

A kiadvány 2020. 06. 10-tól 2025. 08. 31-ig tankönyvi engedélyt kapott a TKV/3168-1/2020. számú határozattal. A tankönyv megfelel a Kormány 5/2020 (I.31.) Korm. rendelete a Nemzeti alaptanterv kiadásáról, bevezetéséről és alkalmazásáról szóló 110/2012. (VI.4.) Korm. rendelet módosításáról megnevezésű jogszabály alapján készült Kerettanterv az általános iskola 5. évfolyama számára megnevezésű kerettanterv Digitális kultúra tantárgy előírásainak.

A tankönyvvé nyilvánítási eljárásban közreműködő szakértő: Kusper László

Tananyagfejlesztők: Lénárd András, Abonyi-Tóth Andor, Turzó-Sovák Nikolett, Varga Péter
Kerettantervi szakértő: Lénárd András

Lektor: Farkasfalvy Judit

Szerkesztő: Pintér Gergely

Fedélterv: Slezák Ilona

Fotók: Shutterstock (9, 34 AlesiaKan; 10 Radu Bercan, 25 urbanbuzz; 64 Rose Carson), Cultiris (37), Demeter László (52, 55), iStock (38, 40, 41, 55), Wikimedia (51)

© Oktatási Hivatal, 2020

ISBN 978-615-81539-4-2

Oktatási Hivatal

1055 Budapest, Szalay utca 10-14.

Telefon: (+36-1) 374-2100

E-mail: tankonyv@oh.gov.hu

A kiadásért felel: Brassói Sándor elnök

Raktári szám: OH-DIG05TA

Tankönyvkiadási osztályvezető: Horváth Zoltán Ákos

Műszaki szerkesztő: Kurucz Klára

Nyomdai előkészítés: WOW Stúdió Kft.

Terjedelem: 7,5 (A/5) ív, tömeg: 230 gramm

1. kiadás, 2024

Gyártás: Könyvtárellátó Nonprofit Kft.

Ez a tankönyv a Széchenyi 2020 Emberi Erőforrás Fejlesztési Operatív Program EFOP-3.2.2-VEKOP-15-2016-00001 számú, „A köznevelés tartalmi szabályozóinak megfelelő tankönyvek, taneszközök fejlesztése és digitális tartalomfejlesztés” című projektje keretében készült. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Nyomtatta és kötötte:

Felelős vezető:

A nyomdai megrendelés

törzsszáma:

SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFECTETÉS A JÖVŐBE

Előszó	5
Robotika, algoritmizálás, programozás	7
Bevezetés a robotikába	7
Készítsünk algoritmust!	9
Ismerkedés a blokkprogramozási környezettel	11
A robot vezérlése	15
Gyakorlás, saját ötletek megvalósítása	17
A robot irányítása utasítások segítségével	19
A fal érzékelése	21
Gyakorlás, saját ötletek megvalósítása	23
Programozzunk micro:biteket!	25
Ismerkedés a felülettel	26
Mire ügyeljünk?	27
Készítsünk animációt!	28
Használjuk az érzékelőket!	29
Készítsünk egy játékot!	31
Gyakorlás, saját ötletek megvalósítása	33
Bemutatókészítés, multimédiás elemek készítése	35
Első bemutatónk	35
A képeket kísérő szövegek	38
Rajzok a bemutatóban	42
Animációk	47
Bemutatók fényképrészletekkel	51
Animációk: mozgásvonalak	58
e-Világ és online kommunikáció	63
Információs társadalom	63
Keresés a világhálón	65
A virtuális személyiség	69
Online kommunikáció	71
Közösségi oldalak, chat	73
Felhőszolgáltatások	75
Rendszerezés, összefoglalás	77
A digitális eszközök használata	79
A számítógépek és főbb alkatrészeik	79
Programok, operációs rendszerek, fájlok és mappák	82

Digitális kultúra 5. okostankönyv:



Digitális kultúra 5–8. okosgyűjtemény:



Kedves Ötödikes!

Bizonyára Neked is feltűnt, hogy mostanában elég nehéz úgy bemenni egy szobába, terembe, hogy ne legyen ott valamilyen számítógép. Igen, ott vannak, még ha nem is mindig számítógép alakúak, hanem okostelefonnak, televízióknak, tabletnek vagy épp mosógépnek vannak álcázva. Igencsak sok mindenben lapul számítógép.

Tehát körbe vagyunk véve számítógépekkel. Ha körbe vagyunk véve számítógépekkel, akkor használjuk is a gépeket, a nap minden órájában. Ha használjuk is őket, miért ne akaránk jól használni? Ezek szerint érdemes megtanulni a számítógépeket *igazán jól* használni?! De még mennyire!

Ez a könyv épp ebben segít neked! Ezért húzza az iskolatáskádat. Ezért foglalja a helyet a padodban a sokkal fontosabb dolgok (tízórai, miegymás) előtt. Azért dolgozott sok tanár sokat a megírásával, hogy a számítógép ne valami gonosz varázslat, hovatovább boszorkányság legyen számodra, hanem egy olyan eszköz, amelynek érted a működését, és amelyet örömmel és ügyesen használsz.

A számítógéppel annyi mindent lehet csinálni, hogy szinte elmondani sem lehet. Jelen sorok írója már harmincöt éve számítógépezik (nem folyamatosan, néha picit abbahagyja), és még mindig nem unta meg! Mi idén – **veled együtt, kedves Ötödikes** – először is megnézzük, miként programozható egy számítógép, írunk egyszerűbb játékot, és robotoknak parancsolgatunk. Ha ezzel megvagyunk, megtanulunk lenyűgözően szép és érdekes bemutatókat készíteni, fényképeket átalakítani. Mikor már ez is megy, jól szétréznünk az interneten, megismerve annak fényes és sötét oldalát is. Év vége felé pedig megnézzük, hogy mi lakik egy számítógép belsejében.

Hát ilyen lesz ez a könyv, és ha már úgyis erről beszélünk, megemlítjük, hogy a könyvhöz tartoznak letölthető fájlok, amiket például a programjaidhoz tudsz felhasználni, vagy az elkészülő bemutatóidba kerülnek majd bele. A <https://www.tankonyvkatalogus.hu/site/kiadvany/OH-DIG05TA> weboldalt megnyitva találod meg őket.

Hogy sok minden kimaradt? Igen, teljesen igazad van, de majd hatodikos korodban is várunk, hogy újabb részeit fedezhessük fel a bennünket körülvevő digitális világnak!



Bevezetés a robotikába

Ebben a fejezetben robotok irányításával, illetve programozásával foglalkozunk. Megismerkedünk a kapcsolódó fogalmakkal, illetve átismételjük ezeket.



Bizonyára már te is láttál robotot, vagy hallottál róluk, láttad a filmekben, híradásokban. De vajon minden olyan eszköz robotnak tekinthető, amelynek a nevében szerepel a robot szó? Robot-e például a konyhai robotgép? Egyáltalán melyek azok a tulajdonságok, amelyek alapján azt mondhatjuk egy eszközről, hogy robot? A következőkben ezzel foglalkozunk.

Feladat

Gyűjtsük össze, hogy eddigi tanulmányaink és tapasztalataink alapján mit tudunk a robotokról! Írjuk rá egy cetlire egy (tényleg létező) robot nevét! Egy másik cetlire pedig írjuk fel azt, hogy ez a robot milyen tevékenységet végez! Helyezzük el a cetliket a tábla megfelelő részein, és mondjuk el hangosan a társainknak a választásunkat! Ha mindenki végzett, beszéljük meg, hogy milyen szempontok szerint lehetne csoportosítani az összegyűjtött robotokat, valamint a cselekvéseket!



A közös ötletelés során bizonyára sokféle eszközt és tevékenységet sikerült összegyűjteni. Ezen tevékenységek lehetnek egyszerűek, de akár nagyon bonyolultak is. Ennek oka, hogy a technológia fejlődésével a robotok is egyre többféle feladatot tudnak ellátni. Az alapján, hogy a robotok milyen jellegű feladatok ellátására képesek, különböző generációkba sorolhatjuk őket.

Robotgenerációk

Az **első generációs robotok** az 1960-as években jelentek meg, és egyszerű, például tárgyak mozgatásával kapcsolatos feladatokat oldottak meg. Ezeknek a robotoknak még nem voltak érzékelők, így még nem tudtak a környezetük jellemzőire, változásaira reagálni. A **második generációs robotok** az 1970-es évekre jellemzőek, és ezek a robotok már érzékelőkkel lettek felszerelve.

Az **érzékelők**, más néven **szenzorok** teszik lehetővé azt, hogy a robot érzékelje a környezetét, és az információk kiértékelése alapján döntéseket hozhasson, más-más cselekvéseket hajthasson végre.

▶ Egy távolságérzékelő szenzor

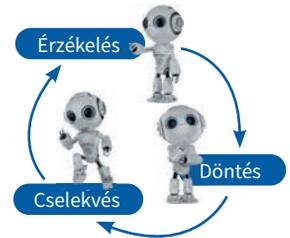


Feladatok

1. Gyűjtsük össze közösen, milyen külvilágból származó információkat érzékelhetnek a szenzorok! Például hőmérséklet, testhő stb.
2. Ötleteljünk azon, hogy a saját iskolánkban milyen feladatot lehetne rábízni egy második generációs robotra! Alkossunk csoportokat, és tervezzünk meg egy robotot! Gyűjtsük össze, hogy milyen érzékelőkkel kellene felszerelni a robotot, hogy elvégezhesse a feladatát!

A **harmadik generációs robotok** már igen fejlettek, nemcsak szenzorokkal rendelkeznek, hanem sokkal okosabbak, mint az elődeik. Akár a korábbi tapasztalataik alapján is képesek tanulni (gépi tanulás), szabályszerűségeket megállapítani. Azt, hogy ilyen bonyolult feladatokra is képessé váltak a robotok, a **mesterséges intelligencia** területén elvégzett kutatások tették lehetővé.

Napjaink korszerű robotjaival szemben tehát **alapvető elvárás**, hogy **érezkeljék** a környezetük jellemzőit, a környezetben bekövetkezett változásokat. Ezek kiértékelése alapján pedig **döntést** kell hozniuk, és az alapján **cselekedniük** kell.



Milyen méretű lehet egy robot?

A robotok nemcsak a tudásuk, hanem a méretük szerint is nagyon változatosak lehetnek. Vannak egészen kicsi (mikro- és nano-) robotok, amelyeket például az orvostudományban alkalmaznak. Képesek lehetnek az emberi test keringési rendszerének feltérképezésére, vagy akár gyógyszer eljuttatására a test megfelelő pontjaiba.

A nagyobb robotok közé tartozik például az az autó méretű marsjáró robot, amelyet a NASA 2011-ben bocsátott fel a Mars felületének megvizsgálására. A robot annyira fejlett volt, hogy a fúrójával mintát tudott venni a kőzetekből, azok összetételét megvizsgálta, és az adatok kiértékelése után megállapította a kőzetek korát.



► Ipari robot egy süteménygyárban



► Robot a Mars bolygón (Curiosity marsjáró)

Feladatok

1. Vitassuk meg, hogy miért nagyon fontos elvárás az a marsjárótól, hogy önállóan képes legyen a kutatási feladatok elvégzésére! Miért nem elegendő az, hogy a Földről irányítsák a robot mozdulatait a mérnökök és kutatók?
2. Gyűjtsük össze, hogy melyik volt a legkisebb, illetve legnagyobb létező robot, amit saját szemünkkel vagy akár dokumentumfilmekben láttunk!

Készítsünk algoritmust!

Ahhoz, hogy a robotokat irányítani, vezérelni tudjuk, ismernünk kell, hogy a robot milyen cselekvésekre képes, illetve milyen utasításokat képes végrehajtani.

Az irányítás kipróbálásához használhatunk egy *virtuális* (fizikai értelemben nem létező, számítógépes programban elérhető) robotot, de akár valós, kézzel fogható oktatási robotot is.

Kezdetben csak arra van szükségünk, hogy a robot képes legyen előrehaladni megadott távolságot, és balra, illetve jobbra fordulni 90 fokkal. A robot lehet akár egy virtuális, számítógépes programban megvalósított robot is.

A **robot vezérlése** leegyszerűsítve azt jelenti, hogy a robot pontosan az általunk kiadott utasításokat hajtja végre. A robot mozgása ilyenkor csak a kiadott parancsoktól függ, és a robot nem végez semmilyen korrekciót. Ha például egy hibás utasítás miatt egy falnak vezetjük a robotot, a kerekei nem állnak le az akadályt érzékelve, hanem folyamatosan működnek tovább. Emiatt akár könnyen meg is sérülhetnek. Ezért is fontos a megoldandó feladat alapos tervezése!

Tervezzük meg, mit fog csinálni a robotunk!

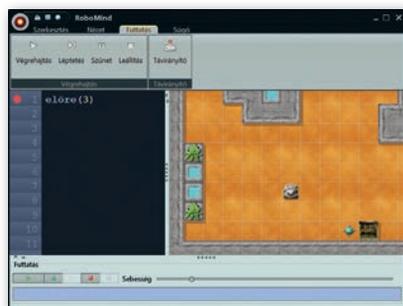
Kezdjünk egy egyszerű példával! A robotot ki kell vezetnünk egy labirintusból. Azt tudjuk, hogy a robot képes előremenni egy távolsággal, jobbra és balra fordulni. A robot kezdetben észak felé néz, vagyis a felülnézeti képen felfelé irányban fog elmozdulni.

Milyen utasításokat kell kiadnunk ahhoz, hogy a robot a kezdeti helyéről eljusson a jobb felső sarokban lévő kijáratig?

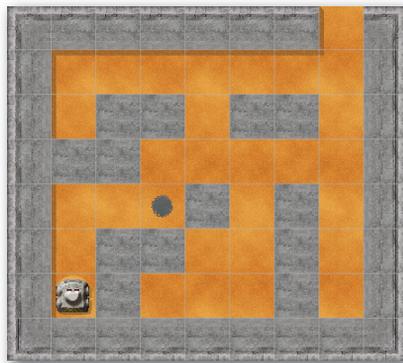
A feladat megoldásához először meg kell terveznünk az egyes lépéseket. Ekkor még nem kell egészen pontosan tudnunk azt, hogy a robot milyen parancsokat tud végrehajtani. Egyszerűen, **mondatszerű elemekkel** írjuk le, hogy milyen lépésekkel oldható meg a feladat.



▶ LEGO Mindstorms EV3 robot



▶ Egy robotszimulátor-program (RoboMind) felülete



▶ A labirintus, amelyből ki kell juttatni a robotot

Ha például a szürke ponttal jelölt mezőre akarnánk eljuttatni a robotot, gondolkodhatnánk így:

Program

Menj előre 2 lépést

Fordulj jobbra

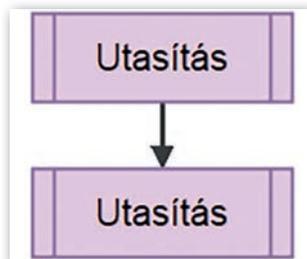
Menj előre 2 lépést

Program vége

Amit most készítettünk, az egy **algoritmus**. Ez nem más, mint a feladat megoldásához szükséges elemi utasítások sorozata.

Ebben az algoritmusban egyszerű tevékenységek, utasítások sorozatát írtuk le, amelyek egymás után hajtnak végre. Ezt más néven **szekvenciának** nevezzük.

Az algoritmust kezdetben szövegesen, mondatszerű leírással adjuk meg. Az algoritmusokat nagyon gyakran grafikusán, **folyamatábra** segítségével is le szokták írni.



► Folyamatábra részlete egy szekvenciával (utasítások sorozatával)

Fontos tudni, hogy az algoritmusokkal nemcsak az informatika területén találkozhatunk, hanem a hétköznapi életben is. Amikor például egy útmutató alapján összerakunk egy építőjátékot, akkor is egy algoritmus szerint dolgozunk.



Feladatok

1. Szervezdjünk három-négy fős csoportokba! Minden csoport kap egy fogalmat, amelyhez kapcsolódóan egy hétköznapi tevékenységhez kell algoritmust készítenie és ismertetnie. A fogalmak: *ital/étel automata, sport, társasjáték, tisztálkodás, utazás, állattartás, szülinapi buli, osztálykirándulás, hírek megosztása, útba igazítás*. Az egyes algoritmusok ismertetésekor beszéljük meg, hogy milyen előfeltételek szükségesek ahhoz, hogy az algoritmus valóban működjön.
2. Készítsük el önállóan azt az algoritmust, amellyel a robot kivezethető a labirintusból! Párokban ellenőrizzük le egymás megoldásait! Valóban sikerül a megadott lépésekkel kijuttatni a robotot az útvesztőből? Mi lehet az oka annak, ha nem egyezik meg (egyéb-ként helyes eredményt adó) két megoldás?

Csoportos feladat

Mennyire egyértelműek az utasítások?

Menjünk olyan helyszínre, ahol nagyobb szabad tér van! Ezen a téren jelöljünk meg egy kiindulópontot és egy végpontot!

Alkossunk 4-5 fős csoportokat!

Minden csoportban legyen egy olyan diák, akinek bekötjük a szemét. Készítsük el azt az algoritmust, amiről úgy gondoljuk, hogy elvezeti a társunkat a kiindulási ponttól a célig! Adjuk ki az algoritmusnak megfelelő szóbeli utasításokat a társunknak! Mit tapasztalunk? Tényleg eljutott a célhoz? Most adjuk ki az utasításokat úgy, hogy a társunk pontosan visszafelé járja be az útvonalat. Visszajutott a kezdőpontig?



Vitassuk meg, hogy a játék során milyen fontos tanulságokat szereztünk az irányítással kapcsolatban! Hogyan lehetett volna egyértelműbbé tenni az utasításokat? Ötleteljünk arról, hogy az igazi robotok irányítása mennyiben lesz más, illetve hasonló, mint amit ebben a játékban tapasztaltunk!

Ismerkedés a blokkprogramozási környezettel

Egy robot irányításának kipróbálására, szimulálására sokféle környezet alkalmas. Elsőként a blokkprogramozási környezetekkel ismerkedünk meg, amelyekben az egyes utasításokat és vezérlőszerkezeteket blokkok jelképezik. Ezeket a blokkokat nagyon egyszerűen egymáshoz illeszthetjük, így előállítva a programot.



▶ Blokkok különböző programozási környezetekben

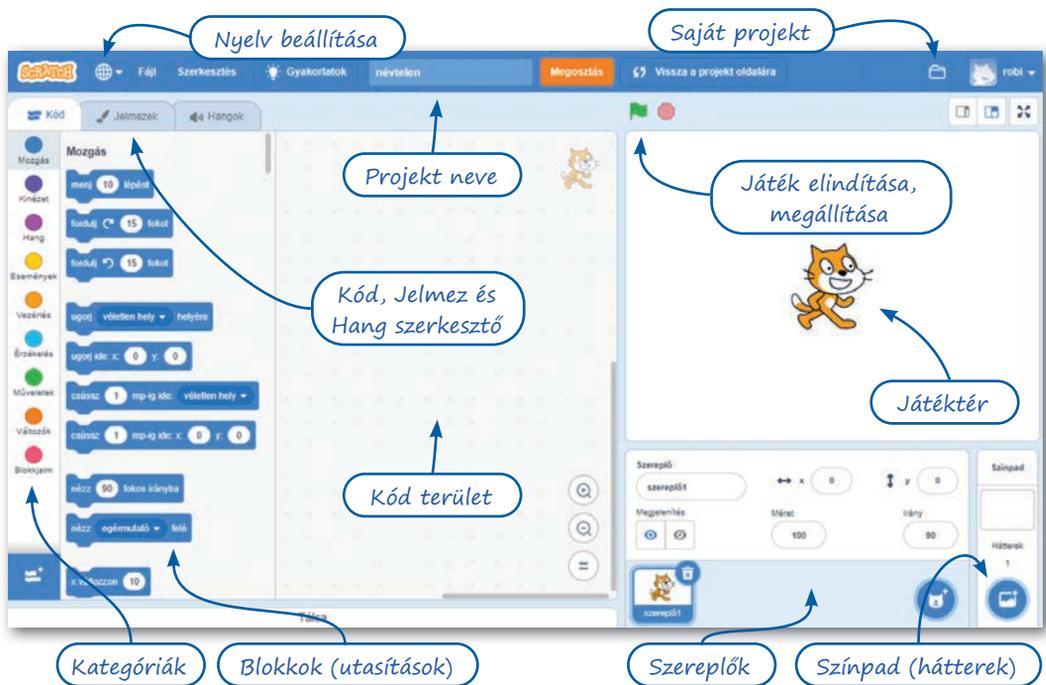
Feladat: Beszéljük meg, hogy az itt szereplő blokkoknak mi lehet a jelentésük! Vajon milyen esetben fognak ezek végrehajtódni? Mi lehet a kiadott utasítások eredménye?

A **blokkprogramozási környezetek** között találunk olyanokat, amelyeknek van letölthető, **számítógépre telepíthető** változatuk, vagy önálló alkalmazásként használhatók okostelefon/tableten, de sok esetben nem is szükséges telepítenünk az alkalmazást, mert online módon, böngészőprogramban is használhatjuk azt. Egyes környezetek pedig többfajta lehetőséget is biztosítanak a felsoroltak közül, vagyis akár telepíthetőek is, de böngészőprogramban is használhatóak.

A **blokkprogramozási fejlesztői környezet** minden olyan eszközt biztosít számunkra, amelyre szükség van a programozás során. Ebbe beletartoznak maguk a blokkok, a különböző beállítási lehetőségek (pl. milyen nyelvű legyen a felület), futtatási és nyomkövetési (tesztelési) lehetőségek, a programok elmentésének és betöltésének lehetőségei, sőt egyes esetekben az elkészült munkákat másokkal is egyszerűen megoszthatjuk a felület segítségével.

A programozási környezet részei

A blokkprogramozási környezetek felületében közös, hogy a programot egy **munkaterületen** vagy **kódrészleten** kell összeállítani a különböző blokkokból. A blokkok kategóriák szerint csoportosítva vannak. A képernyőn megjelenő szereplő kinézetét testre lehet szabni, vagy úgy, hogy mi rajzolunk meg egy alakot, vagy úgy is, hogy felhasználunk egy már létezőt. Szintén beállíthatjuk azt, hogy a háttérben milyen kép legyen elhelyezve. Az összeállított programot le tudjuk futtatni, és akár meg is tudjuk állítani.

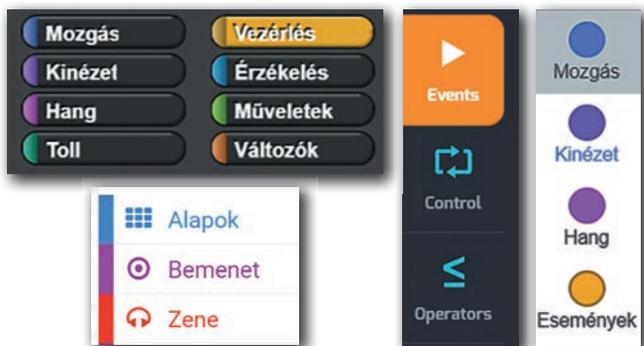


► Példa egy blokkprogramozási fejlesztői környezetre (Scratch)

Kategoriák a blokkok csoportosításához

A blokkok különböző **kategoriákba** vannak besorolva attól függően, hogy milyen célra lehet használni őket.

Ilyen kategoriák lehetnek például a mozgás, kinézet, zene, események, műveletek, toll stb.



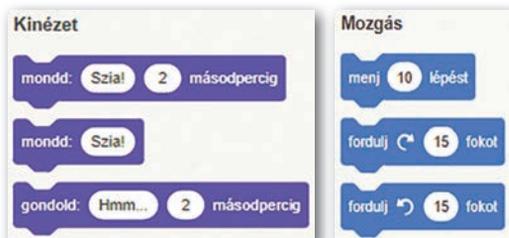
► Blokkok kategoriái a különböző blokkprogramozási környezetekben

A használható blokkok áttekintése

A kategoriákat jellemzően különböző színekkel különböztetik meg. Az azonos csoportokba tartozó blokkok azonos színűek.

Vagyis a blokk színe a blokk szerepére is utal.

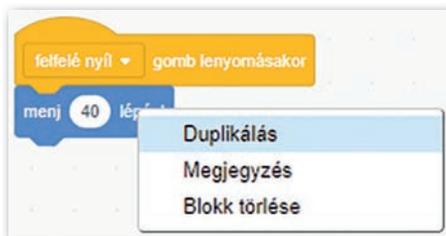
► A blokkok ugyanolyan színűek, mint a kategória (Scratch környezet)



Blokkok másolása, törlése

A munkaterületen elhelyezett blokkokból akár másolatokat is készíthetünk, ha újra fel akarjuk használni őket. Általában ehhez a jobb egérgombbal kell a blokkra kattintani, majd a menüből kiválasztani a duplikálás (megkettőzés) menüpontot.

Ugyanebben a menüben található a törlési lehetőség is. A törlés általában úgy is működik a programozási környezetekben, hogy a blokkot megragadjuk, és visszahúzzuk arra a területre, ahonnan kiválasztottuk.



- ▶ Blokk másolatának készítése (duplikálás)

Program végrehajtása, futtatása

A programot végre is tudjuk hajtatni (futtatni tudjuk). Ennek hatására a programozási környezetben láthatjuk a program eredményét. Ez az eredmény lehet például az, hogy elmozdul egy szereplő a képernyőn, de egy igazi robot esetén az is lehet az eredmény, hogy a robot ténylegesen elindul, és megtesz egy bizonyos távolságot.

A program nem biztos, hogy valóban azt a feladatot oldja meg, amit elgondoltunk, mivel az algoritmus megírása és a kódolás során is követhetünk el hibákat. Ezért a programot **tesztelnünk** kell, hogy meggyőződjünk a helyességéről. A tesztelés során kapott eredményt **elemoznünk** kell annak érdekében, hogy a **hibajavítást** el tudjuk végezni. A hiba lehet magában a kódban is, de már a kigondolt algoritmus is tartalmazhatott hibákat, így vissza kell térnünk a megfelelő algoritmus kitalálásához.

Program mentése, betöltése

Az elkészült projektjeinket el tudjuk **menteni**, illetve a korábban elmentett projekteket **be tudjuk tölteni** a felületre. Így bármikor továbbfejleszhetjük a korábban elmentett alkalmazásokat.

Tipp: Amikor elmentjük a projektjeinket, adjunk nekik olyan beszédes nevet, amely alapján később is tudni fogjuk, hogy a program milyen célt szolgált.

A felület nyelve

Mivel ezek a környezetek főleg gyermekek számára készültek, a fejlesztők ügyeltek arra, hogy az **alkalmazás nyelve beállítható** legyen, így sok esetben magyar nyelvre is átállíthatóak.

A magyar nyelvű felületen kezdetben jobban kiismerhetjük magunkat, és az önálló felfedezést is segíti.



- ▶ Nyelv beállításának lehetősége (Scratch)

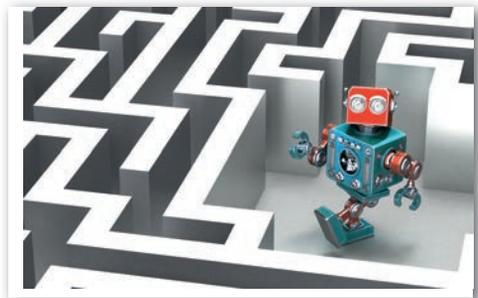
A robot és a pálya elkészítése

Még mielőtt valódi robotokat irányítanánk, érdemes az alapvezérlési és -programozási lehetőséggel megismerkedni olyan környezetben, ahol **szabadon kísérletezhetünk**, és tapasztalatokat gyűjthetünk. Egy valódi robotot mindig sokkal nagyobb körültekintéssel kell programozni, mint egy szimulált, virtuális robotot. Az **igazi robot akár meg is sérülhet**, vagy **sérülést okozhat**, ha például ráesik a lábunkra vagy nagy sebességgel nekünk ütközik.

A későbbiekben egy olyan projektet készítünk el, amelyben egy robotot fogunk kivezetni egy labirintusból.

Első lépésként készítjük el a pályát, és rajzoljuk meg a robotot!

A robotot olyan módon kell megszemélyesítenünk, hogy lássuk, melyik irányba néz. Így egyértelmű lesz, hogy milyen irányban halad előre a megfelelő utasítás kiadásakor.



Feladatok

1. Tekintsük át, hogy milyen beépített jelmezek állnak rendelkezésre a programozási környezetben!
2. Jelmezként rajzoljunk meg egy olyan egyedi robotalakot, amelyet látva eldönthető, hogy melyik irányba néz. Figyeljünk arra, hogy alapesetben milyen irányba néz a szereplő a környezetben (jobbra, felfelé stb.), és annak megfelelően rajzoljuk meg a robotot.

Mi az alábbi robotot rajzoltuk meg, amelynek két kereke van. A kis antennák jelzik, hogy éppen jobbra néz.



► Az általunk megrajzolt robot

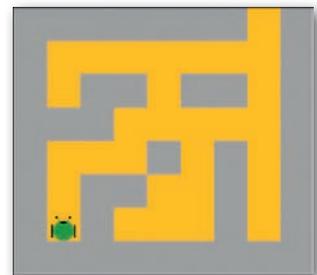
Ha elkészült a robotunk, folytassuk azzal, hogy elhelyezzük az üres labirintus képét a háttéren. Ez a kép megtalálható a letölthető állományok között.

Feladat: Töltsük le ezt a képet a számítógépre! Módosítsuk a robot szereplő helyzetét és méretét úgy, hogy a labirintus bal alsó mezőjére kerüljön, és felfelé nézzen!

A projektet mentjük el a fejlesztői környezetben, hogy később továbbfejleszthessük!



► A szereplő jelmezének beállítási lehetőségei a Scratch környezetben



► A labirintus beillesztése, a robot elhelyezése és átméretezése utáni állapot

A robot vezérlése

Az algoritmus elkészítése után izgalmas látni azt is, hogy az igazi (vagy a virtuális) robot hogyan hajtja végre a leírt lépéseket. Ehhez meg kell tanulnunk a robot nyelvén beszélni, vagyis kódolnunk kell!

A **kódolás** művelete azt jelenti, hogy az algoritmust egy **programozási nyelv** szabályai, a rendelkezésre álló utasítások szerint átfogalmazzuk.

A kódolás eredménye az adott programozási nyelven megfogalmazott **program**. Az így előállt programot már meg lehet próbálni **végrehajtani**, más néven **futtatni**.

A **kódolás nem** teljesen ugyanazt jelenti, mint a **programozás**. A programozás egy összetett folyamat. Nemcsak a kódolási tevékenységet foglalja magában, hanem sok más tevékenységet is: a megoldandó feladat precíz leírását (specifikálását), algoritmizálást, tesztelést, hibajavítást, a program továbbfejlesztését és így tovább.

Előzőleg megrajzoltuk a robotot, és elhelyeztük a pályát a háttérben. Itt az ideje annak, hogy a robot elkezdjen mozogni a pályán!

Lépjünk előre!

Feladat: Keressünk olyan blokkot a környezetben, amellyel előre lehet léptetni a robotot! Helyezzük el ezt a kódterületen!



Tip: A blokkprogramozási környezetek többségében, ha duplán kattintunk az elhelyezett blokkra, akkor az végre is hajtódik, így mindjárt látjuk annak hatását.

Láthatjuk, hogy az előrelépés mértékét a blokk belsejében elhelyezett ovális mezőben adhatjuk meg. Ide nemcsak egy konkrét szám kerülhet, hanem különböző műveletek eredménye is.

Feladat: Tekintsük át, hogy milyen műveletek érhetőek el a programozási környezetben. Próbáljuk ki azt a blokkot, amely véletlenszerűen meghatározott lépéssel lépteti előre a szereplőt!



► Előrelépés véletlenszerűen választott számmal a Scratch környezetben

A paraméter fogalma

A blokk belsejébe beírt számot **paraméternek** nevezzük. A paraméter értéke most azt befolyásolta, hogy hány lépést tett előre a robot. Ha az algoritmust mondatszerű leírással adjuk meg, a paramétert sokszor **kerek zárójel**ek közé tesszük. A folyamatábrában ugyanez a megadási mód látszik.



► Folyamatábra az előre utasítással

Feladat: Fedezzük fel önállóan, hogy mekkora számot kell megadni paraméterként ahhoz, hogy a labirintusban egy mezőnyi (egységnyi) távolságot haladjon előre a robot!

A **paraméter** nem mindig számot jelöl, **akár szöveg is lehet**. Jó példa erre az itt látható blokk. A szöveges paraméter adja meg, hogy milyen szöveg jelenik meg a szereplő mellett, a szám pedig azt, hogy hány másodpercig legyen látható a szöveg.



► Szöveg és szám is lehet paraméter

A próbálgatáson kívül máshogy is meghatározhatjuk, hogy mekkora távolságot kell előrelépni a megadott háttérképen. Ehhez tudnunk kell, hogy a háttérkép mekkora méretű. Ezt a méretet akár a programozási környezet is jelezheti, de az operációs rendszer beépített lehetőségeit használva is kideríthetjük.

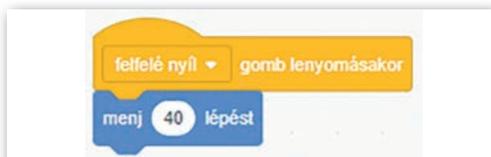
Tegyük fel, hogy a kép magassága 320 képpont. A kép függőlegesen most 8 mezőt tartalmaz, így egy mező $320 / 8 = 40$ képpont magasságú. Vagyis most 40 lépést kell előrelépnie a robotnak, hogy a megfelelő helyre kerüljön.



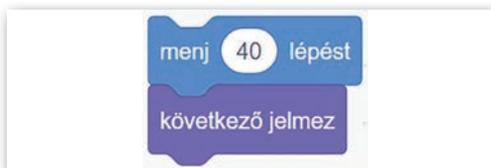
► A háttérkép méretének leolvasása (Scratch)

A robot vezérlése billentyűzetről

Most próbáljuk ki, hogy billentyűzet segítségével hogyan vezérelhetnénk a képernyőn a virtuális robotot! A blokkprogramozási környezetekben az **Események** vagy **Vezérlés** kategóriákban találjuk azokat a vezérlőblokkokat, amelyek lehetővé teszik, hogy az utasítások egy billentyű lenyomásakor hajtsódjanak végre.



► A felfelé nyíl hatására előrelép a szereplő 40 egységet (Scratch környezet)



► Jelmez váltása (Scratch környezetben)

A szereplők helyét (koordinátáját) le lehet olvasni a képernyőről. Az x koordináta azt jelöli, hogy vízszintesen melyik pozícióban helyezkedik el a szereplő. Az y koordináta a függőleges tengelyen lévő pozíciót jelöli.



► A szereplő helyzete, mérete, iránya (Scratch és Snap! környezetekben)

Feladat: Készítsük el azt a projektet, amelyben a felfelé nyíl megnyomásakor a robot előrelép 40 egységgel, és balra fordul 90 fokkal a balra nyíl billentyű megnyomásakor. A jobbra nyíl gomb hatására forduljon jobbra 90 fokkal. A szóköz billentyű megnyomásakor pedig kerüljön vissza a kiindulási helyére, a labirintus bal alsó mezőjébe.

Tipp: Ha egy szereplőnek több jelmezt is megrajzolunk, akkor a szereplő mozgás közben váltogatni tudja ezeket a jelmezeket egy megfelelő utasítás kiadásával.

Ha a programban szeretnénk megadni, hogy pontosan milyen koordinátára kerüljön a robot, akkor tipikusan a **Mozgás** kategóriában találjuk meg az erre szolgáló blokkokat.



► A szóköz billentyű lenyomásakor a szereplő visszatér a kiindulási állapotba, és felfelé néz. (Scratch környezet)

Gyakorlás, saját ötletek megvalósítása

Most már tudjuk azt, hogy hogyan hozhatunk létre szereplőt, hogyan állíthatunk be hátteret, és láttuk azt is, hogy billentyűzetről hogyan irányíthatjuk a szereplőt. Most itt az ideje, hogy önállóan is alkossunk!

Akár egyedül, akár párokban, akár csoportmunkában is sokféle érdekes program készíthető. Ha van kedved, folytasd otthon is, amibe belekezdted! Mutasd meg a szüleidnek, testvéreidnek is, hogy milyen programokat sikerült el készítened!



Projektmunka

1. Ötleteljünk azon, hogy az eddigi tudásunk alapján milyen táblás társasjátékot tudnánk megvalósítani a tanult programozási környezetben! Tervezzük meg a játékot, majd valósítsuk meg! Készítsük el a program algoritmusát, és csak utána álljunk neki a megvalósításnak! Törekedjünk arra, hogy mindenki kapjon olyan részfeladatot, amiért ő a felelős!
2. Ha elkészültünk a munkánkkal, mutassuk be egymásnak a játékokat, és próbáljuk ki egymás programjait!



Egyéni feladatok

1. A nyúl és teknős versenyéről szóló mesét szinte mindenki ismeri. Most valósítsuk meg ezt egy program segítségével! Rajzoljuk meg a szereplőket! Ha a nyúlra vagy a teknősre kattintunk, akkor lépjen előre egy véletlenszerűen meghatározott értéket! Állítsuk be úgy a véletlen szám paramétereit, hogy nagyobb eséllyel nyerjen a nyúl, mint a teknős! De készítsünk egy turbóteknőst is, amelyet nem győzhet le a nyúl!
2. Készítsünk egy olyan projektet, amelyben egy tetszőleges szereplőt lehet jobbra és balra léptetni a billentyűzet segítségével. A szereplő alakját mi találjuk ki és rajzoljuk meg! A szereplőnek legyen több jelmeze is, amit váltogatni tud a lépések során. Próbáljuk úgy elkészíteni a jelmezt, hogy úgy tűnjön, ténylegesen halad a szereplő (pl. egy pálcikaember esetén mozog a lába, egy autó esetén forog a kereke, és így tovább). Készítsük el hozzá a megfelelő hátteret is, amelyen szívesen elhelyeznénk a szereplőt!



Páros feladatok

1. Találjunk ki közösen egy történetet, amelyet különböző szereplőkkel el lehet játszani. Rajzoljuk meg a szereplőket és a szükséges háttereket is! A szereplők gombnyomásokra reagáljanak!
2. Készítsünk egy *Kérdezz-felelek* programot, amelyben a szereplő alakja mellett az Igen, Nem, Talán szövegek jelennek meg az I, N, T billentyűk lenyomásakor. A szereplő jelmeze is változzon meg a választól függően! A játékot próbáljuk ki párokban! Tegyük fel egymásnak eldöntendő kérdéseket, de ezekre ne mi, hanem a szereplők válaszoljanak a megfelelő gombok megnyomásával!



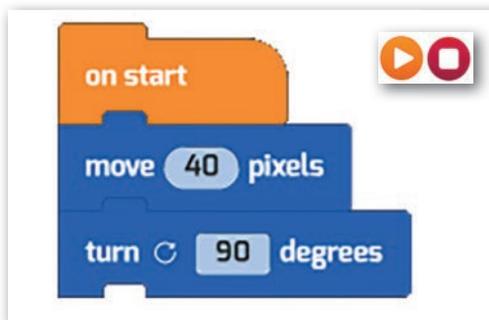
A robot irányítása utasítások segítségével

Korábban a billentyűzettel irányítottuk a robotot. Lépünk tovább, és írjuk meg a kijutáshoz szükséges programot!

Ha azt szeretnénk, hogy a robot önállóan menjen végig az útvonalon, akkor az utasításokat egy olyan vezérlőblokkban kell elhelyeznünk, amely a program indításakor automatikusan lefut. Ez a vezérlőblokk programozási környezetenként eltérő lehet.



- ▶ A Scratch és Snap környezetekben a zöld zászlós vezérlőblokk jelenti a főprogramot. A programot a zöld zászló megnyomásával indíthatjuk el, a piros nyolcszöggel pedig leállíthatjuk.



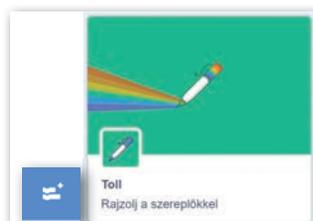
- ▶ A Tynker környezetben az On start (Induláskor) blokkban kell elhelyeznünk a főprogramot. A programot a háromszög ikonnal tudjuk elindítani, és a négyzet ikonnal tudjuk leállítani.

Feladat: Korábban elkészítettük azt az algoritmust, amely a robotot kivezeti a labirintusból. Most próbáljuk ki a megoldásunkat az általunk használt blokkprogramozási környezetben! A program indításakor a robot kerüljön vissza a kiindulási helyzetébe, nézzen felfelé, majd jelenjen meg mellette a „Szia” szöveg. Utána menjen el a kijáratig úgy, hogy nem érhet hozzá a falakhoz! A robot minden elfordulás után várakozzon 1 másodpercnyi időt, hogy nyomon lehessen követni a haladását. Amikor eljutott a robot a kijáratig, jelenjen meg mellette a „Kijutottam!” szöveg!

Merre járt a robot?

Ha meg tudnánk oldani azt, hogy a robot megjelölje, hogy mely mezőkön járt már, akkor a megtett útvonalat akár a program végétével, utólag is elemezni tudnánk.

A blokkprogramozási környezetek között sok olyat találunk, amelyekben a szereplők vonalat húzhatnak maguk után. Van olyan környezet is, amelyben egy bővítmény hozzáadása után tudjuk ezt a funkciót használni!



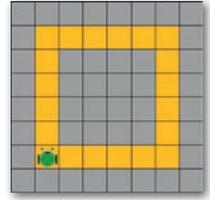
- ▶ Toll bővítmény használata (Scratch)

Feladat: Derítsük ki, hogy az általunk használt programozási környezetben hogyan lehetne megoldani, hogy a mozgása során egy vonalat húzzon maga után a szereplő! Módosítsuk úgy a korábban elkészült programot, hogy a robot húzzon egy vonalat maga után! A program újbóli elindításakor legyen letörölve az előzőleg meghúzott vonal, és a robot kerüljön vissza a kiindulási helyére!

Ismétlődő utasítások, ciklusok

Nézzünk egy másik pályát, amelyet a letöltendő állományok között megtalálunk! Ez nem egy labirintus, hanem egy négyzet alakú pálya. Most vezessük végig ezen a pályán a robotunkat!

A sárgával jelölt pálya hat egység hosszú és hat egység széles. A robot kiindulópontja legyen ismét a pálya bal alsó pontja, és nézzen a robot észak felé!



Ha végig akarjuk vezetni a pályán a robotot, akkor gondolkodhatnánk így:

Program

```
Menj előre 5 lépést
Fordulj jobbra
```

Program vége

Láthatjuk, hogy az algoritmus ismétlődő elemeket tartalmaz. Négyyszer ismételjük meg az öt egységgel történő előrelépést és az elfordulást. Ezt sokkal rövidebben is megfogalmazhatnánk:

Program

```
Ismételd 4 alkalommal
  Menj előre 5 lépést
  Fordulj jobbra
Ismétlés vége
```

Program vége

Az ismétlődő tevékenységeket úgynevezett **ciklusokba** érdemes szerveznünk. A ciklus belsejében – a **ciklusmagban** – helyezük el azokat az utasításokat, amelyeket ismételni kell.

Az algoritmus szöveges leírásában az *ismételd* szó jelöli a ciklus kezdetét, és az *ismétlés vége* a befejezését.

Ciklusokból többféle van. Kezdetben ismerkedjünk meg három fajtájával!

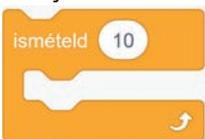
Ezeket általában a *Vezérlés* vagy *Ciklusok* kategóriában találjuk.

Számlálós ciklust akkor használunk, amikor ismerjük, hogy az utasításokat pontosan hány alkalommal kell végrehajtani (pl. *ismételd 3-szor ... ismétlés vége*).

Végtelen ciklust akkor alkalmazunk, ha az utasításokat állandóan ismételni akarjuk. Ilyenkor a program futását például úgy tudjuk leállítani, hogy megnyomjuk a *Leállítás* gombot (pl. *ismételd ... ismétlés vége*).

A **feltételes ciklusokra** az jellemző, hogy az utasítások ismétlése egy feltétel igaz vagy hamis voltától függ (pl. *ismételd amíg feltétel ... ismétlés vége*).

Programozási nyelvenként különbözhet, hogy a megadott feltétel a ciklus leállításának (kilépésnek) vagy a ciklus végrehajtásának feltétele.



▶ Számlálós ciklus



▶ Végtelen ciklus



▶ Feltételes ciklus

Feladatok: Oldjuk meg az alábbi feladatokat:

1. A robot pontosan háromszor menjen végig a körpályán!
2. A robot addig menjen a körpályán folyamatosan, amíg le nem állítjuk a programot.
3. A robot addig körözzön, amíg le nem nyomjuk a „v” billentyűt!

Az utóbbi feladat megoldásához önállóan kell felfedeznünk, hogy a feltételt hogyan kell megadni. Egy biztos: hatszög alakú blokkot kell keresnünk!

A fal érzékelése

Előző projektjeinkben a robot úgy viselkedett, mint egy első generációs robot. Csak a kiadott parancsokat hajtotta végre, nem tudott reagálni a környezetére. Folytatásként egy fejlettebb (második generációs) robot működését fogjuk szimulálni. Ez a robot már érzékeli, hogy szabad terület van-e előtte, így megtaníthatjuk majd arra, hogy elmenjen önállóan a falig.

Feladatok

1. Gondoljuk át, hogy ezen algoritmus alapján milyen útvonalat jár be a robot!
2. Páros munkában keressünk olyan kiindulási helyet a robotnak, amelyre igaz, hogy a fenti parancsok kiadásával szintén kijutna a labirintusból! Keressünk egy olyan kezdőpontot is, ahonnan nem tudna kijutni a robot ezen algoritmus alapján!
3. Van egy olyan útvonal, amely szintén elvezet a kijáratig, de most elhalad mellette a robot. Melyik ez az útvonal? Hogyan kellene módosítani az algoritmust, hogy ezen az útvonalon menjen el a kijáratig?

Program

```
Menj a falig
Fordulj jobbra
Menj a falig
Fordulj balra
Menj a falig
Fordulj jobbra
Menj a falig
Fordulj balra
Menj a falig
Program vége
```

▶ Az új algoritmus

Ahhoz, hogy továbbléphessünk, meg kell értetnünk a robottal azt, hogy a *menj a falig* művelet az elemi utasítások használatával (előre, jobbra, balra), hogyan hajtható végre. Vagyis ezt a műveletet elemi részekre kell bontanunk. Ezt nevezzük **lépésenkénti finomításnak**.

A **lépésenkénti finomítás** elve azt mondja ki, hogy a feladatokat olyan kisebb feladatokra (részfeladatokra) kell bontani, amelyeket már nem tudunk tovább bontani, vagy arra a kisebb feladatra már van működő megoldásunk.

Hogyan érzékeli a robot a környezetét?

Ahhoz, hogy a való világban egy robot érzékeli, hogy fal van előtte, szükség van valamilyen szenzorra (pl. távolságérzékelőre).

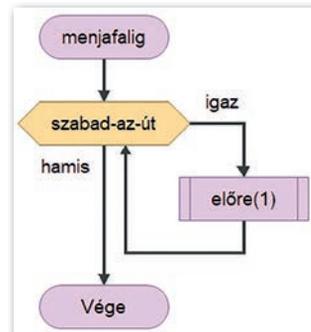
Az érzékelő által mért adatok kiértékelése után a robot már el tudja dönteni, hogy a vele szemben lévő terület szabad-e, vagy fal van rajta.

A *menj a falig* művelet azt jelenti, hogy addig kell előrelépkednie a robotnak, amíg a szemben lévő terület szabad. Ezt folyamatábrán szemléltetve látható, hogy a hatszög alakzat jelenti a feltételt (szabad-e az út?).

Az is szépen látszik, hogy milyen utasítások hajtnak végre, amikor a feltétel igaz, és mi történik, amikor hamis.

Ismételd amíg szabad az út
Menj előre 1 lépést
Ismétlés vége

- ▶ A *menj a falig* művelet algoritmus szövegesen



- ▶ Az algoritmus leírása folyamatábrára segítségével

A színérzékelő használata

A példánkban úgy modelleztük a labirintust, hogy a falat szürke színnel jelöltük, a szabad területeket pedig narancssárga színnel. Vagyis most a szín hordozza azt az információt, hogy haladhat-e előre a robot, vagy sem.

Emiatt most érdekesebb színérzékelőt használni a távolságérzékelő helyett. A különböző blokkprogramozási környezetekben esetenként más-más érzékelők használhatók, de a színérzékelő megléte nagyon gyakori.

A színérzékelő akkor ad vissza **igaz értéket**, ha a szenzor (robotunk) érint egy megadott színt, ellenkező esetben **hamis** értéket kapunk.

Mivel most a falunk szürke színű, a szürke szín beállításával fogjuk tudni ellenőrizni, hogy elértük-e a falat. Vigyázzunk! Szürke színárnyalatból nagyon sokféle lehet, ezért pontosan azt a színt kell beállítanunk a feltétel megfogalmazása során, amely a labirintus háttérképen megtalálható.

Feladatok

1. Páros munkában fedezzük fel, hogy az általunk használt blokkprogramozási környezetben hogyan valósítható meg a színérzékelés!
2. Készítsük el azt a programot, amelyben a felfelé nyíl megnyomásakor a robot egy ciklus segítségével addig lép előre, míg falba nem ütközik!
3. Mit tapasztalunk? Ha eljutott a falig a robot, tovább tud-e haladni egy másik irányba?

Fordulás előtt lépünk vissza!

Mivel a ciklus akkor áll le, amikor a szürke színt elérte a robot, ezért valószínűleg akkor is rá fog lógni valamelyik része a szürke színre, amikor elfordul valamelyik irányba és folytatná az útját.

Ezért a ciklus leállása után hátra kell lépünk valamekkora távolságot, hogy a robot semelyik része ne lógjon rá a szürke területre!

Azt, hogy pontosan mekkora távolságot kell visszalépni, függ attól, hogy milyen robotot rajzoltunk, és hogy például a robotot pontosan a rendelkezésre álló hely közepére rajzoltuk-e vagy sem.

Feladat

Teszteljük le, hogy mi az a legkisebb távolság a visszalépésre, ahol már jól fog működni a programunk, és állítsuk be paraméterként ezt az értéket!



- ▶ A szín kiválasztása Scratch környezetben. A pipetta ikonnal pontosan beállíthatjuk a vizsgálandó színt.

Gyakorlás, saját ötletek megvalósítása

Most már ismerjük azt is, hogy a különböző ciklusokat hogyan lehet használni. Nemcsak billentyűzetről, hanem utasítások alapján is tudjuk irányítani a szereplőket. Sőt, ezek a szereplők akár vonalakat is tudnak húzni, és érzékelhetik azt is, hogy milyen szint érintenek.

Ezek alapján újabb izgalmas programokat lehet megvalósítani.



Alkossunk együtt!

Alkossunk három-négy fős csoportokat! Tervezzünk meg egy olyan játékprogramot, amely azon alapul, hogy egy szereplőt (ami most ne robot legyen!) különböző akadályokon keresztül kell végigvezetni ahhoz, hogy teljesítse a küldetését.

A szereplőket és a háttérrel mi választhatjuk ki, illetve rajzolhatjuk meg. A játékprogramban kapjon szerepet a színérzékelő használata! Készítsük el a program algoritmusát, és csak utána álljunk neki a megvalósításnak. Törekedjünk arra, hogy mindenki kapjon olyan részfeladatot, amiért ő a felelős!

Ha elkészültünk a munkánkkal, mutassuk be egymásnak a játékokat, és próbáljuk ki egymás programjait!



Egyéni feladatok

1. Némelyik robot hangot is képes kiadni. Fedezzük fel önállóan, hogy milyen lehetőségek vannak hangok megszólaltatására az általunk használt környezetben! A robot induláskor és megérkezéskor más-más hangot adjon ki! Van-e esetleg olyan bővítmény a programhoz, amit érdemes lehet telepíteni?
2. Készítsünk olyan programot, amelyben ha egy robot a kék színt érzékeli, akkor jobbra fordulva folytatja az útját, ha pedig a sárgát, akkor balra fordul. Piros szín esetén forduljon vissza!

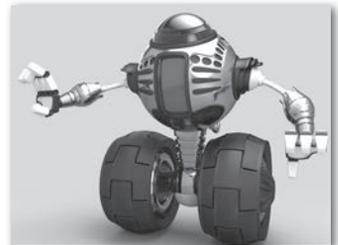
Páros feladat



Az igazi robotok kerekeit motorok mozgatják. Sajnos néha előfordulhat üzemzavar. Tegyük fel, hogy a robotunknak két kereke van, és az egyik kerekét forgató motort nem lehet kikapcsolni, ezért a kerék állandóan a maximális fordulattal forog.

A másik motor teljesítményét tudjuk állítani, de így a robot vagy csak egyenesen tud haladni, vagy jobbra tud fordulni, balra nem.

Pármunkában vitassuk meg, hogy egy hibás működésű (balra fordulásra képtelen) robot el tudna-e jutni a kijáratig a korábban bemutatott pályán! Ha igen, hogyan változik az algoritmus és a programkód ebben az esetben? Próbáljuk ki a gyakorlatban!



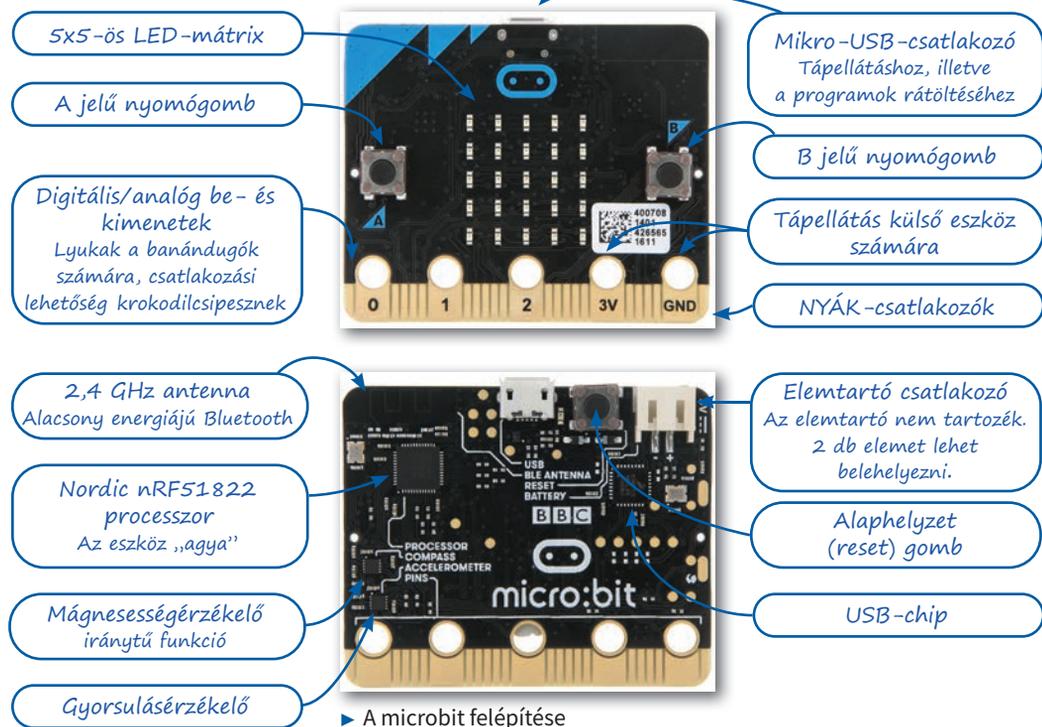
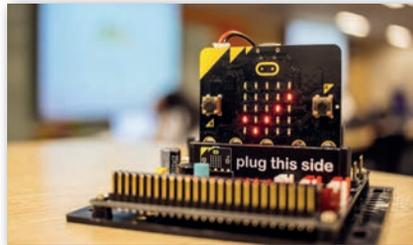
Programozzuk micro:biteket!

A következőkben egy olyan eszközt fogunk megismerni, amely valódi szenzorokkal van felszerelve, és blokkprogramozási környezetben is programozhatjuk. Ez nem más, mint a micro:bit.

A micro:bit egy oktatási célra kifejlesztett, egyetlen lapkán megvalósított miniszámítógép. Található rajta egy 5 × 5-ös LED-kijelző, amelyen számokat, szövegeket és különböző ikonokat, animációkat jeleníthetünk meg.

Sokféle érzékelővel el van látva. Rendelkezik gyorsulásérezkelővel, hőmérséklet-érezkelővel, fényérezkelővel, irányérezkelővel.

Van két gombja (A és B jelű), amelyek megnyomására reagálni tud. A be- és kimeneti csatlakozói lehetővé teszik, hogy más eszközökkel is össze lehessen kötni. A Bluetooth-kapcsolatnak köszönhetően pedig a micro:bitek egymással is képesek kommunikálni.

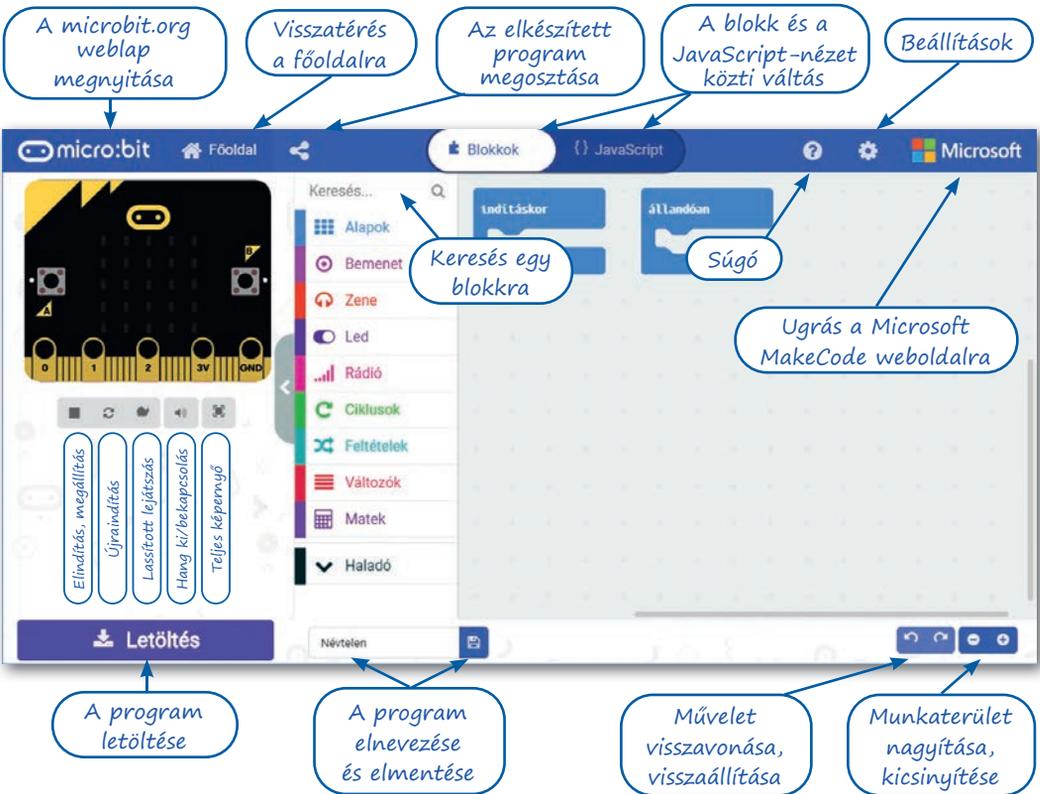


Az eszközt sokféle programozási nyelven lehet programozni, köztük több blokkprogramozási nyelven is. De nagy különbség van abban, hogy az egyes környezetekben milyen funkciókat érhetünk el. A legtöbb funkciót a MakeCode környezet biztosítja. Ennek van online elérhető (<https://makecode.microbit.org/>) és letölthető változata is.

Ismerkedés a felülettel

A MakeCode alkalmazás felülete nagyon hasonlít a korábban látott blokkprogramozási környezetekhez. A fő különbség, hogy most nem egy szereplőt irányítunk a képernyőn, hanem egy fizikailag is létező eszközre készítünk programokat.

Az elkészült programot egy szimulátor segítségével ki is próbálhatjuk, de ha rendelkezünk micro:bittel, akkor a programot az eszközre letölthetjük, és maga az eszköz fogja végrehajtani azt.



► A MakeCode programozási felülete

A beépített szenzorok miatt a micro:bit képes reagálni a környezetből érkező hatásokra. Ilyen lehet, amikor valamelyik irányba megdöntjük az eszközt, megnyomjuk valamelyik vagy mindkét gombját, ha sötét helyre visszük, ha észak felé fordítjuk, ha meg-rázzuk, és így tovább.

- Néhány esemény, amelyekre az eszköz reagálni képes



Mire ügyeljünk?

Amikor a micro:bitekkel dolgozunk, ügyeljünk a következőkre!

- Mindig tiszta asztalon dolgozzunk! Különösen veszélyesek az eszköz számára a rövidzárlatot okozó tárgyak, például a gemkapocs, tűzőgépkapocs, körző, egy ceruza kitört grafithegye, kifolyt üdítőital stb.
- Az elektrosztatikus kisülés akár tönkre is teheti az eszközt, ezért fontos, hogy mielőtt az eszközhöz hozzáérnénk, érintsük meg a számítógép házát, vagy a tanteremben lévő radiátor vagy fűtési csőhálózat felületét!
- Az eszköz úgy legyen elhelyezve az asztalon, hogy ne eshessen le, illetve véletlenül se lehessen lesodorni azt!
- Az egyes gesztusok kipróbálása során (pl. rázás) figyeljünk arra, hogy biztosan tartsuk az eszközt a kezünkben, ne ejtsük azt le!
- A tevékenységek során se az eszközben, se a társainkban ne tegyünk kárt!
- Elemről csak addig működtessük az eszközt, ameddig feltétlenül szükséges, egyébként használjuk az USB-kábelt!
- Ne kössünk az eszköz kivezetéseihez olyan külső eszközöket, amelyek például nagyobb tápfeszültséget igényelnek, mint amit az eszköz biztosítani tud! Ez akár tönkretelheti az eszközt, és az esetleges túlmelegedés miatt égési sérüléseket is okozhat!
- Az eszköz érzékelőit megzavarhatják külső tényezők, például erős mágneses tér, fémtárgyak stb. Ha a szenzorok nem megfelelő értékeket mérnek, vizsgáljuk meg, hogy nincs-e a környezetben valamilyen zavaró tényező!



Készítsünk animációt!

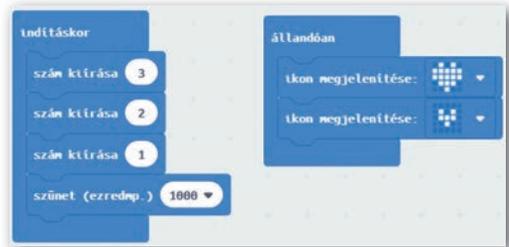
Kezdjük az eszközzel való ismerkedést azzal, hogy a LED-kijelzőn jelenítsünk meg különböző ábrákat és animációkat!

A munkaterületen alapesetben két blokkot is találunk. Az *indításkor* blokk tartalma egyszer, a program elindulásakor hajtódik végre. Az *állandóan* blokk tartalma állandóan ismétlődik, csakúgy, mint a korábban látott végtelen ciklus esetén.

Mindkét blokk kék színű, ez egyben arra is utal, hogy melyik kategóriába tartoznak. Ez most az *Alapok* kategória.

Feladatok

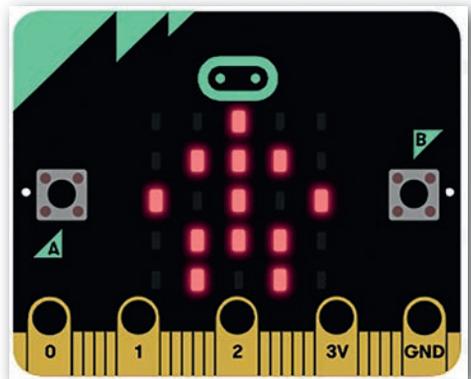
1. Próbáljuk ki, hogy az itt látható blokkok hatására mi fog történni a szimulátorban/eszközön! Mit tapasztalunk?
2. Bővítsük úgy a kódot, hogy indításkor az eszköz kiírja a keresztnévünket! Az ehhez szükséges blokk szintén az *Alapok* kategóriában található. Próbáljuk ki a programot!
3. Láthattuk, hogy a programban nemcsak szívecskét, hanem sok más ikont is meg lehet jeleníteni. Cseréljük le a szív ikonokat több, általunk választott ikonra. Bővítsük az *állandóan* blokk tartalmát úgy, hogy az animáció legalább öt ikonból (fázisból) álljon!
4. Nincs olyan ikon, amire szükségünk van? Semmi gond, rajzoljunk egyet kedvünk szerint! Készítsünk olyan animációt, amelynek minden fázisát mi rajzoltuk meg. Az ehhez szükséges blokkot szintén az *Alapok* kategóriában kell keresni.
5. A micro:bit logója pont a LED-kijelző tetejénél van elhelyezve, és úgy néz ki, mintha egy robot feje lenne, két szemmel. Használjuk ki ezt, készítsünk egy robotot, amely tud integetni!



► Mit csinálnak a blokkok?

A *Bemenet* kategóriában találunk olyan vezérlőblokkokat, amelyekkel megoldható, hogy a blokkok akkor hajtódjanak végre, amikor megnyomtuk az *A* vagy a *B* gombot, vagy mindkettőt egyszerre.

Készítsünk olyan animációt, amelyben az *A* gombot lenyomva a robot az egyik karjával integet, a *B* gomb hatására pedig a másikkal! Ha mindkét gombot egyszerre megnyomjuk, akkor pedig csináljon valami egyedi mozgást, amit mi találunk ki. Az integetés legalább kétszer ismétlődjön. Keressük meg a megfelelő blokkot ehhez a *Ciklusok* kategóriában. Az integetés után a robot vegye fel a kiindulási helyzetét! Próbáljuk ki a projektet! Mentsük el a programot, hogy következő alkalommal folytathassuk!



Használjuk az érzékelőket!

Korábban megismerkedtünk azzal, hogy a LED-kijelzőn hogyan jeleníthetünk meg ikonokat, animációkat, és már az *A* és *B* gombokat is használtuk. Most továbblépünk úgy, hogy a micro:bit érzékelőiben rejlő lehetőségeket is kihasználjuk.

Feladat

Az *A* és *B* gombokkal működtetett animációs projektet fejleszünk tovább úgy, hogy ne csak a gombokra reagáljon a micro:bit. Az eszköz balra döntésekor integessen a robot az egyik kezével, jobbra döntéskor pedig a másikkal! Amikor megrázzuk az eszközt, akkor pedig mindkét karját emelje fel, majd engedje le!



▶ A *Bemenet* kategória egyik blokkja

Tipp: Ezeket a gesztusokat akár a szimulátorban is kipróbálhatjuk a virtuális micro:bit mozgatásával, illetve a Rázás (shake) gomb megnyomásával.

Készítsünk számlálót, ami egyben dobókocka is!

A micro:bitet akár elemről is lehet működtetni, a rátöltött programot pedig egész addig képes végrehajtani, míg egy másik programot rá nem töltünk. Ez azt jelenti, hogy a számítógéptől függetlenül, akár a szabadban is használni tudnánk.



A következőkben egy olyan alkalmazást készítünk, amellyel egyszerűen megszámlálhatunk különböző dolgokat, sőt akár dobókockaként is működhet. Ezt az alkalmazást sokféle célra felhasználhatjuk:

- Megmérhetjük, hogy hány autó haladt el az iskola melletti útszakaszon adott idő alatt. Ha látunk egy autót, növeljük meg a számlálót!
- Egy meccsen megmérhetjük, hogy hány labdaérintés kellett ahhoz, hogy gól szülessen. A gól után lenullázhatjuk a számlálót, és folytathatjuk a mérést.
- Bármilyen társasjátéknál nyilvántarthatjuk, hogy hányszor nyertünk. Sőt, igazi dobókocka helyett használhatjuk a micro:bitet is.
- A tanórai feladatoknál véletlenszerű csoportbeosztást tudunk csinálni (nem csak digitáliskultúra-órán!). Akik azonos számot dobznak a virtuális dobókockával, azok egy csoportba kerülhetnek egy feladat megoldásakor.

Feladat

Gondoljuk át, hogy milyen más célra tudnánk használni egy ilyen eszközt, a saját érdeklődési körünkhöz, hobbinkhoz kapcsolódóan! Beszéljük meg, hogy kinek milyen ötlet jutott eszébe!

Hogyan tároljuk a számláló értékét?

A terv az, hogy egy olyan alkalmazást készítünk, amelyben ha megnyomjuk a *B* gombot, akkor mindig növekedjen eggyel a számláló a micro:bit kijelzőjén. Az *A + B* gomb hatására pedig nullázdójon le a számláló. Az *A* gomb megnyomásakor véletlenszerűen jelenjen meg egy szám 1 és 6 között, mintha egy dobókockával dobtunk volna.

Azt már megismertük, hogy az *A* és *B* gombok lenyomását figyelő vezérlőblokkot hogyan kell használni. De hogyan tudnánk eltárolni, sőt növelni egy számot, valamint azt megjeleníteni a kijelzőn?

Változók használata

Ahhoz, hogy számokat, szövegeket el tudjunk tárolni egy program során, meg kell ismerkednünk a változókkal.

Az adatokat a program végrehajtásakor úgynevezett **változóban** tárolhatjuk. Az adat lehet szám, szöveg, sőt más, összetettebb adat is. A változó onnan kapta a nevét, hogy a program végrehajtása során a tartalma **változhat, módosulhat**.

A változóknak egyedi nevet kell adnunk. A változó értékére pedig a változó neve alapján hivatkozhatunk.

Nézzük meg a számláló alkalmazás kapcsán, hogy a gyakorlatban hogyan használhatnánk a változókat.

A programunkat azért készítjük el, hogy megszámloljunk különböző dolgokat. Ezt az adatot egy változóban fogjuk eltárolni.

Amikor **elnevezzük a változót**, akkor érdemes olyan nevet adni neki, amely alapján később is jól tudjuk majd azonosítani. Most *számláló* néven hozunk létre egy változót!

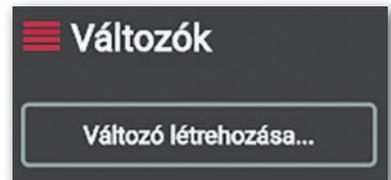
Miután ezt megtettük, a *Változók* között megjelenik egy ovális alakzat, benne a megadott névvel. Ezt a blokkot tudjuk használni akkor, ha le akarjuk kérdezni, hogy éppen milyen értéket tartalmaz a változó.

De nemcsak ez a blokk jött létre, hanem a változó értékének beállítására szolgáló blokk is. Az érték szintén egy ovális alakzatban van elhelyezve. Ide számot is írhatunk majd, de akár szöveg is lehet benne, sőt különböző matematikai műveletek eredménye is.

Szintén létrejön egy blokk, amely a számláló értékét eggyel megnöveli. (Ha negatív előjelet használunk, akkor akár csökkenthetjük is az értéket.)

Feladatok

1. Készítsük el az itt látható programot, és próbáljuk ki a működését!
2. Módosítsuk úgy a programot, hogy az *A + B* gomb hatására is nullázdjon le a változó értéke!
3. Fejlesszük tovább az alkalmazást úgy, hogy az *A* gomb hatására egy véletlenszerűen meghatározott szám kerüljön bele a *dobókocka* nevű változóba, majd jelenítsük meg a változó értékét. Nézzünk körül a *Matek* kategóriában, hogy melyik blokkal lehetne véletlen számot létrehozni!



- ▶ Változó létrehozása a MakeCode-feülületen (Változók kategória)



- ▶ A létrehozott változóhoz tartozó blokkok



- ▶ A számlálóprogram egy részlete

Készítsünk egy játékot!

Van egy eszközünk, amelyen különböző ikonokat jeleníthetünk meg, illetve reagál arra, ha megrázzuk, megdöntjük valamelyik irányba. Ezen egyszerű elemek felhasználásával már akár játékokat is készíthetünk.

Tegyük fel, hogy igazságosan akarjuk eldönteni, hogy mi vagy a testvérünk ehesse meg az utolsó szelet csokit. Vagy a véletlenre szeretnénk bízni, hogy kinek kell kivinnie a szemetet a kukába. Ilyenkor akár kő, papír, olló játékot is játszhatunk. De erre a játékra akár a micro:bitet is megtaníthatjuk.

Ennek kapcsán pedig olyan új dolgokkal is megismerkedünk, amelyeket más játékok készítésénél is fel tudunk használni.



► A kő, papír, olló játék kézjelei

Gondoljuk át, hogy mi lehet egy kő, papír, olló játék algoritmus! Valójában annyi történik, hogy véletlenszerűen kiválasztjuk, hogy három lehetőségből (kő, papír, olló) melyiket fogjuk a kezünkkel mutatni. Ez nagyon hasonló ahhoz, mintha egy 1 és 3 közötti számra gondolnánk, és a szám értékétől függően más-más kézjeleket mutatnánk.

Azt, hogy egy változó értékétől függően más-más dolog történjen, azt elágazással tudjuk leírni. De mi az az elágazás?

Elágazások a hétköznapi életben

Elágazásokkal a hétköznapi életben is gyakran találkozunk. Az alábbi mondatok mind ezt szemléltetik:

Ha hazaérsz 4 óráig, **(akkor)** vidd el a kutyát sétálni!

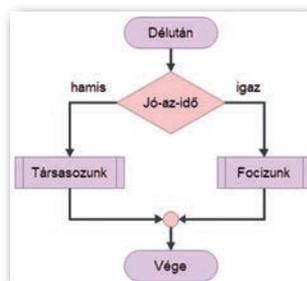
Ha tüzet észlelsz, **(akkor)** nyomd meg a tűzjelzőt!

Ha látsz egy hullócsillagot, **(akkor)** kívánj valamit!

Ezek a mondatok egyszerű elágazásra mutatnak példát. Csak arra az esetre adtunk meg utasításokat, ha a feltétel igaz. Ha hozzátesszük azt is, hogy különben mi történjen, akkor már kétirányú elágazásról beszélhetünk:

Ha délután jó az idő, **(akkor)** focizunk, **különben** társasozunk.

Ha jobban érzed magad, **(akkor)** menj át a nagyihoz, **különben** maradj itthon pihenni.



► Kétirányú elágazás folyamataiban szemléltetve

Az **elágazásban** egy utasítás vagy egy utasításcsoport végrehajtását feltételhez tudjuk kötni. **Egyszerű elágazás** esetén az utasítások akkor hajtódnak végre, ha a megadott feltétel igaz.

Kétirányú elágazásban már azt is megadjuk, hogy milyen utasítások hajtódnak végre akkor, ha a feltétel nem igaz, vagyis hamis.

Feladat

Fogalmazzunk meg mi is olyan mondatokat, amelyek egyirányú, illetve kétirányú elágazásnak felelnek meg!

Elágazások és változók az algoritmusokban

Amikor algoritmizálunk, érdemes a változókkal és elágazásokkal kapcsolatban az itt látható megfogalmazást használni.

A *kő, papír, olló* játékunk algoritmusá így alakul:

Láthatjuk, hogy az 1 és 3 között véletlenszerűen meghatározott számot egy olyan **változóban**

tároljuk el, amelynek neve: *gondoltszám*. Amikor a változónak értéket adunk, akkor nem csak egyenlőségjelet használunk, hanem egy kettőspontot is teszünk az egyenlőségjel elé.

Ha a program algoritmusának első sorát fel kellene olvasnunk, akkor így tehetnénk meg: „*A gondolt szám változó értéke legyen egyenlő egy 1 és 3 közötti véletlen számmal.*”

Az elágazásokat a következő mondatszerű leírásokkal adhatjuk meg:

```
Program
  gondoltszám:=véletlenszám(1 és 3 között)
  Ha gondoltszám=1 akkor kő kirajzolása
    különben
      Ha gondoltszám=2 akkor papír
        kirajzolása
          különben olló kirajzolása
        Elágazás vége
      Elágazás vége
    Elágazás vége
  Program vége
```

Egyszerű elágazás esetén:

```
Ha feltétel akkor
  utasítás(ok)
elágazás vége
```

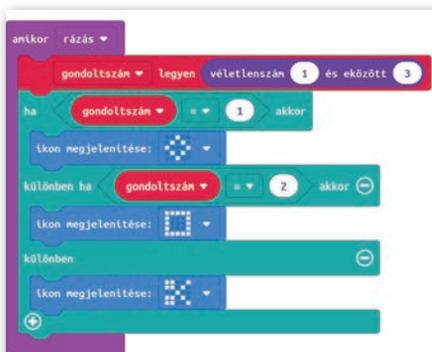
Kétirányú elágazás esetén:

```
Ha feltétel akkor
  utasítás(ok)
különben
  utasítás(ok)
elágazás vége
```

Háromirányú elágazás esetén:

```
Ha feltétel akkor utasítás(ok)
különben ha feltétel2 akkor utasítás(ok)
  különben utasítás(ok)
  elágazás vége
elágazás vége
```

Játékra fel!



► A programkód a MakeCode környezetben

Feladatok

1. Próbáljuk ki az itt látható kódot! Játsszunk le pár játszmat párokban!
2. Fejlesszük tovább a játékot úgy, hogy nyilván lehessen tartani, hogy hány játszmat nyertünk! A B gombbal lehessen növelni a számlálót, az A gombbal lehessen nullázni!
3. Játsszunk tojás, fióka, sas, fönix játékot! Kezdetben mindenki tojás állapotból induljon. Aki megnyer egy játszmat, fióka lesz. Ők egymás ellen játszanak párokban, majd aki nyer, sassá változik. A sasok egymás ellen fognak játszani. Aki nyer a sasok közül, az fönixmadárrá változik. Az nyer az osztályban, aki először eléri a fönixmadár szintet!
4. Hogyan lehetne továbbfejleszteni a játékot? Ötleteljünk párokban, és készítsük el a továbbfejlesztett alkalmazást!

Gyakorlás, saját ötletek megvalósítása

Most már egy új, kézzelfogható eszközt is megismertünk, amelynek kijelzőjén számokat, szöveget, ikonokat és animációkat is meg tudunk jeleníteni. Az eszköz szenzorai segítségével érzékelni tudja a környezetét, és akár gesztusokkal (pl. rázás) is irányítani lehet.

Ezek alapján újabb izgalmas programokat lehet megvalósítani.



Alkossunk együtt!



Alkossunk három-négy fős csoportokat! Gondoljuk át, hogyan lehetne továbbfejleszteni a korábban megvalósított számláló- és dobókocka-alkalmazást úgy, hogy minél több társasjátéknál lehessen használni segédeszközként! Gyűjtsük össze, hogy az általunk játszott és kedvelt társasjátékoknál milyen segédeszközöket kell használni, és azokat hogyan lehetne kiváltani micro:bittel!

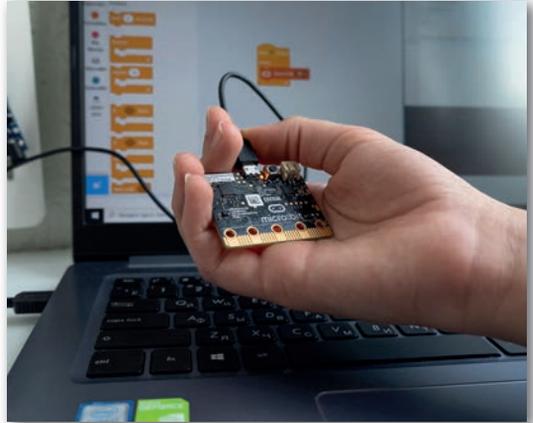
Tervezzük meg az alkalmazást, majd valósítsuk is meg! Törekedjünk arra, hogy mindenki kapjon olyan részfeladatot, amiért ő a felelős!

Ha elkészültünk a munkánkkal, mutassuk be egymásnak a fejlesztéseket!

Otthon pedig, amikor összeül a család egy jó kis társasjátékra, lepjük meg őket azzal, hogy a micro:bitek is szerepet kapnak a játékban!

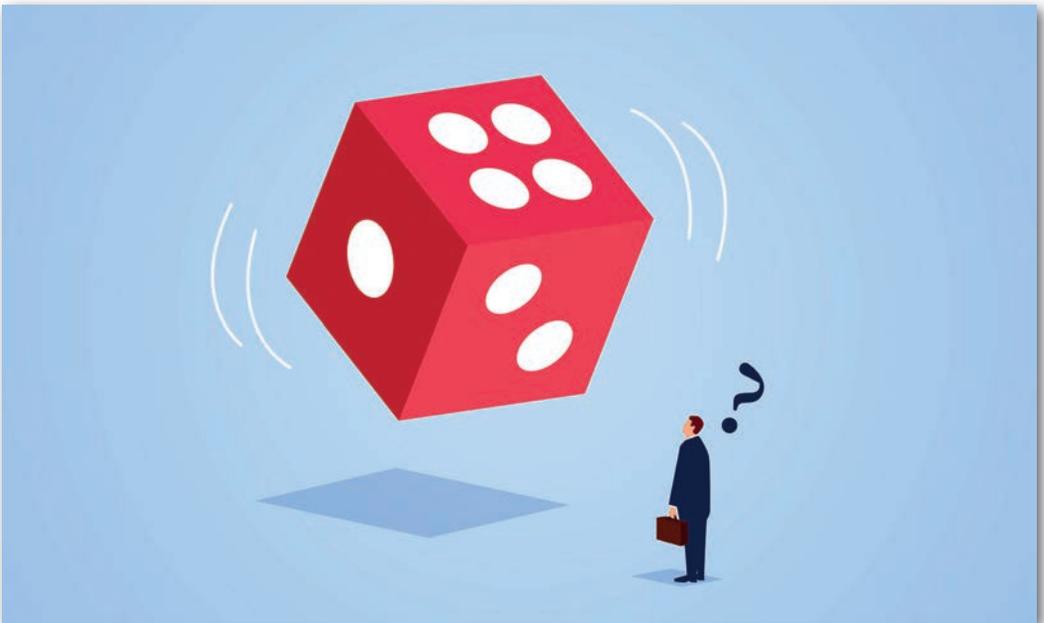
Egyéni feladatok

1. Készítsünk olyan animációt, melynek szereplője egy általunk kiválasztott állat. Az *A* és *B* gombok megnyomásakor, valamint a különböző események hatására (pl. rázás, döntés) más-más animáció játszódjon le (pl. menjen balra, jobbra, ugorjon egyet stb.).
2. Készítsünk egy programajánló alkalmazást, amely különböző tevékenységeket ajánl! Ha unatkozunk, csak megrázzuk a micro:bitet, és a megjelenő ikon alapján már akár cselekedhetünk is (pl. zenehallgatás, sport, kirándulás stb.)!
3. Készítsünk egy időjárás-előrejelző alkalmazást. Az *A* gomb hatására véletlenszerűen napsugár, esőfelhő vagy hópihe alak jelenjen meg. A *B* gomb hatására pedig jelenjen meg számként, hogy a micro:bit hőmérséklet-érzékelője hány fokot érzékel.



Páros feladat

Készítsünk egy számkitalálós játékot! A játék során azt kell megtippelnünk, hogy a micro:bit milyen számra fog gondolni. Aki eltalálja, az kap egy pontot. A játékot felváltva lehet játszani. A program úgy működjön, hogy a micro:biten a *B* gomb megnyomásával lehessen növelni a kijelzött megjelenő szám értékét, az *A* gomb hatására pedig egy véletlen szám jelenjen meg 1 és 9 között.



Első bemutatónk

Az emberek imádnak történeteket hallgatni és történeteket mesélni. Ha módjukban állt, a történeteket már a régi időkben is képekkel kísérték. A fényképezés megjelenésével lehetővé vált a diaképek vetítése, a számítógépek pedig elhozták a bemutatókészítő alkalmazásokat.

A legtöbb bemutatókészítő alkalmazás mind a mai napig egymás után vetített diákból álló bemutatót készít. Mi is ilyeneket készítünk ebben az évben.

A legismertebb bemutatókészítő alkalmazás a Microsoft PowerPoint. Olyannyira összeforrt a neve a bemutatókészítéssel, hogy manapság sokan mondják, hogy „levetített egy PowerPointot”, vagy „a „pendrájvon van egy pépété”. (A „ppt” a PowerPoint-fájlok egyik lehetséges kiterjesztése.)

Ma már egyre többen ismerik és használják a mindenki számára ingyenesen elérhető és letölthető LibreOffice Impress vagy a számos ingyenesen használható internetes bemutatókészítő valamelyikét.

A „bemutató” idegen szóval „prezentáció”. Ez az elnevezés is elterjedt a mindennapokban.

Mire használnak bemutatókat?

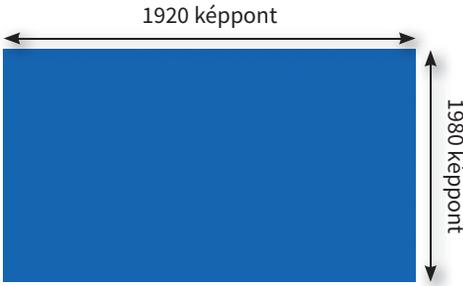
Biztosan volt már olyan tanítási órád, amelyiken tanárod, tanárnőd bemutatót vetített nektek. Mi volt a bemutatón? Miért néztétek meg a bemutatót?

- Valamilyen esemény (kirándulás, farsang) képeit néztétek meg?
- A tananyaghoz kapcsolódó fényképeket néztetek meg?
- Valamilyen magyarázó ábra volt rajta?
- Volt-e szöveg a bemutatón? Ha igen, mennyi?

Képek az interneten és a telefonokon

Bár a bemutatók nem csak képek vetítésre alkalmasak, mi elsőként képeket tanulunk meg elhelyezni egy dián. A könyv példáiban használt képek az Előszóban közölt internetes oldalról letölthetők, de a gyakorlatban többnyire magunk keresünk képet az interneten, vagy saját képeinket használjuk.

A számítógép monitora vagy a telefon kijelzője képpontokból áll. Egy képpont nagyon kicsi, szabad szemmel nem látható. Minden képpontnak külön-külön állítható a színe. A monitor, illetve a kijelző **felbontását** úgy szokás megadni, hogy megmondjuk az egy sorban, illetve egy oszlopban lévő **képpontok** számát. A képpont neve idegen szóval **pixel**.



► Egy képernyő felbontása lehet például 1920-szor 1080 képpont



► A túlnagyított kép „pixeles” lesz

Nagy képeket!

Egy kép beszurása nem olyan nehéz. Az igazi trükk abban rejlik, hogy milyen képet szúrás be, és hogy hova helyezed a képet a dián. Szeretjük a nagy képeket – különösen, ha a vetítéskor a hátsó sorból is láthatók.

- Mekkora annak a monitornak a felbontása, aminél te ülsz? Hány képpont van rajta összesen?
- Mit jelent a HD felbontás? Mit jelent a 4k és mit a 8k felbontás?
- Mekkora a felbontása az iskolában használt projektoroknak?

Egy számítógépes kép is képpontokból áll. A telefonokba épített kameránál is meg szokták adni, hogy a felbontása hány megapixeles, azaz hány millió képpontból áll a vele készített kép.

- Hány megapixeles a monitorod?

Amikor az internetről töltünk le képet a bemutatónkba, figyelniük kell arra, hogy a kép elég sok képpontból álljon. Ha túl kevés képpontból áll, fel tudjuk ugyan nagyítani, de csúnya, „pixeles” lesz az eredmény.

- Hogyan lehet az interneten képeket keresni téma szerint?
- Hogyan lehet az internetes keresőben megtalált képet nagy felbontásban letölteni?
- Hogyan állítható be a keresőben, hogy csak a nagy felbontású képeket mutassa meg?

Tanulmányozzuk az alábbi képelhelyezéseket! Melyik tetszik? Miért?



- ▶ A kép vízszintesen középen van, sok a fel nem használt terület.



- ▶ Függőlegesen is középen a kép, de miért nem nagyítjuk fel?



- ▶ A kép a lehető legnagyobb, de vetítéskor fehér csíkok lesznek az oldalán.



- ▶ A fekete háttér nem zavaró sem a képernyőn, sem vetítéskor – igazság szerint nem is látszik.



- ▶ A kép széléhez hasonló háttérszín is szebbé teheti az eredményt.



- ▶ Ha nem hiányzik a kilógó rész, nyugodtan vegyünk nagyobbra képet a diánál! A kilógó rész vetítéskor nem fog látszódni. A piros keret csak itt, a könyvben segít észrevenni a lényegét.

Feladat

Készítsünk néhány diából álló bemutatót városunkról, kedvenc állatunkról vagy filmhősünkről! Töltsünk le nagy képeket az internetről, és szúrjuk be őket úgy a bemutatóba, hogy ott is minél nagyobbak legyenek! Amikor a képek nagyságát állítjuk, figyeljünk rá, hogy el ne torzuljanak!

A képeket kísérő szövegek

Hogyan kell unalmas bemutatót készíteni?

A recept nagyon egyszerű: használjunk kevés és kicsi képet, és tegyünk a diákra sok-sok szöveget, valamilyen rosszul olvasható betűtípussal. Legyen a diákon jó sok felsorolás, és a bemutatónkat néző közönség már alszik is!

Természetesen szeretnénk, ha a mi bemutatónkat mindenki érdekesnek találná. Az előző leckében megtanultuk ügyesen elhelyezni a minél nagyobb és érdekesebb képeket, ma megtanulunk szép feliratokat készíteni hozzájuk.

Kell egyáltalán felirat? Ha igen, mennyi?

Nem biztos, hogy kell felirat. Ha egy képről csak egy mondatot beszélünk a bemutatónk vetítése közben, akkor valószínűleg nem kell, mert úgyis mindenki emlékszik még, hogy miről van szó.

Semmiképp ne írjunk ki a bemutatóra mindent, amit úgyis elmondunk! A legtöbb esetben pár szónál többre nincs szükség.

Könnyen olvasható feliratok

Készítsünk bemutatót két fecskefajunkról! A képek a könyv weboldaláról letölthetők. Az első képpel könnyű a dolgunk, mert a háttere fehér. A másodikonál sem rossz a helyzet, mert a háttér majdnem egyszínű, és be tudunk állítani egy hasonló színt. A tönk, amin a molnárfecske ül, kerüljön a kép aljára!

Ha bármi, aminek szeme van (ember, állat, rajzfilmfigura, de akár egy autó lámpái is) a képre kerül, és nem felénk néz, akkor szinte biztos, hogy irányítani fogja az néző tekintetét. Úgy érdemes elhelyeznünk, hogy a szöveg felé nézzen.

A bemutatókészítőkből a szöveg szövegdozba kerül. Válasszunk jól olvasható és érdekes, de nem túldíszített betűket! Akkora betűkkel kell írunk, hogy távolabbról is olvasható legyen a felirat – ha megoldható, lépünk egy-két lépést hátra a monitorunktól, és ha ilyenkor is olvasható a szöveg, akkor a kivetítőn sem lesz gond.

A számítógép a kijelző színeit három színből, a vörösből, a zöldből és a kékiből keveri ki. Minden képpont színét úgy tárolja, hogy elteszi a három összetevő mértékét, ez a szín **RGB-kódja**.

A rajzolóprogramokban (például az ingyenesen használható és letölthető GIMP-ben is) általában van „pipetta” eszköz, amivel az összetevők számértéke kiolvasható. Ezeket a számokat megadva tudunk beállítani ugyanilyen színű hátteret a bemutatókészítőben.



Az azonos szerepű feliratok (a példában a fecskék magyar, illetve tudományos neve) készüljenek egyforma betűtípussal és -mérettel! A betű színét állítsuk a háttértől elütőre, és válasszunk olyan színt, ami jól illeszkedik a kép színeihez!

A szövegdobozban a sorokat (bekezdéseket) rendezhetjük balra, jobbra vagy középre. Figyeljük meg, hogy itt szép lett úgy a szöveg, hogy a fecske irányába rendeztük – érdemes mindig kipróbálnunk többféleképp.

A bemutatókészítőknél tudjuk tükrözni a képeket. Fordítsuk meg a fecskéket, hogy a másik irányba nézzenek, és fogalmazzuk meg, hogy milyen érzésünk támad a diát így megnézve!

Nézzük meg a többiek választott betűtípusait, és beszéljük meg, hogy melyiknek mi az előnye, és mi a hátránya!

Fontos, hogy a feliratok ne kerüljenek túl közel sem a fecskéhez, sem a dia széléhez. Próbáljuk a feliratokat ide-oda helyezve megállapítani az ideális távolságot! Meg tudunk állapítani szabályt arra nézve, hogy mekkora távolság kell?

Feliratok sokszínű képen

Bonyolultabb a dolgunk, ha sokszínű kép elé szeretnénk olvasható feliratot tenni. Sokszor nem találunk olyan színt, amivel jól olvasható marad a felirat. Márpedig a felirat olvashatóságáról nem szabad lemondanunk.

Ilyen esetekben megoldás lehet, ha megadjuk a szövegdoboz kitöltésének színét. Ha úgy látjuk, hogy az egyszínű háttér túl sokat takar ki a képből, beállíthatunk átlátszó hátteret is. Az átlátszóság mértékének ügyes megválasztásával a felirat is olvasható marad, és a kép is látszani fog.



Feladatok

Magyar kutyafajták

Készítsünk két diából álló bemutatót három jellegzetes magyar kutyafajtáról! A bemutatóhoz a következő képek állnak rendelkezésre: puli.jpg, vizsla.jpg. (A tankönyv weboldaláról tudod őket letölteni.)

1. Az első diára helyezzük el a puli képét úgy, hogy kitöltse a diát! A kép nagyságának beállításakor soha ne torzuljon a kép! Ha a puli lábaiból csak egy kicsi lóg le, akkor az az érzése van a nézőnek, hogy a készítő figyelmetlen volt. Állítsuk be a puli képét úgy, hogy a lábak ne is látsszanak!
2. A második diára helyezzük el a vizsla képét úgy, hogy kitöltse a diát! A vizslából ne lógjon le semmi, és legyen kényelmes távolság a vizsla és a dia széle között!
3. Készítsük el a Puli feliratot!
 - a. Szúrjunk be szövegdobozt!
 - b. Írjuk bele, hogy „Puli”!
 - c. Állítsunk be jól látható méretű betűket (ha van rá mód, lépünk hátra, és nézzük meg, hogy pár lépésről kényelmesen olvasható-e a felirat)!
 - d. Válasszunk jól olvasható betűtípust!
 - e. A szövegdoboznak állítsunk fekete hátteret és átlátszóságot!
 - f. Állítsunk be fehér betűszínt!
4. A szövegdobozt másoljuk át a második diára! (Így egyforma lesz a betűtípus, -méret és -szín.) Írjuk át benne a feliratot „Vizslá”-ra!
5. Töltsünk le az internetről nagy, komondort ábrázoló képet!
 - a. Szúrjuk be a képet az utolsó diára!
 - b. Jelenítsük meg a kutyafajta nevét az előző diához hasonlóan!

A fájlok kiterjesztése arról tájékoztat, hogy milyen adat (kép, zene, videó, program, szöveg) van a fájlban. A **kiterjesztés** általában 3-4 betű hosszú. A számítógépeden láthatók a kiterjesztések?

Manapság a képet tartalmazó fájlok legtöbbször a .jpg vagy a .png kiterjesztést kapják. A .gif kiterjesztésű képek tartalmazhatnak rövid mozgóképet is, a .tif kiterjesztésű képeket nyomtatott kiadványokhoz szokták használni.



► 1. dia



► 2. dia

A szél

Készítsünk négy diából álló bemutatót a szél hatásairól, felhasználásáról! A bemutatóhoz a tankönyv weboldaláról letöltött képek közül a következők állnak rendelkezésünkre: hogleballon.jpg, megdolt_fa.jpg, villamas.tif, szelturbina.jpg.

1. Az első három dián helyezzük el úgy a képeket, hogy kitöltsék a teljes diát! A képek nagyságának beállításakor a képek nem torzulhatnak.
 - a. Az első dián figyeljünk, hogy a villanypózna ne lógjon ki a képből!
 - b. A második dián a hőlégballonok felül legyenek picit távolabb a dia szélétől!
 - c. A harmadik dián a fa teljes egészében legyen látható!
2. A negyedik dián szeretnénk, hogy a közel lévő szélturbina teljes egészében látható legyen. Ha a dia hátere látható marad, állítsuk az éghez hasonló színűre!
3. Készítsük el az első dia feliratát!
 - a. Az első dia szövegdobozát helyezzük a diára!
 - b. Írjuk be a feliratot!
 - c. Keressünk szép, jól olvasható betűtípust!
 - d. A betűszínt állítsuk fehérre!
 - e. Állítsunk be minél nagyobb betűméretet, de a villanyvezetékekhez ne érjen hozzá a felirat!
4. A szövegdobozt másoljuk át a második diára! (Így egyforma lesz a betűtípus, -méret és -szín.)
 - a. A feliratot írjuk át a mintának megfelelően!
 - b. A szövegdoboznak állítsunk be fekete hátteret és átlátszóságot!
5. A második dia szövegdobozát másoljuk a harmadik diára!
 - a. Írjuk át benne a feliratot!
 - b. A szövegdoboz hátterének színét állítsuk olyan kékre, mint amilyen az ég!
6. Másoljuk át a második diáról az utolsó diára a szövegdobozt!
 - a. Írjuk át benne a feliratot!
 - b. Helyezzük el úgy a szövegdobozt, hogy a közeli turbina lapátjaihoz ne érjen hozzá, de legyen minél magasabban!



▶ 1. dia



▶ 2. dia



▶ 3. dia



▶ 4. dia

Rajzok a bemutatóban

A bemutatónkban eddig csak fényképek és szövegek szerepeltek. A bemutatókészítő alkalmazásokban meg lehet rajzolni egyszerűbb alakzatokat: háromszögeket, négyszögeket, köröket és ellipsziseket (más szóval oválisokat), különféle vonalakat, görbéket, nyilakat.

Szabályos alakzatok

Ha beszúrunk egy téglalapot a bemutatónkba, észrevevesszük, hogy az alakját egerünkkel szabadon változtathatjuk. Ha rajzolás közben lenyomjuk a Shift billentyűt, a téglalapból négyzet lesz. Hasonló a helyzet az ellipszissel is: ha rajzolás közben lenyomjuk a Shift billentyűt, kört kapunk. Ugyanígy az összes többi beszúrható alakzat is szabályos lesz a Shift billentyű lenyomásával.

Vonal és kitöltés

Az alakzataink lényegében két részből állnak:

1. **Vonal** részből, ami a sokszögek, ellipszisek esetében az alakzat körvonala, a vonalak, görbék esetében maga az alakzat.
2. **Kitöltés** részből, ami a sokszögek, ellipszisek esetében az alakzat belseje. Vonalaknak, görbéknek nincs ilyen része.

A két rész külön formázható, színezhető. A kitöltés nem csak egyszínű lehet, használhatunk színátmenetet, képet is. A vonal színe és vastagsága állítható, lehet szaggatott, a végére kerülhet nyíl. A kitöltésből és körvonalból álló alakzatoknál akár a kitöltés, akár a körvonal elhagyható. Ha a kitöltést elhagyjuk, láthatóvá válik, ami az alakzat mögött van.



Szöveg és sorrend

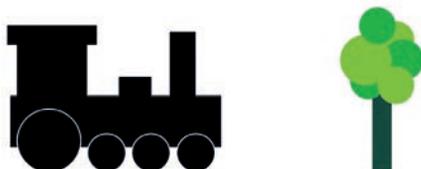
Az alakzatba belekattintva tudunk bele szöveget írni. Az alakzat ilyenkor kicsit úgy működik, mint egy szövegdoboz.

Ha a bemutatókészítővel való rajzolást úgy tekintjük, mint ha színes papírokból vágánk ki ezt-azt, és a kivágott papírokból készítenénk ábrát, jó nyomon járunk. És ahogy a papírokat tudjuk egymás elé és mögé helyezni, az alakzatokkal is megtehetjük ezt.



Rajzoljunk egyszerű alakzatokból!

Mi mindent lehet még egyszerű alakzatokból megrajzolni?



Rasztergrafika és vektorgrafika

Mostanra bizonyára észrevettük, hogy míg a fényképek esetében nagyon figyeltünk, hogy a képeket ne nagyítsuk túl, ezeket a rajzokat akkorára nagyítjuk, amekkorára csak akarjuk, nem lesznek csúnyábbak. A különbséget a képek tárolási módja okozza.

A fényképeket a számítógép sorokra és oszlopokra, azaz képpontokra bontja. Ezután az egyes képpontok színének RGB-kódját tárolja. Ezt raszteres, más szóval bittérképes tárolásnak hívjuk.

A rajzoknál a gépünk inkább olyasmiket jegyez meg, hogy mekkora az alakzat szélessége és magassága, milyen irányban kanyarodik a körvonal, melyik alak van elöl, és melyik hátul. Ez a módszer a vektoros, vektorgrafikus tárolás.

Pontosság

Készítsük el az olimpiai öt karika rajzát!

- Hogyan tudunk pontos kört rajzolni?
- Hogy kell beállítani, hogy a körnek ne legyen ki-töltése?
- Hogy állítható a vonal színe? Miként tudunk *pontosan* olyan színekett állítani, mint az eredetiek?
- Hogyan oldható meg, hogy a többi kör pont akkora legyen, mint az első?
- Hogyan tudjuk az első körrel egy magasságban kezdeni az első sor másik két körét?

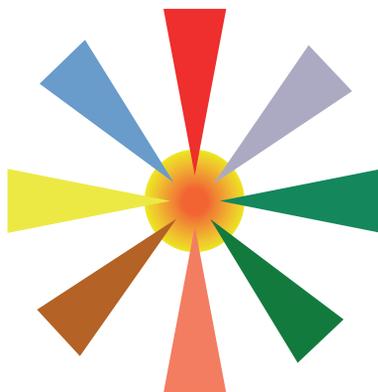


Az öt elkészült karikát érdemes csoportba foglalnunk, mert a csoportot már könnyen tudjuk a dia közepére rendezni.

Tükrözés és forgatás

Rajzoljuk ehhez hasonló virágot!

- Először rajzoljunk egy álló háromszöget, majd készítsünk róla másolatot!
 - Hogyan lehet tükrözni a másolatot?
 - Hogyan tudjuk a tükörképet pontosan az álló fölé tenni?
- Foglaljuk csoportba a felső és az alsó szirmot! Készítsünk a két szirmóról másolatot!
 - Hogyan tudjuk a másolatot úgy forgatni, hogy el tudjuk készíteni a vízszintesen és az átlóban lévő szirmokat?
 - Használunk-e körvonalakat?
 - Milyen színátmenet van a virág közepén?



Vasarely

Victor Vasarely, azaz Vásárhelyi Győző magyar származású művész volt. A képek, amiket alkotott, nagyon modernnek, újszerűnek számítottak akkoriban. Keressük meg interneten a „Planetary folklore participations No 1” vagy az „Orion-Noir” című képét, és próbáljuk meg egy-egy részletét a bemutatókészítőnkkel megrajzolni!

Feladatok

Ablak

A fehér háttér előtt feketével rajzolt ablakot hat téglalapról készíthetjük el.

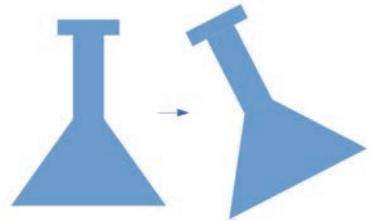
1. Készítsünk egy hosszú, álló téglalapot!
2. A vonalat és a kitöltést állítsuk azonos színűre!
3. Másoljuk le a téglalapot kétszer! Kész vannak a függőleges részek.
4. Készítsük egy újabb másolatot valamelyikről, és forgassuk el vízszintesre!
5. Másoljuk ezt a téglalapot is kétszer! Kész vannak a vízszintes részek.
6. Rakjuk össze az ablakot!



Lombik

A lombik egy háromszögből és két téglalapról készül.

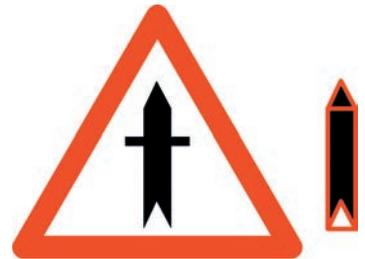
1. Készítsünk egy álló téglalapot! Távolítsuk el a körvonalát!
2. Készítsünk egy háromszöget! Távolítsuk el a körvonalát!
3. Készítsük el a lombik tetejéül szolgáló téglalapot! Ennek is távolítsuk el a körvonalát!
4. Rakjuk össze a lombikot álló helyzetben!
5. Foglaljuk a részeit csoportba!
6. Forgassuk el kissé a csoportot!



KRESZ-tábla: útkereszteződés

A tábla egy egyszerű háromszög, a belseje pedig két háromszög és két téglalap.

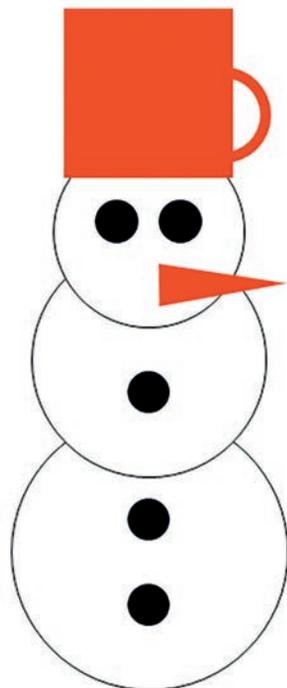
1. Készítsük el a tábla külsejét!
 - a. Rajzoljunk egy háromszöget! Ha a bemutatókészítőben lévő háromszög túl „sovány”, akkor húzzuk szélesebbre!
 - b. Állítsuk be a körvonal vastagságát, és hogy ne legyen kitöltése!
2. Rajzoljuk meg a belsejében lévő alakot!
 - a. Készítsünk egy hosszú álló téglalapot!
 - b. A téglalapról távolítsuk el a körvonalát, a kitöltését pedig állítsuk feketére!
 - c. Készítsünk egy háromszöget, aminek a körvonalát és a kitöltését állítsuk be a téglalaphoz hasonlóan!
 - d. Másoljuk le a háromszöget, és a kitöltését, körvonalát állítsuk fehérre! Takarjuk ki vele a téglalap alját!
 - e. Másoljuk le a téglalapot, és alakítsuk át olyanra, hogy használhassuk az előzőt átszelelő, vízszintes téglalapként!



Hóember

Egy hóember rajzát készítjük el bemutatókészítő alkalmazással.

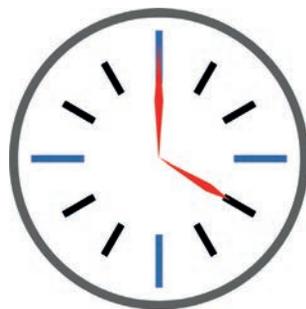
- Rajzoljunk egy kört! Figyeljünk, hogy szabályos kör alakú legyen!
 - A kör kitöltőszíne legyen fehér!
 - A körvonalát állítsuk vékony feketére!
- Készítsünk a körről két másolatot!
 - A két másolatot méretezzük át, figyelve, hogy az alak ne torzuljon!
 - Helyezzük egymásra a három kört! Kész a hóember teste.
- Rajzoljunk egy kicsi kört!
 - A körvonalát kapcsoljuk ki!
 - A kitöltőszín legyen fekete!
- Készítsünk a kis körről négy másolatot! Így összesen öt kis körünk van.
 - Három körből készítsük el a hóember gombjait!
 - Két körből készítsük el a szemeit!
- A téglalaprajzoló eszköz használatával rajzoljunk egy négyzetet!
 - Töltsük ki pirossal!
 - Kapcsoljuk ki a körvonalát!
 - Helyezzük el a hóember fején!
- A hóember kalapjaként szolgáló fazék füle egy kitöltés nélküli kör.
 - Rajzoljuk meg a kört!
 - Állítsunk be számára vastag piros körvonalat!
 - Helyezzük el a négyzet oldalán!
- A hóember orra egy háromszög. Válasszuk ki az alakzatok közül!
 - Ügyeljünk rá, hogy megfelelően vékony és hosszú legyen!
 - A négyzethez hasonlóan állítsuk be a körvonalát és a kitöltését!
 - Forgassuk el, és tegyük a helyére!



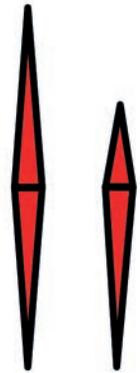
Óra

Az előző oldalon lévő virághoz nagyon hasonlóan tudunk számlapos-mutatós órát is készíteni. A kihívást a számlap és a mutatók jelentik, a kört pedig akár a munka elején, akár a végén is elkészíthetjük.

- Először a 3, 6, 9, 12-es számot jelző téglalapokat készítjük el.
 - Készítsük el a 12-est jelző kék téglalapot!
 - Készítsünk róla másolatot, és helyezzük el a 6-os helyére!
 - Foglaljuk a két téglalapot csoportba!
 - Készítsünk másolatot a csoportról, és a másolat elforgatásával készítsük el a 3-as és a 9-es vonalát!



2. Most a rövidebb, fekete vonalakat készítjük el.
 - a. Bontsuk szét a 12-es és 6-os vonalát adó csoportot!
 - b. A felső vonalról készítsünk másolatot! Színezzük feketére!
 - c. Vegyük rövidebbre a másolatot, de úgy, hogy az alja legyen a hosszú téglalap aljánál!
 - d. Másoljuk le a fekete téglalapot, és húzzuk el a 6-os helyén lévő téglalaphoz! A két téglalap felső oldala legyen egy magasságban!
 - e. Foglaljuk csoportba a két téglalapot, és az előzőekhez hasonlóan alakítsuk ki a többi szám helyét!
3. A mutatók kialakításához háromszögeket használunk. A könyvben lévő ábrán csak a jobb megértés miatt van vastag szegély.
 - a. Készítsünk egy háromszöget! A körvonalát töröljük, a kitöltése legyen piros!
 - b. Másoljuk le, majd tükrözzük a másolatot, és az alapjukat illesszük egymáshoz!
 - c. Foglaljuk őket csoportba! Kész a nagymutató.
 - d. Tegyük a nagymutatót a helyére!
 - e. Másoljuk le a nagymutatót, és bontsuk szét a csoportot!
 - f. A felső háromszöget vegyük alacsonyabbra!
 - g. Foglaljuk őket ismét csoportba! Kész a kismutató.
 - h. Forgassuk el a kismutatót, és tegyük a helyére!



Animációk

Animációk segítségével a dia elemeit – a feliratokat, képeket, rajzokat – tudjuk „életre kelteni”, megmozdítani. Animáció használatával oldható meg, hogy valamelyik elem

- később jelenjen meg, mint ahogy a diát megmutatja a bemutatókészítő program, vagy
- a helyén maradván megmozduljon, ha már látszik a dián, így hívva fel magára a figyelmet, illetve
- eltűnjön a diáról, mikor már nincs rá szükség.

A fenti három animációcsoporton kívül rendelkezésre áll egy negyedik is, amellyel a dia egy-egy eleme az általunk megrajzolt útvonalon tud végighaladni – ez a mozgási útvonal típusú animáció.

Az animációk sorrendje

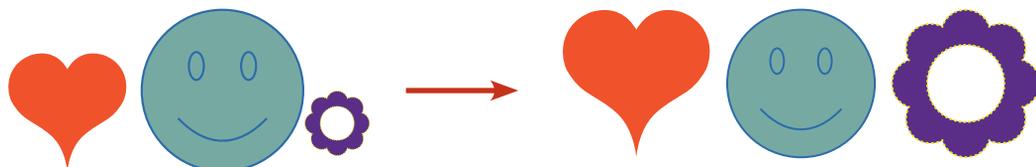
Készítsünk egy diát, olyat, amelyiken három alakzatot helyezünk el! Figyeljünk, hogy az alakzatok ne torzuljanak!

- Melyik billentyűt kell lenyomva tartanunk, ha meg akarjuk akadályozni a torzulást?

Az alakzaton jobb egérgombbal kattintva találunk olyan menüpontot, amelyikkel az alakzat pontos méretét tudjuk megadni. Állítsuk mindhármát egyformára, hogy egymás mellett kényelmesen elférjenek!

Néhány kattintással meg tudjuk oldani, hogy az alakzataink pontosan egy sorban legyenek.

- Az általunk használt bemutatókészítőben az alakzatokon kell kattintani (esetleg jobb egérgombbal), vagy a menüben találunk módot az alakzatok egy sorba rendezésére?



Amikor elkészültünk, foglaljuk csoportba a három alakzatot, és a csoportot igazítsuk a dia közepére! A középre igazítást követően bontsuk szét a csoportot. Mentsük a munkánkat, mert az animációk beállítása során elképzelhető, hogy örülni fogunk, ha van mihez visszatérni.

Állítsunk be animációkat az egyes alakzatokra:

Menjenek a helyükre egymás után az alakzatok! Ehhez valamilyen megjelenési, más szóval bevezető animációra van szükség.

Ha mind a három a helyén van már, mozduljanak meg! Most vesszük hasznát a kiemelési animációknak.

Amikor mindhárman megmozdultak, hagyják el az alakzataink a diát! Itt van szükség a kilépési vagy eltűnési animációkra.

Vegyük észre, hogy az animációk egymás után következése, sorrendje beállítható.

Rendezzük át úgy az animációkat, hogy az első alakzat jöjjön be, mozduljon meg, majd menjen ki, aztán csinálja végig ugyanezt a második és a harmadik alakzat!

Az animációk beállításai

Az animációknál vannak egyedi beállítások, amelyek csak arra az egy animációra érvényesek. Ilyen például, hogy beúszáskor melyik irányból érkezen a dia adott eleme.

Vannak azonban olyan beállítások is, amelyek mindegyik animációnál egyformák. Ilyenek az animációk sebességének vagy késleltetésének beállításai, de ilyenek az animációk indulását meghatározók is. Alaphelyzetben minden animáció egymás után, egérekattintásra (vagy billentyűlenyomásra) történik meg.

- Hányat kellett kattintanunk az előző animáció végigjátszásához?

Ha nem egérekattintásra indul egy animáció, akkor két lehetőséget szoktak kínálni a bemutatókészítők. Mindkettő automatikus indítás, a különbség az, hogy

- az egyik esetben az előző animáció véget értével indul a lejátszás,
- a másikban pedig akkor indul el, amikor az előző is, azaz az előzővel együtt.

Változtassuk meg úgy a diánkon lévő három alakzat animálását, hogy teljesen automatikusan jöjjenek be, mozduljanak meg, és menjenek ki! Ha a legelső alakzat legelső animációját is automatikusan akarjuk elkezdetni a bemutatókészítővel, akkor ott is azt kell beállítani, hogy az előző után kezdődjön. Ilyen esetben az „előző” esemény nem más, mint a dia megjelenése.

Alakítsuk át úgy az animációkat, hogy a három alakzat kattintásra, de egyszerre jöjjön be, egy második kattintásra megint csak egyszerre mozduljon meg, és egy harmadik kattintásra egyszerre menjen ki!

- Hány animációnál kell beállítanunk azt, hogy az előzővel együtt történjen meg?

Az animáció szerepe

Most, hogy áttekintettük az animációk főbb lehetőségeit, megfogalmazunk egy fontos figyelmeztetést is. **Csak akkor használjunk animációt, ha az valóban szükséges a diára, és segít a dia megértésében!** Az önmagáért való animáció csak eltereli a figyelmet a dia igazi mondanivalójáról.

Mikor segít az animáció? Alapvetően akkor, ha

- egy dia részeiről külön-külön beszélünk, vagy
- egy dián látnunk kell valaminek a kialakulását.

Készítsük el az alábbi diát, figyelve arra, hogy a szöveg minden sora külön szövegdobozba kerüljön! A dia elkészítéséhez használjuk a könyv webhelyéről letölthető lakotorony.tif képet!

Amikor elkészültünk, állítsuk be úgy a szövegdobozok animációját, hogy a szövegek alulról kezdve jobbról, egyesével érkezzenek meg!

A lakótorony



várórség lakrésze
kápolna
lakóterem
lovagterem
előcsarnok
csigalépcsők
tömlőc
kút

A lakótorony

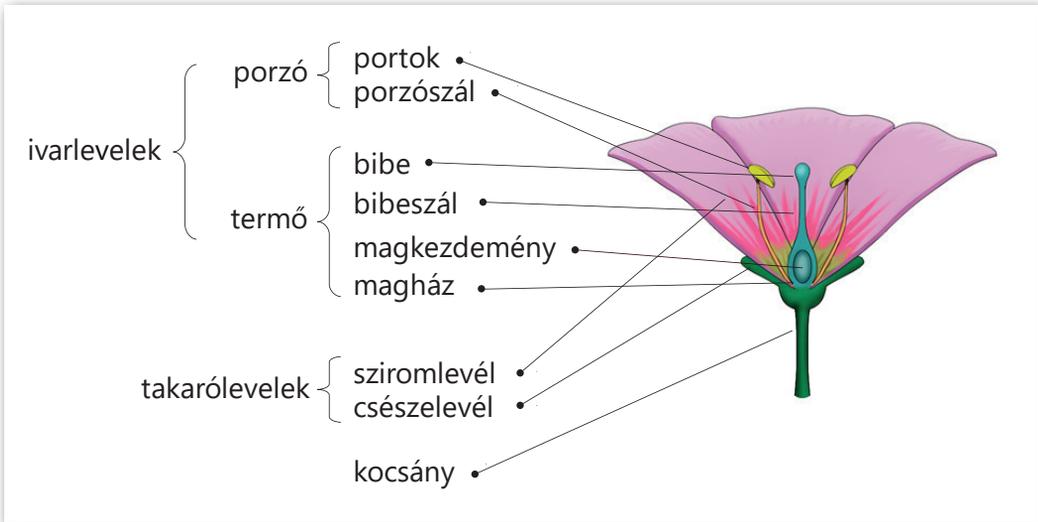


előcsarnok
csigalépcsők
tömlőc
kút

Feladatok

A virág részei

Bemutatót készítünk a kétszikű növények virágának részéről. A munkához rendelkezésre áll a könyv webhelyéről letölthető virag.jpg fájl.

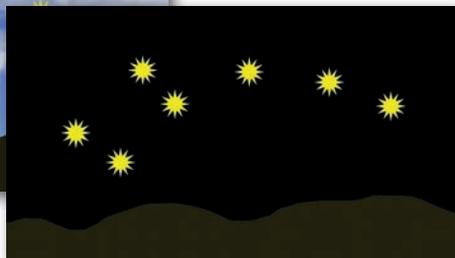
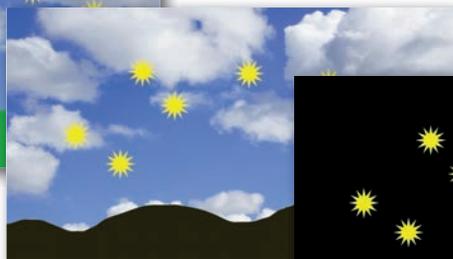
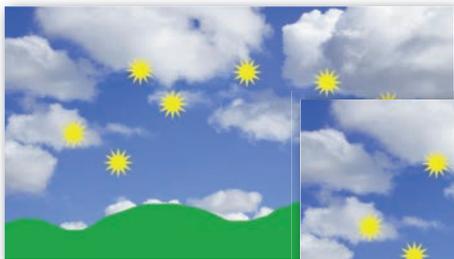


1. Szúrjuk be a képet! Nagyítsuk fel!
2. A kép jobb oldali virágjára nincs szükségünk, lógassuk le a dióról!
3. A szövegek kialakítása és elhelyezése
 - a. Szúrjunk be szövegdobozt! Írjuk bele a „magkezdemény” feliratot (azért ezt, mert ez a leghosszabb)!
 - b. Állítsunk be jól olvasható és elég nagy feliratot, majd másoljuk le nyolcszor, és helyezzük a dobozokat egymás alá!
 - c. Ha jól elfér a kilencsornyi szöveg egymás alatt a minta szerinti csoportosításban, akkor nézzük meg, hogy kényelmesen ki fog-e férni az „ivarlevelek” és a „porzó” felirat is!
 - d. Ha minden jól elfér, töröljük a fölösleges szövegdobozokat, és hozzuk létre a még hiányzókat!
 - e. Helyezzük el őket az ábrán látható módon!
4. A kapcsos zárójeleket a bemutató alakzatai között találjuk. Szúrjuk be mind a négyet, és helyezzük el őket a minta szerint!
5. A nyilak
 - a. Rajzoljuk meg a nyilakat! Mutassanak *pontosan* a virág megfelelő részeire, és induljanak a megfelelő felirattól! Nagy lesz a kavargás, de mindjárt segítünk rajta.
 - b. Állítsunk be a legfelső nyílra egy általunk választott, helyben megjelenést biztosító animációt!
 - c. Tüntessük el a nyilat egy újabb, helyben lejátszódó animációval!
 - d. Állítsunk be hasonló módon különféle, helyben lejátszódó be- és kilépési animációt a többi nyílra is!

Kisgöncöl

Látványos, bár nagyon egyszerű bemutatót készítünk az ismert csillagképről. Valójában bármelyik csillagkép ábrázolható így. Rendelkezésre áll a könyv webhelyéről letölthető felhok.jpg fájl, de letölthetünk magunknak egy nekünk jobban tetszőt is az internetről.

1. Állítsunk be a diának felhős hátteret!
 2. Keressünk képet az interneten egy nekünk kedves csillagképről! Szúrjuk be a diára, és nagyítsuk ki!
 3. Keressünk nekünk tetsző csillag alakzatot a bemutatókészítőben, és helyezzünk el egy-egy csillagot a kép csillagaira!
 4. Töröljük a képet, nem lesz már rá szükségünk.
 5. Rajzoljunk egy, az egész diát eltakaró fekete téglalapot!
 6. Küldjük a téglalapot a csillagok mögé!
 7. Állítsunk be automatikusan lejátszódó, lassú megjelenést biztosító animációt a téglalpra, hogy érzékeltetni tudjuk, ahogy beesteledik.
 8. Kérjünk néhány másodperces késleltetést, hogy a néző egy ideig a felhős hátteret is láthassa.
 9. Állítsunk be lassú, automatikus megjelenést a csillagokra is! Egyszerre vagy egymást követően jelenjenek meg? Melyik az érdekesebb animáció, melyik áll közelebb a valósághoz?
- Tegyük bonyolultabbá az animációt!
10. A felhős háttérre rajzoljunk a bemutató szabad kézi vonal eszközével világoszöld „tájat”! Töltsük ki világoszölddel!
 11. Másoljuk le a tájat, és fedjük el a másolattal az eredetit! A kitöltést állítsuk sötétzöldre!
 12. Állítsunk megjelenési animációt a sötétzöld tájnak úgy, hogy a fekete téglalappal egyszerre jelenjen meg!



Bemutatók fényképrészletekkel

Gyakran előfordul, hogy találunk egy jó képet a bemutatónkhoz, de csak egy részletére van szükségünk. A többi részen olyasmi látható, ami vagy szükségtelen a diánkhoz, vagy egyenesen zavaró hatású.

Képek vágása

Láttuk már, hogy a képet tudjuk úgy állítani, méretezni, hogy a fölös része lelógjon a diáról. Ez azonban nem mindig kielégítő megoldás. Vannak olyan bemutatókészítők, amelyek lehetővé teszik az egyszerűbb képvágások elvégzését, de használhatunk erre a célra rajzolóprogramokat is. A kissé bonyolultabb alakúra vágásokhoz szinte biztosan külön rajzolóprogramot fogunk igénybe venni.

Bemutató készül három híres személyről. Mi a címdiát készítjük el, ahol mindhárman szerepelnek. A bemutatóhoz felhasználható a könyv webhelyéről letölthető három kép: kleopatra.tif, nagy_sandor.tif, periklesz.tif.

A képek mérete jelentősen eltér egymástól. Van, amelyiken szinte csak a fej látszik, van, amelyiken sok egyéb is.



- A képek vágása előtt vagy azt követően érdemes egyforma nagyságúra alakítani a megmutatott képrészleteket?

Motívumok

Az emberek nagyon hamar felismerik a hasonló alakokat, és általában kedvelik is a hasonló alakokból álló mintákat, motívumokat. Hasonló, ismétlődő motívumok képezik sok ruha-, épület- vagy edénydíszítés alapját. Három jellegzetes csík, négy karika vagy két megtört vonal, vagy két egymás melletti ív már lehet egy világmárka logójának alapja. A találmány nem új, hiszen az ókoriak és középkoriak néhány épületét azért találjuk mind a mai napig gyönyörűnek, mert valamilyen ismétlődő mintát fedez fel benne a szemünk.

A fenti megfigyelést kihasználhatjuk a bemutatókészítés során is. Ennek egyik legegyszerűbb módja, hogy az azonos szerepű képeket azonos, a téglalaptól eltérő alakúra vágjuk.

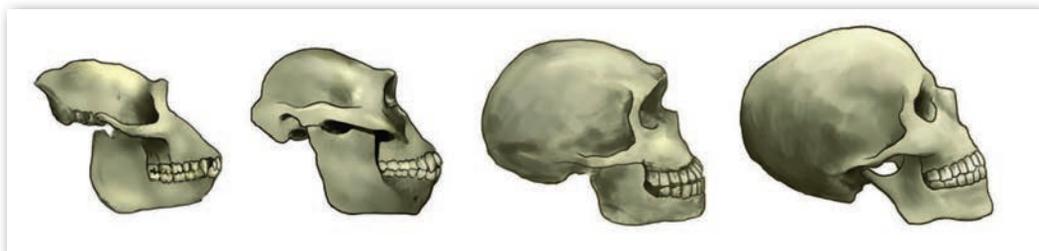


Készítsük el az előző dia új változatát! Ha rajzolóprogramot használunk, érdemes lehet tudni, hogy a másolás/beillesztés a megszokott módon működik akkor is, amikor az egyik alkalmazásban a vágólapra másolt tartalmat egy másik alkalmazásban illesztjük be a helyére.

Képek szétvágása

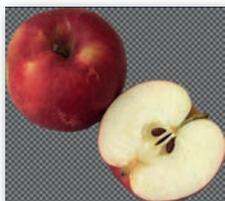
Van, hogy a képnek minden részére szükségünk van, de nem egyben. Ilyekor szinte bizonyosan külön rajzolóprogramra van szükségünk. A képek elérő részeit külön-külön másoljuk be a bemutatókészítőnkbe.

Készítsünk a könyv webhelyéről letölthető `koponyak.jpg` képből négyoldalas bemutatót, melyen egyesével vetítjük le a gorilla, az Australopitecus, a Neander-völgyi ember és a mai ember koponyájának képét!



Átlátszóság a képekben

Az internetes keresőkben képeket keresgélve találunk olyanokat, amely háttérét egy pepita mintázat képezi. Ez a pepita minta valójában nincs a képen, épp azt mutatja, hogy ott nincs semmi, a képnek az a része átlátszó.



Az átlátszó képek különösen jól használhatók sok bemutatóban:

- olyankor, amikor valamilyen háttér elé kell elhelyezni nem onnan származó képet, és
- amikor az ilyen fotó mögé bújtatunk be valamilyen egyéb elemet a dián.

Készítsük el az alábbi diát a könyv webhelyéről letölthető `alma.tif` fájl felhasználásával! Figyeljünk arra, hogy a szövegek jobb oldala (a pontok jobb szélé) legyen egymás alá igazítva!

Beszéltünk már arról, hogy a számítógép a fényképeket alkotó képpontok színét az RGB-kóddal tárolja.

Sok képen azt is tárolni akarjuk, hogy egy kép, képpont mennyire engedi látszani a mögötte lévő színét, azaz mennyire átlátszó. Ilyenkor a fenti három mellett egy negyedik számot is használunk, az úgynevezett **alfa-csatorna**-értéket. Az alfa-csatorna értékének változtatásával tudunk egy képrészletet átlátszóvá alakítani.



Képek vágása színegyezőség alapján

A komolyabb rajzolóprogramokban rendszerint helyet kap a varázspálca eszköz, amelyik a képeken a többé-kevésbé egyforma színű területeket tudja kijelölni. Ilyen terület szokott lenni, ha egy kép egyszínű háttér előtt készül, de ilyen a felhőtlen ég is. Beállítható, hogy mennyire kell egyformának lenniük a színeknek ahhoz, hogy a varázspálca egybefüggőnek tekintse őket.

Az így kijelölt területek eltávolításával magunk is tudunk átlátszóságot varázsolni a képekre. Ha a kijelölt terület eltávolításakor nem pepitát (átlátszóságot), hanem fehér színt kapunk, akkor a képnek nincs alfa-csatornája. Persze ilyenkor sem ijedünk meg, az alfa-csatorna egy-két kattintással a képhez adható.

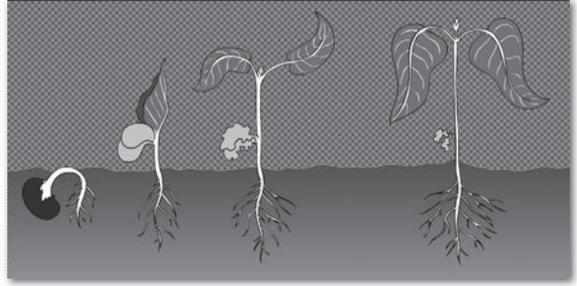
- Az általunk használt rajzolóprogramban miként adunk alfa-csatornát a képhez?
- Készítsünk a könyv webhelyéről letölthető ceger.tif és sotetfelhok.jpg fájl felhasználásával diát!
- Hol van a varázspálca eszköz az általunk használt rajzolóprogramban?
- Melyik billentyűvel törölhető a kijelölt rész?



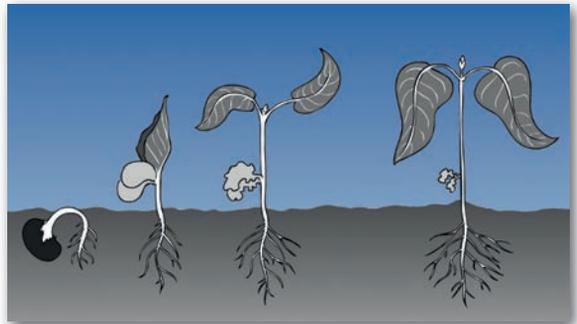
Feladatok

A csírázás folyamata

Animációval szemléltetjük a csírázás folyamatát. A munkához rendelkezésre áll a könyv webhelyéről letölthető csirazas.jpg kép.



1. Az első feladatunk, hogy a kép vas-tag fehér szegélyét eltávolítsuk.
 - a. Nyissuk meg a képet a képszerkesztő alkalmazásunkban!
 - b. Téglalap alakú kijelöléssel jelöljük ki a megmaradó részt!
 - c. Vágjuk a képet úgy, hogy csak a kijelölt rész maradjon meg!
2. Eltávolítjuk a fehér felső részt, hogy látszódhasson a bemutató diájának háttere.
 - a. Adjunk alfa-csatornát a képhez!
 - b. Varázspálcával jelöljük ki az eltávolítandó részt, és távolítsuk is el!
3. A csírázás lépéseit egyesével jelöljük ki téglalap alakú kijelöléssel, és másoljuk át őket a bemutatókészítőbe!
 - a. A kijelölések alul a kép szélénél kezdődjenek!
 - b. Igyekezzünk az újabb kijelöléseket az előző jobb szélénél kezdeni, hogy a „földfelszín” egybefüggő lehessen a dián!
 - c. A bemutatókészítő nem mindig egyforma méretben illeszti be az egyes darabokat. Ha a „föld” alja egymás mellett van, és a teteje nem, akkor méretezzük át a beillesztett részletet! Ügyeljünk arra, hogy ne torzuljanak a képek az átméretezéssel! Egyelőre nem baj, ha a képek kilógnak a diából.
4. Beállítjuk a képek méretét.
 - a. Foglalkozunk csoportba a négy képrészlettel!
 - b. A csoportot méretezzük úgy, hogy minél jobban kitöltse a diát, de felül és oldalt ne lógjon ki! Ha alul nem sikerült egyformán kijelölni a képeket, és néhol látható a dia háttere, mozgassuk kicsit lentebb a csoportot, hogy az egyenetlenség kilógjon a diából! A gyökérnek látszódnia kell.
 - c. Ha a csoport a végső helyére került, bontsuk szét!
5. Állítsunk be animációt az egyes lépések megjelenésére!
 - a. Az animáció egérekattintásra kezdődjön!
 - b. A többi lépés automatikusan játszódjon le, az egyes lépések előtt egy-egy másodperc késleltetéssel!
6. Állítsunk be színátmenetes hátteret!
 - a. Felül a háttér legyen sötétebb (de még világos)kék!
 - b. Alul legyen a háttér egészen világos kék!



A burgonyabogár fejlődése

Animációval szemléltetjük a burgonya kártevőjének fejlődését. A munkához rendelkezésre áll a könyv webhelyéről letölthető *pete.jpg*, *larva.jpg* és *kifejlett.jpg*, illetve a *burgonya.tif* kép.

1. Az első dia

- Állítsunk be világoszöld hátteret a diának!
- Szúrjuk be a *burgonya.tif* képet, és helyezzük el a minta szerint, jobboldalt és alul kicsit lelógatva a diáról!
- A szövegdoboz háttere legyen az a háttérhez illeszkedő, de attól eltérő szín, és állítsunk be rajta átlátszóságot!
- Jól olvasható, nagy fehér betűkkel írjuk be a feliratot!



2. A második dia

- Kérjünk új diát a bemutatókészítőben!
- Állítsuk be az első dián használt világoszöld háttérszínt itt is!
- A dia úgy lesz szép, ha a három szövegdoboz színe, átlátszósága és mérete az előzőhöz hasonló. Azt szeretnénk, ha a betűk típusa és mérete is megegyezne az előző dián lévőkkel. Másoljuk át a szövegdobozt az előző diáról! Ha így járunk el, nem kell újra beállítanunk mindent, amit ott beállítottunk.
- Írjuk át a szövegdobozban lévő feliratot a „pete” szóra! A szövegdoboz szélességét állítsuk be a szövegnek megfelelően!
- Másoljuk le kétszer a szövegdobozt, és írjuk be a „lárva” és a „bogár” szót!

3. A második dia képei

- Nyissuk meg a *pete.jpg* képet képszerkesztő alkalmazásban!
- Jelöljük ki kör alakú kijelöléssel a petéket!
- A kijelölt részt másoljuk a második diára!
- Méretezzük a képet a mintának megfelelően! Az átméretezés során a kép nem torzulhat!
- Helyezzük el a képet a diának megfelelően!
- Ismételjük meg mindezt a *larva.jpg* és a *kifejlett.jpg* képpel!



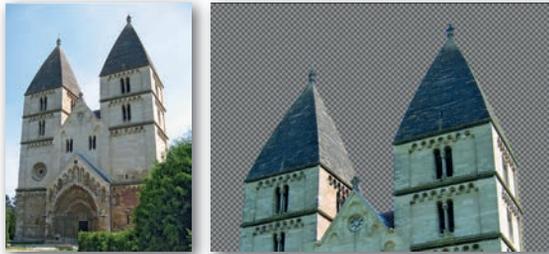
4. A második dia animációja

- Állítsunk be megjelenési animációt az egyes képekre! Úgy lesz szép az animáció, ha egyforma megjelenési hatást használunk a három képen.
- Az első kép kattintásra jelenjen meg!
- A többi lépés automatikusan játszódjon le, az egyes lépések előtt egy-egy másodperc késleltetéssel!

A jáki templom

Érdekes hatást érhetünk el, ha a kép egy részét megkettőzzük, és fedésbe hozzuk az eredeti résszel. A jáki templom egyik tornyával is így teszünk, és így a szöveget be tudjuk bújtatni a két torony közé. A munkához rendelkezésre áll a könyv webhelyéről letölthető `jaki_templom.tif` kép.

1. A kék ég eltüntetése
 - a. Képszerkesztő alkalmazásunkban nyissuk meg a `jaki_templom.tif` képet!
 - b. Téglalap alakú kijelöléssel jelöljük ki a mintán látható képrészletet!
 - c. Vágjuk le a kijelölésen kívüli részeket!
 - d. A varázspálca eszközzel jelöljük ki az ég kékjét! Sokféle kék van a képen, ezért viszonylag nagy küszöbértéket kell beállítanunk az eszköznek. Kísérletezzük ki a helyes értéket!
 - e. Töröljük ki a kék részt!
 - f. A képet másoljuk át a bemutatókészítőbe! A képszerkesztőt még ne zárjuk be!



2. A bemutatóban
 - a. Állítsunk be színátmenetes háttérrel!
 - b. A képet felnagyítva igazítsuk el úgy, hogy az alja a dia aljánál legyen!
 - c. Készítsük el a szöveget! Használjunk nagy és jól olvasható betűket!
 - d. Állítsunk be háttérszínt és átlátszóságot a szövegdoboznak!
 - e. Helyezzük el a szöveget a minta szerint!
 - f. A szövegdoboznak állítsunk be jobbról beúszó animációt!



3. A jobb oldali torony
 - a. A képszerkesztőben téglalap kijelöléssel jelöljük ki a jobb oldal tornyot!
 - b. Másoljuk át a bemutatókészítőbe!
 - c. Hozzuk fedésbe „saját magával”!



Bemutató készítése saját fotókból

Gyakori eset, hogy egy bemutatót a saját magunk készítette fotókból állítunk össze. Ha a bemutató lényegében „csak” diavetítés, például a családi kirándulásról, egyszerű dolgunk van: beszúrjuk a képeket, és úgy rendezzük el őket, hogy minél jobban láthatóak legyenek. Érdekes és nagyon remek bemutató készíthető akkor, ha kifejezetten a bemutató kedvéért készítjük el a fényképeinket, hiszen ilyenkor megtervezhetjük a fotó témáját, a beállításokat.

Emberről csak olyankor készítsünk fényképet, ha azt a képen szereplők, illetve szüleik kifejezetten engedélyezték! Másokat tudtukon kívül vagy engedélyük nélkül fotózni törvénybe ütközik.

Szerencsére tárgyról, tájról, növényekről és állatokról is lehet akár önmagukban is érdekes, akár más kreatív mű kiindulási alpjául szolgáló fotókat készíteni.

A figyelemfelkeltő fénykép elkészítése gyakran csak a szokásostól eltérő beállításon múlik. Ha egy épületet, virágot, utcaképet például hasra fekvé fotózunk, máris érdekesebb a kép. Ha módunkban áll viszonylag közeli képeket készíteni, kis tárgyakat is érdemes a fénykép témájául választani. Ilyenkor is jó ötlet érdekes háttérrel keresni, ami akár a monitoron lévő kép is lehet. Megéri a megvilágítással is kísérleteznünk: az alábbi fotón egy hagyományos asztali lámpa adja a figurán látható sárgás fényt.



Ha a témától elütő egyszínű háttér (akár egy papírlap) előtt fényképezünk, akkor a már ismert varázspálca eszközzel el tudjuk távolítani a háttérrel. Az ilyen fotókat borús napokon, a szabadban vagy nagyobb ablak előtt készítve a tárgyak árnyéka elmosódott lesz, kis szerencsével alig látszik, azaz könnyebben eltávolítható.



Forrásmegjelölés

Ha nem a saját képeinket használjuk, a műveinkben meg kell jelölnünk, hogy honnan származnak, azaz mi a képek forrása. Lehetséges megoldás, hogy az utolsó „rendes” dia után nyitunk egy új diát. Ide kiírhatjuk, hogy ki készítette a művünk alapjául szolgáló képeket. Ha az internetről töltöttünk le valamit, írjuk ki, hogy melyik webcímről és mikor töltöttük le!

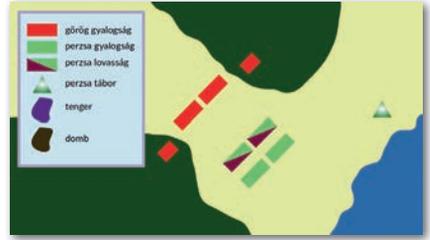
Animációk: mozgásvonalak

Viszonylag ritkán készítünk olyan animációt, amikor a dia egy elemét egyik helyről a másikra **áthelyezzük** a dián. Az áthelyezést a bemutatókészítők a mozgásvonalakkal szokták megvalósítani. A mozgásvonallal lényegében egy útvonalat adnak meg, a dia eleme ezt az útvonalat követi végig az animáció lejátszásakor. Többféle beépített útvonalat találunk, de van olyan eszköz is, mellyel mi magunk rajzolhatunk teljesen egyedi útvonalat.

Helyezzünk el egy alakzatot egy dián, és próbáljuk ki a mozgásvonal működését!

A marathóni csata

Történelemórán ebben az évben ismerkedünk meg a marathóni csatával. A csatában a perzsák és a görögök csapnak össze. A csatát a perzsa lovasok kezdik, megtámadják a görögöket, de kénytelenek visszavonulni. Ekkor roppan össze a két gyalogság. A görögök könnyűfegyverzetű harcosai leereszkednek a csatatér melletti dombokról, és bekerítik a perzsákat. A perzsák elveszítik a csatát, akik túlélnek, elmenekülnek. Ezt a csatát jelenítjük meg bemutatónkban. A csata lefolyását mozgásvonalak használatával animáljuk.



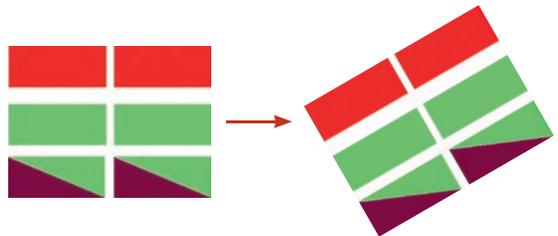
1. Elsőként a lovasságot jelölő ábrát készítjük el.

- A bemutatókészítő alkalmazás alakzatai közül válasszunk egy derékszögű háromszöget! Vegyük le a körvonalát, és állítsuk be a kitöltésének színét!
- Másoljuk le a háromszöget! Tükrözzük a háromszöget vízszintesen és függőlegesen! Állítsuk be az új háromszög színét! A két háromszöget foglaljuk csoportba!



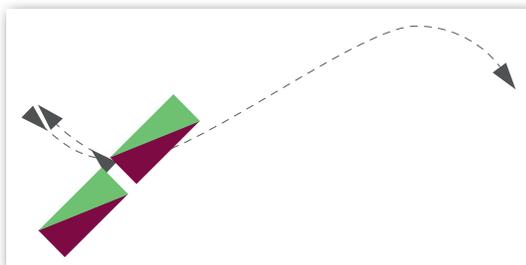
- Másoljuk le a csoportot, és a másolatot helyezzük az előző mellé, hogy két lovassági egység legyen!
2. A gyalogságok elkészítésével folytatjuk.

- Rajzoljunk egy, a lovasság ábrájához hasonló méretű téglalapot! Vegyük le a körvonalát, és töltsük ki az előbb használt színnel!
- A téglalapot másoljuk le még háromszor, és a másolatokat rendezzük el a lenti mintának megfelelően!
- A két felsőt színezzük át a görögök színére!
- Jelöljük ki mind a hat téglalapot, és forgassuk el őket!
- A görögöket helyezzük távolabb, ahogy a kész dia képén látható!
- Készítsük el a dombokon álló két kisebb görög csapat ábráját!

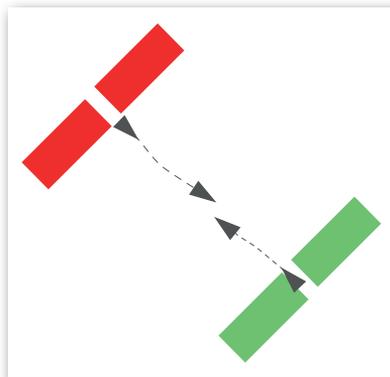


3. A mozgásvonalak kialakítása

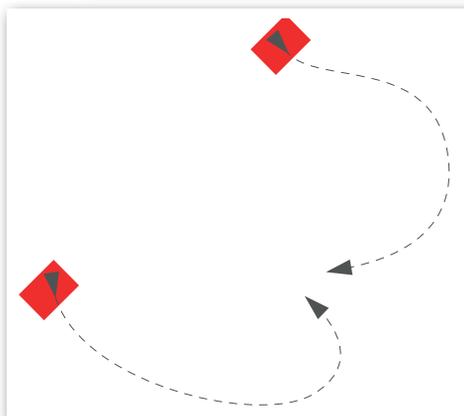
- a. A lovasság csoportja kapjon két mozgásvonalat! Az elsővel közelítsék meg a görög gyalogságot, a másodikkal meneküljenek el! (Figyeljünk arra, hogy a második mozgásvonalat onnan kell indítanunk, ahol a lovasság az első mozgásvonal után *lesz*, és nem onnan, ahol most van!)



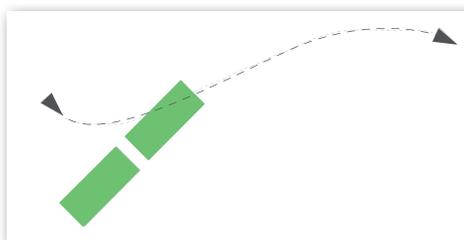
- b. A csata második mozzanata a két küzdőfél gyalogságának támadása. Állítsuk be az animációt úgy, hogy egyszerre játssza le őket a bemutatókészítő alkalmazás!



- c. A harmadik mozzanat a dombokon álló görög gyalogság bekerítő hadművelete. A két csapat külön mozgásvonalat kap. Állítsuk be, hogy egyszerre mozgassa a csapatokat a bemutatókészítő!



- d. Az utolsó esemény a perzsa gyalogság megfutása. Figyeljünk, hogy a mozgásvonal onnan indul, ahová a perzsák a második mozzanatban kerülnek, és nem onnan, ahol a dián vannak!



Ha minden jól ment, elkészültünk a csapatok mozgásával.

4. A dombok, a tenger és a perzsa tábor
 - a. Keressük meg a bemutatókészítő alkalmazásban azt az eszközt, amivel kitöltéssel bíró szabadkézi vonalat rajzolhatunk! Rajzoljuk meg vele a tenger foltját! Nem baj, ha a folt lelóg a diáról.
 - b. A folt kitöltőszíne legyen a tengerre utaló kék szín! A folt körvonalát távolítsuk el, vagy állítsuk a kitöltőszín kékjével azonosra!
 - c. Hasonlóan eljárva készítsük el a két dombot is! Amikor egy-egy dombbal elkészültünk, tegyük a foltot a dombon álló görög gyalogságot jelölő téglalap mögé!
 - d. Rajzoljuk meg a perzsa sereg táborát jelző háromszöget! Töltsük ki olyan színátmenettel, amivel elűt a dombok színétől!
 - e. Helyezzük a háromszöget a perzsa lovasság visszavonulását és a gyalogság megfűtését jelző mozgásvonal végére! Az a cél, hogy a csapatok a csatatérről a táborba érkezzenek meg. Küldjük a háromszöget egészen hátra, hogy a csapatok ne a tábor háromszöge alatt, hanem a háromszögön legyenek, mikor megállnak!
5. A jelmagyarázat



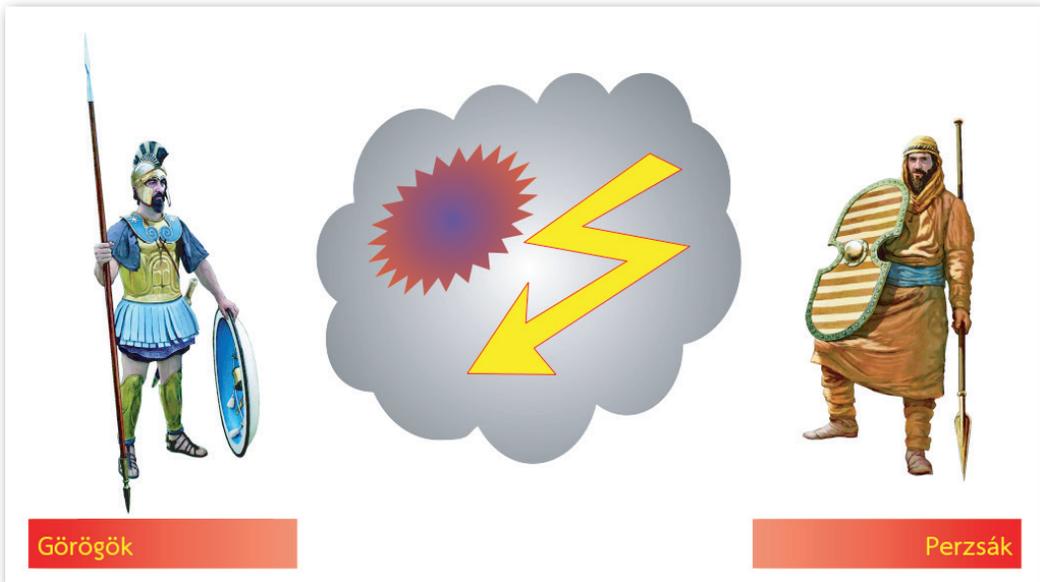
- a. Készítsük el a jelmagyarázat háttéréül szolgáló téglalapot! Válasszunk számára olyan kitöltő- és körvonalszínt, amely a térképtől és a jelmagyarázaton szereplő alakzatoktól egyaránt elűt!
- b. Másoljunk át egy-egy görög, perzsa gyalogos és perzsa lovas csapatot a jelmagyarázatra! Forgassuk el őket vízszintes helyzetbe!
- c. Másoljuk át a perzsa tábor háromszögét!
- d. Rajzoljunk egy „tengert” jelképező foltot, a színét állítsuk a térképen lévő tengerrel azonosra!
- e. Másoljuk le a foltot, és helyezzük el a mintának megfelelően! Állítsuk a színét a „dombokkal” azonosra!
- f. Írjuk meg a jelmagyarázat szövegét!

Elkészültünk ezzel a diával. Remélhetőleg eddig is gyakran megtettük, de most mindenképp mentsük a munkánkat.

A marathóni csata nyitó- és záródiája

A csata elé és a csata után elhelyezünk egy-egy olyan diát, amivel a számítógépes játékok világát idézzük. Rendelkezésünkre áll a könyv webhelyéről letölthető *hoplita.tif* és *perzsa.tif* kép.

1. A nyitódiá



- Kérjünk új diát! Helyezzük át az előzőleg elkészült dia elé!
- Töltsük be és helyezzük el a két katonát az ábrán látható módon!
- Mozgásvonalakkal mozgassuk a két katonát egymás felé, hogy középen találkozzanak!
- Az animációk kezdődjenek egyszerre, automatikusan, kis késéssel a dia megjelenése után!
- Középen helyezzünk el olyan alakzatokat, amik a képregényekhez hasonló módon verekedésre, harcra utalnak! Állítsunk be megjelenő animációt számukra!
- Az animációkat állítsuk be úgy, hogy egymás után automatikusan történjenek meg!
- Készítsük el a görögök erősségét jelző szövegdobozt! Írjuk bele a szöveget, és helyezzük a görög harcos alá!
- A háttérének állítsunk be jobbra világosodó színátmenetet!
- Másoljuk le a szövegdobozt (így biztosan egyforma méretűek), és helyezzük a perzsa harcos alá!
- Cseréljük ki benne a feliratot, és rendezzük a szövegdobozban jobbra!
- Fordítsuk meg a színátmenet irányát!

2. A záródia



- Kettőzzük meg a nyitódiát! A másolatot harmadik diaként helyezzük el!
- Töröljük a fölösleges elemeket!
- Töröljük a két katona animációját!
- Mozgassuk a katonákat oda, ahol az első dián találkoztak!
- A perzsa katonának állítsunk be olyan eltűnési hatást, amivel belehalványul a háttérbe!
- Másoljuk le a perzsa katona rajzát! A másolatot kicsinyítsük le a mintán látható helyre!
- Állítsunk be a kis perzsa katonának olyan megjelenési hatást, amivel előtűnik a háttérből!
- Oldjuk meg, hogy a nagy perzsa harcos eltűnése és a kicsi megjelenése egyszerre történjen, azt érzékeltetve, hogy a nagyból kicsi lett!
- A kis perzsa katonának állítsunk be olyan kilépési animációt, amivel jobbra eltűnik a képről! Az animáció történjen az előzőt követően automatikusan!
- A perzsák erősségét jelző szövegdobozt állítsuk rövidebbre!

Az e–h. pont négy lépését az elterjedtebb bemutatókészítőkben egyetlen animációba sűrítethetjük. Hogyan?

Ebben a fejezetben az internet világgal és az online kommunikációval fogunk megismerkedni. Sok-sok kérdésre keressük együtt a választ. Mi az internet? Hogyan alakult ki? Mit és hogyan tudok keresni az interneten? Mit takar a virtuális személyiség? Hogyan tudom megvédeni az adataimat? Hogyan tudok digitális eszközök segítségével kommunikálni a társaimmal?



Információs társadalom

Az elmúlt évtizedekben a körülöttünk levő világ számos területén lett meghatározó a technológiai fejlődés. A minket körülvevő digitális környezet térnyerése miatt, azt mondjuk, hogy információs társadalomban élünk. Megjelentek például a vezető nélküli metrók, az iskolai értékelés elektronikus naplóban történik, léteznek éttermek, ahol robotok szolgálják ki a vendégeket, és a gyárak robotok segítségével állítják elő termékeiket.

De nemcsak a gazdaságra van hatással a technológiai fejlődés, hanem az egyes emberek, családok életére is. Megjelentek az okostelefonok, okosórák. Sőt, akár az otthonunkat is digitalizálni tudjuk, gondoljunk csak a távolról irányítható fűtésrendszerre, a riasztó- és kamerahálózatra. Léteznek olyan hűtők, amelyek jelzik, ha valamelyik élelmiszernek lejárt a szavatossági ideje. Közlekedésnél segítségünkre van a GPS-rendszer, amely mutatja az útvonalat és az aktuális forgalmat. Elmondhatjuk, hogy a legtöbb háztartásban elérhető széles sávú interneten gyorsan és könnyen jutunk információhoz.

Feladatok

1. Három-négy fős csoportokban gyűjtsétek össze, milyen hétköznapi ügyeket tudunk az interneten elintézni! Miben segíti az életünket az internet? Gyűjtsétek olyan területeket, ahol a technológia segíti az emberek tevékenységét!
2. Gondolkozzatok együtt! Milyenek képzelitek el a jövő technológiáját? A kis csoportok után az egész osztállyal vitassátok meg az ötleteiteket!

Hogyan épül fel az internet?

A világon többféle *hálózat* létezik, például úthálózat, vállalati hálózat, kommunikációs hálózat stb.

Ha több számítógépet összekapcsolunk, akkor szintén *hálózatot* kapunk.

Ezek a számítógép-hálózatok képesek együtt dolgozni, például adatokat tudnak egymással cserélni, megosztani. A számítógép-hálózatok kialakításával energiát és költséget is csökkenthetünk, például a há-



lózathoz csatlakoztatott gépek mind képesek egyazon nyomtatóra nyomtatni. Ha a különböző számítógép-hálózatokat egymással összekapcsoljuk, akkor a világon átívelő pókháló szerű rendszert kapunk. Ez a bonyolult, összefüggő rendszer alkotja az *internetet*. Ha az egyik hálózatból a másikba szeretnénk információt eljuttatni, arra több útvonal is rendelkezésünkre áll. Mivel a hálózatok több szálon is csatlakoznak egymáshoz, az adatok célba érnek akkor is, ha egy-egy elem megsérül vagy kiesik.

Hálózat: Egymással összeköttetésben álló számítógépek.

Internet: Több hálózat összekapcsolásából létrejött hálózati rendszer, amely az egész világra kiterjed.

Böngészés az interneten

Az internet használatához (online szörfözéshez) szükségünk van egy böngészőprogramra, mint amilyen például a *Microsoft Edge*, a *Google Chrome*, a *Mozilla Firefox* stb.



Indítsuk el a kiválasztott böngészőprogramot, és a keresősávba írjuk be a szolgáltató nevét, amely a következőkből épül fel: (pl.: www.okosdoboz.hu)

HTTP: Az angol hypertext transfer protocol (ejtsd: hájper-text transzfer protokoll) kifejezés rövidítése. A szolgáltatók egymáshoz és a felhasználókhoz való kapcsolásához használt rendszer.

Név: A kívánt szerver neve, amellyel kapcsolatba szeretnénk lépni. Ez nagyon sok esetben a *www* szóval kezdődik, ami a *world wide web* (ejtsd: vörl-d vájd veb) rövidítése.

Domén: Ez a cím utolsó szakasza. Ebből akár könnyen kitalálhatjuk, hogy a világ mely részén helyezkedik el a szerver, például a *.hu* – Magyarország; a *.de* – Németország stb. De léteznek: *.com*, *.net*, *.gov* doménnév is.

Feladatok

1. Keressetek rá az iskolátok weboldalára! Milyen információkat tudtatok meg?
2. Nézzetek utána, hogy mi a különbség a *.eu*, *.net*, *.com*, *.gov* és a *.de*, *.it*, *.hu* típusú doménnevek között!

Eszközhasználat

Az otthonainkban növekvő számú digitális eszközkészlet könnyíti meg a hétköznapi életünket. Napjainkban az asztali számítógépek mellett megjelentek a laptopok, tabletek, okostelefonok, okosórák és egyéb, apró számítógépet tartalmazó eszközök.

Azonban több veszély is fenyeget minket: gondoljunk csak a túlzott használatra vagy az adataink védelmére!



Feladat

Vitassátok meg, hogy milyen veszélyei lehetnek az informatikai/digitális eszközök túlzott használatának! Mit javasoltok? Gyűjtsétek össze, beszéljétek meg!

Keresés a világhálón

Az internetre tekinthetünk úgy is, mint egy hatalmas könyvtárra, tudásbázisra. Ha valamilyen információra van szükségünk, különböző *keresőoldalak* segítségével könnyedén megtaláljuk azt. Ezek a szolgáltatók az interneten található több mint kétmilliárd weblap között szisztematikusan folytatnak keresést, a találatokat pedig csoportokba rendezik. A legismertebb és legnépszerűbb keresőoldalak a www.google.hu, a www.yahoo.com és a www.bing.com. Ezek az oldalak keresőmotorok segítségével gyűjtik össze az általunk megadott szempontok alapján az interneten található tartalmakat. A keresőmotorok folyamatosan pásztázzák a weboldalakat, és keresik az új tartalmakat. Ha találnak valamit, azt letöltik, és különböző kategóriákba rendezik. Így tudnak nekünk pillanatok alatt a keresőszavunkhoz kapcsolódó tartalmakat listázni.



Az oldalak a keresési tevékenységünk monitorozásával (követésével, vizsgálatával) hirdetésekkel jelenítik meg.

Kulcsszavas keresés

A keresőoldal megnyitása után a keresősávba egy kulcsszót vagy kifejezést írunk. A találati listában különböző típusú eredményeket találunk.



Megeshet, hogy a keresés eredménye nagyszámú találatot eredményez. Ilyenkor szűkítjük a keresést a következő lehetőségek egyikével!

- és Szerepeljen minden szó a találatban!
- „ ” Ha a keresett szavakat idézőjel közé tesszük, akkor a kulcsszavaknak egymás mellett kell szerepelniük.
- vagy Egyik vagy másik kifejezés szerepeljen a találatban.
- Ez a kifejezés ne szerepeljen a találatok között!

További szűrési lehetőségünk a *Beállítások* / *Speciális keresés* menüpontban is van. Itt például beállíthatjuk, hogy milyen nyelvű találatokat keresünk.

A találati listában felsorakoztatott weboldalak címei alatt rövid összefoglalót olvashatunk arról, hogy miről szól a bejegyzés. Ez segít eldönteni, hogy a megfelelő információt nyújtja-e nekünk az adott weboldal.

Figyelem!

A keresés során kapott találatok egy része nem megbízható oldalakra vezet. Akkor gyanakodhatunk, ha a böngésző figyelmeztető üzenettel hívja fel rá a figyelmünket, vagy nem engedi betölteni az oldalt. Ilyenkor ne kattintsunk rá, mert kárt tehetünk az eszközünkben!

Feladatok

1. Csoportokban keressétek meg a következő kérdésekre a választ!
 - Ki mondta: „A kocka el van vetve.”?
 - Mikor született Kleopátra?
 - Melyik városban található a Festetics-kastély?
2. Keresés közben figyeljétek meg, milyen oldalak kerülnek a találati lista elejére! Miért lehet ez (fizetett oldalak, hirdetések stb.)? Milyen típusú információkat találtatok? Szlektáljatok!

Keresés adatbázisokban

Léteznek olyan weboldalak, amelyek különböző témakörönként gyűjtik össze az interneten található tartalmakat, ilyenek például az online lexikonok, képatadbázisok, fordítóoldalak, menetrendi információs oldalak stb.

Figyelem!

Léteznek olyan adatbázisok, amelyeket a regisztrált felhasználók szerkeszthetnek is, ilyen például a Wikipédia. Ezért az ott talált tartalom nem mindig hiteles. Legyünk kritikusak!

Feladat

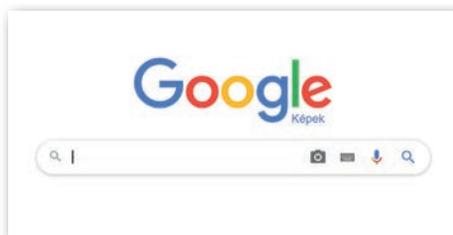
Keressetek rá különböző képatadbázisokra! Keressetek a képek között!

Képkeresés

A munkánk során sokszor képekre van szükségünk, gondoljunk arra, ha egy rövid előadást kell tartanunk, és magyarázatként illusztrálnunk kell valamit. A keresőoldalak segítségével könnyen szűkíthetjük a keresésünket kizárólag a képtalálatokra. Sőt! A szűrési feltételeknél az *Eszközök* menüpontban a kép méretét és típusát is meghatározhatjuk.

A *Beállítások / Speciális keresés* útvonalon további szűrési lehetőségek állnak a rendelkezésünkre.

Arra is figyelniünk kell, hogy az interneten elhelyezett képek egy része magánszemély vagy vállalat tulajdona. Ezért nem használhatunk fel mindent, amit a kulcsszavas keresés során találunk. A keresési feltételeknél beállíthatjuk, hogy a találatok között csak olyan eredmények jelenjenek meg, amelyek probléma nélkül felhasználhatók. Ezt az *Eszközök / Felhasználási jogok* menüpontban tehetjük meg.





Feladat

Keress képet a kedvenc mesehősödről!

- Csak *újrafelhasználható* / *nem kereskedelmi célra újrafelhasználható* képeket keress!
- Szűkítsd a találatokat a nagy felbontású képekre!
- Szűkítsd tovább a találatokat csak .jpg típusú képekre!
- Mit vettél észre? Hogyan alakult a találatok száma?

Útvonaltervezés, keresés online térképen

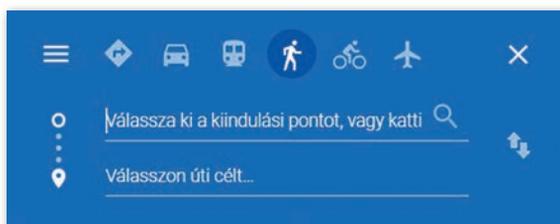
Ha utazást tervezünk, az interneten könnyen tájékozódhatunk arról, hogyan jutunk el a tervezett úti célunkhoz. Ilyen térképes keresőt találunk például a Google és a Bing keresőoldal felületén is. Ha tömegközlekedést használunk, láthatjuk, milyen járatokat kell igénybe vennünk, és tájékozódhatunk a menetrendről. Ha személygépjárművel tervezük az utazást, látjuk a legrövidebb útvonalat.



Keresés során adjuk meg kiindulási pontunkat és az úti célt! Válasszuk ki, milyen eszközt veszünk igénybe!

Feladatok

1. Keress rá, hogy milyen messze vannak autóval a következő városok egymástól! A legrövidebb utat válaszd!
 - Budapest–Pécs
 - Eger–Sopron
 - Debrecen–Hajdúszoboszló
2. Keress rá, hogy a budapesti állatkerttől hogyan jutsz el a budai Várba tömegközlekedési eszközzel! Vizsgáld meg a lehetőségeket!



Ha a talált képeket, weboldalakat, információkat egy-egy munkánk során szeretnénk felhasználni, akkor minden esetben *hivatkozással* kell ellátnunk.

A hivatkozásunk során fel kell tüntetni, hogy MIT találtunk, HOL találtuk és MIKOR találtuk. Például: videó <https://www.youtube.com/xxxxxxx> 2020. 02. 02.

Feladatok

Keress az interneten!

- Készítsd el a kedvenc zenéd videójának hivatkozását!
- Készítsd el a kedvenc ételed receptjének hivatkozását!
- Készítsd el a kedvenc filmed poszterének hivatkozását!

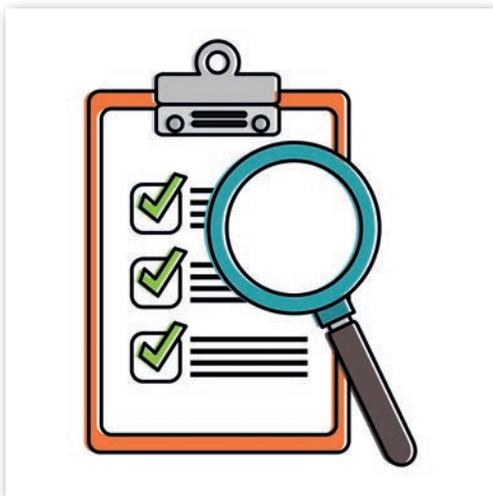


Az információ értékelése

Ahogy arról már korábban volt róla szó, nem minden talált információ származik hiteles forrásból. Mit is jelent ez pontosan? Az internet fejlődésével és bővülésével egyre több olyan szolgáltató jelent meg, amelynek nemcsak fogyasztói, hanem szerkesztői is lehetünk. Ezek az úgynevezett web 2.0-s felületek.

Ha olvasunk egy bejegyzést, hírt, a következők vizsgálata segít minket abban, hogy eldöntsük, hihetünk-e az ott olvasottaknak:

- Hol találtam az információt?
- Ki írta a bejegyzést?
- Milyen forrásra hivatkozik?
- Mikor készült a bejegyzés?



A virtuális személyiség

A webes felületeken való regisztrációhoz különböző adatokat kell megadnunk magunkról, gondoljunk az e-mail-szolgáltatásokra, a közösségi felületekre, online játékokra vagy a csetoldalakra, így tudjuk létrehozni a *profilunkat*. Ne feledjük, hogy némely weboldalak használata törvény által szabályozott életkori határhoz van kötve!

A szülői felügyeleti rendszerek részeként beállítható, hogy a szülők profiljához rendelt 14 év alattiak mire és mikor használhatják a számítógépeket. Ezeknél a kiegészítő funkcióknál a szülők megadhatják, hogy milyen korhatár-besorolás szerinti tartalmat (videó, kép, könyv stb.) jeleníthet meg a gyermekük.



Az adatokat két nagy csoportba soroljuk:

Nyilvános adat: Olyan adatok, melyek más számára is elérhetők, és megadásuk során nem lehet beazonosítani a tulajdonost. Ilyen adat például az órarend, amely a Kréta rendszerben is látható.

Személyes adat: Olyan adatok csoportja, amelyek megadásával be tudjuk azonosítani az adatok tulajdonosát. Ilyen például a név, születési év, lakóhely.

Hogyan védhetjük az adatainkat?

A létrehozott profilunkat védjük *jelszóval*, és időnként változtatunk meg azt!

Az erős jelszó tulajdonságai:

- nem lehet ránk ismerni belőle,
- tartalmaz kis- és nagybetűs karaktereket,
- tartalmaz számjegyeket,
- tartalmaz írásjeleket.



A weboldalakon kialakított profilunkat a jelszón túl *adatvédelmi beállításokkal* tudjuk védeni, illetéktelenek számára láthatatlanná tenni. A megadott személyes adatok soha ne legyenek nyilvánosak mások számára! A kémprogramok, adathalász oldalak visszaélhetnek adatainkkal, és kárt okozhatnak nekünk.



Figyelem!

Amennyiben lehetséges, ne adjuk meg a személyes adatainkat. Ha ez elkerülhetetlen, az adatvédelmi beállításoknál tegyük mások számára láthatatlanná!

Kérjünk segítséget szüleinktől, tanárainktól, hogy biztonságban tudjuk az adatainkat!

Feladat

Vitassátok meg, hogy milyen veszélyek leselkednek ránk, ha nem megfelelő az adatvédelmi beállításunk!

Digitális lábnyom

Biztosan hallottátok már azt, hogy az internet nem felejt. Ez bizony így van, hiszen az online tevékenységünk minden mozzanatát megőrzi. Hosszú évek, évtizedek múlva is visszakéreshetők a feltöltött képeink, videóink, bejegyzéseink. Ezzel úgynevezett *digitális lábnyomot* hagyunk magunk után. A bejegyzéseinken túl olyan tevékenység is megjelenik ebben a lábnyomban, amiről nem is tudunk. Ilyen például a videómegosztó oldalon való médiafogyasztásunk vagy a közösségi oldalakon mások bejegyzéseire tett reakálásunk (pl. lájkolás).



Figyelem!

Azok a bejegyzések, amiket ma megosztunk, évek, évtizedek múlva is megtalálhatók lesznek. Jól gondoljuk át, mit teszünk közre magunkról, barátainkról, családtagjainkról! Ha valaki más készít rólunk fényképet vagy felvételt, és szeretné megosztani az interneten, csak akkor teheti meg, ha engedélyt adtunk rá!

Online kommunikáció

Elektronikus levél

Az internet egyik leggyakrabban használt része az elektronikus levelezés. Az *e-mail* (ejtsd: ímél) a leggyorsabb lehetősége a levélküldésnek. Több szolgáltató is lehetővé teszi az online levelezést, például a Gmail, a Yahoo stb. és léteznek már tabletre, telefonra letölthető alkalmazások is. Azt a szöveget, üzenetet, amit megírunk valamelyik digitális eszközünkön, a címzett pár másodperc múlva már a saját eszközén olvashatja.



A postafiók kezelése

A felhasználó postafiókja több részből tevődik össze:

A *bejövő levelek* között találjuk azokat az e-maileket, amiket nekünk címeztek. Az elolvasásuk után ellenőrizzük, hogy tartalmazzak-e csatolmányt. A levélre a *Válasz/Answer* gombra kattintva azonnal visszajelezhetünk a feladónak. A bejövő leveinket továbbküldhetjük a *Továbbítás/Forward* gombra kattintva.

Az *elküldött levelek* között találjuk a korábban írt és elküldött leveinket.

A *piszkozatok* között vannak azok a levelek, amiket már megírtunk, de még nem küldtünk el a címzettnek.

A *spam* levelek kényszerűen reklámokat tartalmazó e-mailek. Ezeket időnként töröljük ki!

Figyelem!

Egy ismeretlen feladótól származó e-mail melléklete vírust, adathalász vagy kémprogramot is tartalmazhat. Azokat a leveleket, amelyeknek a feladóját nem ismerjük, ne nyissuk meg, mert kárt tehetnek a számítógépünkben, és adatokat szerezhetnek meg rólunk! Csak olyan mellékletet nyissunk meg, amilyenre a levél írója utal a szövegben!

Új levél írása

Ahhoz, hogy e-mailt tudjunk írni, szükség van egy egyedi e-mail-címre, ami a következők szerint épül fel:

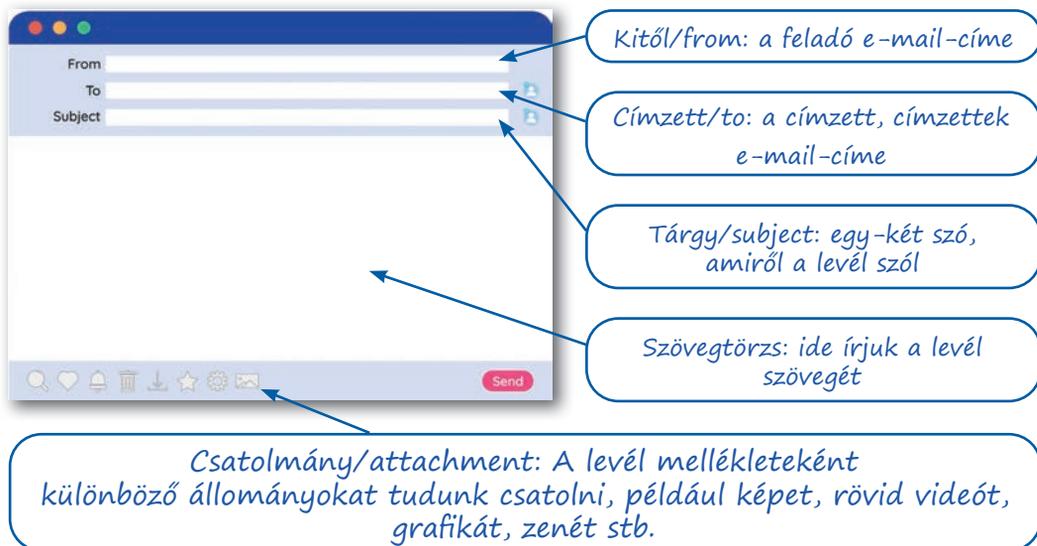
kati1122@probacim.hu

felhasználó egyedi azonosítója	@ (ejtsd: kukac)	szolgáltató neve	domén
--------------------------------	------------------	------------------	-------

Figyelem!

A jelszavunk legyen erős: tartalmazzon kis- és nagybetűs karaktereket, számokat, írásjeleket! Törekedjünk arra, hogy ne lehessen ránk ismerni!

Időnként cseréljük le a jelszavunkat, és soha ne osszuk meg mással!



Új levél írásakor a fejrészben adjuk meg a címzett vagy címzettek *e-mail-címét* és a *levél tárgyát*.

Az e-mail-írás során tartsuk be a már korábban tanult levélírás szabályait!

Kezdjük mindig megszólítással, majd jól felépítetten, logikusan fogalmazzuk meg közlendőnket! Ne felejtünk el a levél végén elbúcsúzni és aláírni!

Az e-maileket két nagy csoportba sorolhatjuk:

1. **Személyes e-mail:** Ebben az esetben az e-mailt barátainknak, családtagjainknak címezzük. Ezek a levelek lehetnek kötetlenebb hangvételűek, azonban itt se írjunk olyat, amit nem írnánk rá egy levelezőlapra! A levél írásakor érzelmeinket egy-egy emoji (ejtsd: emodzsi) segítségével fejezhetjük ki.
2. **Hivatalos e-mail:** Tanárainknak, hivatalos személynek, szervezetnek címezzük. Törekedjünk a rövid, tömör és érthető megfogalmazásra! Kerüljük az emoji használatát!

Figyelem!

A levélírásakor ügyeljünk a megfogalmazás stílusára! A helyesírás szabályai akkor is érvényesek, ha online levelezünk valakivel. Azonban ilyenkor gyakrabban használunk rövidítéseket, hangulatjeleket.

Feladat

1. Képzeld el, hogy többnapos osztálykiránduláson veszel részt! Írj egy e-mailt családtagjaidnak, amiben beszámolsz az elmúlt napok élményeiről!
2. Írj egy e-mailt a tanárodnak, amiben segítséget kérsz egy házi feladatodhoz!
3. Nézz utána, hány e-mailt küldenek naponta a világhálón!

Közösségi oldalak, chat

Az elmúlt évtizedben az interneten számos olyan felület jött létre, amely az emberek közti kapcsolattartást, kommunikálást segíti. Ezek az oldalakon regisztrált felhasználók hálózatot alkotnak. Ilyenek az úgynevezett *közösségimédia*-oldalak, például a Facebook, az Instagram, a Snapchat, a Twitter, a Skype, a Pinterest, a TicToc stb.

Mindegyik oldal más funkciókat biztosít a felhasználói számára. Ismerőseinkkel megoszthatunk képeket, rövid videókat, szöveges bejegyzéseket.

A *digitális lábnyomunk* a közösségi oldalak használatával egyre nő, hiszen ezek azok a felületek, ahol a legtöbb bejegyzést, képet, videót elhelyezzük!

Feladat

Beszélgétek meg, hogy milyen tartalmakat nem szabad online felületre feltölteni, másokkal megosztani! Miért?

Biztosan ti is hallottátok már az *internetes zaklatás*, *bántalmazás* kifejezést (cyberbullying, ejtsd: szájberbálliing). Ez azt jelenti, hogy általában egy közösség összefog egy kiválasztott személy ellen, és őt bántó és fenyegető üzenetekkel, tartalmakkal zaklatja. Az interneten történő kiközösítés, fenyegetés történhet közösségi oldalakon vagy e-mailben is.

Figyelem!

Ha úgy érezzük, internetes zaklatás áldozatai vagyunk, vagy tudomásunk van valakiről, akit zaklatnak, ne féljünk jelezni a szüleinknek, tanárainknak! Kérjünk segítséget!

Online csevegés, beszélgetés

A legtöbb szolgáltató biztosít online beszélgető vagy *csevegő (chat)* lehetőséget. Ezek az oldalakon más regisztrált felhasználókkal kommunikálhatunk írásban.

A csetoldalak segítségével a felhasználók könnyen tudnak egymással kapcsolatot teremteni a világ minden pontjáról.

Ezek a csevegőszolgáltatások nemcsak számítógépen, hanem okostelefonon vagy tableten is elérhetők a felhasználók számára.



A csevegés nyolc fő szabálya:

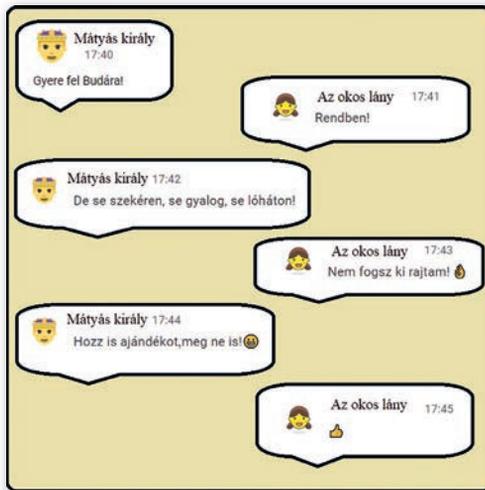
1. Csak olyan személlyel beszélgetsetek, akiket valóban ismertek! Ne feledjétek, bárki rejtőzhet álnév mögött!
2. A beszélgetés elején és végén köszönjétek, mint a valós beszélgetésekben!
3. Ne tegyetek sértő megjegyzést a társatokra! Mások véleményét tartsátok tiszteletben!
4. Ne használjatok trágár szavakat!
5. Ne osszatok meg magatokról olyan képet, felvételt, amellyel bárki visszaélhet!
6. Soha ne osszatok meg személyes adatot magatokról!
7. Érzelmeitek kifejezéséhez használjatok emojikat!
8. A kapott üzenetekre – ha röviden is – reagáljatok! Illetlenség visszajelzés nélkül hagyni egy elolvasott üzenetet!

Figyelem!

A felhasználói fiókok nem mindig a valós egyént takarják. Jusson eszetekbe, hogy egy álnév mögött bárki rejtőzhet! Ne bízzatok meg idegenben! Soha ne adjatok meg személyes adatokat! A gyanús felhasználókat jelentsétek és tiltsátok le!

Feladatok

1. Játsszátok el, hogyan tudjátok megvédeni magatokat az online beszélgetések során! Milyen lehetőségei vannak a beszélgetés lezárásának?
2. Képzeljétek el, hogy a történelmi személyek vagy mesehősök csetelnek egymással! Csoportos munkában alkossátok meg a párosok beszélgetéseit!



Felhőszolgáltatások

A világhálón léteznek olyan weboldalak, melyek különböző méretű tárhelyet biztosítanak a felhasználók számára. Ezek az úgynevezett *felhőszolgáltatások*. Ilyen például a *Google Drive*, az *Apple iCloud*, a *Dropbox* vagy a *Microsoft OneDrive*.

A tárhely meghatározott adatmennyiségig minden regisztrált felhasználó számára ingyenesen hozzáférhető, azonban ha bővíteni szeretnénk, már fizetnünk kell érte.

Regisztrált felhasználóként feltölthetjük képeinket, dokumentumainkat a tárhelyünkre.

A feltöltött fájlokat mappákba tudjuk rendezni, rendszerezni. A felhőszolgáltatások felületén hasonló mappaműveleteket végezhetünk, mint az asztali eszközünkön.

A felhőszolgáltatást nyújtó weboldalak használata lehetőséget biztosít számunkra, hogy ne kizárólag a számítógépünkön tároljuk fájljainkat, takarékoskodva az eszköz kapacitásával. Ha van internetkapcsolatunk, akkor a szolgáltatásba bejelentkezve különböző eszközökön (laptop, telefon stb.) és helyszíneken – otthon, iskola, barátaink – is hozzáférünk az ott tároltakkhoz. A felhőbe feltöltött, akár nagyobb méretű dokumentumainkat, fájljainkat másokkal is meg tudjuk osztani. A velünk megosztott tartalmakat letölthetjük a különböző digitális eszközeinkre.

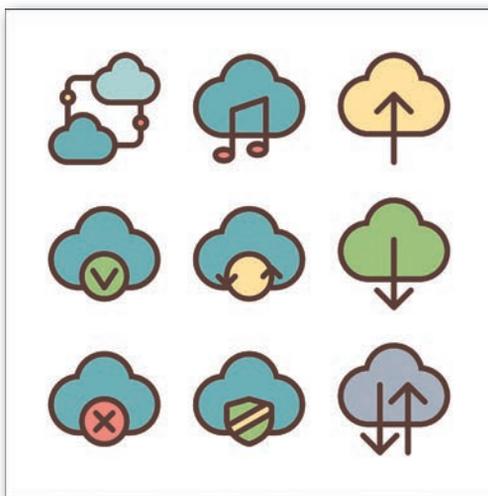
Feladat

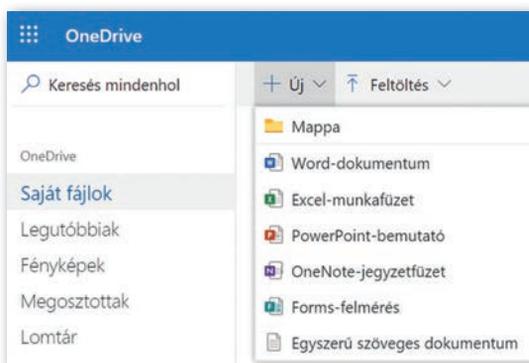
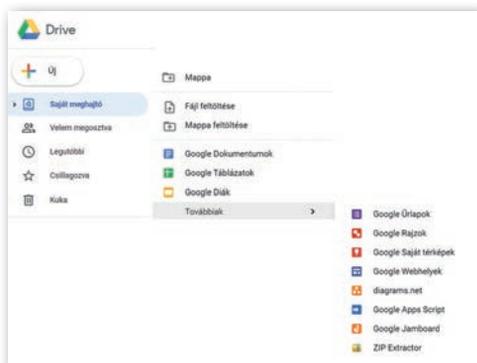
A korábban tanult módon keressetek rá egy általatok választott történelmi személyiséget ábrázoló képre! Letöltés után a tanáraitok vagy szüleitek segítségével helyezétek el a képet a kiválasztott felhőszolgáltatásban! Osszátok meg egymással a találatokat!



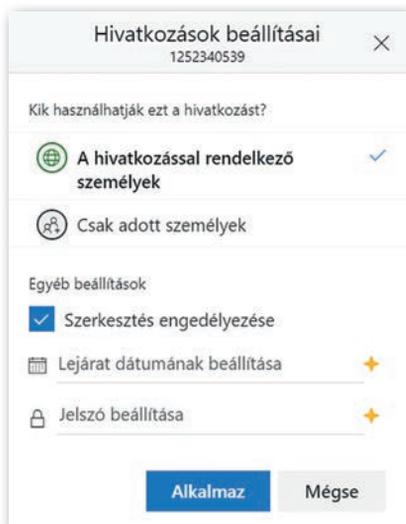
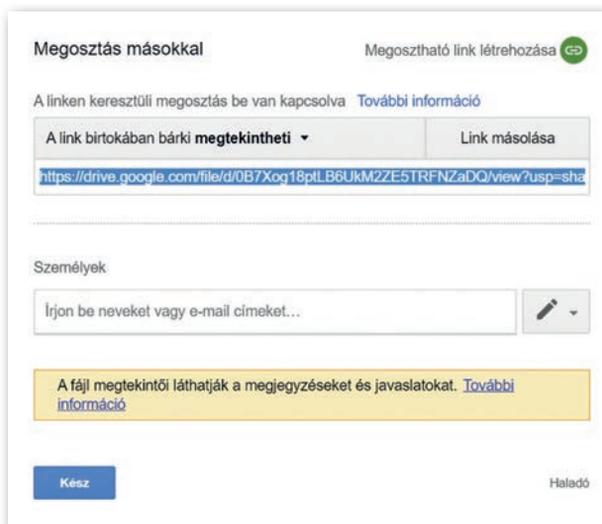
A szolgáltatások lehetőséget biztosítanak számunkra, hogy különböző tartalmakat létre tudjunk hozni: szöveges dokumentumot, táblázatot, prezentációt, kérdőívet.

A következőkben a *Google Drive* és a *Microsoft OneDrive* szolgáltatókkal ismerkedünk meg közelebbről.





A létrehozott dokumentumokat is lehetőség van megosztani más felhasználókkal. A megosztás többféle módon lehetséges: e-mailen keresztül vagy megosztható link létrehozásával.



Lehetőségünk van arra, hogy beállítsuk társaink hozzáférését *megtekintésre*, *megjegyzések elhelyezésére*, illetve *szerkesztésre*.

Feladatok

1. Hozzatok létre három-négy fős csoportokban egy szöveges dokumentumot a kiválasztott felhőszolgáltatásban, és készítsetek rövid ismertetőt az előző feladatban kiválasztott történelmi személyiség életéről! Illusztráljátok képekkel!
2. Hozzatok létre három-négy fős csoportokban egy rövid bemutatót! Használjátok a felhőszolgáltatás funkcióit! Osszátok meg társaitokkal az elkészült munkákat!
3. Három-négy fős csoportokban készítsetek felmérőt az Űrlapok segítségével! Illesszetek bele többféle választási lehetőséget! Osszátok meg társaitokkal!

Rendszerezés, összefoglalás

Feladatok

1. Keresd a párját! Kösd össze!

A matching exercise with the following terms in colored boxes:

- hálózat (light blue)
- elektronikus levél (yellow)
- online zaklatás (purple)
- e-mail (pink)
- cyberbullying (green)
- felhőszolgáltatás (orange)
- Google Drive
Microsoft OneDrive (yellow)
- összekapcsolt számítógépek (grey)

2. Csoportosítsd a tevékenységeket aszerint, hogy igényelnek-e internetkapcsolatot! Figyelj, van, ami mindkét halmazba jó!

Activities to be classified:

- biciklizni
- sakkozni
- útvonalat tervezni
- csetelni
- képet csatolni
- letölteni egy képet
- rajzolni
- e-mailt írni
- keresni az interneten
- telefonálni
- rákeresni egy menetrendre

Categories:

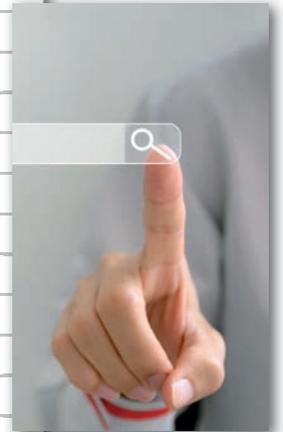
- Kell hozzá internet
- Nem kell hozzá internet

3. Milyen hangulatot fejeznek ki a következő jelek?



4. Három-négy fős csoportokban terveztek meg egy egynapos osztálykirándulást Magyarország egyik nagyvárosába vagy a településeken található nevezetességek egyikehez! Használjátok a fejezetben tanult lehetőségeket!
- Terveztétek meg az útvonalat, időtervet!
 - Kerestétek meg a város nevezetességeit! Készítsetek rövid leírást a látnivalókról!
 - Járjatok utána, milyen költségek merülnek fel a kirándulás során (utazási költség, belépők stb.)!
 - Az összegyűjtött anyagból készíttetek rövid prezentációt társaitok számára!

Jegyzetelj!



A számítógépek és főbb alkatrészeik

Hol vannak körülöttünk számítógépek?

Manapság elég nehéz bemenni egy olyan szobába, ahol nincs számítógép, hiszen számítógép van a legtöbb gépünk belsejében. Számítógép van a telefonban és a tabletben, számítógép van a tévében, a mosógépben, a mikrohullámú sütőben, a bonyolultabb porszívóban, a kenyérsütőben, az autóban, sok termosztátban (hőmérséklet-szabályozóban), kapunyitóban, a boltok pénztárgépeiben és elektromos árkijelzőiben. Bár nem önállóan működő számítógép, de chip van a kutyák bőre alá ültetve, chip van a bankkártyában és a személyi igazolványban is.

Általános célú számítógépek

Az említett számítógépek közül a legtöbb egyetlen feladatot végez. A termosztátban lévő a fűtést és a légkondicionálót kapcsolgatja ki-be. A kapunyitó számítógépe megismeri a mi távirányítónkat, és annak engedelmessé nyitja-zárja a kaput, de érzékelőjével észreveszi, ha valami épp a kapuban állt meg, és olyankor vár.

A hagyományos értelemben vett számítógépek (a laptopok és az asztali gépek), az okos-telefonok és a tabletek sokféle programot tudnak futtatni. Használhatunk rajtuk például internetböngészőt, képnézegetőt, rajzolóprogramot, játékokat. Az ilyen számítógépek általános célú gépek, mert sokféle programot telepíthetünk rájuk, és ezektől a programoktól függ, hogy mire használhatók. A programok mondják el a számítógépnek, hogy mi a számítógép feladata.

E tantárgy óráin a legtöbb esetben mi is általános célú számítógépekkel foglalkozunk.

Hogyan futnak a programok?

Szokás mondani, hogy a számítógép „agya” futtatja a programokat – valójában a számítógépnek nincs agya. A mi emberi agyunk a dolgokat nagyon-nagyon leegyszerűsítve két dolgot tesz:

- megjegyez dolgokat,
- és ezekkel a megjegyzett dolgokkal műveleteket végez.

Például tudjuk az iskola helyét és azt, hogy hol lakunk. Erre a két megjegyzett ismeretre támaszkodva tudunk térképet rajzolni, hogy miként találunk haza. Tudjuk, hogy reggel óta nem evett a háziállatunk, és tudjuk, hogy hány óra alatt szokott megéhezni, így meg tudjuk mondani, hogy mikor érdemes megetetni.

A számítógépben a fent említett két dolgot két külön alkatrész végzi. A két alkatrész a számítógépben másodpercenként akár sok milliószor „beszélget” egymással gépi jelekkel. Nagyon leegyszerűsítve és némileg emberivé formálva a két alkatrészt, valahogy így beszélgethetnek:

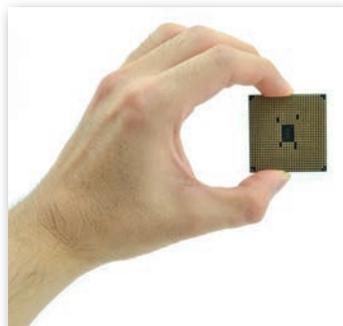
- Te figyelj, memória, most kaptam egy tök izgi feladatot, ki kell számolnom, hogy mennyi kettő meg három! Jegyezd meg, hogy mit kell csinálni, jó?
- Jó, processzor, megjegyzem.
- Juj, imádok számolni! Ööö... Mit is kell kiszámolni?
- Hogy mennyi kettő meg három.
- Ja, tényleg, emlékszem! És mennyi?
- Én tudjam?! Te tudsz számolni, nem én!
- Ja, most, hogy mondod... Kettő meg... Ööö... Mennyit is kell hozzáadnom?
- Hármat.
- Kettő meg három az öt. Jegyezd meg!
- Jó.
- És kell még valamit csinálni?
- Hát, gondolom, el kéne mondani a felhasználónak. Itt a programban az van, hogy írjuk ki a kijelzőre.
- Ja, jó, akkor szólok neki, hogy írja ki, hogy... Ööö, mit is?
- Hogy öt.

- Melyik két alkatrész beszélget?
- Melyiknek mi a feladata?

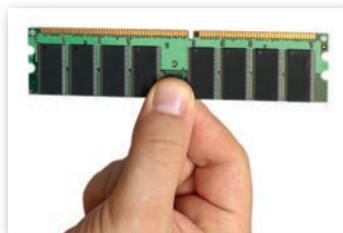
A fenti beszélgetésből tudjuk, hogy a számítógép **memóriája** (angol nevének rövidítésével: RAM) képes adatokat és programokat megjegyezni. A megjegyzett adatokkal pedig a számítógép feldolgozóegysége, idegen szóval **processzora** (angol nevének rövidítésével: CPU) képes műveleteket végezni.

A processzorok **teljesítményét** két számadattal szokás jellemezni. Az egyik a processzor gyorsaságát jellemző **órajel**, amit **gigahertzben** (GHz) mérünk. Két hasonló processzor közül az a gyorsabb, amelyiknél ez a szám nagyobb.

Egy általános célú számítógép több programot is futtat egyszerre. Nézzük csak meg a gépünk feladatkezelőjét! Hány program fut most a gépünkön? Aligha számolunk ötvennél kevesebbet. A számítógép ezeket úgy futtatja egyszerre, hogy váltogat köztük, mégpedig villámgyorsan. Mintha otthon nagytakarításkor fölporszívóznánk a szék mellett, letörölnénk egy polcot, kiterítenénk egy ruhát, aztán porszívóznánk az asztalnál, letörölnénk egy másik polcot, kiterítenénk a következő ruhát, és így tovább. Ha túl sok feladatot végzünk, akkor is mind halad, csak na-



▶ Processzor (CPU)



▶ Memória (RAM)

gyon lassan. A mai processzoroknak több magjuk szokott lenni. A **magok száma** a teljesítményt jellemző másik szám. Az előző gondolatot folytatva a második mag olyan, mintha jönne belőlünk még egy, és ő is segítene takarítani: elmos egy edényt, letöröl egy ablakot, összehajt egy ruhát, és folytatja a következő edénnyel.

A **memória méretét gigabájtban** (gigabájt, GB) szokás megadni. Láttuk, hogy a memória tartalmazza a programokat és az adatokat – legalábbis amíg kap áramot. Ha a gépet, telefont kikapcsoljuk, a memória tartalma egyszer s mindenkorra elvész.

- Hogyan kell megnézni a gépünk feladatkezelőjét? Hány program fut most a gépünkön?
- Mennyi memória van a számítógépünkben, telefonunkban?
- Hány magja van a számítógépünk, telefonunk processzorának?
- Hol készült a számítógépünk, telefonunk processzora?
- Mennyibe kerül egy processzor? Mennyibe kerül annyi memória, amennyi a számítógépünkben van?

Hogyan beszélünk mi a számítógéphez, és hogyan mondja el a gép, amit nekünk szán?

A számítógép vagy az okostelefon processzora és memóriája, valamint az alkatrészeket összekapcsoló **alaplap** már elég ahhoz, hogy az áramot kapó számítógép működjön. Azonban nem sok hasznát vesszük az olyan gépnek, amiben nincs más alkatrész: nem tudunk feladatot adni számára, és nem tudja elmondani nekünk az elvégzett feladat eredményét.

A számítógépnek feladatot, feldolgozandó adatot a **bemeneti perifériákkal** adunk: a billentyűzettel, az egérrel, a mikrofonnal, a kamerával. A processzor az adatokat feldolgozza, és az eredményt a **kimeneti perifériák segítségével** tudatja velünk. Ilyen eszköz a monitor vagy kijelző, a hangszóró, a nyomtató.

Hol tárolja a gép azokat az adatokat, amiket most épp nem használ?

Hova kerülnek az adatok kikapcsoláskor?

A számítógép memóriája hamar betelik, kikapcsoláskor pedig mindent elfelejt. Azok a programok, amelyek épp nem futnak, és azok az adatok (képek, videók, zenék, szövegek), amelyek épp nem kellenek, a gép **háttértárára** kerülnek. Gépünk az adatokat fájlok formájában menti a háttértárra. A háttértár is adatot tárol, ezért kapacitását éppen úgy gigabájtban, illetve az ezer gigabájtval egyenértékű terabájtban fejezzük ki, mint a memóriáét. A számítógépek jellemző háttértára az **SSD**, az okostelefonoké pedig a **memóriakártya**. A háttértárak a tartalmukat áramtalanítás után is megőrzik.

- Mi a különbség a memória és a memóriakártya között?
- Van az okostelefonokban memória is?
- Mi a neve az olyan háttértárnak, amelyen fájlokat szoktunk hordozni például az otthoni és az iskolai számítógép között?
- Mekkora a számítógépünk, telefonunk háttértára? Hány fájl van rajta?
- Mennyibe kerülnek a háttértárak?

Programok, operációs rendszerek, fájlok és mappák

Az előző órán láttuk, hogy milyen alapvető alkatrészekből áll egy számítógép. Ma megtudjuk, mi kell ahhoz, hogy a számítógépünk programokat is tudjon futtatni.

Mi az az operációs rendszer?

Az egyszerűbb, egyetlen célfeladatra szánt számítógépek csak egyetlen programot futtatnak. Ilyen számítógép a termosztát, a robotporszívó vagy a fényképezőgép. Ezt a programot induláskor automatikusan betöltik a memóriájukba, vagy eleve olyan memóriájuk van, amelyben ez mindig rendelkezésre áll – ilyenkor be sem kell tölteniük.

Az általános célú számítógépeknél (okostelefonoknál, táblagépeknél) más a helyzet. Az ilyen számítógépek sokféle alkalmazást, programot képesek futtatni.

- Hogyan tudunk egy alkalmazást elindítani?
- Soroljunk fel a számítógépünkön futó programokat!
- Melyik az a program, amelyiket nem mi indítjuk el, mégis mindig elindul a számítógép vagy a telefon bekapcsolását követően?

Valójában mindig van egy olyan program a számítógépeken, telefonokon, amelyiket nem mi indítjuk el. Ez a program a számítógép **operációs rendszere**. Az általános célú számítógépek operációs rendszer nélkül használhatatlanok.

- Milyen operációs rendszer van a számítógépeden?
- Milyen ismertebb operációs rendszereket használnak az okostelefonok?

Hogyan töltődik be az operációs rendszer?

Amikor a számítógépet bekapcsoljuk, az első dolga, hogy ellenőrzi, megvan-e az összes feltétlenül szükséges alkatrésze.

- Melyek azok az alkatrészek, amik mindenképp kellenek egy számítógépbe?

Ha a számítógép úgy találja, hogy megvan minden, akkor megnézi, hogy milyen háttér-tárak elérhetők számára, és hogy melyiken van operációs rendszer. Az elsőként megtalált operációs rendszert automatikusan betölti. A számítógépnek (telefonnak, táblagépnek) ezt az indulási folyamatát idegen szóval **bootolásnak** nevezzük – amikor a számítógép elindul, akkor idegen szóval bootol, felbootol.

Mire való az operációs rendszer?

Az operációs rendszer arra kell, hogy a többi program, alkalmazás futását felügyelje. Az operációs rendszer

- **elindítja** (automatikusan vagy a mi parancsunkra) és megállítja a programokat;
- figyel arra, hogy melyik program mennyire használhatja a **processzort**;
- figyel arra, hogy az egyes programok a **memória** melyik részét használják (nehogy összekeveredjenek az egyes programok adatai a memóriában);
- hozzáférést biztosít az alkalmazásoknak a bemeneti és a kimeneti **perifériákhoz** (például figyel arra, hogy amit az egyik programba akarunk beírni, az ne a másikba íródjon, vagy hogy amíg az egyik program nyomtat, ne nyúlhasson a nyomtatóhoz a másik); és
- figyel arra, hogy melyik **fájlhoz** melyik program, és a számítógép melyik felhasználója férhet hozzá.

Mik azok a fájlok?

A fájlok (más néven állományok) olyan adatok, amelyeket a számítógép a háttértárra írt.

Ha például készítünk egy rajzot a rajzolóprogramban, akkor a rajz a számítógép memóriájába kerül. Ahogy a rajzot alakítgatjuk, úgy változik a memóriában. Amikor a rajzot **mentjük**, akkor a rajzolóprogram a rajz adatait átadja az operációs rendszernek, és megkéri, hogy mentse el a háttértárra. Az operációs rendszer helyet keres az adatok számára a háttértáron, és egy fájlba, más szóval adatállományba leírja, amit a rajzolóprogramtól kapott. Végül visszajelez a rajzolóprogramnak, hogy sikerült-e a memóriában lévő adatok fájlba írása – ha például betelt vagy elromlott a háttértár, akkor szól, hogy nem sikerült.

Hogy néz ki a fájl neve?

Amikor egy alkalmazásból fájlt mentünk, az alkalmazás általában tőlünk kérdezi meg a fájl nevét. A név nekünk fontos, abból tudjuk, hogy mit mentettünk a fájlba.



A név végére az alkalmazás általában egy kiterjesztést illeszt ponttal elválasztva. A kiterjesztés a tárolt adat jellegére, típusára utal. A számítógép vagy a telefon és a hozzáértő számítógép-felhasználó ember a kiterjesztésből tudja, hogy milyen adat van a fájlban. A kiterjesztés általában nem hosszabb három-négy karakternél, ami azért alakult így, mert a múlt évezredben sok számítógépen legfeljebb hárombetűs kiterjesztéseket lehetett használni. Amikor fájlnevet mondunk, néha csak a kiterjesztés nélküli részt értjük rajta, máskor a kiterjesztést is beleértjük a fájl nevébe.

- A te számítógépeden lévő operációs rendszer megmutatja a kiterjesztéseket? Ha nem, hogyan lehet beállítani, hogy megmutassa?

Mi minden lehet egy fájlban?

A fájlokban mindig adat van, de az adatok nagyon sokfélék lehetnek, sok mindent tárolhatnak. Az alábbi táblázat csak a leggyakoribb adattípusokat és az egyes adattípusoknál is csak a leggyakoribb kiterjesztéseket mutatja be.

Az adat jellege	Az ilyen jellegű adatra utaló néhány lehetséges kiterjesztés
kép	.jpg .png .tif .gif
zene vagy bármilyen hang	.mp3 .mp4 .mkv .wav
mozgóképfilm (videó)	.mkv .mp4 .avi .mpeg
szöveg	.txt .docx .odt
bemutató	.pptx .odp
tömörített adatot tartalmazó fájl	.zip .7z .rar
weblap	.html
program, alkalmazás	.exe .py .sh sokszor nincs kiterjesztésük az ilyen fájloknak

- Találunk-e olyan kiterjesztést, ami többféle adatra is utalhat?
- Vajon mit találunk a Csilus és Zokni.jpg fájlban?

Mire valók a mappák?

A számítógép háttértárán rengeteg, akár milliónyi fájl szokott lenni. Ezek között úgy szokás rendet tartani, hogy azokat a fájlokat, amik valamiért összetartoznak, egy mappában (más néven könyvtárban) helyezük el. Egy mappában lehetnek például egy játékprogram fájljai: maga a program, a szereplők képei és hangjai, a bejárható területet, a pályákat leíró fájl.

Mi magunk tehetjük egy mappába például azokat a képeket, amelyeket egy témáról készítettünk: az egyik mappába kerülnek a kirándulás képei, a másikba a háziállatunkról, a harmadikba a születésnapi buliról készült képek.

- Hogyan kell létrehozni mappát a számítógépünkön?
- Hogyan tudunk egyik mappából a másikba másolni fájlokat?
- Mi a különbség a fájlok másolása és mozgatása között?
- Tudunk-e egész mappákat is másolni, mozgatni?