



OKTATÁSI
HIVATAL

NAT
2020

Digitális kultúra

tankönyv

8



A kiadvány 2022. 12. 02-től 2027. 08. 31-ig tankönyvi engedélyt kapott a TKV/2439-8/2022 számú határozattal. A tankönyv megfelel a Nemzeti alaptanterv kiadásáról, bevezetéséről és alkalmazásáról szóló 110/2012. (VI. 4.) Korm. rendelet alapján készült, 2020. 01. 31. után kiadott, 5–8. évfolyam digitális kultúra tantárgy kerettantervének.

A tankönyvvé nyilvánítási eljárásban közreműködő szakértő: Györgyi Tamás

Tananyagfejlesztők: Abonyi-Tóth Andor, Farkas Csaba, Varga Péter

Kerettantervi szakértő: Farkas Csaba

Lektor: Siegler Gábor

Szerkesztő: Széll Szilvia

Fotók: © Shutterstock (24 AlesiaKan, 77 Lenscap Photography, 82 Poravute Siriphiroon, 88 Barone Firenze, 93 sitthiphong, 96 Olga Popova, 105 ZeroSystem), Pixabay (34, 35, 95, 104), Wikimedia (97 Reinraum)

© Oktatási Hivatal, 2023

ISBN 978-963-436-383-5

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-DIG08TA

Tankönyvkiadási osztályvezető: Horváth Zoltán Ákos

Műszaki szerkesztő: Kurucz Klára

Nyomdai előkészítés: Korda Ágnes

Terjedelem: 10,01 (A/5) ív, tömeg: 230 gramm

1. kiadás, 2023

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.

Gyártás: Könyvtárellátó Nonprofit Kft.

Nyomtatta és kötötte:

Felelős vezető:

A nyomdai megrendelés

törzsszáma:

SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

Előszó	5
Algoritmizálás és blokkprogramozás	7
Bevezetés és ismétlés	7
Játsszunk az algoritmusokkal! Egyszerű algoritmusok tervezése	8
Ciklusok és elágazások	13
Típusalgoritmusok használata	16
Függvények használata	18
Programozzunk micro:bitet!	24
Készítsünk játékokat micro:bitre!	27
Összefoglalás	30
Multimédiás elemek készítése	31
Ballagási tablót készítünk	31
Hangfelvétel készítése és egyszerűbb utómunkálatai	37
Egyszerűbb videószerkesztési műveletek	40
Videóötletek	43
Táblázatkezelés	49
A táblázatkezelés alapjai	49
Táblázat formázása. Cellahivatkozások	53
Diagramok készítése	58
Statisztikai számítások	64
Számformátumok	68
Kétirányú elágazás	73
Gyakorlófeladatok	75
Az e-világ és az online kommunikáció	77
Az információ értéke	77
Az információ ára	81
Mindennapok az információs társadalomban	84
Valós ember – virtuális személyiség	88
A virtuális világ és hatása az egészségünkre	92
Az információ évezredei	95
A digitális eszközök használata	99
Séta a földi hálózatokban és a felhőben	99
Képek és térképek	102
Informatikai kislexikon	106

A digitális kultúra világának áttekintése

Általános iskolai tanulmányaink utolsó évében ideje készítenünk egy mérleget arról, hogy mit tanultunk eddig digitális kultúrából, és mi van még hátra. Az informatika világa rendkívül szer-teágazó, és igen gyorsan fejlődik. Ennek ellenére elég jól körvonalazhatók a szükséges ismeretek.

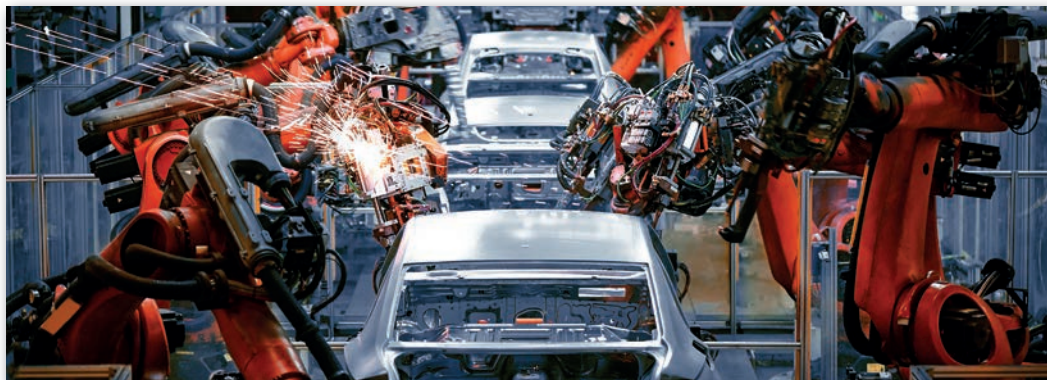
Talán a legszélesebb körben használt terület a dokumentumkészítés. Ha a dokumentumot papíron akarjuk átadni viszonylag szűk körben, akkor *szövegszerkesztő* programot használunk. A szövegszerkesztés a dokumentumkészítés alapja, ezeket az ismereteket a hatodik és a hetedik évfolyamon tanultuk. Ha a dokumentumot például könyv vagy újság formájában szeretnénk terjeszteni, akkor *kiadványszerkesztő* programra van szükségünk. Ennek a könyvnek a papíralapú változata is így készült, azonban a kiadványszerkesztés már olyan speciális ismeret, amellyel csak a szakirányú szakképzésben tovább tanulók fognak megismerkedni. Ha a dokumentumot egy kisebb publikum előtt szeretnénk kivetíteni (például egy másik tanórán vagy egy pár száz fős konferencián), akkor *be-mutatókészítő* programot alkalmazunk. Prezentációkészítéssel már ötödikes korunkban is találkoztunk, az idén ezeket az ismereteket csak átismételjük. Végül, ha a dokumentumot az egész világ számára elérhetővé szeretnénk tenni, akkor azt a webre kell kitennünk, ahogy ennek a könyvnek az esetében is történt. Korábban úgy gondolták, hogy a weblapszerkesztést a szövegszerkesztéshez hasonlóan fogják megvalósítani (ezért akár egy szövegszerkesztővel készített dokumentumot is menthetünk weblapként), azonban a világ más irányba haladt, és a *weblapkészítés* ma már talán közelebb áll a programozáshoz, mint a szövegszerkesztéshez. A weblapkészítéssel középiskolai tanulmányaink során fogunk megismerkedni.

Egy másik fontos terület az adatkezelés: ekkor az adatokat áttekinthető módon, egy vagy több táblázatban rendezzük el. Jóllehet erre egyszerűbb esetben a szövegszerkesztő program (sőt, akár egy négyzetrácsos papír) is elegendő, azonban az adatok megfelelő kezelésére többféle alkalmazás készült. Az idén megismerkedünk a *táblázatkezelő* programokkal. A táblázatkezelő programokban a táblázat – vagy „számlótábla” – képes arra, hogy a táblázat egyes celláiban az adatokat a többi cella tartalmából automatikusan kiszámítsa (például az utolsó sorban összegezze a fölötte lévő számokat). Gyakran előfordul, hogy munkánk elvégzésére nem elegendő egyetlen táblázat (például egy iskolában a *tanulókat* vagy a *tanulók osztályait* érdemes külön táblázatba rendezni), illetve az is, hogy nagyon sok adatunk van (például egy bevásárlóközpont napi eladásai). Ilyenkor *adatbázis-kezelő* rendszereket alkalmazunk. Adatbázis-kezelő rendszerekkel (például menetrendek, internetes vásárlások) ugyan rendszeresen találkozunk, azonban az adatbázis-kezelés olyan ismeret, amelyet alaposabban csak a középiskolában fogunk elsajátítani.



A digitális kultúra világának érdekes és látványos területe a grafika. A képeket, ábrákat kétféleképpen kezelhetjük. A fényképek képpontokból állnak, a fényképek kezelésére *pixelgrafikus* programot használunk. Ilyennel már az ötödik évfolyamon találkoztunk. Egy másik megoldás, amikor az ábrát vonalakból, görbékből, alakzatokból építjük fel, melyek hosszát, szögét és egyéb matematikai jellemzőit adjuk meg. Ez utóbbi esetben *vektorgrafikáról* beszélünk; vektorgrafikus ábrákat szövegszerkesztő programmal magunk is készítettünk. Vannak azonban kifejezetten vektorgrafikus alkalmazások, amelyekkel akár mérnöki tervrajzokat, például háromdimenziós alkatrészeket is tervezhetünk. Ez is olyan ismeret, amellyel csak későbbi tanulmányainkban fogunk találkozni. A képfeldolgozáshoz kapcsolódó, érdekes terület a *hang- és filmfeldolgozás*, ennek alapjait ebben a könyvben tekintjük át.

Az eddig említett ismeretek mindenki számára fontosak („digitális írástudás”). Ugyanakkor a digitális kultúra tantárgynak fontos szerepe van abban, hogy megértsük, *hogyan működnek a számítógépek* (a laptopoktól az okostelefonokon át az autót gyártó robotokig). A legtöbb számítógép valamilyen eszközbe van beépítve a környezetünkben (például autók fedélzeti számítógépe, mobiltelefonok, robotporszívók), és sok esetben nem is gondolnánk, hogy egy számítógép vezérli az adott eszközt (például a meleg vizet előállító gázkazán vagy a mosogatógép). Ezért nagyon fontos volt, hogy megismerkedjünk a *robotika* alapjaival. A mikrokontrollerek, a robotok programozását a korábbi tanévekben a *blokkprogramozás* eszközeivel sajátítottuk el. A programozás folyamatának évtizedek alatt kialakult rendszere van, ebben az évben – általános iskolai ismereteink zárásaként – ezt fogjuk áttekinteni az *algoritmizálás* témakörében. Ma még a legtöbb programot nem



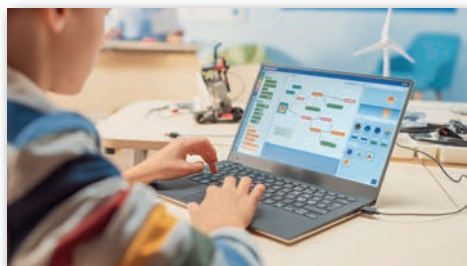
a blokkprogramozás segítségével készítik, hanem úgynevezett *programozási nyelveken*, azonban ezekkel – mint következő lépéssel – már csak a középiskolában fogunk találkozni.

Már eddig is nagyon sok mindent felsoroltunk, de a hardvert, az eszközök fizikai működését, a háttérben lévő matematikai ismereteket, az operációs rendszerek használatát, a hálózatok kezelését, az e-világ előnyeit és veszélyeit – amelyekkel szinte mindennap találkozunk – még meg sem említettük...

Kívánjuk minden olvasónk számára, hogy sikeresen ismerkedjenek meg a számítógépek világával, hasznosan alkalmazzák annak lehetőségeit, és ne féljenek felhasználni őket egy újabb probléma megoldása során! Ehhez kívánnak sok sikert

Bevezetés és ismétlés

Előzetes tanulmányainkban a micro:bit mikrovezérlő számos funkcióját használtuk már. Megtanultunk animációt készíteni, a LED-kijelző pontjait felkapcsolni, lekapcsolni. Játékokat fejlesztettünk, melynek során gesztusokat és gombokat is használtunk, sőt, a rádiókapcsolat segítségével adatokat küldtünk, illetve fogadtunk.



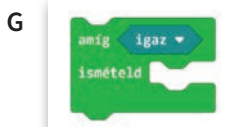
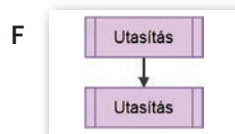
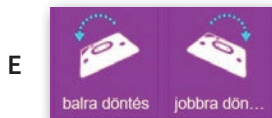
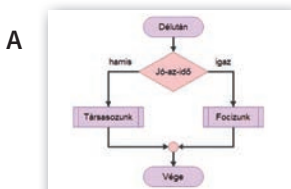
Nemcsak mikrovezérlőt programoztunk, hanem robotjárműveket is. Megtanultuk, hogyan haladhat a jármű egyenes vonalban, hogyan foroghat egy helyben, hogyan kanyarodhat, illetve követhet vonalat. Munkánk során többféle szenzort használtunk (fényérzékelőt, távolságérzékelőt, színérzékelőt), önállóan terveztünk programot valós, gyakorlati probléma megoldására.

Folytatásként elmélyítjük tudásunkat, illetve új, izgalmas eszközökkel és lehetőségekkel is megismerkedünk.

Feladat

Párosítsuk a fogalmakat a képekkel!

1. változó
2. egyszerű elágazás
3. végtelen ciklus
4. kétirányú elágazás
5. gesztusok
6. szekvencia
7. logikai műveletek
8. számlálós ciklus
9. feltételes ciklus



H Ha *feltétel* akkor utasítás(ok) elágazás vége



Játsszunk az algoritmusokkal! Egyszerű algoritmusok tervezése

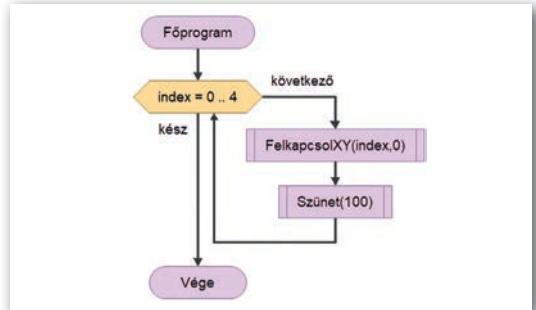
Ismerkedés a Flowgrithm alkalmazással

Korábban számos példát láttunk arra, hogyan lehet az algoritmusokat szöveges, mondat-szerű leírással, illetve folyamatábrával megadni.

Program

```
Ciklus index=0-tól 4-ig
  FelkapcsolXY(index,0)
Szünet(100)
Ciklus vége
Program vége
```

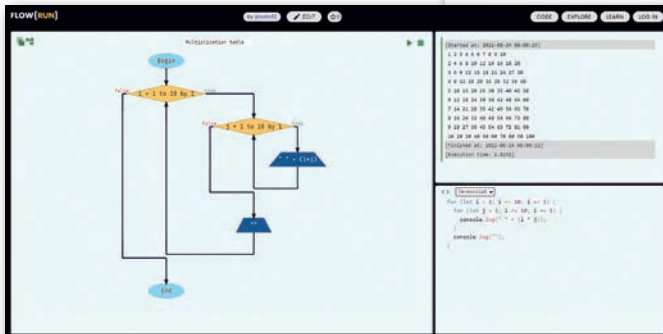
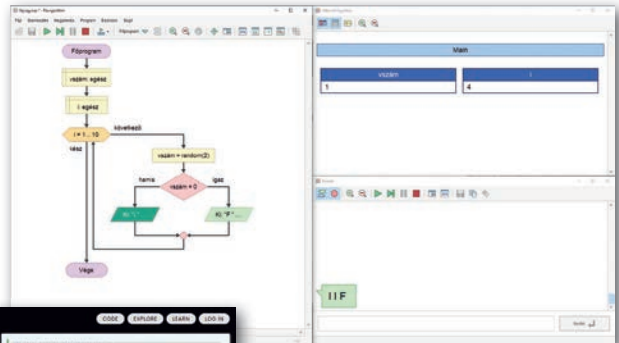
► Egy algoritmus mondat-szerű leírása



► A bal oldali algoritmus leírása folyamatábrával

Milyen jó volna, ha lenne olyan eszközünk, amellyel ezeket a folyamatábrákat egyszerűen elkészíthetnénk, sőt akár ki is próbálhatnánk a működésüket! Ezzel ugyanis meggyőződhetnénk arról, hogy helyesen működik-e az algoritmus, vagyis tesztelni tudnánk. A tesztelés során megvizsgálhatnánk az egyes változók értékeit anélkül, hogy ki kellene őket írni. Sőt, egy korszerű eszközzel a folyamatábrát akár futtatható programkóddá is alakíthatnánk.

A jó hír, hogy van ilyen eszköz (például a *Flowgrithm*), a következőkben ezzel foglalkozunk.

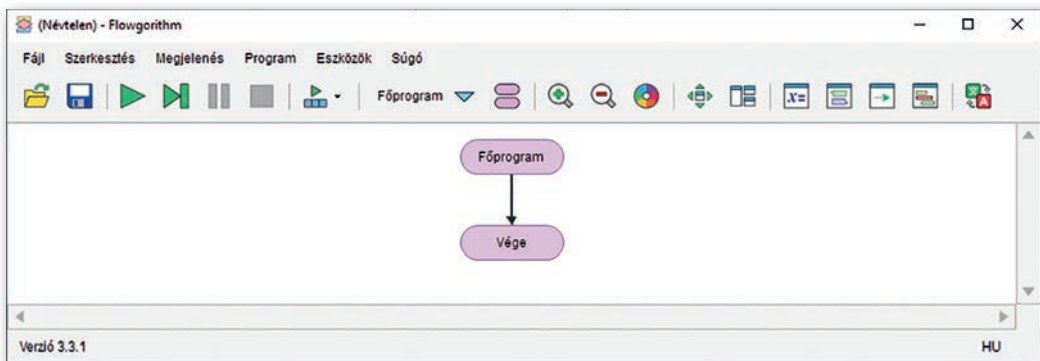


► Folyamatábra-készítő eszközök

A Flowgorithm alkalmazás felépítése, használata

Az algoritmusok készítését és tesztelését a *Flowgorithm* alkalmazással mutatjuk be. Ennek telepítőkészlete a <http://www.flowgorithm.org/> webcímről tölthető le Windows operációs rendszerre. Más operációs rendszereken való futtatáshoz is találunk segítséget az oldalon.

Az alkalmazás neve a *flowchart* (folyamatábra) és az *algorithm* (algoritmus) szavakból származik. A program indítása után egy üres program folyamatábrája jelenik meg a képernyőn, amely két dobozból (*Főprogram*, *Vége*) és egy nyílból áll.

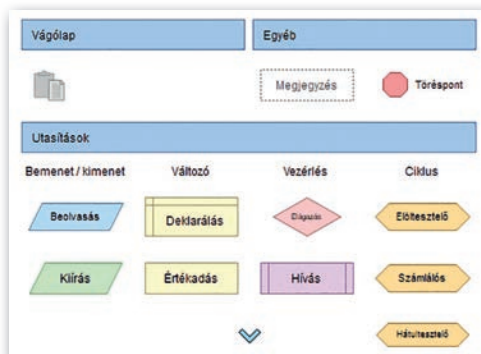


► A Flowgorithm alkalmazás kezdőképernyője

Láthatjuk, hogy a különböző típusú blokkok más-más színekkel jelennek meg a folyamatábrában. A programban számos színséma beállítható, amely befolyásolja az egyes blokkok színét. Érdeemes kezdetben az alapértelmezett színsémát használni, és csak akkor átállítani másikra, ha már megfelelően tudjuk használni a programot.

A folyamatábrát úgy bővíthetjük új elemmel, hogy a két blokk közti nyílra kattintunk. Ekkor az alábbi elemekből választhatunk:

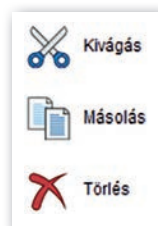
- A *Vágólap* segítségével a korábban vágólapra helyezett elemeket illeszthetjük be.
- Az elhelyezhető *utasítások* – algoritmuselemek – különböző csoportokba (például változó, vezérlés) vannak sorolva.
- Az *Egyéb* kategóriában található elemek segítségével pedig megjegyzést fűzhetünk a folyamatábrához, illetve töréspontot helyezhetünk el, amely a tesztelés során lehet fontos.



► Választható elemek és blokkok

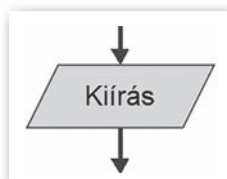
Ha beillesztettünk egy elemet, akkor a jobb egérgombbal rákattintva előhívhatunk egy hasznos menüt. Itt megtaláljuk a *Kivágás*, a *Másolás* és a *Törölés* műveleteket.

Törölni a *Del* billentyűvel is tudunk, kivágáshoz a **CTRL + X**, másoláshoz a **CTRL + C** billentyűkombinációt is használhatjuk Windows operációs rendszerben.



Írjunk ki egy szöveget!

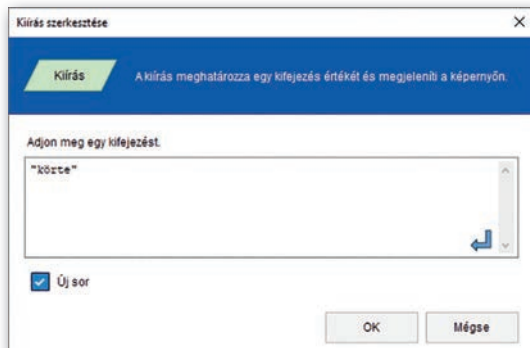
Kezdetnek készítsünk el egy egyszerű algoritmust, amely kiírja a kedvenc gyümölcsünk nevét! Ehhez használjuk a *Kiírás* utasítást! Azt tapasztaljuk, hogy a blokk a beszurása után szürke színnel jelenik meg, az alapértelmezett színsémát használva.



Ez a szürke szín arra figyelmeztet bennünket, hogy a blokkot még módosítani kell ahhoz, hogy használható legyen. Ha ebben az állapotban próbálnánk futtatni az algoritmust, hibaüzenetet kapnánk.

Módosítsuk tehát a *Kiírás* blokkot! Ehhez kattintsunk rá duplán!

A megjelenő ablakban megadhatjuk a kifejezést, amely legyen most a kedvenc gyümölcsünk neve. Vigyázzunk arra, hogy ha szöveget szeretnénk kiírni, akkor azt minden esetben **idézőjelek közé** kell tennünk!

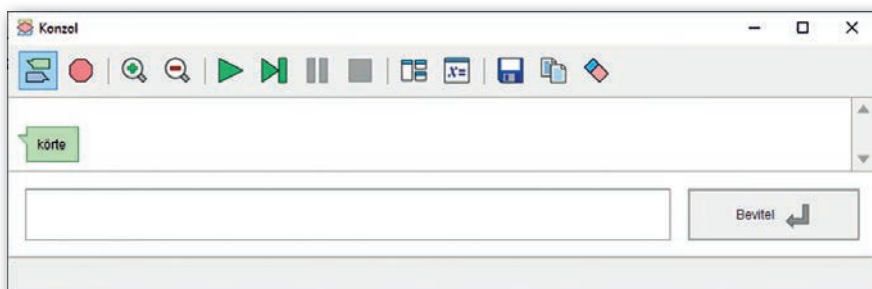
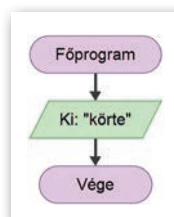


► A *Kiírás* szerkesztési ablaka, benne egy szöveggel

Az *OK* gomb megnyomása után megváltozik az algoritmus, most már zöld színnel jelenik meg a blokk, és láthatjuk benne a kiírt szöveget is.

Ezek után kipróbálhatjuk az algoritmust. Ehhez kattintsunk az eszköztáron a *Futtatás* (▶) ikonra!

A futtatás után a képernyőn megjelenik a *Konzol* nevű ablak, benne pedig a kiírt szöveg, illetve ha a program adatokat kér be, itt gépelhetjük be azokat is.



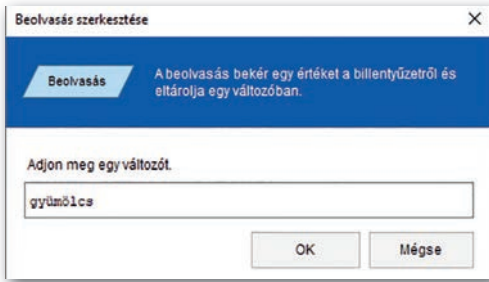
► A konzolablak, benne a kiírt szöveggel

Olvassunk be adatot változóba, és írjuk ki!

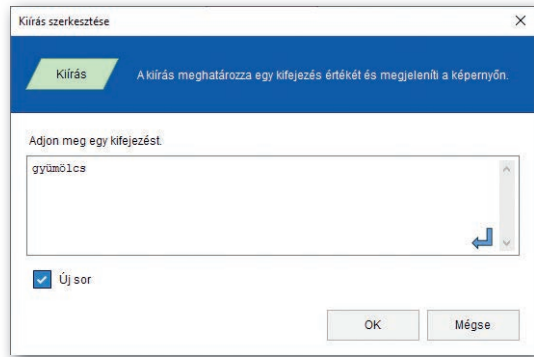
Módosítsuk úgy az algoritmust, hogy egy változóba kérjük be a kedvenc gyümölcsünk nevét, majd írjuk azt ki!

Az adat beolvasásához helyezzük el a *Beolvasás* blokkot! Kattintsunk rá duplán, és adjuk meg annak a változónak a nevét (például `gyumölcs`), amelybe a beírt szöveg kerül majd!

Módosítsuk úgy a *kiírás* blokkot, hogy a `gyumölcs` változó értéke kerüljön kiírásra! Ehhez adjuk meg a változó nevét! Fontos, hogy a **változónevet ne tegyük idézőjelek közé**.



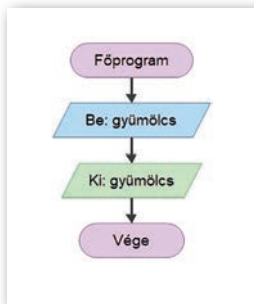
► Beolvasás változóba



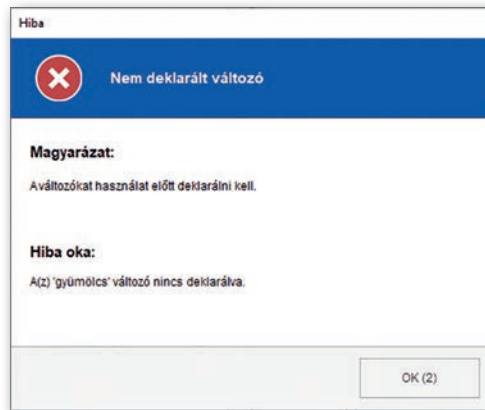
► Változó értékének kiírása

Ezzel elkészült az algoritmusunk. Ne felejtsük el elmenteni a *Mentés* (📁) ikon megnyomásával!

Próbáljuk meg futtatni az algoritmust! Sajnos ekkor azt tapasztaljuk, hogy egy hibaüzenet jelenik meg.



► A futtatandó algoritmus



► A megjelenő hibaüzenet (Nem deklarált változó)

Vajon mi okozza a hibát? Mit jelent az, hogy nem deklarált változó?

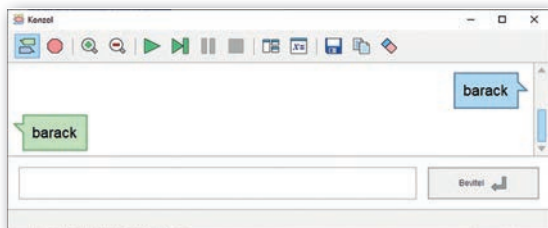
Azokban a blokkprogramozási nyelvekben, amelyekkel korábban dolgoztunk (*Scratch*, *MakeCode* stb.), ahogy a változóknak kezdő értéket adtunk, már használhattuk is őket. Más programozási nyelvekben viszont először **deklarálni** kell a változót, ami azt jelenti, hogy még az első használat előtt meg kell adnunk a változó nevét és típusát. A típus arra utal, hogy milyen jellegű adatot tárolunk el benne. Ez lehet például szöveg, egész szám, valós szám vagy logikai (igaz/hamis) érték.

Folytassuk tehát a változó deklarálásával! A változó értékének beolvasása előtt helyezük el a *Deklarálás* blokkot!

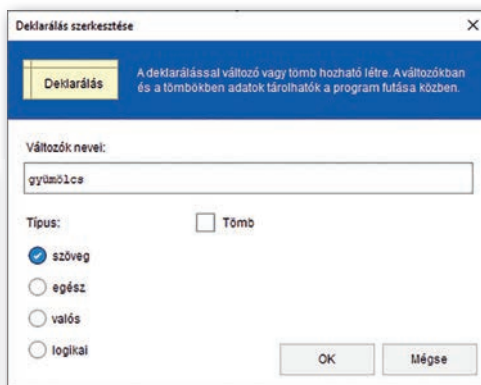
Kattintsunk rá duplán, majd adjuk meg a változó nevét (*gyümölcs*) és típusát (*szöveg*)! Vigyázzunk arra, hogy a változó nevét pontosan úgy írjuk le, mint a beolvasásnál! Ha például *gyümölcs* helyett *gyumölcs*-öt írunk, továbbra is hibaüzenetet kapunk.

Ezek után futtassuk le az algoritmust!

A konzolablakba begépelhetjük a szöveget, amely aztán a kiírás miatt újra megjelenik. Láthatjuk, hogy a beolvasott szöveg jobbra, a kiírt szöveg balra igazítottan jelenik meg, és a színük is különbözik.



► Beolvasás és kiírás a konzolablakban



► Deklarálás szerkesztése

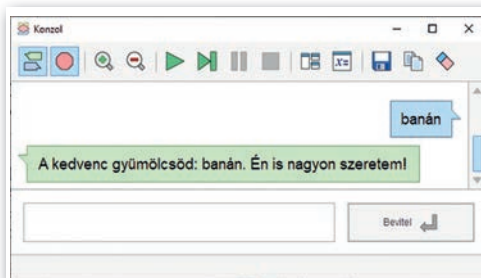
Most módosítsuk a kiírást úgy, hogy ne csak a gyümölcs nevét írjuk ki, hanem az alábbi szöveget:

„A kedvenc gyümölcsöd: *gyümölcs neve*. Én is nagyon szeretem!”

Kattintsunk duplán a *Kiírás (KI)* blokkra, és adjuk meg a kifejezést! Mivel most egymás mellett, egy sorban szeretnénk kiírni „A kedvenc gyümölcsöd:” szöveget, a változó tartalmát és az „Én is nagyon szeretem!” szöveget, ezért egy speciális jelet kell használnunk ezek összefűzéséhez. Ez nem más, mint az & jel. Vagyis írjuk a következőt:

"A kedvenc gyümölcsöd: " & gyümölcs & ". Én is nagyon szeretem!"

Futtatáskor a képen látható eredményt kapjuk, amennyiben a banán szót gépeljük be.



► A megjelenő szöveg a konzolablakban

Feladat

Pármunkában készítsünk egy chatbotot! A chatbot olyan alkalmazás, amellyel chatelni (beszélgetni) lehet. A chatbot adjon szöveges visszajelzést arra, amit írtunk (például „Aha, értem. Miben tudok segíteni?”). Az algoritmus legalább öt kérdés-válasz lehetőséget biztosítson! Próbáljunk olyan általános visszajelzéseket adni, amelyekkel hihető lehet, hogy a chatbot arra válaszol, amit beírtunk!

Gondolkodjunk fordítva is! Előre tudjuk, hogy mit fog válaszolni a chatbotunk, hiszen mi programoztuk be. Írjuk be olyan mondatokat, amelyekkel vicces párbeszéd születhetnek! Próbáljuk ki egymás chatbotjait úgy, hogy ne nézzük meg előtte az algoritmust!



Ciklusok és elágazások

Fej vagy írás? Használjunk ciklust és elágazást!

Pénzfeldobással egyszerűen eldönthetünk kérdéseket úgy, hogy a véletlenre bízunk magunkat. Az eredmény kétfajta lehet: fej vagy írás. A következőkben olyan algoritmust készítünk, amely a pénzfeldobást szimulálja.

A *Flowgorithm* alkalmazásban vannak beépített függvények, amelyeket a kifejezésekben használhatunk.

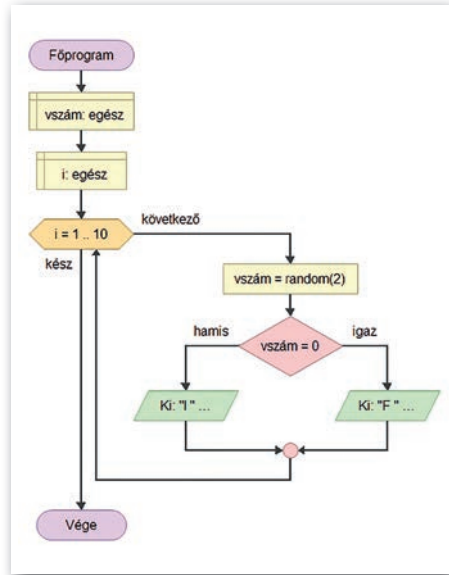
Véletlenszám generálására a `random(n)` függvény alkalmas, amely 0 és $n-1$ között ad vissza véletlenszámot. Vagyis a `random(3)` a 0, 1, 2 számokat adja eredményül, véletlenszerűen.

Készítsük el az ábrán látható algoritmust! Ez tíz alkalommal írja ki véletlenszerűen az **F** és az **I** betűt, egy sorban, szóközzel elválasztva.

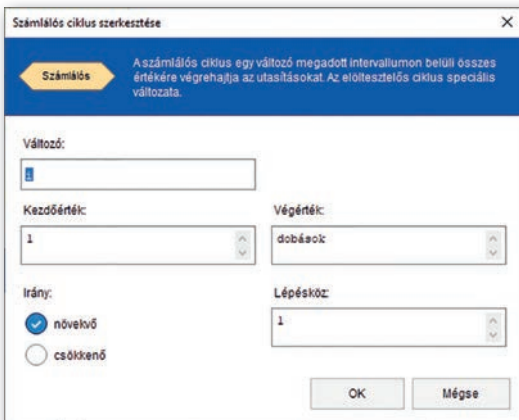
Magyarázat: Mivel tudjuk, hogy tíz alkalommal kell feldobni a „pénzérmét”, számlálós ciklust érdemes használnunk.

A `vszám` változót a véletlenszám tárolására használjuk, az `i`-t pedig a számlálós ciklus ciklusváltozójának. Mindkét változót deklarálnunk kell használat előtt, mégpedig egész típust megadva.

A számlálós ciklus szerkesztésekor meg kell adnunk a ciklusváltozó nevét (`i`), kezdő értékét (1) és végértékét (10). Ezenkívül megadhatjuk még a lépésközt, vagyis azt, hogy hányasával növekedjen a ciklusváltozó. Ez most 1 lesz. A számolás iránya szintén beállítható, ami lehet növekvő, illetve csökkenő.

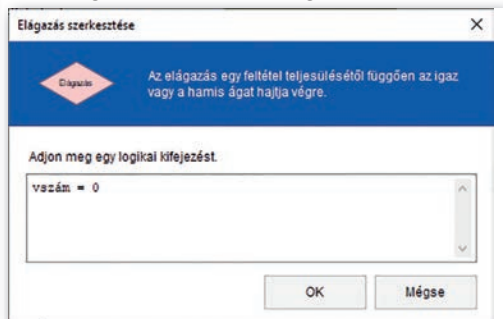


► A fej vagy írás dobás algoritmus





► A számlálós ciklus beállítása

► Az elágazás feltételének megadása



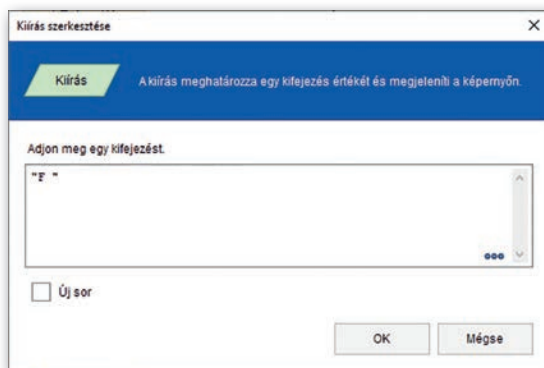
Az elágazásnál egy logikai feltételt kell megadnunk. Most azt vizsgáljuk, hogy a kapott véletlenszám nullával egyenlő-e. Ehhez a `vszám = 0` feltételt kell beírni.

Amennyiben a feltétel teljesül (igaz), akkor az F betűt írjuk ki, ha nem, akkor az I betűt.

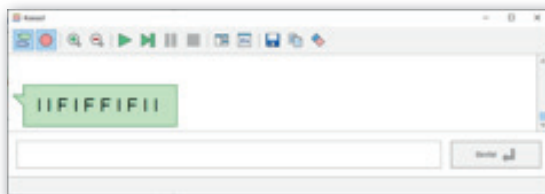
Fontos még, hogy most egy sorba, szóközzel elválasztva szeretnénk kiírni a kapott eredményt, ezért az F és I kiírásánál ne válasszuk ki az Új sor jelölőnégyzetet! Ekkor a kifejezés jobb alsó sarkában a  ikon jelenik meg, nem pedig az  ikon.

Ne felejtjük el beírni az F, illetve I betű kiírásánál a betű utáni szóközt sem!

Ezzel készen is vagyunk. Futtassuk az algoritmust! Eredményül a képen látható kimenethez hasonlókat kapunk a konzolablakban.




► Kiírás sortörés nélkül

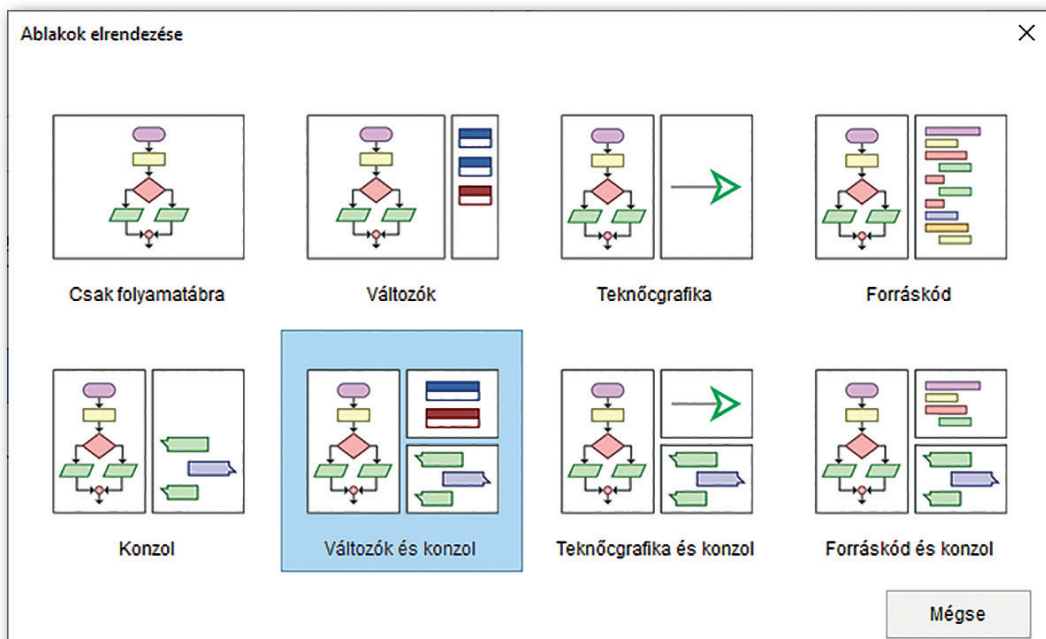


► Egy futtatás eredménye

Az algoritmus lépésenkénti futtatása

Ahhoz, hogy jobban megértsük, mi történik az algoritmus végrehajtásakor, érdemes a *lépésenkénti végrehajtást* választani, és közben ellenőrizni a változók értékét.

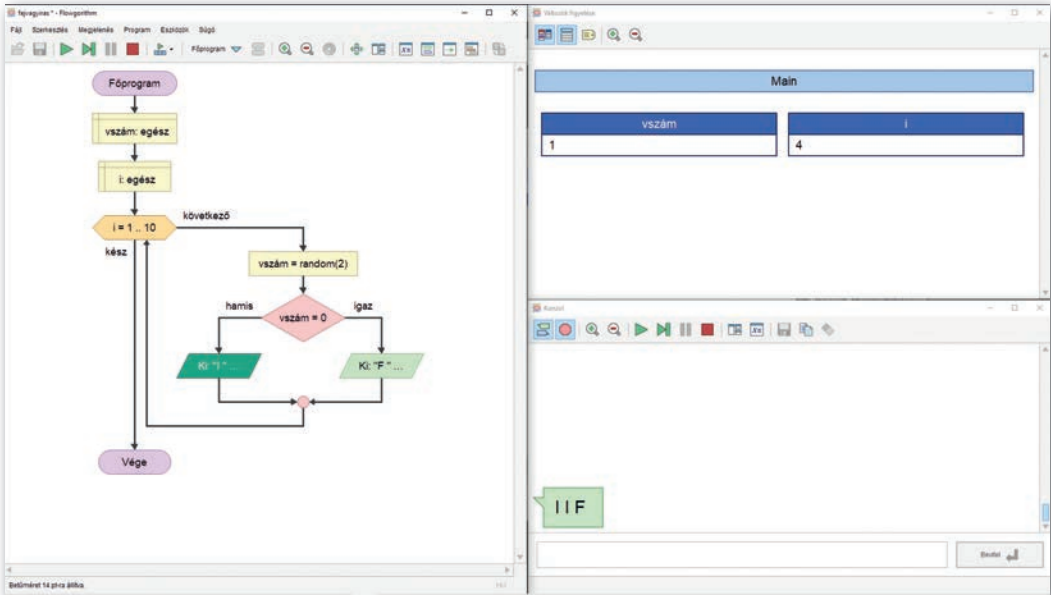
Ahhoz, hogy jól lássunk minden adatot, válasszuk ki a megfelelő ablakelrendezést! Ehhez kattintsunk az *Ablakok elrendezése* ( ikonra, majd válasszuk ki a *Változók és konzol* elrendezést!



► „Változók és konzol” elrendezés

Ezek után elkezdhetjük a lépésenkénti futtatást. Kattintsunk a léptetés ikonra (▶)! Ezzel az algoritmust lépésenként (vagyis blokkról blokkra) tudjuk végrehajtani. A program kiemeli a következő végrehajtandó blokkot, közben pedig láthatjuk a változók értékeit is.

Az alábbi képen azt láthatjuk, hogy az i változó értéke 4, vagyis a negyedik betű kiírásánál tartunk. A $vszám$ változó értéke 1, vagyis az elágazás hamis ága fog végrehajtódni. Sötétzöld színnel a KI: "I" blokk van megjelölve, vagyis ez hajtódik végre a léptetés gomb következő megnyomásakor. A lépésenkénti végrehajtást a leállítás (■) ikonnal állíthatjuk meg. Lépésenkénti végrehajtásból bármikor átválthatunk a normál végrehajtásba a futtatás (▶) ikon megnyomásával.



▶ Az algoritmus lépésenkénti végrehajtása

Típusalgoritmusok használata

Számoljunk! Használjunk típusalgoritmust!

Amikor a fej vagy írás játékról van szó, általában arra gondolunk, hogy a kétféle kimenet közel azonos számban fordul elő. De vajon valóban így van ez? Biztosan nem forulhat elő, hogy tízszer egymás után fejet dobunk? Győződjünk meg róla! Számoljuk meg, hányszor dobunk fejet, illetve írást!

Az, hogy meg kell számolnunk adott tulajdonságú elemeket, nagyon gyakran fordul elő a különböző problémák megoldása során. Azt is mondhatnánk, hogy ez egy úgynevezett **típusalgoritmus**, mivel adott típusú problémára nyújt megoldást. A típusalgoritmusokat más néven **programozási tételeknek** is nevezzük. (Későbbi tanulmányaink során még több ilyenrel fogunk találkozni.)

Nézzük meg először, hogy mi a típusalgoritmus a **megszámlálásnak**! Ha összesen N darab elemünk van, amelyek közül meg akarjuk számolni, hogy hány adott tulajdonságú elem van, akkor az algoritmus a következő lesz:

```
Eljárás Megszámlálás
  DB:=0
  Ciklus I=1-től N-ig
    Ha az I. elem adott tulajdonságú akkor DB:=DB+1
  Ciklus vége
Eljárás vége
```

Az algoritmus lényege, hogy egy cikluson belül megvizsgáljuk az összes elemet. Ha adott tulajdonságú elemet találunk, akkor megnöveljük a darabszám (DB) értékét eggyel. Természetesen a ciklus előtt gondoskodnunk kell a darabszám változó lenullázásáról is. Ha ezt nem tennénk, akkor hibás eredményt kaphatnánk a megszámlálás során.

Példa: Öttel osztható számok 1 és 20 között

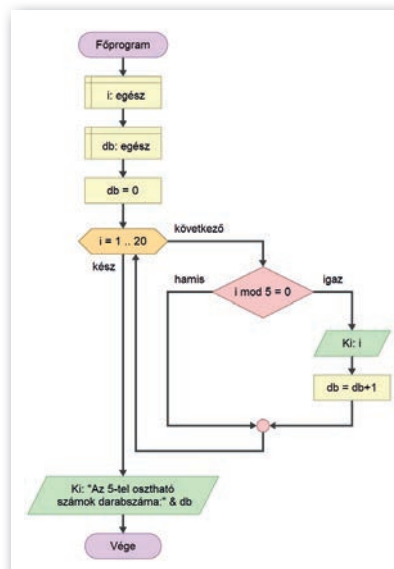
Nézzünk egy példát a típusalgoritmus használatára! Számoljuk meg, hogy 1 és 20 között hány olyan szám van, amely osztható öttel, és ezeket írjuk ki a képernyőre!

Tudjuk, hogy egy szám akkor osztható öttel, ha az öttel való osztás után maradékul nullát kapunk.

Az osztási maradék művelete a *Flowgorithm* programban (és sok más programozási nyelvben is) a *mod* vagy *%* művelet. Vagyis a $10 \bmod 5$ művelet eredménye 0, mert a 10 osztható 5-tel. A $10 \bmod 3$ művelet eredménye pedig 1, mert a 10-ben a 3 háromszor van meg, a maradék pedig 1. A $10 \bmod 3$ műveletet így is írhatjuk: $10 \% 3$.

Próbáljuk ki a képen látható algoritmust a gyakorlatban is!

- ▶ Öttel osztható számok kiírásának algoritmus



Feladat

A megszámlálás típusalgoritmus alapján módosítsuk a korábbi (fej vagy írás) algoritmust úgy, hogy megjelenjen a fej dobások száma is! Használjuk az itt látható mintát a kiírásra!

FFFFFIFIF A fej dobások száma: 8

► Egy lehetséges kimenet

Legyen áttekinthető a kód! Csoportosítsuk a kódrészleteket!

Érdekes lenne tudni, hogyan alakul a fej dobások száma, ha sokszor (például százszor) megismételjük a pénzfeldobás-sorozatot. Lesz-e esetleg olyan, hogy az összes érmedobás fej lesz?

Ehhez a korábban elkészített, teljes algoritmusunkat egy olyan számlálós ciklusban kellene elhelyeznünk, amely százszor hajtódik végre. Ezzel viszont már eléggé hosszú, kevésbé átlátható algoritmust kapnánk. Mi lehet erre a jobb megoldás?

Ha egy algoritmusrészletet többször szeretnénk végrehajtani, érdemes azt a részletet egy önálló blokkban elhelyeznünk. Ezt a blokkot pedig el is nevezhetjük. Ez hasonló ahhoz, mintha különböző építőköc-káink lennének, amelyekből aztán összeállíthatjuk a kész építményt, a teljes algoritmust (vagy programot). Egy fajta építőkockát akár többször is felhasználhatunk.

A különböző programozási nyelvek eltérő lehetőségeket biztosítanak (*eljárások és/vagy függvények*) a kódrészletek létrehozására. A *Flowgorithm* alkalmazásban (és sok más programozási nyelvben) **függvényeket** használhatunk erre a célra. A következő leckében ezekkel ismerkedünk meg.



Függvények használata

Mire jók a függvények?

Beépített függvényt már korábban is használtunk. Ilyen volt a `random()` függvény. A függvénynek meg kellett adnunk az n egész **paramétert** is a kerek zárójelen belül. A megadott paraméter határozza meg azt, hogy a függvény belsejében milyen adatokkal dolgozunk. A `random()` függvény esetén a paraméter azt jelenti, hogy mely számnál kisebb véletlenszámot szeretnénk előállítani. Vannak azonban olyan függvények is, amelyeknek több paraméterük van. Ekkor a kerek zárójelben több adatot kell felsorolni.

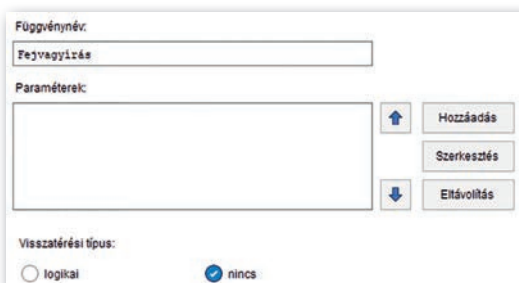
A függvénnyel az a célunk, hogy elvégezzon egy műveletsorozatot, vagy hogy kiszámoljon valamit. A kiszámolt értéket fogja eredményül adni a függvény hívásakor. Ezt az értéket **viszterési értéknek** nevezzük. Ezt az értéket kapjuk eredményül a függvény hívásakor. Például a `random(3)` függvény visszterési értéke olyan egész szám, amely 0, 1 vagy 2 lehet.

Nagyon hasznos, hogy nemcsak beépített függvényt használhatunk, hanem mi is létrehozhatunk függvényt. A függvény nevét mi adhatjuk meg, de a programozási nyelvben már foglalt nevet nem választhatunk.

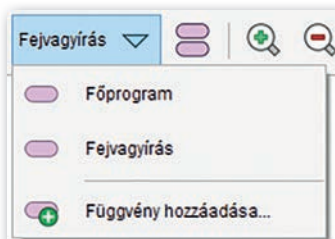
Példa: A Fejvagyírás nevű függvény létrehozása

Hozzuk létre a `Fejvagyírás` nevű függvényt a következő lépésekkel!

1. Kattintsunk a függvénykezelő ikonra az eszköztáron!
2. A megjelenő ablakban kattintsunk a *Létrehozás* gombra!
3. A függvény neveként adjuk meg a `Fejvagyírás` szöveget! Visszterési típust most nem kell beállítanunk, vagyis a *nincs* nevű választási lehetőség legyen kijelölve. Később majd megnézzük, hogy milyen előnyei lehetnek a visszterési értéknek.
4. Kattintsunk az *OK* gombra, majd a megjelenő ablakban a *Kész* gombra!
5. Ezek után az eszköztáron már átválthatunk a *Főprogram*-ra, illetve visszaválthatunk a `Fejvagyírás` nevű függvényre is.
6. A következőkben a *Főprogram*ból fogjuk a blokkokat átmozgatni a `Fejvagyírás` függvénybe. Válasszuk ki a *Főprogram*ot!
7. Egérrel kerítsük körbe az összes blokkot! Ennek hatására a kijelölt blokkok kék színűek lesznek. Amennyiben minden blokkot sikerült kijelölnünk, kattintsunk a jobb egérgombbal, és válasszuk ki a *Kivágás* műveletet! Ennek hatására a *Főprogram*ból eltűnnek a kijelölt blokkok.



► Függvény létrehozása



► Váltás a *Főprogram* és a függvények között

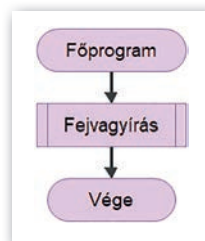


► A korábban kimásolt kód beillesztése

- Váltunk át a *Fejvagyírás* nevű függvényre! Kattintsunk a *Fejvagyírás* nevű blokkból kiinduló nyílra, majd a megjelenő ablakban válasszuk ki a *Vágólap* kategóriában a *Beillesztés* opciót! Ezzel minden korábban kivágott blokk ebbe a függvénybe kerül.

Példa: A *Fejvagyírás* nevű függvény meghívása

- Váltunk vissza a *Főprogram*ra! Kattintsunk a *Főprogram* nevű blokkból kiinduló nyílra, és a megjelenő ablakban válasszuk ki a *Hívás* utasítást! Ezt az utasítások kategória *Vezérlés* csoportjában találjuk.
- Kattintsunk a *Hívás* blokkra, és adjuk meg a meghívni kívánt (vagyis végrehajtandó) függvény nevét: *Fejvagyírás*. Ha mindent jól csináltunk, a *Hívás* blokkban megjelenik a függvény neve.
- Hajtsuk végre az algoritmust! Eredményül ugyanazt kell kapnunk, mint korábban, vagyis egymás után megjelennek a fej és írás dobások, illetve a fej dobások száma.



► A módosult *Főprogram*

Feladatok

- Módosítsuk önállóan a főprogramot úgy, hogy a *Fejvagyírás* függvény 100 alkalommal hajtódjon végre, vagyis egymás alatt 100 alkalommal jelenjenek meg az érmedobások eredményei! Ezután futtassuk le az algoritmust!
- Vizsgáljuk meg az eredményt! Volt-e olyan eset, amikor 10 alkalommal is fejet dobtunk? Ha nem, akkor mi volt a legnagyobb szám? Beszéljük meg, hogy kinek a programja dobta a legtöbb fejet az osztályban!
- Hallottunk-e már a cinkelt érméről? Ezt az érmét szándékosan úgy módosították, hogy sokkal többször lehessen vele fejet dobni, mint írást (vagy fordítva). Hogyan lehetne módosítani az algoritmust úgy, hogy sokkal többször legyen fej az eredmény? Beszéljük meg közösen a lehetséges megoldásokat!



► Részlet az eredményből. Látszik, hogy volt olyan eset, amikor 9 fej dobás volt


Jó tudni!

- A legnagyobb érték meghatározására szintén létezik típusalgoritmus, amelyet *maximum-kiválasztásnak* neveznek. Ezzel (és több más) algoritmussal későbbi tanulmányaink során fogunk megismerkedni. De érdemes már most elgondolkodni a problémán! Vajon milyen algoritmussal lehetne meghatározni a legnagyobb számot sok (például 100) szám közül?
- Bizonyos programozási nyelveken nemcsak függvényt, hanem *eljárást* is lehet készíteni. Az eljárás nagyon hasonló ahhoz, ahogy a függvényt készítjük el (saját névvel rendelkezik, és paraméterei is lehetnek), azonban a függvénytől eltérően az eljárásnak nincs visszatérési értéke. A későbbiekben találkozhatunk olyan programozási nyelvekkel, amelyekben használhatunk ilyeneket.
- A *Flowgorithm* programmal az algoritmust át lehet alakítani különböző programozási nyelvekre. Ehhez a *Forráskód megtekintése* (📄) ikont kell megnyomni, majd a legördülő menüt használni. Sőt, mondatszerű leírássá is átalakíthatjuk a folyamatábrát, amihez a *Magyar pszeudokód* nyelvet kell kiválasztani. Próbáljuk ki!

Példa: A Fejvagyírás függvény paraméterezése

Korábban láttuk, hogy a beépített függvénynek **paraméterben** adjuk át a szükséges adatokat. Ugyanezt a saját függvények esetén is megtehetjük.

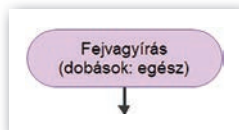
Most a `Fejvagyírás` függvényünk 10 alkalommal dob véletlenszerűen fejet vagy írást. De a 10-es szám lehetne akár a függvény paramétere is, a paraméter megváltoztatásával pedig a dobások száma is változna. Folytassuk ennek megvalósításával!

1. Nyissuk meg a függvénykezelőt () , és kattintsunk a `Fejvagyírás` függvényre!
2. Kattintsunk a `Szerkesztés` gombra!
3. A megjelenő ablakban a `Paraméterek` csoportban kattintsunk a `Hozzáadás` gombra!
4. A `Paraméternév` mezőben adjunk nevet a paraméternek! A név legyen: `dobások`. A típusa legyen egész!



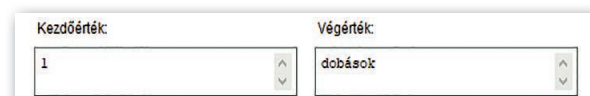
► Paraméternév megadása

5. Kattintsunk az `OK` gombra (két ablakban is kell), majd a `Kész` gombra!
6. Ezek után már látszik a paraméter neve a függvénynél.



► Függvény neve a paraméterrel

7. Most válasszuk ki a `Fejvagyírás` függvényt, és módosítsuk a `számlálós ciklust` úgy, hogy a végértéke ne 10 legyen, hanem a `dobások` változó értéke.



► Ciklus végértékének beállítása

8. Térjünk vissza a főprogramhoz! Módosítsuk a függvényhívást úgy, hogy a paramétert lehessen megadni! Vagyis írjuk például ezt: `Fejvagyírás (15)`
9. Futtatáskor azt tapasztaljuk, hogy most már 15 alkalommal történik érmedobás.



Függvény használata visszatérési értékkel

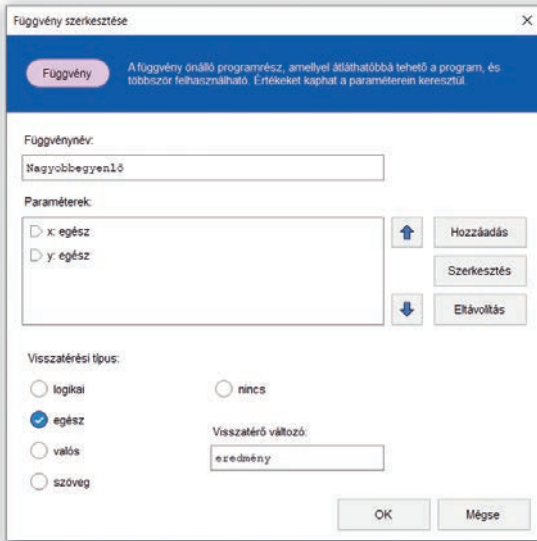
Most nézzük meg azt, hogy hogyan állíthatunk be a függvénynek visszatérési értéket! Ezt a tudásunkat több blokkprogramozási nyelvben felhasználhatjuk, például a `micro:bitek` programozásánál is.

Készítsünk egy `Nagyobbegyenlő` nevű függvényt, amely két egész paraméterrel rendelkezik! Ha a paraméterként kapott két szám különböző, a függvény a nagyobbát adja vissza, ha viszont egyformák a paraméterek, a függvény ezt a számot adja vissza. Vagyis a függvény a táblázatban láthatók szerint működjön.

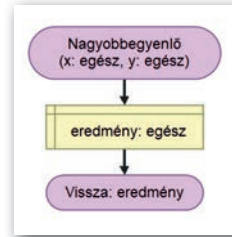
Függvény	Eredmény
<code>Nagyobbegyenlő(2, 8)</code>	8
<code>Nagyobbegyenlő(-1, -6)</code>	-1
<code>Nagyobbegyenlő(5, 5)</code>	5

A létrehozás lépései:

1. Hozzuk létre a `Nagyobbegyenlő` függvényt, és adjunk hozzá két paramétert! Az x és y paraméter lesz a két számunk, amelyek legyenek *egész* típusúak!
2. A függvény visszatérési értékének típusa legyen *egész*! Meg kell adnunk a visszatérő változó nevét is, amely legyen: `eredmény`.
3. Az *OK*, majd a *Kész* gombok megnyomása után megjelenik a függvény folyamatábrája. Láthatjuk, hogy a függvény a *Vége* blokk helyett a *Vissza* blokkal zárul. Ez utal arra, hogy a függvénynek van visszatérési értéke.

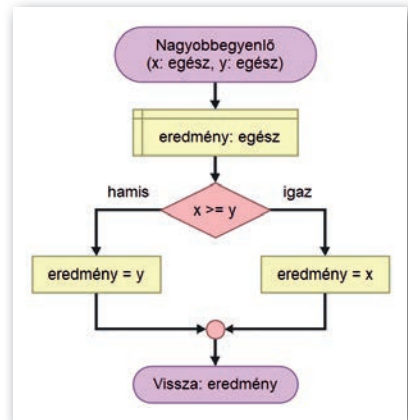


► A függvény szerkesztési ablaka



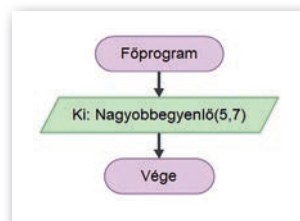
► A létrejött függvény

4. Módosítsuk úgy a függvényt, hogy az `eredmény` változó az x és y számok közül a nagyobb legyen egyenlő! Ha az x és az y megegyezik, akkor az x szám legyen az eredmény! Ehhez használjunk elágazást!



► Egy lehetséges megoldás a függvényre

- Készítsük el a *Főprogramot* is! Írassuk ki a Nagyobb-egyenlő függvény eredményét!
- Teszteljük az algoritmust különböző paraméterek megadásával!



► A *Főprogram*

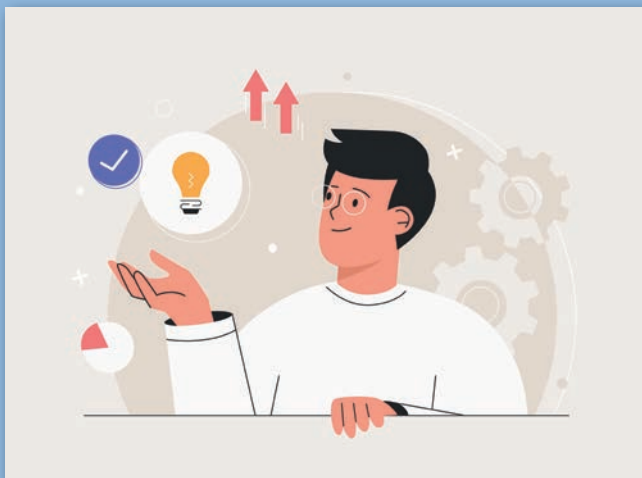
Kérdés

Az elágazásnál most az $x \geq y$ feltételt helyeztük el. Mi történne, ha az $x > y$ feltételt írnánk be? Változna-e az eredmény? Miért?

Jó tudni!

A fenti algoritmus végrehajtása során három eset állhat elő. Vagy az első szám a nagyobb, vagy a második, vagy a két szám egyenlő. A tesztelésnél mindig törekednünk kell arra, hogy a lehetséges eseteket megfelelő paraméterekkel kipróbáljuk.

Ha hibát találunk, használjuk a lépésenkénti végrehajtást, amely segít abban, hogy a változók értékét figyelemmel kísérjük, illetve lássuk, hogy milyen utasítások hajtódnak végre, és hányszor.



Feladatok

- Próbáljuk ki, hogy mi történik akkor, ha elhelyezünk az algoritmusban egy vagy több töréspontot! Milyen hatása van a töréspontnak a futtatáskor, illetve a lépésenkénti végrehajtáskor?
- Lássuk el az algoritmust magyarázó megjegyzésekkel! Ezek a magyarázatok segíthetnek nekünk abban, hogy később is megértsük, mit miért csináltunk. Később látni fogjuk, hogy megjegyzéseket a programok forráskódjában is elhelyezhetünk.
- Pármunkában készítsünk algoritmusokat a következő feladatok megoldására!
 - Készítsünk egy sormintát, amely legfeljebb 10 karakterből (betű, szám, egyéb karakterek) áll. Például: >-----<. Kérjük be a felhasználótól, hogy hányszor szeretné megjeleníteni egymás mellett a sormintát, majd jelenítsük meg azt!
 - Fejlesszük tovább az algoritmust úgy, hogy több sorból álló mintát is ki lehessen rajzolni, és a felhasználó adhassa meg a sorok számát! Az egy sorban megjelenő sorminta kiírását paraméterezett függvény segítségével készítsük el!



Töréspont

Megjegyzés

Korábbi tanulmányainkban a programok készítésekor számos műveletet használtunk (például `és`, `vagy`, `nem`). A *Flowgorithm* programban is használhatók műveletek. A következő feladatok megoldása során ezekre szükség lehet. Összetettebb műveletek esetén a kerek zárójelekkel csoportosíthatjuk a műveleteket.

	Művelet	Példa
egyenlőség vizsgálata	<code>=</code> vagy <code>==</code>	<code>x = 10</code> <code>x == 10</code>
nem egyenlőség vizsgálata	<code>!=</code> vagy <code><></code>	<code>x != 10</code> <code>x <> 10</code>
és művelet	<code>&&</code> vagy <code>and</code>	<code>x > 5 && x < 10</code> <code>x > 5 and x < 10</code>
vagy művelet	<code> </code> vagy <code>or</code>	<code>x = 5 x = 10</code> <code>x = 5 or x = 10</code>
tagadás művelet	<code>!</code> vagy <code>not</code>	<code>!(x < 10)</code>
összefűzés művelet	<code>&</code>	"Az életkorom: " & kor
maradék számítása	<code>%</code> vagy <code>mod</code>	<code>x % 5</code> <code>x mod 5</code>

4. Jelenítsük meg a naptár egy hónapját a következő formátumban: a napok 1-től 31-ig jelenjenek meg, és a sorszámok után legyen kiírva a *nap*: szöveg is. A héttel osztható számok esetén a *Szabadnap* szöveg jelenjen meg, a többinél a *Munkanap*. Vajon hogyan lehetne megoldani azt, hogy a héttel osztható szám előtti napnál is a *Szabadnap* szöveg jelenjen meg? Készítsük el ezt a változatot is!



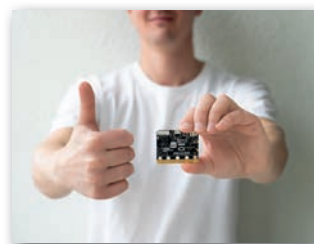
5. Készítsük el az `Ellentett` függvényt! A függvénynek egy paramétere legyen, amely egy egész szám! A függvény adja vissza a szám ellentettjét, vagyis a `-1`-gyel való szorzatát! Teszteljük le az eredményt pozitív, negatív és nulla paraméterre is!

Programozzuk micro:bitet!

Az előzőekben megismertünk egy olyan eszközt, amellyel algoritmusokat készíthetünk, kipróbálhatjuk őket, és akár lépésenként is nyomon követhetjük, hogy mi történik a végrehajtásuk során.

Ez a tudás a későbbiekben is hasznos lesz, amikor új algoritmusokat kell kitalálnunk. Az is előfordulhat, hogy mások által írt algoritmust kell megértenünk, alkalmaznunk, és tesztelnünk a helyes működését.

Használtunk függvényt, amellyel csoportosíthattuk a többször felhasználandó kódot, illetve paramétereket adtunk a függvénynek. A következőkben visszatérünk a micro:bitek programozásához, melynek során kihasználjuk a függvényekben rejlő lehetőségeket is.



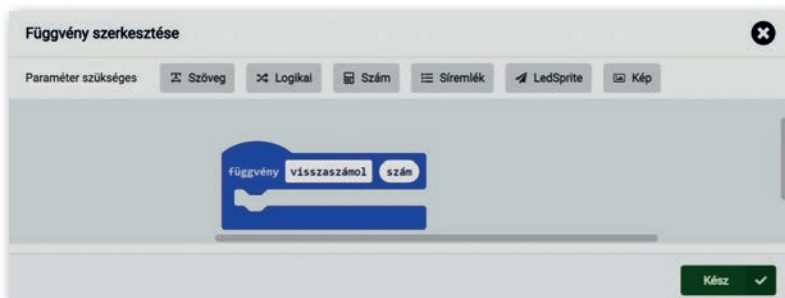
Használjunk függvényeket!

A játékprogramok gyakran visszaszámlálással indulnak, hogy a játékos felkészülhessen a kezdésre. Készítsünk olyan függvényt, amely elvégzi a visszaszámlálást! A függvény paramétere legyen az az egész szám, amelytől indítani szeretnénk a visszaszámlálást egészen nulláig!

Készítsünk egy új projektet a <https://makecode.microbit.org/> felületen! A függvényeket a *Haladó funkciók* között találjuk, ezért kattintsunk a **Haladó** legördülő menüre!

A *f(x) Függvények* kategóriában kattintsunk a **Függvény létrehozása...** gombra!

A *Függvény szerkesztése* ablakban adjunk nevet a függvénynek (*Visszaszámol*), majd kattintsunk a *Paraméter hozzáadása* sorban a *Szám* paraméterre! Ha készen vagyunk, nyomjuk meg a *Kész* gombot!



► A függvény beállításai

A Visszaszámol függvény tartalmának megadása

Ezzel létrejött a függvény! Helyezzük el a visszaszámláláshoz szükséges utasításokat! Például használhatunk számlálós ciklust erre a célra. A paraméterként megjelenő **szám** változót úgy tudjuk elhelyezni ciklusváltozóként, hogy az egérrel megragadjuk, és a megfelelő helyre vonszoljuk.

Ügyeljünk arra, hogy vissza kell számolnunk, ezért a kiírásnál nem a ciklusváltozót kell kiíratnunk, hanem megfelelő matematikai művelettel még elő kell állítanunk a kiírandó számot. Korábbi tanulmányainkban volt már erre példa.

Most már csak gondoskodnunk kell a függvény meghívásáról, és tesztelhetjük is az eredményt.

- ▶ A visszaszámlálás egy lehetséges megvalósítása



- ▶ A függvény meghívása

Feladatok

1. Egy szöveg a micro:bit kijelzőjén jobbról balra gördítve jelenik meg. Pármunkában készítsünk olyan függvényt, amely a szöveget betűnként jeleníti meg, gördítés nélkül! A függvény neve legyen: **Betűzd!** Minden betű kiírása után teljen el egytized másodperc, majd a kijelzőt letörölve, az újabb betű megjelenése előtt teljen el újabb egytized másodperc! A szöveg feldolgozásával kapcsolatos blokkokat a **Szöveg** kategóriában találjuk. Gondoljuk át, mely blokkok szükségesek a feladat megoldásához! A függvényt természetesen úgy kell elkészítenünk, hogy tetszőleges hosszúságú szöveget képes legyen betűzni.
2. Módosítsuk a fenti függvényt úgy, hogy ne csak a szöveg legyen paraméterként átadva, hanem az időzítési érték is, amely az előző feladatban egytized másodperc volt!

Függvény visszatérési értékének beállítása

Az imént létrehoztunk függvényeket, de azoknak nem volt visszatérési értéke. Készítsünk egy paraméterezhetőt is, hiszen a későbbiekben azt sokféle célra felhasználhatjuk.

A **MakeCode** blokkprogramozási felület **Math** blokkjában találunk egy beépített függvényt (**max** ezek közül: **0** és **0**), amely két különböző szám közül a nagyobbat adja vissza. Egyenlő számok esetén az értéke a megadott szám lesz. Készítsünk olyan függvényt, amely hasonlóan működik, de nem két szám közül adja meg a legnagyobbat, hanem négy szám közül!

Hozzunk létre **maximum** néven egy függvényt, amelynek legyen négy darab szám paramétere! Ha négy számból akarjuk meghatározni a legnagyobbat, a következőképpen tehetjük meg. Először határozzuk meg, hogy az első két szám közül melyik a nagyobb! Hasonlítsuk össze ezt a számot a harmadik számmal, és a nagyobbat tároljuk el egy változóban! Végül ezt a számot hasonlítsuk össze a negyedik számmal, és ismét tároljuk el a nagyobbat!

Ahhoz, hogy a függvény visszaadja a legnagyobb értéket, a **Függvények** kategóriából a **Visszaad** blokkot kell elhelyeznünk a függvény végén. Látható is, hogy a **Visszaad** bloknak olyan a kialakítása, hogy alá már nem helyezhetünk el másik blokkot.

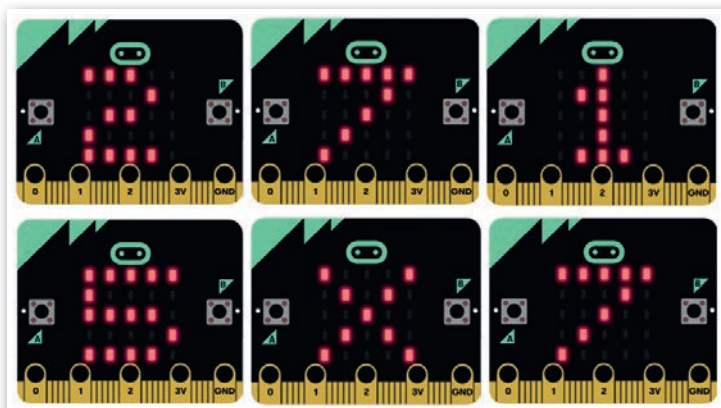
Ezek után próbáljuk ki a függvény működését! Írassuk ki a legnagyobb számot! A **Kiírás** blokkban lekerekített blokkot tudunk csak elhelyezni. Mivel olyan függvényt készítettünk, amelynek van visszatérési értéke, a **Függvények** kategóriában megjelenik a függvényhívás olyan változata is, amely egy lekerekített blokkban helyezkedik el. Használjuk ezt a szám kiírásánál!



- ▶ A feladat egy lehetséges megoldása

Feladatok

1. Korábban elkészítettük a fej vagy írás dobás algoritmusát. Valósítsuk meg ugyanezt a micro:biten is! A függvény neve legyen: `Fej vagy írás`. A függvénynek egy paramétere legyen, amely jelentse azt, hogy hányszor dobunk egymás után! Fej és írás esetén más-más ikonok jelenjenek meg a kijelzőn! Számoljuk meg a függvényben azt is, hogy hányszor dobtunk fejet! A fejek száma legyen a függvény visszatérési értéke! A micro:bit megrázásával hívjuk meg 10-es paraméterrel a függvényt, majd utána jelenjen meg a fej dobások száma!
2. Készítsünk függvényt `Kockadobás` néven! A függvény paramétere jelentse azt, hogy hány oldalú dobókockával dobunk. Egy hagyományos, hatoldalú kocka esetén a paraméter értéke 6. Ha a paraméter 12, akkor már 1 és 12 közötti értékeket kaphatunk. A dobott szám legyen a függvény visszatérési értéke!
Végezzük el négyszer a kockadobást úgy, hogy jelenjenek meg a micro:bit kijelzőjén a dobott számok! Ezt követően töröljük le a kijelzőt, majd jelenítsük meg az X ikont! Ezután írjuk ki, hogy melyik szám volt a legnagyobb a négy dobás közül! Használjuk fel a korábban készített `Maximum` függvényt!



► A program egy lehetséges eredménye

Készítsünk játékokat micro:bitre!

Az alábbiakban ötleteket találunk arra, hogy milyen játékok, alkalmazások készíthetők önállóan, párban vagy akár csoportmunkában a micro:bit eszközre. Törekedjünk arra, hogy a kód legyen átlátható, ahol lehet, csoportosítsuk a kódot kisebb egységekbe a függvények segítségével! Adjunk megjegyzéseket a kódhoz, hogy társaink is megértsék a gondolatmenetünket! Ezt a jobb gomb megnyomása után, a *Mejggyezés hozzáadása* menüponttal tehetjük meg.

Milyen az időérzékünk?

1. Pármunkában készítsünk olyan programot, amely azt méri, hogy milyen jó az időérzékünk! A játék az *A* gomb megnyomásával induljon úgy, hogy jelenjen meg a kijelzőn egy véletlenszám 5 és 9 között, majd 1 másodperc múlva jelenjen meg egy ikon, amely a játék kezdetét jelzi. A játék lényege, hogy a *B* gombot pontosan annyi másodperc után kell megnyomni, amekkora számot láttunk korábban! A gomb megnyomása után jelenítsük meg a kijelzőn azt, hogy hány ezredmásodperc különbséggel nyomtuk meg a gombot az előírt értékhez képest! Természetesen minél kisebb számot látunk, annál ügyesebbek voltunk. Azt, hogy a program mennyi ideje fut, a *Bemenet* kategória `futási idő (ms)` blokkjával kérhetjük le.
2. Fejlesszük tovább a programot úgy, hogy ne csak egy szám jelenjen meg a játék végén, hanem egy ikon is, amely utal az eredményre! Legyen három kategória: nagyon pontos, pontos és átlagos. Például akkor nagyon pontos az eredmény, ha a tényleges idő és a lenyomás között kevesebb mint fél másodperc telt el.
3. Készítsük el a játék azon változatát, amikor két micro:bit egymásnak küldi az adatokat! Az első micro:biten lehessen a játékot elindítani és leállítani, de az eredmény a másik micro:biten jelenjen meg!



Vajon tudunk-e blöffölni?

A játék közbeni blöffölés azt jelenti, hogy félrevezetjük a társainkat azért, hogy ne essünk ki a játékból, vagy hogy jobb eredményt érjünk el.

A más nevű játék eredeti változatát dobókockákkal játsszák. A játék előnye, hogy akár 5-6 fős (vagy nagyobb) csoportokban is játszhatjuk. A játékban minden játékos két dobókockával dob úgy, hogy társai ne láthassák a dobott számokat, vagyis a dobások eredményét el kell takarni. A játék célja, hogy társaink elhiggyék azt, hogy a két dobókockával olyan számot dobtunk, amely nagyobb vagy egyenlő, mint amit az előzőleg dobó társunk dobott.

A két dobott számból speciális szabállyal határozzuk meg a dobás értékét. A nagyobb dobás lesz a kétjegyű számunk első számjegye, a kisebb szám pedig az utolsó számjegye.



Ha két egyenlő számot dobunk, akkor természetesen a két számjegy megegyezik. Ezt nevezik básnak. Az 1-es bás a 11, a 2-es bás a 22, és így tovább.

Vagyis a dobásértékek a következőképpen alakulnak:

Egyik kockadobás	Másik kockadobás	A dobás értéke
1	3	31
4	1	41
3	3	33

► A két kockadobás eredménye és az abból előállt dobási érték

Van azonban egy nehezítés is, amely miatt nagyon kell figyelni a játék során. A dupla dobások nagyobb értékkel bírnak, mint a különböző számjegyekből állók. De ezeknél is többet ér a 21-es dobás, amely a játékban a legtöbbet éri. Vagyis a dobásértékek sorrendje aszerint, hogy melyik érték ér többet az előzőnél:

31, 32, 41, 42, 43, 51, 52, 53, 54, 61, 62, 63, 64, 65, 11, 22, 33, 44, 55, 66, 21

Kezdetben minden játékosnak 5 pontja (vagy zsetonja) van. A játékosok körben ülnek egy asztal mellett vagy a földön. Minden körben más játékos kezdi a játékot az óramutató járásával megegyezően vagy ellenkezően (amelyik jobban tetszik). A játékot kezdő dobó elmondja a dobásának az értékét. A soron következő játékos ezt vagy elhiszi, vagy nem. Ha elhiszi, akkor folytatódik a játék: dob, és elmondja, mennyit dobott, vagy blöfföl, és egy másik számot mond. Ha viszont nem hitt az előző játékosnak, akkor annak fel kell fednie a dobott kockáit. Ekkor kiderül, hogy igazat mondott-e, vagy blöffölt.

Ha valaki rosszul blöffölt, az veszít egy pontot, aki pedig hazugságon érte társát, az 1 ponttal gazdagodik.

Akkor is veszít a játékos 1 pontot, ha olyan számértéket mondott, amely nem fordulhat elő a játékszabály szerint (5, 9, 40, 13, 24, 46 stb.). Ekkor a dobás sorrendjében következő játékos automatikusan kap 1 pontot.

Az új fordulót a következő játékos kezdi. Akinek elfogynak a pontjai, kiesik a játékból. A játék addig tart, míg egy játékos marad, de játszhatunk úgy is, hogy ha valaki elér egy bizonyos pontszámot (például 10), akkor vége a játéknak.

Nézzünk egy példajátékot! Az első játékos 3-at és 4-et dob, ezért a dobás értéke 43. Nem blöfföl, az igazat mondja a társainak. A második játékos 1-et és 3-at dob. Dobásának értéke 31, ami sajnos kevesebbet ér, mint az előző játékosé. Ezért kénytelen blöffölni, és azt mondja, hogy 52-t dobott. A következő játékos ezt nem hiszi el, és kéri, hogy mutassa meg a dobott kockákat. Mivel a játékos hazudott a pontértékről, elveszít 1 pontot, a másik játékos pedig kap 1 pontot.



A bás játék elkészítése

Feladatunk az előző társasjáték algoritmusának elkészítése a *Flowgorithm* alkalmazásban, majd a megvalósítása micro:bit segítségével.

A kockadobás a micro:bit megrázásával történjen! A dobás értékét úgy jelenítsük meg a kijelzőn, hogy egyszerűen leolvasható legyen a két szám! Például a nagyobb dobást megjeleníthetjük a kijelző bal oldalán, a másikat pedig a jobb oldalán. Így a dobásérték egyszerűen leolvashatóvá válik. A kockák képeinek megjelenítéséhez használjunk függvényeket, a megjelenítendő számot paraméterként adjuk át!

Nyerés esetén a *B* gomb megnyomásával lehessen növelni a pontszámot, veszteség esetén pedig az *A* gombbal csökkenteni. Ha a pontszám eléri a nullát, jelenjen meg egy animáció, illetve halljunk egy hangjelzést, amely arra utal, hogy kiesett a játékos a további fordulókból! Az *A + B* gombok megnyomása után jelenjen meg a pontszám, majd kis idő elteltével az utolsó dobás képe. Így akár játék közben is ellenőrizni lehet a pontszámot.

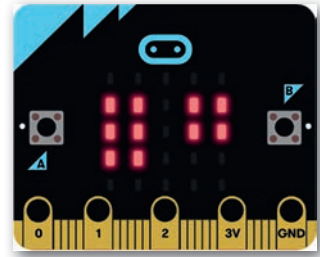
Továbbfejlesztés, ötletelés

3-4 fős csoportokban ötleteljünk arról, hogyan lehetne növelni a játékélményt! Gondolkozhatunk abban is, hogy a micro:bit-ek rádiókapcsolattal kommunikálhatnak egymással vagy egy játékvezetői micro:bittel. Színesíthetjük a játékot különböző hanghatásokkal is.

Valósítsuk meg ötleteinket! Próbáljuk ki egymás megoldásait, és játszunk néhány játszmat a legjobban sikerült programokkal!

Ezek után akár osztálykirándulásra is magunkkal vihetjük a micro:bit-eket, és a szabadidőben is játszhatunk velük.

Ha maradt még időnk, fejlesszünk más társasjátékokat is a korábban tanultak felhasználásával!



▶ A két dobás egyidejű megjelenítése (64-es pontérték)



Összefoglalás

Az általános iskola felső tagozatán sok izgalmas lehetőséggel találkoztunk az algoritmizálás, programozás és robotika területén.

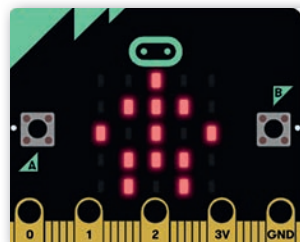
Olyan blokkprogramozási környezeteket használtunk, amelyekben játékosan készíthettük el programjainkat. Ehhez algoritmusokat gyártottunk. Az algoritmusokat mondat-szerűen vagy folyamatábrák segítségével írtuk le, sőt azt is megtanultuk, hogyan tesztelhetjük az algoritmusok működését, és milyen lehetőségeink vannak a hibák azonosítására és kijavítására. Megismerkedtünk olyan programozási fogalmakkal, amelyekre a későbbiekben építeni tudunk.



Valós és nem létező (szimulált) robotokat irányítottunk, és megtanítottuk őket különböző feladatok elvégzésére, például vonal mentén történő haladásra. Áttekintettük és kipróbáltuk, hogy a robotok milyen állapotokkal rendelkeznek, illetve hogyan képesek érzékelni a környezetüket.

A programozás során megismerkedtünk számos utasítással, függvénnyel és vezérlési szerkezettel, használtunk ciklusokat, elágazásokat, hogy minél többfajta feladat megoldására képesek legyünk.

Megismerkedtünk a micro:bit vezérlővel, amely játékosá tette a tanulást. Nemcsak az előírt feladatokat készítettük el, hanem pármunkában és csoportmunkában is megvalósíthattuk saját ötleteinket, amelyeket aztán megoszthattunk a többiekkel.



A környezetünkből származó adatokat (például fényerősséget) mértünk és dolgoztunk fel úgy, hogy rádiókapcsolatot is használtunk az eszközök között az egyes feladatok elkülönítésére.

Animációkat és ezekre épülő játékokat fejlesztettünk úgy, hogy a szereplőket különböző gesztusokkal vagy gombnyomással lehetett irányítani. Rajzoltunk a kijelzőre, és valós, életbeli problémákat szimuláltunk.

Megnéztük, hogyan tudjuk hatékonyabbá tenni a programjainkat, illetve milyen lehetőség van a kód újrafelhasználására és átláthatóbbá tételére.

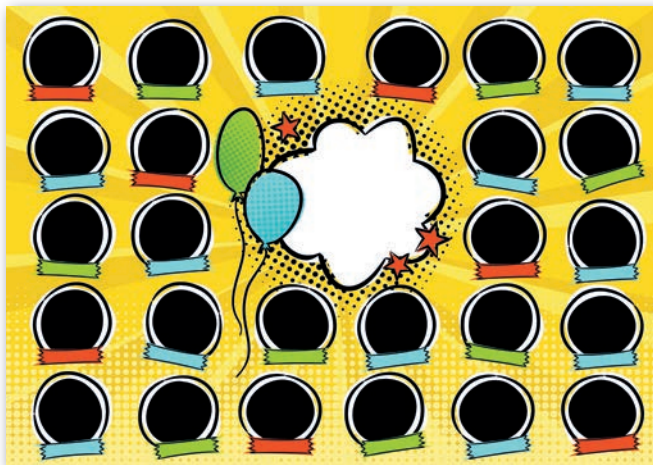
Későbbi tanulmányainkban már nem (feltétlenül) blokkprogramozási környezetet fogunk használni, hanem például betekintést kapunk a modern, objektumorientált programozásba is. De azt, amit eddig megtanultunk, a későbbiekben is tudjuk használni. Addig is, ha találkozunk egy problémával, amelyet meg tudunk oldani az általunk ismert környezetekben, álljunk neki bátran! Ezzel nemcsak a számítógépes gondolkodásunk fejlődik, hanem hasznos alkalmazásokat hozunk létre, amelyek gyorsabbá, hatékonyabbá, esetleg élvezetesebbé teszik a munkát. Ehhez kívánunk sok sikert a továbbiakban!



Ballagási tablót készítünk

Ebben a tanévben befejeződik az általános iskolai éveink. Remek alkalom arra, hogy emléket állítsunk saját magunknak, annak az osztálynak, amelyben gyermekkorunk jelentős részét töltöttük. Számítógépes ismereteink nagyjából elegendőek ahhoz, hogy a tablót mi magunk tervezzük meg, és elkészítsük azt az alapanyagot, amelyből egy nyomda már előállíthatja a végterméket.

A képfeldolgozással és alakzatok készítésével a korábbi években már megismerkedtünk. Ebben a fejezetben egy összetett példán keresztül áttekintjük ezeket az ismereteket. A pixelgrafika és a vektorgrafika összehasonlításáról *A digitális eszközök használata* című fejezetben olvashatunk részletesebben.



A tabló motívuma

A tabló azért lesz több, mint egymás mellé tett fényképek összessége, mert egyes részei egységesek. Ez az egység nemcsak a szemnek kellemes, de kifejezi a tabló szereplőinek összetartozását is. A hagyományos tablókon a diákok ruhája és tartása éppúgy egységes, mint a képek alakja és a nevek felírására használt betűtípus. Némileg oldottabb hangulatú tabló készíthető, ha ezek közül valamelyiktől eltekintünk, de ha mindről lemondunk, akkor ismét csak egymás mellé helyezett képeink lesznek, és nem tablónk.

Mielőtt hozzákezdenénk...

Ha a közelben nem volna nagy képek nyomtatását vállaló szolgáltatás – hiszen a tabló lényegében egyetlen nagy kép lesz –, az interneten aligha lehet probléma ilyet találni.

A teljes tabló összeállítását – csakúgy, mint a képketrek megrajzolását, a tabló címének és a diákok neveinek felírását – célszerűen vektorgrafikus eszközökkel végezzük. Céljainknak megfelel akár egy bemutatókészítő alkalmazás vektorgrafikus eszköz-készlete is. Mielőtt képzeletünket szabadjára engednénk, egyeztessünk a kiválasztott nyomdával, hogy milyen formátumban fogadják be a nyomtatnivalót (a PDF általában használható), és mekkora tabló nyomtatását vállalják. Ha a PowerPoint maximális diaméterét túl kicsinek találjuk, a LibreOffice megfelelő választás lehet.

A tabló képeinek elkészítése

Ha magunk készítjük a tablóképeket, érdemes az alábbiakra figyelni.

Mielőtt elkészítenénk a képeket, tudnunk kell, hogy milyen lesz a képek kerete – pontosabban azt, hogy milyen méretarányú képekre van szükség. Négyzet alakúakra? A nyomdászatban elterjedten használt 4:5 arányúra? A telefonok képernyőjén megszokotthoz hasonló 16:9-esre (amely, minthogy valószínűleg álló képet készítünk, inkább 9:16-osnak mondanó)? Mielőtt nekilátunk az osztály minden tagjáról fotót készíteni, próbáljuk ki, hogy mikor nem fog kilógni (vagy mikor fog mindenkinél nagyjából egyformán kilógni) a már méretek alakított képből a váll, a fejtető és a kéz.

A képeknek legyen egységes, semleges háttér, ugyanis nem szeretnénk, ha elterelné a figyelmet a szereplőről. A fényképezőgépeket be lehet állítani úgy, hogy a háttér elmosódjon, de manapság a legtöbb telefon is képes hasonló hatás elérésre. Természetesen a háttér elmosható a rastergrafikus rajzolóprogramunkban is, illetve ingyenes webes szolgáltatások is rendelkezésre állnak ilyen célra.

A jó tablóképekhez sok fényre lehet szükségünk. Alapszabály, hogy ha csak szemből érkezik a fény, akkor nem lehet jó portrékat készíteni, mert az arc térbeliségét a szemből érkező fény tompítja, meghamisítja. Portrékészítésre a fényképezőgépbe, telefonba épített vaku emiatt nem felel meg. Tovább bonyolítja a feladatot, hogy az arcra érkező fény nem lehet túl éles, mert akkor túl éles lesz a szemek és az orr körüli árnyék. Mindkét problémát egyszerűen megoldjuk, ha a fényképezésre egy világos, de felhős napon, amikor az éles napsugarak szóródhatnak, például egy ablak mellett, az ablaknak valamennyire oldalt fordulva kerítünk sort. Természetes színű, szórt fényt kapunk, ha az ablakból ragyogó napsütéses napon érkező erős fényt lepedőn vagy terítőn szűrjük át. Az emberi bőr színeit napfényben vagy ahhoz hasonló színű mesterséges fényben készült képeken találjuk a legszebbnek.

A fényképünk nem fog bemozdulni, ha állványra tesszük a telefonunkat. Hasznos trükk, ha két-három másodpercre időzített felvételt készítünk – ilyenkor a telefont nem mozdítja meg a fotós keze sem.

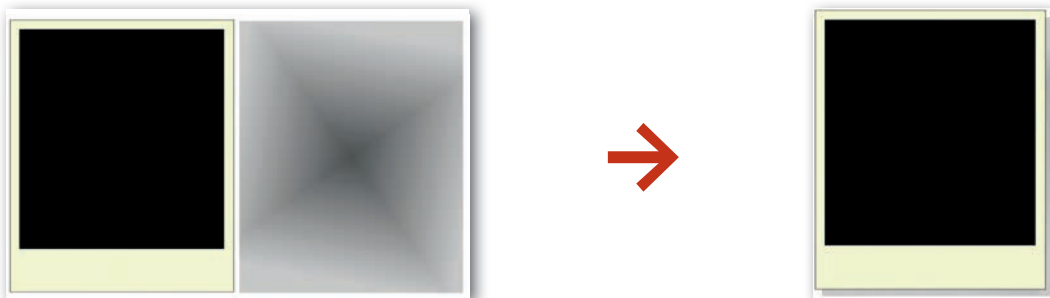
Rastergrafikus programban végezzük el a képek alapvető retusálását, például a pattanások vagy anyajegyek eltávolítását, esetleg a szemek világosítását (GIMP-et használva az eszköztár *Javítás*, *Másoló*, *Elmosás*, *Élesítés*, *Fakítás* vagy *Sötétítés* eszközei is megfelelőek a feladat elvégzésére).



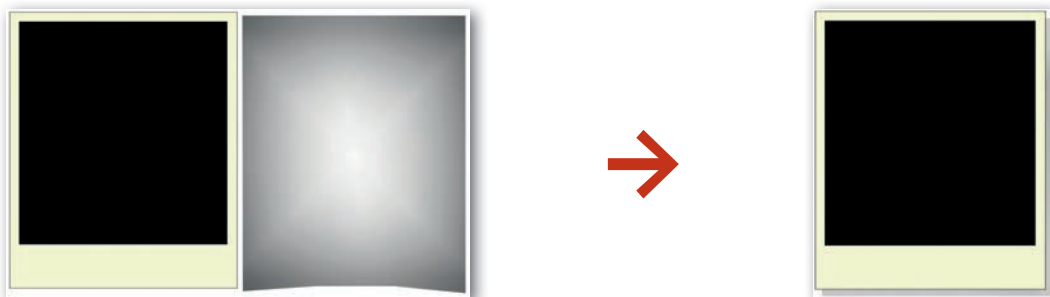
A képek keretei

A keretek kreatív megválasztása is segít érdekessé tenni a tablónkat. Mielőtt a keretek elkészítésének nekifognánk, tudnunk kell, hogy milyen színű, esetleg milyen mintájú lesz a tábló háttere, hiszen nem szerencsés, ha utólag kell minden képkeretet átszíneznünk, mert a színük nem harmonizál a háttérrel. Az is hasznos, ha van elképzelésünk arról, hogy hova kerülnek a képek feliratai – a nevek. Az alábbiakban bemutatott három keret egyikén magára a keretre kerül a név. Próbáljuk ki, hogy elfér-e a leghosszabb nevű osztálytársunk neve is, mégpedig akkora méretben, hogy távolabbról is olvasható legyen.

Képeinket elhelyezhetjük úgy, mintha valódi fényképeket ragasztgattunk volna a táblóra. A képkeret ilyenkor legegyszerűbb esetben egy egyszerű téglalap, amelyet a fénykép mögé helyezünk el. A „felragasztott” hatást erősíti, ha a képnek enyhe vetett árnyékot adunk. Ha a bemutatókészítő vetett árnyékait túl mesterségesnek találjuk, mi magunk is készíthetünk „árnyékot” a keret alá helyezett másik téglalappal. Ezt a téglalapot a bemutatókészítő eszközeivel valamilyen egyenetlen mintával vagy színátmenettel töltjük ki, esetleg alkalmazunk átlátszóságot! Kísérletezzünk bátran az eszközökkel!



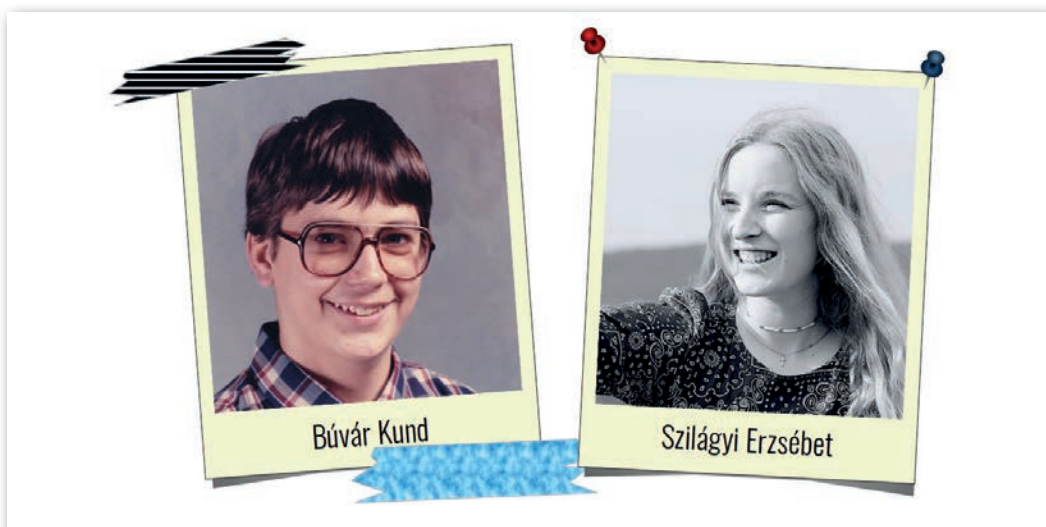
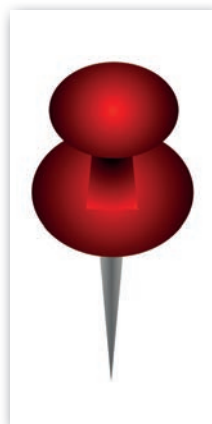
Felkunkorodó képszél hatását érhetjük el, ha az „árnyékot” sokszögmintára rajzoljuk meg, amely kilóg a „fényképpapír” alól. Ha nem bemutatókészítőben dolgozunk, hanem vektorgrafikus szerkesztőben, akkor utólag is van lehetőség a csomópontok áthelyezésére, új csomópontok felvételére, illetve a csomópontok közötti vonal ívelésére. Így valószerűbbé tehető a keretünk felkunkorodása. A bemutatókészítőben további lehetőségként kínálkozik még a görbék használata is.



Amikor elkészültünk a képkerettel, próbáljuk bele a fényképet is. Ha alapvetően elégedettek vagyunk, foglaljuk csoportba a kész részt.

Készíthetünk rajzszőgeket is a képünk sarkaiba. Egy rajzszőg két ellipsziszből, egy közöttük lévő trapézból és a hegyként szolgáló háromszögből áll. Az alakzatokat színátmenettel töltjük ki, hogy hamis térbeliséget kölcsönözzünk nekik. A kész rajzszőg elemeit foglaljuk csoportba, és döntsük meg, így helyezzük el a kép egyik sarkánál. Ha a rajzszőg másolatát a függőleges tengelyre tükrözzük, precízen előállítható a másik sarokba való szög, de lehet, hogy valóságosabbnak hat, ha a két rajzszőg nem pontos tükröképe egymásnak. A másolatot akár át is színezhettük.

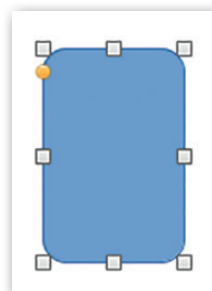
Ha pedig elég már a rajzszőgekből, rajzolhatunk mindenféle ragasztócsíkot, sebtapaszt mintával, vagy valamilyen egyszerű, sormintaként használható képpel kitöltött sokszögeket.



Bemutatjuk még egy keret elkészítését: ezúttal telefonon tekinthetjük meg magunkat – úgy, ahogy a valóságban a legtöbbször amúgy is szoktuk. Rajzoljunk lekerekített sarkú téglalapot, majd távolítsuk el az alakzatot szegélyező vonalat, a kitöltést pedig állítsuk sötét színűre. A téglalap mérete legyen $5,2 \times 10,6$ cm. A téglalap valamelyik sarkánál láthatunk egy kört, ezzel tudjuk a görbület erősségét szabályozni – állítsuk olyanra, amilyenek a telefonunk sarkát látjuk.

Készítsünk másolatot a téglalapról, és mindkét kiterjedését csökkentjük 4-4 milliméterrel. A kitöltést állítsuk világosra, és az új téglalapot igazítsuk a régihez képest függőlegesen és vízszintesen is középre.

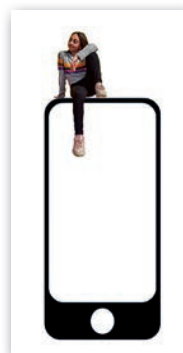
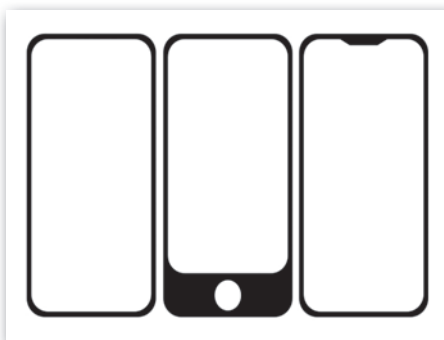
Ha van kedvünk, a világos téglalap magasságának csökkentésével alakíthatunk ki helyet a „gomboknak”. A belső téglalap tetejéhez igazított trapézalappal pedig a szembenéző kamerának is helyet adó „notch” valósítható meg.



A telefon „képernyőjén” már el tudjuk helyezni a képünket.

Érdekesebbé tehetőek ezek a képek, ha valakit „ráültetünk” a telefonra. Először keressünk olyan képet, amelyen a szereplő valamilyen jól határolt vízszintes vagy függőleges terepen ül, fekszik, vagy annak támaszkodik, esetleg próbálja eltolni. A példán ilyen a fal, amelyen a lány ül. Rastergrafikus alkalmazásban jelöljük ki a szereplőt, és a kép kijelölésre vágásával távolítsuk el a fölös részeket. Adjunk alfa-csatornát a képhez, hogy ha valamit kiradírozunk, átlátszóság maradjon alatta. Nekünk kézre álló módszerrel távolítsuk el a további szükségtelen részeket is.

Másoljuk át a képet a bemutatókészítőbe, illetve a vektorgrafikus alkalmazásba, vagy exportáljuk a képet PNG-formátumban – így nem vész el az átlátszóság –, majd szűrjük be az exportált fájl tartalmát.



Természetesen nem kell mindenkinek a saját ballagási képét mutató telefon környékén lennie. „Henyélhetünk” a barátunk, padtársunk telefonján, és „odébbtolhatjuk” a barátnőnk telefonját. Különösen érdekesek lehetnek a többszereplős képek, de elkészítésükkor figyelniük kell arra, hogy minden telefonnak jusson hely.

Vegyük észre, hogy akkor tudjuk elvégezni ezeket az egymáshoz illesztéseket, ha a képen, amelyből kiindulunk, pontosan a fotóssal szemben húzódik az a függőleges vagy vízszintes határolóvonal, amelyiknek az ottlétét kihasználjuk. A képet hatásosabbá teszi, ha a határvonalat valami átlépi – ilyen a képen látható lány jobb lába és a fiú bal keze.

Járható út az is, ha a telefonok belsejébe kisgyerekkori képeket helyezünk el. Ezek lehetnek még papírképek, amelyeket erre szolgáló berendezéssel, a szkennerral, magyar nevén lapolvasóval tudunk digitális jelek sorozatává alakítani. A szkennerbe helyezett fényképet pedig erre szolgáló alkalmazással tudjuk rastergrafikus formátumban menteni. A beolvasás során a szkennel végső soron lefényképezi a fényképet, de lényegesen szebb képet ad, mint ha a mobiltelefonunkat használtuk volna e célra.

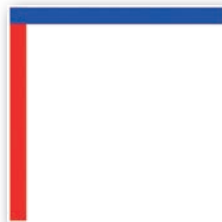


A tábló befejezése

Ha mindenki elkészült a saját képével, keretével, feliratával, és gondosan csoportba is foglalta az objektumokat, akkor érdemes a munkát úgy folytatni, hogy a képeket egyetlen gépen egyetlen dokumentumba másoljuk, majd egymás mellé és alá rendezzük őket. Beállítjuk a tábló hátterét, és kiírjuk a tábló címeként szolgáló feliratot.

Rajzolhatunk – akár csak jelzésként, a négy sarokban látszódo – keretet a táblónkra. Legegyszerűbb esetben egy vastag vonalból vagy egy vékony téglalapról indulunk ki. Kettőzzük a téglalapot, és forgassuk el, majd a sarkokat illesszük egymáshoz (az ábránkon csak azért más színű a két téglalap, hogy jobban láthassuk a keret felépülését). Foglaljuk csoportba a két téglalapot, majd kettőzzük a csoportot, és tükrözzük a másolatot! Két keretsarok elkészült, és egy újabb kettőzéssel, majd a másolat tükrözésével megvan mind a négy szükséges keretrészletünk. Természetesen kiindulhatunk bonyolultabb ábrából, vagy akár kézzel rajzolt és beszkenelt részletből.

A kész művet exportáljuk a nyomdával megbeszélte formátumba, és ha elkészült a nyomtatás, keressünk olyan helyet a művünknek, ahol csodájára járhat mindenki!



Hangfelvétel készítése és egyszerűbb utómunkálatai

Hangfelvétel készítéséhez mikrofonra van szükségünk, amely manapság a legtöbb ember zsebében ott lapul, hiszen a telefonunkba, egyéb mobilkészülékünkbe építve bizonyosan van ilyen eszköz. Van mikrofon a mai laptopokban is, ha pedig asztali számítógépen készítünk hangfelvételt, tudunk hozzá külső mikrofont csatlakoztatni.

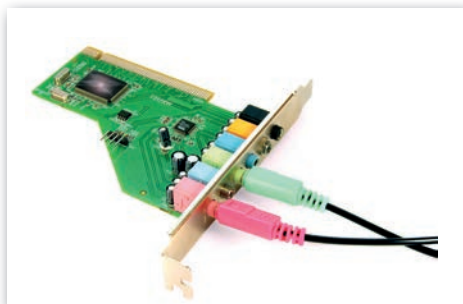
Kérdések

1. Hogyan terjed a levegőben a hang?
2. Milyen közeg továbbíthatja még a hangot?
3. Hogyan érzékeli a hangot a fülünk?
4. Mennyiben hasonlít a mikrofon és a fülünk?
5. Milyen jelátalakítást végez az emberi fül, illetve a mikrofon?
6. Hova továbbítja az átalakított jelet a fül és a mikrofon?
7. Milyen eseményeknél és eszközökben használunk mikrofont?

A hang feldolgozása

A légnemzés változását követve, ahhoz hasonlóan – azzal analóg módon – változik a mikrofonból elvezethető, belőle érkező áram. A mikrofon analóg jelét a számítógépeken nem is olyan régen egy külön eszköz, a hangkártya alakította digitális jellé. A „hangkártya” ma is benne van a számítógépekben, mobilkészülökben, de felkészült szakembernek kell lennünk, ha meg akarjuk találni a sok egyéb kicsi alkatrész között.

A hangkártyának, illetve a szerepét ellátó chipnek természetesen a hang visszaalakítása is feladata. A számítógépen lejátszott hangfájlokban digitálisan tárolt információt kell analóg jelként, esetünkben feszültségingadozásként továbbítani az eszköz kivezetésén a fülhallgató, illetve hangszóró felé. Professzionális megoldásoknál az analóg-digitális és a digitális-analóg átalakítást külön eszköz végzi.



► Hangkártya

Feladat

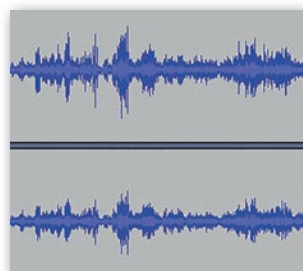
Keressünk és nézzünk meg a mobiltelefonban lévő hangchipet ábrázoló képet, videót!

Alapvető műveletek hangszerkesztő alkalmazással

Az eszközünkbe érkező, immár digitális jelet valamilyen alkalmazás dolgozza fel. Talán a leghétköznapiabb eset, amikor a megérkező digitális jelet telefonálás során továbbítjuk beszélgetőpartnerünk készülékére. Van azonban, amikor a hangot rögzíteni, későbbi felhasználásra szánva tárolni szeretnénk. Telefonunkkal vagy a számítógépünkkel alighanem kapunk előtelepítve valamilyen, alapvető feladatokat ellátni képes hangrögzítő alkalmazást. Ha pedig sokoldalúbb, ügyesebb eszközre van szükségünk, remek választás lehet a nyílt forráskódú *Audacity*, amelyet a továbbiakban használni fogunk.

Ha az Audacityben megnyitunk valamilyen hangfájlt, például egy zenesámot tartalmazó, MP3-formátumú állományt, akkor a sztereó felvétel jobb és bal csatornáján szóló hanghullámot mutató ábrát kapunk. Ezen tudunk dolgozni: itt van lehetőségünk a hanghullám egyes részeit kijelölni, átalakítani.

Játsszuk le a hangot, és ha sikerült, vegyük észre, hogy a lejátszást kétféleképp állíthatjuk meg: a szünet és a leállítás gombbal. A hanghullám szerkesztése csak leállított lejátszás esetén lehetséges, a szünet nem elegendő.



► Hanghullámok ábrázolása Audacityben

Feladatok

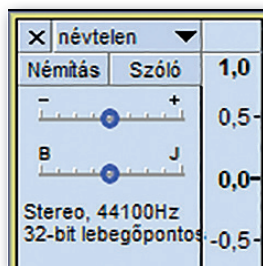
1. Figyeljük meg, hogy a jobb és a bal csatorna hulláma más alakú ott, ahol a jobb és a bal oldalon lejátszott hang is eltérő! Keressük meg az eszköztáron az alapesetben aktív *kijelölés* eszközt! Jelöljük ki a hullám egy részét, és figyeljük meg, hogy ekkor csak ez a rész játszódik le. Keressük meg, miként szüntethető meg a kijelölés!
2. A *nagyítóval* közelítsünk rá annyira a hullámra, hogy a képernyőn csak egyetlen jellegzetes zenerészlet – egy dobütés, gitárpendítés, zongorahang – hulláma legyen látható! Hogyan tudunk ismét „távolabbra menni”, nagyobb részletet megnézni a zeneszámból?
3. Jelöljünk ki részleteket a hanghullámból, és nézzünk szét a *Hatások* menüben! Próbáljunk ki néhány hatást! A fel-, illetve a lekeverést a felvétel vagy felvételrészlet elején és végén szokás használni. Az erősítés, a megfordítás és a visszhang bárhol használható. Mi a különbség a *tempó változtatása* és a *sebesség változtatása* hatás között?

Hangsávok és csatornák, a munkánk mentése

A sztereó hangsávnak két csatornája van, a jobb és a bal oldalon szóló. Ha a *Sávok* menüben új sávot kérünk, látható, hogy monó hangsávot is készíthetünk, ennek egyetlen csatornája lesz. A monó felvételek menthetők sztereó hangsávban is, ilyenkor a jobb és a bal csatorna azonos információt tárol.

Ha egy sztereó hangsáv csatornáit külön-külön szerkeszteniénk, az ábránkon látható lefelé mutató, fekete nyílra kattintva válasszuk a *Sztereósáv felosztása* lehetőséget.

A szerkesztés befejeztével, félbehagyásával a művünk mentésére több lehetőségünk is van. Ha *projektként* mentünk, akkor megmaradnak a sávjaink, a beállításaink, és legközelebb ott folytathatjuk a munkát, ahol félbehagytuk – de csak az Audacityben hallgathatjuk meg, ami eddig elkészült. Ha a fájlt *exportáljuk*, egyéb alkalmazások is képesek lesznek megnyitni és lejátszani. A WAV-formátumú fájlok az adatot tömörítetlenül tárolják, az ilyen fájljaink igen méretesek lesznek. A FLAC-formátum veszteségmentes, az MP3- és az OGG-formátum veszteséges tömörítést valósít meg.



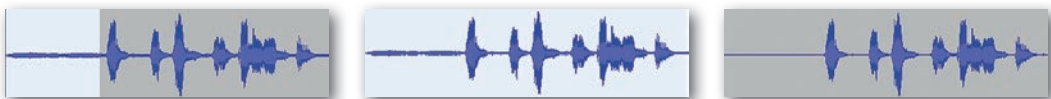
Kérdések

Mit jelent a 2.1-es, az 5.1-es és az 5.2-es hangsáv? Hol és miért használnak ilyet? Hány csatorna van bennük?

Hangfelvétel

Vegyük fel a saját hangunkat! Ha van a számítógépünkhöz csatlakoztatott mikrofon, akkor egyszerűbb azt használnunk. Teljesen megfelel az is, ha a mobil eszközünkre telepítünk alkalmazást, és a kész fájlt nyitjuk meg a hangszerkesztőben. Érdekes monó hangszávvá dolgozunk, hiszen egyszerűbb a szerkesztése, és jó eséllyel úgyis monó a mikrofonunk is.

Hangfelvétel készítésekor jól jöhet a *normalizálás* hatás. Olyankor lesz hasznunkra, ha több, eltérő hangerejű részletből állítunk össze egyetlen hangszávvot, és azt szeretnénk, hogy egyik se legyen sokkal hangosabb, mint a többi. A *zajcsökkentés* hatás arra való, ha valamilyen alapzajt szeretnénk eltüntetni a felvételtől. A használata két lépésben történik. Az első lépésben (ahogy az első képen látható) kijelölünk egy olyan részt a felvételen, ahol csak zaj van, majd a hatásnál a *zajprofil készítése* lehetőséget választjuk. Így tanítjuk meg az Audacityt arra, hogy mi számít zajnak. A második lépésben (második kép) kijelöljük azt a részt, amelyből a zajt kell eltávolítani, és újra használjuk a hatást. A harmadik képen látható, hogy az alapzaj hulláma eltűnik.



Feladatok

1. Vegyük fel, ahogy elszámolunk tízig, majd oldjuk meg, hogy a páros számokat a bal, a páratlanokat a jobb fülünkben halljuk! A nagyobb számokat hatások használatával alakítsuk egyre mélyebb hangúvá!
2. Készítsünk sztereó párbeszédet két monó felvételtől! Használhatjuk például Rómeó és Júlia erkélyjelenetének szövegét, amelyben Rómeó beszél a jobb oldalon, Júlia a balon.
3. A tanulócsoportban válasszunk ki tíz ismert zeneszámot! Alkossunk párokat! A csapat mindkét tagja állítson össze egy-egy, a tíz zeneszám mindegyikének részletét tartalmazó, legfeljebb félperces fájlt, amelyet majd a csapat másik tagja hallgat meg. A másik csapat tag jegyezze le, hogy milyen sorrendben hangzottak el az egyes számok részletei! Az a csapat nyer, amelyik összesítve a legtöbb találatot éri el a húszból.

Egyszerűbb videószerkesztési műveletek

Videóformátumok, codecek, videófájlok, streaming

Napjainkban már legtöbbször zsebünkben ott lapul (legalább) egy kamera, ezért igazán hasznos megismerkednünk a felvételek utólagos szerkesztésének, vágásának egyszerűbb lehetőségeivel.

Bevezetesként és a tananyag kiegészítéseként megemlítünk néhány olyan témát, amelyek megkönnyítik a videószerkesztők használatának megértését, és mindennapi műsorfogyasztóként is hasznukat vesszük.

A videófájljaink az esetek túlnyomó többségében hangot is tárolnak. A hang- és a videóinformáció a fájlokban jól elkülönül, azaz beszélhetünk külön hangsávokról és képsávokról. Hangsávból egy fájlban gyakran van több, mert például a tárolt film eredeti nyelve mellett több szinkronsáv is helyet kapott az állományban. Sokszor kerülnek feliratsávok a fájlba, így akár több nyelven is megjeleníthetjük a film szereplőinek szövegét.



Napjainkban a legelterjedtebb a Full HD felbontás, amelyben a képek 1920 képpont szélesek és 1080 képpont magasak. Ezt a felbontást gyakran 1080p-ként vagy 1080i-ként említik. (A p betű arra utal, hogy minden képkocka tartalmazza a kép összes képpontját, az i pedig arra, hogy az egyik képkockán csak a kép páros, a következőn pedig csak a páratlan sorok képpontjai láthatók.) Kisebbségi felbontás például a HD, másképp 720p vagy 720i formátum, nagyobb pedig a 4K, más jelöléssel a 2160p vagy 2160i felbontás, amely mindkét dimenzióban kétszer annyi képpont terjedelmű, mint az elsőként említett Full HD. Mindhárom felbontás 16:9-es képarányú. A régebbi, 4:3 képarányú műsorok készítése századunk első évtizedében szűnt meg, azóta lényegében a 16:9-es képarány az egyeduralgoló.

Fontos jellemzője a videósávoknak, hogy másodpercenként hány képkockát tárolnak. Az elterjedtebb ipari szabványok másodpercenként 24, 25 és 30 képkockát írnak elő. Ha például 60 képkockát veszünk fel másodpercenként, a videófájlunk lényegesen nagyobb lesz – cserébe viszont jó lassításokat tudunk készíteni.

A videófájljaink kiterjesztése e könyv készítésének idején (2022-ben) az esetek túlnyomó többségében .mp4, .mkv vagy .webm. Ezek egyszerűen úgynevezett konténerformátumok is, és a videólejátszó alkalmazásunknak megnyitásukkor elmondják, hogy milyen eljárással vannak bennük kódolva az egyes videó-, hang- és feliratsávok.

Alkalmazásaink a videó-, hang- és feliratsávok kódolásához és dekódolásához (kibontásához, visszafejtéséhez) úgynevezett codeceket használnak. A codecek olyan szoftverek, amelyek az adott formátumba való kódolást, illetve a kódolt információ megfejtését teszik lehetővé. Nem önálló alkalmazások, hanem a lejátszó- és szerkesztőprogramok használják őket. Egyszerűbb esetekben az alkalmazással együtt települnek az eszközünkre, más esetekben külön telepíthetők.

Míg a képeknél és a hangoknál gyakrabban előfordul, hogy veszteségmentes tömörítést használunk, a videók esetében ez a profi filmkészítők kivételével nagyon ritka – pusztán a képi információ hatalmas mérete miatt. A képsorokat ritkábban állóképek sorozataként

tároljuk. Bonyolultabb, de helytakarékosága miatt lényegesen elterjedtebben használt módszer, hogy csak minden sokadik (például minden ötvenedik, századik) képet tároljuk teljes egészében (ezek az úgynevezett kulcsképkockák, keyframe-ek), két kulcsképkocka között pedig csak a változások kerülnek a fájlba.

Információfogyasztóként egyre ritkábban történik meg az, hogy először teljes egészében letöltjük a videófájlt az eszközünkre, majd egy alkalmazással megnyitjuk és megnézzük. Sokkal gyakoribb, hogy a videót (vagy hangot) az internetről nyitjuk meg, és miközben a videó vagy a hang érkezik az eszközünkre, már nézzük, hallgatjuk is. Ezt a streamingtechnológia teszi lehetővé. (A stream angol szót informatikai környezetben jelfolyamnak fordíthatjuk, a streaming pedig a jelfolyam áttöltését, kezelését, elérhetővé tételét is jelenti.) A „hagyományos” internetes videószoftverek mellett egyre elterjedtebbek a streaming műsorszolgáltatók. A videóhívások és az internetes hanghívások során is streaming történik.



Kérdések, feladatok

1. Nyissunk meg filmet tartalmazó fájlokat valamilyen lejátszóprogrammal! Hány hangsáv és hány feliratsáv van bennük?
2. Milyen kiterjesztéssel menti a mobilkészülökön használt felvevőalkalmazás a videókat? Mekkora a felbontása a felvételnek? Másodpercenként hány képkockát tartalmaz a felvétel? Mekkora időközönként van benne kulcsképkocka?
3. Melyek a leghíresebb videószerkesztő-alkalmazások?

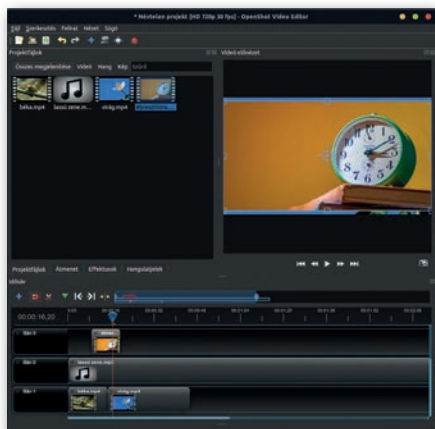
Videószerkesztés

Szerkesztési feladatainkhoz ebben a könyvben egy nyílt forráskódú, mindenki számára ingyenesen használható és viszonylag kis gépigényű alkalmazást, az *OpenShot* videószerkesztőt használjuk.

A filmkészítés első lépése a videó profiljának – felbontásának, illetve másodpercenkénti képkockaszámának – beállítása. Ezt a Fájl menüben tehetjük meg.

Második feladatunk azoknak a videó- és hangfájloknak az összegyűjtése – elkészítése, letöltése, beszerzése –, amelyekből a videónkat összerakjuk. Ezeket például a fájlkezelőből az OpenShot bal felső részébe, a projektfájlok közé húzva tudjuk megmutatni a szerkesztőprogramunknak.

A szerkesztés lényegében a lenti *Idősáv* részen történik. Több időszávunk van, amelyek egyaránt tartalmazhatnak videókat, állóképeket, feliratokat és hangokat – vagy ahogy az Open-



► Az OpenShot alkalmazás

Shot összefoglalóan nevezi őket: *klipeket*. A klipeket a projektfájlok közül az egérrel tudjuk az idősvárra húzni.

Az egy sávon egymás mellé helyezett klipek egymás után következnek majd a végső videóban. Ha fedésbe helyezzük őket, akkor az átfedés részen megjelenik egy új téglalap, amely az átmenetet jelképezi.

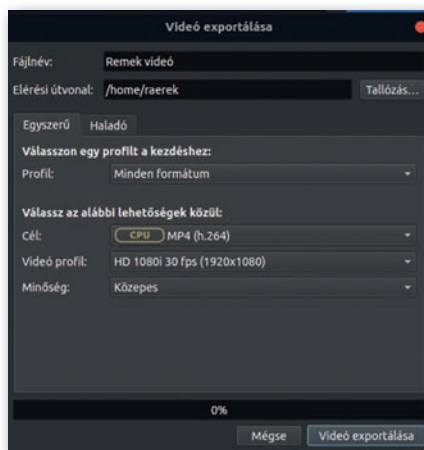
Az átmenetek – eddigi tanulmányainkból leginkább a bemutatókészítéskor megismert – két dia egymásba való átalakulásának módját megadó áttűnésekhez hasonló szerepűek. Ha nem vagyunk elégedettek az alapértelmezettként odakerülő *Keverés* átmenettel, választhatunk helyette másikat.

Az idősvárra helyezett klipek, átmenetek és effektusok vágását, megfordítását, gyorsítását, lassítását és egyéb átalakításait az egyes klipek bal felső sarkában lévő lefelé mutató nyílra kattintva tudjuk elvégezni.

Ha a szerkesztést több idősváron végezzük, akkor a felül lévőkre – a nagyobb sorszámúakra – helyezett klipek elfedik az alul lévők képét, illetve hangját. Az előző oldali ábránkon látható, hogy nagyjából 16 másodpercnél van a lejátszófej és a belőle „lelógó” vonal. Az első sávon lévő virágot elfedi a harmadik sávon található ébresztőóra. A jobb felső részen lévő előnézetben azt is látjuk, hogy az elfedés nem teljes – a virág videója alul és felül kilóg –, mert az órák videó magassága kisebb. Az OpenShot középre helyezte az órák videót, de a fogantyúkkal áthelyezhetjük és átméretezhetjük.

Az OpenShot – hasonlóan az Audacityhez – *projektfájlokként* tárolja az eddig megadott beállításainkat: azt, hogy mi mindent helyeztünk el az idősváron, és milyen beállításokat alkalmaztunk az egyes klipeknél. (Ezek a projektfájlok nem azonosak az ablak bal felső nyelvében lévő projektfájlokkal – azokat érdemesebb lenne például „a projekt forrásfájljai” néven emlegetni.) Ahogy arra számítottunk, az OpenShot projektjeit nem tudja egy egyszerű lejátszóprogram lejátszani, hanem exportálnunk kell a művünket.

*Exportálás*kor figyeljük meg, hogy beállíthatjuk a videó konténerének formátumát (a képen MP4), a videó tömörítésére használt codecet (a képen h.264), a videó felbontását és a másodpercenkénti képkockák számát. A minőségnél azt mondhatjuk meg, hogy mennyire figyelmesen valósuljon meg a veszteséges tömörítés. Amíg csak próbálgatjuk a videóinkat, érdemes gyengébb minőséget választani, a végső változatot pedig a hosszú időt igénybe vevő, de szebb képet adó módon tömöríteni.



Videóötletek

Ebben a leckében néhány olyan rövid videó ötletét olvashatjuk, amelyeket akár mi magunk is el tudunk készíteni. Az alábbiak tényleg csak ötletek, inkább gondolatébresztők, mint kidolgozott tervek – azaz engedje szabadon mindenki a kreativitását. Beszéljük meg, hogy magunkban vagy csapatban dolgozunk, hogy mikor készítjük el a felvételeket, és hogy hol vágjuk össze a videókat! Ahogy az igazi filmek készítésekor, nekünk sem jelent kellemetlenséget, ha egy jelenetet sokszor kell fölvenni. Ha még nem vagyunk gyakorlott videókészítők, bőven elegendő, ha ezek a videók három perc körüli időtartamúak.

Recept

Ebben a videóban egy étel elkészítését mutatjuk be. Majdnem minden mozzanatot felirattal kísérvük, a videóban beszéd nem hangzik el. A képsorokat valamilyen semleges zenével festjük alá.

A videó a kész étel képével indul. Ez lehet állókép, de mozgalmasabbá is tehetjük az étel bemutatását. Ha az étel torta jellegű, akkor levehetjük róla azt a búrát, ami alatt tortát, süteményt szokás tárolni. Ha valamilyen kelt tésztából készült sütemény, akkor azt a konyharuhát libbentjük fel róla, amivel letakarjuk. Ha leves, akkor ki lehet szedni egy tányérba, ha ital, akkor kitölteni, vagy például a díszítés utolsó darabját a helyére tenni.

A következő képen jelenjen meg címként az étel neve.

A hozzávalókat szintén kiírjuk. Alattuk – arra az egy-két másodpercre, amíg a felirat látható – a hozzávaló beszerzését mutató képet vetítünk. Ha gyümölcs, akkor látszódjon, ahogy leszedjük a fáról, vagy betesszük a piacon a kosárba. Ha hús, akkor bemutatathatjuk még élő korában, ha liszt, akkor felvehetjük, ahogy belefújunk, és az arcunkra vagy a kamera előtt lévő üveglapra rakódik, ha tojás, akkor vegyük ki a tyúk alól, amely jobb híján lehet éppen plüssmadár is.

Az étel elkészítésének lépéseit illusztráljuk, például a meggy kimagozását, a szárzestésza kiöntését, a tojás feltörését. Törekedjünk arra, hogy önmagunkban is érdekes képsorokat készítsünk, aminek egyik legegyszerűbb módja, ha a teljesen hétköznapi cselekedeteket meglepő szögből vesszük fel. A tojástörés történjen szemmagasságban, és – ha van rá mód – készüljön 60 vagy több képkockával másodpercenként, hogy legyen módunk a drámai pillanat lassítására. A tésztát öntsük átlátszó edénybe, amely alatt van a kamera, azaz a tészta hulljon a néző arcába. A meggy kimagozását vegyük fel közelről, és fröccsenjen a meggylé, vagy állóképek sorában szaporodjanak a meggymagok. Ha szükségét érezzük, írjuk ki, hogy épp mit csinálunk, mi történik a képen.

Illusztráljuk külön a főzés vagy sütés elejét és végét! Remek módszer, ha a – még hideg – sütőben helyezzük el a telefont, majd felvesszük, ahogy nyílik a sütőajtó, és szemből megérkezik a tepsi. A kivételkor csalhatunk: ha látszik, hogy mit veszünk ki, akkor várjuk meg, míg kihűl a sütő, és vegyük ki belőle az immár aranybarna süteményt vagy az elfolyt sajttól roskadozó rakott krumplis tepsit még egyszer. Ha úgyszem látszik, hogy mit veszünk ki, akkor a sütés megkezdése előtt ki is vehetjük a „kész” ennivalót. Aztán persze ne feledjük tényleg megsütni!



A tálalás is helyet kaphat a filmben – akár ebből a képsorból is kiválaszthatjuk a nyitóképet. Ha a videó hangulata engedi, helyezzünk el pár képsort elégedetten táplálkozó ember-társainkról vagy magunkról.

A zárókép lehet néhány megmaradt morzsa, hatalmas torony a használt tányérokból, vagy a konyhában maradt rendetlenséget bemutató képsor. Rosszul sikerült főzőcske végére kerülhet egy pár másodperces felvétel a boldogan csámcsogó kutyánkról.



Variációk a recept témára

A recepthez nagyon hasonlóan készíthető el bármilyen, több lépésből álló munkáról szóló videó, legyen az egy akvárium berendezése, egy frizura elkészítése, egy asztal lefestése, egy köröm kilakkoztatása vagy egy makett elkészítése.

Riport

Riportból készíthetünk olyat, amelyben egyetlen riportalany válaszol több kérdésre, de olyat is, amelyben ugyanazt az egy-két kérdést válaszolja meg sok riportalany.

Egy riportalany

Több iskolában szokás, hogy a diákok sorozatot indítanak „Bemutatjuk tanárainkat” vagy „Bemutatjuk diáktársainkat” témában. Az ilyen videók általában a diákság és a tanárok érdeklődésére is számot tarthatnak. Ha megijednénk a feladat nagyságától, vigasztalódjunk azzal, hogy elég, ha az osztályunk csak egy-két riportot készít el, hisz igazán hagyni kell valamit azoknak is, akik jövőre lesznek nyolcadikosok. A különösen jól sikerült riportok akár kikerülhetnek az iskola honlapjára is.

Írjuk össze a kérdéseinket előre, és vitassuk meg, hogy melyek várhatóan valóban érdekelni fogják a leendő nézőinket! Így nem fogjuk fölösleges, a készre vágott videóban már nem szereplő, fel nem használt kérdésekre pazarolni a riportalany idejét – nem is beszélve a sajátunkról.

Döntsük el már az elején, hogy benne lesz-e a végső változatban az is, ahogy a riporter felteszi a kérdést, vagy csak kiírjuk a kérdést, vagy ki sem kell majd írni, mert a válaszból úgyis kiderül.

Az egyetlen riportalannyal készülő videók esetében különösen könnyű unalmas és hosszú videókat készíteni. Hogyan kerülhető el ez a csapda?

Válasszunk érdekes háttérrel a szereplő mögé! Ha nem igazán sikerül ezt megoldani, akkor keressünk példát kedvenc videósatornáinkon látható interjúkból: egy-egy érdekes színpad fény a háttérben, egy-egy jól berendezett polc – különösen, ha a riportalanyra jellemző –,



akár személyes tárgyakkal megrakva, jó szolgálatot tehet. Érdekes lehet egy mozgalmas háttér is, de ez esetben nagyon kell figyelnünk arra, hogy a mikrofonon, illetve a felvételt készítő alkalmazás hatékonyan szűrje ki a zajt.

A riportot több kamerával, többféle állásból vegyük fel. Egy kamera veheti azt a képet, amit a riporter lát – általában szemben vagy majdnem szemben a riportalannyal –, de ha van rá mód, állítsunk be több kamerát is, akár meglepő szögekből. Készülhet például felvétel a riportalany vállá fölött, ahogy a riportert látja, vagy oldalról, az arc profilját rögzítve. Ha a riportalany szokott gesztikulálni, akkor egy kamerával érdemes lehet a kezét venni. Ezeket a képeket hosszabb válaszok alatt, legfeljebb néhány másodpercnyi időben vágjuk be.

Próbáljuk ki a helyszínt – még azelőtt, hogy a riportalanyt odacitálnánk. Stábunk valamelyik tagja lesz a „riportalany”. Érdemes olyasvalakit választanunk, aki termetre hasonlít a leendő riportalanyunkhoz. Próbáljuk ki, hogy hova tesszük a kamerákat, miként fogjuk negyedórán keresztül egy helyben tartani őket – azaz milyen támasztékot, állványt használunk. Keressünk jó beállításokat! Nézzük meg, hogy elegendő-e a fény, érdekes-e a háttér!

Hogyan vesszük fel a hangot? Ha van mikrofonunk, az nagyon jó, de mit tudunk tenni, ha nincs? Mennyire kell közel tenni a mikrofont? Ha egy telefonba kell beszéltetnünk a riportalanyt, hogyan oldjuk meg, hogy a telefon elég közel legyen, de ne nagyon látsszon a felvételen? Mennyire lesz zajos a felvétel? Keletkezik-e visszhang? Ki tudjuk-e szűrni a zajt?

Készítsünk vágóképeket! Ezek a képek sokszor csak távolabbról kapcsolódnak a riport témájához, és a riporttól függetlenül is elkészíthetjük őket. Ha tanár vagy diák a riportalanyunk, felvehetjük, ahogy reggel megérkezik az iskolába – kinyit egy ajtót, lezár egy biciklit, összecsup egy esernyőt, megköt egy tornacipőt, bekapcsol egy számítógépet. Ha valamelyik szertárban vagy az iskolakönyvtárban készül a riport, készíthetünk olyan felvételt, ahol a kamerát végighúzzuk valamelyik polc előtt, megmutatva, hogy mi minden van rajta. Ha a szakáccsal készítünk riportot, legyen olyan kép, ahogy begombolja a munkaruha felső gombját, és fejébe húzza a sapkát. Ezeket a képeket ismét csak pár másodpercre villantjuk majd fel a válaszok alatt.

Az utómunkák során a szüneteket vágjuk ki, a nem sikerült félmondatokat, mondatokat úgyszintén! Ha szükséges, és nem töri meg a hangulatot, halk zenével feshetjük alá a riportot.

Sok riportalany

Az előzőekben említett gondos előkészületre itt aligha lesz mód, hiszen jó eséllyel az iskola folyosóján fogunk megkérdezni sok diákot, például arról, hogy melyik a kedvenc étele a büféből, mit kellene tenni még az iskolai honlapra, vagy hogy szerinte ki nyeri majd a tanár-diák meccset.

Itt is érdemes kipróbálnunk a szituációt. Mennyire kell közel tenni a mikrofont, hogy biztosan érthető legyen, amit mond a riportalany? Kell-e valamilyen megvilágítás? Melyik az a lépcsőforduló, amelyikben úgyszintén lelassítanak az emberek? El fogunk-e félni a menza előtti sor mellett, a sorban állókat kérdezgetve?



Szokás az ilyen videókat úgy összevágni, hogy az azonos válaszok egymás után jöjjenek – már csak azért is, mert érdekes hatást kelthet, ha tizedjére mondja be valaki, hogy „tutira a diákok”, „naná, hogy a diákok”.

Út az iskolába

Ez a videó a személyes élmények videója lesz. A hozzá való felvételeket akár több napon keresztül is érdemes készíteni – az „igazi” videók sokszor készülnek hetekig. Ennyi idő kell ugyanis, mire elegendő érdekes esemény történik pár perc kitöltéséhez, és ráadásul a felvételek minőségével is elégedettek vagyunk.

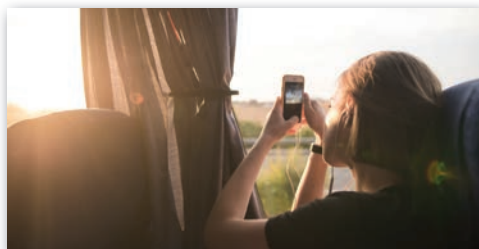
Amikor élményekről, érdekes eseményekről beszélünk, nem kell nagyon nagy dolgokra gondolni, szerencsére viszonylag ritkák az akciófilmekbe illő jelenetek egy iskolába vezető út, utazás alatt. Ez a videó így a pillanatok élvezetéről is szól majd – lehet, hogy a végén csak olyan felvételek lesznek benne, amelyek a riportos videóban vágóképek lennének.

Kezdődhet a filmünk az ébredésünkkel. Ha van, aki felveszi álmos fejünket, akkor jó, de készíthetjük mi is a felvételt. Vegyünk fel mozzanatok: ahogy a helyére akasztjuk a törülközőt, ahogy a reggeli vajjas kenyeret kenjük, a tejeskávét kavarjuk, a fogunkat mossuk, a macskát etetjük, a zárban elfordítjuk a kulcsot. És hogy kinek van reggel mindezt ideje felvenni? Senkinek, csalnunk kell – szerencsére a zár nem nyikorog délben sem másként, és a macska is fog örülni egy extra porciónak.

Menet közben felvehetjük utunk állandó szereplőit: a harmadik ház kertjében a jácintokat vagy épp az ugató kutyát, a beforduló buszt, a megérkező metrót, a kirakatot. Ha az út egyhangú, vagy épp vidám ismétlődéseit szeretnénk kiemelni, bizonyos jeleneteket több napon, többször is felvehetünk, és több felvételt is szerepeltethetünk: a vonat ajtajának csukódását, a bérlet felmutatását, ahogy a zebránál zöldre vált a lámpa, és elindulnak a szembejövők.

Ha szeretnénk, a felvételeink lehetnek nyugodtan lüktető, kellemes aláfestő zenével, de kísérletezhetünk mással is. Felvehetjük és gyorsítva, érdekes hanghatással aláfestve bevághatjuk azt, ahogy lemegyünk a mozdólépcsőn, vagy ahogy öt másodperc alatt teszi meg a vonat a negyedórás utat.

Lehet az egész út egyetlen felvétel is, amelyben a hosszabb részeket gyorsítjuk, az érdekesebbeknél pedig megállunk egy-egy pillanatra. Ilyenkor akár meg is nyilazhatjuk a képen a látnivalókat: a híres épületet éppúgy, mint a különös falfirkát, a betört ablakot,



a belvárosban közlekedő lovas kocsit, a kreatívan átrajzolt plakátot, a képbe csak épp belógó csodaautót vagy kedvenc gerlénket a faágon.

Elképzelhető olyan változat is, amelyet oda nem illő vagy csak részben odailló hanghatásokkal teszünk érdekessé. Ilyen hatás lehet a szemünk nyitását kísérő csikorgás, a tojás feltörését aláfestő üvegcsörömpölés, a kutyaugatás helyett elhangzó macskanyávogás, a villamos indulásakor felhangzó vonatfütty és az a zenedobozból szóló altató, amely akkor csendül fel, amikor a padunkba ülünk a záró képsoron.

Bizonyos témák esetében nagyban elősegítheti a videó elkészítését egy akciókamera – kérdezősködjünk, hátha tud valaki kölcsönadni egyet. Az akciókamerával egyszerűbb elkészíteni a teljes utat bemutató videókat, és a kerékpárral közlekedőknek sem kell lemondaniuk kedvenc közlekedési eszközük használatáról.

Variációk az Út az iskolába témára

Az iskolába vezető útról szóló rövidfilmhez nagyon hasonlóan készíthető el bármilyen, egyik pontból a másikba jutást bemutató videó. Filmre vehetjük kirándulásunkat, ahogy odaérünk a hétvégi meccsre vagy az esti színházi előadásra.

Ennél a témánál a felvételünkre hamar rákerülnek olyan emberek, akiknek a beleegyezését ehhez nem kaptuk meg, és így törölnünk kell a felvételeket. Szerencsére lesz elég olyan mozzanat, amelyekből hamar összeállítható egy-két percnyi kisfilm.

Kedvenc háziállatom

Az ebben a videóban szereplő háziállat természetesen nem csak a hagyományosan háziállatnak tekintett fajok közül kerülhet ki. Mindenféle házi kedvenc remek téma lehet, az aranyhórcsögtől a papagájokon keresztül a tarantuláig. Sőt, nem is kell házi kedvencnek lennie, mindössze annak a feltételnek kell teljesülnie, hogy tudjunk róla egy-két percnyi érdekes felvételt készíteni. Lehet a főszereplő egy cipekedő hangya vagy néhány szorgos méh.

A videóban pár mondat szentelhető az alábbi témáknak:

- A faj története – honnan származik, mióta tarják háziállatként vagy házi kedvencként?
- Mikor és mivel kell etetni az állatot?
- Milyen ellátást igényel az etetésen felül?
- Mennyi költséggel jár a beszerzése, és mennyivel az ellátása?
- Mennyi foglalkozást igényel?
- Kinek, milyen természetű, temperamentumú embernek ajánljuk?
- Kinek nem ajánljuk?
- Miért szeretjük az állatunkat?
- Mik a számunkra legkedvesebb ténykedései?

Természetesen nem kell mindennek megjelennie a videóban, elég, ha néhány témát érintünk a felsoroltak közül.



Döntsük el, hogy mi magunk fogjuk mindezt elmondani, vagy egy-egy mondatban kiírjuk, és inkább zenével kísérvük a képeket. Megoldás lehet az is, ha alapvetően zenét használunk, de az egy-egy elhangzó mondat idejére a zenét lehalkítjuk.

Remélhetőleg sikerül sok érdekes felvételt készíteni. A macskás videóknak hosszú sora van már az interneten, és valószínűleg mindenki látott már belőlük eleget, de remekül tud olvasni a közönség a tollászokodó madarak és az ugrándozó kutyák láttán is. Vadászó pókból vagy vacsoráját fogyasztó gekkóból alighanem kevesebbet látott a nagyérdemű, és kellően hátborzongató zenével egy másfajta, de az előzőeknél cseppet sem kisebb élményt adhatnak képsoraink.



A rövidfilm fő témája – az állat cukiságának vagy épp vérfagyasztóságának bemutatása – mellett illusztráljuk pár perces képsorokkal a videóban említett állatunkkal kapcsolatos tevékenységeket. Néhány másodpercben mutassuk be, hogy miként tartjuk tisztán a kutyaházat vagy a terráriumot, hogyan készítjük el az ennivalót.

Zárásképp

A videóknak végén pár másodpercben jelenjenek meg azok az információk, amelyekkel a szerzői jog tiszteletben tartásának teszünk eleget – például az, hogy a videóknak legálisan letölthető és felhasználható zenéi, hanghatásai honnan származnak.

A videók elkészítése során figyeljünk arra, hogy az alkotás lázába belemerülve ne okozunk és ne szenvedjünk balesetet!

A táblázatkezelés alapjai

Az adatokat könnyebben áttekinthetjük, ha táblázatosan rendezzük el őket. Erre használhatjuk például a szövegszerkesztő programok 7. évfolyamon megismert táblázatkészítő funkcióját is, de gyakran elegendő egy közönséges négyzetrácsos („kockás”) papír. Ugyanakkor, ha sok adatunk van, és ezekből összefüggéseket kell számolnunk, vagy ha diagramot kell készítenünk, akkor már célszerűbb a **táblázatkezelő program** alkalmazása.

Ebben a fejezetben előbb áttekintjük egy példán keresztül a táblázatkezelő programok legfontosabb funkcióit, majd újabb példák segítségével részletesebben megismerkedünk velük.

Eszközként a Microsoft Excel és a LibreOffice Calc programok asztali változatait használjuk, de a legtöbb példa könnyen megvalósítható akár a Google Sheets, akár a Microsoft Excel online változatában is.

Példa: Szeptemberi piknik

Az Irka Iskola Diákönkormányzata minden év szeptemberében iskolai pikniket szervez. Ennek előkészítésébe és lebonyolításába minden osztály bekapcsolódik. A hagyományok szerint a 8. osztály például büfét szervez, ahol az általuk készített ételeket és italokat árúsják. A bevételt a Diákönkormányzat iskolai programokra fordítja.

Az osztály DÖK-képviselői a nap végén táblázatkezelő programmal összesítették a napi bevételt termékenként:

	A	B	C	D	E
1	Termék	Eladott	Egység	Egységár	Összeg
2	Zsíros kenyér	240	szelet	120 Ft	
3	Vajas-mézes kenyér	112	szelet	180 Ft	
4	Limonádé	198	3 dl	160 Ft	
5	Csipkebogyótea	77	2 dl	180 Ft	
6	Sós muffin	84	db	150 Ft	
7	Édes muffin	124	db	170 Ft	
8	Palacsinta (túrós)	160	db	250 Ft	

► A 8. a osztály bevételei a szeptemberi pikniken

Indítsunk el egy táblázatkezelő alkalmazást, és írjuk be a fenti adatokat! Beírás közben észrevehetjük, hogy például a *Zsíros kenyér* nem fér ki az adott cellában: látszólag a szomszéd cellában folytatódik. Ha azonban beírjuk oda az

eladott szeletek számát (240), már csak a szó egy része jelenik meg. Ilyenkor **az egérrel** (a bal gomb nyomva tartása mellett) **megnövelhetjük az oszlop szélességét**.

	A	B	C	D	E
1	Termék	Eladott	Egység	Egységár	Összeg
2	Zsíros ken	240	szelet	120 Ft	
3	Vajas-mézi	112	szelet	180 Ft	

► Oszlop szélességének növelése

A szöveg bevitelekor ügyeljünk a pénznem (120 Ft) beírására! A szám és a pénznem jele között egy szóköz van, a forint jele pedig Ft (és nem ft vagy FT). A hibát utólag úgy javíthatjuk, hogy kettőt kattintunk az egér bal gombjával a hibás adatot tartalmazó cellára, majd átírjuk a tartalmát.

A táblázat celláiban a szövegszerkesztésben megszokott módon formázhatjuk a karaktereket, ezúttal például az első sorban szereplő címeket félkövér betűstílusúra állítottuk.

A táblázatkezelő program az adatokat egy *munkafüzetben* tárolja, a munkafüzet *munkalapokból* áll.

A munkalap oszlopait az *A* betűvel kezdve betűkkel, sorait *1*-től kezdve számokkal azonosítjuk. Az egyes *cellákra* oszlopuk betűjelével és soruk számával hivatkozhatunk, például *B2*-es cella.

Azt a cellát, amelyikbe éppen adatot írhatunk be, *aktív cellának* nevezzük. A következő cellába a kurzormozgató gombokkal vagy a tabulátor gombbal léphetünk.

Egy cellába mindig csak egy adat kerül. A **táblázatkezelő program alapértelmezés szerint a cellákban lévő szöveget balra, a számot jobbra igazítja.**

Képletek, függvények beszúrása

Számítsuk ki az *Összesen* oszlopban, hogy mennyi volt az egyes termékekből származó bevétel! Mivel ez az érték az *Eladott* mennyiség és az *Egységár* szorzata, az *E2*-es cellába például a *B2*-es és *D2*-es cellák szorzata kerül. A táblázatkezelő egyik erőssége, hogy ha az ennek megfelelő $=B2*D2$ képletet írjuk be az *E2*-es cellába, akkor nemcsak elvégzi a számítást, hanem az eladott mennyiség vagy egységár változtatása esetén azt automatikusan újra is számítja.

	A	B	C	D	E
1	Termék	Eladott	Egység	Egységár	Összeg
2	Zsíros kenyér	240	szelet	120 Ft	$=B2*D2$
3	Vajas-mézes kenyér	112	szelet	180 Ft	20 160 Ft
4	Limonádé	198	3 dl	160 Ft	
5	Csipkebogyótea	77	2 dl	180 Ft	
6	Sós muffin	84	db	150 Ft	

► Képlet beillesztése és másolása

Hasonló módon az *E3*-as cellába az $=B3*D3$ képletet írhatjuk. Minden képlet egyenlőségjellel (=) vagy pluszjellel (+) kezdődik.

A sort folytathatjuk ugyan az *E4*-es, *E5*-ös stb. cellákkal, de egy hosszú táblázat esetén valamennyi képlet beírása így nehézkes. A képletet azonban másolhatjuk is: kattintsunk egyet a képletet tartalmazó *E3*-as cellára, és húzzuk az annak jobb alsó sarkában megjelenő + jelet lefelé! Ez a művelet úgy másolja át a képletet az érintett cellákba, hogy a képletben lévő cellahivatkozás sorszáma minden sorban – a másolás irányának megfelelően – eggyel nő, rendre $=B4*D4$, $=B5*D5$ stb. lesz. A másolást a szokásos módon, a **CTRL + C**, **CTRL + V** billentyűkombinációval is elvégezhetjük.

Határozzuk meg a teljes napi bevételt! Ez az *E* oszlopban lévő cellák tartalmának összege, vagyis kiszámolható például az $=E2+E3+E4+E5+E6+E7+E8$ képlettel. Nyilván kell

lennie más megoldásnak, hiszen ez a képlet egy több ezer soros táblázatban nem alkalmazható. A probléma megoldására a **SZUM** függvényt használhatjuk, amelynél paraméterként a zárójelben elegendő kettősponttal elválasztva megadni az összeadandó értékek első és utolsó celláját. Az E9-es cellába tehát írjuk inkább a következő képletet: **=SZUM(E2:E8)**

A függvény neve után zárójelben lévő kifejezés, az E2:E8, egy téglalap alakú területet, a táblázatkezelő programok szóhasználatában *tartományt* jelöl ki, amelynek bal felső sarka az E2-es, jobb alsó sarka az E8-as cella.

6	Sós muffin	84 db	150 Ft	12 600 Ft
7	Édes muffin	124 db	170 Ft	21 080 Ft
8	Palacsinta (túrós)	160 db	250 Ft	40 000 Ft
9	Teljes bevétel:			=SZUM(E2:E8)

► A teljes bevétel meghatározása a SZUM függvény alkalmazásával

A táblázatkezelő programokban **képletek** használatával végezhetünk számításokat.

A képletek eredményét a programok az adatok módosulása esetén **automatikusan újraszámítják**.

A képletet az egér húzásával másolhatjuk. Másoláskor az érintett cellákba – ha mást nem írunk elő – a *másolás irányának megfelelően* módosított képlet kerül.

Egy téglalap alakú területet **tartománynak** nevezünk, és a bal felső, valamint jobb alsó sarkában lévő cellákra való hivatkozással adjuk meg, például A3:G8.

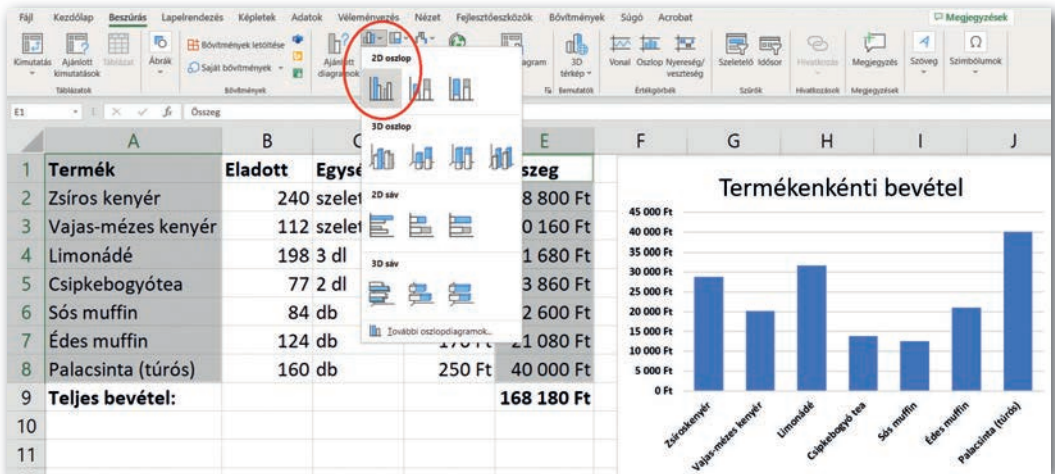
Összetettebb számítások elvégzésére a táblázatkezelő programok **függvényeket** kínálnak. A **SZUM** függvény például a zárójelben megadott tartomány celláinak tartalmát összegezi.

Az adatok szemléltetése diagramon

Az osztály DÖK-képviselői szeretnék a bevétel adatait látványosan bemutatni az Iskolai Diákönkormányzat tagjainak. Azt akarják szemléltetni, hogy az osztály egyes termékei milyen kapósak voltak. Ezért az egyes termékek bevételi adatait egy oszlopdiagramon ábrázolják.

A táblázatkezelő programok automatikusan elkészítik a kiválasztott diagramot. Ehhez először ki kell jelölnünk a megfelelő adatokat, például a termékek nevét tartalmazó A1:A8 és a termékenkénti bevételt tartalmazó E1:E8 tartományt (az első sorban lévő címekkel együtt). A két tartományt úgy jelölhetjük ki egyszerre, hogy végighúzzuk az egeret az A1:A8 tartományon, majd a CTRL gombot lenyomva tartva végighúzzuk az egeret az E1:E8 tartományon is.

Az adatok kijelölése után a *Beszűrés* lapon (illetve a *Beszűrés > Diagram* menüponttal) választhatjuk ki a megfelelő diagramtípust, ezúttal egy „hagyományos” (nem térhatású) oszlopdiagramot. A besűrt diagram helyét és méretét – a szövegszerkesztőben az objektumok mozgatásához, átméretezéséhez hasonlóan – egérrel módosíthatjuk. Ugyanígy a diagram egyes részeit kijelölve, a menürendszer segítségével megváltoztathatjuk azok színét, a tengelyeken lévő betűk típusát és méretét stb. Példánkban a diagram címét átírtuk *Termékenkénti bevételre*. (Ehhez előzőleg a táblázatkezelő program által felajánlott diagramcímre kell kattintanunk.)



► A bevétel megoszlásának szemléltetése oszlopdiagrammal

Feladatok

- A Diákönkormányzat nem a teljes befolyt összeget kapja meg az eladások után, hanem csupán az italok eladásából származó bevételek felét és az ételek utáni bevételek harmadát.
 - Írjuk be az F1-es cellába a DÖK szót, és számítsuk ki alatta termékenként a DÖK-nek fizetendő összeget!
 - Függvény segítségével határozzuk meg a DÖK-nek fizetendő teljes összeget az F9-es cellában!
 - Ábrázoljuk kördiagramon a DÖK-nek fizetendő összeg termékenkénti megoszlását!
- A napi bevétel összegezéséhez az egyik DÖK-képviselő táblázatot készített az alábbiak szerint.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1 címlet		5	10	20	50	100	200	500	1000	2000	5000	10000	20000	
2 darab		40	212	108	56	87	76	26	18	3	8	2	2	
3 összeg		200	2120	2160	2800	8700	15200	13000	18000	6000	40000	20000	40000	168180

A táblázat első sorában a forgalomban lévő címletek szerepelnek, a másodikban az adott címletből befolyt pénzürmék vagy bankjegyek száma, a harmadikban pedig a teljes befolyt összeg címletenként. Például 20 Ft-osból 108 db jött össze, és ez 2160 Ft-ot jelent.

- Készítsük el a minta szerinti táblázatot!
- Képlet segítségével határozzuk meg a befolyt összeget címletenként!
- Számoljuk ki függvény segítségével a teljes napi bevételt az M3-as cellában!
- Formázzuk meg a táblázatban lévő karaktereket a mintának megfelelően!

Táblázat formázása. Cellahivatkozások

Példa: Biciklitúra

A 8. b osztály a nyáron egyhetes biciklitúrát tett. A tervezett út 300 km volt, amelyet az alábbi táblázat A és B oszlopa szerint bontottak napokra. A napi terv a látnivalóktól függően különböző volt, és ha kellett, többet pihentek, vagy kihagytak egy-egy látnivalót. Így a ténylegesen megtett utakat a C oszlop tartalmazza; azonban az osztályfőnök azt is kiszámolta a D oszlopban, hogy hány százalékra teljesítették az aznapi tervet. Először az A1:D9 tartományban lévő táblázatot készítjük el.



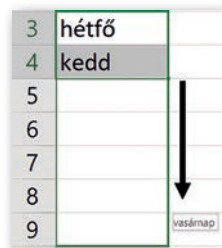
	A	B	C	D	E	F	G	H
1	Kerékpártúra							
2	nap	terv	tény	teljesítés		defekt	egységár	díj
3	hétfő	40	45	113%		1	1 698 Ft	1 698 Ft
4	kedd	50	55	110%		2	1 698 Ft	3 396 Ft
5	szerda	60	50	83%		4	1 698 Ft	6 792 Ft
6	csütörtök	50	45	90%		1	1 698 Ft	1 698 Ft
7	péntek	20	20	100%		3	1 698 Ft	5 094 Ft
8	szombat	50	20	40%		6	1 698 Ft	10 188 Ft
9	vasárnap	30	50	167%		0	1 698 Ft	0 Ft
10	Összesen	300	285	105%		17		28 866 Ft

► A 8. b osztály nyári biciklitúrájának adatai

Első lépésként töltsük fel az A1:D2 tartomány celláit a megfelelő címekkel! A következő lépés a napok beírása lenne az A3:A9 tartományba, de ezúttal nem kell minden napot kézzel beírunk. Írjuk be csak az első két napot (hétfő, kedd), majd jelöljük ki ezt a két cellát, és a kijelölt tartomány jobb alsó sarkában lévő kis négyzetet húzzuk lefelé, amíg meg nem jelennek a szükséges napok!

Hasonlóan készíthetnénk el a hónapok nevét tartalmazó sorozatot vagy akár egy számsorozatot is. Számok esetén ugyanis a táblázatkezelés programok az első két számból indulva, azok különbségével egy számtani sorozatot hoznak létre, például 1 és 8 esetén: 1, 8, 15, 22, 29...

A naponta teljesített út arányát például a D3-as cellában az =C3/B3 képlet adja. Ezt a korábban megismert módon másolhatjuk a D4:D9 tartomány celláiba. A mintán a kapott értékeket százalékban kifejezve látjuk. Ehhez jelöljük ki a C3:C9 tartományt, majd válasszuk a % ikont! (Az alkalmazott programtól függően lehet, hogy az eredmények tizedesjegyekkel jelennek meg.)



► Kitöltés sorozattal

A 10. sorban az adatok összegzésére érdemes a már megismert SZUM függvényt alkalmaznunk, így például a B10-es cellába az =SZUM(B3:B9) képlet kerül. (Vajon milyen képlet került a D10-es cellába?)

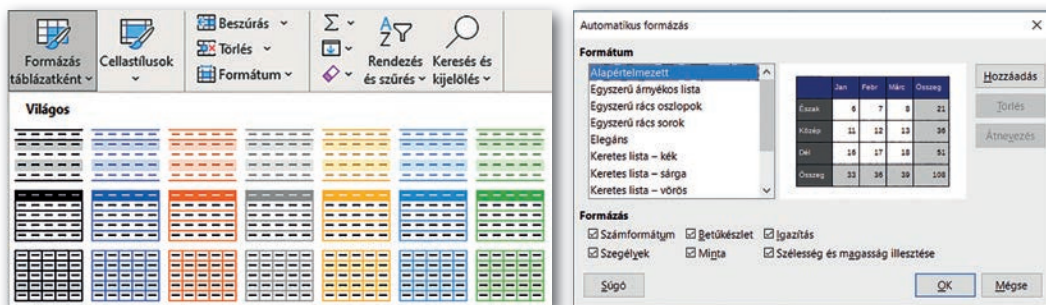
Sajnos a biciklitúrák természetes velejárója a defekt. Az osztályt kísérő osztályfőnök-helyettes ezt a legegyszerűbb módon oldotta meg: a defektes belsőket kicserélte, az eredetit pedig javításra visszaadta a kerékpár tulajdonosának. A napi defektek számát az F oszlopban találjuk, az új belsőök értékét pedig a G oszlopban. Hozzuk létre önállóan a H oszlopban szereplő defektek javítására kiadott összegeket naponta! (Mivel az osztályfőnök nagy mennyiségben előre beszerezte a szükséges belsőket, azok értéke minden esetben ugyanannyi volt.)

A 10. sorban ezúttal is határozzuk meg képlet segítségével a defektek számát, illetve a belsőkre összesen kifizetett díj összegét!



A táblázat formázása



A táblázatkezelő programok általában rendelkeznek előre elkészített táblázatformátumokkal, amelyeket egyszerűen „ráhúzhatunk” a kijelölt tartományra. Ezt a listát elérhetjük például a Kezdőlap > Formázás táblázatként paranccsal (illetve a Formázás > Automatikus formázási stílusok menüponttal). Az így kialakított formátumot utólag módosíthatjuk.



► Automatikus formázási lehetőségek (balra Microsoft Excel, jobbra LibreOffice Calc)

A továbbiakban megvizsgáljuk, melyek a legfontosabb lehetőségeink, ha „kézzel”, és nem automatikusan formázunk.

Először alakítsuk ki a címet! Jelöljük ki az A1:D1 tartományt, és **egyesítsük a tartomány celláit**, például a Cellaegyesítés (illetve a Cellák egyesítése és...) ikonnal. A program a táblázat címét automatikusan középre zárja az egyesített tartományban. (Az egyesített cellákat szükség esetén ugyanezzel az ikonnal tudjuk később cellákra bontani.) Alkalmazunk a címre félkövér, sötétzöld betűformátumot, és a betűk méretét növeljük meg!

A cím alatti mezőnevekre érdemes hasonló színeket alkalmazni: állítsuk a **háttért** sötétzöldre a  **Kitöltőszín** (illetve a  **Háttérszín**) ikonnal, a cellák szövegét pedig félkövér, fehér színűre. Hasonló elrendezést érdemes az **Összesen** sorban is létrehozni.

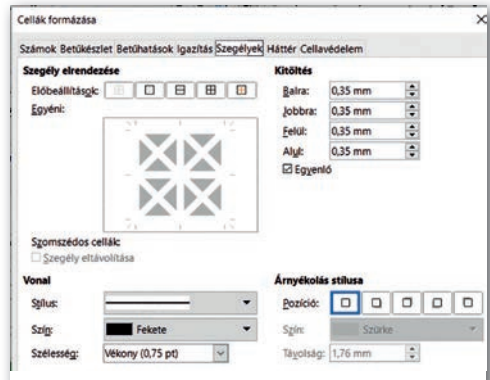
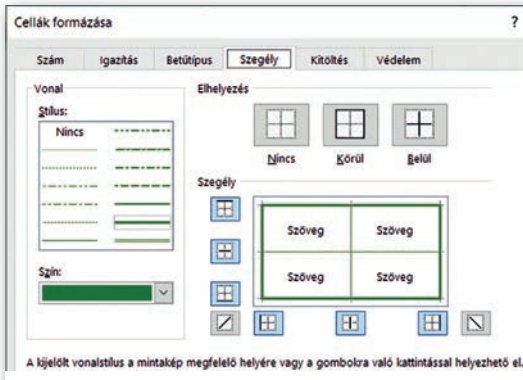
Kerékpártúra			
nap	terv	tény	teljesítés
hétfő	40	45	113%
kedd	50	55	110%
szerda	60	50	83%

► A formázott táblázat egy részlete

A táblázat összetartozó elemeit **szegély** alkalmazásával különítjük el.

A munkalap celláit a képernyőn alapértelmezés szerint egy **cellarács** határolja, amelyet akár nyomtatáskor is megjeleníthetünk. Ugyanígy a **Nézet > Rácsvonalak** (illetve a **Nézet > Rácsvonalak megjelenítése**) menüponttal a cellarácsot a képernyőn ki-, illetve bekapcsolhatjuk.

Az alapértelmezett cellarácsot a szegélyek beállításával testre szabhatjuk, ekkor a szegélyek a képernyőn, illetve nyomtatáskor a papíron is megjelennek. A szegélyeket beállíthatjuk például a **Kezdőlap > Szegélyek** (illetve a **Formátum > Cellák > Szegélyek**) menüpont lehetőségeivel. A szegélyek beállításakor megadhatjuk a vonal vastagságát, színét, illetve stílusát (szaggatott, dupla stb.). Ezeket a beállításokat munkalaponként végezhetjük, így a címsorok és az adatok esetében eltérő módon formázhatjuk. Például az **F2:H10** tartomány celláit határolhatjuk belül sötétzöld, vékony szegéllyel, magát a tartományt és azon belül az első és utolsó sort pedig vastag, sötétzöld szegéllyel.



► Szegély kialakítása (balra: Microsoft Excel, jobbra: LibreOffice Calc)

Előfordulhat, hogy egy táblázat sorait vagy oszlopait törölnünk kell. Ilyenkor a **DELETE** gomb lenyomása csak az adatokat törli, a formázási beállítások megmaradnak. A végleges törléshez, illetve új sorok vagy oszlopok beszúrásához a menü megfelelő pontját kell alkalmaznunk. Sorok beszúrása esetén alkalmazhatjuk például a **Kezdőlap > Beszúrás** (illetve **Munkalap > Sorok beszúrása**) parancsot.

Abszolút cellahivatkozás

A **G** oszlop celláiban minden napnál ugyanazt az értéket látjuk. Adódik, hogy a kerékpárbelsők árát csak egy helyen szerepeltessük, például a **G2**-es cellában, és az alatta lévő cellák értékét ennek felhasználásával számoljuk ki. Például a **G3**-as cellában az **=F3 * G2** képlettel, amelyet az eger húzásával másolhatunk lefelé.

Azonnal láthatjuk, hogy hamis eredményt kaptunk, teljesen kizárt például, hogy ez az összeg 6 defekt esetén, szombaton 244 512 Ft legyen. A táblázatkezelő programok hasznos szolgáltatása, hogy a cellában a képlet értéke helyett magát a **képletet jelenítik meg**, tipikusan az **Alt .** (alt pont) **billentyűkombinációval**. Az eredmény nem meglepő: a képlet másolásakor a cellacímek a másolás irányának megfelelően módosultak. Így a G2-ből rendre G3, G4 stb. lett, miközben azt szerettük volna, hogy végig a G2 cellahivatkozás szerepeljen.

	F	G	H
2	egységár:	1 698 Ft	
3	1	1 698 Ft	=F3*G2
4	2	3 396 Ft	=F4*G3
5	4	13 584 Ft	=F5*G4
6	1	13 584 Ft	=F6*G5
7	3	40 752 Ft	=F7*G6
8	6	244 512 Ft	=F8*G7
9	0	0 Ft	=F9*G8
10	17	317 526 Ft	

	F	G	H
2	egységár:	1 698 Ft	
3	1	1 698 Ft	=F3*\$G\$2
4	2	3 396 Ft	=F4*\$G\$2
5	4	6 792 Ft	=F5*\$G\$2
6	1	1 698 Ft	=F6*\$G\$2
7	3	5 094 Ft	=F7*\$G\$2
8	6	10 188 Ft	=F8*\$G\$2
9	0	0 Ft	=F9*\$G\$2
10	17	28 866 Ft	

- ▶ A G3-as cellában szereplő képlet módosulása másolás során, ha a képletben a G2-es cellára relatív (balra), illetve abszolút (jobbra) cellahivatkozás szerepel

Mivel a cella értékének rögzítése gyakran szükséges, ezért a táblázatkezelő programok ezt a műveletet lehetővé teszik. Ha a képletbe G2 helyett \$G\$2-t írunk, másolásakor a képletben a cella címe nem változik, így az helyes eredményt ad.

Egy cellahivatkozást *relatív cellahivatkozásnak* nevezünk, ha az a képlet másolásakor a másolás irányának megfelelően módosul. *Abszolút cellahivatkozás* esetén a cella címe másolásakor nem változik. Az abszolút cellahivatkozást a sor-, illetve oszlophivatkozás elé írt \$ jel jelzi, például \$G\$2.

Ha tehát példánkban a G3-as cellába az =F3*\$G\$2 képletet írjuk, akkor az másolásakor már helyes értéket ad, mivel másolásakor a \$G\$2 cellacím fog szerepelni.

Abszolút cellahivatkozás esetén a táblázatkezelő program azt tárolja, hogy a hivatkozott adat melyik cellában van, esetünkben azt, hogy a „G2-es cellában van”. Ezzel szemben relatív cellahivatkozás esetén azt tárolja, hogy az merre van az aktuális cellához képest, például hogy „az eggyel feljebb lévő cellában”.

Néha előfordul, hogy a cellacímnek csak az egyik koordinátáját kell rögzítenünk, ilyenkor **vegyes cellahivatkozást** használunk. Például a \$G2 cellahivatkozás esetén az oszlop rögzített, vízszintes másolásakor nem változik, a sor viszont nincs rögzítve, vagyis függőleges másolásakor módosul.

Feladatok

1. Egy fagylaltárus kifüggesztette az ábrán látható táblázatot, hogy a vevők azonnal lássák, mennyit kell fizetniük. A táblázatot táblázatkezelő programmal készítette el úgy, hogy később bármikor módosítani tudja egy gombóc árát. Készítsük el a táblázatot!

	A	B
1	Egy gombóc	525 Ft
2		
3	Gombóc	Ár
4	1	525 Ft
5	2	1 050 Ft
6	3	1 575 Ft
7	4	2 100 Ft
8	5	2 625 Ft

2. Egy család lakásfelújításba kezd. Ennek keretében szeretnék valamennyi helyiségben a falakat kifestetni, az ajtókat, csöveket stb. pedig mázoltatni. A család az alábbi árajánlatot kapja:

	A	B	C	D	E	F	G
1	Árajánlat			Helyiség	Festés (m²)	Mázolás (m²)	Összeg
2	Alapárak 1 m ² -re:			konyha	100	4	198 000 Ft
3	Festés:	1 800 Ft		nappali	180	8	360 000 Ft
4	Mázolás:	4 500 Ft		háló	120	8	252 000 Ft

Készítsük el az ábrán látható árajánlatot lakásunk adataival! Ügyeljünk arra, hogy ha változik a négyzetméterenkénti alapár, vagy valamelyik helyiséget rosszul mértük fel, az adatok módosíthatók legyenek! Határozzuk meg a fizetendő végösszeget is!

3. Az alábbi feladatot papíron, táblázatkezelő program használata nélkül oldjuk meg! Hogyan módosul
- a D4-es cellában lévő $=B5\$2+C2$ képlet, ha átmásoljuk a D5-ös, C4-es, D6-os, E5-ös, illetve az F6-es cellába?
 - a G5-ös cellában lévő $=B10/\$D\6 képlet, ha átmásoljuk az F5-ös, G6-os, H5-ös, G4-es, H6-os, illetve az I8-as cellákba?
 - a D4 cellában lévő $=\$B3*\$C2$ képlet, ha átmásoljuk a D5-ös, E4-es, C4-es, D3-as, E5-ös, F6-os, illetve az A1-es cellákba?

Diagramok készítése

Példa: Gyümölcsstermelés

Az alábbi ábrán hazánk gyümölcsstermelésének megoszlását látjuk az elmúlt harminc évben, ötévenkénti bontásban. Gépeljük be az adatokat, vagy nyissuk meg a könyv weblapjáról letöltött források közül a *Gyümölcsstermelés_nyers.xlsx* állományt! Formázzuk meg a táblázatot a mintának megfelelően!

Az adatok ezres tagolással jelenjenek meg. Ezt az elrendezést az adatok kijelölése után például a *Kezdőlap > Szám* dialógusdobozban a *Szám* fülön (illetve a *Formátum > Cellák > Szám* ablakban) választhatjuk ki.

	A	B	C	D	E	F	G	H
1	Hazánk 1990 és 2020 közötti gyümölcsstermelése (tonna)							
2								
3		1990	1995	2000	2005	2010	2015	2020
4	Alma	945 455	352 984	694 586	510 361	496 916	511 460	398 710
5	Körte	64 235	41 192	36 908	19 736	24 176	36 793	20 330
6	Cseresznye	26 768	20 123	18 274	6 159	5 758	11 312	8 614
7	Meggy	61 175	47 627	48 894	48 082	51 870	76 792	61 455
8	Szilva	152 319	104 938	91 285	35 956	70 854	46 023	27 038
9	Kajszi	42 325	18 102	21 408	25 655	27 059	19 855	10 623
10	Őszibarack	71 992	42 371	64 121	48 390	52 912	37 372	11 823
11	Málna	27 208	18 032	19 804	6 724	3 184	1 534	492
12	Szamóca	15 700	11 600	12 056	3 925	3 844	5 313	5 197

► Hazánk gyümölcsstermelése (Forrás: https://www.ksh.hu/stadat_files/mez/hu/mez0017.html)

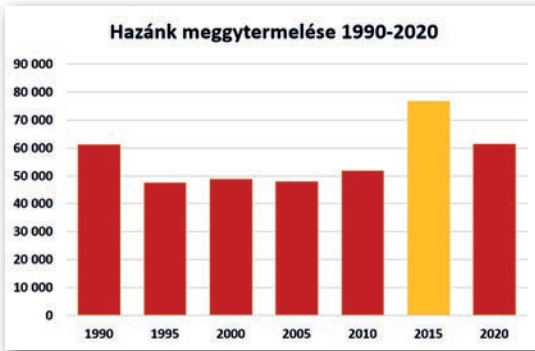
Az adatok változásának szemléltetése

Szemléltessük **oszlopdiaagramon**, hogyan változott hazánk meggystermelése az adott időszakban! A **diagramon ábrázolni kívánt adatokat úgy kell kijelölnünk, hogy a kijelölt cellák egy téglalapot alkossanak**. Ezúttal például kijelöljük egérrel az *A3:H3* tartományt, majd a CTRL gomb nyomva tartása mellett az *A7:H7* tartományt.

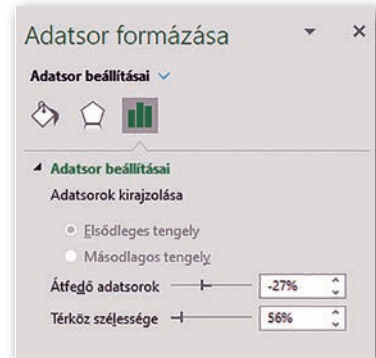
Az oszlopdiaagramot például a *Beszűrés > Diagramstílusok > Oszlopdiaagram* (illetve a *Beszűrés > Diagramtípus > Oszlopdiaagram*) paranccsal illeszthetjük be. (LibreOffice Calc alkalmazása esetén további beállítások is szükségesek lehetnek a *Diagramtündér > Adattartomány* pontjával.)

A diagram összetevőit utólag módosíthatjuk. Az ábrán például a **cím** az eredeti címre kattintva (*Meggy*) utólag kézzel írtuk át. Ugyanígy módosítottuk a cím, illetve a tengelyek feliratának **betűformátumát** feketére és félkövérre, valamint az **oszlopok színét** sötétvörösre. Az oszlopok színét és szegélyét akár egyesével is megváltoztathatjuk, például a legnagyobb értékhez tartozó oszlopot kiemelhetjük narancssárgával. A módosításokat elvégezhetjük például a kijelölt elemre kattintva a helyi menü *Adatsorok formázása* (illetve a diagramra kattintva megjelenő helyi menü *Formátum > Kijelölés*) pontjával.

Néha az **oszlopok szélességét és egymástól való távolságát** is célszerű módosítani. Ezúttal például az oszlopok szélességét megnöveltük az adatsor kijelölése után a helyi menü *Adatsor formázása* menüponttal (illetve a *Formátum > Kijelölés formázása > Beállítások* fülön).



▶ Hazánk meggytermelése 1990–2020 között oszlopdiagramon

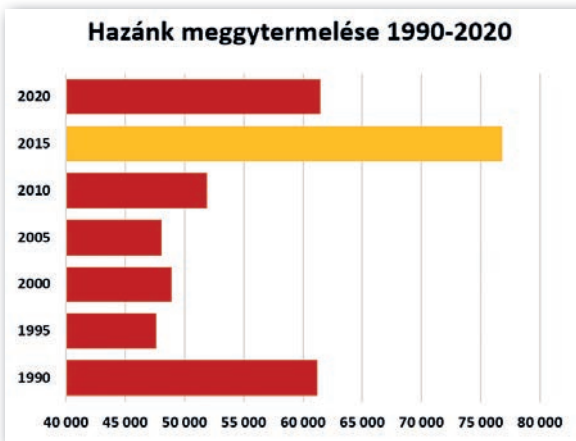


▶ Oszlopszélesség állítása (Microsoft Excel)

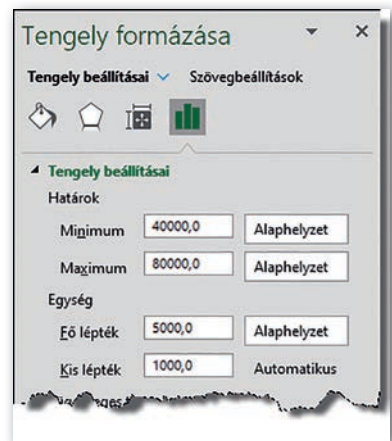
Az oszlopdiagramhoz hasonlóan szűrhatjuk be a **sávdiagramot**, ebben az esetben az adatokat szemléltető téglalapok nem vízszintesen, hanem függőlegesen követik egymást.

A diagramokon való eligazodást a **tengelyeken lévő feliratok** segítik, így ezek elrendezésére nagyon sok lehetőségünk van. Ezeket a beállításokat megtehetjük például a tengelyre kattintva a helyi menü *Tengely formázása* menüponttal. Az alábbi ábrán például a sávdiagram értéktengelyen a kezdő értéket 40 000-re állítottuk, illetve a lépésközt lecsökkentettük 5000-re.

Érdeemes az eredményt összehasonlítani az oszlopdiagrammal. Egy felületes szemlélő, aki nem figyel a tengelyfeliratokra, könnyen téves következtetést vonhat le, hiszen most a 2020-as termés ránézésre csupán a 2015-ös termés felének tűnik.



▶ Hazánk meggytermelése 1990–2020 között sávdiagramon



▶ A tengely formázása (Excel)

Az adatok arányának szemléltetése

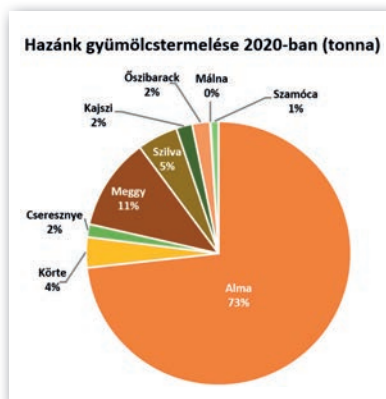
Vizsgáljuk meg, hány tonna termelt 2020-ban az egyes gyümölcsfajtákból! Ezúttal a fajtánkénti termés mennyiség egymáshoz viszonyított arányát szeretnénk szemléltetni, ezért a **kördiagramot** választjuk.

A diagram címét utólag például a *Diagramtervezés* > *Diagram-összetevő hozzáadása* (illetve *Beszűrés* > *Címek*) menüponttal vehetjük fel a diagramra.

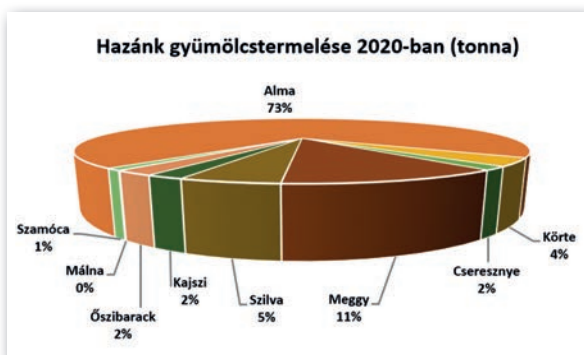
Az ábrán látható diagramon olyan elrendezést választottunk, ahol a körcikkekben megjelenik a gyümölcsfajták neve és a fajtánkénti termés százalékos aránya a teljes gyümölcsstermeléshez képest. A diagram beszűrése után ezt például megtehetjük a *Diagramtervezés* > *Kész elrendezés* (illetve a *Beszűrés* > *Adatfeliratok*) menüponttal.

Az oszlop- és a kördiagramnak egyaránt van térhatású változata. Ennek választása esetén lehetőségünk van a diagram térbeli forgatására, perspektívájának módosítására, ami sok esetben megtehető a felületes szemléltető. Az alábbi térhatású kördiagramon például az alma elveszti meghatározó arányát azzal, hogy hátraforgattuk, helyette a meggy és a szilva szelete dominál.

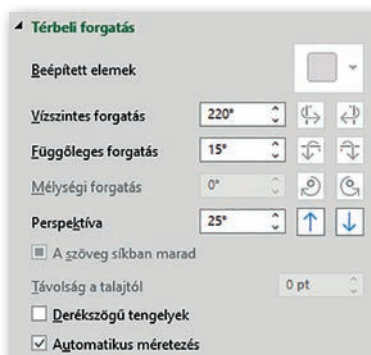
A térhatású jellemzőket beállíthatjuk például a helyi menü *Térbeli forgatás* (illetve a helyi menü *Térbeli nézet*) menüpontjával.



► Adatok ábrázolása kördiagramon



► A 2020-as adatok térhatású kördiagramon

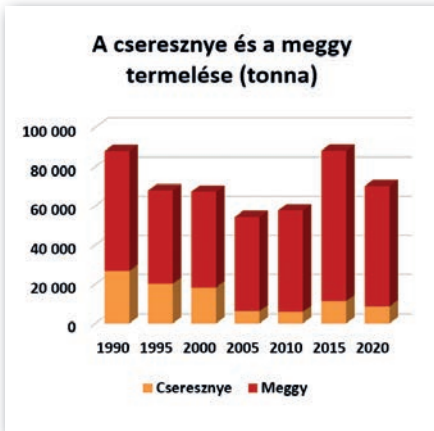


► A térbeli forgatás beállítása (Excel)

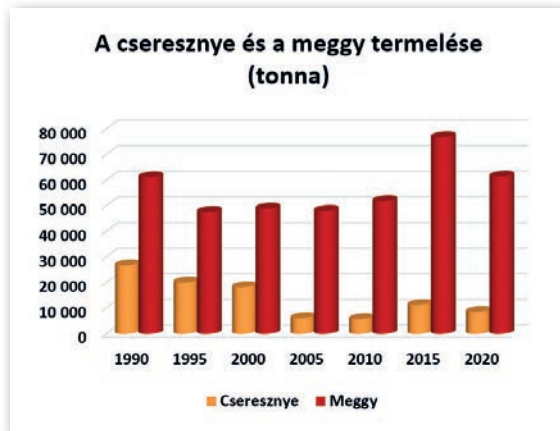
Az adatok arányának és változásának együttes ábrázolása

Következő példánkban szeretnénk ábrázolni a meggy és a cseresznye termésmennyiségének változását 1990 és 2020 között.

A kördiagram csak egyetlen adatsor ábrázolására alkalmas. Az oszlopdiagram azonban több lehetőséget is ad két adatsor együttes ábrázolására: az oszlopokat elhelyezhetjük egymás mellett (*csoportosított oszlop*) vagy egymáson (*halmozott oszlop*). Mindkét elrendezés lehetőséget ad a két adatsor összetartozó értékeinek összehasonlítására. Az alábbi ábrákon két térhatású oszlopdiagramot látunk.



► A cseresznye és a meggy termelése térhatású csoportosított oszlopdiagramon



► A cseresznye és a meggy termelése térhatású halmazott oszlopdiagramon

Grafikonok készítése

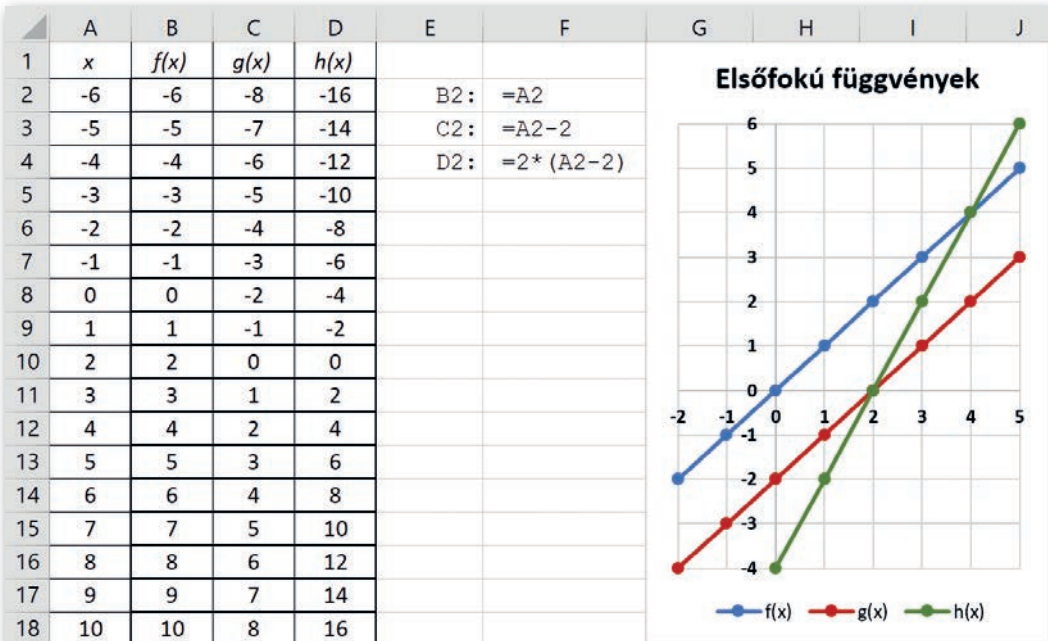
A táblázatkezelő programok segítségével könnyen szemléltethetünk matematikai és természettudományos összefüggéseket, ábrázolhatunk függvényeket.

$$f: x \mapsto x$$

$$g: x \mapsto x - 2$$

$$h: x \mapsto 2(x - 2)$$

Ábrázoljuk például közös koordináta-rendszerben a következő függvényeket:



► Az elsőfokú függvény transzformációi

Első lépésként értéktáblázatot készítünk, az alkalmazott képleteket az ábrán az F oszlopban láthatjuk. Ezt követően kerül sor az adatok – az ábrán látható példában csupán az $A6:D13$ tartomány adatainak – ábrázolására.

Grafikon beszúrására több lehetőségünk van, matematikailag azonban az adatokat **pontdiagramon** célszerű ábrázolni. A pontdiagram választása esetén megadhatjuk, hogy az adatpontokat egyenes vagy görbített vonalak kössék-e össze, megadhatjuk a vonalak és a jelölők tulajdonságait, valamint az oszlopdiagramhoz hasonlóan a tengelyek jellemzőit.

A diagram típusát attól függően választjuk meg, hogy milyen összefüggést szeretnénk szemléltetni.

Oszlopdiagramon (vagy sávdiagramon) az adatok nagyságának változását, illetve az adatok arányát egyaránt tudjuk szemléltetni. Több adatsort *halmozott* vagy *csoportosított* oszlopdiagramon hasonlíthatunk össze.

Kördiagramon (vagy tortadiagramon) az adatok egymáshoz és azok összegéhez viszonyított arányát is szemléltethetjük.

Pontdiagramon matematikai összefüggéseket vagy árfolyamváltozást ábrázolhatunk.

A diagram megválasztásánál és formázásánál ügyelnünk kell arra, hogy az lehetőleg ne tévessze meg a felületes szemlélőt sem.

Kérdések, feladatok

1. Vajon mi lehet annak az oka, hogy hazánk gyümölcstermelése a különböző években ilyen eltérő mértékű volt?
2. Ábrázoljuk a gyümölcstermelésre vonatkozó adatok felhasználásával
 - a. hazánk almatermelésének alakulását sávdiagramon;
 - b. hazánk szilvatermelésének alakulását oszlopdiagramon;
 - c. hazánk gyümölcstermelését gyümölcsfajtánként összehasonlítva kördiagramon, az ezredfordulón (2000-ben);
 - d. a kajszis- és az őszibarack együttes termését a teljes időszakban térhatású oszlopdiagramon;
 - e. valamennyi gyümölcs termelését fajtánként a teljes időszakban térhatású csoportosított oszlopdiagramon;
 - f. valamennyi gyümölcs termelését fajtánként a teljes időszakban oszlopdiagramon;
 - g. a teljes gyümölcstermést együttesen sávdiagramon!
 - h. A táblázatkezelő programok nagyon sok diagramtípust tartalmaznak. Ilyen például a perezdiagram vagy a 3D területdiagram. Próbáljuk ki ezeket a diagramokat! Ábrázoljuk a c. feladatot perezdiagramon, az f. feladatot pedig 3D területdiagramon is!



3. Az alábbi táblázatban hazánk villamosenergia-termelésének megoszlását látjuk évenként és energiahordozónként.

	A	B	C	D	E	F	G	H	I	J	K
1	Hazánk villamosenergia-termelése (gigawattóra)										
2	Megnevezés	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
3	Nukleáris	15 793	15 370	15 649	15 834	16 054	16 098	15 733	16 288	16 055	15 990
4	Szén és széntermékek	6 492	6 384	6 114	5 908	5 758	5 098	4 834	4 184	3 841	3 084
5	Földgáz	9 401	5 541	4 240	5 108	6 479	7 838	7 255	8 566	9 077	9 447
6	Kőolajtermékek	185	78	76	77	63	85	90	70	42	52
7	Biomassza	1 333	1 429	1 702	1 660	1 493	1 646	1 799	1 769	1 666	1 779
8	Biogáz	211	267	287	293	333	348	336	318	320	279
9	Kommunális hulladék megújuló része	111	136	137	208	245	160	162	137	167	149
10	Víz	213	213	301	234	259	220	222	219	244	208
11	Szél	770	718	657	693	684	758	607	729	655	651
12	Nap	8	25	67	141	244	349	629	1 497	2 450	3 793
13	Geotermikus	–	–	–	–	–	1	12	18	16	11
14	Egyéb	118	133	173	204	290	284	360	359	391	363
15	Összesen										

- Magyarország bruttó villamosenergia-termelése 2012 és 2021 között (Forrás: https://www.ksh.hu/stadat_files/ene/hu/ene0009.html)

Nyissuk meg a tankönyv weboldaláról letöltött források közül a *Villamosenergia_nyers.xlsx* állományt, és formázzuk meg a minta szerint! Határozzuk meg képlettel az *Összesen* sorban a villamosenergia-termelés éves értékeit! Az adatok felhasználásával ábrázoljuk

- hazánk villamosenergia-termelésének megoszlását 2021-ben energiahordozónként kördiagramon;
 - a napenergia-termelés változását a 2012–2021-es időszakban sávdiaagramon;
 - a szén-, földgáz- és kőolajtermékek elégetéséből származó energiát 2012-től 2021-ig csoportosított oszlopdiaagramon;
 - hazánk teljes villamosenergia-termelését a jelzett időszakban grafikonon!
4. Ábrázoljuk értéktáblázat segítségével, közös grafikonon az alábbi függvényeket!

$$f: x \mapsto x^2 \quad g: x \mapsto x^2 + 2 \quad h: x \mapsto (x - 2)^2$$

Statisztikai számítások

Példa: Magyarország megyéi

Az alábbi ábrán Magyarország megyéinek adatait látjuk a Központi Statisztikai Hivatal 2021. január 1-jén közzétett adatai alapján. (Az Alaptörvény 2022-es 11. módosítása a megyét mint közigazgatási területi egységet visszanevezte a történelmileg kialakult vármegye elnevezésre.) Az adatokat a tankönyv web-lapjáról letöltött *Megyék2021_nyers.xlsx* forrásfájlban találjuk. A fájl nem tartalmazza a világoskék háttéren látható számított adatokat, ezeket az adatokat képlettel fogjuk meghatározni.



	A	B	C	D	E	F	G	H	I	J
	Mege	Székhely	Járás	Település	Város	Népesség (fő)	Terület (km ²)	Népsűrűség (fő/km ²)		
1	Bács-Kiskun	Kecskemét	11	119	22	500 026	8 444,89	59,21		
2	Baranya	Pécs	10	301	14	356 819	4 429,66	80,55	Legnagyobb terület:	8 444,89
3	Békés	Békéscsaba	9	75	22	326 530	5 629,69	58,00	Második legnagyobb:	7247,09
4	Borsod-Abaúj-Zemplén	Miskolc	16	358	29	632 722	7 247,09	87,31	Harmadik legnagyobb:	6391,97
5	Csongrád-Csanád	Szeged	7	60	10	395 617	4 264,42	92,77		
6	Fejér	Székesfehérvár	8	108	17	418 701	4 358,57	96,06	Legkisebb népesség:	187 574
7	Győr-Moson-Sopron	Győr	7	183	12	478 281	4 207,72	113,67	Második legkisebb népesség:	213341
8	Hajdú-Bihar	Debrecen	10	82	21	526 164	6 210,77	84,72	Harmadik legkisebb népesség:	253494
9	Heves	Eger	7	121	11	291 967	3 637,17	80,27		
10	Jász-Nagykun-Szolnok	Szolnok	9	78	22	363 646	5 581,63	65,15	Magyarország népsűrűsége:	104,603
11	Komárom-Esztergom	Tatabánya	6	76	12	299 772	2 264,33	132,39	Népsűrűségek átlaga:	246,65
12	Nógrád	Salgótarján	6	131	6	187 574	2 544,46	73,72		
13	Pest	Budapest	18	187	54	1 309 802	6 391,97	204,91	Legtöbb település:	358
14	Somogy	Kaposvár	8	246	16	300 945	6 065,06	49,62	Legtöbb város:	54
15	Szabolcs-Szatmár-Bereg	Nyíregyháza	13	229	28	545 321	5 935,87	91,87		
16	Tolna	Szekszárd	6	109	11	213 341	3 703,21	57,61	Megeyék száma:	19
17	Vas	Szombathely	7	216	13	253 494	3 336,12	75,98		19
18	Veszprém	Veszprém	10	217	15	341 113	4 463,78	76,42		
19	Zala	Zalaegerszeg	6	258	10	265 101	3 783,82	70,06	100 fő/km ² -nél nagyobb:	4
20	Budapest		23	23	1	1 723 836	525,14	3282,62		
21	Összesen		197	3177	346	9 730 772	93 025,37			

► Magyarország megyéinek néhány adata (Forrás: https://www.ksh.hu/stadat_files/fol/hu/fol0007.html)

Formázzuk meg a táblázatot a mintának megfelelően! A fejlécben a kétsoros megjelenítést például a *Kezdőlap > Igazítás > Sortöréssel több sorba* (illetve a *Formátum > Cellák > Igazítás > Automatikus szövegtördelés*) menüponttal hozhatjuk létre.

A 22. sorban található adatokat a már ismert SZUM függvénnyel határozhatjuk meg, például a C22-es cellában: $=SZUM(C2 : C21)$. Mivel a népsűrűség a népesség és a terület hányadosa, ezért például a H2-es cellába az $=F2 / G2$ képlet került.

A J oszlopban statisztikai számításokat kell végeznünk. Ezeket a számításokat a táblázatkezelők függvényekkel segítik. A statisztikai függvények sok esetben – a SZUM függ-

vényhez hasonlóan – egy tartományt kapnak paraméterként. Egy adatsor maximumát például a **MAX**, minimumát a **MIN** függvény segítségével határozhatjuk meg. Így a *J3*-as cellában az `=MAX(G2:G21)`, a *J7*-esben pedig az `=MIN(F2:F21)` képlet szerepel.

A második legnagyobb, illetve a harmadik legkisebb értéket például a **NAGY** és a **KICSI** függvények segítségével kaphatjuk meg. Ezeknek a függvényeknek két paraméterük van: az első az adatsort tartalmazó tartományt adja meg, a második pedig, hogy hányadik értékre vagyunk kíváncsiak. Ennek megfelelően a *J4*-es cellába az `=NAGY(G2:G21; 2)`, míg a *J9*-es cellába az `=KICSI(F2:F21; 3)` képletet írtuk.

Magyarország népsűrűsége a *J11*-es cellában Magyarország népességének és területének átlaga: `=F22/G22`. Ez az adat eltér a megyék népsűrűségének átlagától, amelyet az **ÁTLAG** függvénnyel határozhatunk meg (a *J12*-es cellában: `=ÁTLAG(H2:H21)`). Érdemes meggondolnunk, hogy miért értelmetlen a megyék népsűrűségének átlagát jellemzőként használni.

A darabszámok meghatározására több függvény is a rendelkezésünkre áll. A **DARAB** függvény a számokat tartalmazó cellák számát, a **DARAB2** pedig a nem üres cellák számát adja vissza. Így tehát a megyék számát a *J17:J18* tartományban megkaphatjuk akár az `=DARAB(C2:C20)`, akár az `=DARAB2(B2:B20)` képlettel (Budapestet nem sorolják a megyék közé).

Végül egy érdekes számítás: a *J20*-as cellában arra vagyunk kíváncsiak, hogy a *H2:H21*-es tartományban hány 100-nál nagyobb érték van. Erre egy összetettebb függvényt használhatunk, amelynek első paramétere a tartomány, második a feltétel:

`=DARABTELI(H2:H21; ">"100)`.

A **DARABTELI** függvényben a feltétel bonyolultabb is lehet. Megvizsgálhatjuk például, hogy van-e olyan megye, amelynek nagyobb a népsűrűsége, mint Magyarországé. Ekkor a megoldást az

`=DARABTELI(H2:H21; ">"&J11)`

képlet adja. A feltételben a relációs jel szöveges adat, ezért idézőjelek közé kell tenni. Ezt a részt egy `&` jel fűzi a cellahivatkozáshoz.

A leggyakoribb függvények

SZUM (tartomány)	A tartomány elemeinek összege
ÁTLAG (tartomány)	A tartományban lévő számok átlaga
MAX (tartomány)	A tartomány legnagyobb értéke
MIN (tartomány)	A tartomány legkisebb értéke
DARAB (tartomány)	A tartományban lévő számok száma
DARAB2 (tartomány)	A tartomány nem üres celláinak száma
NAGY (tartomány; k)	A tartomány k-adik legnagyobb eleme
KICSI (tartomány; k)	A tartomány k-adik legkisebb eleme

Feladatok

1. Az iskola Diákönkormányzata a Föld napja alkalmából PET-palackok kupakjait gyűjtő versenyt hirdet Kupak Kupa néven a felső tagozatosok részére. Az eredményt az alábbi táblázat tartalmazza. Gépeljük be az adatokat, vagy másoljuk be a könyv weblapjáról letöltött *Kupak Kupa_nyers.xlsx* fájlból!

	A	B	C	D	E	F	G	H
1	Osztály	kg	fő	egy főre			osztályokra	egy főre
2	5.a	25,20	20			Teljes mennyiség (kg):	216,63	
3	5.b	21,28	19			Legtöbb kupak (kg):	35,42	
4	5.c	29,45	19			Második legtöbb (kg):	34,02	
5	6.a	14,30	26			Harmadik legtöbb (kg):	29,45	
6	6.b	9,18	27			Legkevesebb kupak (kg):	9,18	
7	7.a	34,02	21			Egy osztályra jutó átlag (kg):	24,07	
8	7.b	26,22	23			Osztályok száma:	9	
9	8.a	21,56	22			30 kg-nál többet gyűjtött:	2	
10	8.b	35,42	23					

- Határozzuk meg a tanult függvények segítségével a G oszlopban a kupakgyűjtés néhány statisztikai adatát!
 - Formázzuk meg a táblázatot a mintának megfelelően!
 - Egyes osztályok szerint az adatok – az osztályok eltérő létszáma miatt – nem fejezik ki a tanulók tényleges hozzáállását. Határozzuk meg ezért a D oszlopban, hogy mennyi kupak jut egy-egy tanulóra osztályonként! Az eredményként kapott adatok alapján határozzuk meg a megfelelő értékeket a H oszlopban!
 - Ábrázoljuk mindkét esetben az adatokat diagramon! Melyik esetben van értelme kördiagramot használni? Válaszunkat indokoljuk!
2. Az iskolában nagyon sok tanuló tanul zenét, ezért az eredmények bemutatására az iskolában működő zeneiskola minden évben tehetségkutató gálát hirdet. A gálán egy tíztagú zsűri 1-től 30 pontig terjedő skálán értékeli a növendékek teljesítményét. A leadott pontszámokból azonban levonják a legnagyobb és a legkisebb értéket, továbbá az előadó rokonai, tanárai és osztálytársai sem pontozhatnak, így a táblázatban az ő celláik üresek. Az adatokat a tankönyv weblapjáról letöltött *Tehetségkutató_nyers.xlsx* fájlban találjuk.
- A fentieknek megfelelően töltsük ki képlettel az L:Q oszlopokban lévő cellákat!
 - Az egyes zsűritagok szigorúságát határozzuk meg a B14:K14 tartományban az általuk adott pontszámok átlagával!
 - Formázzuk meg a táblázatot a mintának megfelelően, továbbá ábrázoljuk a tanulók eredményét általunk megválasztott diagramon!

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Előadó	ZS1	ZS2	ZS3	ZS4	ZS5	ZS6	ZS7	ZS8	ZS9	ZS10	Össz	Max	Min	Pont	Fő	Pontszám
2	Békési Szervác	21	29		20	23	28	23	22	24	30	220	30	20	170	7	24,29
3	Kutassy Henrik	28	19	20	22	21	19	24	26	18	30	227	30	18	179	8	22,38
4	Békési Paula	18	19	20	18			25	23	26	26	175	26	18	131	6	21,83
5	Bimbó Inéz	20	23	18	18	18	30	21	18	24	19	209	30	18	161	8	20,13
6	Chlebovics Maja	30	28	29	24	29	22	22			25	209	30	22	157	6	26,17
7	Gera Johanna	30	15		23	19	30		15	17	27	176	30	15	131	6	21,83
8	Kriston Borbála	21	25	17	14	24	26	25	16	11	22	201	26	11	164	8	20,50
9	Spitczér Titanilla	26	16	26	23	17		25	28	22	16	199	28	16	155	7	22,14
10	Geisler Katrina	26	23	22	20	16	30	25	22	25	17	226	30	16	180	8	22,50
11	Possert Armanda	14	26	27		16	25	16	12	27	18	181	27	12	142	7	20,29
12	Bobrovski Árpád	25	15	15	16	28	27	21		25	20	192	28	15	149	7	21,29
13	Kávai Valér	27	23	26	19		25	19	17	23	20	199	27	17	155	7	22,14
14	Átlag	23,83	21,75	22,00	19,73	21,10	26,20	22,36	19,90	22,00	22,50						

3. Az alábbi példában összefoglaljuk a statisztikai függvényekről tanult ismereteinket. Számítógép alkalmazása nélkül adjuk meg, hogy milyen képletekkel kell válaszolni az F oszlopban olvasható kérdésekre a G oszlopban! Milyen másolható képletet kell írunk a D oszlopba?

	A	B	C	D	E	F	G
1	Osztálylétszámok:					Összes tanuló:	
2						Fiúk száma:	
3	osztály	létszám	fiú	lány		Az "a" osztályba járók száma:	
4	5.a	20	10			Átlagos osztálylétszám:	
5	5.b	19	11			Fiúk átlagos száma:	
6	5.c	19	10			Legnagyobb osztálylétszám:	
7	6.a	26	14			Második legnagyobb osztálylétszám:	
8	6.b	27	13			Legkisebb osztálylétszám:	
9	7.a	21	10			Második legkisebb osztálylétszám:	
10	7.b	23	12			Legkisebb fiúlétszám:	
11	8.a	22	10			Osztályok száma:	
12	8.b	23	11			20 főnél nagyobb létszámú osztályok száma:	
13						Átlagosnál kisebb osztálylétszámok száma:	

4. A feladatokban szereplő statisztikai függvények nem függetlenek egymástól. A Kupak Kupa adatait elemző feladatban például a $G3$ -as cella tartalmát megadhatjuk a MAX és a $NAGY$ függvénnyel is. Hogyan? Milyen további példákat tudunk felsorolni, amelyekkel az egyes statisztikai függvényeket más statisztikai függvényekkel helyettesíthetjük?
5. Az alábbi táblázatban Miskolc meteorológiai adatait találjuk havi bontásban. A táblázat a napi középhőmérsékletek átlagát, az adott hónapban lehullott csapadék mennyiségét és a napfénytartamot, azaz a napsütéses órák számát tartalmazza. (A napi középhőmérséklet az 1 óraker, a 7 óraker, a 13 óraker és a 19 óraker mért hőmérsékletértékek átlaga).

	A	B	C	D	E	F	G
1	Miskolc éghajlati jellemzői						
2							
3	Hónapok	Közép-hőmérséklet [°C]	Csapadék [mm]	Napfénytartam [óra]			
4	január	-1,9	25	63		Legmagasabb középhőmérséklet:	
5	február	-0,2	30	91		Legalacsonyabb középhőmérséklet:	
6	március	5,1	30	142		Legtöbb csapadék:	
7	április	11,0	48	190		Második legtöbb csapadék:	
8	május	16,0	64	241		Legkevesebb csapadék:	
9	június	19,1	78	245		Második legkevesebb csapadék:	
10	július	21,0	76	267		Átlagos havi csapadék:	
11	augusztus	20,4	69	261		Napsütéses órák éves összege:	
12	szeptember	15,7	54	183			
13	október	10,2	39	142			
14	november	4,1	41	72		Fagyos hónapok száma:	
15	december	-0,9	39	49		Napos hónapok száma:	

► Forrás: www.met.hu/eghajlat/magyarorszag_eghajlata/varosok_jellemzoi/Miskolc/

- Adjuk meg képlet segítségével a $G4:G11$ tartomány megfelelő celláiban az F oszlopban kért adatokat!
- Hány hónapban volt a középhőmérséklet $0\text{ }^{\circ}\text{C}$ fok alatt? Hány hónapban volt a napsütéses órák száma 200-nál több? A választ képlet segítségével adjuk meg a $G14:G15$ tartomány celláiban!
- Formázzuk meg a táblázatot a minta szerint!
- Ábrázoljuk kördiagramon az éves napfénytartam havonkénti megoszlását!
- Ábrázoljuk oszlopdiagramon az éves lehullott csapadék havonkénti megoszlását!

Számformátumok

A táblázatkezelő programok a számokat egyféleképpen tárolják, ám többféleképpen képesek megjeleníteni: pénznemként, százalékként, dátumként stb. Ezekkel a lehetőségekkel korábban már találkoztunk, de most alaposabban körbejárjuk őket.

Példa: Bruttó és nettó bér

A minimálbér a kötelező legkisebb munkabér havi összege, egy napi nyolc órában dolgozó munkavállaló legalább ennyit keres. Összegét minden évben jogszabály határozza meg, például 2022. január 1-jén 200 000 Ft volt.

Ez az összeg a *bruttó bér*, ennyi szerepel a munkaszerződésben. Ténylegesen azonban mindenkinek hozzá kell járulnia a közterhekhez, azaz bizonyos kiadások közös viseléséhez. Ilyen a *személyi jövedelemadó*, amelynek befizetésével az állami feladatok ellátását segítjük, illetve a *társadalombiztosítási járulék*, amellyel a nyugdíjak kifizetéséhez és az egészségügyi rendszer fenntartásához járunk hozzá.

2022-ben a személyi jövedelemadó a bruttó bér 15 százaléka, a társadalombiztosítási járulék pedig a bruttó bér 18,5 százaléka volt. Készítsünk táblázatot, amelynek segítségével meghatározhatjuk, hogy egy adott bruttó bér esetén mennyit fizetünk erre a két közteherre, illetve számoljuk ki, hogy a végén mennyi pénzünk marad (*nettó bér*)!

Megjegyzés: Bizonyos feltételek teljesülése esetén a személyi jövedelemadó fizetése alól részlegesen vagy teljes mentesülést kaphatunk.

	A	B	C	D
1	A minimálbér után fizetendő közterhek 2022-ben			
2				
3	Bruttó minimálbér		200 000 Ft	
4	Személyi jövedelemadó	15,00%	30 000 Ft	=C3*B4
5	Társadalombiztosítási járulék	18,50%	37 000 Ft	=C3*B5
6	Nettó minimálbér		133 000 Ft	=C3-C4-C5

► A minimálbér után fizetendő közterhek 2022-ben. Hogyan alakulnak ezek az összegek ma?

Készítsük el a táblázatot a mintának megfelelően, és formázzuk meg!

A bruttó minimálbért **pénznem formátumban**, forintban adjuk meg, és ebben a formában írjuk be: a szám után egy szóközt teszünk, ezt követi a forint rövidítése: Ft. Hasonló módon, egy szóközzel elválasztva írhatjuk be az eurót (€), amelynek jelét az ALT GR és az U billentyűk együttes lenyomásával vihetjük be.

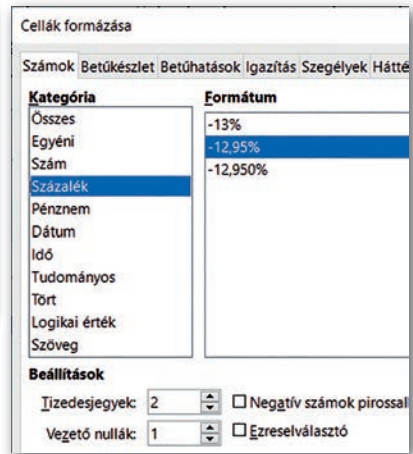
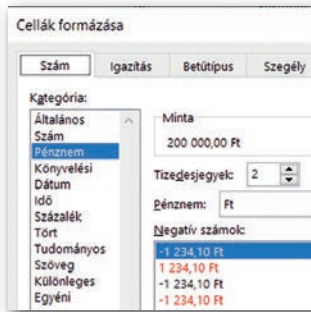
A B oszlopban a számok **százalék formátumban** szerepelnek. A % jel nem mértékegység, hanem a „század” nevezőt helyettesíti, ennek megfelelően 15% = 0,15.

A C4:C6 tartományban képleteket alkalmaztunk, ezek az ábrán a D oszlopban szerepelnek. A képletek beírása után a cellákban a számok nem feltétlenül a mintának megfelelő számú tizedesjeggyel jelennek meg: a tizedesjegyek számát a (illetve a) ikonokkal növelhetjük, illetve csökkenthetjük.

Mint láttuk, a táblázatkezelő programok a számokat egyformán tárolják, a számformátumok beállításával „csupán” különböző módon jelennek meg. Vajon mi a tárolt szám az egyes esetekben? Ezt megtudhatjuk, ha a számformátumokat töröljük, vagyis az *Általános számformátumot* alkalmazzuk a teljes táblázatra a *Kezdőlap* > *Szám* listájában (illetve a *Formátum* > *Számformátum* menüpontjában).

Jól látható, hogy ekkor a 15% valóban 0,15-ként jelenik meg, valamint az is, hogy a 200 000 Ft vagy 200 000 € egyaránt 200 000 lesz.

	B	C
Fizetendő közterhek 2022-ben		
		200000
Idő	0,15	30000
Járulék	0,185	37000
		133000



► Az előző ábra adatai Általános számformátumban

► A PéNZnem formátum beállításai (Microsoft Excel)

► A Százalék formátum beállításai (LibreOffice Calc)

A közös színeként beírt számokat utólag a megfelelő formátumra alakíthatjuk, például a *Kezdőlap* > *Szám* csoportjában a dialógusdoboz megjelenítésével (illetve a *Formátum* > *Cellák* > *Számok* fülön). A megjelenő párbeszédpanel általában további beállításokat is felkínál, például megadhatjuk a tizedesjegyek számát, pénznem esetén a pénzformátumot, valamint azt is, hogy a negatív számok eltérő színnel jelenjenek-e meg.

Példa: Biciklikölcsönző

Egy tóparti vállalkozás kerékpárokat kölcsönöz a tó környéki kerékpártúrákhoz és a helyi látóivalok megtekintéséhez.

A kölcsönzéseket a tulajdonos táblázatkezelővel tekinti át. Külön táblázatban tartja nyilván a többnapos kerékpártúrákat (ilyenkor a kölcsönbe vevő napdíját fizet), illetve az egy napos kölcsönzéseket (ilyenkor a kölcsönbe vevő óradíját fizet).

A bérleti díj a kerékpár típusától függ (például túra-, tandem- vagy gyermekkerékpár, illetve elektromos kerékpár).

A többnapos kölcsönzéseket tartalmazó táblázat tartalmazza a kölcsönző nevét, a kölcsönzés első és utolsó napját, a napok számát, a napdíját, valamint a kölcsönzés után fizetendő összeget. A táblázatot az alábbi ábrán látjuk, a nyers fájlt a tankönyv weblapjáról letöltött *Kölcsönző_nyers.xlsx* forrásfájl tartalmazza. A zöld háttérű mezőket (napok száma, kölcsönzési díj) képlettel fogjuk meghatározni.

	A	B	C	D	E	F
1	Többnapos kölcsönzés					
2	Kölcsönző	Kezdet	Vége	Napok	Egységár	Fizetendő
3	Halász Timót	2022.07.01	2022.07.02	2	3 000 Ft	6 000 Ft
4	Gerencsér Dávid	2022.07.01	2022.07.01	1	3 000 Ft	3 000 Ft
5	Gerencsér Villó	2022.07.01	2022.07.01	1	2 500 Ft	2 500 Ft
6	Fabinyi Vazul	2022.07.01	2022.07.04	4	5 000 Ft	20 000 Ft

► Többnapos kölcsönzés nyilvántartása

A táblázatkezelő programok a dátum formátumú számok esetén az 1900. január 1. óta eltelt napok számát tárolják, de a megszokott módon jelenítik meg őket. Az egység az egy nap, tehát két dátum különbsége a közöttük eltelt napok száma. Például $2022.07.02 - 2022.07.01 = 1$, mivel a két dátum között 1 nap telik el. Kölcsönzés esetén az elvitel és a visszahozatal napját is beszámítják, a különbséget tehát 1-gyel meg kell növelni, így a D2-es cellába az $=C3-C2+1$ képlet kerül.

	A	B
1	1900.01.01	1
2	1900.01.02	2
3	1900.01.31	31
4	1900.02.01	32
5	1900.12.31	366
6	1901.01.01	367
7	2000.01.01	36526

► Dátumok-számok

A dátumok kezelése alapján tehát 1900.01.01. 1-nek felel meg, 1900.01.02. 2-nek, 1900.02.01. 32-nek, 2000.01.01. pedig 36 526-nak. A dátumot a *Kezdőlap* > *Szám* csoportjában a dialógusdoboz megjelenítésével (illetve a *Formátum* > *Cellák* > *Számok* fülön) az *Általános* számformátum választásával alakíthatjuk „számmá”, és ugyanitt a *Dátum* lehetőséggel alakíthatjuk vissza dátum formátumúvá. A dátumformátumok között többféle megjelenítés között választhatunk, például megjeleníthetjük a hónap nevét, sorszámát vagy rövidítését, megadhatjuk, hogy az elválasztójel pont vagy kötőjel legyen-e, stb.

Vajon alkalmas-e ez a dátumkezelés arra, hogy egy adott napon belül az időpontot is megadjuk? Például milyen szám felel meg 2000.01.01 12:00-nak? A táblázatkezelők egy adott napon belül az időpontot a nap törtrészeként kezelik. Példánkban délig a nap fele telik el, így a válasz 36526,5. Az átalakítást a táblázatkezelők megjelenítéskor automatikusan elvégzik, így a fenti időpont esetén a megfelelő cellában 2000.01.01 12:00 jelenik meg.

	A	B
1	2000.01.01 0:00	36526,00000
2	2000.01.01 8:00	36526,33333
3	2000.01.01 12:00	36526,50000
4	2000.01.01 16:00	36526,66667
5	2000.01.01 20:00	36526,83333
6	2000.01.01 23:59	36526,99931

► Az időkezelés egysége a nap

	A	B	C	D	E	F
1	Néhány órás kölcsönzés					
2	Kölcsönző	El	Vissza	Óradíj	Óra	Fizetendő
3	Zelics Manfréd	8:00	14:00	400 Ft	6,00	2 400 Ft
4	Erdős Sámuel	9:12	11:44	450 Ft	2,53	1 140 Ft
5	Erdős Ubul	9:12	11:44	450 Ft	2,53	1 140 Ft
6	Buzási Tamás	9:15	14:50	400 Ft	5,58	2 233 Ft
7	Molnár August	9:47	14:00	500 Ft	4,22	2 108 Ft

► Néhány órás kölcsönzés nyilvántartása

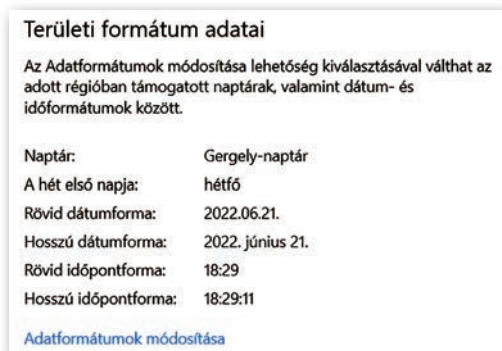
Ha a biciklikölcsönzés egy napnál rövidebb, az egyes kerékpárok után óradíjat számolnak fel. Ebben az esetben a táblázat tartalmazza a kölcsönbe vevő nevét, az elvitel és a visszahozatal időpontját, az óradíjat, valamint a fizetendő összeget. Az időpontot többnyire *óra:perc* formátumban írhatjuk be, például 9:12 vagy 11:44. (A nyers táblázat a könyv weboldaláról letöltött *Kölcsönző_nyers.xlsx* fájl második munkalapján található.)

Vajon milyen képlet kerül az E3-as cellába? Két időpont különbsége ($C3-B3$) az eltelt időtartamot az alapértelmezett nap egységben adja meg, például 8:00-tól 14:00-ig a nap egynegyede (6 óra / 24 óra) telik el, így $14:00 - 8:00 = 0,25$.

A fizetendő összeget azonban ezúttal nem egy napra, hanem egy órára tartalmazza a D oszlop. Egy nap 24 óra, tehát az E3 cellába az $=(C3-B3) * 24$ képletet kell írunk. A célformátumot az ábrán két tizedesjegy pontosságú megjelenítésre állítottuk.

Vajon honnan tudják a táblázatkezelő programok, hogy az adott országban milyen dátumformát kell alkalmazniuk? Ezeket a beállításokat az operációs rendszer beállításából kérdezik le, amelyeket például Windows 10 esetén a *Start menüből* a *Gépház > Idő és nyelv* beállításainál érünk el.

A táblázatkezelő programok további számformátumokat is ismernek, sőt, akár magunk is készíthetünk egyéni formátumokat. Gyakran találkozhatunk például a normálalaknak megfelelő **tudományos számformátummal** vagy a közönséges törtet kezelő **törtformátummal**.



► Regionális beállítások (Windows 10)

Feladatok

- Magyarországon 1870 óta tízévenként tartanak népszámlálást. A hazánk jelenlegi területére vonatkozó adatokat a tankönyv weblapjáról letöltött *Népszámlálási adatok_nyers.xlsx* fájlban találjuk. Az *A* oszlop a népszámlálás évét, a *B* oszlop a népességet tartalmazza az adott évben. Határozzuk meg képlet segítségével
 - a *C* oszlopban, hogy hány fővel változott a népesség az előző népszámláláshoz képest;
 - a *D* oszlopban, hogy a népesség hány százalék volt az előző népszámláláshoz képest;
 - az *E* oszlopban, hogy a népesség hány százalék volt az 1870-es népszámláláshoz képest;
 - az *F* oszlopban Magyarország népsűrűségét az adott évben, ha hazánk területe 93 026,33 km²!
- Formázzuk meg a táblázatot a mintának megfelelően, és szemléltessük az adatokat megfelelően választott diagramon!

	A	B	C	D	E	F
1	Hazánk népessége a népszámlálási adatok alapján					
2	Év	Népesség	Változás	Előzőhöz képest	1870-hez képest	Népsűrűség fő/km²
3	1870	5 011 310	–		100%	53,90
4	1880	5 329 191	317 881	106%	106%	57,30
5	1890	6 009 351	680 160	113%	120%	64,60
6	1900	6 854 415	794 376	113%	136%	73,70
7	1910	7 612 114	757 699	111%	152%	81,80

► Forrás: https://www.ksh.hu/nepszamlalas/tablak_területi_00

2. A következő ábrán egy szálloda újévi forgalmának néhány sorát látjuk. Készítsük el és formázzuk meg a táblázatot a mintának megfelelően! A sárga háttéren lévő adatokat képlet segítségével határozzuk meg!
- A *D* oszlopban (*Vendégéj*) határozzuk meg, hogy hány éjszakát töltött az adott vendég a szállodában!
 - Az *E* oszlopban a szoba egy éjszakára vonatkozó ára szerepel. Határozzuk meg ennek felhasználásával az *F* oszlopban a fizetendő összeget forintban!
 - Néhány külföldi szívesebben fizet euróban. Írjuk be a *G1*-es cellába az euró árfolyamát, és fejezzük ki a *G* oszlopban a fizetendő összeget euróban! (A napi euró/Ft árfolyamot például a Nemzeti Bank honlapján nézhetjük meg.)

	A	B	C	D	E	F	G
1	Szállásdíj					1 euró: 388 Ft	
2	Szoba	Érkezett	Távozott	Vendégéj	Szobaár	Fizetendő (Ft)	Fizetendő (€)
3	1012	2022.12.28	2023.01.02	5	49 500 Ft	247 500 Ft	637,89 €
4	1016	2022.12.28	2023.01.14	17	37 000 Ft	629 000 Ft	1 621,13 €
5	1017	2022.12.30	2023.01.02	3	37 000 Ft	111 000 Ft	286,08 €
6	6066	2022.12.30	2023.01.06	7	16 000 Ft	112 000 Ft	288,66 €

3. Módosítsuk a *néhány órás kölcsönzéseket* tartalmazó táblázatot úgy, hogy minden megkezdett óra egész órának számítsen! (A megoldáshoz használhatjuk például a KERÉKÍ-TÉS függvényt. Keressük meg a *Súgó*ban ennek a függvénynek a használatát!)

Kétirányú elágazás

Programozási tanulmányaink során már megismerkedtünk az elágazással. Ebben a vezérlési szerkezetben egy adott feltételtől függően hoz döntést a program, hogy az egyik vagy a másik megoldást válassza-e. Ez a lehetőség a táblázatkezelő programokban is rendelkezésünkre áll, a magyar nyelvű táblázatkezelőkben az ilyen feladatokat a **HA** függvény segítségével oldhatjuk meg.

Példa: Papírgyűjtési verseny

Az alábbi táblázat a papírgyűjtési verseny eredményét tartalmazza osztályonként. A *B* oszlopban szerepel, hogy az adott osztály hány kilogramm papírt gyűjtött tanulónként. A verseny minden résztvevőjét díjazták: amelyik osztály 50 kg/fő-nél több papírt gyűjtött, az egy egynapos kirándulást kap; amelyik annál kevesebbet, az egy tortát. Készítsük el a táblázatot az alábbi mintának megfelelően! Ügyeljünk arra, hogy a *B* oszlopban az adatok két tizedesjegy pontossággal jelenjenek meg! Az adatokat a tankönyv weblapjáról letöltött *Papír_nyers.xlsx* állományban találjuk.

Határozzuk meg a táblázat *C* oszlopában, hogy melyik osztály kapott tortát, és melyik kirándulást! Az iskola alapítványa a 60 kg/fő-nél több papírt gyűjtött osztályok kirándulását 100 000 Ft-tal támogatja. Jelenjen meg a támogatott osztályoknál ez az összeg a *D* oszlop megfelelő celláiban!

	A	B	C	D	E	F	G
1	Osztály	kg/fő	jutalom	támogatás			
2	1.a	69,12	kirándulás	100 000 Ft		Legtöbb:	
3	1.b	49,99	torta			Legkevesebb:	
4	2.a	50,01	kirándulás				
5	2.b	62,56	kirándulás	100 000 Ft		Osztályok száma:	
6	3.a	21,00	torta			Kirándul:	
7	3.b	54,80	kirándulás			Tortát kap:	
8	4.a	66,30	kirándulás	100 000 Ft			
9	4.b	22,12	torta			Összes támogatás:	
10	5.a	47,20	torta				

► Az iskolai papírgyűjtés eredménye és összesítése

A táblázat *C2*-es cellájában a versenykiírásnak megfelelően a következő képlet szerepel:

```
=HA (B2>50; "kirándulás"; "torta")
```

A **HA** függvénynek három paramétere van. Első paramétere tartalmazza a feltételt, ez példánkban a $B2 > 50$ kifejezés. Második paramétere azt, hogy mi kerüljön a cellába, ha a feltétel igaz (*igaz ág*), harmadik paramétere pedig, hogy mi legyen a cella értéke, ha a feltétel hamis (*hamis ág*). Mivel $69,12 > 50$, ezért ezúttal a cella tartalma a "kirándulás" szó lesz, de a *C3*-as cellában már a "torta" szerepel.

Hasonló módon a *D2*-es cellába a következő képlet kerül:

```
=HA (B2>60; 100000; "")
```

Ezúttal abban az esetben, ha a feltétel hamis, a cellában ne jelenjen meg semmi, s ehhez a hamis ágban két egymás mellett szereplő macskaköröm szükséges: "".

Példa: Abszolút érték

Az abszolút érték fogalmával matematikaórán már találkozunk. Ha a szám negatív, akkor abszolút értéke a szám ellentettje (-1 -szerese), egyébként önmaga.

Például az A2-es cellában szereplő szám abszolút értékét a B2-es cellában a következő képlettel adhatjuk meg:

$$=HA(A2<0; -A2; A2)$$

Ebben a példában mind az igaz, mind a hamis ág egy-egy kifejezést tartalmaz. A HA függvény tehát feltételtől függő számítások esetén is használható.

Megjegyzés: Egy szám abszolút értékét az ABS függvény segítségével is megkaphatjuk.

	A	B
1	x	x
2	-3	3
3	3,14159	3,14159
4	0	0

► Abszolút érték

Feladatok

- Végezzük el a papírgyűjtési verseny adatait tartalmazó táblázat G oszlopában az alábbi statisztikai számításokat! Határozzuk meg
 - a G2-es cellában a legtöbb, a G3-as cellában pedig a legkevesebb egy főre jutó összegűjtött papír mennyiségét;
 - a G5-ös cellában az osztályok számát;
 - a G6-os és G7-es cellában, hogy hány osztály kap kirándulást, illetve tortát;
 - a G9-es cellában az alapítványi támogatás teljes összegét!Ábrázoljuk az adatokat egy, a feladat szövegéhez illeszkedő diagramon!
- Egy önkormányzat szeretné bevezetni az ebadót. Az összeg a kutya tömegétől függene, ám a településen működő pártok többféle eltérő megoldást javasolnak. Az alábbi ábrán mintaként szerepel néhány adat. Készítsük el a táblázatot, és határozzuk meg az egyes elképzelések szerint fizetendő adó összegét!
A képletet a párt nevének megfelelő oszlopba írjuk!

	A	B	C	D	E	F	G
1	Tulajdonos	Eb	Tömeg (kg)	D párt	E párt	F párt	G párt
2	Somos Irma	Bogánics	28	4 000 Ft	2 800 Ft	2 800 Ft	3 800 Ft
3	Hármori Jakab	Sanyi	16	2 000 Ft	2 000 Ft	1 600 Ft	1 800 Ft
4	Csellós Gábor	Vahur	44	4 000 Ft	4 400 Ft	4 000 Ft	5 400 Ft
5	Imre Katalin	Tappancs	3	2 000 Ft	2 000 Ft	300 Ft	1 150 Ft
6	Juhász Zoltán	Csuti	56	4 000 Ft	5 600 Ft	4 000 Ft	6 600 Ft

- A D párt szerint az ebadó mértéke a 20 kg-nál nagyobb tömegű kutyák után legyen 4000 Ft, egyébként 2000 Ft.
 - Az E párt szerint 20 kg-nál kisebb kutyák után az adó legyen 2000 Ft, azonban a 20 kg-nál nagyobb kutyák után kilogrammonként 100 Ft-ot kelljen fizetni.
 - Az F párt a nagy testű kutyákat támogatná. Szerintük az adó legyen kilogrammonként 100 Ft, de 40 kg fölött már egységesen csak 4000 Ft.
 - A G párt javaslata kicsit összetettebb. Egységesen minden kutyatulajdonos fizessen 1000 Ft minimáladót, és azon túl még kilogrammonként 50 Ft-ot, ha a kutya 20 kg-nál kisebb, egyébként pedig kilogrammonként 100 Ft-ot.
- Készítsünk képletet, amely a papírgyűjtési verseny esetén a *Jutalom* oszlopban háromféle kategóriát alkalmaz: 50 kg/fő fölött egynapos kirándulás jár, de ennél kisebb mennyiség esetén a torta csak akkor, ha az összegűjtött papír mennyisége legalább 25 kg/fő!

Gyakorlófeladatok

1. Az alábbi táblázat a 8. osztály tanév végi tanulmányi eredményeit tartalmazza. Az adatokat a tankönyv weblapjáról letöltött *Osztálystatisztika_nyers.xlsx* forrásállomány tartalmazza. A szürke háttérű cellák adatait képlettel kell meghatároznunk.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
1	Osztálystatisztika (8.a)																		
2	Név	Magyar	Angol nyelv	Matematika	Történelem	Állampolgári ism.	Etika	Természettudomány	Földrajz	Ének-zene	Vizuális kultúra	Digitális kultúra	Testnevelés	Átlag	Igazolt	Igazolatlan	4,5 fölött	Elégtelent szerzett	
3	Ábrahám Borbála	4	5	4	4	5	5	4	4	5	4	4	4	4,33	16	0			
4	Bajai Valter	4	4	5	4	5	5	5	5	5	4	5	4	4,58	55	1	x		
5	Bárkányi Jónás	4	3	5	4	4	5	3	4	4	4	3	5	4,00	78	0			
6	Bodzás Oleg	5	4	1	4	5	5	3	4	4	5	4	5	4,08	12	0		!!!	
7	Délczeg Jadviga	4	3	5	5	5	5	4	5	3	4	5	4	4,33	98	1			
25	Völgyesi Kristóf	4	4	3	4	4	5	4	2	3	3	4	5	3,75	24	1			
26	Vezér Franciska	4	5	4	3	5	5	4	3	5	5	4	5	4,33	48	1			
27	Tantárgyi átlag	4,25	4,08	3,96	3,83	4,63	4,79	3,67	3,88	4,46	4,21	4,17	4,55						
28	Összesen													1096	31	8	4		
29	1 főre													45,67	1,29				
31	Tanulók száma:	24						Legjobb átlag:	5					Legrosszabb átlag:	2,75				
32	Osztályátlag:	4,20						Második legjobb:	4,9					Második legrosszabb:	3,18				
33	Bukások száma:	5						Harmadik legjobb:	4,8					Harmadik legrosszabb:	3,58				

- Számoljuk ki függvény alkalmazásával a tanulók átlagait az N oszlopban, illetve a tantárgyi átlagokat a 27. sorban!
- Határozzuk meg függvény segítségével a tanulók számát a $B31$ -es cellában!
- Az osztályátlag valamennyi osztályzat átlaga. Határozzuk meg az osztályátlagot a $B32$ -es cellában!
- Melyik a legjobb, második legjobb, harmadik legjobb; illetve legrosszabb, második legrosszabb, harmadik legrosszabb átlag? A kérdésekre a $H31:H33$, illetve az $O31:O33$ tartományban függvény segítségével válaszoljunk!
- Határozzuk meg az $O28:P29$ tartományban, hogy hány igazolt és hány igazolatlan hiányzás volt összesen, és ebből hány jut egy-egy tanulóra!
- A Q oszlopban képlet segítségével jelenítsünk meg egy \times karaktert, ha az adott tanuló jeles rendű (azaz 4,5-nél magasabb az átlaga)!
- Az R oszlopban jelenjen meg $!!!$ azoknál, akik elégtelent szereztek!
- Hány bukás volt az osztályban? Válaszoljunk képlettel a $B33$ -as cellában!
- Határozzuk meg a jeles rendű, illetve a bukott tanulók számát a $Q28:R28$ tartományban!
- A táblázatban az átlagok, illetve az egy főre jutó hiányzások két tizedesjegy pontossággal jelenjenek meg!
- Alakítsuk ki a szegélyt és a háttérszínt a mintának megfelelően! A $G31:G33$, illetve az $N31:N33$ tartomány cellái jobbra igazítottak. A tantárgynevek elforgatva, alulról fölfelé olvashatóan jelenjenek meg!
- Ábrázoljuk a tantárgyi átlagokat megfelelően megválasztott diagramon!

2. Egy önkormányzat ottalvós nyári erdei tábort szervez a településen lakó tanulók részére. A tábor a tanulók június 20. és augusztus 20. között csak egyszer vehetik igénybe, ám akkor tetszőlegesen hosszú ideig.

A tábor térítési díja 3000 Ft naponta, ebből a tanulók annyi százalék kedvezményt kapnak, ahány napot a táborban töltenek. A kedvezmény mértéke azonban a 20 százalékot nem haladhatja meg.

A fizetendő díjakat az alábbi táblázattal számolják ki. Az adatokat a tankönyv weblapjáról letöltött *Erdei tábor_nyers.xlsx* forrásállomány tartalmazza.

	A	B	C	D	E	F	G
1	Nyári erdei tábor			Napi térítési díj: 3 000 Ft			
2	Név	Érkezés	Távozás	Napok	Összeg	Kedvezmény	Fizetendő
3	Kertész Karina	június 21.	július 12.	22	66 000 Ft	20%	52 800 Ft
4	Dóka Andor	június 21.	július 18.	28	84 000 Ft	20%	67 200 Ft
5	Ábrahám Borbála	június 21.	július 1.	11	33 000 Ft	11%	29 370 Ft
6	Kakas Hugó	június 24.	július 26.	33	99 000 Ft	20%	79 200 Ft
7	Szemán Adina	június 24.	július 11.	18	54 000 Ft	18%	44 280 Ft
8	Bajai Valter	június 24.	július 12.	19	57 000 Ft	19%	46 170 Ft

	A	B	C	D	E	F	G
70	Tóbiás Róbert	augusztus 15.	augusztus 19.	5	15 000 Ft	5%	14 250 Ft
71	Istenes Vanda	augusztus 15.	augusztus 20.	6	18 000 Ft	6%	16 920 Ft
72							
73	Tanulók száma:	69					
74	Igénybe vett napok száma:	898					
75	Átlagos táborozási idő:	13,01					
76	Leghosszabb időtartam:	33					
77	Teljes bevétel:	2 280 360 Ft					

- Határozzuk meg képlettel a *D* oszlopban, hogy az egyes tanulók hány napot töltöttek a táborban! (Az érkezés és a távozás napja is beleszámít.)
- Másolható képlet segítségével számoljuk ki az *E* oszlopban a kedvezmény nélküli térítési díjat!
- Határozzuk meg a kedvezmény százalékos értékét az *F* oszlopban!
- Adjuk meg a kedvezményes térítési díjat a *G* oszlopban!
- A táblázat alatt, az *A73:B77* tartományban függvényt határozzuk meg, hogy
 - hány tanuló vett részt a táborban;
 - összesen és átlagosan hány napot vettek igénybe a tanulók;
 - mennyi volt a leghosszabb ott-tartózkodás időtartama;
 - mennyi volt a tábor teljes bevétele!
- A táblázat celláiban alkalmazzunk a mintának megfelelően dátum-, százalék-, illetve pénznemformátumot!
- A *B75*-ös cella tartalma két tizedesjegy pontossággal jelenjen meg!
- A mintának megfelelő helyeken vonjuk össze a cellákat, és alkalmazzunk a mintának megfelelő betűstílust és betűszínt!
- A táblázat szegélyét és háttérszínét a mintának megfelelően, zöld színnel alakítsuk ki!
- Milyen diagramon tudnánk az adatokat szemléltetni?

Az információ értéke

Az információ mint érték egyidős az emberiséggel. Ha valaki tudta, hogy mikor várható az állatok vonulása, mikor fog áradni a folyó, mikor lesz napfogyatkozás, akkor sikerre vezethette törzsének vadászatát, jól tudta időzíteni a vetést és az aratást, és akár isteni hatalom birtokosaként tetszeleghetett. Ma sincs ez másképp: aki ismeri a pontos buszmenetrendet, vagy tudja, hogy reggelente merre nincs csúcsforgalom, az ráér később is elindulni iskolába. Az információ értéke könnyen kifejezhető pénzben is – ilyen például, ha tudjuk, hogy hol kapható olcsóbban az, amit venni szeretnénk.

Az információt az emberek igen régóta megpróbálják elrejtteni egymás elől, és nagy erőfeszítéseket tesznek az elrejtett információ megszerzésére.

Kérdések, feladatok

1. Keressünk az interneten ismertetőt a Caesar-kód működéséről! Készítsünk Caesar-kóddal rejtjelezett üzeneteket tanuló társainknak, és fejtjük meg, amit mi kapunk tőlük! Meg tudjuk-e oldani egy szöveg megfelelő átalakítását táblázatkezelővel? Nézzünk utána, milyen egyéb híres rejtjelezési módszerek léteznek!
2. Egy világhírű detektívregény-író egyik leghíresebb novellájában is szerepet kap a titkosírás. Nézzünk utána, hogy hívják az írot, és mi a novella címe! Hogy fejt meg a főszereplő a titkosírást?
3. Háborúban és békében egyaránt fontos szerepet kaptak a hírszerzők, a kémek. Melyik kötelező olvasmányunkban szerepelt kém? Keressünk az interneten információt valóban létező, híres vagy híresen ügyetlen kémekről! Melyik hatalomnak kémkedtek? Milyen fontos információt szereztek meg? Hogyan buktak le?
4. Nézzünk utána, mi volt a szerepe az Enigma nevű készüléknek! Hogyan sikerült megfejteni a működését, és mi volt a jelentősége ennek? Ha tehetjük, nézzünk meg egy erről szóló filmet!
5. Mi az az ipari kémkedés?



► Enigma készülék

Az információ tehát értékes, és értéke a mai világunkban talán nagyobb, mint valaha. Az interneten történő kommunikáció általánossá válásával az információ megszerzésének és elrejtésének újabb és újabb módzatai alakulnak ki. Az információszerezés célja sem mindig olyasmi, amit a régi korok államférfiai, hadvezérei és tudósai könnyűszerrel megérthetnének.

Adathalászat

Az adathalászat, a „phishing” céljáról és néhány módszeréről a 7. évfolyamon már bővebben tanultunk. Megtudtuk, hogy általában a személyes adatok megszerzése a cél, a fő eszköz pedig a leendő áldozat megtévesztése, például ügyesen fogalmazott üzenetekkel, hamis weboldallal. A megszerzett adatokkal törvénytörően visszaélők más nevében tudnak beszélgetni, levelet írni, ügyeket intézni, adásvételeket lebonyolítani.

Telemetria

A telemetria szó „távoli mérést” jelent, és az adathalászattal szemben legális tevékenység. Telemetrikus rendszerek a mérőeszközök távoli leolvasását lehetővé tevő eszközök, például az egyszerű lakossági villanyórák vagy a nagyvárosok tömegközlekedésében részt vevő járművek pontos helyét, sebességét jelző műszerek. Telemetrikus mérőkkel működnek például a légzés, a szívverés vagy az alvás távoli megfigyelésére szolgáló eszközök, akár csak az elkóborolt kisállatok megtalálását segítő elektronikus nyomkövetők.



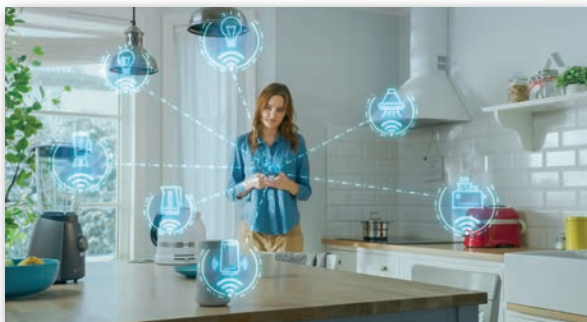
A szorosabb értelemben vett számítógépek és a mobil eszközök világában a telemetriát jó esetben felhasználói szokásokról készült statisztikák készítésére használják. Milyen alkalmazásokat futtatnak az emberek az eszközükön? Melyik fogyasztja túl gyorsan az akkumulátort? Hányszor lép ki egy alkalmazás, és milyen hibaüzenettel? Hány percig játszanak átlagosan egy játékkal? Hány pontot érnek el vele? Az ilyen adatok hasznosak lehetnek például az alkalmazás következő verziójának tervezésekor.

Ha például túl kevés pontot érnek el egy játékban, a következő verziót érdemes könnyebbre alakítani.

Az adatokat a mérőeszközök, a számítógépek és a mobil eszközök valamilyen felhőszolgáltatás szervereihez küldik, ahol aztán feldolgozzák őket, és az eredményekből levonják a megfelelő következtetéseket.

Kérdések, feladatok

1. Gyűjtsünk össze olyan adatokat, amelyeket érdemes lehet egy választott alkalmazás, egy mobil eszköz vagy egy operációs rendszer fejlesztőinek gyűjteniük és feldolgozniuk!
2. Miközben szoftverfrissítéseket tölt le vagy telemetria-adatokat küld ki az eszközünk, már kommunikál is a nagyvilággal. Vajon mennyire nehéz megoldani, hogy ne csak a törvény által engedélyezett adatokat küldje el magáról és a felhasználóról, hanem személyes, bizalmas adatokat is? Honnan tudjuk, hogy ez nem történik meg?



► Mi minden küldhet magáról információkat a lakásunkban?

3. Nézzünk utána az interneten, hogy mi mindenre használnak kémprogramokat a mindennapi életben!
4. Mi az az IoT (*Internet of Things*: „dolgok internete”)? Keressünk róla rövid videót az interneten!

Kereskedelmi célú adatgyűjtés

A kép- és videómegosztók, a keresők és a közösségi alkalmazások a legtöbbször abból élnek, hogy reklámokat nézetnek meg velünk. Kiemelten fontos számukra, hogy az érdeklődésünknek megfelelő reklámokat juttassanak el hozzánk. Ennek érdekében folyamatosan figyelik, milyen kereséseket futtatunk, milyen videókat nézünk meg, milyen képeknél időzünk el, vagy hogy miről chatelünk ismerőseinkkel. Az e-kereskedelem fontosabb szereplői, a nagyobb webshopok is figyelik vásárlási szokásainkat, és azok ismeretében ajánlanak újabb termékeket.

Kérdések

1. Észrevettük-e már, hogy a kereséseink eredménye vagy a megtekintésre felajánlott videók köre az előző kereséseinknek, videómegtekintéseinknek megfelelően változik?
2. Az említett webes alkalmazásokban ma már sok esetben ki lehet kapcsolni a személyre szabott, nekünk válogatott reklámokat. Hogyan tudjuk ezt megtenni az egyes alkalmazásokban? Szeretnénk-e kikapcsolni őket? Vajon jelenti-e az ilyen reklámok kikapcsolása azt, hogy adatot sem gyűjt rólunk az alkalmazás?

Bűnmegelőzési célú adatgyűjtés

Vannak, akik az internetet különféle bűncselekmények előkészítésére használják. Interneten szerzik be a szükséges eszközöket, ott gyűjtenek adatot a leendő áldozatokról, vagy tartanak kapcsolatot bűntársaikkal. A különféle hatóságoknak az internetes kommunikáció figyelésével merényleteket is sikerült már megakadályozniuk.

Feladat

Keressünk olyan esetekről szóló tudósításokat, ahol a bűnözők kommunikációjának megismerésével sikerült megelőzni vagy felderíteni bűneseteket!

Big data

A kifejezés „nagy adat”-ot jelent, de nem szoktuk lefordítani, mert a szó szerinti fordításnál bővebb a jelentése. Az informatikának arról az ágáról van szó, amelyben nagyon sok, nagyon gyorsan érkező, illetve nagyon sokféle adatot kell feldolgozni, és azokból következtetéseket levonni. A sok és sokféle adat érkezhethet például telemetriából, és a feldolgozásuk jelentheti egy felhasználó érdeklődési körének feltérképezését éppúgy, mint az épp száguldó versenyautó különféle mű-



szereiből érkező adattömeg folyamatos elemzését annak érdekében, hogy idejében észrevegyenek például egy közelgő katasztrófát.

Adatok titkosítása szállítás közben és tároláskor

Amikor az adataink az interneten utaznak, elképzelhető, hogy valaki igyekszik a mi eszközünk és a kommunikációs csatorna másik végén lévő eszköz (például egy webszerver vagy az ismerősünk mobiltelefonja) közé ékelődni, és lehallgatni a kommunikációt. Az utazó adatainkat tehát érdemes titkosítani. A titkosítás egyik legelterjedtebb formáját nap mint nap használjuk mi is. Ha megnézzük az általunk meglátogatott weboldalak URL-jét (például: <https://www.nkp.hu/>), akkor a *https* protokollmegjelölés s betűje épp arra utal, hogy a kommunikáció „secure”, magyarul titkos, biztonságos. A régebbi változat használatakor (*http*, s nélkül) az adatok lehallgatása lényegesen egyszerűbb.



▶ *Https* protokoll-megjelölés URL-ben

Az adatokat a számítógépünkön, mobilszközünkön is érdemes lehet titkosítva tárolni. Ha betör valaki a gépünkre, vagy ha a gépünk megfertőződik egy vírussal, akkor elképzelhető, hogy megkísérik például a böngészőnkben megadott jelszavainkat tartalmazó fájlt kijuttatni az internetre. Ezek az adatok védhetők mesterjelszóval, vagy tárolhatók külön jelszókezelő alkalmazásban. Ilyenkor a böngésző vagy az alkalmazás a háttérben egy olyan algorit-mussal titkosítja az adatainkat, amely csak

a mesterjelszó ismeretében képes a titkosított adatok visszafejtésére. Ha valaki mégis hozzáfér a jelszavainkat tároló fájlhoz, akkor sem tudja – legalábbis egykönnyen – megnyitni azt. Természetesen ilyenkor szóba jöhet a nyers erő (angolul *brute force*) módszere, és az illetéktelen behatoló végigpróbálgathat milliárdnyi lehetséges jelszót. Céljának eléréséhez azonban esetleg évekre lenne szükség – minél hosszabb a jelszó, annál több időbe telik a feltörése.

Kérdések, feladatok

1. Állítsunk be mesterjelszót az iskolai számítógépen futó böngészőben!
2. Hogyan érhetjük el, hogy akkor se tudjanak bejutni például az e-mail-fiókunkba, ha megismerik a jelszavunkat?

Az információ ára

A világ társadalmi ezekben az években kezdenek hozzá szokni ahhoz, hogy korlátlanok tűnő mennyiségű információt vagyunk képesek gyűjteni, tárolni és feldolgozni. Az előző leckében láttuk, hogy ezt a rengeteg információt jóra és rosszra egyaránt fel lehet használni. De valójában mennyi információról van szó?

Bitek és byte-ok

Az információ mennyiségét megadhatjuk byte-ban. Egy byte nyolc bitből (jele: b), azaz nyolc 0-ból vagy 1-ből álló, kettes számrendszerű szám. Így 2^8 , azaz 256 különböző értéket vehet fel, tehát alkalmas lehet például egy-egy európai nyelv összes kis- és nagybetűjének, írásjeleinek és számainak ábrázolására. A következő példában nem tévedünk nagyot, ha egy-egy leírt karaktert egy-egy byte hosszú információnak tekintünk.

Byte-okban (jele: B) kifejezhetjük egy könyv hosszát is. Például a Biblia a maga nagyjából három és fél millió betűjével közel háromezer-ötszáz kilobyte (kB) vagy három és fél megabyte (MB) helyet foglal el.

Régen kőtáblára vésték, papírra írták, majd nyomtatták az információt, vagy az szájról szájra járt (népmesék, népdalok), de napjainkban az információ tárolására és továbbítására az esetek messze túlnyomó többségében számítógépet használunk. A számítógép egyaránt lehet általános célú vagy speciális, egy konkrét feladatra fejlesztett eszköz.

A képek rögzítéséről beszélve érdemes tudnunk, hogy egy-egy képpont vörös, zöld és kék összetevőjét is egy-egy byte-on írjuk le, azaz mindegyiknek 256-féle értéke lehet. Egy képpont így három byte-ot foglal a számítógép memóriájában vagy háttértárán. Egy ma szokásos Full HD-monitor felbontása 1920×1080 képpont. Egyetlen ekkora felbontású kép $1920 \times 1080 \times 3 = 6\,220\,800$ B, azaz 5,93 MB információ, tehát majdnem kétszer annyi, mint a Biblia teljes szövege.

Az adatok mennyisége hatalmas, még akkor is, ha tudjuk, hogy mind a képeket, mind a videókat tömörítve tároljuk, és így lényegesen kevesebb helyet foglalnak, mint amit előzőleg kiszámoltunk.

Kérdések

A napjainkban szokásos másodpercenkénti ötven képpel számolva hány byte-nyi képi információ van egy ötperces videóklipben? Ez hány bibliányi adat?

Környezetterhelés

A fenti számításunkkal közelítő képet alkothatunk arról, hogy mi az eszközigénye egyetlen jelentős felhőszolgáltató tárolókapacitásának. Pontosabban a kapacitás éves bővülésének, hiszen csak egyetlen év feltöltéseivel számoltunk. Ha átgondoljuk azt is, hogy az adatokat redundánsan, azaz több másolatban és több kontinensen tárolják, az előbbi számok többszörösével kell tovább számolnunk. Gondoljuk át azt is, hogy eddig csak a tárolókapacitásra koncentráltunk. Nem vettük figyelembe a további szükséges számítógép-alkatrészeket, illetve a hálózati kapcsolatokat megvalósító és még sok egyéb alkatrészt.



► Mekkora kapacitású lehet egy ilyen eszköz? Mennyibe kerül? Hányszor fér el rajta a tankönyvünk?

Ha pedig azt is meggondoljuk, hogy egyetlen felhőszolgáltatásról beszéltünk, pedig ki tudja, hány van, akkor egészen szédítő, szinte felfoghatatlan hardver- és áramigények kezdenek előttünk körvonalazódni.

Kérdések, feladatok

1. Hány gigabyte (GB) méretű egy egész estés – nagyjából százperces – filmet tároló fájl?
2. Mekkora kapacitású és mennyibe kerül egy internetes boltban egy számítógépekbe való közepes méretű adattároló? Hány percnyi videó fér el egy ilyen adattárolón?
3. Hány percnyi videót töltenek fel a legnépszerűbb videómegosztóra percenként? Hány percnyi videó kerül fel egy év alatt? Az előzőleg látott adattárolóból mennyi kellene ehhez? Mekkora helyet foglal el, és mekkora tömegű ennyi adattároló? Mennyi időnként használnának el és szorulnak cserére az ilyen adattárolók?
4. Mennyi áramot fogyasztanak az adattárolók? Mekkora árammennyiség ez Magyarország legjelentősebb erőművének kapacitásához képest?
5. Az adatok tárolása és feldolgozása manapság speciálisan erre kialakított épületekben, úgynevezett adatközpontokban történik. Keressünk és nézzünk meg adatközpontot bemutató videót! Nézzünk utána, hogy országunkban hol található jelentősebb adatközpontok!
6. Nézzünk utána, hogy az Európai Unió országaiban jellemzően milyen energiahordozóból állítják elő az áramot!

Az előzőekben távoli, de az arányok érzékeltetésére valamennyire alkalmas képet kaptunk információs társadalmunk infrastruktúrájának felhőbeli végéről. Lássuk most a másik végét: azokat az eszközöket, amelyekkel az információt fogyasztjuk és előállítjuk! A legfontosabb ilyen eszközeink a számítógépeink és a mobileszközeink, illetve a televízióink. Ne felejtsük el a kiegészítő berendezésekről: a lakások, iskolák, cégek internetkapcsolatához szükséges kábelmodemokről, routerekről, switchekről, WiFi access pointokról, illetve a televíziók dekódereiről és set-top boxairól!

Kérdések, feladatok

1. Kérdezzük meg a családtagokat, ismerősöket, hány évig szoktak egy okostelefont használni, mielőtt lecserélik! Átlagoljuk ki ezeket az értékeket a tanulócsoportban!
2. Hány mobiltelefon-előfizetés van ma Magyarországon? Egy átlagos okostelefon méretével és tömegével számolva mondjuk meg, hogy hány köbméter, illetve hány kilogramm telefon van üzemben ma országunkban! Hogyan aránylik a kapott térfogat az iskolánk méretéhez, mennyire töltené meg az iskolát?
3. Az előzőleg kapott, a telefonok élettartamát mutató átlaggal kalkulálva hányszor töltenék meg iskolánkat a tíz év alatt vásárolt telefonok? És ha ezt Európa lakosság számára vetítjük? És ha a telefonokhoz a hálózati adaptereket is hozzászámítjuk?
4. Nézzünk utána, mi a feladata a kábelmodemnek, a switchnek és a WiFi access pointnak!



► Hány telefont dobnak ki a lakóhelyeden évente?

Ha aggódnánk az előző számítási feladatokban kapott eredmények miatt, talán arra gondolnánk, hogy mindez nem olyan jelentős környezeti teher, hiszen az eszközök újrahasznosíthatók. Érdeemes tudnunk, hogy a legtöbb eszközt hosszú-hosszú évekig tartják még otthon, mielőtt átadnák újrahasznosításra. Az igazi probléma azonban az, hogy az elektronikai hulladék újrahasznosításának hatékonysága nagyon kicsi. Minél jobban miniatürizált egy termék, minél többféle anyag van benne nagyon kis helyre összezsúfolva, annál nehezebb és költségesebb az elemek szétválasztása és újrahasznosítása. Mindez ne tartson vissza bennünket az elektronikai hulladék megfelelő helyeken való leadásától, hiszen még mindig ez a legjobb, amit tehetünk a környezetvédelem szempontjából.

Feladat

Keressünk az interneten olyan átvételi pontokat, ahol lakhelyünk közelében szelektív hulladékgyűjtés történik, és olyat, ahol elektronikai hulladékot is átvesznek!

Információöozön és társadalom

Szokták mondani, hogy sohasem az eszköz jó vagy rossz, hanem az ember, aki használja. Minden találmánnyal lehet élni és visszaélni. Ennek megfelelően két oldala van annak is, hogy az emberiség történetében először elképzelhetetlen mértékű információ áll minden ember rendelkezésére.

Kérdések, feladatok

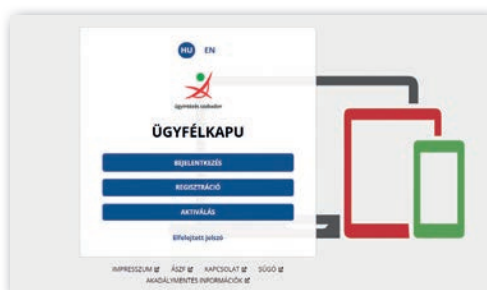
1. Vajon mekkora az olyan információ aránya a bárki számára hozzáférhető adatokból, amelyek pusztán a szórakozásunkat szolgálják?
2. Az iskoláinkban oktatott legtöbb tény – évszámok, folyók nevei, fővárosok, állatok és növények adatai, híres emberek életrajza – mind elérhető és pár kattintással megtalálható az interneten. Van-e értelme ilyesmiket megtanulni, bebiflázni ezek után?
3. A fordítóprogramok nagyon sokat fejlődtek az elmúlt években. A legjobban fejlődők valamilyen mesterséges intelligenciával folyamatosan tanulnak: összevetik a több nyelven is fellelhető internetes írásokat, hangfelvételeket, videókat. Várhatóan hamar eljön az idő, amikor nehéz lesz megkülönböztetni a gépi fordításokat az ember által készítettől. Lesz-e értelme ezt követően nyelveket tanulnunk?
4. Vannak olyan vélemények, melyek szerint megfelelő önfegyelem és elszántság mellett napjainkban minden szakma megtanulható az internetről. Mi a véleményünk erről az állításról?
5. Nézzünk utána, hogy mai világunkban melyek a pénzügyileg legsikeresebb cégek, és kik a legvagyonosabb emberek! Mivel foglalkoznak?
6. Éreztük-e már úgy egy-egy téma internetes kutatásakor, hogy túl sok a fellelhető információ? Hogyan tudunk válogatni belőlük?
7. Vajon mennyi a hamis információ az interneten? Mi segíthet eldönteni azt, hogy
 - amit épp olvasunk, az igaz-e;
 - a cikkben leírt tudományos tények valóságosak-e;
 - a bemutatott események tényleg úgy történtek-e, ahogy nekünk tálalják?
8. Szeretnénk-e olyan országban élni, ahol minden internetes – és esetleg a való világban megtett – lépésünket megfigyelik, rögzítik, elemzik? Sokak szerint máris vannak ilyen országok. Keressünk erről szóló cikkeket, tudósításokat!

Mindennapok az információs társadalomban

E-állampolgár

Két évszázaddal ezelőtt egy ember könnyen leélhette az egész életét úgy, hogy alig került kapcsolatba a különféle hivatalok ügyintézésével. Elképzelhető, hogy egész életében csak annyi hivatalos feljegyzés született róla, hogy a lakóhelyének papja bejegyezte az anyakönyvbe a születését, majd – jó esetben már egy másik pap – a halálát.

Mára ez a helyzet gyökeresen megváltozott a modern társadalmakban. Hivatalok egész sora intéz különféle ügyeket, de olyan intézményekben is rengeteg hivatalos feljegyzés készül rólunk, amelyekre nem tekintünk hivatalként – gondoljunk például az iskolára vagy egy egészségügyi intézményre.



Magyarországon 2005-ben kezdte meg működését egy olyan webes szolgáltatás, amely hivatali ügyek személyes megjelenés nélküli, online intézését teszi lehetővé. A szolgáltatás a <https://magyarorszag.hu>, illetve a <https://mo.hu> címről kiindulva érhető el. A hétköznapiakban mind a mai napig elsőként elterjedt nevén, **Ügyfélkapuként** hivatkozunk rá – így teszünk mi is.

Kérdések

1. Miért jó, ha a hivatali ügyeket online, például egy webes felületen intézhetjük személyes megjelenés helyett?
2. Ki használhatja az Ügyfélkaput?
3. Nézzünk utána a következőknek:
 - Hogyan lehet regisztrálni az Ügyfélkapura?
 - Miben tér el az eljárás a legtöbb honlapon szokásostól?
 - Mi az említett eltérés oka?
 - Neked lehet-e már Ügyfélkapu-hozzáféréсед?

Az Ügyfélkapun ma már sokkal többféle ügyintézés végezhető el, mint indulásának idején, és a körük szerencsére egyre bővül. Vannak olyan részei is a felületnek, amelyek nem használhatók ügyintézésre, de tájékoztatást adnak arról, hogy milyen adatokat tartanak nyilván rólunk.

Kérdések, feladatok

1. Milyen ügyek intézhetők az Ügyfélkapun?
2. Milyen személyi okmányok megújítása kezdeményezhető itt?
3. Kérdezzük meg szüleinket, felnőttkorú családtagjainkat, hogy ők – ha használják – mire használják az elektronikus ügyintézésnek ezt a módját! Ha van rá módunk, nézzük meg, hogy milyen egészségügyi adatokat tárol rólunk az innen kiindulva megnyitható rendszer!

Közlekedés

A közlekedésben segítségünkre lévő alkalmazásokat három csoportba sorolhatjuk.

Az első csoportba az egyszerű **elektronikus menetrendek** tartoznak. Megmondjuk, hogy honnan, hova és mikor utaznánk, és kiderül, hogy mikor megy oda vonat vagy autóbusz. Az ilyen alkalmazásokkal manapság már sokszor a menetjegyet is meg tudjuk vásárolni.

A második csoport tagjai az **útvonaltervezők**. A nevükkel ellentétben nemcsak megtervezni segítenek az útvonalat, hanem végigmenni is rajta. Ehhez GPS (*Global Positioning System*: globális helymeghatározó rendszer) segítségével találják meg a mindenkori helyzetünket. Nagyon sok autóvezető használja őket, de jól jöhetnek, ha gyalog kell odatalálnunk egy ismeretlen helyre. Van olyan változatuk is, amely elektronikus menetrenddel egybeépítve a nagyvárosi tömegközlekedés részvevőit segíti – akár jegyvásárlási lehetőséget is biztosítva.

A harmadik csoportot a **túraútvonal-tervezők** alkotják, amelyeket akár az előző csoport részeként is említhetnénk. Ezek az alkalmazások a túrázókat segítik. Útvonalat nem feltétlenül terveznek, de pontosan megmutatják a helyünket olyan részletes turistatérképeken, amelyek a második csoportba tartozó alkalmazásokban nem állnak rendelkezésünkre. További előnyük, hogy képesek rögzíteni a bejárt utunkat, amelyet aztán útvonalajánlatként, részletes szintemelkedési adatokkal kísérve megoszthatunk az interneten – így mások is hasznát vehetik tapasztalatainknak a következő kirándulásuk megtervezésekor.



► Mit látunk a képen? Miért fontosak az ilyen eszközök a GPS működésében? Mi a jelentőségük a mindennapi adatforgalmazásban?

Kérdések, feladatok

1. A MÁV nem olyan régen adta ki az utolsó nyomtatott éves menetrendjét. Melyik évben történt ez? Bár egykor sok-sok példányban nyomtatták az ilyen menetrendeket, lassan gyűjtők féltett kincsévé válnak. Kérdezzük meg otthon, hogy van-e ilyen menetrendünk, és ha igen, akkor lapozzunk bele!
2. Azon kívül, hogy könnyebb használni, milyen előnnyel jár még az elektronikus menetrend használata a nyomtatottéhoz képest? Mikor jöhet jól mégis a nyomtatott menetrend?
3. Milyen szempontokat vesz figyelembe egy útvonaltervező alkalmazás az optimális útvonal megállapításakor?
4. Az útvonaltervezők meg tudják tervezni a legrövidebb és a leggyorsabb utat is, de a kettő nem feltétlen esik egybe. 2021 tavaszán jelentős hír volt, hogy az egyik legnevesebb útvonaltervező már a legrövidebb utat mutatja alapértelmezetten. Miért fontos ez a hír, és minek köszönhető a változás?
5. Készítsünk bemutatót a GPS működéséről!
6. Nézzünk utána, mik azok a GPX-fájlok, és milyen alkalmazásokkal tudjuk használni őket!
7. Milyen webhelyeken találunk GPX-fájlokat? Van-e köztük olyan, amely a környékünkről készült?
8. Telepítsünk mobilszközre GPX-megjelenítő alkalmazást, és próbáljuk ki a működését!
9. Készítsünk GPX-fájlt a környezetünk néhány látványosságát bejárva! Néhány jól sikerült fotóval és rövid kísérőszöveggel együtt osszuk meg például az iskola honlapján!

Pénzügyek

Alig-alig van már olyan ember, akinek ne lenne bankszámlája, és meglehetősen sokan használják a bankok internetes felületét, akár böngészőből, akár mobilszközön futó alkalmazásból. Az internetes banki ügyintézés szerepe hasonló a hivatali ügyek internetes intézéséhez – gyorsabb és kényelmesebb.

A vásárlásainknak egyre nagyobb részét bonyolítjuk le webshopokban.



Kérdések, feladatok

1. Milyen banki ügyeket szokás a leggyakrabban intézni a bankok internetes szolgáltatásával? Intéznék-e így ügyeket felnőtt családtagjaink?
2. Beszéljük meg közösen, milyen hasonlóságok vannak a *Magyarország.hu* weboldalra és a bankok weboldalára történő regisztrációban! Mi ennek az oka?
3. Szoktunk-e a családban interneten vásárolni? Ha igen, elsősorban milyen árucikkeket?
4. Miért szoktak és miért tudnak webshopok olcsóbban árulni termékeket, mint az „igazi” boltok?
5. Mi adott az évtized elején újabb lendületet a webes vásárlás elterjedésének?
6. Tudunk-e olyan árucikkeket megnevezni, amelyeket eszünkbe sem jutna „igazi” boltban keresni, vásárolni?
7. Milyen lehetőségektől esnek el azok, akik nem tudják vagy nem akarják pénzügyeiket e-bankban intézni?

► Milyen lehetőségre hívja fel a figyelmet a kép? Használánk-e így a telefonunkat? Mi az az NFC?

Szakmai internetes jelenlét, online szakmai életrajzok

Bár mindannyian tudjuk, hogy „a jó bornak nem kell cégér”, azaz ami igazán jó, azt nem kell hirdetni, azt is látjuk, hogy a legtöbb cégnek, vállalkozásnak van valamekkora internetes jelenléte. Az egészen kicsi, egyszemélyes cégek, egyéni vállalkozók általában bejáratott ügyfélkörrel rendelkeznek. Ennek ellenére őket is gyakran megtaláljuk az interneten – ha nincs is önálló webhelyük, legalább a közösségi oldalakon elérhetők. Így a már meglévő ügyfeleik is könnyebben kapcsolatba tudnak velük lépni.

A nagyobb cégek a fontosabb beosztásokba gyakran próbálnak interneten hozzáértő munkavállalókat találni. Különösen igaz ez a multinacionális vállalatokra, ahol sokféle nyelvet beszélő emberek dolgoznak együtt – bár csak ritkán egy helyen –, és ahol éppen ezért gyakran angolul beszél egymással a magyar és a más országokban élő munkatárs.

Akik azt szeretnék, hogy leendő munkaadóik könnyebben rájuk találjanak, és így nagyobb eséllyel kapjanak jó állást, gyakran helyezik el az interneten szakmai önéletrajzukat. Az online önéletrajzokban – offline megfelelőikhez hasonlóan – megírják, hogy eddigi életükben milyen munkákat végeztek, milyen szakmai tapasztalataik vannak. Az önéletrajzok, online portfóliók nyelve is gyakran az angol.

Kérdések, feladatok

1. Melyik környékünkön lévő kis cégnek, vállalkozónak van honlapja? Van-e esetleg valamelyik családtagunknak, ismerősünknek? Hol üzemeltetik a honlapot, hol tárolják a fájljait? Mennyibe kerül a fenntartásuk?
2. Melyek azok a környékbeli kis cégek, vállalkozások, amelyeket közösségi oldalakon érünk el? Be tudunk-e közösségi oldalon fodrászhoz, kozmetikushoz, fogászhoz jelentkezni, az autószerelővel, a virágboltossal, a könyvtárral és a pizzériával kapcsolatot tartani? Kivel még? Van-e, akivel inkább e-mailezünk?
3. Mi mindent helyeznénk el a saját honlapunkon, közösségi oldalunkon, ha nekünk volna valamilyen, akár a fentebb felsorolt területeken működő vállalkozásunk?
4. Keressünk szakmai önéletrajzokat az interneten! Milyen elemei vannak egy önéletrajznak? Melyik önéletrajz kelt bennünk bizalmat, és miért? Készítsünk az Europass honlapon szakmai önéletrajzot!
5. Van-e a környékünkön multinacionális vállalat? Mivel foglalkozik?
6. Melyik a legjelentősebb olyan internetes oldal, amely online önéletrajzok tárolásával foglalkozik? Ha van rá módunk, nézzük meg rajta híres emberek szakmai életrajzát! Vajon mi hoz bevételt ennek a weboldalnak?

Valós ember – virtuális személyiség

Játékok mint virtuális valóságok



- Milyen eszközt látunk a képen?
Mire való?

tányér, és házat, várost építettünk játékgúráinknak, akik részben mi magunk voltunk. Ilyenkor egy virtuális világot teremtettünk magunknak.

A virtuális világok közé tartoznak a számítógépes játékokban életre kelő valóságok is, melyekben hosszabb-rövidebb ideig sokan szívesen töltik idejüket. Talán azokat a játékokat élik át hosszabb időre (hetekre, hónapokra) a játékosok, amelyekben valamennyire ők irányítják a sorsukat. Fejlesztik képességeiket, felszereléseket vásárolnak maguknak, kapcsolatokat építenek ki – virtuális életet kezdenek el élni. A múlt század '80-as éveiben kezdődött ez az élmény, amely a sokszereplős, nyílt világú játékokon keresztül a valós élethez hasonló kapcsolati háló és a magunk alkotta világok megteremtéséig ível.

Kérdések, feladatok

1. Melyik volt az első olyan, híres űrhajós játék, amelyben felszereléseket lehetett vásárolni a masinánkra, küldetéseket vállalni és elutasítani, és ezernyi világba elutazni? Mikori ez a játék, és milyen számítógépeken játszották? Nézzünk meg róla videót!
2. Mit várunk el ma egy játéktól, amellyel sok időt töltenénk el? Mennyire fontos a grafika, a zene, a hangulat, a játék világa? Mennyire fontos az elérendő cél? Mik voltak a legfurcsább játékelményeink?
3. Mik azok a nyílt világú játékok? Játszottunk-e már ilyet? Aki igen, meséljen az élményéről!

A játékbeli szereplőnk, a hősünk, a házunk, a farmunk, a kosárlabdacsapatunk sok esetben annyira fontos számunkra, hogy élünk a játékon belüli vásárlások lehetőségével. Valós pénzért veszünk a virtuális szereplőnek tárgyakat, a farmunkra állatokat, az autónkra kiegészítőket, „fodrászhoz” küldjük a focistánkat, és „kutyakozmetikushoz visszük” ölebünket.

Kérdések

1. Van-e számítógépünkre, mobilkészülökünkre alkalmazásokon belüli vásárlást lehetővé tevő alkalmazás telepítve? Szoktunk-e élni a lehetőséggel? Ha nincs, azért még ismerünk ilyen játékokat? Lehet-e bennük előrejutni vásárlások nélkül?
2. Sokéves múltira tekint vissza a Second Life című szoftver. Járjunk utána és foglaljuk össze, hogy mi is ez az alkalmazás! Hány aktív felhasználója van? Milyen korosztályúak használják? Mondjuk el a véleményünket róla! Milyen hasonló virtuális valóságok léteznek?
3. Tekintható-e virtuális világnak a Minecraft? Fejtsük ki véleményünket!

A közösségi oldalak virtuális világa

Nem mindenki kívánczik annyira nyilvánvalóan virtuális világokba, mint amilyeneket az előzőekben bemutatunk. Sőt, sokan nem is tekintik virtuális világnak a közösségi oldalakon folytatott életet. És nem is lesz az, ameddig valós kapcsolataink megerősítésére, valódi ismerőseinkkel való kommunikációra használjuk, és ameddig általában véve nem engedünk túlzottan nagy teret életünkben a közösségi oldalaknak.

Vannak azonban, akik túl komolyan veszik a virtuális világot. Az ő számukra nagyon fontos, hogy elegendő tetszésnyilvánítást kapott-e egy-egy videójuk, képük, hozzászólásuk. Egy-egy negatív kritika sokszor mély kétségbeesésbe hajszolja őket.

A „lájkvadászat” azonban hamar vezet olyan tettekhez, megnyilvánulásokhoz, amelyekről egyébként tartózkodna az az ember, aki helyén kezeli az efféle tetszésnyilvánítás értékét.



Kérdések

1. Vajon miért annyira fontos sok embertársunknak a minél több internetes tetszésnyilvánítás? Mit egészítenek ki, mit pótolnak vele?
2. Ismerünk-e olyan embereket, akik gyökeresen másképp „élnek” a közösségi hálók, a kép- és videómegosztók világában, mint a valóságban?
3. Neveznénk-e személyiségtorzítónak ezeket a „valóságokat”? Fejtsük ki véleményünket!

A közösségi oldalak virtuális világa

Korábban volt már róla szó, hogy sok ingyenesen használható internetes szolgáltatás fő bevételi forrása a reklámok bemutatása a felhasználóknak. Arról is volt már szó, hogy igyekeznek feltérképezni érdeklődési körünket, hogy a bennünket talán jobban érdeklő reklámokat mutathassanak. Az érdeklődésünk feltérképezése azért is fontos e weboldalak számára, mert így olyan híreket, videókat, képeket tudnak megmutatni, amelyekre kíváncsiak vagyunk. Amíg ezeket a tartalmakat fogyasztjuk, addig is az ő reklámjaikat nézzük, és nekik termeljük a bevételt, nem pedig a konkurens weboldaloknak. Az ő „ingyenes” tartalmukért mi a reklámok nézegetésével eltelt időnkkel és figyelmünkkel fizetünk.

Az, hogy folyamatosan bennünket érdeklő tartalmakat tesznek elénk, több erős „mellékhatással” is jár. Elkezd körülvenni



bennünket egy kis világ, amelyben minden arról szól, ami bennünket is érdekel. De ami ennél is fontosabb: ebben a kis világban mindenki a mi véleményünket hangoztatja – hiszen a weboldal algoritmusai ilyen tartalmakat válogatnak össze nekünk. Sok rajzolás videót nézünk? – ilyeneket fog nekünk ajánlani a videómegosztó. Sok extrém sportos hírre kattintunk? – ezekkel lesz teli a hírfolyamunk. Nem értünk egyet valamivel? Hirtelen mindenki ugyanazt pocskondiázza majd az internet bennünket körülvevő részén.

Emberként lassan hajlamosak leszünk túlzott szerepet tulajdonítani annak, ami ebben a kis világban történik. Azt gondolhatjuk, az a tévképzet alakulhat ki bennünk, hogy mindenki hozzánk hasonlóan gondolkodik – pedig ez a legritkább esetben van így. Ezt különösen akkor fontos szem előtt tartanunk, ha egyszer csak azon kapjuk magunkat, hogy valamilyen rossz érzésünket, félelmünket, szorongásunkat, esetleg a bennünk bujkáló ártó szándékot felerősítő tartalmakkal van tele a napunk, mert újra meg újra ilyen cikkek, képek és videók kerülnek elélnk.

Igyekezzünk kiszakadni ebből a világból, és visszatérni a valóságba! Figyeljünk oda jobban a bennünket *valóban* körülvevő világra.

Influenszerek és mellékhatásaik



► Mennyire tudjuk értékelni egymás társaságát? Hány éves korra lehetünk mobileszközünk rabjai?



Influenszereknek (magyarul talán a „befolyásolók” szó adja vissza a kifejezés jelentését leginkább) azokat az embereket nevezhetjük, akiknek interneten hangoztatott véleményére nagyon sokan figyelnek oda. Nap mint nap sokan nézik a képeiket, videóikat, vagy olvassák a cikkeiket. Vannak közöttük olyanok, akiknek tényleg fontos mondanivalójuk van, és akik kiérdemelték embertársaik figyelmét. Másokat a híressé válás, a becsvágy mozgat, megint mások egyszerűen pénzeskereseti lehetőséget látnak benne – nekik ez a „munkahelyük”.

Kérdések

1. Ki Sir David Attenborough? Minek az érdekében emelte fel sok filmjében és megszólalásában a szavát? Mennyi ideig tartott „influenzsersége” a leghíresebb képmegosztón? Hány évesen lett internetes influenszer? Mekkora sikert ért el? Hogyan vetett véget influenzerei életszakaszának? Tudunk-e hasonló, nemes célokért küzdő influenszerekről?
2. A műsoraikon kívül milyen bevételi forrásuk szokott lenni az influenszereknek?
3. Ismerünk-e olyan influenszereket, akiket máshonnan ismer a világ? Kiket követnek a legtöbben?
4. Szeretnénk-e influenszerként élni? Miért igen, és miért nem?
5. Vajon miben változik meg egy internetes influenszer élete ahhoz képest, mint amilyen influenszerré válása előtt volt? Milyen előnyei és hátrányai lehetnek az influenzerségnek?
6. Mi a különbség a vlogger és az influenszer között?
7. Mik azok az unboxing videók? Miért szeretik nézni az ilyen videókat az emberek? Mennyire valós helyzet, amikor valakinek hetente akár többször is akad kicsomagolnivalója?

Az internetes videóknak, cikkeknek, képeknek nagyon nagy része szerencsére nem akarja ránk erőszakolni a készítője véleményét. Az ilyen internetes tartalmak készítői a legtöbbször nem influenszerek. Sok esetben egyszerűen csak felveszik, hogy merre mentek a kamionjukkal, milyen új számítógépük, munkaruhájuk, szempillaspiráljuk vagy könyvük van. Elmesélik, hogy milyen filmet láttak, és az miért tetszett vagy nem tetszett nekik, hogy hol vonatoztak, vagy milyen volt a hétvégi kirándulás. Animációt készítenek híres csatákról, megmutatják, mit láttak a mikroszkópjukban, vagy lefilmezik egy halpaprikás elkészítését. Nekik köszönhetően sok mindent meg is tanulhatunk az ablakszigeteléstől kezdve a hímzésen, a vászonképek festésén és a gördeszkatrűkkökön át a fényképezésig vagy a weboldal-készítésig. Felnyitják szemünket igazi világunk sokféle, számunkra másféleképp nehezen megismerhető szépségére és érdekességére.

A virtuális világ és hatása az egészségünkre

Amit magunkról közvetítünk, és amit a többiekből látunk



- Mennyire hasonlít az emberek internetes arca az igazira? Mennyire hasonlít rá a miénk?

Amikor virtuális személyiségünk, online identitásunk alakításáról beszélünk, tudatosítani kell magunkban, hogy sosem a valóságot látatjuk magunkból az interneten. Amikor kiteszünk valamit az internetre – legyen az egy kép, egy videó, egy hozzászólás vagy egy blogbejegyzés –, mindig azt mutatjuk meg magunkból, amit meg is akarunk mutatni. Általában azt, amikor valami sikerült, valamire büszkék vagyunk, valaminek örülünk, valamit érdekesnek találunk, vagyis arról posztolunk, hogy boldogok vagyunk, és jól érezzük magunkat.

És – a legtöbbször – mélyen hallgatunk arról, amikor nem sikerült valami, amikor pocsékul érezzük magunkat a bőrünkben, amikor csalódtunk valakiben, betegek vagyunk, egyes lett a dolgozatunk, vagy lekéstük a buszt, elromlott a mosógép, bemozdult az összes fényképünk, és nem vettek fel abba a középiszkolába, ahová menni szerettünk volna.

Kérdés

Vajon mi lehet a fent olvasott viselkedés oka?

Minthogy a legtöbben a szépet és jót osztjuk meg magunkról, mások megosztásait olvasva és nézve az a kép alakulhat ki bennünk, hogy mindenki mindig felhőtlenül boldog – bennünket kivéve, akiknek nem mindig csupa öröm az életünk. Ez a kép – ha picit is belegondolunk – nyilvánvalóan hamis. Arra mégis jó lehet, hogy rossz érzést, szorongást, irigységet alakítson ki bennünk, elszomorítson, és rosszabb színben tüntesse fel a világunkat, mint amilyen. A téma kutatói felnőtteknél és gyerekeknél egyaránt megfigyelik ezt a reakciót.

Kérdések

1. Észleltünk-e már magunkon valamit ezekből az érzésekből?
2. Mit tehetünk azért, hogy minél kevesebbszer érezzük magunkat így?

A virtuális éniünket érő agresszió

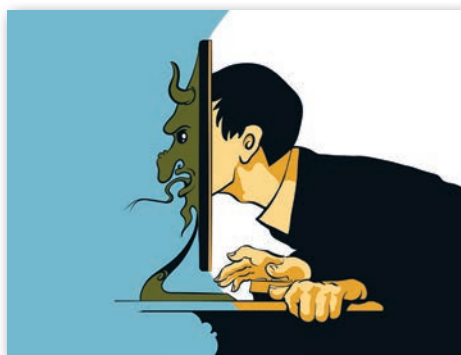
Általában fontos lenne, hogy mielőtt megszólalunk, gondoljuk át szavaink hatását. Ha valamit közölni szeretnénk, de bántóan, sértően fogalmazunk, a beszélgetőpartnerünk automatikusan védekezni kezd, vagy ellentámadásba lendül, és a legritkább esetben lesz nyitott a mondandónk elfogadására.

Gyakori, hogy az interneten lényegesen keményebben, sértően fogalmaznak a megszólalók, mint amikor nem virtuális, hanem valódi közösségben kell kimondaniuk a véleményüket. Ezeket a megszólalásokat a valóságban sokszor nem hallgatnánk végig. Ha mégis elhangzottak, a hallgatók a lesújtó arckifejezéstől kezdve a hangos elutasításig terjedően

fejeznék ki nemtetszésüket, aminek általában van visszatartó ereje. Az internetes hozzászólásoknál az ilyen reakciókra nincs lehetőség, és a hozzászóló nem kapja meg ezeket a nagyon fontos visszajelzéseket.

Tovább rontja az internetes agresszió áldozatainak a helyzetét, hogy az internet nem felejt. Ha valós énként éri agresszió, a hatása idővel csökken, pusztán azért, mert az emlék elhalványul, és a régebben történt események súlyukat veszítik. Egy sértő internetes megszólalást újraolvasva, egy bántó képet, egy megálázó videót újra megnézve a rossz érzés és a keserűség ereje alig csökken.

Ha csak meggondolatlanságból viselkedünk sértően, és bántunk meg ezzel valakit, az is járhat messzemenő következményekkel. Ha szándékosan tesszük, akkor érdemes lehet elgondolkodni rajta, hogy mit éreznénk, ha más tenné velünk ugyanezt.



► Tudunk-e önmagunkhoz hűek maradni az internetes kommunikációban?

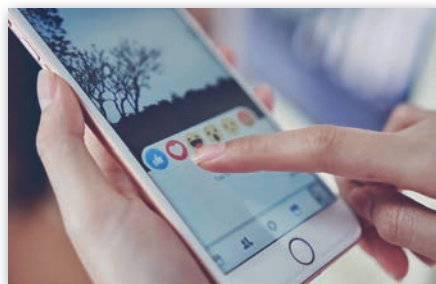
Kérdések

1. Tapasztaltuk-e már, hogy internetes hozzászólásokban sokkal durvábban fogalmazznak az emberek, mint egymás mellett állva és úgy beszélgetve? Mi lehet ennek az oka?
2. Előfordult-e már, hogy bántó, sértő megszólalások, képek, videók, esetleg zaklatás célpontjává váltunk az interneten? Hogyan érdemes kezelni az ilyen eseteket? Kitől kérhetünk segítséget?
3. Tudjuk-e jelezni a webes alkalmazás üzemeltetőjének, ha egy tartalom bántó, sértő, megálázó? Hogyan? Szoktunk-e élni ezzel a lehetőséggel?
4. Tudunk-e olyan esetről, amikor tragikus következménnyel járt, hogy valakit számára elviselhetetlenül sok internetes elutasítás vagy elviselhetetlenül komoly agresszió ért?

Virtuális jutalmak

Az emberek nagy többsége általában szeret kapni valamit. Sokféle dolognak tudunk örülni, amit a kereskedelem alaposan ki is használ. Gondoljunk csak át, hány olyan tárgyunk van, amely lényegében semmire sem kell, sőt, az idő legnagyobb részében csak útban van, mégsem szeretnénk megválni tőle. Elismerést is szívesen kapunk, jó érzést kelt bennünk, ha megdicsérnek bennünket vagy a munkánkat.

Amikor egy-egy közzétett írásunkat, képünket, videónkat ismerik el az adott internetes alkalmazásban, ugyanígy tudunk neki örülni. Azonban nagyon fontos megértenünk, hogy az elismerés kifejezése a legtöbbször egy egyszerű egérgattintással, a kijelző megérintésével lehetséges, azaz az elismerést adótól nem igényel különösebb erőfeszítést. Mindez nem jelenti azt, hogy a virtuális elismerések értéktelenek lennének – mindössze meg kell tanulnunk ezeket is a valós értéküknek megfelelően szemlélni.



Kérdések

1. Hányan fejezik ki tetszésüket ismerőseink közül egy-egy megosztásunkkal kapcsolatosan az említett egyszerű módon, és hányan teszik meg ugyanezt gondolataikat hozzászólásként megfogalmazva? Hányan mondják el érzéseiket élőszóban? Számunkra melyik elismerés ér többet? Miért?

Láthattuk tehát, hogy szeretünk jutalmakat, elismeréseket kapni, még ha csekély jelentőségűeket is. Alighanem ezzel is összefüggésben áll, hogy vannak, akiknél a tartalmak közzététele, posztolása (az angol *post*, azaz 'elküld' szóból) már-már rutinná vagy rossz szokássá, kényszerré válik.

Amit nem fényképeztünk le, az nem is történt meg?

Jártunkban-keltünkben újra meg újra látjuk, hogy valaki épp fotót készít. Amióta gyakorlatilag mindenkinél és minden pillanatban ott van egy fényképezőgépként is használható telefon, hihetetlen mennyiségű fotó készül. Ma már nehéz elképzelni, hogy négy-öt nemzedékkel ezelőtt csak akkor készült valakiről fotó, ha fényképészhez ment – azaz sokakról sohasem. Még a szüleink gyerekkorából is csak szekrények, dobozok rejtette papírképek tanúskodnak.

Kérdések, feladatok

1. Hol és mikor szabad másokat lefényképeznünk? Szabad-e megosztanunk másokról készült fotókat? (Segít a hatodikos könyv.)
2. Hány fotó van a legnépszerűbb képmegosztó oldalon? Mennyi kerül oda másodpercenként?
3. Hány magunk készítette képet nézünk meg egyetlen alkalomnál többször? Mekkora az ilyen képek aránya az összes képünkhöz képest?
4. Sokszor látjuk azt, hogy emberek egy-egy látnivalónál, eseménynél megállnak, elkészül a kép, és már lépnek is tovább. Vajon van idő az ilyen esetekben megélni az élményt? Miért készülnek el ezek a képek?
5. Tudunk-e olyan helyekről a világban, amelyekhez csak azért utaznak el emberek, hogy elkészítsék „a tökéletes képet”?
6. Hány képet néztünk meg az előző három napban? Hányra emlékszünk?
7. Volna-e esélyük a régi korok vagy épp a múlt század híres festőinek a mai körülmények között híressé válniuk?
8. Ki volt Robert Capa? Vajon lennének-e ma követői?
9. Ismerjük-e a Fortepan.hu vagy a Kezai.hu weboldalt? Keressünk tíz-tíz képet egy-egy témáról: a városunkról, az autózásról, a színházakról, a boltokról, a divatról! Mutassuk be a képeket a tanulócsoporthoz, és meséljük el, miért ezeket választottuk!



Az információ évezredei

Néhány leckével ezelőtt kimondtuk, hogy az információ érték, és láttuk azt is, hogy a megszerzésére vagy az ellopására, illetve a megőrzésére a régi korokban ugyanúgy jelentős erőfeszítéseket tettek az emberek, mint napjainkban. Azt is látjuk, hogy az emberiség és az egyes emberek előtt álló mai kihívások némelyike épp a ránk zúduló, a történelemben megközelítőleg sem tapasztalt mennyiségű, hihetetlen gyorsan és változatosságban érkező információözönnel kapcsolatos.

A számítógépek nagyon rövid története

Az információ csak akkor érték, ha értelmezni tudjuk, ha fel tudjuk dolgozni. Van olyan információ, amelynek értelmezését, feldolgozását egy pillanat alatt megtesszük – például azonnal értelmezhető, hogy az ellenség hajnalban támad, vagy hogy elmarad az utolsó óra. Más információk tömegével érkeznek, és már nagyon korán felmerült az igény olyan eszközökre, amelyek ilyenkor lehetnek a segítségünkre. Ma az ilyen eszközöket tekintjük a **számítógépek elődeinek**.

Az első ismert számolóeszközök az abakuszok. Sok évezred óta használnak abakuszt az emberek, és csak az elmúlt évtizedekben tűnt el majdnem teljesen, amióta az elektronikus számológépek nagyon olcsókká váltak.

Wilhelm Schickard az 1600-as évek első negyedében tervezett mechanikus számológépet a kor jelentős csillagásának, Johannes Keplernek, hogy segítse a munkáját. Fogaskerekes gépével a leírások szerint mind a négy alapműveletet el lehetett végezni.

Blaise Pascal 1642-ben készített újabb mechanikus számológépet. Szintén fogaskerekekkel működik, és automatikus helyiértékátvitel-képesséssel rendelkezik, de csak összeadni és kivonni lehet vele. A jelen idő használata helyes a mondatban, mivel ebből a gépből néhány napjainkra is fennmaradt. Ez volt az első számológép, amelyből viszonylag sok, körülbelül ötven darab készült.

Még a század vége előtt elkészült Gottfried Wilhelm Leibniz első számológépe. Pascal gépét alakította át úgy, hogy szorozni és osztani is tudjon. A második (és utolsó) gépe is elkészült 1720-ra. Jó ötven évvel később javítani küldték a Göttingeni Egyetemre, ahol elfelejtkeztek róla, hogy csak újabb száz év múlva találják meg egy padlásán. Restaurálták, és lassan százharminc éve a Leibnizről elnevezett Alsó-szászországi Országos Könyvtárban csodálható meg.



► Abakusz



► Pascal mechanikus számológépe

1851 és az első világháború között gyártották Thomas de Colmar négy alpműveletes számológépét. Az igazi újdonság, hogy a számológép a kereskedelemben kapható volt, bárki vásárolhatott magának, feltéve, hogy volt rá pénze. Körülbelül ötezer példány készült belőle az évtizedek alatt. Mire a gyártása megszűnt, már többféle, sikerebb típus is kapható volt a világpiacon.



► Jacquard szövőszékének programja – az információt a lyukak hordozzák



► Neumann János az USA-ban készült bélyegen

A teljesen elektronikus számítógépek története a második világháború alatt kezdődött, amikor többek között az atombomba kifejlesztéséhez szükséges számításokat is szívesen végeztették volna ilyen eszközökkel. A gépek első generációjának tagjai elektroncsövekkel működtek, teremnyi méretűek és több tíz tonna tömegűek voltak. A létrehozásukon fáradozó talán legjelentősebb szakember a Magyarországról kivándorolt amerikai tudós, **Neumann János** volt. A mai számítógépek nagy része is az általa lefektetett elveket követve készül.

A második generációs gépek mérete a múlt század '60-as éveinek első felében, a kor jelentős újításának, a tranzisztornak köszönhetően számottevően csökkent (bár több közülük még mindig szekrény méretű és több tonnás volt), megbízhatóságuk pedig nőtt (akár napokig is működtek meghibásodás nélkül). Áruk akkori dollárban mérve is az egymilliót közelítette, és sok mérnök dolgozott egyetlen gépnek a kiszolgálásán. Ekkoriban jelentek meg az első, a mai programozási nyelvekre hasonlító, magas szintű nyelvek.

A számítógép-programozás jelentős előzményeként értékeljük Joseph Marie Jacquard 1805-ben elkészült szövőgépét. A szőtt minták „programját” – azaz a szövőszék beállításait – lyukkártyán tárolta (ahogy százharminc évvel később a számítógépek is), hogy ne kelljen újra meg újra kézzel beállítani az egyes mintákat.

Charles Babbage-nek az 1800-as évek első felében megálmodott programozható számológépéből működő változatot csak sok évtizeddel a halála után tudtak építeni, de több alapelve máig meghatározó a számítógépek tervezésekor. A képzeletbeli gép programozásának leírását Ada Lovelace készítette el, akire ma az első ismert programozóként tekintünk – nem csak a nők között.

A 19. század végén készült el Herman Hollerithnek az a gépe, amellyel az USA népszámlálási adatait évek helyett hetek alatt fel tudták dolgozni. Hollerith cégéből alakult ki később a 20. század második felében az információs társadalom fejlődésére döntő hatással lévő IBM.



- ▶ Az IBM 7070-es sorozatú számítógép részlete – a lényegi rész a szekrényekben van. A jobb oldali asztal három világos téglalapja egy-egy lyukkártya, rajtuk a gép programjával



- ▶ ZX Spectrum 1982-ből. A gép akkora, mint egy kisalakú iskolai füzet, a memóriája 48 kB. Hányszor ennyi memória van egy mai telefonban? Keressünk ilyen gépen futó játékokat bemutató videót!

A harmadik generációban a milliárdnyi tranzisztort magába sűrítő integrált áramköröknek, azaz IC-knek köszönhetően a gépek mérete látványosan csökkent, a megbízhatóság pedig tovább nőtt.

A negyedik generációban, nagyjából ötven éve jelent meg a mikroprocesszor – az olyan processzor, amelyet lényegében mind a mai napig használunk (persze a maiak több százegyszer gyorsabbak). A számítógépek egyre gyakoribbá váltak az otthonokban és a munkahelyeken.

1982-ben elkészült az a később Magyarországon is elterjedt két számítógép (a ZX Spectrum és a Commodore 64), amelyeknek köszönhetően nálunk is megismerkedtek az emberek a viharos sebességgel teret hódító számítógépes játékokkal.

Ma a számítógépek ötödik generációjával élünk egy korban. A kor bő harminc éve tart, és nem a technológiai fejlődéssel határozzuk meg, hanem a számítógépek használatának módjával. Sok-sok, túlnyomó többségében hálózatba kötött számítógépet használunk szinte minden percben és szinte minden feladat elvégzéséhez. Egyre több feladat ellátását segíti mesterséges intelligencia.

Kérdések, feladatok

1. Nézzünk meg a fent említett eszközökről szóló videókat!
2. Volt-e a szüleinknek, nagyszüleinknek negyedik generációs gépe? Kinél van otthon viszonylag öreg, húsz-harminc-negyven éves számítógép? Láttuk-e őket működni?
3. Bár nehéz összevetni, de érdekes számokat kapunk, ha az után kutakodunk, hogy mennyivel gyorsabb vagy mennyivel több memóriát tartalmaz a telefonunk az egyes generációk néhány híres számítógépénél. Keressünk ilyen adatokat!

Mi vár ránk a jövőben?

Információs társadalmunkra sok és sokféle érdekes kihívás vár már a közeli jövőben is. Első helyen a **környezetterhelést** említhetjük. Volt már róla szó, hogy az elektronikai hulladék újrahasznosításának megoldása távol jár az optimálistól, és Európában nagyon sok elektronikai hulladék keletkezik. A számítógépek (ideértve mindent, amiben valamilyen értelemben számítógép működik) anyagainak bányászata és alkatrészeinek gyártása is rendkívül környezetszennyező. Bár a gyártásból eredő szennyezés Európában nem látványos – az

ásványkincs kevés, a gyártás túlnyomórészt nem itt történik –, a termékek nagyon jelentős fogyasztójaként bennünket is felelősség terhel, a megoldás kereséséből pedig ki kell vennünk a részünket.

Szintén komoly kihívást jelentenek a jövő tekintetében a **technológiai korlátok**. A mikroprocesszorok mérete tovább már nem nagyon csökkenhet. Nem arról van szó, hogy az integrált áramköreinkben nem tudjuk jobban összesűríteni a második generáció idejében még egyesével készített tranzisztorokat, hanem hogy a fizikai korlátok miatt nem is lehet. A határt a kvantumszámítógépek kialakításával próbálják legyőzni. Ha minden az elképzelések szerint alakul, ezek a számítógépek a mostaniak számítási teljesítményének sokszorosával dolgoznak majd. A gyorsaságnál is jelentősebb különbség talán, hogy a kvantumszámítógépek belső működése alapvetően eltérhet a mostani gépeketől.

Láttuk, hogy hihetetlen mennyiségű adat és számítási kapacitás kerül a **felhőbe**. A nagy szolgáltatók egymással versengő tempóban építik az újabb és újabb adatközpontokat. Lehet, hogy a földrajzilag szétszórta működő gépek helyett megszámlálható mennyiségű helyre kerül a világ számítási kapacitását jelentő számítógépek nagy része. Ez segítségünkre lehet a környezetünket érő károkozás csökkentésében. Van olyan nagy cég, amely a terveit szerint az adatközpontjainak áramellátását teljes egészében megújuló energiaforrásokra állítja át a következő egy-két évben. Mások a sarkkörön túlra helyezik az adatközpontjait. Bár a gépek itt sem fogyasztanak kevesebb áramot, de messze nem kell annyi energiát használni a gépek hűtésére, ami sok millió számítógép esetén jelentős árammegtakarítással járhat.

Gyakran látjuk, hogy az emberiség számára hosszú idő, mire megtanulja egy-egy új eszközt felelősen használni. A korábban soha nem látott mennyiségben előállított, gyűjtött és feldolgozott **információ felelős használatát** sokáig kell még tanulnunk. Az információ gyűjtését, felhasználását egyre több rendelkezés szabályozza. Más a helyzet a **mesterséges intelligenciával**, amelynek fejlődése az elmúlt évtizedben vett egyre gyorsuló, új lendületet. A lehetőségek itt is beláthatatlanok, és először 2021 áprilisában nyújtott be tervezetet az Európai Bizottság a mesterséges intelligenciával kapcsolatos szabályok megalkotására.



Feladatok

1. Milyen egyéb kihívások állhatnak még az információs kor társadalma előtt? Végezzünk kutatómunkát, és írjunk rövid fogalmazást a fűzetünkbe!
2. Készítsünk egy, az előzőekben tárgyalt témáról, részterületről bemutatót, és mutassuk be az osztálytársainknak!

Séta a földi hálózatokban és a felhőben

Hogyan működik az iskolai hálózat?

Az iskolában hálózatba vannak kötve a gépek – éppen úgy, ahogy az egyes cégeknél, hivatalokban, kórházakban, vagyis szinte minden kicsit nagyobb intézményben. Az ilyen hálózatokban alapvetően kétféle számítógépet különböztünk meg.

Az egyik típusba sok gép tartozik, az intézményben lévő számítógépek nagyobbik része. Ezekkel a gépekkel dolgozunk a tanítási órákon, de ilyet használnak a tanárok és az iskolában dolgozó többi munkavállaló is. E gépek neve **kliens** – a szót ebben a szöveggörnyezetben az ügyfél szóval fordíthatjuk magyarra, máskor pedig munkaállomásnak hívjuk ezeket.

A másik típusba kevés, esetleg egyetlen gép tartozik az intézményben, ezeket szervernek – magyarul kiszolgálónak – nevezzük. A szerverek az iskolában általában legalább három feladatot látnak el. Mindeközben az iskolai számítógépes hálózaton kommunikálnak, „beszélgetnek” a kliensekkel és egymással, de ezen a hálózaton az ügyfelek közvetlenül is tudnak egymással kommunikálni.



► Számítógépes hálózat

Kérdések

1. Miért kötjük hálózatba a gépeinket? Milyen hálózattal kapcsolódnak a tanterem gépei az iskolai hálózathoz: vezetékessel vagy vezeték nélkülivel?
2. Melyik kábel a gépek hálózati kábele? Hova, milyen eszközbe csatlakoznak ezek a kábelek?
3. Milyen eszközhöz csatlakoznak a gépek a vezeték nélküli hálózat használata esetén? Hol van elhelyezve ez az eszköz?
4. Melyik hálózathoz hasonlíthatjuk a főként okostelefonokon használt mobilinternetes kapcsolatot?
5. Gondoljuk végig, mi minden történik egy számítógép vagy egy mobilkészítők bekapcsolásakor! Vajon mikortól éri el a számítógépünk a hálózatot?

A szerverek feladatai

A szerverek első feladata eldönteni, hogy ki léphet be a kliens számítógépekre. Az esetek túlnyomó többségében a klienseken megadjuk a felhasználóneveinket és a jelszavunkat, majd a kliens továbbítja ezeket a szervernek. A szerver eldönti, hogy jó jelszót adtunk-e meg, és ha igen, akkor elfogadja, illetve szól a kliensnek, hogy beengedhet bennünket –

azaz bejelentkezhetünk a kliensre. A szerver első feladata tehát a hálózati hozzáférés engedélyezése.

A másik két feladat az iskolai hálózaton elérhető erőforrások használatával kapcsolatos. Az első ilyen erőforrás a megosztott hálózati meghajtókat, hálózati mappákat jelenti. Egy hálózati megosztás nem más, mint egy mappa a szerver számítógépen, amelyet a hálózaton megosztva a szerverhez csatlakozó számítógépek felcsatlakoztathatnak magukra. A felcsatolt megosztást, megosztásokat a kliensekre bejelentkezett felhasználó használatba veheti – olvashat róluk fájlokat, vagy akár írhat is rájuk, attól függően, hogy mire van jogosultsága. A hálózati megosztáson lévő fájlok a valóságban a szerver háttértárán találhatóak. Amikor megnyitjuk őket, átutaznak a hálózaton, és rendelkezésünkre állnak. A programjaink nem tudják, hogy egy ilyen fájl a számítógépünkön van-e, vagy a hálózaton érkezett – a fájlok megnyitása és mentése az operációs rendszer feladata.

Az iskolai szerver harmadik alapvető feladata általában az iskola nyomtatóinak elérhetővé tétele a hálózaton – így nem csak egy számítógépről érhető el ez az erőforrás. Elképzelhető, hogy ugyanez a szerver teszi elérhetővé az iskola internetkapcsolatát a többi gép számára. Jobb esetben ezt a feladatot külön eszköz látja el.

Érdeemes tudnunk, hogy a szerver számítógépek nem feltétlenül különleges számítógépek – egy iskolai szerver feladatát szinte biztosan el tudná látni bármelyik nem túl régi laptop is. A szerver attól lesz szerver, hogy olyan szoftvereket telepítünk rá, amelyek lehetővé teszik, hogy ellássa szervert szerepét – példánkban a felhasználók azonosítását, a hálózati meghajtók és a hálózati nyomtató elérhetővé tételét.

Kérdések

1. Milyen hálózati megosztás található a tantermi számítógépeinken?
2. Melyik megosztáson mire van jogosultságunk?
3. Hol van az iskolai szerver elhelyezve? Ha van rá mód, nézzük meg! Mennyiben tér el az iskola máségeitől?
4. Hogyan szokás olyan esetekben azonosítani a felhasználókat, amikor nem felhasználónév és jelszó megadásával teszik?

Vigyázzunk a jelszavunkra! Ha valaki megtudja, és a felhasználásával rosszat cselekszik, azt a mi nevünkben teszi, és ez ránk vet rossz fényt, bennünket hoz kellemetlen helyzetbe. Ha a mi tudomásunkra jutna valakinek a jelszava, akkor sem használhatjuk! Ahogy másnak a házába akkor sem mehetünk be hívás nélkül, ha találtunk hozzá kulcsot, vagy nyitva hagyták az ajtót, úgy ezzel a helyzettel sem szabad visszaélni. Szóljunk a jelszó tulajdonosának, hogy változtassa meg a jelszavát, mert mi már ismerjük!

Hogyan működik a felhő?

Az iskolai hálózat ismeretében könnyebb elképzelnünk az internet, a felhő működését. Nagyon leegyszerűsítve, az internet egy olyan hálózat, amelyhez rengeteg kliens és szerver eszik – azaz a „felhőt” alkotó eszközök sokasága – kapcsolódik. Kliensek lehetnek például az iskolai, otthoni számítógépek, okostelefonok, autók GPS-készülékei, időjárás-érzékelők, pénztárgépek, vonatok, internetkapcsolattal rendelkező fényképezőgépek és kamerák. A szerverek pedig azok a számítógépek, amelyek ezúttal nem néhány mappát, nyomtatót szolgálnak ki a

klienseknek, hanem weboldalakat, videókat, képeket, amelyekhez a klienseken futó GPS-ek, üzenetküldő alkalmazások, pénztárgépek vagy internetes játékok kapcsolódnak.

A szerverek most is attól lesznek szerverek, hogy olyan alkalmazást telepítenek rájuk, amelyet futtatva betölthetik a szerver szerepét. A sokak által látogatott szerverek ugyanakkor általában komolyabb hardveren futnak, mint az iskolai szerver. Az ilyen gépek arra vannak felkészítve, hogy akár sok éven át megszakítás nélkül, teljes kihasználtsággal működjenek. Sokszor különleges termekben, úgynevezett **adatközpontokban** helyezik el őket.

Kérdések, feladatok

1. Egy webszerver attól lesz webszerver, hogy webszerver-alkalmazást telepítünk rá. Melyek a leggyakrabban használt webszerver-alkalmazások? Milyen operációs rendszeren futnak? Hogy hívjuk azokat az alkalmazásokat, amelyekkel a webszerverekhez csatlakozunk? Nézzük meg egy webszerver telepítését akár videóról, akár tanári bemutatással!
2. Keressünk az interneten olyan videót, amely szerver számítógépet mutat be! Miben tér el az ilyen gépek felépítése az általunk használtakétól?
3. Keressünk az interneten olyan videót, amely adatközpontot mutat be! Hol vannak ilyen épületek hazánkban?

Felhőbeli identitásunk védelme

A felhőszolgáltatók alkalmazásai – a közösségi oldalaktól a videómegosztókon keresztül a chatszerverekig – általában ugyanúgy felhasználónévvel és jelszóval azonosítanak bennünket, mint az iskolai számítógép-hálózat. Az igazán nagyok lehetőséget nyújtanak a bonyolultabb azonosításra is, amelyet **két- vagy többfaktoros azonosításnak** nevezünk. Ilyen esetekben általában a jelszavunkon felül egy másik azonosítót is meg kell adnunk. Ez lehet például egy SMS-ben kapott vagy egy telefonos alkalmazásban megjelenő kód. A másik azonosító csak egy-két percig él, vagy csak egyszer használatos, és a jelszavunkkal együtt azonosít bennünket.

Kérdések, feladatok

1. Soroljunk fel felhőalkalmazásokat! Beszéljük meg közösen, ki melyiket használja! Milyen feltételei vannak a felhőalkalmazások használatának? Mit tudunk a felhőalkalmazásokba való regisztráció magyarországi jogi hátteréről? (Segít a hatodikos könyv.)
2. A felhőtárhelyeken tárolt fájloknak általában kétféle megosztása lehetséges: megoszthatjuk őket konkrét felhasználókkal vagy mindenkivel. Ha módunkban áll, próbáljuk ki a kétféle megosztást!
3. Beszéljük meg, hogy mi a kétfaktoros azonosítás előnye és hátránya!
4. Derítsük ki, hogy mi az az identitásslolgáltató! Használtunk-e már ilyet? Mi a használat előnye és hátránya?

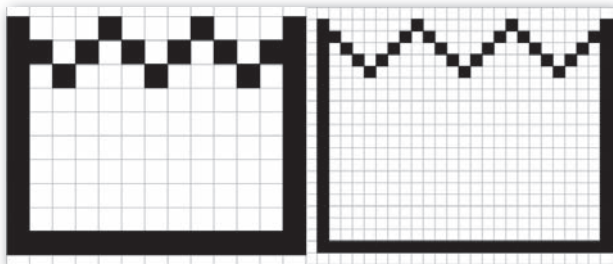
Képek és térképek

Hogyan készülnek a digitális képek?

Manapság digitális képek készítésére elsősorban okostelefonokat használunk, de az alábbiakban ismertetett jellemzők ugyanígy érvényesek a digitális fényképezőgépekre és a kamerákra – a kép tárolásáról írtak pedig a rasztergrafikus rajzolóprogramokra is.

Amikor exponálunk, fény érkezik a lencse mögött, a telefon vagy a fényképezőgép belsőjében elhelyezkedő fényérzékeny területre. Ez a terület oszlopokra és sorokra van osztva, mindegyik helyen egy-egy apró érzékelővel. A kép digitalizálásakor az érzékelő feljegyzi, hogy hányas kódú szín jutott rá (például egy RGB-kóddal), és a fényképezőgép ezeket a színekódokat egymás után felírja egy fájlba.

Az, hogy hány színekód kerül a fájlba, elsősorban attól függ, hogy mekkora a kamera fényérzékelőjének a felbontása. Nézzük meg például az alábbi két képet! Az elsőnek kisebb a felbontása, a másodiknak nagyobb.



Ha a fekete színt nulla jelöli, a fehéret pedig egy, akkor a bal oldali kép felső sorának kódja: 0111011101110, a jobb oldalié pedig 0111111101111111011111110. A nagyobb felbontású kép rögzítése sokkal több jelet igényel. A teljes bal oldali koronát százharminc számmal tudjuk leírni, míg a jobb oldali pontosan ötszázzal. A digitalizálás kifejezés jelentése: leírhatóvá alakítás számokkal. A digitális szó tehát itt azt jelenti, hogy „számokkal leírt”.

Látjuk, hogy a kisebb felbontású kép nem olyan szép. A nagyobb felbontású, bár szebb a szemnek, látványosan nagyobb méretű. Ha figyelembe vesszük, hogy egy manapság készített kép 16 777 216 színt különböztet meg, ahol egyetlen képpont színekódja is huszonnégy darab egyesből és nullából álló jelsort jelent, akkor az eltérés a fenti két korona esetében még látványosabb. A bal korona leírásához egy $130 \times 24 = 3120$ tagú számsor kellene, a jobb oldaléhoz pedig egy $500 \times 24 = 12\,000$ tagú.

Kérdések

1. Mi az az RGB-kód?
2. Milyen mértékegységben adjuk meg egy telefon kamerájának vagy egy fényképezőgépnek a felbontását? Mit fejez ki ez a szám? Mekkora a mobilkészülökünk kamerájának a felbontása? (Az eszköz típusát megadva megkereshetjük az interneten is.)
3. Vajon meddig érdemes növelni a kamerák felbontását? Meddig érdemes növelni a televízió-képernyők, monitorok felbontását?
4. A fentiek alapján hány tagú, nullából és egyesekből álló számsor szükséges a kameránk által készített kép leírásához?

Vannak olyan képformátumok, amelyek lényegében így, az egyes képpontok színkódjának felsorolásával tárolják a képeket a háttértáron. Ilyen adat van például a .bmp kiterjesztésű fájlokban. A legtöbbször azonban tömörítve tároljuk az adatainkat, legyen szó akár képekről, akár más adatokról.

Mi az a tömörítés, és hogyan működik?

Akár hisszük, akár nem, megoldható az, hogy az említett jelsorozatokat információtartalmát – csakúgy, mint a legtöbb jelsorozatát – kevesebb jellel is leírjuk. Mindez nem boszorkányság, hanem számítástudomány. A legegyszerűbb módszer lényege mindössze annyi, hogy ismétlődéseket keresünk, és megmondjuk, hogy hány ismétlődő jelsor követi egymást. Ha például beütjük a kezünket, nem mondjuk el nyolcvanhétszer, hogy „jaj-jaj-aú”, hanem csak annyit mondunk, hogy „nyolcvanhét jaj-jaj-aú”. És ez kevesebb helyet foglal. Az ilyen tömörítés sajátossága, hogy a tömörített forma alapján pontosan vissza tudjuk állítani az eredeti jelsorozatot, azaz a **tömörítés veszteségmentes**. Így tömörítjük a szövegeinket, táblázatainkat, a programjainkat, és sokszor a képeket, hangokat, néhány esetben pedig a videókat is. Veszteségmentesen tömörített képet tárolnak például a .gif és a .png kiterjesztésű fájlok, veszteségmentesen tömörített hang van például a .flac kiterjesztésűekben. A .zip kiterjesztésű tömörített fájlok manapság többnyire az operációs rendszer segédprogramjával készülnek, és bármilyen adatot tárolhatnak – veszteségmentesen.



Sokkal hatékonyabban tömöríthetünk adatokat, ha csak az eredetihez hasonlót akarunk visszakapni. Az ilyen, **veszteséges tömörítéssel** tárolt képeken, hangokban kihasználjuk az emberi látás és hallás véges érzékenységét, és óvatosan megszabadulunk olyan adatoktól, amelyeket úgysem látnánk vagy hallanánk. A veszteséges tömörítésekkel lényegesen kisebb fájlokat kapunk, mint a veszteségmentes eljárásokkal. Ilyen adatot tárolunk a .jpg kiterjesztésű képekben, az .mp3 vagy az .ogg kiterjesztésű hangfájlokban. Az elvesztett információ mennyisége szabályozható. A balra látható lufis kép viszonylag jó minőségű. A jobb oldali tömörítések több információt hagyunk el. A fájl sokkal kisebb, a képminőség sokkal gyengébb így.

A mozgóképek tömörítésekor általában nemcsak annyit teszünk, hogy a veszteségesen tömörített képek sorozatát tároljuk, hanem arra is figyelünk, hogy ha egyforma rész van az egymást követő képeken, arra ne pocsékoljuk a helyet. Ha például egy percig rögzítjük,

hogy egy pók ereszkedik le a fal előtt, akkor a legtöbb egymást követő képen nem tároljuk a változatlan falat, csak a változó pókot – így spórolunk a tárterülettel, és így a hálózaton is gyorsabban le- vagy áttölthető a videó.

Kérdések, feladatok

1. Veszteségesen csak képet, hangot és videót tömörítünk. Vajon miért?
2. Készítsünk vagy keressünk az interneten egy nagy, .bmp kiterjesztésű képet! Jegyezzük fel a méretét, majd tömörítsük veszteségmentesen, például .png kiterjesztésű fájlba! Mekkora a méretcsökkenés?
3. Amikor egy rasztergrafikus programmal JPG-formátumba mentünk, beállítható, hogy milyen minőséget szeretnénk. Mekkora lesz az előző fájl, ha 90%-os, 70%-os, 50%-os beállítással mentjük? Hol válik látványossá a minőségromlás? És ha nem a monitoron, hanem kivetítőn nézzük?
4. Mit nevezünk adatnak, információnak, hírnek? Hogyan változik veszteségmentes és veszteséges tömörítéskor a tárolt adat, illetve a tárolt információ mennyisége?
5. Melyik felhőalkalmazással tárolhatunk képet, hangot és videót? Melyikkel könnyű úgy tárolni, hogy könnyen megosztható legyen? Melyikkel tudjuk úgy tárolni, hogy csak azok nézhessék meg, akikkel megosztjuk?
6. Ismerünk-e olyan megosztási lehetőséget, ahol többnyire művészi igénnyel készült alkotásokat osztanak meg?
7. A tömörítést nem csak azért alkalmazzuk, mert helyet szeretnénk spórolni. A rövidebb jelsorozatok, a kisebb fájlok az interneten gyorsabban célba érnek. Indítsunk internetes hanghívást vagy videóhívást egyik eszközről a másikra vezeték nélküli kapcsolat használatával! Ha nem áll módunkban, akkor kezdjük el egy nagy felbontású videó megtekintését az internetes videómegosztók valamelyikéről! Menjünk a készülékünkkel olyan helyre, ahol romlik a vételi lehetőség, vagy óvatosan vegyük körül a készüléket fémhállóval, illetve alufóliával! Mi tapasztalunk a kép-, illetve a hangminőség tekintetében? Indokoljuk a tapasztalatunkat!

A vektorgrafikus képek tárolása

A vektorgrafikus rajzok esetében a képek tárolása alapvetően más. Nem képpontokat, hanem objektumokat – köröket, sokszögeket, görbéket – és azok jellemzőit – színüket, vastagságukat, átlátszóságukat – tároljuk. Nem azt mondja el a fájl, hogy hol vannak például egy vonal pontjai, hanem azt, hogy „rajzolj vonalat az ilyen és ilyen koordinátájú pontok közé”. Az egyszerűbb alakzatoknak a fontos pontjait, a körnek a középpontját és a sugarát tároljuk, a görbéket pedig bonyolultabb függvényekkel írja le a vektorgrafikai elemeket tároló fájl.

Mint ahogy lényegében nem a kész ábrát, hanem az ábra megrajzolásához szükséges információkat tároljuk, a számítógép az ábrát erősebb ráközelítés esetén is úgy tudja újrajzol-



ni, hogy az nem pixelesedik. Az ábra objektumai utólag egyszerűen módosíthatók. Könnyen megváltoztatható az, hogy melyik objektum fedi el a másikat. Ha pedig háromdimenziós koordináta-rendszert használunk, térbeli objektumok kiterjedésének, helyzetének megadására is módunk nyílik.

Digitális térképek

Az egyik olyan felhőalkalmazás, ahol nem beszélünk képmegosztásról, de rengeteg képet is tárol, az az internetes térképek világa. Az ilyen alkalmazások folyamatosan frissítik a tárolt adatokat, így tartva lépést például az úthálózat változásaival. Tehetünk velük virtuális sétákat – az ál-háromdimenziós környezet ábráját sok állóképből rakják össze számunkra.



Kérdések, feladatok

1. Hogyan kerülnek a fényképek az ilyen alkalmazásokba?
2. Keressük meg a házunkat, iskolánkat, városunk nevezetes helyeit! Keressük meg híres események, csaták helyszínét! Van-e lehetőségünk háromdimenziós sétára? Becsüljük meg, hogy mikor készültek a képek!
3. A térképalkalmazások mobilszközünkben a GPS-jelek segítségével találják meg a mindenkori pozícióinkat. Mutassuk be bemutatón a GPS működését!

A szószedet az 5–8. évfolyamos tankönyvsorozat legfontosabb informatikai szakkifejezéseit és a digitális kultúra világában leggyakrabban használt fogalmakat tartalmazza. A meghatározásokban szerepelő *dólt betűs* kifejezések magyarázatát lásd az adott fogalomnál.

adat: Tények, fogalmak olyan megjelenési formája, tulajdonsága, amely alkalmas eszközökkel történő értelmezésre, feldolgozásra, továbbításra.

adathalászat: Olyan tevékenység, amelyben a támadók hamis üzenettel próbálják rávenni a felhasználót arra, hogy érzékeny (például személyes, banki) adatokat adjon meg.

adathordozó: Adatok tárolására alkalmas eszköz. Ilyenek például a *háttértárak*, de adathordozó lehet egy papírlap vagy egy üzenetet tartalmazó agyagtábla is.

alaplap: Az alaplapon a *mikroprocesszor*, a *memória* és a számítógép működéséhez elengedhetetlen áramkörök, valamint a számítógép bővítéséhez szükséges csatlakozók találhatók.

analóg: Az analóg jelek folytonosan változnak, szemben a digitális jelekkel, amelyek csak adott lépésközzel növekedhetnek vagy csökkenhetnek.

Android: Mobilkészülékek (például okostelefonok és táblagépek) népszerű *operációs rendszere*. Szabadon használható, *nyílt forráskódú*.

ATX (Advanced Technology eXtended): A számítógépházakra és alkatrészekre vonatkozó szabvány. Megadja többek között a számítógép legfontosabb geometriai jellemzőit és egymáshoz történő csatlakoztatásuk módját.

behúzás: A *bekezdés* és a függőleges margók közötti távolság.

bekapcsolás folyamata: A számítógép elektronikus alkatrészeinek indításának és az *operációs rendszer* betöltődésének folyamata. A felhasználónevünk és jelszavunk megadása után indíthatjuk el azokat a programokat, amelyeket használni szeretnénk. Azokat a *perifériákat*, amelyeket nem kívánunk használni (például nyomtató, szkener, hangfal), nem kell bekapcsolni.

bekezdés: A szöveg összefüggő, önálló gondolatot hordozó része. Szövegszerkesztő alkalmazá-

sokban a *bekezdés* az Enter billentyű két leütése között található szövegrész.

beviteli periféria: A számítógép központi egységéhez kívülről csatlakozó eszköz (*billentyűzet*, egér, mikrofon, szkener stb.), mely az adatok bevitelét teszi lehetővé.

billentyűzet: A számítógép beviteli eszköze (*beviteli periféria*). Segítségével *adatokat*, illetve utasításokat viszünk be a számítógépbe. További elnevezései: keyboard, klaviatúra, tasztatúra.

bit: Az *információ* alapegysége. Egy igen-nem (1 vagy 0) választást tesz lehetővé. Egybitnyi információt képvisel a kettes számrendszer egy-egy számjegye.

bitcoin: Elektronikus fizetőeszköz, kriptovaluta.

böngészés: A *weboldalak* megtekintése, más szóval szörfözés.

böngésző: Olyan program, amellyel böngészhetünk, webes tartalmakat letölthetünk a saját számítógépünkre.

byte: Egy byte 8 bit. Egy byte-on – kettes számrendszerben – 0 és 255 közötti, összesen 256-féle számértéket tudunk tárolni.

Captcha-kód: Biztonsági kód, amelyet a *webhelyek* felhasználóinak kell beírniuk egy-egy űrlapon. A webhelyen működő szoftver ennek segítségével bizonyosodik meg arról, hogy valóban egy ember tölti ki az adatokat, nem pedig egy program.

cella: A táblázatkezelő programban az adatokat mindig cellákba írjuk. Minden cella egy sor és egy oszlop metszéspontjában található. A cella azonosításához oszlopának betűjelét, majd sorának sorszámát adjuk meg (például C7-es cella).

chat: Magyarul csevegés; több felhasználó között zajló online beszélgetés. Webes szolgáltatásokon zajlik szöveges üzenetek vagy videómegbeszélések formájában. Az utóbbit *videóchat*nek nevezzük.

cyberbullying: Internetes zaklatás. A bántalmazó lehet egy személy vagy egy csoport. Zaklatást követ el, „aki abból a célból, hogy mást megfélemlítsen, vagy más magánéletébe, illetve mindennapi életvitelébe önkényesen beavatkozzon, őt rendszeresen/tartósan háborgatja” (Btk. 222. §). Akár három évig terjedő szabadságvesztéssel büntethető.

csoporthatás eszközei: Olyan alkalmazások, amelyek segítik, hogy a csoport tagjai együttműködhessenek, közösen alkothassanak. Például a Microsoft Office lehetővé teszi a OneDrive-ra feltöltött dokumentumok, prezentációk és táblázatok közös szerkesztését. Ehhez hasonló módon a Google Drive is lehetőséget kínál a közös munkára.

diagram: Grafikus formában mutathatjuk be rajta az adatok változását vagy egymáshoz való viszonyát.

digitális számítógép: A digitális szó a *digit* szóból származik, amelynek jelentése: számjegy. A digitális számítógépek két számjegyet ismernek: a nullát és az egyet. Minden beérkező információt először csak nullákból és egyesekből álló számokká alakítanak. Ezekkel a számokkal műveleteket végeznek. A műveletek eredményét az ember számára értelmezhető formában jelenítik meg.

digitalizálás: Analóg (folytonosan változó) jelek digitális jelekké alakítása. A digitális eszközök két számjegyet ismernek: a nullát és az egyet. Kettes számrendszerben azonban minden szám felírható. Fekete-fehér képek digitalizálásakor minden képponthoz egy számot tárolunk, amely megadja, hogy mennyire világos az adott pont. Hangok digitalizálásánál meghatározott időközökben alakítjuk számokká a hang erősségével arányos analóg jelet.

dokumentumformátumok: A dokumentumokat típusuknak megfelelően különböző fájlformátumokban tároljuk. Erre a fájlnev után ponttal elválasztva írt kiterjesztés utal, például PDF-állomány esetén a .pdf, bemutató esetén a .pptx, a .ppt vagy az .odp; szöveges dokumentumnál a .docx, a .doc vagy az .odt, táblázatnál pedig az .xlsx, az .xls vagy az .ods. A .png, .jpg, .bmp kiterjesztés képekre, a .wav, .mp3, .flac, .ogg hangfájlokra utal.

domain: Az internet egy meghatározott része, tartománya. A domainnevek (tartománynevek) használata teszi lehetővé, hogy egy-egy webkiszolgáltató címét könnyen megjegyezhesük.

e-állampolgárság: Az e-állampolgár elektronikusan intézi hivatali ügyeit, például erkölcsi bizonyítványt, rendszámtábla-cserét igényel, interneten adja meg népszámlálási adatait.

elsőfej: A felső margónak az oldal szövegtükör felfeltri része, ahol a dokumentummal kapcsolatos információkat helyezhetünk el, például a fejezet címét.

elsőláb: Az alsó margónak az oldal szövegtükör alatti része, ahol a dokumentummal kapcsolatos információkat helyezhetünk el, például az oldal számát.

e-mail: Elektronikus levél.

e-mail-cím: Az elektronikus levelezésnél használt postaláda címe (azonosítója).

ergonómia: Az ember-gép-munkakörnyezet kapcsolatát vizsgáló tudományág. Célja az ember testi és lelki tulajdonságainak leginkább megfelelő eszközök és munkakörnyezet kialakítása.

eszközkezelés: A számítógép belsejében található vezérlés, amely a külső *hardver*eszközök kezelését végzi. Elsősorban az *operációs rendszer* feladata. Bekapcsolásnál ellenőrzi a működéshez elengedhetetlen eszközök (*memória, videokártya, háttértárolók* stb.) meglétét és működőképességét. Működés közben a *beviteli perifériákról* érkező adatokat a *processzor* számára érthető formára alakítja, és előállítja a *kiviteli perifériák* által értelmezhető jeleket.

etikus információkezelés: Mások adatait, képeit, videóit csak az érintettek engedélyével szabad az interneten közzétenni. Saját műveinkben mások írásait, fotóit csak akkor használhatjuk fel, ha erre engedélyünk van. Használatkor feltüntetjük származásukat.

fájlkezelő: Az *operációs rendszer* egyik leggyakrabban használt segédprogramja. Segítségével megkereshetjük a *háttértáron* lévő fájlokat és mappákat. Megnyithatjuk vagy új helyre másolhatjuk, vagy – megfelelő jogosultság birtokában – áthelyezhetjük és törölhetjük is őket.

fejezet: A könyvek és nagyobb dokumentumok tartalma rendszerint fejezetekre van bontva. A tartalomjegyzék a fejezetek címeit és kezdő oldalszámait sorolja fel.

felbontás: A vízszintes és függőleges irányban megjeleníthető *képpontok* száma. *Monitorok, kivetítők, nyomtatók* és más megjelenítőeszközök fontos jellemzője.

felhőszolgáltatások: A felhőalapú szolgáltatásoknál adatainkat a szolgáltató cég távoli számítógépeire (szervereire) töltjük fel, programjainkat, *weboldalainkat* az ilyen gépeken futtatjuk. A felhőszolgáltatások segítségével például megoszthatjuk másokkal és közösen szerkeszthetjük a feltöltött dokumentumokat.

fórum: Közösségi oldal, ahol a felhasználók beszélgethetnek az őket érdeklő, az adott fórum jellegének megfelelő témákról.

függvény: Összetettebb számítások elvégzésére a táblázatkezelő programok függvényeket kínálnak. A SZUM függvény például a zárójelben megadott tartomány celláinak tartalmát összegezi: =SZUM(A2:B6).

GDPR (General Data Protection Regulation): Az Európai Unió általános adatvédelmi rendelete. Meghatározza a személyes adatok kezelésének szabályait.

gigabyte: Egy gigabyte = 1000 megabyte. Jelölése: GB (1 GB = 1000 MB).

hacker: Nagy tudású számítástechnikai szakember, akinek a különféle biztonsági rések megtalálása és korrigálása a feladata. Ha tudását jó célra használja, akkor etikus hackernek nevezzük, ha a tudásával visszaél, akkor fekete kalapos hackernek (black-hat hacker).

hardver: A számítógép megfogható (elektronikus és mechanikus) elemei. Idetartoznak a számítógépházban található modulok és a számítógéphez kapcsolódó be- és kiviteli eszközök is.

háttértár: A háttértárolók – a számítógép kikapcsolása esetén is – tárolják a programokat és az adatainkat. A leggyakrabban használt háttértár-típusok közé tartozik a *merevlemez* (HDD) és a szilárdtest-meghajtók (SSD).

helyi hálózat: Egy intézmény (iroda, iskola stb.) falain belül vagy esetleg egymáshoz közeli épületekben összekötött számítógépek együttese.

hír: Közölt vagy továbbított üzenet, tájékoztatást tartalmazó jelsorozat.

hivatkozás szabályai: Minden esetben, amikor mások által készített műből idézünk szövegrészeket saját dokumentumunkban, akkor azt jól látható módon jelezniük kell. Az idézett szövegrész idézőjelek közé kerül. Ezenkívül meg kell adnunk a mű címét, szerzőjét, kiadóját és azt is, hogy a mű hányadik oldalán található az idézett szöveg. Webes tartalmak esetén a *weboldal linkjét* (hivatko-

zását) kell megadnunk, és azt az időpontot is, amikor az idézett szöveget utoljára láttuk rajta. (A weboldalak tartalma változhat.)

HTML (HyperText Markup Language): Hiperszöveges jelölőnyelv. A weblapok tartalmának leírását rögzítő szabvány.

HTTP (HyperText Transfer Protocol): Hiperszöveg-átviteli protokoll. A világhálón keresztüli adatátvitel módját meghatározó szabvány. Az adatok továbbítása nem titkosított formában történik.

HTTPS (HyperText Transfer Protocol Secure): Biztonságos hiperszöveg-átviteli protokoll. A világhálón keresztüli adatátvitel biztonságos módját meghatározó szabvány. Az *adatok* továbbítása titkosított formában történik.

információ: Olyan új ismeret, amely megszerzése számára hasznos és értelmezhető. Egy *adat* annak, aki nem érti a jelentését, nem információ – annak viszont, aki érti, és szüksége van rá, igen.

inkognitó mód: Más néven privát böngészési vagy rejtett mód. Ilyenkor a *böngésző* nem tárolja a böngészési és letöltési előzmények között a meglátogatott *webhelyek* adatait és a letöltéseket. A felhasználó ekkor sem marad ismeretlen.

internet: Az egész világot behálózó számítógéphálózat.

internetfüggőség: Szenvedélybetegség, amely már nem egyszerűen sok internethasználatot jelent. Ha egy internetfüggőnek a munkahelyi, iskolai, magánéleti konfliktusai szaporodnak, és a problémák ellenére az internet köti le a figyelmét, akkor az online világ nyert.

iOS: Mobilkészülékek (például okostelefonok és táblagépek) népszerű *operációs rendszere*. Kereskedelmi célú *szoftver*, amit csak azokon az eszközökön szabad használni, amelyekre gyárilag telepítették.

IP-cím: Olyan 32 jegyű, kettes számrendszerbeli azonosító szám, amellyel az internetre kapcsolódó számítógépek bármelyikét egyértelműen azonosítani lehet. A könnyebb olvashatóság miatt ezeket ponttal elválasztott, tízes számrendszerbeli számmal írjuk le (például 84.206.104.74).

Jegyzetömb: A Windows *operációs rendszer* segédprogramja, amelynek segítségével gyorsan és egyszerűen készíthetők nem formázott szöveges dokumentumok. Mindegyik operációs rendszer kínál hasonló alkalmazást.

Jogosultságok: A számítógépek, okoseszközök és online szolgáltatások felhasználói bejelentkezéskor felhasználónév, jelszó megadásával vagy más módon, például ujjlenyomatukkal való azonosítást kérnek. Ezt követően a felhasználók hozzájuthatnak saját adataikhoz, dokumentumaikhoz, használhatják a számukra engedélyezett alkalmazásokat, amelyekhez jogosultságot kaptak. A kiemelt felhasználók, moderátorok vagy rendszergazdák a saját adataikon túl engedélyt kapnak a rendszerbeállítások, esetleg más felhasználók adatainak kezelésére is.

karakter: Rendszerint képernyőn vagy nyomtatásban megjelenített betű, szám, írásjel vagy speciális karakter (például @). A nem nyomtatható karakterek a képernyőn bár megjeleníthetők, a kinyomtatott dokumentumban nem láthatók. Ilyen például a szóköz, a tabulátor, az Enter stb.

karakterkódolás: A számítógép minden *karaktert* egy-egy számként tárol. A karakterkódolás határozza meg, hogy a képernyőn vagy nyomtatásnál egy-egy számérték helyett melyik karaktert fogjuk látni.

képlet: Ha egy táblázatkezelő programban a *cella* tartalmát egyenlőségjellel kezdjük, akkor utána képleteket írhatunk. A képletekben használhatunk számokat, matematikai jelöléseket, és hivatkozhatunk egy másik cella tartalmára.

képpont: Más szóval pixel. A monitoron megjeleníthető, illetve képállományban tárolható legkisebb egység.

kiberbűnöző: Olyan bűnöző, aki informatikai eszközöket használ áldozatai átverésére, megkárosítására vagy bántalmazására.

kikapcsolás folyamata: Kikapcsolás előtt az összes dokumentumot, amelyet szerkesztettünk, el kell mentenünk. Minden operációs rendszer rendelkezik egy kikapcsolás menüponttal vagy ikonnal. Mindig ezt használjuk! Ha a *háttértárolón* éppen írásművelet zajlik, egy áramtalanítás esetén az eszköz meghibásodhat. A számítógép kikapcsolását követően ne feledkezzünk meg a *perifériák* (nyomtató, *monitor* stb.) kikapcsolásáról sem.

kilobyte: Egy kilobyte = 1000 *byte*. Jelölése: kB (1 kB = 1000 B).

kiviteli periféria: A számítógép központi egységéhez kívülről csatlakozó eszköz (*monitor*, nyomtató, projektor, hangfal stb.), mely az adatok kivitelét teszi lehetővé.

kód: A kód megállapodás szerinti jelek vagy szimbólumok rendszere, mellyel valamely *információ* egyértelműen megadható. Ilyen például az írás vagy a jelbeszéd.

kódolás: Valamely információ átalakítása egyezményes jelekké. Írásnál gondolatainkat rögzítjük az ábécé betűi és írásjelek segítségével. Programozásnál a fordító vagy parancsértelmező programok számára érthető utasítások segítségével írjuk le, hogy milyen műveletek elvégzését várjuk el a számítógéptől.

közösségi média: Olyan internetes médiaeszköz, amelynek segítségével bárki tartalomszerkesztővé válhat. Feltölthet és megoszthat szöveges tartalmakat, képeket, hanganyagokat és videókat.

kulcsszavas kereső: Olyan *szoftver* vagy online szolgáltatás, amely segítségével az általunk megadott szavak (keresőszavak) vagy kifejezések (keresőkifejezések) előfordulására kereshetünk.

lánclevél: Rendszerint valamilyen érdekesnek vagy hasznosnak tűnő információt tartalmaz. Arra kéri olvasóját, hogy ossza meg ismerőseivel. Hatékony módja az *e-mail-címek* begyűjtésének.

laptop: Olyan hordozható számítógép, amely egyben tartalmazza a számítógép működéséhez szükséges elemeket (*processzor*, *memória*, *háttértároló* stb.), valamint a kijelzőt és *billentyűzetet*.

LCD: Folyadékkristályos megjelenítő. Használják *monitorokban*, okostévékben, mobiltelefonokban és egyéb megjelenítőeszközökben. Működési elve: a képernyő egy-egy pontján a folyadékkristály és az elé helyezett polárszűrő vagy átengedi a háttérvilágítást, vagy nem. Egy *képpont* három különböző színű (vörös, zöld, kék) elemi pontból áll. A három alapszínből keveri ki a képpont valódi színét.

lényegkiemelés: Dokumentumokban és bemutatókban a fontos információkat kiemelhetjük eltérő betűtípus-beállítások használatával. Ügyelni kell azonban arra, hogy a dokumentumnak csupán pár elemét emeljük ki, különben a tartalom átláthatatlan, a megjelenés pedig csúnya lesz.

LibreOffice: *Nyílt forráskódú*, ingyenes irodai programcsomag.

LibreOffice Writer: *Nyílt forráskódú*, szabadon használható szövegszerkesztő. Több operációs rendszeren is működik. A *LibreOffice* irodai *szoftvercsomag* része.

link: Hivatkozás egy dokumentum vagy egy *weboldal* valamely pontjára. A linkek gyorsan (kattin-

tással) elérhetővé teszik a dokumentumnak vagy weboldalnak az aktuális tartalomhoz kapcsolódó részét.

Linux: Nyílt forráskódú operációs rendszer. Nagyon sok Linux-változat (disztribúció) létezik. Többségük szabad szoftver.

megabyte: Egy megabyte = 1000 kilobyte. Jelölése: MB (1 MB = 1000 kB).

memória: A számítógép működése közben tárolja a számítógép működéséhez szükséges programokat és adatokat.

menüsor: Az aktív alkalmazásban, szoftverben használható parancsok csoportjai.

menüszalag: A felhasználói felület egy része, amelyet a Microsoft az Office 2007 irodai szoftvercsomaggal vezetett be. A programablak fölött elhelyezkedő terület, amely fülekből, csoportokból és parancsokból áll. A menüsört helyettesíti.

merevlemez: A merevlemezek már régóta megbízható háttértárolók. Angol elnevezésének (hard disk drive) elterjedt rövidítése: HDD. Működési elve: az adatokat mágnesezhető réteggel bevont, forgó lemez tárolja. Az egyik legfontosabb jellemzője a tárolókapacitása, ami több terabyte is lehet.

mikroprocesszor: Egy token (egy áramkörön) belül megvalósított központi feldolgozó egység (CPU). Vezérli a számítógép működését. Aritmetikai és logikai műveleteket végez.

monitor: A személyi számítógépek elsődleges kiviteli eszköze. Biztosítja a számítógéppel való állandó, azonnali kommunikáció és ellenőrzés lehetőségét a felhasználó számára. Az egyik legfontosabb jellemzője a felbontása. Minél nagyobb a felbontása, annál több képpontból áll össze, és annál szebb a megjelenített kép. Ma a monitorok nagy része LCD-technológiával készül.

multimédiás objektum: Az irodai szoftverek által készített dokumentumokba, táblázatokba és bemutatókba az írott szöveg mellett más objektumokat (képeket, alakzatokat, táblázatokat, videókat és hangokat) is beilleszthetünk.

netikett: Az internet használatának illemtana.

nyílt forráskódú program: Számítógépes program, amelynek a forráskódját a készítője közzéteszi, és szabadon terjeszthető, módosítható. Így mások megismerhetik az adott program működését, illetve saját igényeiknek megfelelően módosíthatják.

objektumok: Az irodai szoftverek által készített dokumentumokba, táblázatokba és bemutatókba az írott szöveg mellett más objektumokat (képeket, alakzatokat, táblázatokat, hiperhivatkozásokat) is beilleszthetünk. A programozás témakörében az objektum az információt hordozó és azokkal műveleteket vagy számításokat elvégezni képes egységet jelenti.

okostelefon: Az okostelefonok nemcsak telefonálásra használhatók, hanem komplett számítógépet is tartalmaznak, amelyeken az operációs rendszerükhöz készült alkalmazások futnak. Az internet böngészésére és online kapcsolattartásra is alkalmasak.

online identitás: Az internetfelhasználók által létrehozott online személyiség, amelyet az online térben használnak. Nagymértékben eltérhet a felhasználók valódi személyiségétől.

online piactér: Célja összehozni az online kereskedelem szereplőit: az eladókat, a szállítókat, a szolgáltatókat és a vásárlókat, fogyasztókat.

operációs rendszer: Programok gyűjteménye, amelyek elősegítik a számítógép hardverének könnyű, sokoldalú és biztonságos használatát. Elérhetővé teszik az egyéb szoftvereket, és kezelik a számítógép hardverelemeit.

PDF (Portable Document Format): Hordozható dokumentumformátum. A PDF-formátumban tárolt dokumentumokat a különböző eszközökön futó különböző szoftverek közel azonos módon jelenítik meg.

periféria: A számítógép központi egységéhez kívülről csatlakozó eszköz (monitor, billentyűzet stb.), mely az adatok be- vagy kivitelét teszi lehetővé.

processzor (Central Processing Unit, CPU): A számítógép központi feldolgozó egysége. Vezérli a számítógép működését. Aritmetikai és logikai műveleteket végez.

rendszergazda: Informatikai eszközök és hálózatok működtetésével, az adminisztrációs feladatok ellátásával foglalkozó szakember. Joga van programok telepítésére vagy törlésére, és módosíthatja valamennyi rendszerbeállítást.

robothálózat (botnet): Kártékony, rejtőzködő kóddal fertőzött informatikai eszközökből álló hálózat. A megfertőzött zombigépeken a kártékony kód sokszor hosszú ideig nem csinál semmit. Egy-egy után a robothálózaton érkező utasítás hatására feléled, és munkához lát.

sorköz: A sorok közötti távolság.

SSD: A szilárdtest-meghajtók (Solid-State Drive) lényegesen gyorsabbak a merevlemezeknél. Fogyasztásuk és méretük is kisebb. Egyre inkább átveszik a *merevlemez* (HDD) szerepét. Az SSD-gyártók olyan félvezető memóriákat használnak, amelyek kikapcsolás után is megőrzik a tárolt információt. Mivel az SSD-k nem tartalmaznak mozgó alkatrészeket, kevésbé sérülékenyek, mint a HDD-k. Az egyik legfontosabb jellemzője a *tárolókapacitása*, ami több *terabyte* is lehet.

streamelés (streaming): Adatátviteli technika, amely lehetőséget kínál videók és hanganyagok folyamatos továbbítására és lejátszására.

sütik (cookie-k): A böngészőben a *webhelyek* által tárolt szöveges információk.

szabad szoftver: Szabadon használható számítógépes program, amelynek a forráskódja is ismert. Szabad másolni, terjeszteni, tanulmányozni és módosítani.

Számológép (segédprogram): Az *operációs rendszerek* egyik segédprogramja. Alapértelmezetten egyszerű matematikai műveletek végrehajtására alkalmas. Menüjében azonban több üzemmód közül is lehet választani, így már programozhatóvá válik, és alkalmassá tehető a tudományos számítások elvégzésére is.

személyi számítógép: Angol elnevezésének (Personal Computer) rövidítése alapján gyakran csak PC-nek hívjuk. Nagyon elterjedt számítógéptípus. A működéséhez feltétlen szükséges részegységek (*processzor*, *memóriák*, *háttértárolók* stb.) egy számítógépházban vannak elhelyezve. A számítógép és a külvilág kapcsolatát a hozzá csatlakoztatott külső *be- és kiviteli perifériák* (*billentyűzet*, *egér*, *monitor* stb.) biztosítják.

szerver: A számítógépes hálózatok kiszolgáló-számítógépe. Szerverek szolgálják ki a *weboldalakat* böngészőnk számára, és szervereken tároljuk azokat a dokumentumokat, amelyeket másokkal meg szeretnénk osztani.

szimulátor: Egy helyzetet, tevékenységet vagy egy eszköz működését utánzó *szoftver* és/vagy *hardver*.

szoftver: Egy feladat megvalósítására készített program és a használatához szükséges kiegészítők (leírások, súgók, képtárak, sablonok stb.).

szövegtükör: Az oldalon az a – margókkal körülvett – terület, ahová a szöveg kerül.

táblagép: Kis méretű, könnyen kézbe vehető számítógép, amely nem tartalmaz valódi billentyűzetet. Az érintőképernyőjén látható ikonokra koppintva lehet a különböző funkcióit elérni. Szövegek beírását virtuális (látszólagos) *billentyűzet* teszi lehetővé. A képernyőn megjelenített karaktereket megérintve szövegeket lehet begépelni.

tárolókapacitás: Megadja, hogy egy *memória* vagy *háttértároló* milyen mennyiségű adatot képes tárolni. Mértékegysége memóriáknál rendszerint *gigabyte* (GB). A háttértárolóknál gyakran találkozunk *terabyte* (1000 GB) vagy annál is nagyobb tárolókapacitással.

tartomány (domain): Az internet egy meghatározott része, sok számítógép együttese.

tematikus kereső: Az adminisztrátorok által összeállított *linkgyűjtemény*. A készítői által hasznosnak gondolt *weboldalakat* teszi elérhetővé témakörök és altémakörök alapján csoportosítva.

terabyte: Egy terabyte = 1000 *gigabyte*. Jelölése: TB. (1 TB = 1000 GB).

térköz: A *bekezdések* közötti távolság.

torrent protokoll: Különböző fájlok (például videók) gyors letöltését teszi lehetővé úgy, hogy az egyes számítógépek (kliensek) között zajlik adat-továbbítás. Folyamatosan használja az eszköz erőforrásait, így lassíthatja a számítógép működését.

tömörítés: A tömörített fájlok kevesebb helyet foglalnak, és gyorsabban továbbíthatók. A leggyakrabban használt fájlkezelő programok a helyi menüben lehetővé teszik a kijelölt fájl vagy teljes könyvtár tömörítését a megfelelő parancs kiválasztásával. Tömörített állományt hasonlóképpen lehet kibontani. Dokumentumokat, táblázatokat veszteségmentes tömörítéssel, hang-, kép- és videófájlokat veszteséges tömörítéssel tömöríthetünk.

trójai programok: Nem megbízható forrásból származó *szoftverek*, amelyek nem csak az elvárt feladatokat látják el. Céljuk lehet az adatgyűjtés vagy a károkozás.

troll: A trollok sértő, agresszív, rosszindulatú módon, a tárgyhoz nem tartozó üzenetekkel bombáznak egy-egy online közösséget. Céljuk a zavar-keltés.

tűzfal (firewall): Megakadályozza, hogy a számítógépet illetéktelen behatolás (támadás) érje a számítógépes *hálózaton* keresztül.

Ügyfélkapu: A magyar kormányzat elektronikus ügyfél-azonosító rendszere. Lehetővé teszi az elektronikus közigazgatási ügyintézését.

VGA (Video Graphics Array): Grafikuskártya-szabvány, 320 × 200 vagy 640 × 480 pixeles felbontást és 16 vagy 256 szín megjelenítését teszi lehetővé.

videóchat: Több felhasználó között zajló online videóbeszélgetés. Webes szolgáltatásokon zajlik.

videokártya: A számítógép által küldött képi információkat a monitor számára értelmezhető jelekké alakítja.

vírus: Olyan kártevő program, amely rejtőzködve saját másolatait helyezi el más, végrehajtható programokban vagy dokumentumokban.

VR (virtual reality): A virtuális (látszólagos) valóság egy informatikai eszközök által létrehozott környezet, amelybe a felhasználó beleélheti magát, úgy érezheti, mintha a valóságban lenne.

webhely: Egymáshoz tartozó *weboldalak* csoportja, amelyeket linkek kötnek össze. Üzemeltetőjük lehet cég, szervezet vagy magánszemély.

weboldal: A weboldal (weblap) a világhálón (*WWW*) publikált dokumentum, amelyet egyedi internetes cím (URL) segítségével érünk el. A weboldalak szöveget, hivatkozásokat, képe-

ket, videókat és más erőforrásokat is tartalmazhatnak.

webszerver: Állandó *IP-címmel*, illetve *domain-névvel* rendelkező kiszolgáló, amely állandóan csatlakoztatva van az internethez. Ezekben publikálhatjuk a honlapokat (*weboldalak*at).

wifi: Vezeték nélküli kommunikációt (WLAN) megvalósító, széleskörűen elterjedt szabvány.

Windows: *Zárt forráskódú*, kereskedelmi, széles körben használt *operációs rendszer*. Megvásárlása esetén használható személyi számítógépeken és laptopokon. Van mobilkészülöket működtető változata is, de ma már elvétve találkozhatunk vele.

WWW (World Wide Web): Az interneten elérhető világméretű hálózat. Röviden: web vagy világháló. Olyan információs rendszer, amelyre jellemző, hogy *böngésző* programok vagy más alkalmazások segítségével különböző dokumentumokat (weboldalakat, *PDF*-dokumentumokat stb.) és egyéb állományokat (például képeket, videókat, zenéket) érhetünk el.

zárt forráskódú program: Számítógépes *szoftver*, amely a felhasználónak csak a program futtatását teszi lehetővé, de a forráskód zárt, azaz nem módosítható.

zsarolóvírus: Kártevő program, amely váltságdíj reményében blokkolja az eszközt, vagy titkosítja a rajta tárolt adatokat.