Általában részei egy szélesebb irodai csomagnak, így nem igényelnek külön kiadást.

16. téTEL: Vektorgrafikus szerkesztés

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

Vektorgrafikus szerkesztőprogram felhasználói felülete

A bemutatáshoz az **Inkscape** nevű nyílt forráskódú vektorgrafikai szerkesztőt használjuk. Az Inkscape felhasználói felülete több részből áll:

Eszköztár (Toolbar): Bal oldalon található, és a különböző rajzeszközöket tartalmazza (pl. kiválasztó eszköz, rajzeszközök, alakzatok).

Tulajdonságssáv (Property Bar): Az ablak tetején található, és a kiválasztott eszközhez kapcsolódó beállításokat jeleníti meg.

Rétegek panel (Layers Panel): Jobb oldalon található, és a rétegeket kezeli.

Színek paletta (Color Palette): Az ablak alján található, és a színek gyors kiválasztására szolgál.

Dokumentum ablak (Document Window): Középen található, ahol a grafikai munka folyik.

ÁllapotSOR (Status Bar): Az ablak alján található, és a jelenlegi műveletről ad információkat.

A szerkesztőprogram eszközeinek kategorizálása

Kiválasztó eszközök:

Kiválasztó eszköz (Select Tool): Elemek kiválasztása, áthelyezése és méretezése.

Csoportosítási eszköz (Group/Ungroup Tool): Több elem csoportosítása vagy szétbontása.

Rajzeszközök:

Vonaleszköz (Line Tool): Egyenes vonalak rajzolása.

Szabadkézi eszköz (Freehand Tool): Szabadkézi rajzolás.

Bezier-görbe eszköz (Bezier Tool): Precíziós görbek rajzolása.

Alakzat eszközök:

Négyzet eszköz (Rectangle Tool): Négyzetek és téglalapok rajzolása.

Ellipszis eszköz (Ellipse Tool): Körök és ellipszisek rajzolása.

Többszög eszköz (Polygon Tool): Többszögek és csillagok rajzolása.

Szöveg eszközök:

Szöveg eszköz (Text Tool): Szöveg hozzáadása és szerkesztése.

Átalakító eszközök:

Forgatás eszköz (Rotate Tool): Elemek forgatása.

Tükörzés eszköz (Flip Tool): Elemek tükrözése.

Szín és stílus eszközök:

Kitöltés és kontúr eszköz (Fill and Stroke Tool): Kitöltési szín, kontúrszín és vastagság beállítása.

Színválasztó eszköz (Color Picker Tool): Színek kiválasztása a dokumentumból.

Néhány eszköz használatának szemléltetése

Kiválasztó eszköz (Select Tool):

Használata: Kattintsunk egy elemre a kiválasztásához. A fogópontokat használva átméretezhetjük vagy áthelyezhetjük az elemet.

Példa: Egy téglalap áthelyezése és méretének módosítása.

Bezier-görbe eszköz (Bezier Tool):

Használata: Kattintsunk a kezdőpontra, majd kattintással adjuk meg a következő pontokat. Kattintsunk és húzzuk a görbe formálásához.

Példa: Egy ívelt vonal rajzolása egy logóban.

Szöveg eszköz (Text Tool):

Használata: Kattintsunk a dokumentumra, írjuk be a szöveget, majd a tulajdonságávon állítsuk be a betűtípusat, méretet és színt.

Példa: Egy cég neve hozzáadása egy brosúrához.

A klónok használata

A klónok olyan másolatok, amelyek az eredeti objektumtól függnek. Ha az eredeti objektumot módosítjuk, a klónok automatikusan frissülnek.

Használata:

Kattintsunk a kiválasztandó objektumra, majd válasszuk a "Szerkesztés" menüből a "Klón készítése" opciót.

A klónok elhelyezhetők és méretezhetők, de a szín és a forma az eredeti objektumtól függ.

Unió, metszet, különbség műveletek

Unió (Union): Két vagy több objektum összeolvasztása egyetlen objektummá.

$$A + B = AB$$

Metszet (Intersection): Két vagy több objektum közös területének létrehozása.

$$A \cap B = C$$

Különbség (Difference): Az első objektumból a második objektum által lefedett terület eltávolítása.

$$A - B = D$$

Útvonalak (Paths)

Az útvonalak (paths) olyan vektoros elemek, amelyek pontokból és a közöttük húzódó görbékből állnak. Az útvonalak rugalmasan szerkeszthetők és átalakíthatók.

Használata:

Útvonalak létrehozása Bezier-görbékkel vagy vonaleszközökkel.

Útvonalak szerkesztése pontok és szegmensek hozzáadásával, eltávolításával vagy áthelyezésével.

Példák:

Logók, ikonok és más grafikai elemek létrehozása.

Egyedi alakzatok és görbék rajzolása.

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!

Mi az adat?

Az adat olyan információ, amelyet számítógépes rendszerekben tárolunk és kezelünk. Az adatok lehetnek számszerűek, szövegesek, időpontok, dátumok vagy egyéb formátumok, amelyek feldolgozhatók és elemzhetők.

Adattípusok

Három különböző adattípust ismernek a táblázatkezelők:

Szám (Number): Egész számok és tizedes tört számok (pl. 42, 3.14).

Dátum és időpont adatokat a táblázatkezelők valós számként tárolnak.

Szöveg (Text): Karakterláncok, amelyek betűket, számokat és egyéb karaktereket tartalmaznak (pl. "Hello", "123ABC").

Logikai (Boolean): Igaz vagy hamis értékek (pl. TRUE, FALSE).

Dátum és idő adatok

A dátum és idő adatok különleges adattípusok, amelyek lehetővé teszik az időpontok kezelését, számítását és formázását. A táblázatkezelőkben általában előre definiált formátumokat használunk, és különféle műveleteket végezhetünk rajtuk, például dátumok közötti különbségek kiszámítása.

Példák:

2024-06-02 (dátum)

14:30:00 (idő)

2024-06-02 14:30:00 (dátum és idő)

Cellahivatkozás

A cellahivatkozás azt jelenti, hogy egy adott cellát (vagy cellatartományt) megnevezünk egy táblázatban. Különböző típusai vannak:

Relatív hivatkozás: A cellahivatkozás az aktuális pozícióhoz képest változik, amikor másoljuk. Példa: **A1**.

Abszolút hivatkozás: A cellahivatkozás fix marad, függetlenül attól, hova másoljuk. Példa: **\$A\$1**.

Vegyes hivatkozás: A cellahivatkozás egy része fix, más része relatív. Példa: **\$A1** vagy **A\$1**.

Kifejezések összeállítása

Kifejezések segítségével végezhetünk számításokat és műveleteket a táblázatkezelőben. Az alapvető matematikai műveleteken kívül logikai és szövegfüggvényeket is használhatunk.

Példák:

Összeadás: **=A1 + B1**

Kivonás: **=A1 - B1**

Szorzás: **=A1 * B1**

Osztás: **=A1 / B1**

Feltételes függvény: =HA(A1 > B1, "Nagyobb", "Kisebb")

Dátumok különbsége: =A1-A2 (ha minden cella dátumot tartalmaz)

Formázási beállítások táblázatkezelőben

A táblázatkezelők számos formázási lehetőséget kínálnak, hogy az adatok jól olvashatók és esztétikusak legyenek:

Szövegformázás: Betűtípus, betűméret, félkövér, dőlt, aláhúzott.

Színezés: Cellák háttérsíne, szöveg színe.

Szegélyek: Cellák körül vonalak hozzáadása és stílusának beállítása.

Igazítás: Szöveg igazítása balra, közepre, jobbra, függőleges igazítás.

Számformátumok: Számok formátumának beállítása, például pénznem, százalék, dátumformátum.

Diagramok és diagramfajták

A diagramok vizuálisan ábrázolják az adatokat, segítve az elemzést és a megértést. Különböző típusú diagramok használhatók, attól függően, hogy milyen adatokat akarunk megjeleníteni.

Gyakori diagramtípusok:

Oszlopdiagram (Column Chart):

Függőleges oszlopokkal ábrázolja az adatokat.

Például havi értékesítési adatok összehasonlítása.

Sávdiagram (Bar Chart):

Vízszintes sávokkal ábrázolja az adatokat.

Például különböző kategóriák összehasonlítása.

Vonaldiagram (Line Chart):

Adatok időbeli változását ábrázolja vonalak segítségével.

Például havi hőméréséleti adatok ábrázolása.

Kördiagram (Pie Chart):

Egy teljes értéket részekre bontva ábrázol.

Például egy költségvetés részeinek arányos megjelenítése.

Pontdiagram (Scatter Plot):

Két változó közötti kapcsolatot ábrázolja pontok segítségével.

Például kétféle mérés közötti korreláció.

Területdiagram (Area Chart):

Vonaldiagramhoz hasonló, de a terület színezi ki az adatokat.

Például összesített adatok időbeli változásának ábrázolása.

Diagramok előnyei:

Az adatok könnyebb megértése és értelmezése.

Az adatok közötti kapcsolatok és trendek azonosítása.

A döntéshozatali folyamat támogatása vizuális ábrázolással.

18. téTEL: Függvények

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

Mi az a függvény?

A függvény olyan előre definiált képlet vagy művelet, amely egy vagy több bemeneti értéket vesz fel (argumentumok), és egy kimeneti értéket ad vissza. A függvények megkönnyítik az összetett számítások és adatmanipulációk végrehajtását a táblázatkezelőkben.

Függvény használatának általános bemutatása

Egy függvény használata általánosan a következőképpen történik:

Függvény neve: A függvény neve (pl. **SZUM** a summázásra).

Nyitó zárójel: A függvény neve után egy nyitó zárójel (következik.

Argumentumok: Az argumentumok vagy bemeneti értékek, amelyeket a függvény használ. **Ezek lehetnek cellahivatkozások, értékek, vagy más függvények.**

Záró zárójel: Az argumentumok után egy záró zárójel) következik.

Példa: =SZUM(A1:A10) összegzi az A1-től A10-ig terjedő cellák értékeit.

Függvénytípusok és jellemzésük

Matematikai és trigonometriai függvények: Olyan függvények, amelyek különböző matematikai és trigonometriai számításokat végeznek.

Példa: HATVÁNY(), SZUM(), SZORZAT()

Statisztikai függvények: Olyan függvények, amelyek statisztikai elemzéseket végeznek az adatokon.

Példa: ÁTLAG(), MEDIÁN(), MAX(), MIN()

Szövegfüggvények: Olyan függvények, amelyek szöveges adatokat manipulálnak.

Példa: BAL(), JOBB(), KÖZÉP(), HOSSZ()

Dátum és idő függvények: Olyan függvények, amelyek dátumokkal és időpontokkal dolgoznak.

Példa: DÁTUM(), ÉV(), HÓNAP(), NAP()

Logikai függvények: Olyan függvények, amelyek logikai értékeket adnak vissza, vagy feltételek alapján számítanak.

Példa: HA(), ÉS(), VAGY()

Keresőfüggvények: Olyan függvények, amelyek adatbázis-szerű műveleteket végeznek a táblázatban.

Példa: FKERES(), INDEX(), HOL.VAN()

Függvények használata

HATVÁNY()

Leírás: Kiszámítja egy szám adott hatványát.

Szintaxis: =HATVÁNY(szám, kitevő)

Példa: =HATVÁNY(2, 3) eredménye 8, mert $2^3 = 8$.

SZUMHATÖBB()

Leírás: Összeadja azokat az értékeket, amelyek megfelelnek egy vagy több kritériumnak.

Szintaxis: =SZUMHATÖBB(tartomány, kritérium1, [tartomány2], [kritérium2], ...)

Példa: =SZUMHATÖBB(A1:A10, ">5") összeadja az A1

tartományban lévő azon értékeket, amelyek nagyobbak 5-nél.

BAL()

Leírás: Egy szöveg bal oldaláról adott számú karaktert ad vissza.

Szintaxis: =BAL(szöveg, [karakterek_száma])

Példa: =BAL("Hello", 2) eredménye "He".

INDEX()

Leírás: Egy tömb vagy tartomány adott pozíójában lévő értéket adja vissza.

Szintaxis: =INDEX(tömb, sor_szám, [oszlop_szám])

Példa: =INDEX(A1:B2, 2, 1) eredménye az A2 cella értéke.

ÉV()

Leírás: Egy dátumból visszaadja az év részét.

Szintaxis: =ÉV(dátum)

Példa: =ÉV("2024-06-02") eredménye 2024.

Nagy adathalmazok átláthatóvá tétele

Szűrés (Filter): Az adatok szűrése bizonyos feltételek alapján.

Rendezés (Sort): Az adatok rendezése növekvő vagy csökkenő sorrendbe.

Pivot táblák (Pivot Tables): Az adatok összegzése és csoportosítása egy új táblában.

Kondicionális formázás (Conditional Formatting): Az adatok vizuális kiemelése színezéssel, ikonnal vagy egyéb formázási elemekkel, feltételek alapján.

Képletek és feltételes formázás kapcsolata

A képletek és a feltételes formázás szorosan összefüggnek, mivel a feltételes formázás gyakran képleteken alapul.

Például a feltételes formázás lehetővé teszi, hogy egy cella színe megváltozzon, ha egy adott képlet igaz értéket ad vissza.

Példa:

Képlet: =A1 > 100

Feltételes formázás: Ha a fenti képlet igaz, akkor az A1 cella háttérszíne pirosra változik.

Ez lehetővé teszi az adatok gyors vizuális kiemelését és azonosítását a táblázatban, segítve ezzel a fontos információk gyors felismerését.

Adatbázis-kezelés

19. tétel: Az adatbázis-kezelés alapjai

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

Mit nevezünk adatbázisnak?

Az adatbázis egy strukturált gyűjteménye adatoknak, amelyeket úgy tárolnak, hogy azok hatékonyan elérhetők, kezelhetők és frissíthetők legyenek. Az adatbázisok lehetnek relációsak (táblázatos formában) vagy nem relációsak (pl. dokumentum-orientált, grafikus adatbázisok).

Mit nevezünk adattáblának?

Az adattábla az adatbázis alapvető szerkezeti egysége, amely sorokból (rekordok) és oszlopokból (mezők) áll. minden oszlop egy adott adattípust és mezőt képvisel, míg minden sor egy egyedi rekordot tartalmaz.

Hogyan kapcsolódhatnak egymáshoz az adattáblák?

Az adattáblák közötti kapcsolatok általában kulcsok segítségével jönnek létre. Ezek a kapcsolatok lehetnek:

Egy-egy (One-to-One): Egy rekord csak egy másik rekordhoz kapcsolódik egy másik táblában.

Egy-több (One-to-Many): Egy rekord több rekordhoz kapcsolódik egy másik táblában.

Több-több (Many-to-Many): Több rekord több rekordhoz kapcsolódik egy másik táblában, amelyhez kapcsolótáblára van szükség.

Mit nevezünk mezőnek, rekordnak, adatnak?

Mező (Field): Egy oszlop egy adattáblában, amely egy adott típusú adatot tartalmaz. Példa: egy "Név" mező a dolgozók nevét tárolja.

Rekord (Record): Egy sor egy adattáblában, amely egy adott entitás összes adatát tartalmazza. Példa: egy dolgozó összes adata egy sorban.

Adat (Data): Az egyes mezők értékei a rekordokban. Példa: egy dolgozó neve "Kovács János".

Milyen kulcsokat ismersz? Ismertesd őket!

Elsődleges kulcs (Primary Key): Egy egyedi mező vagy mezők kombinációja, amely egyértelműen azonosítja az adattábla minden rekordját. Példa: egy dolgozói azonosító (ID).

Idegen kulcs (Foreign Key): Egy mező vagy mezők kombinációja, amely kapcsolatot hoz létre két adattábla között, hivatkozva az egyik tábla elsődleges kulcsára. Példa: egy dolgozói osztály azonosítója, amely az osztályok táblájának elsődleges kulcsára hivatkozik.

Külső kulcs (External Key): Hasonló az idegen kulcshoz, de külső kapcsolatok létrehozására használják több adatbázis között.

Mit nevezünk kapcsolótáblának?

A kapcsolótábla egy speciális adattábla, amely több-több kapcsolatok létrehozására szolgál két vagy több másik adattábla között. Ez a tábla legalább két idegen kulcsot tartalmaz, amelyek az összekapcsolt táblák elsődleges kulcsaira hivatkoznak. **ACCESS**

Példa:

Hallgatók tábla: **hallgató_id**, név

Kurzusok tábla: **kurzus_id**, **kurzus_név**

Hallgató_kurzus kapcsolótábla: **hallgató_id**, **kurzus_id**

Milyen adattípusokat használunk adatbázis-kezelőben?

Szöveg (Text/String): Betűk, számok és egyéb karakterek. Példa: VARCHAR, CHAR.

Szám (Number): Egész számok és tizedes számok. Példa: INTEGER, FLOAT, DECIMAL.

Dátum és idő (Date and Time): Dátumok és időpontok tárolására. Példa: DATE, TIME, DATETIME, TIMESTAMP.

Logikai (Boolean): Igaz vagy hamis értékek. Példa: BOOLEAN.

NULL: Hiányzó vagy ismeretlen érték jelölésére.

Példák adattípusokra:

VARCHAR(255): Szöveges adat legfeljebb 255 karakter hosszúságban.

INT: Egész szám.

DATE: Dátum formátumban.

BOOLEAN: Igaz/hamis érték.

20. tétel: SQL utasítások és függvények

Ismertesd az adatbázis létrehozásához szükséges utasítást! Ismertesd a táblák importáláshoz 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 SQLben? Mit tudsz az allekérdezések ról?

Adatbázis létrehozásához szükséges utasítás

Az adatbázis létrehozásához az SQL **CREATE DATABASE** utasítást használjuk.

Szintaxis:

CREATE DATABASE adatbazis_neve;

Példa:

CREATE DATABASE tanuloi_adatbazis;

Táblák létrehozásához szükséges SQL utasítások

A táblák létrehozásához az SQL **CREATE TABLE** utasítást használjuk.

Szintaxis:

CREATE TABLE tabla_neve (oszlop1_neve adat_tipus, oszlop2_neve adat_tipus, ...);

Példa:

CREATE TABLE hallgatok (hallgato_id INT PRIMARY KEY, nev VARCHAR(100), születési_dátum DATE);

Táblák bővítéséhez, frissítéséhez, törléséhez szükséges utasítások

Táblák bővítése (INSERT):

Szintaxis:

INSERT INTO tabla_neve (oszlop1, oszlop2, ...) **VALUES** (ertek1, ertek2, ...);

Példa:

INSERT INTO hallgatok (hallgato_id, nev, születési_dátum) **VALUES** (1, 'Kovács János', '2000-05-20');

Adatok frissítése (UPDATE):

Szintaxis:

UPDATE tabla_neve **SET** oszlop1 = ertek1, oszlop2 = ertek2, ... **WHERE** feltétel;

Példa:

UPDATE hallgatok **SET** nev = 'Nagy Anna' **WHERE** hallgato_id = 1;

Adatok törlése (DELETE):

Szintaxis:

DELETE FROM tabla_neve **WHERE** feltétel;

Példa:

DELETE FROM hallgatok **WHERE** hallgato_id = 1;

Lekérdezések szintaxisa

A lekérdezésekhez az SQL **SELECT** utasítást használjuk.

SELECT: a megjelenő mezők kiválasztása

FROM: a lekérdezésben használt táblák megadása

WHERE: szűrési feltételek megadása

Szintaxis:

SELECT oszlop1, oszlop2, ... FROM tabla_neve WHERE feltétel;

Példa:

SELECT nev, születési_dátum FROM hallgatók WHERE hallgató_kor > 20;

SELECT DISTINCT

Egyéb záradékok

SELECT DISTINCT: az eredménytábla rekordjai egyediek lesznek

GROUP BY: csoportosítja az adatokat a megadott mező szerint, kategóriákat hoz létre a mező adataiból.

HAVING: csoportosítás utáni feltételek megadására használható, általában összesítő függvények eredményeire a feltételt

ORDER BY: rendezési szempontokat lehet a segítségével beállítani

Aggregáló függvények

Az aggregáló függvények csoportos műveleteket hajtanak végre az adatokon.

COUNT(): Az elemek számát adja vissza.

SELECT COUNT(*) FROM hallgatók;

SUM(): Az értékek összegét adja vissza.

SELECT SUM(jövedelem) FROM alkalmazottak;

AVG(): Az értékek átlagát adja vissza.

SELECT AVG(jövedelem) FROM alkalmazottak;

MAX(): A legnagyobb értéket adja vissza.

SELECT MAX(jövedelem) FROM alkalmazottak;

MIN(): A legkisebb értéket adja vissza.

SELECT MIN(jövedelem) FROM alkalmazottak;

Dátum- és időfüggvények

YEAR(): Egy dátumból az évet adja vissza.

SELECT YEAR(datum);

HOUR(): Egy időpontból az órát adja vissza

SELECT HOUR(datum);

Hasonlóképpen működnek az alábbi függvények:

MONTH(), DAY(), MINUTE(), SECOND()

Szövegek kezelése SQL-ben

CONCAT(): Két vagy több szöveg összefűzése.

```
SELECT CONCAT(nev, ' ', vezeteknev) AS teljes_nev FROM hallgatok;
```

UPPER(): Szöveg nagybetűssé alakítása.

```
SELECT UPPER(nev) FROM hallgatok;
```

LOWER(): Szöveg kisbetűssé alakítása.

```
SELECT LOWER(nev) FROM hallgatok;
```

SUBSTRING(): Szöveg egy részének kivágása.

```
SELECT SUBSTRING(nev, 1, 3) FROM hallgatok;
```

LENGTH(): Szöveg hosszának meghatározása.

```
SELECT LENGTH(nev) FROM hallgatok;
```

Allekérdezések

Az allekérdezések olyan lekérdezések, amelyeket másik lekérdezésbe ágyazunk.

Példa egyszerű allekérdezésre:

```
SELECT nev FROM hallgatok WHERE hallgato_id = (SELECT MAX(hallgato_id) FROM hallgatok);
```

Példa összetett allekérdezésre:

```
SELECT nev FROM hallgatok WHERE hallgato_id IN (SELECT hallgato_id FROM kurzus_hallgatok WHERE kurzus_id = 1);
```

Az allekérdezéseket gyakran használják összetett adatlekérdezésekhez és -szűrésekhez, valamint adatok elemzéséhez.