MINISTERUL EDUCAȚIEI ȘI CERCETĂRII

LITERA

Luminița Ciocaru
Ștefania Penea
Oana Rusu
Claudia-Elena Mitrache

Informatika és IKT 8

Tankönyv a VIII. osztály számára

Acest manual școlar este proprietatea Ministerului Educației și Cercetării.
Acest manual școlar este realizat în conformitate cu Programa școlară
aprobată prin OM nr. 3393 din 28.02.2017.

116.111 – numărul de telefon de asistență pentru copii

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII

LITERA

Luminița Ciocaru

Ștefania Penea

Oana Rusu

Claudia-Elena Mitrache

Informatika és IKT 8

Tankönyv a VIII. osztály számára

Manualul școlar a fost aprobat de Ministerul Educației și Cercetării prin ordinul de ministru nr. 5615/23.09.2020.

Manualul este distribuit elevilor în mod gratuit, atât în format tipărit, cât și digital, și este transmisibil timp de patru ani școlari, începând cu anul școlar 2020–2021.

Inspectoratul școlar .....

Școala/Colegiul/Liceul .....

ACEST MANUAL A FOST FOLOSIT:

Anul	Numele elevului	Clasa	Anul școlar	Aspectul manualului*			
				format tipărit		format digital	
				la primire	la predare	la primire	la predare
1							
2							
3							
4							

** Pentru precizarea aspectului manualului se va folosi unul dintre următorii termeni: nou, bun, îngrijit, neîngrijit, deteriorat.*

- Cadrele didactice vor verifica dacă informațiile înscrise în tabelul de mai sus sunt corecte.
- Elevii nu vor face niciun fel de însemnări pe manual.

Informatică și TIC. Manual pentru clasa a VIII-a
Luminița Ciocaru, Ștefania Penea, Oana Rusu, Claudia-Elena Mitrache

Referenți științifici: Conf. univ. dr. Radu-Eugen Boriga, Facultatea de Matematică și Informatică – Universitatea din București
Prof. Marilena Frîncu, Liceul Greco-Catolic „Timotei Cipariu", București

Traducere în limba maghiară: Hibácskó Gizella-Elisabeta, Jakab Irma-Tünde, Kondert Enikő, Veress Éva
Corector: Toth Domokos

Copyright © 2020 Grup Media Litera
Toate drepturile rezervate

Editura Litera
O.P. 53; C.P. 212, sector 4, București, România
tel.: 021 319 63 90; 031 425 16 19; 0752 548 372
e-mail: comenzi@litera.ro

Ne puteți vizita pe
www.litera.ro

Editor: Vidrașcu și fiii
Redactor: Gabriela Niță
Corector: Carmen Bîtlan
Credite foto: Dreamstime, Shutterstock
Copertă: Vlad Panfilov
Tehnoredactare și prepress: Pontlab Srl

Descrierea CIP a Bibliotecii Naționale a României
Informatică și TIC: manual pentru clasa a VIII-a /
Luminița Ciocaru, Ștefania Penea, Oana Rusu,
Claudia-Elena Mitrache. - București: Litera, 2020
ISBN 978-606-33-5476-2
I. Ciocaru, Luminița
II. Penea, Ștefania
III. Rusu, Oana
IV. Mitrache, Claudia-Elena
004

TARTALOMJEGYZÉK

A tankönyv szerkezete 4
Kedves diákok! 6
1. Lecke A VII. osztályban tanultak ismétlése és felmérése. 7

1. Egység TÁBLÁZATKEZELÉS 11
2. Lecke Táblázatkezelő alkalmazás. 12
3. Lecke Formázási műveletek és adattípusok 17
4. Lecke Képletek és függvények 21
5. Lecke Adatsorok és diagramok. 25
ISMÉTLÉS – Csapatmunka! 29
FELMÉRÉS – Légy felelősségteljes, védd! 30

2. Egység WEBLAPOK 31
6. Lecke A weblapszerkesztő 32
7. Lecke A weblap elemeinek szerkesztése. 37
8. Lecke A weblap elemeinek a formázása 41
ISMÉTLÉS – Csapatmunka! 45
FELMÉRÉS – Légy megfontolt, takarékoskodj!. 46

3. Egység ALGORITMUSOK 47
9. Lecke A számsor 48
10. Lecke Képzett számsorok 53
11. Lecke Beolvasott számsorok. 57
12. Lecke Interdiszciplináris algoritmusok 61
ISMÉTLÉS – Csapatmunka! 65
FELMÉRÉS – Légy kíváncsi, kutass! 66

4. Egység ROBOTIKA 67
13. Lecke Az akadályok felfedezése és elkerülése 68
14. Lecke A kijelölt útvonal követése 73
15. Lecke A kijelölt útvonal bejárása 77
ISMÉTLÉS – Csapatmunka! 81
FELMÉRÉS – Légy okos, fedezd fel!. 82

16. Lecke Ismétlés és felmérés!. 83
Válaszok és magyarázatok 87

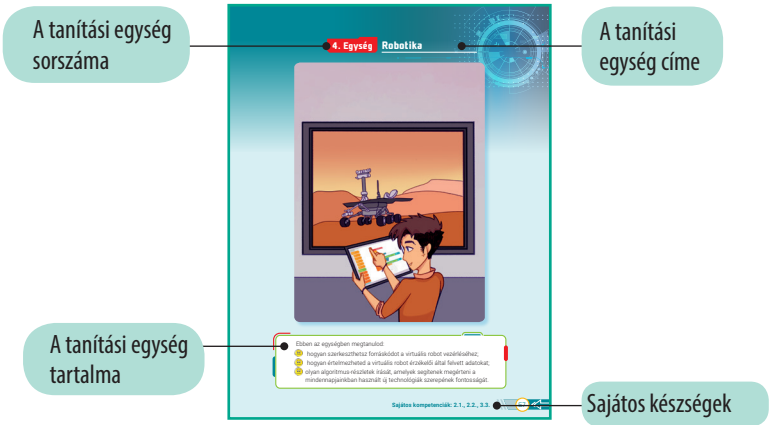
Függelék – HTML szöveg-formázások. 94
Függelék – HTML színkódok 95
Az Informatikában és IKT-ben használt kifejezések szótára 96

A tankönyv szerkezete

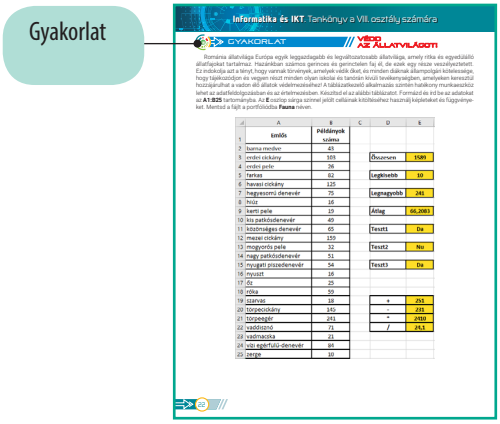
Nyomtatott változat

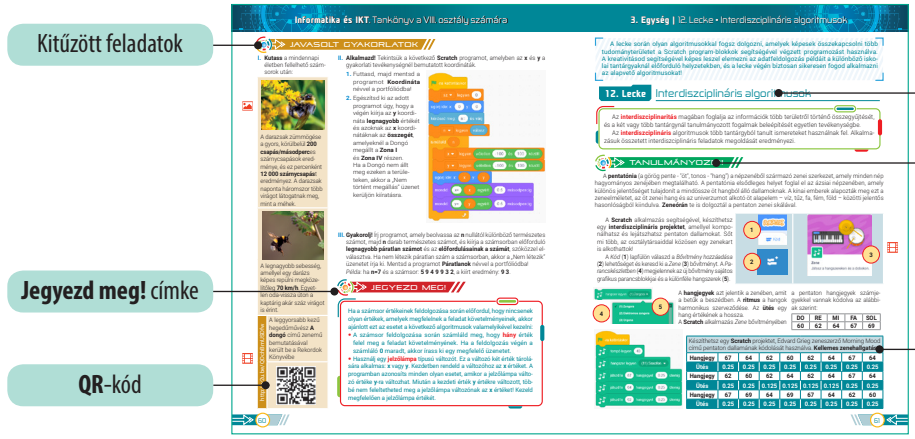
Az *Informatika és IKT. Tankönyv a VIII. osztály számára* négy egységből áll, amelyek tiszteletben tartják a tanterv szerkezetét és tartalmát. A tanórákat gyakorlati, alkalmazható jellegű interaktív tanulási-értékelési tevékenységek követik, ezáltal biztosítva a hozzájuk kapcsolódó kompetenciák kialakulását.

Tanítási egységet bemutató oldal

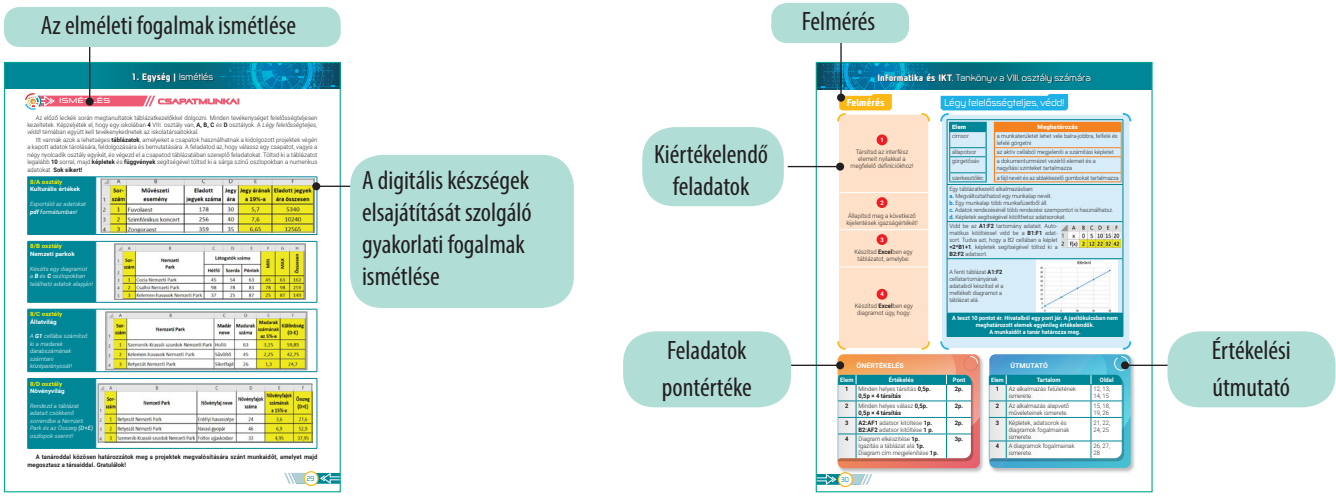


Leckét tartalmazó oldalak





Kiegészítő értékelési módszerek



Digitális változat

A tankönyv digitális változata tartalmazza a nyomtatott változat teljes tartalmát, emellett interaktív gyakorlatokat, oktató játékokat, animációkat, filmeket és szimulációkat is tartalmaz.

Mindezek célja a kognitív érték növelése.

A digitális változat megnyitható asztali számítógépen, laptopon, táblagépen, telefonon, ennek köszönhetően kiváló böngészési élményt nyújt.

A digitális változatban történő navigálás lehetővé teszi, hogy a tankönyv áttanulmányozása után bármikor visszatérhessünk az előző tanulási tevékenységhez.

A tankönyv elektronikus változata megegyezik a nyomtatott változat tartalmával, ugyanakkor kiegészíti azt egy sor interaktív, multimédiás - statikus, animációs, interaktív - tanulási tevékenységgel.

Statikus IMTT	Rajzokat, fényképeket, statikus diagramokat, statikus térképeket tartalmaz.
Animációt tartalmazó IMTT	Animációkat és filmeket tartalmaz.
Interaktív IMTT	Magas szintű interaktivitással járó oktatási elemeket tartalmaz (folyamatszimulációk, feladatmegoldások, kísérletezés és felfedezés, oktató jellegű játékok), amelyek révén a tanuló egy magasabb kognitív szintre juthat.

Kedves diákok!

Jelen tankönyv célkitűzése, hogy végigvezessen benneteket azon az izgalmas úton, amelyet végigjárva okos, megfontolt és felelősségteljes polgárokká váltok. Lehetséges, hogy még ismeretlen számotokra az a mondás, hogy az ember hasonló a fához. Természetesen ennek kapcsán most egyszerre gondolhattok mindarra, ami körülvesz bennünket, valamint a jótékony természetre, amit mindannyiunknak szent kötelessége megóvni abban a formában, ahogy kaptuk. Azt már tudjátok, hogy nagyon sokan aktívan részt vesznek népek és nemzetek anyagi és kulturális értékeinek a megőrzésében, a polgárok egyediség és szépségigényének az alakításában, valamint a mindennapjainkból hiányozhatatlan újdonságok népszerűsítésében. Ugyanakkor sok esetben szintén az emberek voltak azok, akik felrúgták a természet törvényeit, és így nem kívánt következményeket generáltak: a világunk harmóniájának, egyensúlyának a felborulását. Mi a teendő? Számtalan válaszlehetőség létezik erre a kérdésre, de a tudományok és művészetek egyöntetűen azt javasolják, hogy térjünk vissza a reményt adó természetes ártatlansághoz. Ennek értelmében mindannyiunknak van tanulnivalója - mindeki saját ritmusa szerint - mindabból, ami körülvesz minket: a legjelentéktelenebbtől a legösszetettebbig, és miért ne, a leghatékonyabb robotig. Hihetetlennek tűnhet, de ez a valóság. Ahogy beszélünk fák, madarak, hegyek nyelvéről, úgy beszélünk akár vízmolekula nyelvezetről is, de létezik egy sokkal szigorúbb és titokzatosabb nyelv is, ami megnemesíti az embert.

Az Informatika és az IKT tantárgy nem pusztán egy iskolai tantárgy az órarendben, hanem készségeket, képességeket és konstruktív attitűdöt fejlesztő eszköz is mindaddig, amíg lenyűgöző ötleteket használ és alkalmaz azokon a tevékenységi területeken, amelyeken az embernek van mondanivalója!

A Vár lakójaként sok sikert kívánuk ehhez a felfedező úthoz! És ne feledd: mindenből tanulhatsz!

A tudás örömével teljes sikeres tanévet kívánunk!

Általános és specifikus kompetenciák az OMEN 3393/28.02.2017-es rendelettel jóváhagyott Számítástechnika és IKT - VIII. osztályos tantervének megfelelően.

1. **Az információs és kommunikációs technológia felelős és hatékony felhasználása.**
   - 1.1. A táblázatok használata az egyszerű feladatok megoldásához.
   - 1.2. A megfelelő webszerkesztő használata különböző témájú weblapok készítéséhez.
2. **Alapfeladatok megoldása az információfeldolgozás intuíciós módszereivel.**
   - 2.1. Az algoritmusok létrehozásához szükséges számsorok felismerése.
   - 2.2. Egyszerű feladatok megoldása számsorokat feldolgozó algoritmusok segítségével.
3. **Társadalmi, kulturális és személyes kérdésekkel foglalkozó kreatív miniprojektek létrehozása, figyelve az információ hitelességére és a szerzői jogokra.**
   - 3.1. Táblázatkezelő segítségével előállított informatikai termékek előállítása.
   - 3.2. Weblapok szerkesztése/frissítése megadott követelmények szerint.
   - 3.3. Programok gyakorlatba ültetése adott programozási környezetben.

Az elmúlt iskolai évben megtanultad kezelni a dokumentumok és szövegek szerkesztésére használt szövegszerkesztőt. Ugyanakkor használtad a multimédiás szerkesztőt is ahhoz, hogy kreatív és interaktív környezetben dolgozhasd fel és állíthasd össze a különböző médiák eszközeit. Az algoritmusok gyakorlatba ültetéséhez programozási felületeket használtál, és programokat írtál különböző nyelvezetekben. Ügyes vagy, csak így tovább!

1. Lecke A VII. osztályban tanultak ismétlése és felmérése

ISMÉTLÉS

I. Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést!

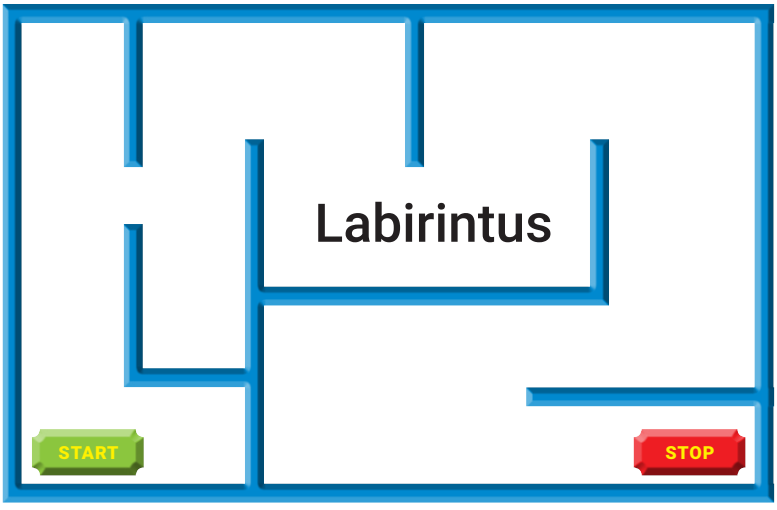
1. A **LibreOffice** csomag ingyenes és a www.libreoffice.org weboldalról lehet letölteni..
2. A **Microsoft Word** szövegszerkesztőben a program nem viszi át automatikusan a következő oldalra, ha a beírt szövegünk túlhaladja a jelenlegi oldal alsó margóját.
3. Az együttműködő platformok közelebb viszik az iskolai környezetet a diákokhoz; a diákok kedvére való modern eszközök megkönnyítik a tanulást.
4. A számítógépre telepíthető multimédiás szerkesztő programok kiegészülnek a nagyon fontos és könnyen hozzáférhető telefonos alkalmazásokkal.

II. Egészítsd ki az alábbi kijelentéseket, felhasználva a zárójelben levő szavakat (közösségi, csoport, elemek, együttműködő)!

1. A **Microsoft Word** szövegszerkesztőben készített dokumentumban az **.....** lehetnek: szövegek, képek, táblázatok, vonalak, mértani formák, grafikonok, diagramok, stb.
2. A **Microsoft Word** menüszalagja több lapot is tartalmaz, amelyek a maguk során több **.....** tartalmaznak.
3. A **.....** alkalmazások elősegítik a csoportmunkát, mert a csoporton belüli felhasználók együttműködhetnek, megjegyzéseket, ímeléket, üzeneteket küldhetnek egymásnak.
4. A dokumentumok és a multimédiás termékek megosztása manapság nagyon könnyen történik a **.....** platformok segítségével.

III. Alkalmazd! Használd a Word szövegszerkesztőt, hozz létre és ments el Labirintus névvel a saját mappádba egy dokumentumot, amelyben:

1. A lapod beállítása **Fekvő tájolású** legyen, és tartalmazzon egy **vízszintes** szöveget **Labirintus** néven.
2. Az **Insert/Shapes** menüből vonalakból készítsd el a mellékelt labirintust: a vonalak színe **kék**, mérete pedig **20** legyen.
3. Jobb és bal oldalra beszúrsz egy-egy alakzatot, és beleírod, hogy **START**, valamint **STOP**. **Csoportosítsd** az alakzatokat!



Innen le tudsz tölteni egy labirintusmintát:
http://bit.ly/383OITn

IV. Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést!

1. Az **Open Roberta** programozási felületen lehet programot írni, ez tesztelhető egy virtuális roboton, majd a program feltölthető egy valódi robotra is.
2. A **MakeCode** programozási alkalmazást nem szükséges telepíteni, ennek ellenére meg lehet nyitni számítógépről és bizonyos mobil eszközről is.
3. **Kompilálás**sal egy **C++/Python** program lefordítható a processzor által érthető nyelvre.
4. A **C++/Python** programban nem írhatunk megjegyzéseket.
5. A **Code::Blocks** és **Python** programozási felületek a felhasználónak nem nyújtanak segítséget a programban felmerülő hibák javítására.
6. A **C++/Python** programban az **x, y, z** egész típusú változók beolvasása a **cin>>x>>y>>z; / x,y,z=int(input()),int(input()),int(input())** paranccsal történik.
7. A **C++/Python** programban két változó, **x** és **y** különböző sorokba való kiíratását **cout<<x<<endl<<y; / print(x,'\n', y)** paranccsal érjük el.

V. Egészítsd ki az alábbi kijelentéseket a zárójelben levő szavakkal (F9/F5, Ctrl+S, Alt+F4, Ctrl+N):

1. A **C++/Python** program mentése ..... billentyű/billentyűkombináció lenyomásával történik.
2. A **Code::Blocks/Python** programozási felületen egy új állomány létrehozása ..... billentyű/billentyűkombináció lenyomásával történik.
3. Egy **C++/Python** program futtatása ..... billentyű/billentyűkombináció lenyomásával történik.
4. A **Code::Bloks/Python** programozási felület bezárása ..... billentyű/billentyűkombináció lenyomásával történik .

VI. Végezd el!

Kijelentés	C++ nyelv
Adott a mellékelt **C++/Python** programozási nyelvezetekben megírt algoritmus.	

Python nyelv

```
n=int(input())
x=0
p=1
while n!=0:
    szamjegy=n%10
    if szamjegy%2==0:
        x=x+szamjegy*p
        p=p*10
    n=n//10
print(x)
```

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,x,p,szamjegy;
  x=0;p=1;
  cin>>n;
  while(n!=0)
  { szamjegy=n%10;
    if(szamjegy%2==0)
    { x=x+szamjegy*p;
      p=p*10;
    }
    n=n/10;
  }
  cout<<x;
  return 0;
}
```

1. Határozd meg, milyen érték íródik ki az algoritmus futtatása után, ha a beolvasott érték **4152039**!
2. Határozd meg azt a legnagyobb 4 jegyű számot, amelyre az algoritmus a **68** értéket írja ki!
3. Fogalmazz meg egy feladatot az adott algoritmusra!

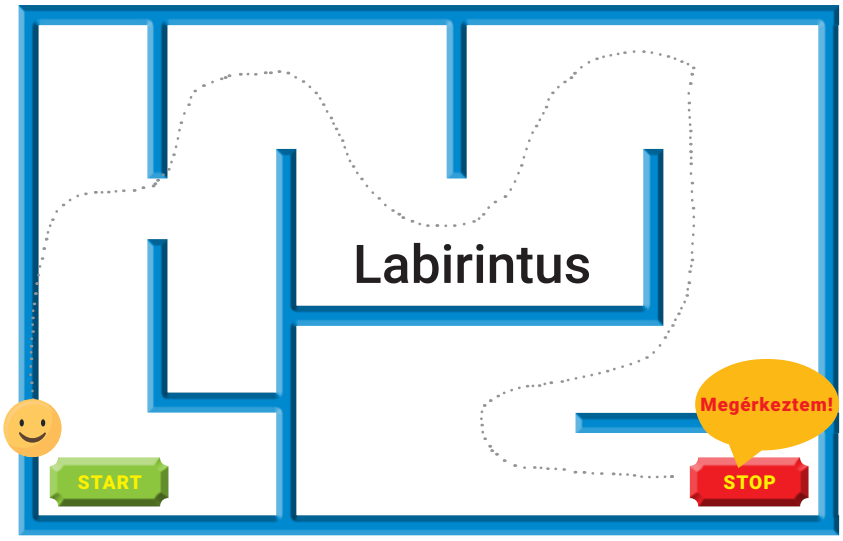
FELMÉRÉS

I. Társítsd a kifejezést a helyes meghatározással!

Kifejezések	Meghatározás
1. Szöveg-szerkesztő program	**a.** egy olyan digitális tárhely, amely elősegíti a közös munkát egy biztonságos oktatási környezetben. Meg lehet valósítani online módon is interneten keresztül, valamint helyileg is egy hálózaton keresztül.
2. Programozási nyelvezet	**b.** olyan utasítás-sorozat, amely meghatározza, hogy a számítógép milyen sorrendben, milyen műveleteket végezzen a bemeneti adatokkal.
3. Program	**c.** egy alkalmazás keretén belül barátságos, grafikus környezetben segít programokat írni.
4. Multimédiás szerkesztő program	**d.** szövegek szerkesztésére használt alkalmazás. Általában üzenetek, programok írására, weboldalak készítésére, valamint az operációs rendszer kezelésére használják.
5. Programozási felület	**e.** az algoritmusoknak a számítógép által érthető és elvégezhető utasításssorozattá való átalakítását valósítja meg.
6. Együttmű-ködési felület	**f.** egy olyan alkalmazás, melynek segítségével olyan média-eszközöket tudunk feldolgozni és egybefogni, mint: szöveg, kép, animációk, 3D rajzok, hang, grafika, és még sok más. Az eszközök összefogásának célja egy érdekes és interaktív végső audió-videó termék bemutatása.

II. A PowerPoint alkalmazással készíts egy prezentációt, mentsd Labirintus néven a saját mappádba, amibe majd:

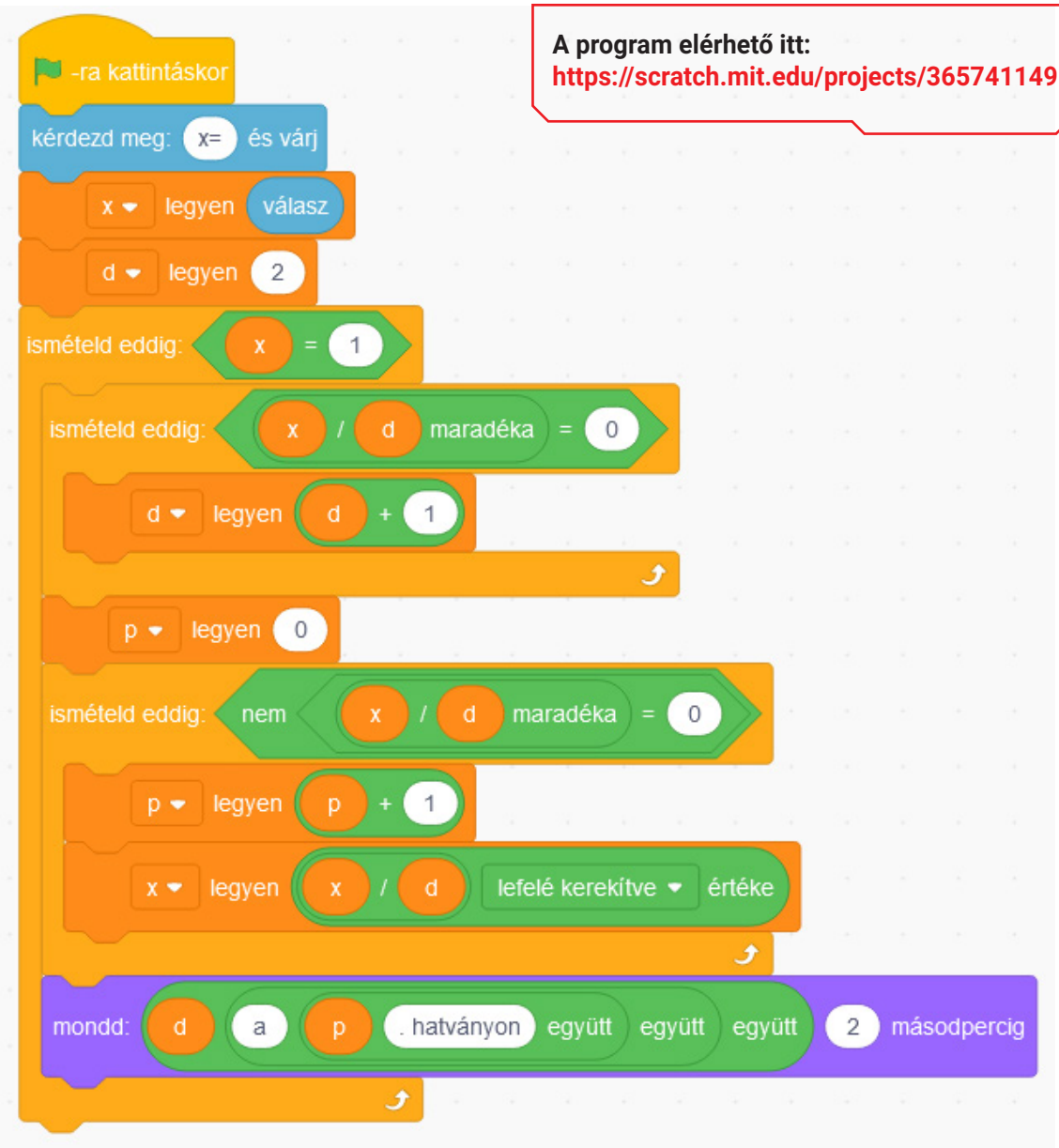
1. Szúrj be egy **Blank** (üres) típusú diát.
2. Erre a diára szúrj be egy másolatot a **Labirintus** nevű dokumentum megfelelő oldaláról.
   A **START** formára tegyél be egy **Smiley face**-t. A **Motion Path** animációs effektust használva végezd el a megfelelő lépéseket ahhoz, hogy a **Smiley face** eljusson a **STOP** formához anélkül, hogy elérné a kék vonalakat. A **sebesség 5 másodperc** legyen (**nagyon lassú**).
3. Végezd el a megfelelő beállításokat ahhoz, hogy amikor a **Smiley face** eléri a **STOP** alakzatot, írja ki azt az üzenetet, hogy „Megérkeztem!”
4. Exportáld a prezentációt a mappádba mint multimédiás termék ***.mp4** kiterjesztéssel.



Itt megnézhetsz egy mintát:
http://bit.ly/31ufVfk

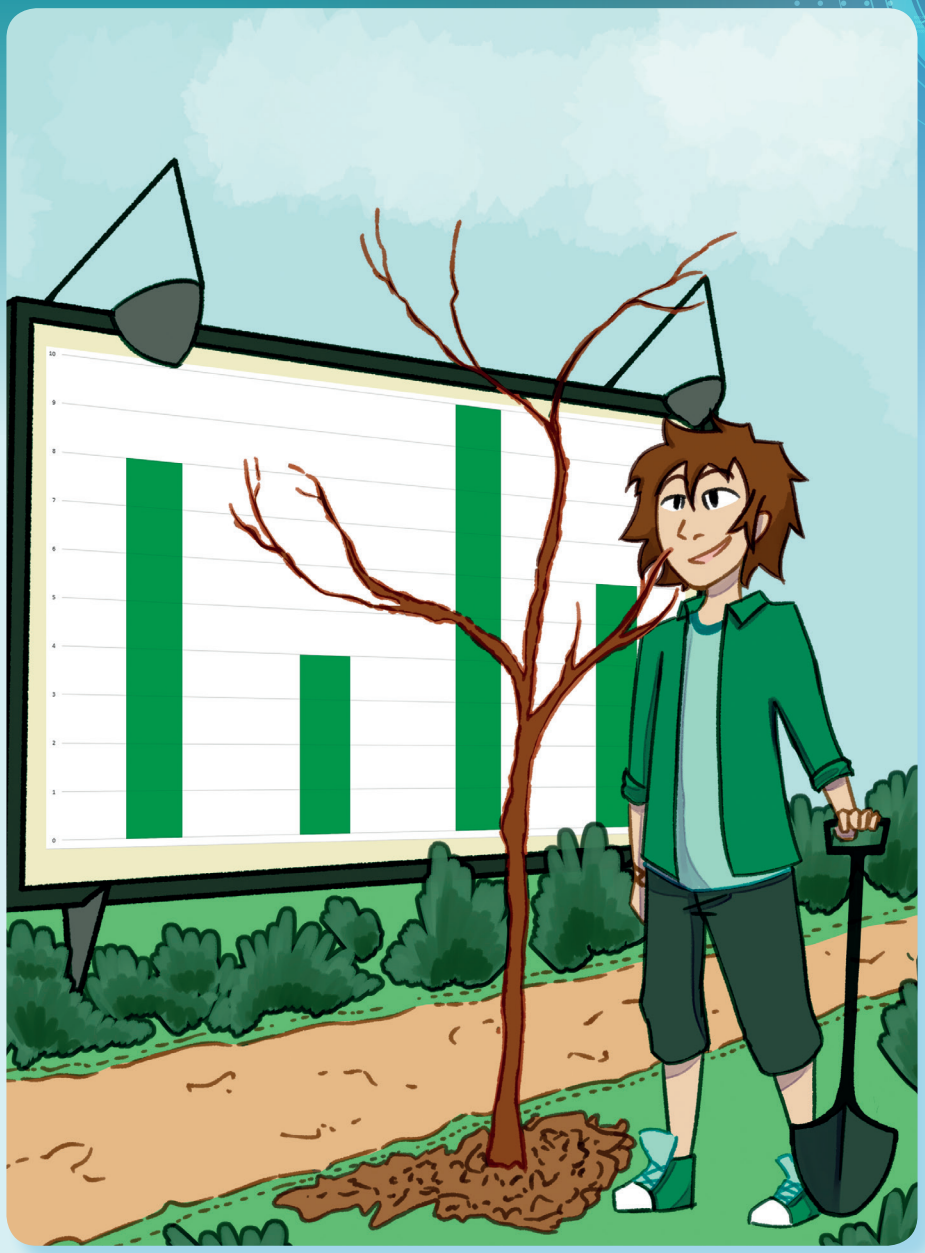
III. Végezd el!

Adott az alábbi Scratch – ben írt program (**x** természetes szám, **x>1**):



1. **Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést!**
   - **a.** A programban használt változók száma **4**.
   - **b.** A programban az ismétlő ciklusok száma **3**.
   - **c.** A program elején a **d** változó kezdő értéke **2**.
   - **d.** A **d** változó kiírt értéke mindig egy prímszám.
2. Határozd meg, hogy milyen érték íródik ki akkor, ha a beolvasott szám **300**.
3. Határozz meg egy értéket az **x** változónak, amelyre egyetlenegyszer íródik ki a **d**, és amelynek értéke **5**.
4. Fogalmazz meg egy feladatot az adott algoritmusra!

1. Egység Táblázatkezelés



Ebben az egységben a következőket fogod elsajátítani:

- adatok gyűjtése és formázása táblázatkezelő alkalmazások segítségével;
- adatok feldolgozása és értelmezése a táblázatkezelőkre jellemző műveletekkel;
- interdiszciplináris projektek kidolgozása.

Sajátos kompetenciák: 1.1., 2.1., 3.1.

A lecke során megismerkedsz a táblázatkezelő alkalmazások kezelőfelületének specifikus elemeivel. Megismerkedsz a munkafüzet szerkezetével, és az ilyen típusú állományokban használható műveletekkel. Munkalapokat fogsz készíteni valós adatok felhasználásával, és így meg fogod érteni a táblázatkezelő alkalmazások hasznosságát.

2. Lecke Táblázatkezelő alkalmazás

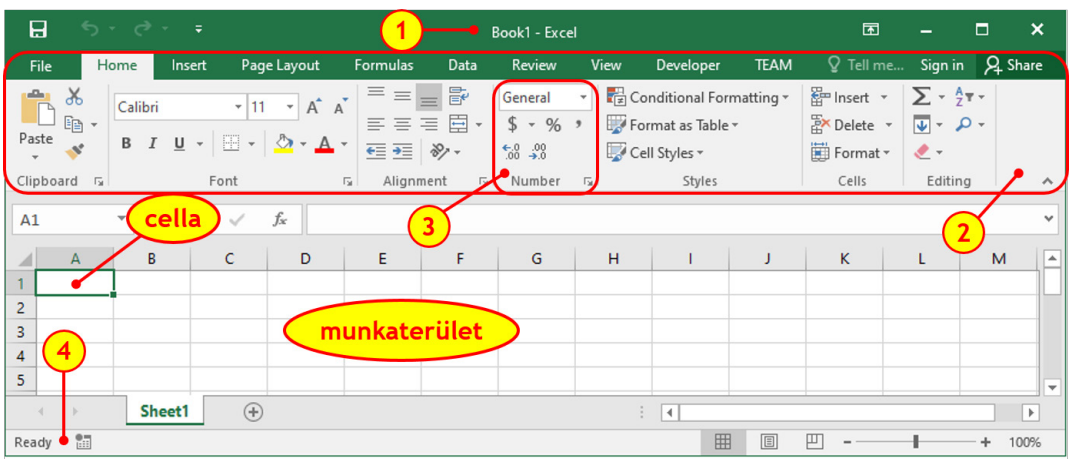
A **táblázatkezelő alkalmazás** olyan program, amelyet olyan helyzetek létrehozására és kezelésére alkalmaznak, amelyekben táblázatokat használnak számítások elvégzésére, diagramok készítésére és a táblázat adatainak egyéni feldolgozására.
A **munkafüzet** egy táblázatkezelő alkalmazással létrehozott állomány.
A **munkalap** a táblákba rendezett adatok tárolásának és feldolgozásának alapdokumentuma.

Az előző tanévek során különböző alkalmazásokkal dolgoztál online vagy offline módban. Gyakorlati alkalmazásokat és projekteket készítettél a következő típusú fájlokkal: kép, dokumentum, bemutató, hang, videó és multimédia. Idén a táblázatkezelő fájlok is megjelennek a portfóliódban. Természetes, hogy egy táblázatkezelővel sokkal könnyebben össze tudod állítani a szükséges kellékek listáját, meg tudod határozni a lista egyes tételeinek teljes árát, valamint a teljes készlet megvásárlásához szükséges költségvetést.

A táblázatkezelő alkalmazás kezelőfelülete ismerősnek fog tűnni majd a számodra, és nagyon hamar meg fogsz barátkozni vele. A többi szerkesztőhöz hasonlóan a táblázatkezelő alkalmazásokhoz is hozzátartozik egy menüszalag, amelyek különféle gombokat tartalmaznak az elvégezhető tevékenységek szerint csoportosítva.

A Microsoft Excel táblázatkezelő alkalmazás kezelőfelülete

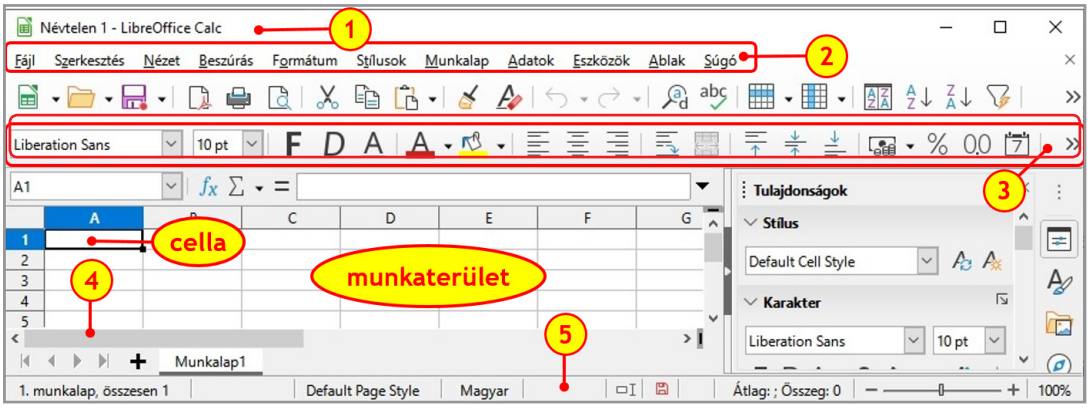
Az alkalmazás része a **Microsoft Office** csomagnak, és telepíteni kell a számítógépre. **Windows** operációs rendszeren működik, de van online verziója is.



1. **Címsáv:** tartalmazza a megnyitott állomány nevét és az ablakkezelési menüt *(Minimalizálás, Maximalizálás/Előzőméret, Bezárás)*.
2. **Menüszalag:** több **Fülből** áll, mindegyik egy aktivitási zónát jelképez. A fülek rendre több **Gombcsoportot** tartalmaznak.
3. **Gombcsoport:** egy fül aktivitási zónáját alzónákra osztja fel. Mindegyik csoport olyan gombokat tartalmaz, amelyek egy tevékenységet hajtanak végre, vagy egy parancsmenüt jelenítenek meg. Néhány csoport nevének jobb oldalán egy speciális *Launcher* gomb található, ez a csoport párbeszédablakát jeleníti meg.
4. **Állapotsor:** az aktuális oldal számát tartalmazza, a dokumentum összes oldalainak számát, a dokumentum lehetséges nézeteit és a munkalap nagyítására szolgáló csúszkát.

A LibreOffice Calc táblázatkezelő alkalmazás kezelőfelülete

Az alkalmazás része az ingyenes **LibreOffice** csomagnak, amelyet telepíteni kell a számítógépre, és kompatibilis a következő operációs rendszerekkel: **Windows, Linux** és **MacOS**.



1. **Címsáv:** tartalmazza a megnyitott fájl nevét és az ablakkezelési menüt *(Csökkentés, Növelés/ Visszaállítás, Bezárás)*.
2. **Menü:** több opciót tartalmaz, mindegyik egy függőleges almenüt képvisel, amelyek hozzáférést biztosítanak a **Calc** alkalmazás parancsaihoz.
3. **Eszközsor:** több gombot tartalmaz, ezek lehetnek egyszerű parancsokat végrehajtó gombok vagy parancsmenü gombok. Egy eszközsor be- és kikapcsolható a *Nézet → Eszköztárak* menüben.
4. **Görgetősávok:** segítségükkel a munkafelületet balra-jobbra, fel-le lehet görgetni.
5. **Állapotsor:** tartalmazza az aktuális oldal számát, a dokumentum összes oldalainak számát, a dokumentum lehetséges nézeteit és a munkalap nagyítására szolgáló csúszkát.

Munkafüzet – szerkezete és specifikus műveletei

Egy munkafüzet egy vagy több **munkalap**ból áll. Kezdetben az állomány egy munkalapot tartalmaz. Egy munkalap sorokból és oszlopokból áll. **Aktív munkalap**nak nevezzük azt a lapot, amelyikben éppen dolgozunk.

A **sorok** vízszintesen helyezkednek el, és a munkafelület bal oldalán folyamatos számozással vannak megcímkézve. A számozás így kezdődik: **1, 2, 3,...** Nyomd le egyszerre a **Ctrl** és ↓ gombokat, hogy láthatóvá tedd az **Excel** utolsó sorát. Nyomd le egyszerre a **Ctrl** és **Home** gombokat, hogy újra az **A1** cellába kerülj.

Az **oszlopok** függőlegesen helyezkednek el, és a munkaterület felső részén betűkkel vannak megcímkézve. A címkézés az **A** betűvel kezdődik, az ábécé végéig tart, majd kétbetűs csoportok következnek, szintén ábécé sorrendben: **A, B, ..., Z, AA, AB, AC, ...** Nyomd le egyszerre a **Ctrl** és → gombokat, hogy lásd az utolsó oszlop címkéjét.

A **cella** az **adatok tárolásának alapvető egysége**, és egy sor és oszlop metszéspontjában található.

A cellát az oszlop és sor metszéspontjakét azonosítjuk, az oszlop és a sor címkéjét használjuk fel, mint például: **A1, F24, XFD1048576.** A cellákat egymástól *rácsvonalak* választják el.

Új munkafüzet készítése: kiválasztjuk a *File → New → Blank workbook* opciót, vagy a **CTRL** és **N** billentyűkombinációt.

Munkafüzet megnyitása: kiválasztjuk a *File* menüből az *Open* opciót, vagy használjuk a **CTRL** és **O** billentyűkombinációt.

Munkafüzet mentése: a *File* menüből kiválasztjuk a *Save* vagy a *Save As* opciót, vagy használjuk a **CTRL** és **S** billentyűkombinációt.

Munkafüzet bezárása: a *File* menüből kiválasztjuk a *Close* opciót, vagy használjuk az **ALT** és **F4** billentyűkombinációt.

GYAKORLAT /// VÉDD A KULTURÁLIS ÉRTÉKEKET!

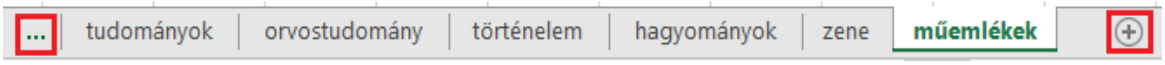
A kultúra az anyagi és szellemi értékek létrehozásának tevékeny folyamata. Az értékeket az emberi igények határozzák meg, ezek válaszok a vágyakra és szükségletekre, az emberek munkájának és kreativitásának a kifejezői, ugyanakkor társas jellegűek is. Egy nemzet fennmaradása érdekében ezeket az értékeket nemzedékről nemzedékre kell átadni, és meg kell újítani.

1. lépés

Nyisd meg az **Excel** alkalmazást, kattints a jobb egérgombbal a munkalap eredeti nevére, válaszd ki a *Rename* opciót, és írd be a **szobrászat** nevet, majd nyomd le az Entert.

2. lépés

Szúrd be a következő munkalapokat: **festészet, tudományok, orvostudomány, történelem, hagyományok, zene, műemlékek.** Használd a *lapfülek görgetősávjáról* a **+** opciót. Ezen a sávon szerepelnek a munkalapok nevei. Nem minden esetben jelenik meg az összes munkalap neve. Több munkalap megjelenítéséhez használd a *három pontot* tartalmazó gombot. A két előbb említett opció az alábbi képen ki van emelve. Mentsd le a fájlt a portfóliódba **Kulturalis_ertekek_1** névvel.



3. lépés

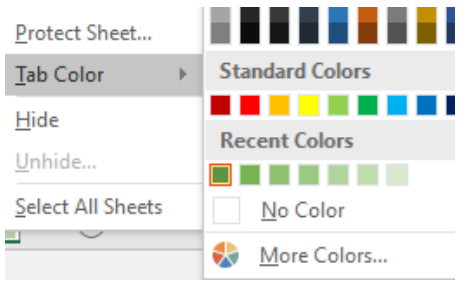
Mentsd le a **Kulturalis_ertekek_1** állományt **Kulturalis_ertekek_2** névvel.
Ebben az állományban módosítsd a munkalapok sorrendjét úgy, hogy azok nevei legyenek ábécé sorrendben. Ehhez a bal gombbal kattints a munkalapra, majd vidd a megfelelő helyre!

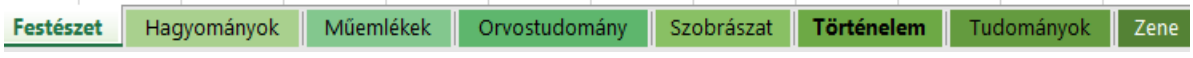
4. lépés

Nevezd át a munkalapokat úgy, hogy nagybetűvel kezdődjön a nevük. Ehhez kattints a jobb egérgombbal a munkalap lapfülére, és a helyi menüből válaszd ki a *Rename* opciót.

5. lépés

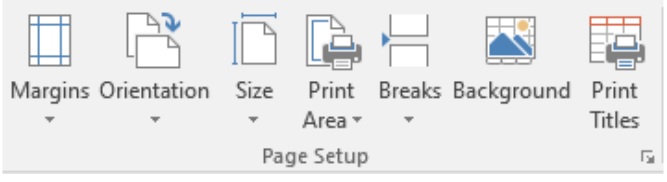
A zöld különböző árnyalataival színezd át az összes munkalapfület. Ehhez kattints a jobb egérgombbal a lapfülre, és válaszd ki a *Tab Color* opciót. Változatos árnyalatok használatához kattints a *More Colors…* opcióra!

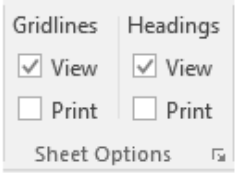




6. lépés

Formázd egyénire a munkalapjaid! Ehhez használd a *Page Layout* menüszalag opcióit. A *Page Setup* csoportból válaszd ki a *Background* gombot minden munkalapra. Az *Insert Pictures* ablakban szúrj be egy, a munkalap címkéjének megfelelő képet. A *Sheet Options* csoportból kikapcsolhatod a rácsvonalak megjelenítését (*Gridlines > View*).

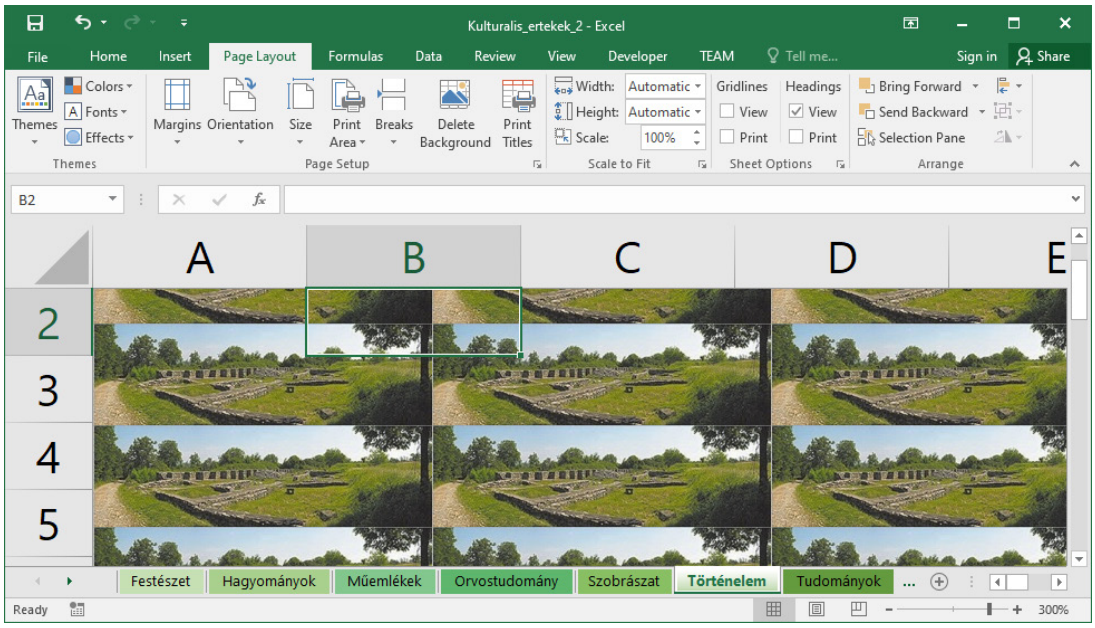


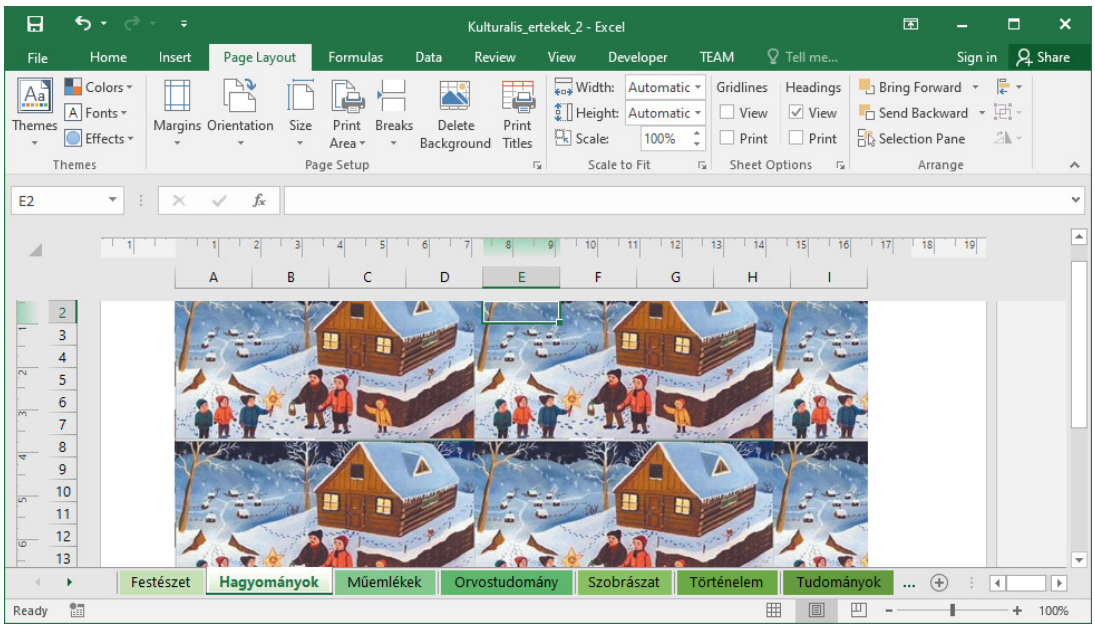


7. lépés

Folytasd a munkalapok személyre szabását! Használd az *állapotsor* opcióit. Az első négy munkalap esetében alkalmazd a *Page Layout* nézetet. Figyeld meg, hogy a munkalap alapértelmezett nézete *Normal*. Az első munkalapra állíts be **50%** nagyítást *(Zoom level)*, majd növeld a szintet minden lapon **50**-el, míg az utolsónál a nagyítás **400%** lesz.

Íme a gyakorlati tevékenység eredménye!

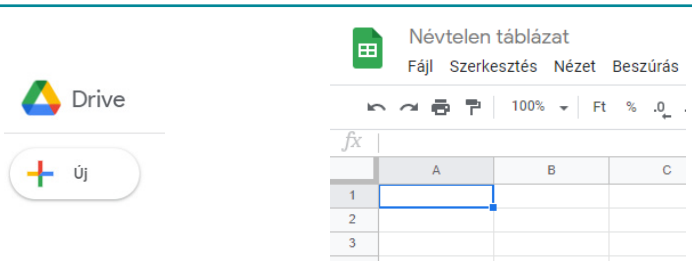




FEDEZD FEL!!

Google Táblázatok alkalmazás

Lépj be a **Google** fiókodba és nyisd meg a **Drive** alkalmazást. Nyomd le az *Új* gombot és válaszd ki a *Google Táblázatok*at. Fedezd fel a munkalapok beszúrás, átnevezés és színezés műveleteit!



JAVASOLT GYAKORLATOK

I. Fedezd fel! Romániában 6 olyan helyszín található, amelyek az **UNESCO Világörökség** részét képezik.

Moldvai templomok

Horezu kolostor

Szászvárosi templomvár

Segesvár történelmi központja

Máramarosi fatemplomok

Erdélyi templomok

II. Gyakorolj!

1. Határozd meg, melyik válasz helyes:

Egy munkalap lapfülén a jobb egérgomb lenyomásakor kapott helyi menüben nem szerepel a következő opció:

a) beszúrás	b) törlés	c) kivágás	d) átnevezés

2. Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést!

a. Egy munkafüzet egy vagy több munkalapból áll.

b. A **Microsoft Excel** alkalmazás menüszalagja több *Fülből* áll, mindegyik egy tevékenységi zónát képvisel.

c. A táblázatkezelő alkalmazásokban a sorok függőlegesen vannak elrendezve, és számokkal vannak címkézve.

d. A **LibreOffice Calc** alkalmazás menüje több opciót tartalmaz, mindegyik egy függőleges almenüt képvisel, amelyek hozzáférést biztosítanak az alkalmazás parancsaihoz.

III. Alkalmazd!

1. A portfóliódban készíts egy **Táblázatkezelés** nevű mappát. Ebbe a mappába hozz létre 3 munkafüzetet a következő nevekkel: **Történelem, Hagyományok** és **Orvostudomány**.

2. Mindegyik munkafüzetbe szúrj be **3** munkalapot. A munkalapokat nevezd el a táblázatnak megfelelően:

Történelem	Hagyományok	Orvostudomány
Ștefan cel Mare	Márciuska	Nicolae Paulescu
Mihai Viteazul	Húsvét	Victor Babeș
Regina Maria	Esküvői szertartás	Ana Aslan

3. Másold le a három munkafüzetet és nevezd át őket így: **Szobrászat, Festészet, Tudományok.** Nevezd át a munkafüzetekben szereplő munkalapokat a táblázatnak megfelelően. A munkalapok lapfül címkéinek állíts be tetszőleges háttérszínt.

Szobrászat	Festészet	Tudományok
Constantin Brâncuși	Nicolae Grigorescu	Grigore Antipa
Ion Irimescu	Ștefan Luchian	Henri Coandă
Dimitrie Paciurea	Constantin Rosenthal	Petrache Poenaru

TARTSD ÉSZBEN!

- **A Microsoft Excel alkalmazással egy munkafüzet alapértelmezett módban .xlsx kiterjesztéssel lesz lementve. A LibreOffice Calc alkalmazásban egy munkafüzet .ods kiterjesztéssel kerül mentésre.**
- **Microsoft Excel**ben **menthetünk .ods kiterjesztésű fájlt. A LibreOffice Calc alkalmazásban menthetünk .xlsx kiterjesztésű fájlt.**
- **A munkafüzeteknek különböző kiterjesztései lehetnek. Válaszd ki a Save As… opciót akkor, ha más kiterjesztésű állományt szeretnél, mint például: .xls, .pdf, .csv vagy egyebet.**

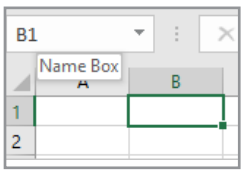
A lecke során megismered, hogyan kell szerkeszteni és formázni a cellát, amely a táblázatkezelőkben az adattárolás alapegysége. A sorok és oszlopok formázását adatgyűjtési táblák létrehozásával fogod gyakorolni. Megtanulod az adatok feldolgozását és táblázatokba rendezését is. Így alkalmazni fogod a táblázatkezelők lehetőségeit!

3. Lecke Formázási műveletek és adattípusok

Az **aktív cella** a munkalapon kijelölt cella. Mivel a cella egy sor és oszlop metszéspontjában található, ezért **az aktív cella címét** az **oszlop betűjele** és a **sor száma** adja meg. Az aktív cellába különböző típusú adatokat írhatunk be, ezekre sokféle formázási műveletet alkalmazhatunk: sortörés, igazítás, színezés, egyesítés és egyéb műveletek.

Az aktív cella formázási műveletei

Ha ki akarsz **jelölni** egy cellát, akkor *kattints* rá. A koordinátája a munkalap tetején a bal oldalon jelenik meg a *Name Box* (Név) mezőben. A mellékelt képen a kijelölt cella a **B1**. Az adatbevitelhez *kattints duplán* a cellára. A begépelt információk az aktív cellában jelennek meg. Az adatok beírása után a véglegesítés az **Enter** vagy **Tab** gombokkal történik.



Ha ki szeretnéd törölni egy cella teljes tartalmát, akkor *kattints rá,* és nyomd le a **Delete** gombot.
Ha cella tartalmának csak egy részét akarod megváltoztatni, akkor *kattints rá duplán* vagy nyomd le az **F2** gombot.
A munkalapon az elmozdulás megoldható az egérrel; de használhatod a vízszintes és függőleges görgetősávokat is.

A munkalapon a billenyűzet segítségével is el lehet mozdulni a következő gombok használatával:
Tab = egy cellával jobbra lép
Home = a sor legelső cellájára lép
Nyilak = elmozdulás egy cellával balra/jobbra/fel/le

Ha az egérkurzort az aktív cella fölé viszed, akkor három formát vehet fel, mégpedig:

(1) a cella belsejében		(2) a cella körvonalán		(3) a cella jobb alsó sarkában	

(1) Cella kijelölése. Kattints a cellára a jobb egérgombbal, és megjelenik a helyi menü.
(2) Cella elmozdítása. A bal gomb lenyomva tartásával a cella áthelyezhető.
(3) Cella másolása. A bal gomb lenyomva tartásával a cella lemásolható.

Több cellára vonatkozó formázási műveletek

Egy **sor** kijelöléséhez kattints a sor címkéjének gombjára. Az **első sor** összes cellájának kijelöléséhez a *Name box* mezőbe írd az **1:1** hivatkozást. Egy **oszlop kijelöléséhez** kattints az oszlop címkéjére. Az **A** oszlop összes cellájának kijelölésére írd a *Name Box* mezőbe az **A:A** hivatkozást.

Egy **téglalap alakú terület összefüggő** celláinak **kijelöléséhez** először jelöld ki a terület *bal belső sarkát* meghatározó cellát, majd az egér bal gombját lenyomva tartva húzd a *jobb alsó sarok* felé. Az **A1** és **B10** közötti cellák kijelöléséhez írd a *Name Box* mezőbe az **A1:B10** hivatkozást.
Több **nem összefüggő cella kijelöléséhez** a **CTRL** gombot lenyomva tartva jelöld ki a kívánt cellákat.
A **teljes munkalap kijelöléséhez** kattints a *Mindent kijelöl* gombra vagy nyomd le egyszerre a **Ctrl** és **A** gombokat.

GYAKORLAT /// VÉDD A NEMZETI PARKOKAT!

Megtanultad már, hogy a **Nemzeti Park** egy olyan nagy kiterjedésű terület, amelyet törvény nyílvánít védetté a természetes környezet, valamint a növény- és állatvilág sokszínűsége által képviselt biodiverzitás megőrzésének érdekében. A nemzeti parkok felbecsülhetetlen értékét a változatos tájrészek, az itt előforduló ritka növény- és állatfajok és az élőlények élettere (a biotóp) határozza meg. A Kárpátok hegyvonulata, a Duna folyam és a Fekete - tenger által határolt Románia a természeti értékek valódi kincsestára.

A gyakorlat során elkészíted az alábbi táblázatot. Nyisd meg a táblázatkezelő alkalmazást, kövesd lépésről lépésre a gyakorlati tevékenységet, és a végén mentsd el a fájlt a portfóliódba **Nemzeti_Parkok** néven.

1. lépés	Formázd az **A1:E18** tartomány (blokk) celláit a következő módon: módosítsd az **A, B, C, D** és **E** oszlopainak a szélességét. Egyesítsd a következő tartományokat: **A1:A2, B1:B2, C1:D1, E1:E2** és **B18:E18**.
2. lépés	Az **A1:E2** tartományra alkalmazd a *Sortöréssel több sorba (Wrap text)* formázást. Az **E1:E2** tartományban forgasd el a szöveget **45°** fokkal az óramutató járásával ellenkező irányba. Az **A1:E16** és **A18:E18** tartományokra alkalmazz vastag, zöld színű szegélyt, és a képen látható mintának megfelelően módosítsd a háttérszínt és a betűk színét.
3. lépés	A **B3:B16** tartományban formázd a számokat ezreselválasztó használatával és két tizedessel. A **B18:E18** tartományban írd ki az aktuális dátumot, a képen látható formátumnak megfelelően.
4. lépés	Rendezd növekvő sorrendbe az **A3:E16** tartomány adatait, a *Terület (ha)* oszlop szerint.

Sorok és oszlopok formázása **(1. lépés)**

Szöveg és szegély formázása **(2. lépés)**

Adattípus személyre szabása **(3. lépés)**

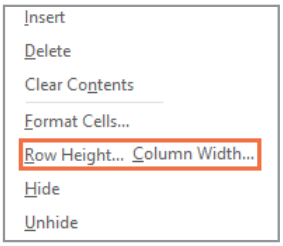
Adatok rendezése **(4. lépés)**

	A	B	C	D	E
1	Információk	Terület (ha)	Évszám		Website
2	Nemzeti Park neve		alapítás	bejegyzés	
3	Békás-szoros-Nagyhagymás Nemzeti Park	6.575,00	1990	2000	cheilebicazului-hasmas.ro
4	Buila-Vânturarița Nemzeti Park	4.186,00	2005	2005	buila.ro
5	Cozia Nemzeti Park	17.100,00	1966	2000	cozia.ro
6	Csalhó Nemzeti Park	8.396,00	1995	2000	ceahlaupark.ro
7	Domogled-Cserna Nemzeti Park	61.211,00	1982	2000	domogled-cerna.ro
8	Duna-delta Nemzeti Park	580.000,00	1991	2000	ddbra.ro
9	Kelemen-havasok Nemzeti Park	24.041,00	1975	2000	calimani.ro
10	Királykő Nemzeti Park	14.733,00	1938	2000	pcrai.ro
11	Măcin-hegység Nemzeti Park	11.321,00	2000	2000	parcmacin.ro
12	Néra-szurdok-Beusnica Nemzeti Park	36.758,00	1990	2000	cheilenereibeusnita.ro
13	Radnai-havasok Nemzeti Park	46.599,00	1990	2000	parcrodna.ro
14	Retyezát Nemzeti Park	38.047,00	1938	2000	retezat.ro
15	Szemenik-Krassói-szurdok Nemzeti Park	36.664,00	2000	2000	pnscc.ro
16	Zsil-völgy Nemzeti Park	11.127,00	2005	2005	defileuljiului.ro
17					
18	A táblázat készítésének dátuma:	10-Oct.			

1. lépés

Formázd a sorokat és oszlopokat!

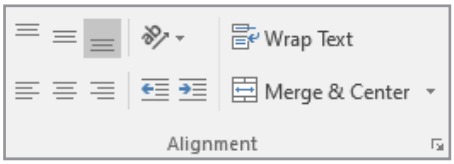
Egy oszlop szélességének vagy egy *sor magasságának* a **módosításához** az egérkurzort az oszlop- vagy sorcímkék közé kell helyezni, és „húzni" kell *balra-jobbra,* illetve *fel-le*. Egy oszlop további formázási opcióinak eléréséhez az oszlopcímkén nyomd le a jobb egérgombot, és a helyi menüben válaszd ki a kívánt opciót. Hasonló opciók léteznek a *sorokra* is. A *Home* fülön, az *Alignment* csoportban **elforgathatod a szöveget** vagy **egyesíthetsz** két vagy több cellát (jelöld ki a cellákat és kattints a *Merge&Center* gombra; általában a táblázat első sora lesz a táblázat fejléce).



2. lépés

Formázd a cella szövegét!

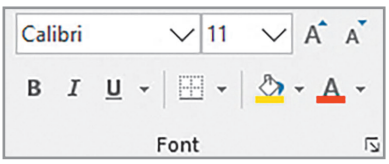
Egy cella tartalmának igazításához a Home fülön keresd meg az Alignment csoportot. Azt a szöveget, amelyik nem „fér be" a cellába, a Wrap text opcióval **sortöréssel több sorba igazítjuk.**



A cellában a **szöveg igazításához** a rendelkezésre álló hat gombot használhatjuk *függőleges igazításra*: felülre, függőlegesen középre, alulra és vízszintesen: balra, vízszintesen középre, jobbra.

A *Home* fülön *a Font* csoportban:

- Átalakíthatod a szöveget a betűcsalád, a stílus, a méret, a szín és az aláhúzás módosításával.
- Módosíthatod a háttérszínt.
- Egyéni szegélyeket alkalmazhatsz.



3. lépés

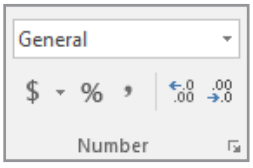
Adattípusok testreszabása!

Egy cellába adatokat írhatsz be. A legfontosabb **adattípusok**:

szám: számok megjelenítésére használjuk. Alapértelmezés szerint a cella tartalma jobbra igazított;

szöveg: a cella tartalmát úgy jeleníti meg, ahogy beírták. A számokat szövegként értelmezi. Alapértelmezés szerint a cella tartalma balra igazított;

dátum: a dátumok megjelenítésére szolgál.



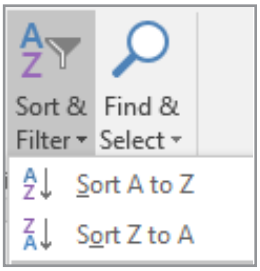
Egy adattípus feldolgozásához használhatod a *Home* fülről a *Number* csoport gombjait. További opciókért használd a *Launcher* gombot a párbeszédablak eléréséhez.

4. lépés

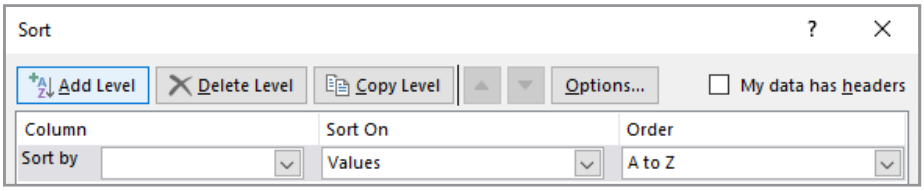
Rendezd a táblázat adatait!

Egy táblázat adatainak rendezése azt jelenti, hogy a sorait a megadott szempontok szerint más sorrendbe rendezzük, mégpedig:

- **növekvő** vagy **csökkenő** sorrendbe (numerikus adatokat);
- **ábécé** (az ábécé betűi) szerint, **A-tól Z-ig** vagy **Z-től A-ig** (nem numerikus adatokra). Rendezni egy legalább két oszlopot tartalmazó téglalap alakú területet fogunk.



A tartomány kijelölése után a *Home* fül *Editing* csoportjából válaszd ki a *Custom Sort...* gombot. A *Sort* ablakban be tudod állítani a rendezési szempontokat.

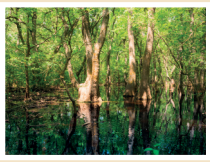


Rendezni több szempont szerint is lehet. Az *Add Level* gomb egy új rendezési szempontot hoz létre. Egy szemponthoz a következőket kell beállítani: *Sort by* (az oszlop, amelyik szerint rendezünk) és az *Order* (a rendezés típusa). Ha a táblázat fejlécét is kijelölted, akkor jelöld be a *My data has headers* opciót is.

JAVASOLT GYAKORLATOK

I. Fedezz fel Románia legszebb nemzeti parkjai közül néhányat:

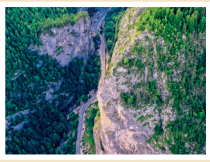
Letea erdő



Măcin-hegység

Néra-szurdok

Békás-szoros



Máramarosi havasok

Sárkányok kertje

II. Gyakorolj!

Határozd meg, melyik válasz helyes:

1. A *Border* gombon nincs a következő opció:
a. Bottom Border
b. Right Border
c. Outside Borders
d. Complex Borders

2. Egy táblában az adatok rendezése **nem** lehet:
a. Növekvő sorrendben
b. Csökkenő sorrendben
c. Hosszúság szerint
d. Ábécé szerint

3. Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést.

Egy munkalapon:

a. Megváltoztathatod a kijelölt terület szegélyének típusát.
b. A mozgás megvalósítható az egér vagy a billentyűzet használatával.
c. Nem lehet kijelölni több összefüggő cellát.
d. Az elrejtett oszlopokat újra láthatóvá lehet tenni.

III. Alkalmazd!

Nyisd meg a **Nemzeti_Parkok** állományt és végezd el az alábbi követelményeket. Minden követelményhez másold a táblázatot egy új munkalapra, és abban old meg a követelményeket. A feladatok elvégzése után a munkafüzet **hat** munkalapot fog tartalmazni.

1. Jelenítsd meg a **B** oszlop adatait *tizedesvessző* és *tizedesjegyek nélkül.*
2. Az **A3:E16** cellatartomány adatait rendezd csökkenő sorrendbe az *Alapítás évszáma* oszlop (első szempont) és a *Bejegyzés évszáma* szerint.
3. A **B18:E18** cellatartományban a dátumot úgy jelenítsd meg, hogy legyen látható az évszám is.
4. Egy tetszőleges színt felhasználva színezd be az **A3:E16** cellatartományból **10** nem összefüggő cella hátterét.
5. Alkalmazz az **A3:E16** cellatartomány adataira dupla szegélyt.
6. A *Format cells* ablakban a *Fill effects* opciót felhasználva alkalmazd az alapértelmezett mintát az **A17:E17** cellatartományra.

JEGYEZD MEG!

- **Az aktív cellába írt szöveg megjelenik a *szerkesztőlécen (Formula bar)* is.**
- **Két szomszédos címke közé duplán kattintva a sorok vagy oszlopok mérete automatikusan átméretezésre kerül.**
- **Ha „eltűntek" sorok vagy oszlopok, jelöld ki azokat a címkéket, amelyek közül eltűntek a sorok vagy az oszlopok, jobb kattintás a kijelölésen és *Unhide*.**
- **A tizedesjel lehet a pont vagy a vessző.**
- **Egy cella formátumának lemásolásához használhatod a *Home* fül *Clipboard* csoportjából a *Format Painter* gombot.**
- **Az aktív cellában az új sorba lépéshez használd az Alt és Enter billentyűkombinációt.**

Ebben a leckében megtanulsz következtetéseket levonni a táblázatok adatai alapján a táblázatkezelő egyedi függvényeinek segítségével. Megtanulsz képleteket alkalmazni néhány fontos számításhoz, amelyek aritmetikai operátorokat használnak. Hasznos lehet egy adott témára vonatkozó adatokat bevinni egy táblázatba, és a megtanult függvények segítségével kinyerni az eredményeket.

4. Lecke Képletek és függvények

Hivatkozás egy cellára egy azonosító, amellyel az adott cella a munkalapon egyértelműen azonosítható. Az **5**-ös sor és a **B** oszlop metszéspontjában található cella hivatkozása **B5**.

Hivatkozás egy tartományra egy olyan azonosító, amellyel azonosítani lehet a munkalapon megtalálható, téglalap alakú tartományt. **C3:E10** annak a tartománynak az azonosítója, amelyik a **C3** bal felső saroktól az **E10** jobb alsó sarokig tart.

Számítási képletek

Aritmetikai operátorok és más cellákra való hivatkozás segítségével számítási képleteket hozhatsz létre!
Az aktív cellába akkor írunk egy **képletet**, ha a cellában egy számítás eredményét szeretnénk kiíratni. **Minden képlet = jellel kezdődik.**
Egy képletben a cella hivatkozásának használata nagyon fontos. Ha egy cellában található érték megváltozik, akkor **automatikusan** módosulni fog az összes olyan cella adata is, amelyek hivatkozást tartalmaznak arra a cellára.
Egy képlet tartalmazhat: számokat, cellahivatkozásokat, aritmetikai operátorokat **(+, -, *, /)** stb.
Az aritmetikai operátorok prioritása a matematikából ismert sorrend.

A mellékelt képen az **A4:C4** tartomány celláit egyesítettük, és egy olyan képlet szerepel benne, amiben a cellahivatkozás színe megegyezik az adott cella színével! A képlet írása közben az megjelenik a *szerkesztőlécen* is.
A képlet szerkesztése leállítható az **Esc** gomb lenyomásával.
Ha befejezted a szerkesztést, akkor nyomd le az **Enter** gombot. Ha helytelenül írtad be a képletet, hibaüzenet jelenhet meg a cellában.
Egy képlet módosítása elvégezhető közvetlenül a *szerkesztőlécen* vagy dupla kattintással a képletet tartalmazó cellában.

Szerkesztőléc

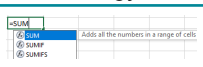
	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9
4	=A1+B2+C3+10		

A mellékelt képen a **D1** cellába először az **=A1+C1** képletet írtuk, majd a **D1** cellát lemásoltuk a **D2** és **D3** cellákba.
Ha módosul egy olyan cella pozíciója, amelyik képletet tartalmaz, akkor módosulnak a benne található cellahivatkozások is. Ha a képletet a **sorok mentén** másoljuk, akkor csak a *sor száma* változik, ha az **oszlopok mentén** másoljuk, akkor csak az *oszlop betűjele* változik meg.

	A	B	C	D
1	1	2	3	=A1+C1
2	4	5	6	=A2+C2
3	7	8	9	=A3+C3

Függvények

Képleteket írhatunk a következő függvényekkel: összeg, maximum, minimum, átlag és döntés!
Egy képletben használhatod a táblázatkezelő előre definiált saját **függvényeit**. A legismertebb függvények: **SUM, MAX, MIN, AVERAGE** és **IF**.
Amikor az aktív cellába egy függvényt írsz, a **táblázatkezelő** segít! Ha elkezded írni a függvény nevét, akkor megjelenik egy lista a hasonló nevekkel. Ha megtaláltad a keresett függvényt, akkor kattints rá duplán, és a lista helyén a függvény általános alakja jelenik meg, ez segít a függvény **argumentumainak** a kitöltésében. A Home fül Editing csoportból használhatod az AutoSum gombról elérhető függvényeket is. **Figyelem!** Az elválasztó jel lehet **;** vagy **,** ezt az operációs rendszer beállításai határozzák meg.



=SUM(A1;C1:D2;A3;B2)

GYAKORLAT /// VÉDD AZ ÁLLATVILÁGOT!

Románia állatvilága Európa egyik leggazdagabb és legváltozatosabb állatvilága, amely ritka és egyedülálló állatfajokat tartalmaz. Hazánkban számos gerinces és gerinctelen állatfaj él, de ezek egy része veszélyeztetett. Ez indokolja azt a tényt, hogy vannak törvények, amelyek védik őket, és minden diáknak állampolgári kötelessége, hogy tájékozódjon és vegyen részt minden olyan iskolai és tanórán kívüli tevékenységben, amelyeken keresztül hozzájárulhat a vadon élő állatok védelmezéséhez! A táblázatkezelő alkalmazás szintén hatékony munkaeszköz lehet az adatfeldolgozásban és az értelmezésben. Készítsd el az alábbi táblázatot. Formázd és írd be az adatokat az **A1:B25** tartományba. Az **E** oszlop sárga színnel jelölt celláinak kitöltéséhez használj képleteket és függvényeket. Mentsd a fájlt a portfóliódba **Fauna** néven.

	A	B	C	D	E
1	**Emlős**	**Példányok száma**			
2	barna medve	43			
3	erdei cickány	103		**Összesen**	**1589**
4	erdei pele	26			
5	farkas	82		**Legkisebb**	**10**
6	havasi cickány	125			
7	hegyesorrú denevér	75		**Legnagyobb**	**241**
8	hiúz	16			
9	kerti pele	19		**Átlag**	**66,2083**
10	kis patkósdenevér	49			
11	közönséges denevér	65		**Teszt1**	**Da**
12	mezei cickány	159			
13	mogyorós pele	32		**Teszt2**	**Nu**
14	nagy patkósdenevér	51			
15	nyugati piszedenevér	54		**Teszt3**	**Da**
16	nyuszt	16			
17	őz	25			
18	róka	59			
19	szarvas	18		+	**251**
20	törpecickány	145		-	**231**
21	törpeegér	241		*	**2410**
22	vaddisznó	71		/	**24,1**
23	vadmacska	21			
24	vízi egérfülű-denevér	84			
25	zerge	10			

1. lépés

Állítsd be a Text formátumot **D19:D22** tartomány celláira.
Írd be, sorrendben, az **E19, E20, E21** és **E22** cellákba az alábbi képleteket

`=E7+E5` `=E7-E5` `=E7*E5` `=E7/E5`

2. lépés

A **SUM** függvény kiszámítja az argumentumként megadott értékek **összegét**. A mellékelt képen az **E3** cellában szereplő képlet látható.

`=SUM(B2:B25)`

3. lépés

A **MIN** függvény kiszámítja az argumentumként megadott értékék **minimumát**. A mellékelt képen az **E5** cellába írt képlet látható.

`=MIN(B2:B25)`

4. lépés

A **MAX** függvény kiszámolja az argumentumként megadott értékek **maximumát**. A mellékelt képen az **E7** cellába írt képlet látható.

`=MAX(B2:B25)`

5. lépés

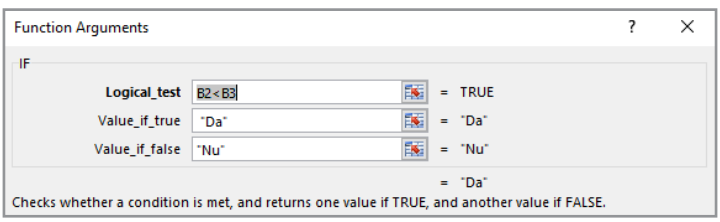
Az **AVERAGE** függvény kiszámolja az argumentumként megadott értékek **számtani középarányosát** (átlagát). A mellékelt képen az **E9** cellába írt képlet látható.

`=AVERAGE(B2:B25)`

6. lépés

Az **IF** függvény egy, a logikai teszt kiértékelése után meghatározott értéket térít vissza. A logikai teszt eredménye lehet TRUE (igaz) vagy FALSE (hamis). A függvény szerkeszthető:

- az aktív cellában: **=IF(logikai_teszt; Érték_ha_igaz; Érték_ha_hamis)**
- a *Function Arguments* ablakban, ami megnyitható a *Home* fül *Editing* gombcsoportjából az *AutoSum* gombról a *More Functions…* opcióval.



Logical_test: egy olyan érték vagy kifejezés, aminek eredménye TRUE vagy FALSE.
Value_if_true: a kifejezés TRUE végeredményére visszatérített érték.
Value_if_false: a kifejezés FALSE végeredményére visszatérített érték.

A mellékelt képen az **E11** cella képlete látható, ennek eredménye „Da”, ha **B2<B3** és „Nu” ellenkező esetben.

`=IF(B2<B3;"Da";"Nu")`

Az alábbi képen látható az **E13** cellából származó képlet, amelyik eredménye „Da”, ha **B17+B18>B19+B20** és „Nu” ellenkező esetben.

`=IF(B5+B6>B7+B8;"Da";"Nu")`

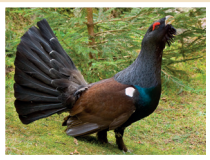
Az alábbi képen látható az **E15** cellából származó képlet, amelyik eredménye „Da”, ha **SUM(B10:B12)>SUM(B13:B15)** és „Nu” ellenkező esetben.

`=IF(SUM(B10:B12)>SUM(B13:B15);"Da";"Nu")`

JAVASOLT GYAKORLATOK

I. Fedezd fel Románia törvény által védett vadállatait!

Siketfajd



Hiúz

Európai bölény

Gímszarvas

Eurázsiai hód

Zerge

II. Gyakorolj!

1. Határozd meg, hogy melyik válasz helyes:

Az a függvény, amelyhez szükséges a *More Functions...* opció, ami elérhető a *Home* fül *Editing* csoportjából az *AutoSum* gombról:

a) SUM	b) IF	c) MAX	d) AVERAGE

2. Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést.

a. Minden képlet az **=** jellel kezdődik.
b. **W2** egy hivatkozás, amelyik a **W** sor és **2** oszlop metszéspontjában található cellát határozza meg.
c. Az **AVERAGE** függvény kiszámolja az argumentumként megadott értékek összegét.
d. A **MIN** függvény kiszámolja az argumentumként átadott értékek maximumát

3. Egészítsd ki a következő mondatokat a zárójelbe írt szavakkal (pozíciója, képletet, előre definiált, érték):

a. Az aktív cellába akkor írunk **.....**, ha a cellában egy számítás eredményét akarjuk kiíratni.
b. Egy képletben használhatjuk a táblázatkezelő **.....** függvényeit.
c. Ha megváltozik egy cellában az **.....** , akkor automatikusan módosulni fog az összes olyan cella értéke is, amelyek hivatkozást tartalmaznak erre a cellára.
d. Ha módosul egy képletet tartalmazó cella **.....**, akkor módosulnak a benne található cellahivatkozások is.

III. Alkalmazd!

Nyisd meg a **Fauna** állományt, egy új munkafüzetbe másold át a Védd az állatvilágot! gyakorlatnál készített munkalapot, és mentsd le a portfóliódba **Fauna2** néven. Töröld ki a **D** és **E** oszlopokat és végezd el a következő feladatokat:

a. A **B27** cellába írj egy olyan függvényt, amelyikkel kiszámolod a **B2:B9** cellatartomány **minimumát**.
b. A **B28** cellába írj egy olyan függvényt, amelyikkel kiszámolod a **B10:B17** cellatartomány **maximumát**.
c. A **B29** cellába írj egy olyan függvényt, amelyikkel kiszámolod a **B18:B25** cellatartomány **számtani középarányosát (átlagát)**.
d. A **B30** cellába írj egy olyan függvényt, amelyik a „Da" szót írja ki, ha a **B29** cellában levő érték kisebb a **B27** és **B28** cellák értékeinek számtani középarányosnál.

JEGYEZD MEG!

- Egy képletben a cellahivatkozásokat kisbetűvel is lehet írni.
- Képlethibák: **#DIV/0** (nullával való osztás), **#NAME!** (ismeretlen azonosító), **#NUM!** (érvénytelen számérték), **#N/A** (a cella nem található), **#REF!** (már nem létező cella), **#VALUE!** (nem megfelelő adat vagy azonosító).

Ebben a leckében megtanulhatod, hogyan lehet a táblázatkezelőben automatikusan és kreatívan adatsorokat létrehozni. A diagram grafikus objektum létrehozása lehetővé teszi egy táblázat adatainak pontos és érdekes bemutatását. Lehetőséged adódik az adott témának megfelelően különféle diagramok kiválasztására.

5. Lecke Adatsorok és diagramok

Az **adatsor** olyan értékekből áll, amelyek leírják mennyiség változását. A táblázatkezelő segít abban, hogy egy táblázat soraiban és oszlopaiban **egyéni adatsorokat** készítsünk.

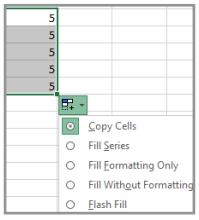
A **lineáris sorozat** egy olyan számokból álló adatsor, ahol az első szám adott és a másodiktól kezdve az összes többi szám kiszámolható az előző értékből, hozzáadva a **lépést.** Példa egy lineáris sorra: **5, 15, 25, 35, 45,…** ahol az első szám az **5** és a lépés **10**.

A **diagram** egy nagyon hatékony grafikus eszköz, amellyel a táblázatkezelők adatai egyértelműen bemutathatóak.

Adatsor készítése másolási művelettel

Annak elkerülésére, hogy az adatokat manuálisan írjuk be, **cellamásolási művelettel** automatikusan kitölthetjük az adatsort.

1. Az aktív cellába írd be a sorozat első értékét.
2. Vidd az egérkurzort az aktív cella jobb alsó sarkába, tartsd lenyomva a bal egérgombot és húzd felfelé vagy lefelé, *balra* vagy *jobbra*. Észre fogod venni, hogy az aktív cella értékét másolja le! Figyelem a **lépésre**!



Adatsor készítése a *Fill* gomb használatával

Egy adatsor **készítése**:

1. Írd be az aktív cellába az adatsor kezdőértékét és jelöld ki.
2. Válaszd ki a *Home* fül, *Editing* csoportjából a *Fill* gombról a *Series...* opciót.
3. Válaszd ki a kitöltési opciókat (sorokban vagy oszlopokban), a növelés értéket (a lépést) és a megállási értéket.

Ha *Dátum* típust jelöltél ki, akkor kiválaszthatod a *dátum* egységeit.

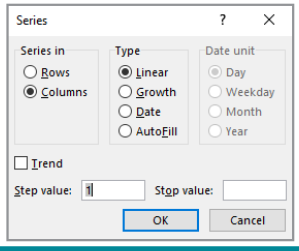


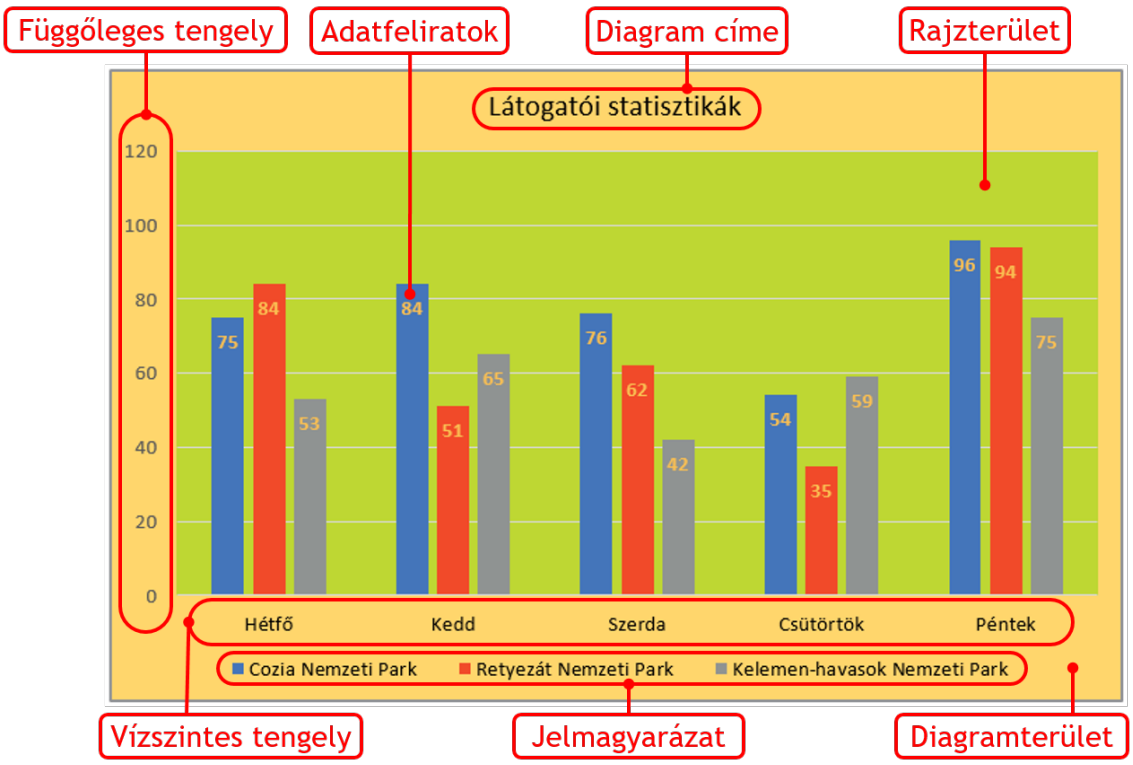
Diagram beszúrása

Egy diagram beszúrásához kijelölhetsz egy tartományt és az *Insert* fül *Charts* csoportjából kiválaszthatsz egy opciót. Ha látni szeretnéd, hogy néz ki a diagram mielőtt a munkalapra kerül, akkor használd az *Insert Chart* párbeszédpanelt *(Launcher)*.

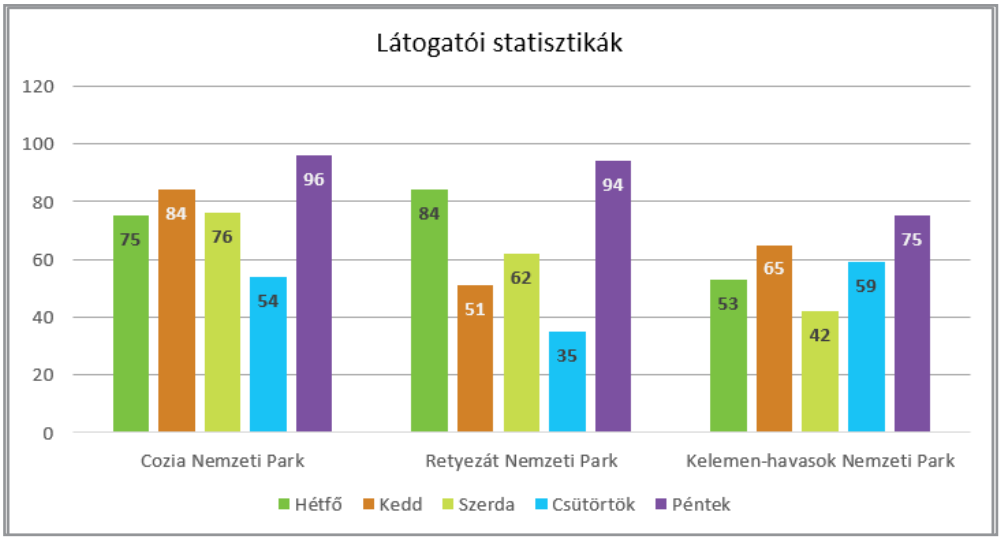
Diagram testreszabása

Kattints a diagramra, majd a **+** jelre, amelyet a diagram jobb felső sarkában találsz. **Adj hozzá új elemet, törölj vagy módosítsd** a *diagram-összetevőket!*

	A	B	C	D	E	F
1		Hétfő	Kedd	Szerda	Csütörtök	Péntek
2	Cozia Nemzeti Park	75	84	76	54	96
3	Retyezát Nemzeti Park	84	51	62	35	94
4	Kelemen-havasok Nemzeti Park	53	65	42	59	75



Egy tengely lehet **értéktengely** (numerikus adatokat képviselhet) vagy **kategóriatengely** (szöveges adatokat, dátumot stb. képviselhet).
A **vízszintes tengelyt** kategóriatengelynek is nevezik. Az adatsorok a **függőleges tengelyen** vannak ábrázolva. Jelöld ki a diagramot, a *Design* fülön a *Data* csoportban kattints a *Switch Row/Column* gombra, és megkapod az alábbi diagramot.
Az első diagram soronként ábrázolja az adatsort, és **3** oszlopot tartalmaz az **5** nap mindegyikére.
A második diagram oszloponként ábrázolja az adatsort, és mind a **3** rezervátum **5** oszlopot tartalmaz.



GYAKORLAT // VÉDD A NÖVÉNYVILÁGOT!

România növényvilágát a hazánk területén azonosított és osztályozott növények összessége képezi. Amint azt a földrajzórákon tanultad, a növényzet rétegekben oszlik el, a talaj és az éghajlat tulajdonságainak, valamint az egyes földrajzi területek magasságának megfelelően. A természeti környezet jelentős romlása miatt Románia növényvilága is valós veszélyben van, kipusztulhatnak ritka vagy akár egyedülálló növényfajok! Ezért léteznek azok a törvények, amelyek bizonyos növényfajokat védenek. Nagyon fontos az egyéni hozzájárulásod a növényvilág védelméhez, mert azok a gyakorlati tevékenységek, amelyekbe bekapcsolódsz, az adatfeldolgozás és -értelmezés eszközeként a táblázatkezelőt használják.

Nyisd meg a **táblázatkezelő alkalmazást** és készítsd el a projektet, amelyben Románia növényvilágával kapcsolatos adatokat dolgozol fel. Mentsd le az állományt a portfóliódba **Flora** névvel!

	A	B	C	D	E
1	**Sor-szám**	**Ritka virágok**	**1. Nemzeti Park**	**2. Nemzeti Park**	**3. Nemzeti Park**
2	1	**Havasi gyopár**	53	58	63
3	2	**Bánáti bazsarózsa**	86	96	106
4	3	**Boldogasszony papucsa**	21	36	51
5	4	**Erdélyi májvirág**	23	43	63
6	5	**Mocsári kockásliliom**	10	35	60

1. lépés

Képletek segítségével töltsd ki az adatsorokat!

Formázd és írd be az adatokat az **A1:E5** cellatartományba. Töltsd ki az A oszlopban az adatsort a következő módon: az A3 cellába írd be az **=A2+1** képletet és másold le azt az **A4:A6** tartományba.

A **C** oszlopban minden ritka virágfajra rögzítettük a turisták által az **1. Nemzeti Parkban** készített fényképek számát.

A **C2:E2** cellatartomány adatsorában a *lépés* **5**. Írd be a **D2** cellába a **=C2+5** képletet és másold le az **E2** cellába.

A **C3:E3** cellatartomány adatsorában a *lépés* **10**. Írd be a **D3** cellába a **=C3+10** képletet és másold le az **E3** cellába.

Fedezd fel a lépések értékeit, és fejezd be a következő sorozatokat **C4:E4, C5:E5** és **C6:E6.**

2. lépés

Készíts egy diagramot!

Miután beírtad az adatokat a táblázatba, készítsd el a **B1:C6** cellatartomány adatainak grafikus megjelenítését egy diagramon. Változtasd meg minden sorozat kitöltőszínét egy jellemző kép használatával. Ehhez jelölt ki az adatsort és a *Format* fülön, a **Shape styles** csoportból válaszd ki a *Shape fill* gombot, majd innen a *Picture…* opciót.

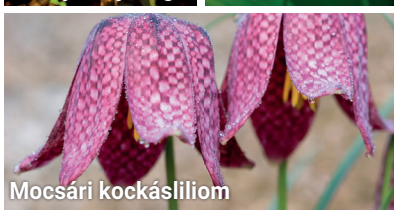
1. Nemzeti Park

Bánáti bazsarózsa

Havasi gyopár

Erdélyi májvirág

Boldogasszony papucsa


Mocsári kockásliliom

JAVASOLT GYAKORLATOK

I. Fedezd fel Románia ritka, törvény által védett virágait:

Havasi gyopár

Erdélyi havasszépe

Turbánliliom

Hegyi kökörcsin

Mocsári kockásliliom

Vékonylevelű szegfű

II. Gyakorolj!

Határozd meg, melyik a helyes válasz:

1. Nem eleme a diagramnak:
a. Jelmagyarázat
b. Diagram címe
c. Tengelyek
d. Képlet

2. Nem diagramtípus:
a. Sugár
b. Terület
c. Nagyító
d. Kombinált

3. Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést.

Egy munkalapon:

a. Az adatsor olyan értékekből áll, amelyek leírják egy mennyiség változását.

b. Egy adatsor képletek segítségével tölthető ki.

c. Az Excel alkalmazásban nincsenek beépített adatsorok.

d. Egy lineáris sorozatban a második számtól kezdve bármely számot kiszámíthatunk az előtte lévő számhoz egy lépés hozzáadásával.

III. Alkalmazd!

A portfóliódban készítsd el a **Látogatók.xlsx** nevű munkafüzetet, töltsd ki adatokkal az **A1:E1** tartományt.

	A	B	C	D	E
1	1	Hétfő	Január	100	1-Jan.
2	2	Kedd	Február	200	2-Jan.
3	3	Szerda	Március	300	3-Jan.
4	4	Csütörtök	Április	400	4-Jan.
5	5	Péntek	Május	500	5-Jan.

1. Minden oszlopban a sárga színnel jelölt cellát tekintjük aktív cellának. A *Home* fül, *Editing* csoport, *Fill* gomb használatával töltsd ki az adatsorokat a következő tartományokban: **A2:A12, B2:B12, C2:C12** és **D2:D12.**
2. A **C1:D12** tartomány adataiból készíts egy *X Y (Scatter)* diagramot *(pontokból álló diagram)*.

JEGYEZD MEG!

- Egy munkalapon **adatsorokat** másolással vagy a *Fill* gombbal készíthetünk.
- A diagramokat két dimenzióban (**2D**) vagy térben (**3D**) lehet ábrázolni.
- A diagram típusa módosítható a *Design* fül, *Type* csoport, *Change Chart Type* gomb segítségével.

ISMÉTLÉS /// CSAPATMUNKA!

Az előző leckék során megtanultatok táblázatkezelőkkel dolgozni. Minden tevékenységet felelősségteljesen kezeltetek. Képzeljétek el, hogy egy iskolában **4** VIII. osztály van, **A, B, C** és **D** osztályok. A *Légy felelősségteljes, védd!* témában együtt kell tevékenykednetek az iskolatársaitokkal.

Itt vannak azok a lehetséges **táblázatok**, amelyeket a csapatok használhatnak a kidolgozott projektek végén a kapott adatok tárolására, feldolgozására és bemutatására. A feladatod az, hogy válassz egy csapatot, vagyis a négy nyolcadik osztály egyikét, és végezd el a csapatod táblázatában szereplő feladatokat. Töltsd ki a táblázatot legalább **10** sorral, majd **képletek** és **függvények** segítségével töltsd ki a sárga színű oszlopokban a numerikus adatokat. **Sok sikert!**

8/A osztály
Kulturális értékek

*Exportáld az adatokat **pdf formátumban**!*

	A	B	C	D	E	F
1	Sor-szám	Művészeti esemény	Eladott jegyek száma	Jegy ára	Jegy árának a 19%-a	Eladott jegyek ára összesen
2	1	Fuvolaest	178	30	5,7	5340
3	2	Szimfónikus koncert	256	40	7,6	10240
4	3	Zongoraest	359	35	6,65	12565

8/B osztály
Nemzeti parkok

*Készíts egy diagramot a **B** és **C** oszlopokban található adatok alapján!*

	A	B	C	D	E	F	G	H
1	Sor-szám	Nemzeti Park	Látogatók száma			MIN	MAX	Összesen
2			Hétfő	Szerda	Péntek			
3	1	Cozia Nemzeti Park	45	54	63	45	63	162
4	2	Csalhó Nemzeti Park	98	78	83	78	98	259
5	3	Kelemen-havasok Nemzeti Park	37	25	87	25	87	149

8/C osztály
Állatvilág

*A **G1** cellába számítsd ki a madarak darabszámának számtani középarányosát!*

	A	B	C	D	E	F
1	Sor-szám	Nemzeti Park	Madár neve	Madarak száma	Madarak számának az 5%-a	Különbség (D-E)
2	1	Szemenik-Krassói-szurdok Nemzeti Park	Holló	63	3,15	59,85
3	2	Kelemen-havasok Nemzeti Park	Süvöltő	45	2,25	42,75
4	3	Retyezát Nemzeti Park	Siketfajd	26	1,3	24,7

8/D osztály
Növényvilág

*Rendezd a táblázat adatait csökkenő sorrendbe a Nemzeti Park és az Összeg **(D+E)** oszlopok szerint!*

	A	B	C	D	E	F
1	Sor-szám	Nemzeti Park	Növényfaj neve	Növényfajok száma	Növényfajok számának a 15%-a	Összeg (D+E)
2	1	Retyezát Nemzeti Park	Erdélyi havasszépe	24	3,6	27,6
3	2	Retyezát Nemzeti Park	Havasi gyopár	46	6,9	52,9
4	3	Szemenik-Krassói-szurdok Nemzeti Park	Foltos ujjaskosbor	33	4,95	37,95

A tanároddal közösen határozzátok meg a projektek megvalósítására szánt munkaidőt, amelyet majd megosztasz a társaiddal. Gratulálok!

Felmérés

1

Társítsd az interfész elemeit nyilakkal a megfelelő definíciókhoz!

Elem	Meghatározás
címsor	a munkaterületet lehet vele balra-jobbra, felfelé és lefelé görgetni
állapotsor	az aktív cellából megjeleníti a számítási képletet
görgetősáv	a dokumentumnézet vezérlő elemeit és a nagyítási szinteket tartalmazza
szerkesztőléc	a fájl nevét és az ablakkezelő gombokat tartalmazza

2

Állapítsd meg a következő kijelentések igazságértékét!

Egy táblázatkezelő alkalmazásban:
a. Megváltoztathatod egy munkalap nevét.
b. Egy munkalap több munkafüzetből áll.
c. Adatok rendezésénél több rendezési szempontot is használhatsz.
d. Képletek segítségével kitölthetsz adatsorokat.

3

Készítsd **Excel**ben egy táblázatot, amelybe:

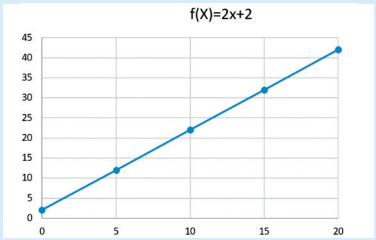
Vidd be az **A1:F2** tartomány adatait. Automatikus kitöltéssel vidd be a **B1:F1** adatsort. Tudva azt, hogy a B2 cellában a képlet **=2*B1+1**, képletek segítségével töltsd ki a **B2:F2** adatsort.

	A	B	C	D	E	F
1	x	0	5	10	15	20
2	f(x)	2	12	22	32	42

4

Készítsd **Excel**ben egy diagramot úgy, hogy:

A fenti táblázat **A1:F2** cellatartományának adataiból készítsd el a mellékelt diagramot a táblázat alá.



A teszt 10 pontot ér. Hivatalból egy pont jár. A javítókulcsban nem meghatározott elemek egyénileg értékelendők.
A munkaidőt a tanár határozza meg.

Légy felelősségteljes, védd!

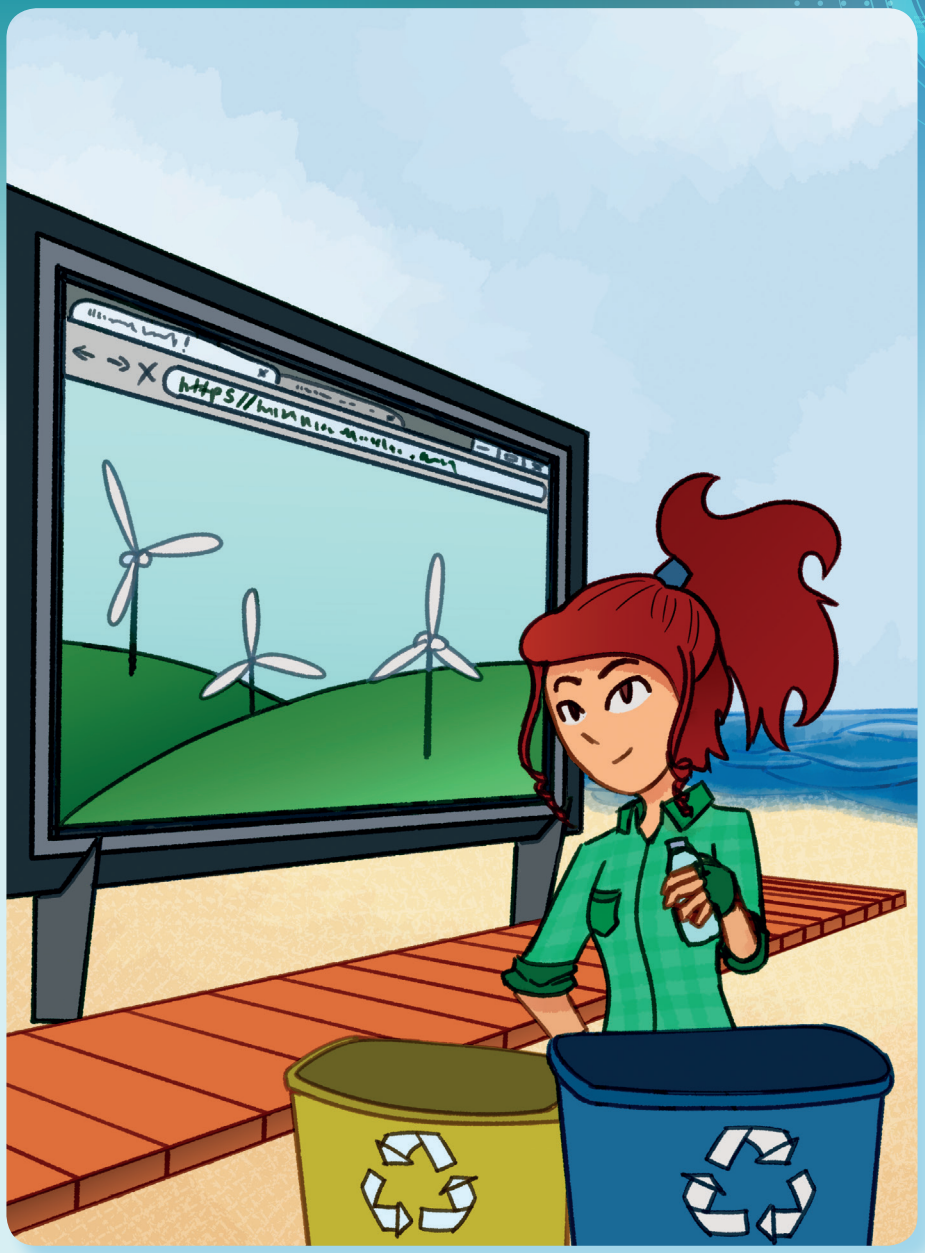
ÖNÉRTÉKELÉS

Elem	Értékelés	Pont
1	Minden helyes társítás **0,5p.** **0,5p × 4 társítás**	**2p.**
2	Minden helyes válasz **0,5p.** **0,5p × 4 társítás**	**2p.**
3	**A2:AF1** adatsor kitöltése **1p.** **B2:AF2** adatsor kitöltése **1 p.**	**2p.**
4	Diagram elkészítése **1p.** Igazítás a táblázat alá **1p.** Diagram cím megjelenítése **1p.**	**3p.**

ÚTMUTATÓ

Elem	Tartalom	Oldal
1	Az alkalmazás felületének ismerete.	12, 13, 14, 15
2	Az alkalmazás alapvető műveleteinek ismerete.	15, 18, 19, 26
3	Képletek, adatsorok és diagramok fogalmainak ismerete.	21, 22, 24, 25
4	A diagramok fogalmainak ismerete.	26, 27, 28

2. Egység Weblapok



Ebben az egységben megtanulsz:

- kihasználni és felismerni weblapok által kínált lehetőségeket;
- elemezni és szerkeszteni egy weblapot;
- biztonságos feltételek mellett dolgozni és védekezni a számítógépes csalások ellen.

Sajátos készségek: 1.2., 2.1., 3.2.

Ebben a leckében a webszerkesztők sokféleségét fedezheted fel. Szerkeszthetsz egy weblapot online vagy a saját gépeden. Lépésről lépésre megérted a HTML kódot és megismerkedsz egy weboldal alapvető elemeivel. Légy kreatív, válassz egy témát, és szerkeszd meg életed első weblapját! Légy felelősségteljes az online térben!

6. Lecke A weblapszerkesztő

A weblap változatos információkat tartalmaz: szöveget, képet, hangot, filmet és még sok más egyebet. Az információk egy jól meghatározott témakört járnak körbe. Több, egymással összekapcsolt weblap, ami egy adott témakörrel foglalkozik, **webhely**et alkot.

A weblapszerkesztő az az alkalmazás, amivel úgy szerkesztjük, formázzuk az összegyűjtött információkat, hogy ezáltal vonzó weblapot hozunk létre.

Egy weblap elkészítése nem a világ legnehezebb dolga, de nem is a legbanálisabb, viszont időt és sok türelmet igényel. Sőt, ami még fontosabb: mindig a jobbra kell törekedni, folyamatosan fejleszteni kell a már elkészített munkát!

Egy weblapkészítés algoritmusának lépései:

1. lépés	**Válassz ki egy szövegszerkesztőt!** Használhatod az operációs rendszer beépített szövegszerkesztőjét, dolgozhatsz online vagy telepíthetsz egy weblapszerkesztőt. A leggyakrabban használt egyszerű szövegszerkesztők a **Notepad** vagy **Notepad++**. Készíthetsz online weblapokat a **GoogleSite, WebWex, WebWave, Bitrix24, PageBreeze, Mobirise, OpenElement** szerkesztőkkel, vagy telepítheted a **WordPress, Microsoft FrontPage, Microsoft Office Publisher, Macromedia Dreamweaver, NVu, KompoZer** weblapszerkesztő programokat és még számtalan más programot.
2. lépés	**Fedezz fel kódokat!** Az újabb webszerkesztők lehetővé teszik, hogy a HTML kódolás ismerete nélkül készíts weblapot, de hasznos lehet a HTML kód ismerete, ha idővel tökéletesíteni szeretnéd azt az oldalt, amit készítettél. A HTML (Hyper Text Markup Language) a weblapszerkesztéshez használt nyelv. Használhatsz egy online szerkesztőt a **https://html5-editor.net/** -et vagy telepíthetsz egy alkalmazást, mint például: **Fresh HTML, WebMaster's HTML Editor, Alleycode** stb.
3. lépés	**Töltsd meg tartalommal!** A legvonzóbb az egészben a weblap tartalmának a megszerkesztése, mivel a munkád azonnal látható eredménnyel jár. Ha egy szövegszerkesztőt használsz, akkor **HTML** kódot kell írj, amit azonnal ellenőrizhetsz egy böngésző segítségével. Ha megbarátkoztál egy online webszerkesztővel, vagy telepítettél egyet a gépedre, és ezáltal megismerted az általuk felkínált lehetőségeket, ezekkel is megszerkesztheted az oldalad tartalmát.
4. lépés	**Teszteld az oldalt!** Ellenőrizd a tartalom elrendezését az oldalon és a formatervezést. Mentsd el az oldalt a mappádba. Ha ellenőrizted, feltöltheted. Indulásként ingyenes tárhelyet is használhatsz. Oszd meg a megvalósított munkádat az osztálytársaiddal és a barátaiddal!
5. lépés	**Folytasd!** Tökéletesítsd a megvalósított oldalakat. Rakd össze a weboldalakat, és hozz létre egy olyan oktató jellegű webhelyet, ami hasznos és vonzó!

Kiberbiztonság

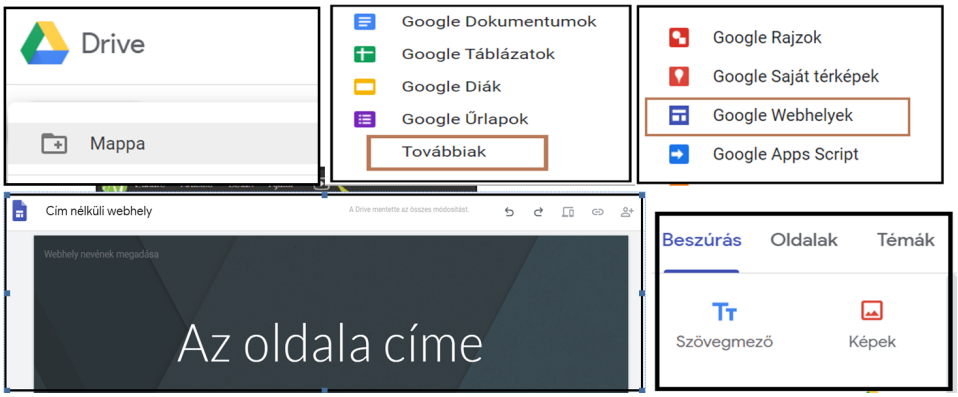
Mielőtt hozzálátnál egy weblap szerkesztéséhez, tisztába kell lenned azzal, hogy az online környezet használata a felkínált lehetőségek mellett kockázatokkal is jár. A kibertérben rosszakarókkal is találkozhatsz, akik cselekedeteikkel befolyásolhatják úgy a magánéletedet, mint az általad használt eszközök és programok biztonságát.

A kiberbiztonság normál állapotot jelent, amely biztosítja az elektronikus információk és erőforrások bizalmas, feddhetetlen és hiteles kezelését. Országos szinten vannak olyan akkreditált intézmények, amelyek kiberbiztonsági stratégiákat dolgoznak ki és alkalmaznak. **A kiberbiztonsággal foglalkozó mérnökök** olyan megoldásokat fejlesztenek ki és alkalmaznak, amelyek folyamatosan védik a webhálozatokat a kibertámadásoktól és fenyegetésektől.

Figyelem! Vigyázz az általad megosztott weblap tartalmára! Tartsd tiszteletben a szerzői jogot, légy tisztelettudó, védd a személyes adataidat, és csak olyasmit ossz meg, amibe minden érdekelt beleegyezett!

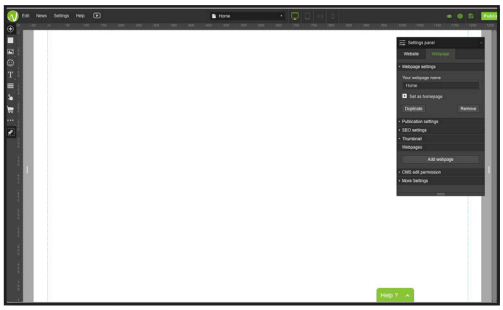
A Google Site szerkesztői felület elemei és alapvető eszközei

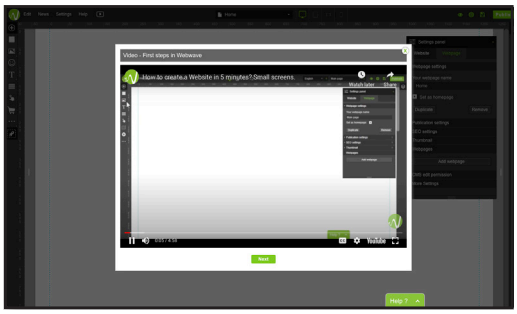
A saját **Google fiókodat használva** lépj be a **Drive**-ra és válaszd ki a Továbbiak opcióból az ingyenes Google Webhelyek alkalmazást. Az alkalmazás felülete barátságos. Rákkattintva a weblapra beszúrhatsz szöveget és képeket is



A WebWave szerkesztői felület elemei és alapvető eszközei

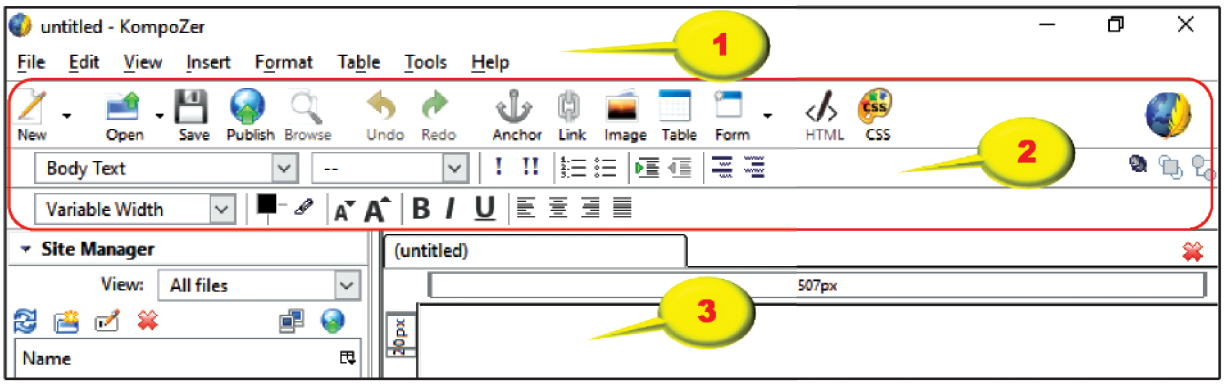
A **WebWave** alkalmazás ingyenes, és online elérhető a **https://ro.webwave.me/** címen. Miután bejelentkeztél egy érvényes e-mail címmel, máris készítheted a weblapokat. Az alkalmazás sok beépített sablonnal rendelkezik, de a nulláról kezdve is építhetsz weblapot.





A KompoZer szerkesztői felület elemei és alapvető eszközei

Ez az alkalmazás letölthető a **https://kompozer.software.informer.com/0.8/** címről. A weblapot a **Szerkesztő felületen (3)** kell készíteni a **Menüszalag (1)** és az **Eszköztár(2)** segítségével. A View menüpont segítségével megjeleníthető a weblaphoz rendelt **HTML kód**.



A weblap szerkezeti elemei

Egy böngésző segítségével weboldalak sokaságát lehet megjeleníteni. A használt böngészőtől függetlenül egy weboldal megnyitása után a **CTRL+U** billentyűkombinációval megnézheted az oldal **forráskódját**. Láthatod, hogy maga az oldal egy **HTML dokumentum**, a forráskód a **<html>** címkével kezdődik, és a **</html>** címkével ér véget. Minden **< >** jelből alkotott zárójel közé helyezett címke parancsot küld a böngészőnek az oldal megjelenítési módjáról. Érdemes megjegyezni, hogy nem minden böngésző ismer fel minden HTML címkét, éppen ezért ajánlott az elkészült oldalt minél több böngészővel tesztelni.

A HTML címkék jellemzői:

- **< >** jelből alkotott zárójel közé kell írni;
- a címkék általában párban jelennek meg, van egy nyitócímke **< >**, és egy zárócímke **</ >**;
- a szöveg, amit a jelek közé írunk, a böngészőnek küldött tulajdonképpeni parancs;
- mindegy, hogy az írásnál kisbetűs vagy nagybetűs írásmódot használunk.

A HTML címkék tulajdonságai:

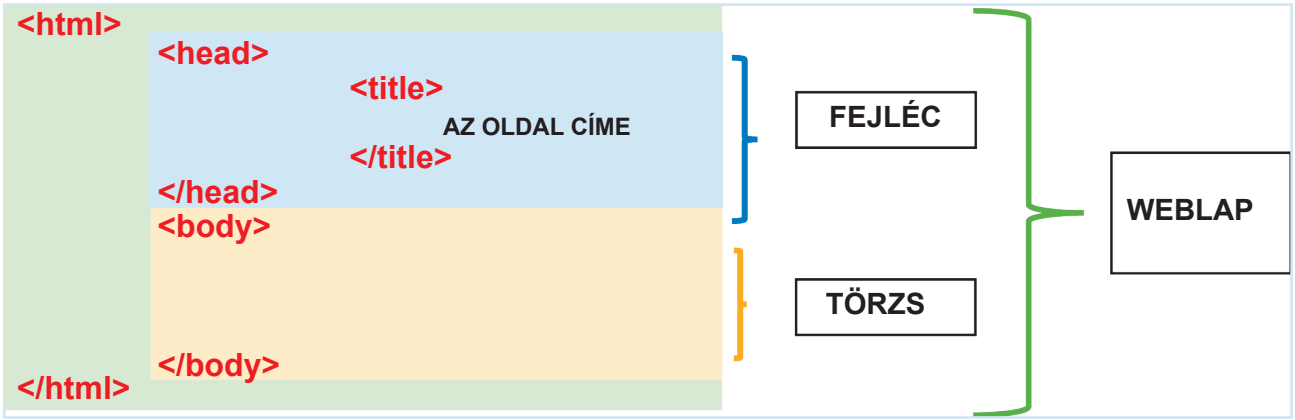
- bővebb információkat küld a böngészőnek a weblap elemeiről;
- mindig csak a nyitócímke zárójelei **< >** között jelennek meg;
- formailag így néznek ki: **tulajdonság neve = tulajdonság értéke**, és a tulajdonság értékét mindig **" "** (idézőjel) közé tesszük.

Bármely weblap tartalmazza az alábbi címkéket:

<html>	a **HTML kód** kezdete	**<title>**	**a weblap címe**
</html>	a **HTML kód** vége	**</title>**	a cím vége
<head>	**a weblap fejléce**	**<body>**	**a weblap törzse**
</head>	a fejléc vége	**</body>**	a törzs vége

Fontos! A weblap fejléce csak a forráskód megnyitásakor látható.
A weblap törzsét anélkül is mutatja a böngésző, hogy a forráskódot megnyitnánk.

A weblap szerkezete



A <body> címke tulajdonságai

bgcolor	a háttér színe	**link**	a meg nem látogatott linkek színe
background	a háttérkép	**alink**	egy aktív link színe
text	a szöveg színe	**vlink**	a meglátogatott linkek színe

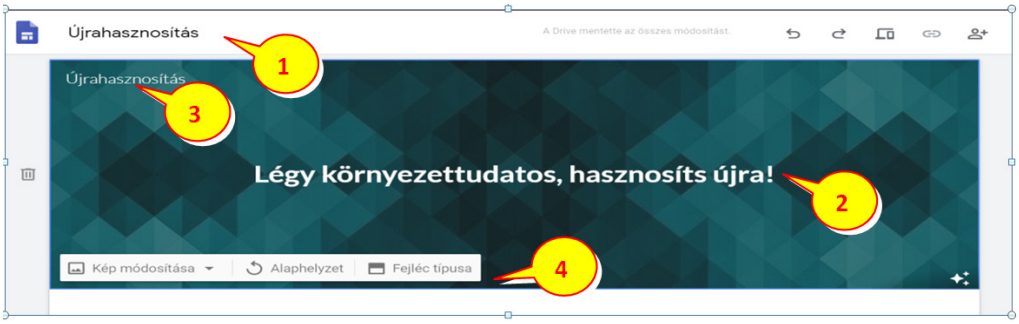
GYAKORLAT

LÉGY KÖRNYEZETTUDATOS, HASZNOSÍTS ÚJRA!

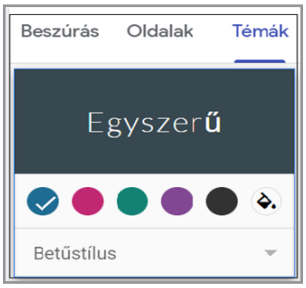
A környezetvédelem egyik módja az újrahasznosítás. Ezt úgy tudatosítsd, hogy az első **weblapod** címe legyen **Újrahasznosítás**.

1. lépés

Nyisd meg a **Google Webhelyeket!** A **Fejléc rész (1)**-be írd be a weblap címét: „Újrahasznosítás”. Figyeld meg, hogy ugyanezt a címet az alkalmazás automatikusan betölti a **(3)**-as részbe is; ez lesz az általad készített **Google Webhely** neve..

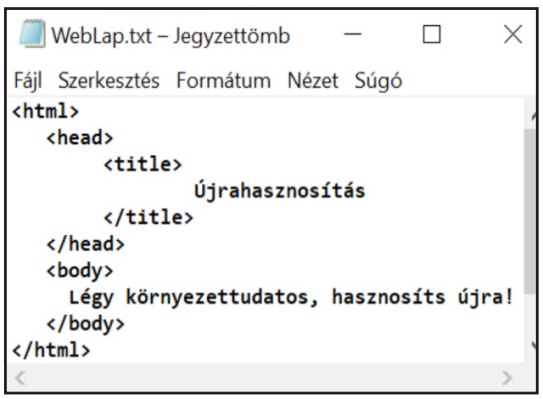


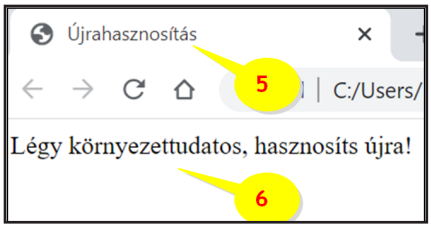
A **Menüből(4)** válaszd ki a fejléc megfelelő típusát. Szúrj be egy háttérképet az alkalmazás galériájából. Írd be a *„Légy környezettudatos, hasznosíts újra!”* **(2)** szöveget. Keress a beírt szövegnek egy témát, egy stílust, egy színt. Az oldalt az alkalmazás automatikusan menti a saját fiókod Drive-jába. Jobb klikk az oldalon, válaszd ki az **Oldal forrásának a megtekintését** és értelmezd a **HTML kódot.**

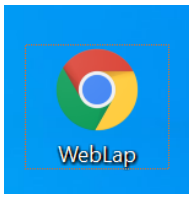


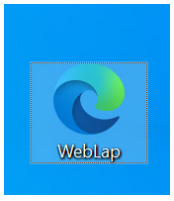
2. lépés

Hogy jobban megértsd a HTML kódot, nyiss meg egy Notepad (Jegyzettömb) állományt, és írd be az alábbi sorokat:









3. lépés

A **Jegyzettömb** állományt mentsd el a saját mappádba. Az állomány kiterjesztése **.html** legyen. Válaszd ki a Fájl típusa: **Minden fájl (*.*)**. Nyisd meg a **WebLap.html**-t egy böngésző segítségével. Figyeld meg a weblap címét a böngésző **címsorán(5)** és a tartalmat a weblap **törzsében(6)**.

Ügyes vagy! Elkészítetted és elmentetted az első weblapod a saját mappádba!

KITŰZÖTT FELADATOK

I. Védd a környezetet azzal, hogy újrahasznosítasz!

Újrahasznosítás jele	**Védd a bolygót!**
Elektromos hulladék újrahasznosítás	**Védd a környezetet!**
Az újrahasznosítási fa	**Az újrahasznosítás előnyei!**
http://y2u.be/ffbACw20BdY	**Hatékony újrahasznosítás.**
http://y2u.be/_E4QJViBcY0	**A műanyag újrahasznosítása!**
http://y2u.be/Gs2vgc_CFes	**Az újrahasznosítás, otthon kezdődik!**

II. Gyakorolj!

1. Határozd meg a helyes választ!

A **HTML** kifejezés jelentése:

a) egy kapcsolat	b) egy szabvány	c) egy programozási nyelv	d) egy dokumentum

2. Határozd meg a következő állítások igazságértékét, használd az I (Igaz) vagy H (Hamis) jelölést!

a. Az összekapcsolt weblapok halmaza weblinket alkot.
b. A weblap törzse megjelenik a böngészőben.
c. A weblap fejléce hozzátartozik a weblap törzséhez.
d. A kiberbiztonság az elektronikus információk feddhetetlenségét szavatolja.

III. Alkalmazd!

1. Írd be a mellékelt **HTML kódot** egy állományba a **Jegyzettömb**öt használva. Mentsd el az állományt **SajátOldal.html** néven a mappádba.
2. Nyisd meg az állományt egy szabadon választott böngészővel, és tekintsd meg a sorok hatását.
3. Tetszőlegesen változtasd a **<body>** címke tulajdonságait, és figyeld meg a változtatások hatását.

SajátOldal.txt – Jegyzettömb

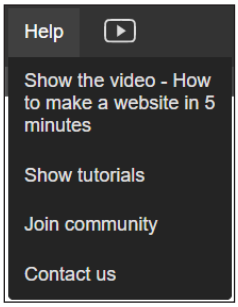
Fájl Szerkesztés Formátum Nézet Súgó

```
<html>
   <head>
        <title>
                Légy bölcs!
        </title>
   </head>
   <body bgcolor="green" text="white">
     Ne szennyezd a környezetet!
   </body>
</html>
```

8. sor, 23 100% Windows (CRLF) UTF-8

IV. Fedezd fe a WebWave-t.

1. Nézd meg a tanítóvideót, hogy megtanulj weblapot készíteni vele.
2. Az *Help* menüből válaszd ki az *a Show tutorials* opciót, hogy megértsd a különböző elemek szerepét.



JEGYEZD MEG!

- Rengeteg olyan online vagy letölthető program létezik, aminek segítségével **weblapot** lehet szerkeszteni.
- A weblap nem más, mint egy **HTML** kiterjesztésű állomány. A forráskód a **<html>** címkével kezdődik, és a **</html>** címkével ér véget.
- Egy weblap feltöltése az internetre nagy felelősséggel jár! Tisztelettudónak, udvariasnak kell lenned, nem szabad diszkriminálni, és tiszteletben kell tartani a szerzői jogot!

Ebben a leckében megtanulod, hogyan építs weblapot egy általad választott témában. Ahogy eddig megfigyelhetted, számtalan webszerkesztő létezik. Tehát egyaránt dolgozhatsz online vagy offline módban. A HTML kód segítségével formázhatod a szöveget és beszúrhatsz képeket is. Válaszd ki felelősségteljesen a weboldal elemeit, és szerkeszd türelemmel! Légy kreatív!

7. Lecke A weblap elemeinek szerkesztése

A weblap elemei: szöveg, kép, táblázat, lista, linkek stb.
A weblap elemeinek a szerkesztésével beilleszthetted és úgy módosíthatod az elemeket, hogy vonzó és működő weboldalt hozol létre.

Alapvető szerkesztési műveletek HTML kód használatával

A cím címke – **6** ilyen címke létezik: **<h1> </h1> <h2> </h2> … <h6> </h6>**. Ezeket a szöveg különböző szintű címeinek a beállítására használhatjuk.

A főcímet a **<h1>** címkével készítjük, ami a legnagyobb betűméretet eredményezi, és meg is vastagítja a szöveget. Az első alcím a **<h2>** címkével készül, és így tovább egészen a legkisebb címig, amit **<h6>** címkével készítünk.

Példa:

```
<body bgcolor="green" text="white">
        <h1>Takarékoskodj az árammal!</h1>
        <h2>Takarékoskodj az árammal!</h2>
        <h3>Takarékoskodj az árammal!</h3>
        <h4>Takarékoskodj az árammal!</h4>
        <h5>Takarékoskodj az árammal!</h5>
        <h6>Takarékoskodj az árammal!</h6>
</body>
```

Takarékoskodj az árammal!

Takarékoskodj az árammal!

Takarékoskodj az árammal!

Takarékoskodj az árammal!

Takarékoskodj az árammal!

Takarékoskodj az árammal!

Figyelem! Az ékezetes betűk helyes megjelenítéséhez szükséges, hogy a böngészőnek megüzend a használt állomány típusát. Tehát, mielőtt elkezdenéd a weblap szerkesztését, használd a **<meta>** címkét az alábbi példa szerint:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<html>
```

A bekezdés címke – a beírt szöveg bekezdésekre tagolható a ** ** címke segítségével.
Figyelem! Ha egy bekezdésen belül szeretnél sortörést, akkor a **
** címkét kell használni.
Ennek a címkének nincs zárócímkéje! *Példa:*

```
<body bgcolor="green" text="white">
        <p> A zöld energia:<br> megújuló és nem környezetszennyező
energia.</p>
      <p>A szélenergia egyfajta zöld energia.</p>
</body>
```

A zöld energia:
megújuló és nem környezetszennyező energia.

A szélenergia egyfajta zöld energia.

A pontozott lista címke – egy pontozott listát az ** ** címkével hozunk létre, a lista minden elemét a ** ** címkével készítjük el. A lista elemeinek a formáját a type tulajdonsággal állítjuk be. Lehetséges értékei: **disc (korong), circle (karika), square (négyzet)**.
Példa:

Megújuló energiaforrások:
- a napenergia;
- a szélenergia;
- a folyóvíz energia.

```
<body text="green">
     <ul>Megújuló energiaforrások:
       <li type="square">a napenergia;</li>
       <li type="circle">a szélenergia;</li>
       <li type="disc">a folyóvíz energia.</li>
     </ul>
 </body>
```

A számozott lista címke – egy számozott listát az ** ** címkével hozunk létre, és a lista minden elemét a ** ** címkékkel készítjük el. A számozás típusát a **type** tulajdonsággal állítjuk be. Lehetséges értékei:
A (nagybetűk), a (kisbetűk), I (nagy római számok), i (kis római számok), 1 (alapértelmezett).
Példa:

```
<body text="red">
     <ol>Meg nem újuló energiaforrások:
      <li type="i">a szén;</li>
       <li type="i">a kőolaj;</li>
       <li type="i">a földgáz.</li>
     </ol>
   </body>
```

Meg nem újuló energiaforrások:
i. a szén;
ii. a kőolaj;
iii. a földgáz.

A kép címke– egy képet az **<img>** címkével tudunk elhelyezni egy weblapon. Ennek a címkének a tulajdonságai:
src = forrás, a kép elérési útja
width = a kép szélessége pixelben
height = a kép magassága pixelben
alt = az a szöveg, ami a kép helyett jelenik meg addig, amíg a kép betöltődik, vagy akkor, ha a kép nem található.
border =keret vastagsága (0 = nincs keret)
align = a kép igazítása, lehetséges értékei:
left = bal, **right** = jobb, **center** = közép

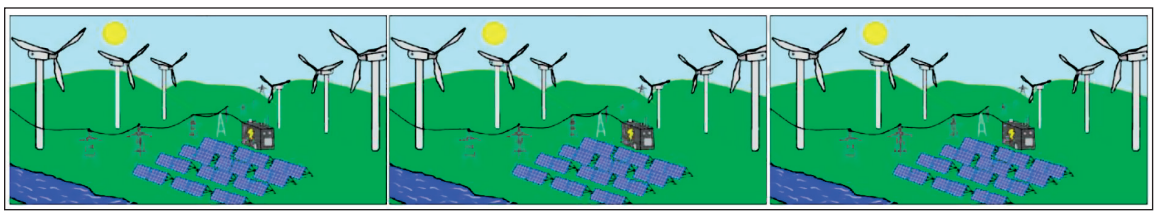
Figyelem! Ajánlott, hogy egy kép beszúrásánál mindig legyen jelen a **width** és **height** tulajdonság, így a böngésző helyesen jeleníti meg a képet. Amennyiben a két tulajdonsággal a **width** és **height**- al beállított érték különbözik a kép eredeti méretétől, a böngésző átméretezi a képet.

Vigyázz! A weboldalnak és a beszúrt képnek ugyanabban a mappában kell lenni, a saját mappádban, különben a böngésző az alt tulajdonságnál megadott üzenetet fogja kiírni.

A weboldalak által megtűrt képformátumok: **.jpg, .jpeg, .gif, .bmp, .png.**

Példa:

```
<body>
 <img src="poza.png" width="502" height="250" alt="poza"  border=1 align="left">
 <img src="poza.png" width="502" height="250" alt="poza"  border=1 align="center">
 <img src="poza.png" width="502" height="250" alt="poza"  border=1 align="right">
</body>
```



GYAKORLAT

LÉGY MEGGONDOLT, TAKARÉKOSKODJ AZ ENERGIÁVAL!

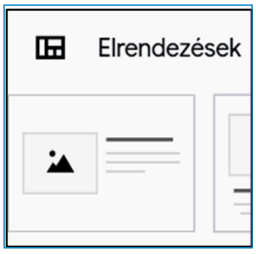
Mivel a Föld szén-, kőolaj- és földgázkészletei megcsappantak, világszinten kutatnak a megújuló energiaforrások után. Tudatosítsd ezt, és készíts egy **weblapot Légy meggondolt!** címmel.

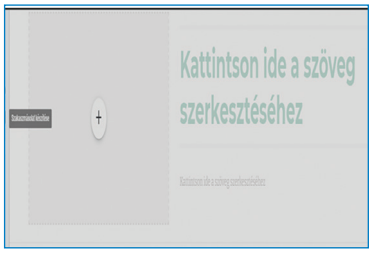
1. lépés

Nyisd meg a **Google Webhelyek**et, add a **Weblapod**nak a **Légy meggondolt!** nevet. Az **Oldal címe** részbe írd be a „Zöld energia” szöveget! Szúrj be egy találó képet a saját mappádból vagy az online térből. Válassz egy háttérszínt és egy témát az **Oldal címe** résznek úgy, ahogy azt a *Légy környezettudatos! Hasznosíts újra!* gyakorlatnál megtanultad.

2. lépés

Válassz egy találó Elrendezést a **weblap törzsének**. Szúrj be megfelelő szöveget és képeket. A beszúrt elemeket egyszerűen az egér görgetésével is szerkesztheted. Légy kreatív! Felelősségteljesen válaszd ki a szöveget, használj pontozott listát!

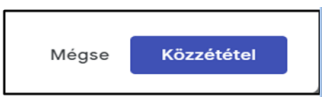


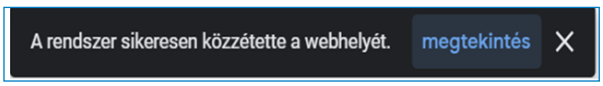


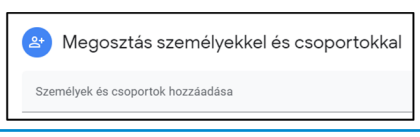
- a napenergia és a szélenergia, zöld energiaforrások, amikkel helyettesíteni lehet a hagyományos energiaforrásokat.
- A napenergia egy olyan megoldás amit napjainkban egyre gyakrabban használnak. Ennek az energiaforrásnak köszönhetően megtakarítjuk a hagyományos energiaforrásokat(kőolaj, földgáz,szén stb.) és ráadásul még környezetkímélőek is.

3. lépés

A **Google Webhelyek** lehetőséget ad arra, hogy a weblapodat feltedd az internetre. A weblapod lehet nyilvános, vagy csak azok láthatják, akiket meghívsz! **Ügyes vagy! Folytasd!**







FEDEZD FEL!

A definiciós lista címkéje

Egy definiciós lista létrehozása a **<dl> </dl>** címkékkel történik, a meghatározott elemek a **<dt> </dt>** címkék közé kerülnek, míg magát a meghatározást a **<dd> </dd>** címkékkel készítjük.

```
<body text="blue">
      Jelölések:
      <dl>
        <dt>LED</dt>
        <dd>Light Emited Diode</dd>
        <dt>OLED</dt>
        <dd>Organic Led</dd>
      </dl>
 </body>
```

Jelölések:

LED
Light Emitting Diode
OLED
Organic LED

KITŰZÖTT FELADATOK

I. Fedezd fel a megújuló energiaforrásokat:

Szélenergia!

Napenergia!

Vízenergia!

Árapályenergia!

Geotermikus energia!

Biomassza energia!

II. Gyakorolj!

1. Határozd meg a helyes választ!

A HTML kóddal készített lista nem lehet:

a) számozott	b) definíciós	c) pontozott	d) véletlenszerű

2. Határozd meg a következő állítások igazságértékét! Használd az I (Igaz) vagy H (Hamis) jelölést!

a. A **Google Webhelyek** egy táblázatkezelő program.
b. Egy weblap törzsébe képeket is beszúrhatunk.
c. A **
** címke párja a **</br>**.
d. Az ** ** címkékkel sorszámozott listákat készíthetünk.

III. Alkalmazd!

1. Válassz egy weblapszerkesztőt, és szerkeszd meg az alábbi szöveget. Szerkeszd úgy, hogy egyetlen bekezdésben egy pontozott lista legyen.

BODY

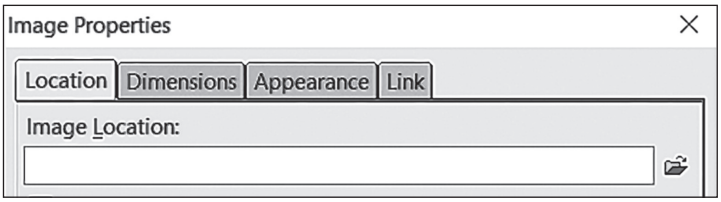
A zöld energiát előállító termékek:

- vékonyfilm-rétegű napelemek
- szélerőművek
- napelemek

IMG

2. Keress egy megfelelő képet is ugyanarra az oldalra. Mentsd a weblapod a saját mappádba **Zöld termékek** névvel.

3. Dolgozz csapatban, a társaddal együtt fedezd fel a **KompoZer** alkalmazás azon eszközét, amellyel képeket lehet szerkeszteni és beszúrni egy weblapra.



IV. Fedezd fel!

Nyisd meg a Légy meggondolt! gyakorlatnál általad elkészített weblapot. A megnyitáshoz használd a hivatkozást. Fejleszd tovább a weblapod újabb szövegek és képek hozzáadásával. Oszd meg osztálytársaiddal és barátaiddal!

JEGYEZD MEG!

- **Egy weblap elemei** lehetnek szövegek, képek, táblázatok, hivatkozások stb.
- A weblapelemek a **HTML kód** segítségével szerkeszthetőek.
- A beszúrt képek a weblappal egy mappába kell legyenek ahhoz, hogy a böngésző helyesen jelenítse meg.
- Ékezetes betűk használatához a **<meta>** címkére van szükség.

Ebben a leckében megtanulod, hogyan szerkeszd meg és formázd a weblapod elemeit. Dolgozhatsz online - megosztva társaiddal a megvalósított munkád-, vagy dolgozhatsz a saját gépeden, felfedezve ezáltal a HTML kód ismerete kínálta lehetőségeket. Végül összefűzöd a weblapjaidat, és így megvalósítasz egy webhelyet. Haladj tovább türelmesen, fejleszd magad!

8. Lecke A weblap elemeinek a formázása

A weblap elemeinek a formázása lehetővé teszi a weblap megszemélyesítését és az elemek elrendezését az oldalon. Ezen elemeknek az összekapcsolásával létrehozhatsz egy **webhely**-et.

A weblap alapvető elemeinek a formázásai HTML kód használatával

Szöveg címkék – Rengeteg lehetőség nyílik a szöveg formázására. Tanulmányozhatod a tankönyv végén levő **Függelék – HTML szövegformázás** és **Függelék - HTML színek** részeket. A leggyakrabban használt szövegformázások: **félkövér **, **dőlt **, **aláhúzott <u> </u>**, **nagy <big> </big>**, **kicsi <small> </small>**. A karakterek formázására a ** ** címkét használjuk, amelynek tulajdonságai a **size** = karakter mérete, **face** = a karakter formája és a **color** = a karakter színe.

Példa:

```
<body bgcolor="blue">
        <font face="Courier" size="6" color="white"><b>Takarékoskodj a vízzel!</b></font>
        <font face="Verdana" size="4" color="yellow"><i>Takarékoskodj a vízzel!</i></font>
        <font face="Arial" size="2" color="red"><u>Takarékoskodj a vízzel!</u></font>
</body>
```

Takarékoskodj a vízzel! ***Takarékoskodj a vízzel!*** Takarékoskodj a vízzel!

```
<body bgcolor="blue">
        <font face="Courier" size="6" color="white">
                <b>Képletek:<br>H<sub>2</sub>O,
                CO<sub>2</sub></b></font>
</body>
```

Képletek:
H_2O, CO_2

A hivatkozás címke – egy hivatkozás létrehozására az **<a> ** címkét használjuk. A hivatkozás nagyon fontos, rajta keresztül történik az átlépés egyik oldalról a másikra. Általában a hivatkozás, a link alá van húzva, és megváltozik a formája és a színe, amikor az egér mutatóját rágördítjük.

A hivatkozás tulajdonságai:	A **target** tulajdonság lehetséges értékei:
href = a célállomány címe **target** = melyik ablakban nyíljon meg a célállomány **title** = a hivatkozás leírása, akkor jelenik meg, ha az egér mutatója a hivatkozásra mutat	**blank** = az oldal egy új ablakba töltődik be **parent** = az oldal az aktuális ablakot elindító ablakba töltődik be **top** = az oldal a legfelső ablakba töltődik be **self** = az oldal az aktuális ablakba töltődik be

A célállomány címe a következő formákban lehet megadva:
1. *állomány neve,* ha ugyanabban a mappában van az aktuális oldallal: **<a href="allomany.html">**
2. *mappa/állománynév,* ha más mappában van **<a href="mappa/allomany.html">**
3. *a teljes URL cím* **<a href="http://www.konyv.ro/mappa/allomany.html">**

A táblázat címke – Egy táblázat beszúrására a **<table> </table>** címkét használjuk. Egy sor létrehozására a **<tr> </tr>** címkét, egy cella létrehozására a **<td> </td>** címkét használjuk.
A <table> címke **tulajdonságai:**
border = keret vastagsága (0 = keret nélkül)
width = a táblázat szélessége
height = a táblázat magassága
bgcolor = háttérszín
background = háttérkép
cellspacing = cellák közötti távolság
cellpadding = cella széle és tartalma közötti távolság

A <td> címke tulajdonságai:
align = a tartalom vízszintes igazítása, lehetséges értékek: **left** = bal, **right** = jobb, **center** = közép,
valign = a tartalom függőleges igazítása, lehetséges értékek: **top** = fent, **bottom** = lent, **middle** = középen,
width = cellaszélesség, **height** = cellamagasság, **bgcolor** = cella háttérszín, **background** = cella háttérkép, **colspan** = cellaegyesítés a szomszédos(jobbról) cellákkal, **rowspan** = cellaegyesítés az szomszédos(alulról) cellákkal

Figyelem! A táblázat első sorát kiemelhetjük, ha a **<td> </td>** címkék helyett a **<th> </th>** címkéket használjuk.

Példa:

piros	sárga
fehér	zöld

```
<body bgcolor="blue">
        <table border="1">
        <tr><td width="160" bgcolor="white">piros</td>
            <td width="160" bgcolor="yellow">sárga</td></tr>
         <tr><td bgcolor="red">fehér</td>
            <td bgcolor="green">zöld</td></tr>
    </body>
```

Év	2010	2015	2020
Hőmérséklet °C	0,48	0,52	0,57

```
<body><table border="3" bordercolor="blue" bgcolor="azure"
        align="center" cellspacing="10" cellpadding="15" style="width:30%">
        <tbody><tr><td bgcolor="LightSkyBlue"><b>Év</b></td>
                <td><b>2010</b></td>
                <td><b>2015</b></td>
                <td><b>2020</b></td></tr>
              <tr><td bgcolor="Aqua"><b>Hőmérséklet C<sup>o</sup></b></td>
                <td><b>0,48</b></td>
                <td><b>0,52</b></td>
                <td><b>0,57</b></td></tr>
        </tbody></table></body>
```

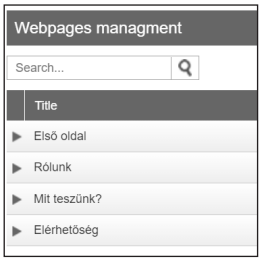
GYAKORLAT

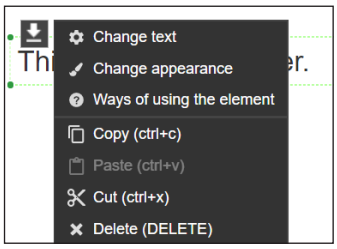
LÉGY MEGGONDOLT, TAKARÉKOSKODJ A VÍZZEL!

A jelen és a jövő egyik nagy globális kihívása a vízkérdés megoldása. Napjainkban közel 1,5 milliárd ember él vízhiány sújtotta területen. Tudatosítsd ezt, és készíts egy **weblapot *Takarékoskodj a vízzel!*** címmel!

1. lépés

Indítsd el a **WebWave** alkalmazást, és készíts néhány weblapot az adott témában úgy, hogy létrehozhasd a ***Takarékoskodj a vízzel!* webhelyet**. Minden oldalnak találd ki a megfelelő tartalmat. Az oldalt az alkalmazás kínálta eszközökkel szerkeszd és formázd!





2. lépés

A **Fejléc részbe** helyezz el egy logót, egy képet és egy üzenetet, ami kifejező a választott témával kapcsolatban.

3. lépés

A webhelyed tedd ki a **https://tek17n.webwave.me** címre

Szerkezd meg és formázd az oldalakat. Publikáld és oszd meg a **webhelyed. Gratulálunk!**

KITŰZÖTT FELADATOK

I. Fedezd fel a víz halmazállapotait:

Felhők – gáz halmazállapot!

Jég – szilárd halmazállapot!

Pára – gáz halmazállapot!

Dér – szilárd halmazállapot!

Hó – szilárd halmazállapot!

Az óceánok vize – folyékony halmazállapot!

II. Gyakorolj!

1. Határozd meg a helyes választ!

Melyik nem tulajdonsága a karakterek formázása címkének HTML kódban:

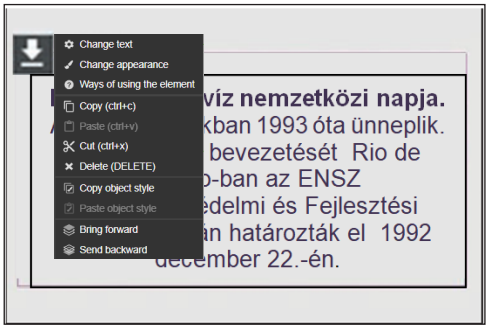
a) align	b) cell	c) big	d) small

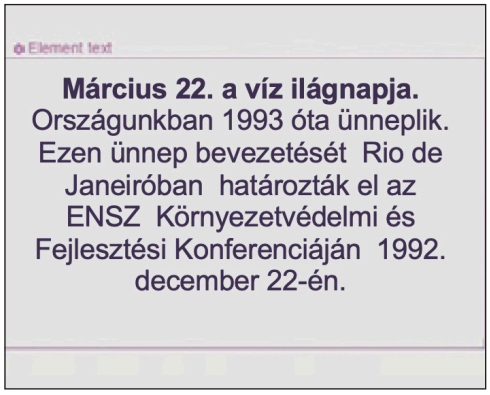
2. Határozd meg a következő állítások igazságértékét! Alkalmazd az I (Igaz) vagy H (Hamis) jelölést!

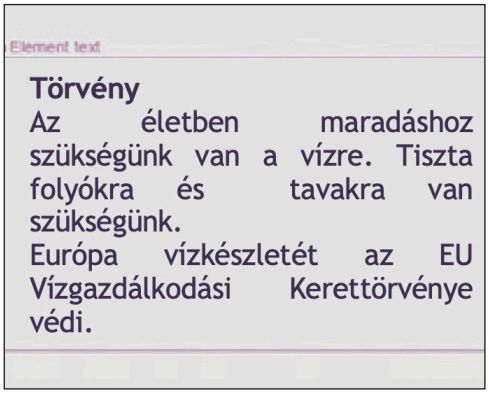
a. **WebWave** egy online webszerkesztő alkalmazás.
b. A dőlt betűs szövegformázás egyenértékű az aláhúzott szövegformázással.
c. A **** címke a szövegből felsőindexet készít.
d. Egy sor beszúrását a táblázatba a **<tr> </tr>** címkével készítjük.

III. Alkalmazd!

Látogasd meg az előző gyakorlatban elkészített **https://tek17n.webwave.me/**linket, nyisd meg a **Takarékoskodj a vízzel!** webhelyet. Fejleszd tovább az egyes oldalakat újabb szövegek és képek beszúrásával!







JEGYEZD MEG!

- **A weblap elemeinek a formázása** változatos **webhelyek** készítését eredményezi.
- Amikor egy weblapról hivatkozást(linket) készítesz egy másik weblapra, a link **HTML címkéjének** az illető weblap teljes címét kell tartalmaznia.
- Egy táblázat első sora kiemelhető fejlécként, ha a **<th> </th>** címkét használjuk.

ISMÉTLÉS /// CSAPATMUNKA!

A gyakorlatok során weblapot szerkesztettél és formáztál meg. A **Google Site** alkalmazást használva az osztálytársaiddal együtt csapatban, online dolgozhattok.

Képernyőfelvétel a webhelyről

Ghid pentru economisirea apei!

http://bit.ly/2SGwPDu

Készítsétek el a **Takarékoskodj!** webhelyet útmutatóul a vízzel való takarékoskodáshoz!

Datorită încalzirii globale, a poluării şi a suprapopularii, 1,4 miliarde km3 din apa globului se află în pericol. Resursele de apă devin limitate şi astfel trebuie să avem grijă cum consumăm apa. În prezent, 31 de state au dificultăţi în obţinerea apei potabile. În următorii 20 de ani se aşteaptă că acest număr să ajungă la 50. Această înseamna că 2,8 miliarde de oameni nu vor putea găsi apa potabilă.

Képernyőfelvétel a webhelyről

1. Csapat feladata:
Gyűjtsd össze és használd újra!

Képernyőfelvétel a webhelyről

2. Csapat feladata:
Használj kevesebbet!

1. Csapat – meghatározza a szükséges oldalak számát és a Fejléc formáját.
2. Csapat – elkészíti a webhely logóját, kitalálja a szükséges képgalériák számát.
Mindkét csapat megkeresi a webhely megfelelő információit. A webhelyre felkerülő információk legyenek hitelesek és tartsuk tiszteletben a szerzői jogot.

Folosirea apei în mod eficient nu resursele de apă, ci diminuează emisiile de gaze cu efect de seră, pentru că sistemele publice de apă utilizează multă energie pentru a purifica apa şi pentru a o distribui către consumatori. Cu cât foloseşti mai puţină apă caldă, cu atât se consumă mai puţină energie pentru a încălzi apa, ceea ce reduce emisiile.

Képernyőfelvétel a webhelyről

Miután formáztunk és beszúrtunk minden elemet, a tanár jóváhagyja a webhely feltöltését az online térbe.
Gratulálunk! Sikerült az együttműködés és létrehoztatok egy oktató jellegű webhelyet!

Hirdessétek és osszátok meg az oldalt!

- **Colectează apa de ploaie!**

De ce să îti uzi peluza cu apă potabilă? Fă rost de un butoi în care să strângi apa de ploaie şi apoi foloseşte-o la udatul plantelor.

Képernyőfelvétel a webhelyről

Felmérés

1

Kösd össze nyilakkal a **HTML címkét** a megfelelő hatással!

Címke	Hatás
**<a> **	megszerkeszti a legnagyobb méretű címsort
** **	sorszámozott listát készít
<th> </th>	beállítja fejlécként a táblázat első sorát
** **	hivatkozást készít egy weblapon

2

Határozd meg a mellékelt kijelentések igazságértékét!

A weblapszerkesztő:

a. Egy olyan alkalmazás, amivel weblapokat lehet készíteni.
b. **.ods** kiterjesztésű állományokat készít.
c. Csak online használható.
d. Kizárólag csak egy **HTML kód**szerkesztővel működik.

3

Alakítsd át a mellékelt **HTML kódot** úgy, hogy:

a. a weblap háttérszíne **piros** legyen, a szöveg színe **fehér**.
b. a lista legyen **pontozott**, a listajelek korong típusúak.

```
<body><text="blue">
   <ol>Halmazállapotok:
      <li type="i">szilárd;</li>
      <li type="i">folyékony;</li>
      <li type="i">gáz;</li>
   </ol>
</body>
```

4

Írd be a mellékelt **HTML kódot**, majd egészítsd ki úgy, hogy beszúrsz egy kétsoros és háromoszlopos táblázatot!

```
<body><table border="3" bordercolor="blue" bgcolor="azure"
 align="center" cellspacing="10" cellpading="15" style="width:30%">
 <tbody><tr><td bgcolor="LightSkyBlue"><b>Év</b></td>
          <td><b>2015</b></td>
          <td><b>2020</b></td>
 </tbody></table></body>
```

A táblázat második sorában az jelenjen meg: **Hőmérséklet 20 30**

A teszt 10 pontra van értékelve. Egy pont jár hivatalból. Azokat az elemeket, amelyekre a teszt nem tér ki, szabadon állíthatod be. A munkaidőt a tanár szabja meg.

Légy megfontolt, takarékoskodj!

ÖNÉRTÉKELÉS

Elem	Értékelés	Pont
1	Minden helyes társítás **0,5p.** **0,5p × 4 társítás**	**2p.**
2	Minden helyes válasz **0,5p.** **0,5p × 4 válasz**	**2p.**
3	Lista beszúrása **1p.** Elemek beállítása **1p.**	**2p.**
4	Kód beírása **1p.** Táblázat beszúrása **1p.** Táblázat feltöltése **1p.**	**3p.**

ÚTMUTATÓ

Elem	Tartalom	Oldal
1	A HTML kód ismerete	37, 38, 42, 44
2	Az alkalmazás kínálta lehetőségek ismerete	33, 34, 35
3	A lista elembeszúrásával és formázásával kapcsolatos ismeretek	34, 38, 39
4	A táblázat beszúrásával kapcsolatos ismeretek	34, 42, 44

3. Egység Algoritmusok

Ebben az egységben a következőkről fogsz tanulni:

- számsorok azonosítása és feldolgozása sajátos alkalmazásokkal;
- algoritmusírás a számsorok megépítéséhez és feldolgozásához;
- interdiszciplináris algoritmusok írása.

Sajátos kompetenciák: 2.1, 2.2, 3.1, 3.3

A lecke során vissza fogsz emlékezni az alapvető algoritmusokra, amelyek segítségével olyan programokat írhatsz, amelyek számsorokat képeznek. A programokat különböző programozási nyelveken fogod írni és különféle fejlesztői környezetekben fogod futtatni. A lecke végén képes leszel algoritmikusan felépíteni egy számsort figyelembe véve bizonyos szabályokat, valamint feldolgozni a számsor elemeit.

9. Lecke A számsor

A számsor számok sorozatát jelöli. A számsor lehet számok egyszerű felsorolása vagy megépíthető bizonyos tulajdonságok alapján. Léteznek olyan számsorok, amelyek megépíthetők és feldolgozhatóak algoritmusokkal, figyelembe véve bizonyos szabályokat.

Egy számsor megépítése olyan szabályok azonosításával történik, amelyek érvényesek a számsor összes elemére, vagy csak bizonyos elemekre egy adott intervallumból.

EMLÉKEZZ!

A mindennapi életben gyakran felmerülő problémák megoldásához **algoritmus**ra van szükség. Az algoritmus egy bizonyos sorrendben végrehajtott műveletek olyan sorozata, amely a **bemeneti adatok**ból kiindulva, véges idő alatt, helyes **kimeneti adatok**at eredményez. Többször használtál úgynevezett **alapvető algoritmus**okat, melyeket alkalmazhatsz valahányszor szükséges. Az algoritmusokat átírtad **programkód**ra, majd a **fejlesztői környezet**ek segítségével, különböző **programozási nyelv**eken írt programokat készítettél.

A leckék során megtanulsz összetett algoritmusokat készíteni, ezek tartalmazhatnak alapvető algoritmusokat is, amelyeket nem kell minden alkalommal megírni. Például egy algoritmus elkészítéséhez használhatod a következő **alapvető algoritmusok**at:

1. két változó értékének **felcserélése**
2. **egy szám számjegyeinek feldolgozása**
3. **egy szám osztóinak feldolgozása**
4. **számlálás**
5. **összegszámolás**
6. **szorzatszámolás**
7. **legkisebb/legnagyobb érték meghatározása**
8. **legnagyobb közös osztó (lnko) és legkisebb közös többszörös (lkkt) meghatározása**
9. **adott szabályok szerint alkotott számsorok feldolgozása**
10. **ismétlő szerkezettel beolvasott számsorok feldolgozása**

Két változó értékének felcserélése

Az algoritmust akkor használod, ha két változó értékét szeretnéd **felcserélni**. **Feladat**: Írj programot, amely beolvassa az **x** és **y** két valós számot és az értékeik **felcserélése** után kiírja az **x** és **y** változók értékeit, szóközzel elválasztva. Mentsd a programot **A1** névvel a portfóliódba! Példa: ha **x = 4** és **y = 18**, a kiírt értékek: **18 4**.

Pszeudokód

1. Módszer: A cserét a **seged** segédváltozó használatával végezzük el.	**2. Módszer:** A cserét segédváltozó használata nélkül végezzük el.	
beolvas x, y **seged** ← **x** // elmenti az **x** értékét a **seged**-be **x** ← **y** // az **y** értékét megfelelteti **x**-nek **y** ← **seged** // a **seged**-be elmentett értéket megfelelteti **y**-nak **kiír x,' ',y**	**beolvas x, y** **x** ← **x + y** **y** ← **x − y** **x** ← **x − y** **kiír x,' ',y**	**beolvas x, y** **x** ← **x − y** **y** ← **x + y** **x** ← **y − x** **kiír x,' ',y**

Figyelem! Ha az **x** és **y** változók értékeinek felcseréléséhez csak az **x** ← **y** és **y** ← **x** utasításokat használjuk, akkor az **x** változó értéke elvész!

Egy szám számjegyeinek feldolgozása

Az algoritmust akkor használod, ha egy **n** természetes **szám számjegyeihez szeretnél hozzáférni**.
Feladat: Írj programot, amely beolvas egy **n** természetes számot, meghatározza és kiírja a **tükörképét**. Mentsd a programot **A2** névvel a portfóliódba! *Példa*: ha **n = 1234**, a kiírt szám: **4321.**

Pszeudokód

```
beolvas n
tükör ← 0
amíg n≠0 végezd el
    számjegy ← n mod 10
    tükör ← tükör *10 + számjegy
    n ← n div 10
amíg vége
kiír tükörszám
```

Magyarázat

Az algoritmus végrehajtása lépésenként

n	n ≠ 0	számjegy	tükörszám
1234	igen	4	0 → 4
123	igen	3	4 → 43
12	igen	2	43 → 432
1	igen	1	432 → 4321
0	nem		
kiír 4321			

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,szamjegy,tukor=0;
  cin>>n;
  while(n!=0)
  { szamjegy=n%10;
    tukor=tukor*10+szamjegy;
    n=n/10;
  }
  cout<<tukor;
  return 0;
}
```

Python nyelv

```
n=int(input())
tukor=0
while n!=0:
    szamjegy=n%10
    tukor=tukor*10+szamjegy
    n=n//10
print(tukor)
```

Magyarázat

Ahhoz, hogy jobbról-balra, sorrendben hozzáférj az **n** számjegyeihez, ismeretlen lépésszámú, elöltesztelő ismétlő szerkezetet használhatsz. A **számjegy** változó tárolja az **n** utolsó számjegyét. A **tükör** változóba megépíted az **n** tükörképét. Az utolsó lépésben eltávolítod az **n** utolsó számjegyét.

Egy szám osztóinak feldolgozása

Az algoritmust akkor használod, ha egy **n természetes szám osztóit** szeretnéd meghatározni. **Feladat:** Írj programot, amely beolvas egy **n** természetes számot (**n>1**) és kiírja az **n** összes osztóját, növekvő sorrendben, szóközzel elválasztva. Mentsd a programot **A3** névvel a portfóliódba! Példa: ha **n = 18**, a kiírt értékek: **1 2 3 6 9 18**.

Pszeudokód

```
beolvas  n
minden i ← 1, n div 2 végezd el
    ha n mod i = 0 akkor
        kiír i, ' '
minden vége
kiír n
```

Ahhoz, hogy meghatározd az **n** összes lehetséges osztóját, ismert lépésszámú ismétlő szerkezetet használhatsz. Az **i** változó első értéke **1**, az **i** változó utolsó értéke **n div 2**. Az **n div 2+1** és **n-1** között, az **n**-nek nincs osztója. A program végén az **n** utolsó osztója kerül kiíratásra.

Magyarázat

Az algoritmus végrehajtása lépésenként

n	i	n mod i = 0	kiír
18			
	1	igen	1
	2	igen	2
	3	igen	3
	4	nem	
	5	nem	
	6	igen	6
	7	nem	
	8	nem	
	9	igen	9
			18

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{    unsigned int n,i;
     cin>>n;
     for(i=1;i<=n/2;i++)
          if(n%i==0)
               cout<<i<<' ';
     cout<<n;
     return 0;
}
```

Python nyelv

```
n=int(input())
for i in range(1,n//2+1):
    if n%i==0:
        print(i,end=' ')
print(n)
```

Magyarázat

Figyelem! Az **1** és **n** számokat **nem valódi osztók**nak nevezzük. A szám többi osztóját **valódi osztó**nak nevezzük. Példa: a **18** esetén az **1** és **18** számok nem valódi osztók, a **2, 3, 4, 6, 9** pedig valódi osztók.

Számláló algoritmus

Főként akkor használod az algoritmust, amikor meg szeretnéd **számolni**, hogy egy esemény hányszor fordul elő egy ismétlő szerkezet törzsében. **Feladat:** Írj programot, amely beolvas egy **n** természetes számot és kiírja az **n számjegyeinek a számát**. Mentsd a programot **A4** névvel a portfóliódba! Példa: ha **n = 42587**, a kiírt szám: **5**.

Magyarázat

Használhatod a **db** változót, amelynek a **számlálás** lesz a feladata. A program elején a számláló mindig a **0** kezdeti értéket kapja. Minden ismétlődésnél, amikor az ismétlő szerkezet törzsében található utasítások végrehajtódnak, a számláló értéke eggyel nő és az **n** utolsó számjegye eltávolítódik. A program végén kiíratod a számláló értékét.

Az algoritmus végrehajtása lépésenként

n	n≠0	db
42587	igen	0 → 1
4258	igen	1 → 2
425	igen	2 → 3
42	igen	3 → 4
4	igen	4 → 5
0	nem	
kiír 5		

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,db=0;
     cin>>n;
     do
     {    db++;
          n=n/10;
     }while(n!=0);
     cout<<db;
     return 0;
}
```

Figyelem! Az algoritmus hátultesztelő ismétlő szerkezetet használ. A **db** változó kezdeti értéke **0** lesz.

Python nyelv

```
n=int(input())
if n==0:
    db=1
else:
    db=0
while n!=0:
    db=db+1
    n=n//10
print(db)
```

Figyelem! Az algoritmus elöltesztelő ismétlő szerkezetet használ. Ha **n=0**, akkor **db** változó kezdeti értéke **1**, különben **0** lesz.

GYAKORLATI TEVÉKENYSÉG // A FIBONACCI - SZÁMSOR

Írj **C++/Python** programot, amely beolvassa az **n** természetes számot (**n>2**) és kiírja a számsor első **n** tagját. A számsorra vonatkozó szabály: az első két eleme **0** és **1**, és a **3.**-tól kezdődően minden további elem az **őt közvetlenül megelőző két szám összege**. Mentsd a programot **Fibonacci** névvel a portfóliódba! Példa: ha **n=20**, a kiírt számsor:

0 1 1 2 3 5 8 13 21 34 55 89 144 233 337 610 987 1597 2584 4181

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int i,n,a=0,b=1,c;
  cin>>n;
  cout<<a<<' '<<b<<' ';
  for(i=3;i<n+1;i++)
  { c=a+b;
    cout<<c<<' ';
    a=b;b=c;
  }
return 0;
}
```

Python nyelv

```
n=int(input())
a=0
b=1
print(a,b,end=' ')
for i in range (3,n+1):
    c=a+b
    print(c,end=' ')
    a=b
    b=c
```

Magyarázat

A számsor első két eleme **0** és **1**, a harmadik elem az első két elem összege: **0+1=1**. A negyediket, a harmadik és a második összegeként kapjuk: **1+1=2** és így tovább.

A **Fibonacci - számsor** valószínűleg a természetben előforduló leggyakoribb értéksor. Láthatjuk a méhek hatszögű viaszsejt építésében, a szitakötők szárnyain lévő mintázatban, a napraforgómag elrendezésében a virágsapkán stb. A számsorban, ha egy számot elosztunk az előzővel (**1/1**, **2/1**, **3/2**, **5/3**...), a **14.**-el kezdődően az **1.61803** értéket kapjuk, melyet **aranymetszésnek** vagy **aranyaránynak** nevezünk.

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int i,n,a=0,b=1,c;
  cin>>n;
  for(i=1;i<=n;i++)
  { c=a+b;
    cout<<(float)c/b<<' ';
    a=b;b=c;
  }
  return 0;
}
```

Feladat

Módosítsd a **Fibonacci** programot úgy, hogy olvassa be az n természetes számot (**n>14**), és írja ki a számsor azon első n tagját, amely kimutatja az **aranyarányt**. Mentsd a programot **Arany** névvel a portfóliódba!

Python nyelv

```
n=int(input())
a=0
b=1
for i in range(1,n+1):
    c=a+b
    print(c/b,end=' ')
    a=b
    b=c
```

TANULMÁNYOZD!!!!

A számsor a nevét az olasz matematikusról, **Leonardo Pisano Bogolloról (1170–1250) kapta**, más néven **Fibonacci**. A matematikus a meglepő számsort a következő kérdésből kiindulva fedezte fel: Hány pár nyúl származhat egy évben egyetlen pártól, ha minden pár havonta új párnak ad életet, és minden pár a következő hónaptól ivarérett, tenyészképes lesz?

JAVASOLT GYAKORLATOK

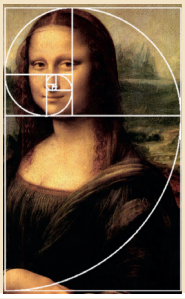
I. **Fedezd fel** a **Fibonacci - számsor** és az **aranyarány** jelenlétét:

http://y2u.be/xLaw05WmhBw

Természet

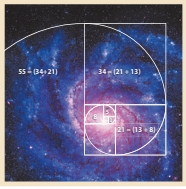
Művészet

Mona Lisa Leonardo da Vinci



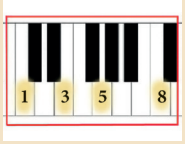
Világegyetem

Tejút



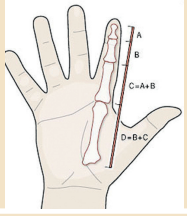
Zene

A hangjegyek dallama



Az emberi test felépítése

A csontok méreter



http://y2u.be/8KxduKwO7N8

Történelem

II. **Alkalmazd!** Legyen a mellékelt **Scratch** program:

1. Add meg az algoritmus végrehajtását követően megjelenített számsor értékeit, ha a bevitt értékek **5** és **3**.
2. Add meg, hogy milyen értéket vehetnek fel az **n** és **x** változók ahhoz, hogy a program a **2 4 8 16 32 64** számsort jelenítse meg.
3. Fogalmazd meg az adott programmal megoldott feladat kijelentését.
4. Módosítsd az adott programot úgy, hogy az új program az **x** első **n** nem nulla hatványait jelenítse meg.
   Példa: ha **n=5** és **x=3**, a kiírt értékek: **3 9 27 81 243.** Mentsd a **Scratch** programot **E1** névvel a portfóliódba!

III. **Gyakorolj!** Válassz egy fejlesztői környezetet és egy programozási nyelvet. Készíts programokat az alábbi feladatokhoz. Mentsd a programokat megfelelő nevekkel a portfóliódba!

1. Írj programot, amely beolvas egy **n** természetes számot, meghatározza és kiírja a **páratlan tükörképét**. Ha az **n**-nek nincs páratlan számjegye, a „Nem létezik" üzenet lesz kiírva. *Példa*: ha **n=1234**, a kiírt eredmény **31**.
2. Írj programot, amely beolvas egy **n** természetes számot és kiírja az **n páros számjegyeinek a számát** illetve a **páratlan számjegyeinek a számát**, szóközzel elválasztva. *Példa*: ha **n=2104639**, a kiírt számok: **4 3**.
3. Írj programot, amely beolvas egy **n** természetes számot (**n>1**) és kiírja az **n valódi osztóit, csökkenő sorrendben**, szóközzel elválasztva. *Példa*: ha **n=12**, a kiírt számok: **6 4 3 2**.
4. Írj programot, amely beolvas egy **n** természetes számot (**n>2**), majd az **a** és **b** két egész számot és kiírja a következő szabály szerint megépített számsor első **n** tagját: **a** és **b** a számsor első két tagja és a harmadik taggal kezdődően, bármelyik tag egyenlő az **őt közvetlenül megelőző két tag** összegével.

Példa: ha **n=6**, **a=2** és **b=7** a kiírt értékek: **2 7 9 16 25 41**.

JEGYEZD MEG!!

- Számok egyszerű felsorolását **számsor**nak tekintjük.
- A számsorok megépíthetők **algoritmikusan**, figyelembe véve bizonyos szabályokat.
- A **Fibonacci - számsor** egy olyan értéksor, amely nagyon gyakran előfordul a mindennapi életben, a természetben.
- Az **aranyarányt** alkalmazzák a művészetben, építészetben, matematikában, sőt a pénzügyi rendszerekben is.

A lecke végén sikeresen fogod alkalmazni azokat az alapvető algoritmusokat, amelyek segítségével feldolgozhatod az olyan számsorokat, amelyek egy szabály szerint voltak képezve. A számsorokat képező szabályokat felismerheted vagy megadhatod te magad. A szokványos napi teendőid felsorolása képzett számsort alkot. A lecke gyakorlati tevékenységeiből kiindulva kidolgozhatsz saját projekteket!

10. Lecke Képzett számsorok

A **képzett számsor**okat úgy kapjuk, hogy alkalmazunk egy számítási szabályt a számsor minden tagjára. A számsor egy tagja megadható: **a számsorban elfoglalt sorszáma függvényében, az őt megelőző egyes tagok segítségével, egy feltétel segítségével stb.**

EMLÉKEZZ!

A **Fibonacci - számsor** egy képzett számsor. A számsor első két eleme **0** és **1**. A további tagjait, a **3.**-tól kezdődően képezzük, minden tag az **őt közvetlenül megelőző kettő összege**. A lecke során megtanulsz olyan bonyolult számsorokat képezni, amelyek alkalmazhatnak **alapvető algoritmusokat.**

Összegszámolás

Az algoritmust akkor használod, amikor egy ismétlő szerkezet törzsében **összeget** szeretnél **számolni**.
Feladat: Írj programot, amely beolvas egy **n** nullától különböző természetes számot és kiírja az első sorba az első **n nullától különböző teljes négyzetet** tartalmazó számsor elemeit, szóközzel elválasztva, majd a második sorba a számsor **4**-es számjegyre végződő tagjainak az **összegét**. Mentsd a programot **A5** névvel a portfóliódba! *Példa*: ha **n=12**, az **1.** sorba írt számsor: **1 4 9 16 25 36 49 64 81 100 121 144** és a **2.** sorba írt szám: **212**. **A számsor tagjainak értékét csupán a számsorban elfoglalt sorszámuk alapján számold ki!**

Pszeudokód

```
beolvas n
s ← 0
minden i ← 1, n végezd el
    kiír i*i, ' '
    ha (i*i) mod 10=4 akkor s ← s + i*i
minden vége
kiír új sor, s
```

Python nyelv

```
n=int(input())
s=0
for i in range(1,n+1):
    print(i*i,end=' ')
    if (i*i)%10==4:
        s=s+i*i
print()
print(s)
```

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{unsigned int i,n,s=0;
 cin>>n;
 for(i=1;i<=n;i++)
  {cout<<i*i<<' ';
   if((i*i)%10==4)
        s=s+i*i;
  }
 cout<<endl<<s;
 return 0;
}
```

Magyarázat

Az összeg kiszámításához használhatod az **s** változót, amely kezdeti értékét **0**-ra állítod (**s ← 0**). Minden ismétlődésnél, amikor az ismétlő szerkezet törzsében található utasítások végrehajtódnak, ha az összeg **i.** tagja (**i*i**) a **4**-es számjegyre végződik, akkor hozzáadod az összeghez (**s ← s + i*i**). Az ismétlő szerkezet végrehajtása után kiíratod az összeget.

Szorzatszámolás

Az algoritmust akkor használod, amikor egy ismétlő szerkezet törzsében szorzatot szeretnél **számolni**.
Feladat: Írj programot, amely beolvassa az **n, x** és **y** nullától különböző természetes számokat és kiírja az első sorba az **x, x+y, x+2y, x+3y,** … számsor első **n** elemét, szóközzel elválasztva, majd a második sorba a képzett számsor tagjai **szorzatának** az utolsó számjegyét. Mentsd a programot **A6** névvel a portfóliódba! *Példa*: ha **n=6, x=2** és **y=5**, az **1.** sorba írt számsor: **2 7 12 17 22 27** és a **2.** sorba írt számjegy: **4. A számsor tagjainak értékét az őt közvetlenül megelőző tagok segítségével számold ki!**

Pszeudokód

```
beolvas n, x, y
p ← 1
minden i ← 1, n végezd el
   kiír x
   p ← (p*x) mod 10, x ← x + y
minden vége
kiír új sor, p
```

Python nyelv

```
n=int(input())
x,y=int(input()),int(input())
p=1
for i in range(1,n+1):
   print(x,end=' ')
   p=(p*x)%10
   x=x+y
print(),print(p)
```

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{unsigned int i,n,x,y,p;
 p=1;cin>>n>>x>>y;
 for(i=1;i<=n;i++)
  {cout<<x<<' ';
   p=(p*x)%10;x=x+y;
  }
 cout<<endl<<p;
 return 0;
}
```

Magyarázat

A szorzat kiszámításához használhatod a **p** változót, melynek kezdeti értékét **1**-re állítod (**p ← 1**). Minden ismétlődésnél, amikor az ismétlő szerkezet törzsében található utasítások végrehajtódnak, módosítod a **p** értékét (**p ← (p*x) mod 10**). Az ismétlő szerkezet végrehajtása után kiíratod a szorzatot.

Adott szabály szerint képzett számsorok feldolgozása

Feladat: Írj programot, amely beolvassa az **n** és **k** nullától különböző természetes számokat és kiírja azt az első **n** számot, szóközzel elválasztva, amely **számjegyeinek összege osztható k-val**. Mentsd a programot **A7** névvel a portfóliódba! Példa: ha **n=7** és **k=8**, a kiírt számsor: **8 17 26 35 44 53 62**. **A számsor tagjait egy adott feltétel segítségével határozd meg!**

Pszeudokód

```
beolvas n, k
x ← 1, nr ← 0
amíg n ≠ nr végezd el
   cx ← x, s ←0
   amíg cx ≠ 0 végezd el
      s ← s + cx mod 10; cx ← cx div 10
   amíg vége
   ha s mod k=0 akkor nr ← nr + 1; kiír x,' '
   ha vége
   x ← x + 1
amíg vége
```

Magyarázat

A számsor megépítésénél használod az **x** változót (a kezdeti értéke **1**). Amíg a teljes számsor nem készült el, az **x** változtatja az értékét (**x ← x + 1**). Egy ismétlő szerkezettel kiszámolod az **x** számjegyeinek az összegét. Ha a számjegyek összege a **k** többszöröse, akkor számlálod és kiíratod az **x** számot, majd módosítod az **x** értékét a következő ellenőrzéshez. A kilépés az ismétlő szerkezetből az **n** darab szám kiíratása után történik meg.

GYAKORLATI TEVÉKENYSÉG /// A PIRAMIS SZÁMSOR

A **piramis számsor** természetes számokból alkotott csoportokból épül fel, a következő eloszlás szerint: az első csoport egy számot tartalmaz, a második csoport két számot tartalmaz, a harmadik csoport három számot tartalmaz és így tovább, minden csoport ugyanazzal a tulajdonsággal rendelkezik.

A **piramis sorozat** a piramis számsor ábrázolása, az alábbiak szerint felépítve:

A minta: az első sor tartalmazza az első csoportba tartozó számokat, a második sor tartalmazza a második csoportba tartozó számokat és így tovább.			**B minta**: az első sor tartalmazza az utolsó csoportba tartozó számokat, a második sor tartalmazza az utolsó előtti csoportba tartozó számokat és így tovább.		
Példák n = 4 esetén épített piramis sorozatokra:					
1 1 1 1 1 1 1 1 1 1	1 2 2 3 3 3 4 4 4 4	1 1 2 1 2 3 1 2 3 4	1 1 1 1 2 2 2 3 3 4	4 3 2 1 3 2 1 2 1 1	1 2 3 4 2 3 4 3 4 4
S1 sorozat	**S2 sorozat**	**S3 sorozat**	**S4 sorozat**	**S5 sorozat**	**S6 sorozat**

Feladat

Írj **C++/Python** programot, amely beolvassa az **n** nullától különböző természetes számot, és kiír két piramis sorozatot a fent bemutatott **S1** és **S6** minta szerint. Mentsd a programot **Sorozatok** névvel a portfóliódba!

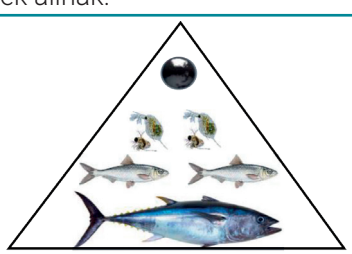
Python nyelv

```
n=int(input())
for i in range(1,n+1):
    for j in range(1,i+1):
        print(1,end=' ')
    print()
for i in range(1,n+1):
    for j in range(1,n-i+2):
        print(i+j-1,end=' ')
    print()
```

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,i,j;
  cin>>n;
  for(i=1;i<=n;i++)
  { for(j=1;j<=i;j++)
      cout<<1<<' ';
    cout<<endl;  }
  for(i=1;i<=n;i++)
  { for(j=1;j<=n-i+1;j++)
      cout<<i+j-1<<' ';
    cout<<endl;  }
  return 0;
}
```

TANULMÁNYOZD!!

Táplálékpiramis		**Hulladékpiramis**	
A környezetben a táplálékpiramis tetején található organizmusok száma a legalacsonyabb.		A környezetben a hulladékpiramis tetején a mérgező elemek állnak.	
	Tápláléklánc: **Sólyom** **Rigó** **Csiga** **Lóhere**		**Hulladéklánc:** **Higany** **Plankton** **Makréla** **Tonhal**

JAVASOLT GYAKORLATOK

I. **Tanulmányozd**, hogy tudományos igazságokról vagy csak véletlen egybeesésekről van szó:

A **gízai nagy piramis** koordinátái megegyeznek a fény terjedési sebességével!
Koordináta:
é. sz. 29,9792458 fok
Fénysebesség:
299.792.458 m/s

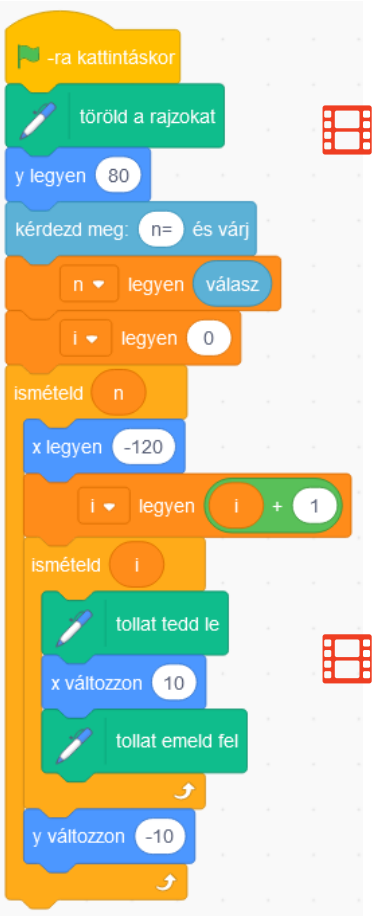
A **Kheopsz - piramis** csúcsán áthaladó hosszúsági kör a föld szárazföldjeit 2 egyenlő részre osztja.
A **Föld tömege** körülbelül 5.300.000.000.000 milliárd tonna, a **Nagy Piramis** tömege pedig körülbelül 5.300.000 tonna.

http://y2u.be/86FyWTKzxpI

Kheopsz nagy piramisa egyike az **ókori világ hét csodájának**.

II. **Dolgozz csapatban!**

1. Kövesd az egyik diáktársaddal a mellékelt program működését
http://bit.ly/39iM5gm
2. Készítsd el a mellékelt **Scratch** programot! Ahhoz, hogy rajzolhass a szereplőkkel, válaszd a *Bővítmény hozzáadása* lehetőséget, majd válaszd a *Toll* bővítményt. A *Kód* lapfülen megjelenik a *Toll* blokk. Válassz szereplőt! Keresd ki a *ceruzát (Pencil)* és rendeld hozzá a mellékelt programban látható feladatokat! Futtasd, majd mentsd a programot **Sorozat** névvel a portfóliódba!
3. Módosítsd az adott programot úgy, hogy a *ceruza* (*Pencil*) egy piramis számsort jelenítsen meg a *Játéktéren* a **B minta** szerint!



III. **Gyakorolj! Válassz egy fejlesztői környezetet és egy programozási nyelvet.**

1. Írj programot, amely beolvassa az **n** nullától különböző természetes számot és kiírja az első sorba a **nullától különböző páros számok**ból alkotott számsor első **n** elemét szóközzel elválasztva, majd a második sorba a számsor **6**-tal osztható tagjainak az **összegét**. *A számsor tagjainak értékét csupán a számsorban elfoglalt sorszámuk alapján számold ki!* Mentsd a programot **Tagok** névvel a portfóliódba! Példa: ha **n=6**, az **1.** sorba írt számsor: **2 4 6 8 10 12** és a **2.** sorba írt szám: **18**.
2. Írj programot, amely beolvassa az **n** nullától különböző természetes számot és kiírja a gyakorlati tevékenységeknél – A PIRAMIS SZÁMSOR bemutatott **S2**, **S3**, **S4** és **S5** mintáknak megfelelő piramis sorozatokat. Mentsd a programot **Számsorok** névvel a portfóliódba!

JEGYEZD MEG!

- Ha egy piramis számsorban minden csoport elemeinek a száma egyenlő a csoport számával, akkor **négyzetes számsort** kapunk.
- Egy ilyen számsort egy **négyzetes sorozattal** lehet ábrázolni.

1 1 1 1 1 2 3 4 4 4 4 4
1 1 1 1 1 2 3 4 3 3 3 3
1 1 1 1 1 2 3 4 2 2 2 2
1 1 1 1 1 2 3 4 1 1 1 1

A lecke során vissza fogsz emlékezni az alapvető algoritmusokra, amelyek segítségével olyan programokat írhatsz, amelyek beolvasott számsorokkal dolgoznak. Különböző programozási nyelveken, különféle fejlesztői környezetekben futtatható programokat fogsz írni. Ezek olyan számsorokat dolgoznak fel, amelyek elemeit te magad adhatod meg. Ezeket a programokat a javasolt tevékenységek alapos tanulmányozásával gyakorolhatod, míg a megfelelő algoritmusokat új helyzetekben alkalmazhatod.

11. Lecke Beolvasott számsorok

A **beolvasott számsorok**at ismétlő szerkezettel lehet feldolgozni és lehetnek **ismert** elemszámú vagy **ismeretlen** elemszámú számsorok.

A **beolvasott számsorok** ismétlő szerkezettel történő **feldolgozásához** ismerned kell vagy az értékek számát, vagy azt az értéket, amely a számsor végét jelöli, vagy a feladatban meghatározott bizonyos feltételt teljesülését, stb.

Legkisebb/legnagyobb érték meghatározása

Az algoritmust akkor használod, amikor egy ismétlő szerkezet törzsében **legkisebb/legnagyobb** értéket szeretnél **meghatározni**. **Feladat**: Írj programot, amely beolvassa az **n** természetes számot, majd **n** darab egész számot, és kiírja a **legkisebb számot a számsorból**. Mentsd a programot **A8** névvel a portfóliódba! *Példa*: ha **n=5** és a számsor: **8 4 2 9 1**, a kiírt szám: **1.**

Pszeudokód

```
beolvas n, Min
minden i ← 2, n végezd el
   beolvas x
   ha x<Min akkor Min ← x
minden vége
kiír Min
```

Magyarázat

Javasolt, hogy a **Min** változó kezdeti értéke a **számsor első eleme** legyen.

Az ismétlő szerkezet törzsében beolvasod az **n-1** darab értéket, és frissíted a **Min** változó értékét, valahányszor szükséges.

Az algoritmus végrehajtása lépésenként

i	x	x<Min	Min
2	4	igen	8 → 4
3	2	igen	4 → 2
4	9	nem	2
5	1	igen	2 → 1
kiír 1			

Ismert elemszámú számsorok feldolgozása

Feladat: Írj programot, amely beolvassa az **n** nullától különböző természetes számot, majd **n** darab természetes számot, és kiírja, hogy a számsor elemei közül **hány szám a 2 hatványa**. Mentsd a programot **A9** névvel a portfóliódba! *Példa*: ha **n=6** és a számsor: **8 1 6 4 24 64**, a kiírt szám: **4**.

```
beolvas n // a számsor elemeinek a száma
minden i←1, n végezd el
   beolvas x  // a számsor eleme
   ...
minden vége
```

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,i,x,db=0;
    cin>>n;
    for(i=1;i<=n;i++)
    {  cin>>x;
       while(x%2==0)x=x/2;
       if(x==1)db++;
    }
    cout<<db;
    return 0;
}
```

Python nyelv

```
n=int(input())
db=0
for i in range(1,n+1):
    x=int(input())
    while x%2==0:
        x=x//2
    if x==1:
        db=db+1
print(db)
```

Magyarázat

A feladatban megadott tulajdonság ellenőrzéséhez megszámlálhatod, hogy hány olyan érték van a számsorban, amely a **2** hatványa.

Ismeretlen elemszámú számsorok feldolgozása

Mivel nem ismered a számsor elemeinek a számát, használhatod az **amíg … végezd el** elöltesztelő ismétlő szerkezetet. A számsor beolvasásának végét leggyakrabban egy olyan érték jelöli, amely nem tartozik a számsor elemei közé.

w = a számsor végét jelölő érték (nem tartozik a számsor elemei közé)
x = a számsor egy értéke

```
beolvas  w, x
amíg  x≠w végezd el
    …
    beolvas  x
amíg vége
```

Feladat: Írj programot, amely nullától különböző természetes számokat olvas be a **0 érkezéséig** (a **0** nem eleme a számsornak), és kiírja a számsor elemei közül azt a **legnagyobb** számot, amelynek **minden számjegye páros**. Ha nem létezik olyan szám, amelynek minden számjegye páros, akkor a „Nem létezik” üzenetet írja ki. Mentsd a programot **A10** névvel a portfóliódba! *Példa*: legyen a számsor: **94 24 84 68 0**, a kiírt eredmény: **84**.

C++ nyelv

```
#include <iostream>
using namespace std;
int main()
{ unsigned int x,cx,db,Max=1;
  cin>>x;
  while(x!=0)
    {db=0;cx=x;
     while(cx!=0)
      {if(cx%2==1)db++;cx=cx/10;}
     if(db==0 && x>Max)Max=x;
     cin>>x;
     }
  if(Max==1) cout<<"Nem letezik";
  else cout<<Max;
    return 0;
}
```

Python nyelv

```
x=int(input())
Max=1
while x!=0:
    db=0;cx=x;
    while cx!=0:
        if cx%2==1:
            db=db+1
        cx=cx//10
    if db==0 and x>Max:
        Max=x;
    x=int(input())
if Max==1:
    print('Nem letezik')
else:
    print(Max)
```

Magyarázat

Ha létezik legalább egy olyan szám, amelynek minden számjegye páros, akkor **Max ≠ 1**.

Az lnko meghatározása

Az algoritmust akkor használod, amikor két szám **l**eg**n**agyobb **k**özös **o**sztóját szeretnéd meghatározni.
Feladat: Írj programot, amely beolvassa az **n** természetes számot (**n>1**), majd **n** darab nullától különböző természetes számot, és kiírja a számsor elemeinek legnagyobb közös osztóját. Mentsd a programot **A11** névvel a portfóliódba! *Példa*: ha **n=5** és a számsor: **36 12 20 8 52**, a kiírt szám: **4**.

Pszeudokód

```
beolvas n, d
minden i ← 2, n végezd el
 beolvas x
  amíg x ≠ d végezd el
   ha x>d akkor x ← x - d
 különben d ← d – x
 amíg vége
minden vége
kiír d
```

Magyarázat

A **minden … végezd el** ismétlő szerkezet előtt, a **d** változóba beolvasod a számsor első elemét.
A **lnko(d,x)** meghatározásánál használhatod az ismételt kivonást: a nagyobb számból kivonod a kisebbet, amíg különböznek. Végül mindkét szám azonos lesz a lnko-val. A **minden … végezd el** ismétlő szerkezet törzsében frissíted a **d** értékét.

Az algoritmus végrehajtása lépésenként

i	x	d
2	12	36 → 12
3	20	12 → 4
4	8	4
5	52	4
kiír 4		

GYAKORLATI TEVÉKENYSÉG /// A DONGÓ SZÁMSOR

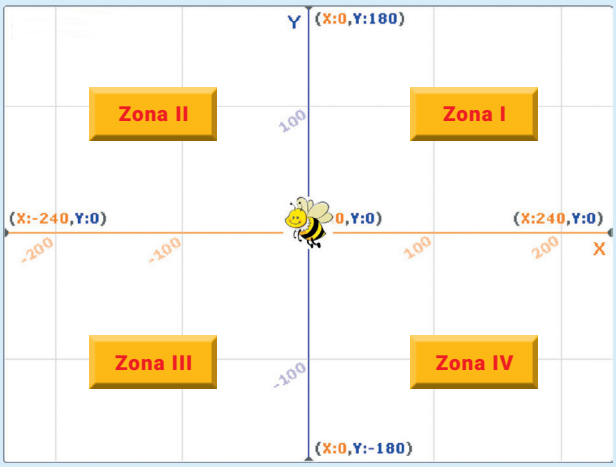
A darázs repkedése a virágok között kaotikusnak tűnik, de a londoni Queen Mary Egyetem kutatói arra a következtetésre jutottak, hogy tulajdonképpen egy koreográfia szerint történik. A kutatók által vizsgált dongók a térmemóriájukat használták és rövid idő alatt sikerült megtalálniuk a legrövidebb útvonalat vissza a kaptárba, miközben minden virágnál csak egyszer haladtak el.

A játéktér koordináta rendszere

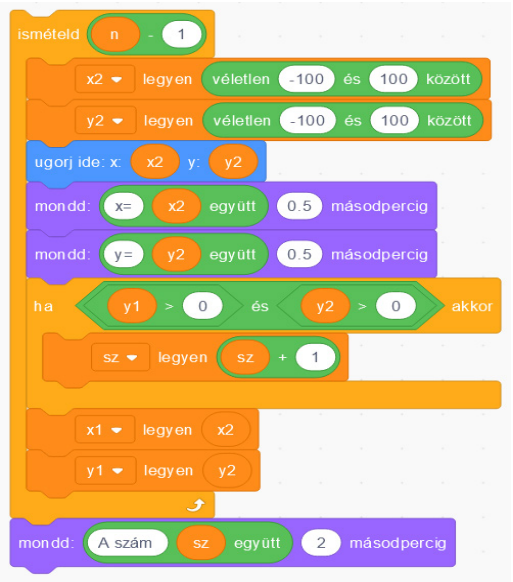
A **Scratch**-ban dolgozva észrevetted, hogy a *Játéktér* az **x tengely** és az **y tengely** segítségével négy téglalap alakú részre, **zónákra** oszlik. Az (**x, y**) értékpárt, amely megadja egy pont elhelyezkedését a színpadon, **koordinátának** nevezik.

Helyezz el a színpadon, az *XY tengelyen* egy dongót ábrázoló szereplőt, és méretezd megfelelően. Nézd a mellékelt mintát!

A Dongó **n** darab egymást követő megállást fog végezni, az **n** értékét te magad választod. Minden megálláskor a Dongó közli veled a koordinátáit. Az **x** és **y** legnagyobb értéke **100**, a legkisebb értéke **-100**.



Tanulmányozd figyelmesen az alábbi programot, hogy megtudd, hány egymást követő megállást végzett a Dongó a **Zona I** és a **Zona II** részen. Mentsd a programot **Dongó** névvel a portfóliódba!



Észrevetted, hogy a programban két koordinátapár jelenik meg: (**x1, y1**) és (**x2, y2**). Az első megálló koordinátái az ismétlő szerkezeten kívül vannak megválasztva és megjelenítve.

Az (**x2, y2**) koordináta az ismétlő szerkezet törzsében van megválasztva és megjelenítve, és itt kerül ellenőrzésre az, hogy az egymást követő (**x1, y1**) és (**x2, y2**) koordináták a **Zona I** illetve **Zona II** területen vannak. Ezt követően módosul a (**x1, y1**) koordináta értéke, felveszi az (**x2, y2**) koordináta értékét.

A program végén azoknak az egymást követő pároknak a száma lesz kiírva, amelyek tiszteletben tartják az adott tulajdonságot.

JAVASOLT GYAKORLATOK

I. **Kutass** a mindennapi életben fellelhető számsorok után:

A darazsak zümmögése a gyors, körülbelül **200 csapás/másodperc**es szárnycsapások eredménye, és ez percenként **12 000 szárnycsapás**t eredményez. A darazsak naponta háromszor több virágot látogatnak meg, mint a méhek.

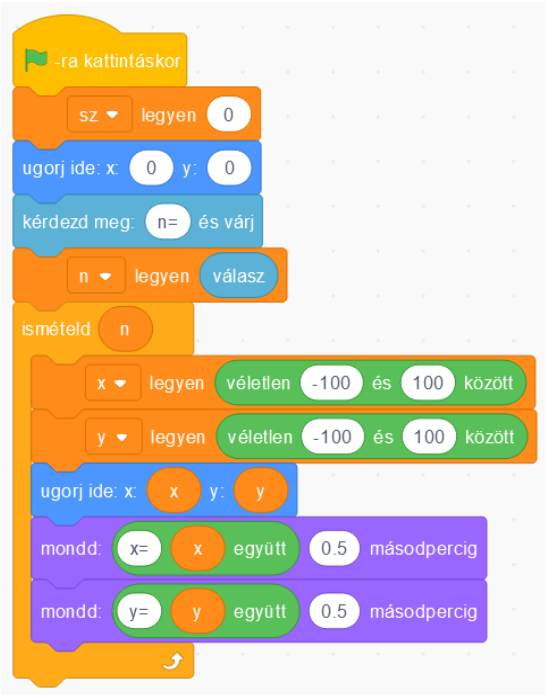
A legnagyobb sebesség, amellyel egy darázs képes repülni megközelítőleg **70 km/h**. Egyetlen oda-vissza úton a kaptárig akár száz virágot is érint.

A leggyorsabb kezű hegedűművész **A dongó** című zenemű bemutatásával került be a Rekordok Könyvébe

http://y2u.be/QDchBmUSDfw

II. **Alkalmazd!** Tekintsük a következő **Scratch** programot, amelyben az **x** és **y** a gyakorlati tevékenységnél bemutatott koordináták.

1. Futtasd, majd mentsd a programot **Koordináta** névvel a portfóliódba!
2. Egészítsd ki az adott programot úgy, hogy a végén kiírja az **y** koordináta **legnagyobb** értékét és azoknak az **x** koordinátáknak az **összegét**, amelyeknél a Dongó megállt a **Zona I** és **Zona IV** részen. Ha a Dongó nem állt meg ezeken a területeken, akkor a „Nem történt megállás” üzenet kerüljön kiíratásra.



III. **Gyakorolj!** Írj programot, amely beolvassa az **n** nullától különböző természetes számot, majd **n** darab természetes számot, és kiírja a számsorban előforduló **legnagyobb páratlan számot** és az **előfordulásainak a számát**, szóközzel elválasztva. Ha nem létezik páratlan szám a számsorban, akkor a „Nem létezik” üzenetet írja ki. Mentsd a programot **Páratlanok** névvel a portfóliódba!
Példa: ha **n=7** és a számsor: **5 9 4 9 9 3 2**, a kiírt eredmény: **9 3**.

JEGYEZD MEG!

Ha a számsor értékeinek feldolgozása során előfordul, hogy nincsenek olyan értékek, amelyek megfelelnek a feladat követelményeinek, akkor ajánlott ezt az esetet a következő algoritmusok valamelyikével kezelni:

- **A számsor feldolgozása során számláld meg, hogy hány érték felel meg a feladat követelményének. Ha a feldolgozás végén a számláló 0 maradt, akkor írass ki egy megfelelő üzenetet.**
- **Használj egy jelzőlámpa típusú változót. Ez a változó két érték tárolására alkalmas: x vagy y. Kezdetben rendeld a változóhoz az x értéket. A programban azonosíts minden olyan esetet, amikor a jelzőlámpa változó értéke y-ra változhat. Miután a kezdeti érték y értékre változott, többé nem feleltetheted meg a jelzőlámpa változónak az x értéket! Kezeld megfelelően a jelzőlámpa értékét.**

A lecke során olyan algoritmusokkal fogsz dolgozni, amelyek képesek összekapcsolni több tudományterületet a Scratch program-blokkok segítségével végzett programozást használva. A kreativitásod segítségével képes leszel elemezni az adatfeldolgozás példáit a különböző iskolai tantárgyaknál előforduló helyzetekben, és a lecke végén biztosan sikeresen fogod alkalmazni az alapvető algoritmusokat!

12. Lecke Interdiszciplináris algoritmusok

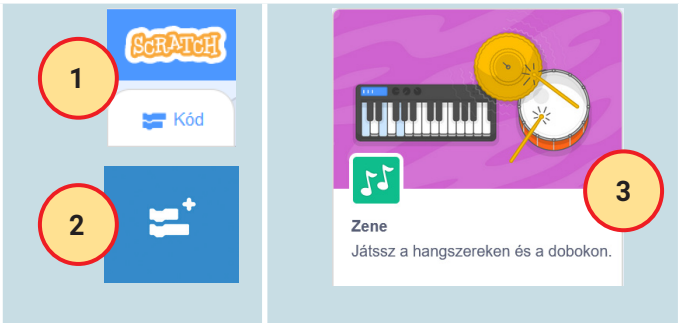
Az **interdiszciplinaritás** magában foglalja az információk több területről történő összegyűjtését, és a két vagy több tantárgynál tanulmányozott fogalmak beleépítését egyetlen tevékenységbe.
Az **interdiszciplináris** algoritmusok több tantárgyból tanult ismereteket használnak fel. Alkalmazásuk összetett interdiszciplináris feladatok megoldását eredményezi.

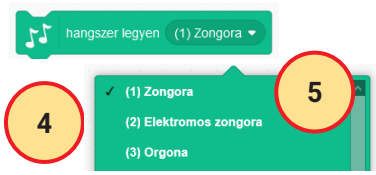
TANULMÁNYOZD!!

A **pentatónia** (a görög pente - "öt", tonos - "hang") a népzenéből származó zenei szerkezet, amely minden nép hagyományos zenéjében megtalálható. A pentatónia elsődleges helyet foglal el az ázsiai népzenében, amely különös jelentőséget tulajdonít a mindössze öt hangból álló dallamoknak. A kínai emberek alapozták meg ezt a zeneelméletet, az öt zenei hang és az univerzumot alkotó öt alapelem – víz, tűz, fa, fém, föld – közötti jelentős hasonlóságból kiindulva. **Zeneórán** te is dolgoztál a pentaton zenei skálával.

A **Scratch** alkalmazás segítségével, készíthetsz egy **interdiszciplináris projektet**, amellyel komponálhatsz és lejátszhatsz pentaton dallamokat. Sőt mi több, az osztálytársaiddal közösen egy zenekart is alkothattok!

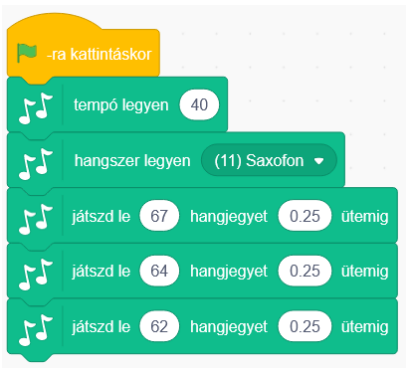
A *Kód* (**1**) lapfülön válaszd a *Bővítmény hozzáadása* (**2**) lehetőséget és keresd ki a *Zene* (**3**) bővítményt. A *Parancskészletben* (**4**) megjelennek az új bővítmény sajátos grafikus parancsblokkjai és a különféle hangszerek (**5**).





A **hangjegyek** azt jelentik a zenében, amit a betűk a beszédben. A **ritmus** a hangok harmonikus szerveződése. Az **ütés** egy hang értékének a hossza.
A **Scratch** alkalmazás *Zene* bővítményében a pentaton hangjegyek számjegyekkel vannak kódolva az alábbiak szerint:

DO	RE	MI	FA	SOL
60	62	64	67	69

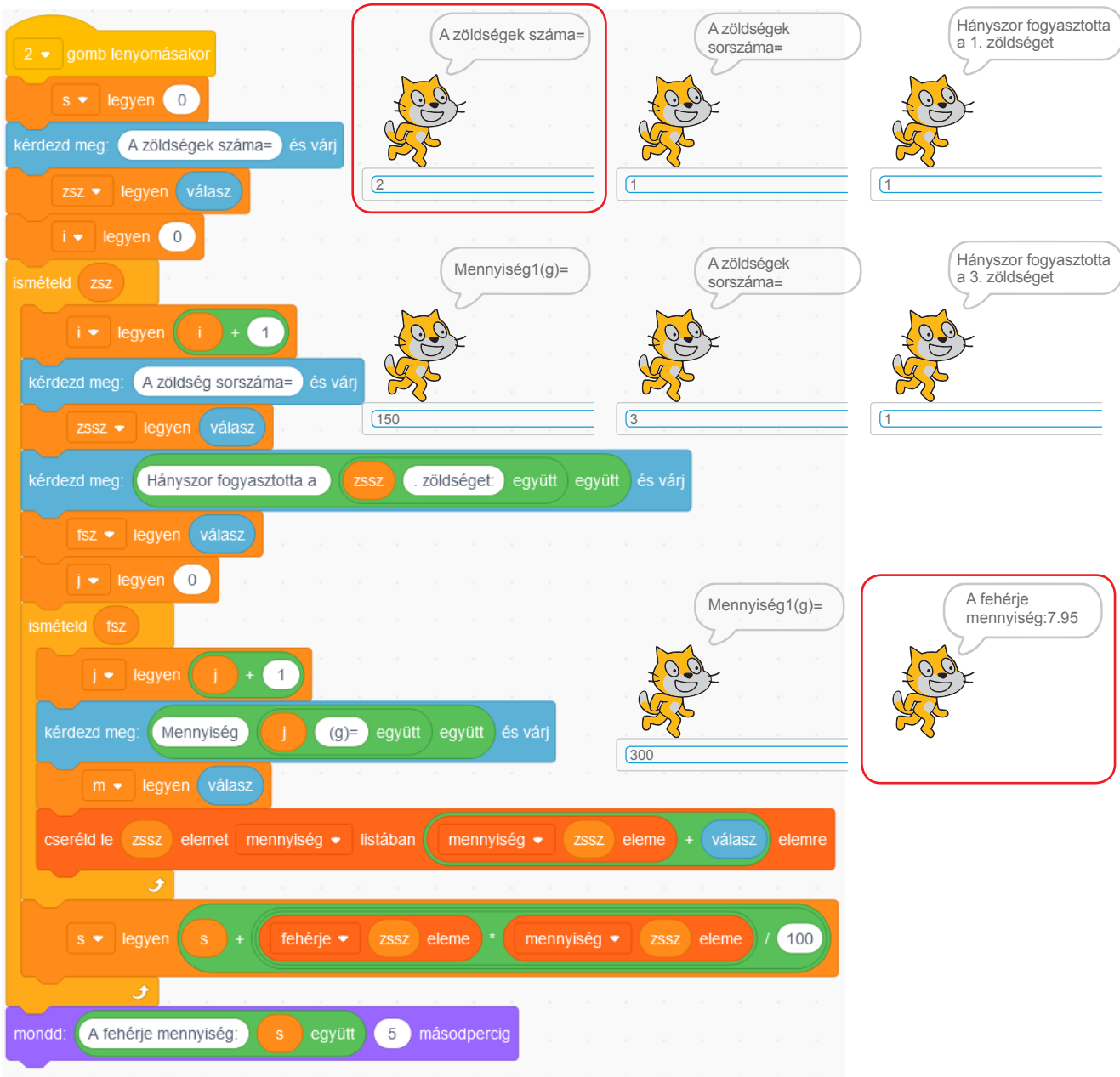


Készíthetsz egy **Scratch** projektet, Edvard Grieg zeneszerző Morning Mood című pentaton dallamának kódolását használva. **Kellemes zenehallgatást!**

Hangjegy	67	64	62	60	62	64	67	64
Ütés	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Hangjegy	62	60	62	64	62	64	67	64
Ütés	0.25	0.25	0.125	0.125	0.125	0.125	0.25	0.25
Hangjegy	67	69	64	69	67	64	62	60
Ütés	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

GYAKORLATI TEVÉKENYSÉG // AZ EGÉSZSÉGES TÁPLÁLKOZÁS ALGORITMUSA

A **biológia** és a **technológia** órákon jobban megértetted, hogy mennyire fontos az egészséges és kiegyensúlyozott táplálkozás. Úgyhogy készíthetsz egy **Scratch** projektet, amelyben kiszámolod, hogy **mennyi fehérjét tartalmaz az egy személy által, egy hét alatt elfogyasztott friss zöldség mennyiség**. A projekt elkészítéséhez ismerned kell a fogyasztott zöldségek megnevezését, minden fogyasztás mennyiségét, grammban kifejezve. Ugyanakkor azt is fontos ismerni, hogy egy-egy zöldségfélét hányszor fogyasztott a hét nap alatt. Mentsd a projektet **Zöldség** névvel a portfóliódba!



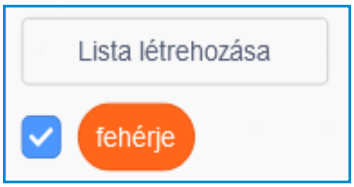
1. lépés

A fogyasztható zöldségek nevét és minden zöldség esetén a 100 g zöldségben található fehérje mennyiséget beírhatod egy táblázatba egy táblázatkezelő alkalmazást használva. A **http://bit.ly/2OFOSsi** linkre kattintva, letölthetsz egy adattáblát. Exportáld az állományt **zöldségek.csv** névvel a portfóliódba!

	A	B
1	Piros kaliforniai paprika	1.3
2	Piros kaliforniai paprika	1.1

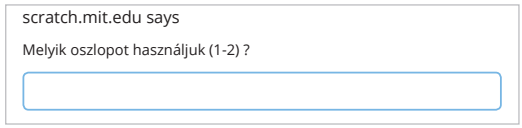
2. lépés

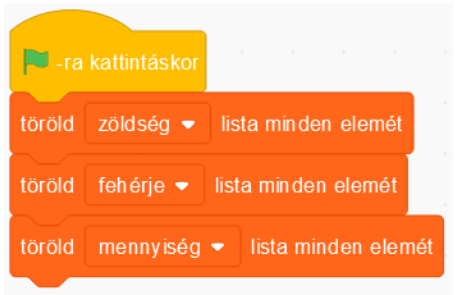
A **Scratch** alkalmazás *Változók* parancskészletéből válaszd a *Lista létrehozása* lehetőséget, és készíts három listát: **zöldségek, fehérjék** és **mennyiség**. Ha szükséges, törölheted a meglévő adatokat a listákból a *töröld* ***listanév*** *lista minden elemét* parancsblokkal.

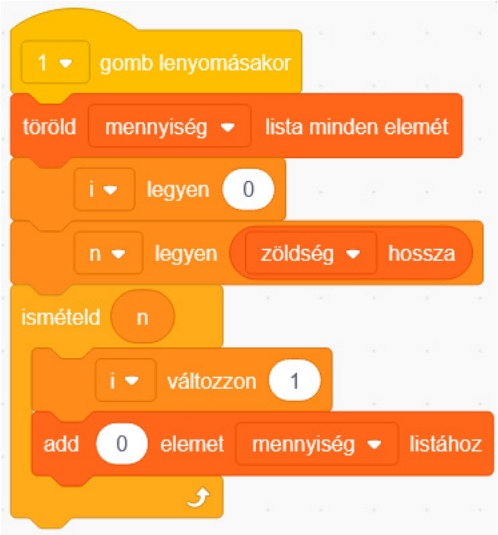


3. lépés

Importáld az adatokat a **zöldségek.csv** állományból a **zöldségek** és **fehérjék** listákba, felhasználva a táblázat első illetve második oszlopát. Kattints jobb gombbal az egyes listákra és a megjelenő menüből válaszd az *importálás* lehetőséget.







Figyelem! A fehérje listában a tizedes elválasztó jel a pont kell legyen. A harmadik lista, az *1-es billentyű leütésekor* csupa **0** értékkel lesz feltöltve. Mindhárom listának azonos számú eleme lesz. A listák elemeinek számát az **n** változó tárolja. A *Játéktéren* csak a 3 lista átméretezése és megjelenítése történik.

4. lépés

Az algoritmus végrehajtása a *2-es billentyű lenyomásakor* történik. Az **zsz** változóba a fogyasztott zöldségek számát fogod megadni. Minden fogyasztott zöldség esetén az **zssz** változóba megadod a zöldség sorszámát jelölölő számot a zöldségek listájáról. Minden zöldségfogyasztás esetén, az **m** változóba az illető zöldségből elfogyasztott mennyiséget adod meg, grammban kifejezve. Az **s** változóhoz, minden fogyasztás esetén hozzáadod azt az értéket, amely a szóban forgó zöldség fehérje tartalmát jelöli, grammban kifejezve.

Figyelem! Az **i** és **j** változók a **minden ... végezd el** ismétlő szerkezetek vezérlői.

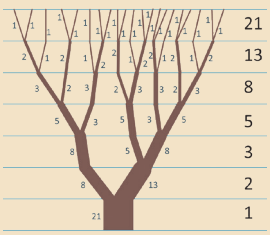
Gratulálok! Sikerült lefuttatni az algoritmust, fogyassz továbbra is sok zöldséget!

JAVASOLT GYAKORLATOK

I. **Kutasd** a **Fibonacci - számsor** létezését a mindennapi életben:

Csigahéj

Faágak



Fenyőtoboz

Napraforgó

Levelek

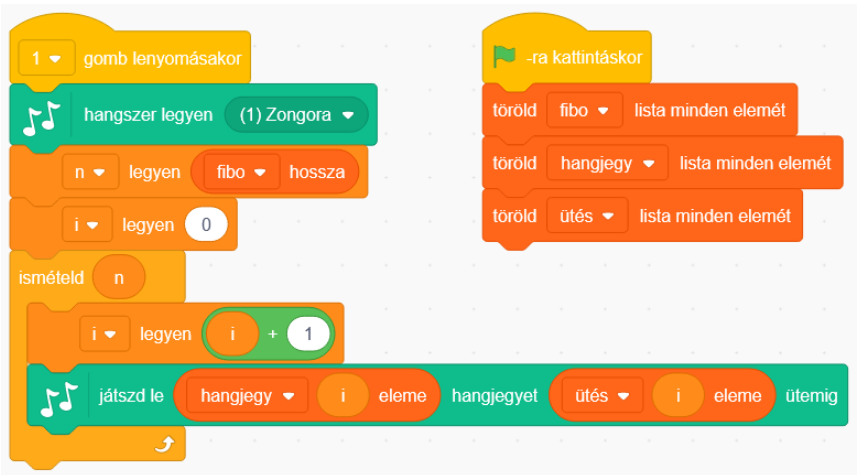
Pávafarok

II. **Alkalmazd!** A **Fibonacci** - számsor első **14** tagja: **0 1 1 2 3 5 8 13 21 34 55 89 144 233**. Írd egy **Excel** munkalap *első oszlopába*, az előfordulás sorrendjében, a fenti számsor elemeit alkotó összes **számjegyet**. Ez **23** számjegyet jelent. A *második oszlopba* írj egy-egy számot, amely az első oszlopban található számjegy **kódja**, az alábbi táblázat szerint:

Számjegyek	0	1	2	3	4	5	6	7	8	9
C-dúr	SI	DO	RE	MI	FA	SOL	LA	SI	DO^2	RE^2
Kódolás	59	60	62	64	65	67	69	71	72	74

A harmadik oszlopba írd a **0.125** értéket, kivéve a **3, 6, 7, 14, 16, 20, 23** sorokat, ahová a **0.25** értéket kell beírni. Exportáld a táblázatot **fibonacci.csv** névvel a portfóliódba!

III. **Gyakorolj! Készíts programot, amely felhasználja az adatsort ahhoz, hogy lejátsza zenére a Fibonacci - számsort!**
A **Scratch** alkalmazás *Változók* parancskészletéből válaszd a *Lista létrehozása* lehetőséget, és készíts három listát: **fibo**, **hangjegy** és **ütés**. Importáld az adatokat a **fibonacci.csv** állományból. Az állományt letöltheted a **http://bit.ly/2UGR9au** linkre kattintva. Importáld az állományt a **Scratch** alkalmazásba: a **fibo** listának az **első oszlop**, a **hangjegy** listának a **második oszlop** és az **ütés** listának a **harmadik oszlop** adatai felelnek meg.
Mentsd a programot **Fibonacci** névvel a portfóliódba!



JEGYEZD MEG!

- Készíthetsz **interdiszciplináris** algoritmusokat, ötvözve a különféle tantárgyakból tanult ismereteket.
- Joseph Schillinger egyes műveiben felhasználva a **Fibonacci-számsor** ihlette ábrázolást, a komponáláshoz kidolgozott egy matematikai stratégiát, tiszteletben tartva a természet elemeinek harmóniáját, összhangban a Fibonacci-számsor elemeivel.

ISMÉTLÉS /// CSAPATMUNKA!

Gratulálunk! Megtanultál bonyolult algoritmusokat készíteni egy vagy több alapvető algoritmus felhasználásával. Most jó alkalom nyílik arra, hogy az osztálytársaiddal együtt kreatívan alkalmazd is őket új és érdekes összefüggésekben. Dolgozzatok csapatban! Minden csapat készít egy **C++/Python** programot, amely megoldja az alábbi feladatok valamelyikét. Tulajdonképpen a feladatok mindenike kihívást jelent a számsorokra alkalmazott alapvető algoritmusok számára.

Képzett számsor	Beolvasott számsor
1. Feladat: Írj programot, amely beolvassa az **n** természetes számot (**n>3**), majd az **a, b** és **c** három valós számot, és kiírja, szóközzel elválasztva, a következőképpen megépített számsor első **n** tagját: az **a, b** és **c** a számsor első a három tagja és a negyedik taggal kezdődően, bármely tag az őt közvetlenül megelőző **3** tag számtani középarányosa. *Példa*: ha **n = 6, a = 2, b = 5** és **c = 7**, a kiírt számsor: **2 5 7 4.66 5.55 5.74**.	**2. Feladat**: Írj programot, amely beolvassa az **n, a, b** nullától különböző természetes számokat (**a<b**), majd **n** darab természetes számot, és kiírja azoknak a számsorban található számoknak a számtani középarányosát, amelyek nem elemei az **[a, b]** intervallumnak. Ha minden szám eleme az adott intervallumnak, akkor a „Nem létezik” üzenet lesz kiírva. *Példa*: ha **n = 6, a = 20, b = 30**, a számsor pedig: **24 5 26 32 27 22**, a kiírt eredmény: **18.5**.

C++ program

```
#include <iostream>
using namespace std;
int main()
{unsigned int i,n;
 float a,b,c,d;
 cin>>n>>a>>b>>c;
 cout<<a<<' '<<b<<' '<<c;
 for(i=4;i<=n;i++)
  { d=(a+b+c)/3;
    cout<<' '<<d;
    a=b;b=c;c=d;
  }
  return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,x,i,s,db,a,b;

    cin>>n>>a>>b;
    s=0;db=0;
    for(i=1;i<=n;i++)
    {  cin>>x;
       if(x<a || x>b)
       {s=s+x;db++;}
    }
      if(db==0) cout<<"Nem letezik";
      else cout<<s/float(db);
    return 0;
}
```

Python program

```
n=int(input())
a=float(input())
b=float(input())
c=float(input())
print(a,b,c,end=' ')
for i in range(4,n+1):
    d=(a+b+c)/3
    print(d,end=' ')
    a=b
    b=c
    c=d
```

```
n=int(input())
a=int(input())
b=int(input())
s=0;db=0
for i in range(1,n+1):
    x=int(input())
    if x<a or x>b:
        s=s+x
        db=db+1
if db==0:
    print('Nem letezik')
else:
    print(s/db)
```

Módosítsátok a programot úgy, hogy az új program egy másik alapvető algoritmust használjon!

Határozzátok meg a tanárral együtt a programok megvalósításához és teszteléséhez szükséges munkaidőt, majd osszátok meg az elkészült programokat. Sok sikert!

Felmérés

1

Nyilak segítségével társítsad az algoritmust a megfelelő művelettel.

2

Határozd meg a következő állítások igazságértékét.

3

Írj programot, amely kiírja a mellékelt **4** számsort.

4

Oldd meg a mellékelt algoritmusra vonatkozó követelményeket (**n, x, y, z** nullától különböző természetes számok, **n>3**).

Légy kíváncsi, kutass!

Algoritmus	Akkor használod az algoritmust, amikor azt szeretnéd, hogy:
Összeg	felcseréld két változó tartalmát.
Szorzat	megszámláld, hogy egy esemény hányszor fordul elő.
Számlálás	kiszámolj egy összeget.
Felcserélés	kiszámolj egy szorzatot.

a. A képzett számsorokat úgy kapjuk, hogy alkalmazunk egy számítási szabályt a számsor minden tagjára
b. A beolvasott számsorok elemeinek száma mindig ismert.
c. Egy számsor beolvasásának a végét jelölheti egy olyan érték, amely nem tartozik a számsor elemei közé.
d. Egy számsor legnagyobb elemét mindig meg tudod határozni.

4 3 2 1	**4 3 2 1** **4 3 2** **4 3** **4**	**4** **4 3** **4 3 2** **4 3 2 1**	**4 3 2 1** **4 3 2 1** **4 3 2 1** **4 3 2 1**
4 elemű számsor	**10** elemű számsor	**10** elemű számsor	**16** elemű számsor

```
beolvas n, x, y
nr ← 0
minden i ← 3, n végezd el
beolvas z
ha x=y és y=z akkor db ← db + 1
ha vége
x ← y, y ← z
minden vége
kiír nr
```

a. Add meg mit ír ki az algoritmus, ha **n=7** és a számsor: **5 5 5 2 2 2**.
b. Fogalmazd meg az adott algoritmussal megoldott feladat kijelentését.
c. Alakítsd át az algoritmust úgy, hogy a kapott algoritmusban ne legyen adott a számsor elemeinek száma és a beolvasás a **0** érkezéséig történjen.

A felmérés értékelése 10 ponttal történik. 1 pont jár hivatalból.
A kijelentésben meg nem határozott elemek választhatók.
A munkaidőt a tanár határozza meg.

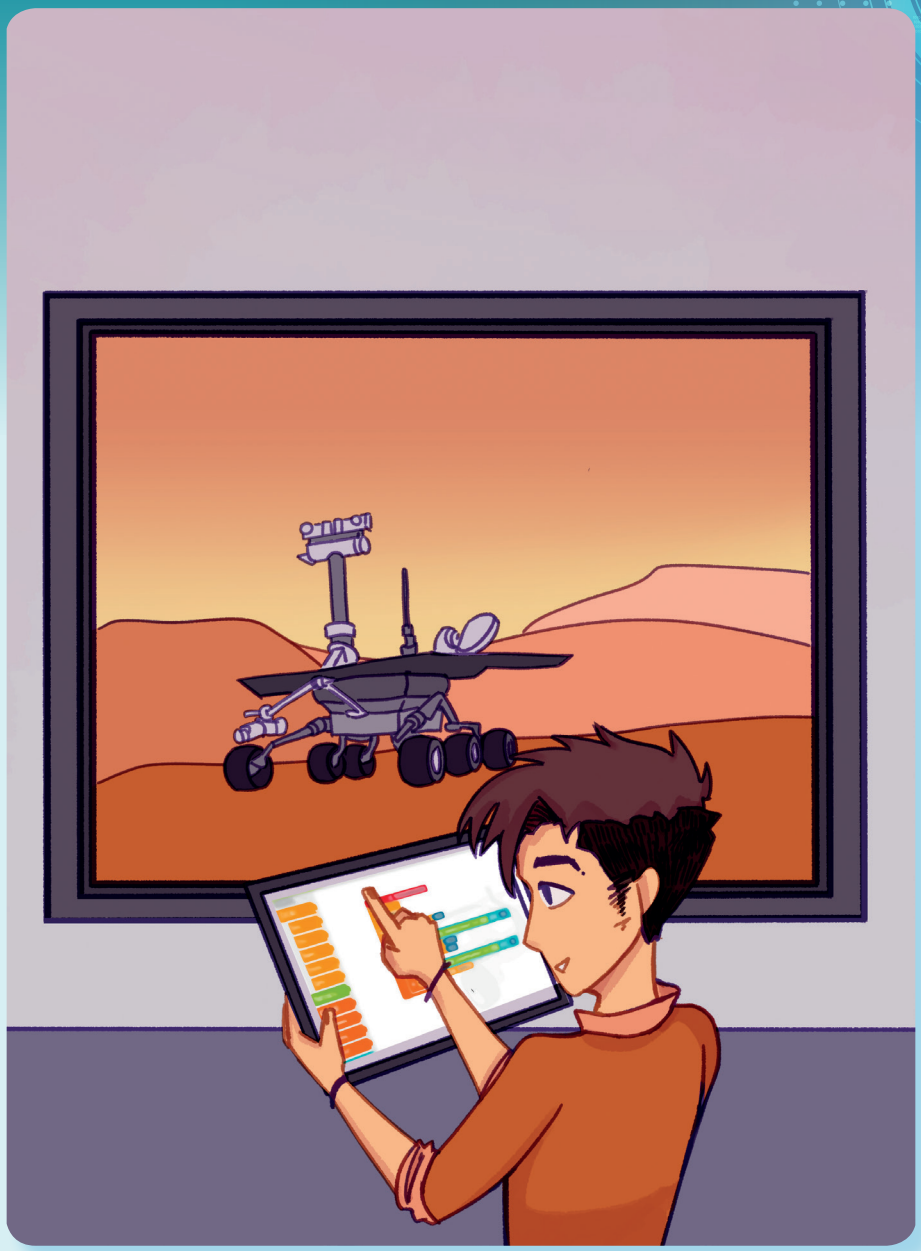
Önértékelés

Kérdés	Értékelés	Pontszám
1	Minden helyes társítás **0,5p.** **0,5p × 4** társítás	**2p.**
2	Minden helyes válasz **0,5p.** **0,5p × 4 válasz**	**2p.**
3	Minden helyes kiíratás **0,5p.** **0,5p × 4 kiíratás**	**2p.**
4	Kiíratás **1p** A kijelentés megfogalmazása **1p**. Programkészítés **1p**.	**3p.**

ÚTMUTATÓ

Kérdés	Tartalom	Oldalszám
1	Az alkalmazás felületének ismerete	48, 50, 53, 54
2	A számsor fogalmának ismerete	53, 57, 58
3	Programkészítés	55, 56
4	Programkészítés	50, 51, 59

4. Egység Robotika



Ebben az egységben megtanulod:

- hogyan szerkeszthetsz forráskódot a virtuális robot vezérléséhez;
- hogyan értelmezheted a virtuális robot érzékelői által felvett adatokat;
- olyan algoritmus-részletek írását, amelyek segítenek megérteni a mindennapjainkban használt új technológiák szerepének fontosságát.

Sajátos kompetenciák: 2.1., 2.2., 3.3.

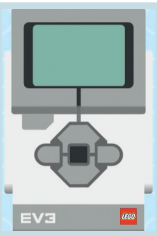
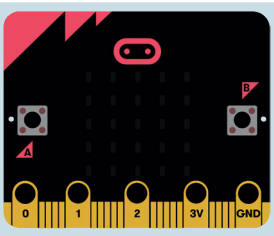
Ebben a leckében a robotot különböző virtuális környezetekben fogod programozni. A szimulációk által megértheted a környezetedben található jelenségeket. A grafikus tömbök segítségével gyakorlatba ültetheted a tanulmányozott algoritmusokat. A lecke végére felfedezed a környezeted, és képes leszel értelmezni a robot segítségével összegyűjtött információkat.

13. Lecke Az akadályok felfedezése és elkerülése

A virtuális robot az az eszköz, melynek segítségével modellezni tudod a mindennapi életben előforduló tevékenységeket és jelenségeket és ezeken keresztül megérted az új technológiák által használt informatikát.
A robot programozása megoldható egy programozási nyelvezetben grafikus tömbök segítségével is. A programozás grafikus megközelítése hozzásegít a tudományok megértéséhez, a feladatok megoldásához és az algoritmusok virtuális gyakorlatba ültetéséhez.

EMLÉKEZZ!

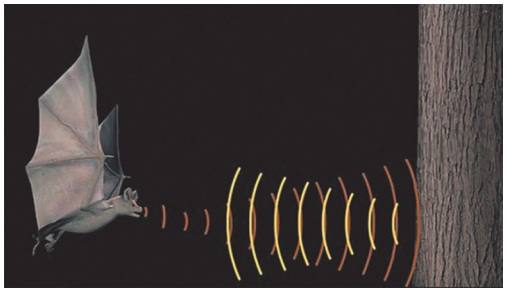
Az alábbi programozási felületeken a robotot online tudod grafikus tömbökkel programozni:

Open Roberta Lab		MakeCode	
https://lab.open-roberta.org		https://makecode.mindstorms.com	https://makecode.microbit.org
Open Roberta Sim	EV3 Ev3		
micro:bit	Calliope mini		

A robotnak olyan érzékelői vannak, amelyeket be tudsz programozni úgy, hogy tájékozódni tudjon, elkerülje az akadályokat, színeket ismerjen fel, mérjen hőmérsékletet, nyomást, fényerősséget vagy egyéb fizikai mennyiséget.

A robot az **echolokációs** módszert alkalmazza, és ultrahangos érzékelőjével képes úgy tájékozódni, hogy elkerüli az útjába kerülő akadályokat. Az **EV3** (**EV**olution) robotnál ezt az

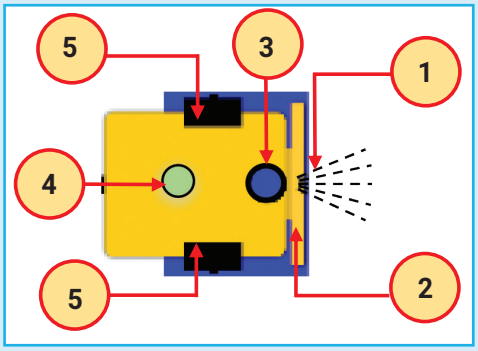
Open Roberta Lab		MakeCode	
Kategória:	Sensors	Kategória:	Sensors
get distance cm ultrasonic sensor Port 4		ultrasonic 4 distance	



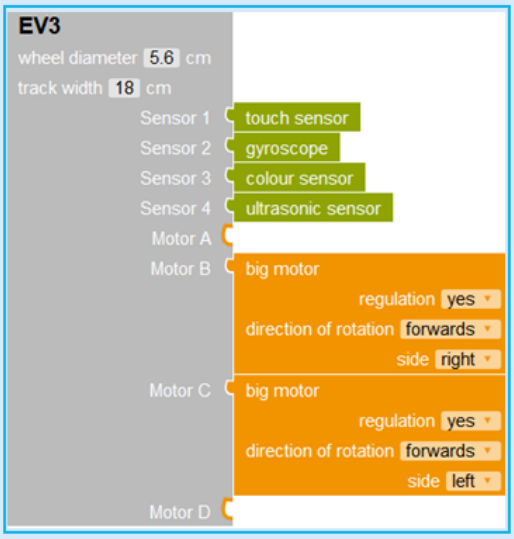
érzékelőt a mellékelt módon érhetjük el. Egyes állatfajok **echolokációval** képesek tájékozódni és érzékelni az előttük lévő akadályokat. Az állatok hangokat adnak ki, és érzékelik a visszhangot. A visszhang érzékelésének időpontjától függően meg tudják határozni, hogy milyen távolságra vannak az elkerülendő akadályok. Echolokációt alkalmazó állatok például a delfinek, bálnák és a denevérek. Az általuk kibocsátott hangokat, amelyeket tájékozódásra, élelemszerzésre és az akadályok elkerülésére használnak, az emberi fül nem érzékeli.

Az **Open Roberta** modellezési felületen az **EV3** robot a következő összetevőkből áll

1. **Ultrahangos érzékelő**
2. **Az előlapra felszerelt érintkezésérzékelő**
3. **Színérzékelő**
4. **LED: piros, narancssárga és zöld színeket jelenít meg**
5. **Motorok által meghajtott kerekek**



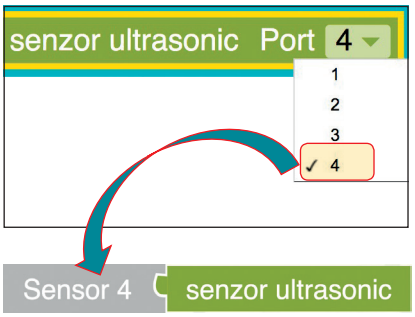
Modellezési felület



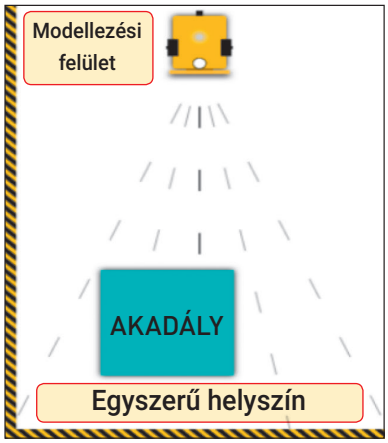
Beállítási felület

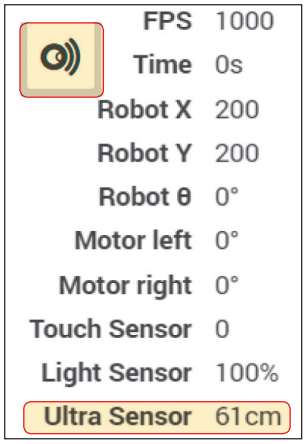
Az ultrahangos érzékelő **centiméterben méri a távolságot** a közelben levő tárgyakig. Így, ha a robotnak különböző feladatokat adunk, a tárgyakhoz képest meg tud tartani egy bizonyos távolságot, és sikerül elkerülnie is ezeket.

Ahhoz, hogy láthassuk az érzékelők által felvett adatokat, a *Modellezési Felületről (Simulation View)* aktiválni lehet a *Megnyit/Bezár érzékelők ablak (Open/Close the senzors data view)* gombot.

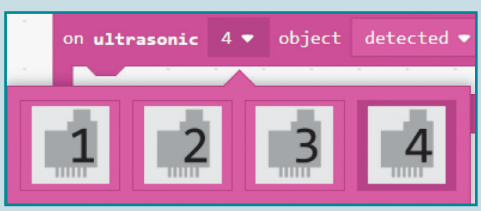


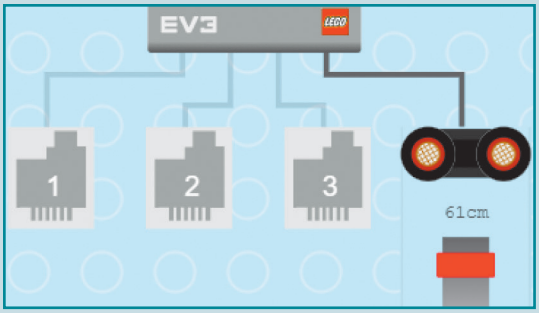
Vigyázat! Az érzékelő helye *Beállítási Felületen* meg kell egyezzen a *Modellezési Felületen* lévő **Port** számával.





A MakeCode felületen, az EV3 robot ultrahangos érzékelője alapból a 4-es Portra van beállítva , de át lehet állítani más Portra.



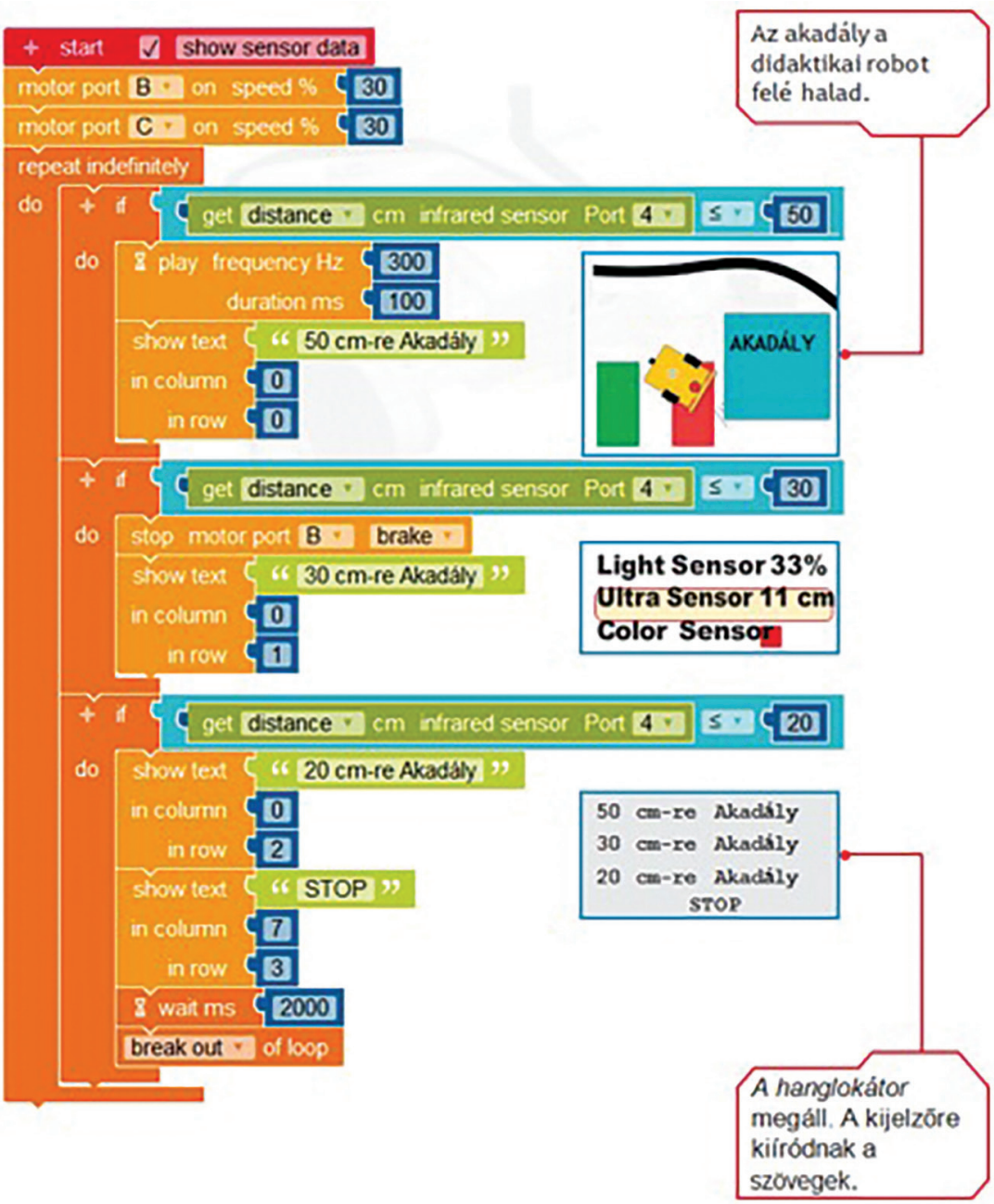


GYAKORLAT // A HANGLOKÁTOR

Nyisd meg az **Open Roberta Lab programozási felületet, válaszd ki az EV3 didaktikai robotot és az Egyszerű helyszínt (Scene)**. Programozd az ultrahangos érzékelővel rendelkező EV3-as robotot úgy, hogy ha a robot és az akadály között a távolság kisebb vagy egyenlő mint:

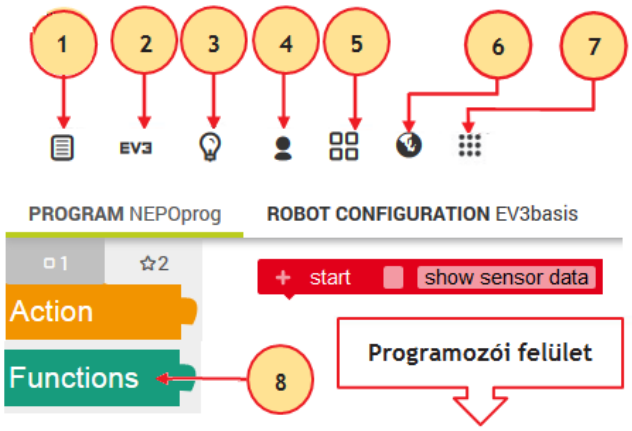
- **50 cm**, akkor adjon ki egy **100 milliszekundum** időtartamú, **300 Hertz**-es vészjelző hangot, és írja ki az "50 cm-re Akadály" szöveget;
- **30 cm**, akkor állítson le egy motrot, és írja ki a „30 cm-re Akadály" szöveget;
- **20 cm**, akkor írja ki a „20 cm-re Akadály" és a „STOP" szövegeket, és 2 másodperc múlva álljon le a robot.

Aktiváld a robot képernyőjét, és a szövegeket soronként, egymás alá írasd ki. Kezdetben a két motor egyforma hatással bír a robot kerekeire. Mentsd a programot **Hanglokátor** névvel a te mappádba.



1. lépés Nyisd meg a Menü-sorból a **Robottípusokat**, a listából válaszd ki az **EV3 (c4ev3)** robotot.

1. Szerkesztés
2. Robottípusok
3. Súgó
4. Felhasználó
5. Galéria
6. Nyelv beállítása
7. Felülettípusok
8. A grafikus tömbök szín, tevékenység és szintek szerinti csoportosítása: **Kezdő szint** - 1 zóna, **Haladó szint** – 2 zóna



2. lépés Húzd át a 2-es zónában található grafikus tömböket a *Programozói felületre*. Tájékozódj a színek és a formák szerint!

3. lépés Nyisd meg a *Robot szerkesztési felületet (ROBOT CONFIGURATION)* és ellenőrizd le az ultrahangos érzékelő, valamint a motrok helyét. Ezeknek egyezniük kell a *Programozói felület* Portjaival.

4. lépés Nyisd meg a *Modellezési Felületet*. Nyomd meg az *Érzékelők* gombot *(Senzor)*, hogy tudd követni a megnyílt ablakban az érzékelők értékeit. Nyomd meg az *EV3* gombot, hogy láthasd a robot kijelzőjét. Nyomd meg a *Start*-ot.

5. lépés A felületen úgy mozgasd el az akadályt, hogy az akadály és a robot közötti távolság kevesebb legyen mint **50 cm**. Fedezd fel a *Hanglokátor* által kiadott figyelmeztető hangot. Folytasd a távolság csökkentését. Amikor ez kevesebb lesz mint **30 cm**, a *Hanglokátor* felméri a körülötte levő felületet. El fog fordulni, mivel csak az egyik motor működik. Csúsztasd az akadályt addig, amíg a távolság kevesebb lesz mint **20 cm**. A *Hanglokátor* megáll. Figyeld meg a kijelzőre kiírt üzeneteket.

6. lépés Nyisd meg az Edit menüt és exportáld a programot **Hanglokátor** néven a saját mappádba. Lementheted a programot online verzióban is. Nyisd meg a *Felhasználó (User)* menüt és kapcsolódj be. Ha nincs fiókod, akkor létrehozhatsz egyet, csak az e-mail címedre lesz szükség. Utána nyisd meg az *Edit* menüből a *Mentés másként (Save as)* utasítást.
Gratulálunk! A virtuális hanglokátor felfedezte, felkutatta az útjába került akadályt!

FEDEZD FEL!

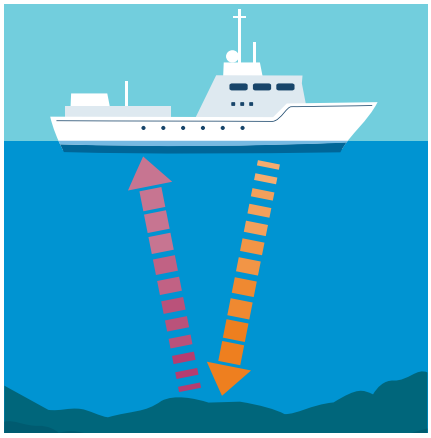
A hanglokátor, angolul **SONAR**, (**SO**und **N**avigation **A**nd **R**anging) olyan technika, amely hanghullámokat használ fel arra, hogy felfedezze a tárgyakat, vagy a tengerfenék domborzati formáit.

A **hang** négy jellemzője: magasság, erősség, időtartam és hangszín. A frekvenciát **Hertz(Hz)**-ben mérik és a hang magasságának felel meg.

A hang erősségét **deciBell (dB)**-ben mérik.

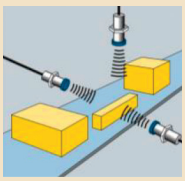
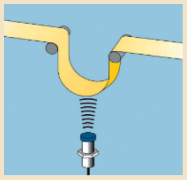
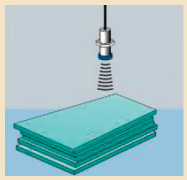
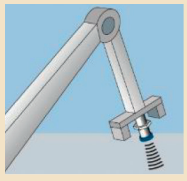
A passzív hanglokátor egy olyan műszer, amely a tengeralattjárók, hajók, egyes állatok, stb. által kiadott hangokat érzékeli. Nem szennyezi a környezetet, mivel nem ad ki káros hangokat.

Az aktív hanglokátor egy csoport hanghullámot bocsát ki, a környezet visszaveri ezeket a hanghullámokat, amit a hanglokátor érzékel. Ennek működése káros hatással lehet a tenger élőlényeire. Emiatt a hanghullámok frekvenciáját olyan szintre kell állítani, ami nem zavarja a környezetet.

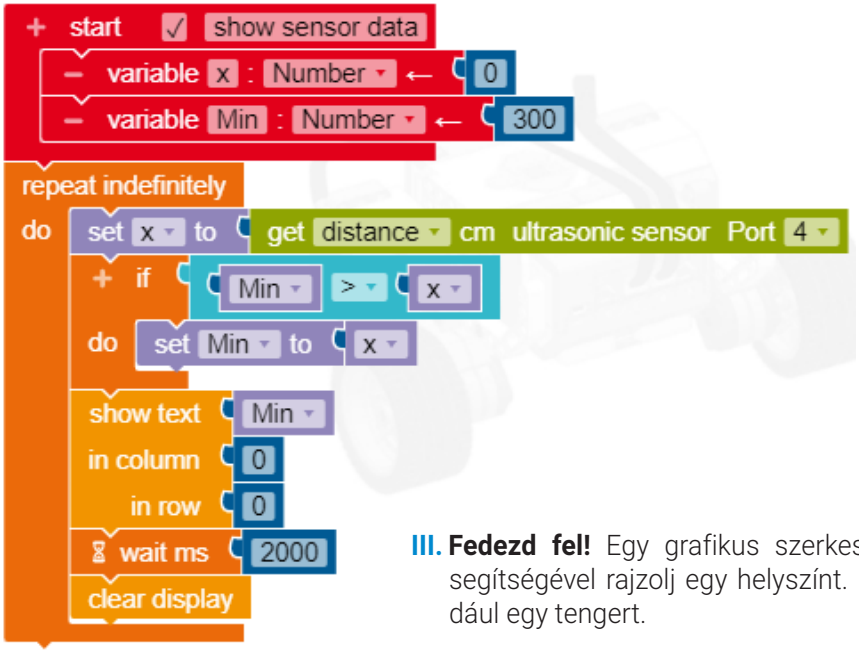


JAVASOLT FELADATOK

I. **Fedezd fel** az ultrahangos érzékelők használatát a mindennapokban!

http://y2u.be/cbssaA8nIF8	**Ultrahangos érzékelők**
Dobozok szkennelése	**Mérések** 
A forgók sebességének korlátozása	**A tűrődés ellenőrzése** 
A rakás magasságának meghatározása	**A magasság meghatározása** 
A robotkar helyzekedése	**A robotkar** 
http:// y2u.be/P-ArXkuXIfo	**Minőség-ellenőrzés**

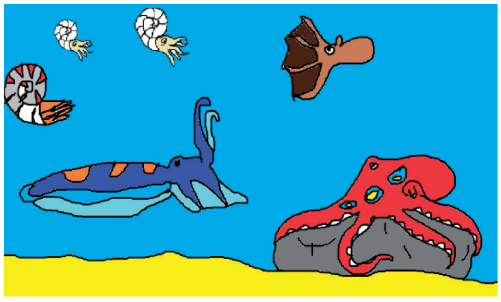
II. **Gyakorolj!** Nyisd meg az **Open Roberta Lab** programozási felületet. Válaszd ki az **EV3** robotot és az *Egyszerű helyszínt*, amelynek van egy mozgó akadálya. Készíts egy programot, amely kiírja a robot kijelzőjére a robot és az akadály közötti minimális távolságot, tudva azt, hogy a robot nem mozog, de az akadályt elmozdíthatod az egér segítségével. A minimális érték két másodpercenként frissül a robot kijelzőjén. A program futtatása közben, hogy követhesd a minimális érték változását, nyisd meg az *érzékelők ablakát és a kijelző ablakát*. Mentsd a programot a saját mappádba **Felfedezés** néven.



Az elkészített kép feltöltéséhez használd a *Helyszínfeltöltő* gombot az **Open Robertá**ban. Úgy programozd az EV3 robotot, hogy pásztázza végig a tengerfenéket oly módon, hogy ne zavarja a környező állatvilágot.

Mentsd a programot a saját mappádba **Feltérképezés** néven.

III. **Fedezd fel!** Egy grafikus szerkesztő segítségével rajzolj egy helyszínt. Például egy tengert.



JEGYEZD MEG!

- **A virtuálisan programozott didaktikai robot képes arra, hogy felfedezze a környezetét és információkat szolgáltasson róla.**
- **Az ultrahangos érzékelő segítségével a didaktikai robot képes felfedezni és elkerülni az akadályokat.**
- **Ahhoz, hogy egy robot egy fal mellett folyamatosan tudjon mozogni, az ultrahangos érzékelőt használja, amellyel képes megmérni a közte és a fal közötti távolságot.**

Ebben a leckében még többet fogod használni a robot érzékelőit, hogy életet lehelj a robotba. Sok gyakorlással és türelemmel sikerülni fog úgy programozni a robotot, hogy az egy kijelölt útvonalat kövessen. Végül sikerülni fog, hogy egyedül tervezz meg egy pályát, és az általad beprogramozott robot pedig sikeresen bejárja azt, megtartva az egyensúlyát.

14. Lecke A kijelölt útvonal követése

Az robotika lépésről lépésre rávezeti a diákokat az oktatási feladatok kidolgozására és megoldására, és segíti az eredmények ellenőrzésében.

A virtuális modellezéssel való munka nagyon hasznos, mivel számos kísérletet lehet végezni. Változtathatunk a robot dinamikáján, többször is tesztelhetjük, a kapott eredményeket feljegyezhetjük, és folyamatosan változtathatunk rajta, míg az összes hibát kiküszöböltük.

A robot a **színérzékelőjét** használva képes a kijelölt útvonal követésére. Az **EV3** robotnál az érzékelő fényt bocsát ki. Ennek az érzékelésre három lehetősége van: **szín, fény** és **RGB**.

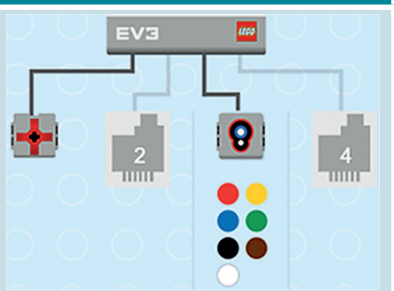
Open Roberta Lab	MakeCode
Kategória: Sensors	Kategória: Sensors
get colour — colour sensor Port 3	color sensor 3 color
get light — colour sensor Port 3	color sensor 3 reflected light
get RGB — colour sensor Port 3	color sensor 3 RGB raw

A színérzékelő	Észleli az érzékelő alatt elhelyezkedő alapszíneket: **fekete, barna, kék, zöld, piros, sárga, fehér** és **szürke**.
A fényérzékelő	Méri a visszaverődő fény erősségét. A kapott érték szám típusú, **0** és **100 között, ahol a 0 = nagyon sötét, és a 100 = nagyon világos.**
Az RGB	Méri a fény erősségét egy **0-tól 100-ig terjedő skálán**; minden alkotóelem esetében: **R**ed-Piros, **G**reen-Zöld és **B**lue-Kék.

Néha a robot által végigjárt pálya zárt körű, amelyet megállás nélkül be tud járni. Így a programban léteznie kell egy megállási feltételnek. A robot egy idő után megállhat akkor, ha például egy akadálytól, vagy a faltól egy bizonyos távolságra van, vagy ha egy bizonyos billentyűt lenyomunk, stb. Az **érintésérzékelő** bekapcsolásával a robot érintésre érzékennyé válik, így meg tud állni, ha elér egy akadályt vagy a falat. Az **érintésérzékelő** által visszatérített érték logikai típusú: **Igaz**, ha **be van kapcsolva**, és **Hamis**, amikor **ki van kapcsolva**.

MakeCode

Alapból a **színérzékelő** a **3-as Porthoz** van kapcsolva, és az **érintésérzékelő** az **1-es Porthoz**.

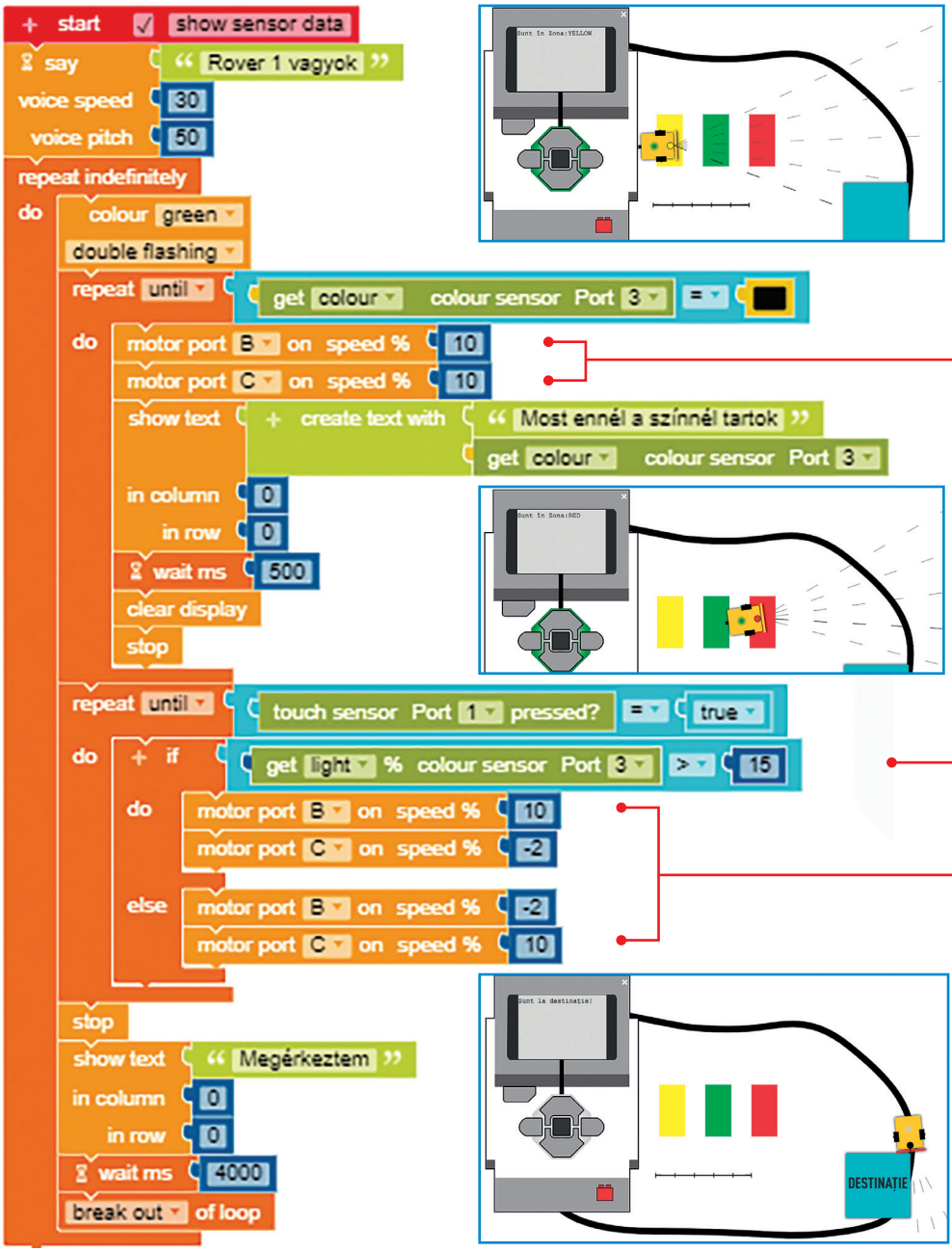


GYAKORLAT // A ROVER

Az **Open Roberta Lab** programozási felületen válaszd ki az **EV3** robotot. Nyisd meg az *Egyszerű helyszínt* és helyezd el a mozgó akadályt a körpályán. Az EV3 robotot, amelynek van egy **színérzékelője** és egy **érintésérzékelője**, úgy programozd, hogy:

- az elején mutatkozzon be, és mondja, hogy „Rover 1 vagyok ";
- a program teljes ideje alatt zölden és dupla villogással égjen a LED-je;
- addig menjen előre, amíg a fekete színű körpályához ér. Ez idő alatt a robot kijelzőjén jelenjen meg az aktuális felület színe, 500 milliszekundumonként frissítve a helyzetét;
- menjen végig a körpályán addig, amíg eléri az akadályt;
- miután elérte az akadályt, 4 másodpercig jelenítse meg a kijelzőn a „Megérkeztem!" üzenetet.

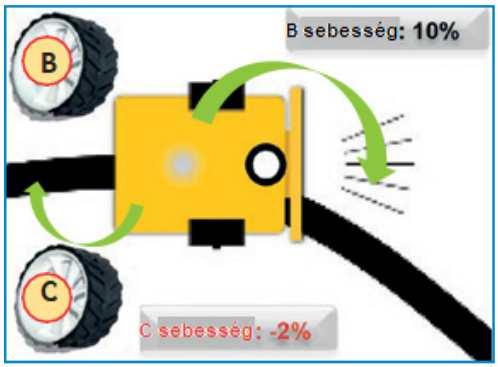
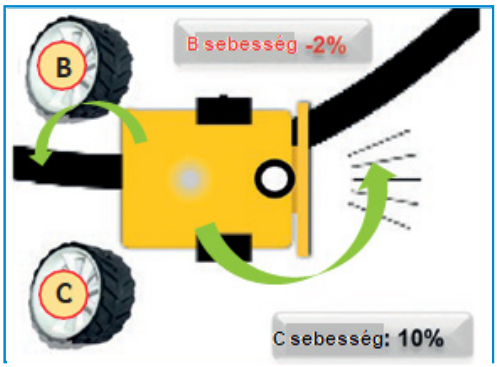
Kezdetben a két motor egyforma hatást gyakorol a robot kerekeire. Mentsd a programot **Rover** néven a saját mappádba.



A sebesség értékeinek beállítása több próbával történik *(2. lépés)*.

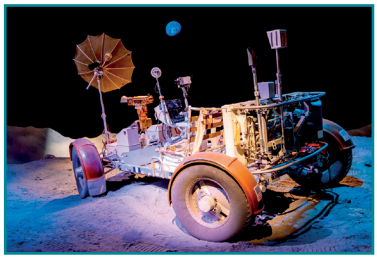
A körpályavonal középtengelyének követése *(3.lépés)*.

A vonal követése az útvonal kanyargósságától függetlenül *(4.* lépés).

1. lépés	Válaszd ki az **EV3 (c4ev3)** robotot és az *Egyszerű színhelyet*, amely mozgó akadály tartalmaz. Az egér segítségével húzd az akadályt a körpályára.
2. lépés	Húzd az alkalmazásnak megfelelő grafikus tömböket a 2-es zónából a Programozási felületre. **Figyelem! Határozd meg, többszöri próbával, a motorok sebességét.** Ha a sebesség túl nagy, a színes területek bejárása után a robot túl hamar átlép a pálya vonalán és nem találja meg azt, és az *Egyszerű színhely* falában fog megállni.
3. lépés	A körpálya vonala **fekete** színű, így a fényérzékelő értéke **0** kell legyen **(nagyon sötét)**. Ahhoz, hogy elérjük azt, hogy a robot a pálya vonalának a közepét kövesse, figyelembe kell venni a **visszaverődő fényt.** Így a fényérzékelőnek állíts be **15**-nél nagyobb értéket.
4. lépés	**Figyelem! Vedd rá a robotot, hogy kövesse a vonalat.** Amikor a motor a robot mozgásával ellentétes irányába forog, a kijelzőn negatív értékek jelennek meg. Amikor a kerekek ugyanabba az irányba forognak, mint a robot mozgási iránya, akkor a motor pozitív értékeket jelez. A sebesség beállítása közelítéssel, több próbával, finomítással történik annak érdekében, hogy a robot a vonalat kövesse, függetlenül a kanyargós pályától.
	Jobb kanyar 
	Bal kanyar 
5. lépés	Nyomd meg az *Edit* gombot és *exportáld* a programot **Rover** névvel a saját mappádba. **Gratulálunk! A virtuális felületen levő Rover felfedezte a környezetét és megérkezett a célhoz!**

FEDEZD FEL!

Lunar Roving Vehicle	Opportunity Mars Rover
A három űrkísérletben: Apollo 15, 16 és 17 a **LRV** (**L**unar **R**oving **V**ehicle) robotot használták. Az Apollo 17 küldetésben a robot **35,9 km-t tett meg 4 óra és 26 perc alatt.**	Jelenleg az **Opportunity** a legidősebb Rover. Őt azért vetették be, hogy vizet keressen a Marson. **15 évig kutatott aktívan a Mars felszínén.**



FELADATOK

I. **Fedezd fel** a vonalkeresés technikáját a mindennapokban!

http://y2u.be/peOM_Nk4AEY

Konténer-szállítás

http://y2u.be/WIlS3vNSuQ4

A vonalak automatizálása

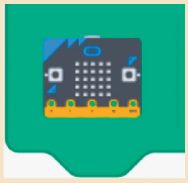
II. **Figyeld meg,** hogyan kommunikálnak a Scratch 3.0 projektek fizikai eszközökkel akkor, amikor a bővítményblokkok engedélyezve vannak.

https://en.scratch-wiki.info/wiki/Scratch_Extension

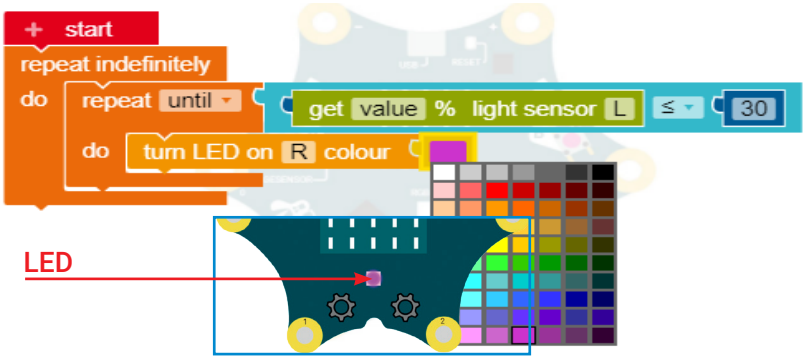
Bővítmények aktiválása

Az EV3 robot

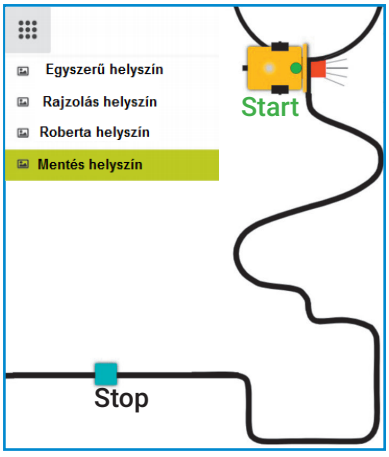
Micro:bit



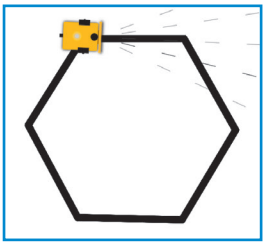
III. **Fedezd fel!** Nyisd meg az **Open Roberta Lab** programozási felületet és válaszd ki **Calliope mini** didaktikai robotot. Ennek van egy olyan fényérzékelője, amely érzékeli a környezet fényét. Az érzékelő szám típusú értékeket jegyez **0 = nincs fény** és **100 = erős fény** között. Írj egy programot, amely kigyújtva tartja a LED-et addig, amíg a környezeti fény sötét.
Mentsd a programot a saját mappádba **Szentjánosbogár** néven!



IV. **Gyakorolj!** Nyisd meg a *Modellezési felületet* az **Open Roberta Lab** programozási felületről, és válaszd ki a *Mentés helyszínt*. Helyezd a helyszínre az **EV3** robotot az alábbi ábrához hasonló módon. Készíts egy programot, amely elindítja a robotot a **Start**-tól(zöld zóna), majd végigviszi a pályán a **Stop**-ig (mozgó akadály). Mentsd a programot a saját mappádba **Pálya** névvel!



V. **Alkalmazd!** Nyisd meg a *Modellezési felületet* az **Open Roberta Lab-**on és válaszd ki a *Rajzolás helyszínt*. Programozd az **EV3** robotot úgy, hogy rajzoljon egy hexagont. Ments a programot a saját mappádba **Hexagon** névvel.



JEGYEZD MEG!

- A robot a színérzékelő használatával képes a kijelölt pálya vonalának követésére.
- Az egyensúly miatt a motorok sebességét többszörös próbával kell beállítani.
- Amikor a fényérzékelőt használod, figyelembe kell venned a visszavert fény erejét.
- Amikor a robot egy zárt körpályán jár, használhatod a érintésérzékelőt ahhoz, hogy a megállási feltételeket beállíthasd.

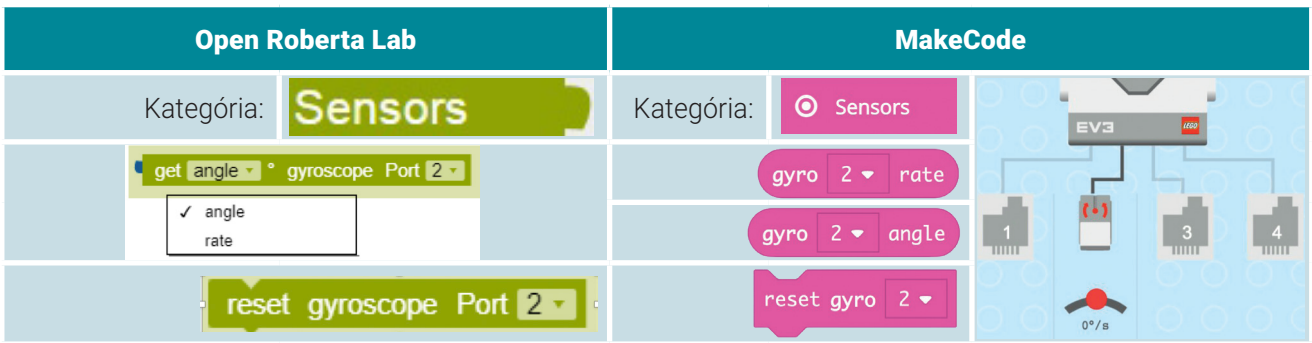
Ebben a leckében az osztálytársaiddal közösen arra fogjátok programozni a virtuális robotot, hogy bejárjon egy kijelölt útvonalat. A robot egy giroszkópos érzékelőt használhat ahhoz, hogy tájékozódni tudjon a kijelölt pályán. Többszöri próbálkozás után látni fogod, hogy a didaktikai robot hogyan mozog biztonságosan, megtartva az egyensúlyát.

15. Lecke A kijelölt útvonal bejárása

A technikai készségek elsajátítása fontos tényező az új technológiák megismerésében, elsajátításában és a más tantárgyak keretén belüli felhasználásában.

A csoportmunka, amelyet a robot programozásánál fogunk használni, nagyon hasznos. A kommunikáció és az ismeretek megosztása segít a társadalmi kapcsolatok kialakításában és a feladatok gyorsabb megoldásában.

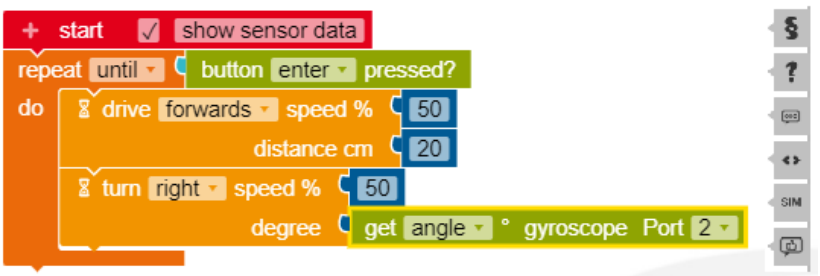
Nagyon sok módszer létezik arra, hogy egy robot hogyan járja be a kijelölt utat. Nagyon fontos az akadály és a pálya útvonala által megkövetelt mozgási módszer. A kijelölt útvonalon való mozgásnál figyelembe kell venni a robot beállításait és a biztonságos bejáráshoz szükséges időt. A robot a biztonságos mozgáshoz a **giroszkópos érzékelőt** használja. Ez az érzékelő segíti a robotot a saját tengelye körüli forgásában és a helyes útvonal kiválasztásában.



A **giroszkópos iránytű** egy olyan eszköz, amelyet a járművek a tájékozódás céljából használnak. Általában a repülők, hajók, űrhajók, stb. fedélzetére szerelik.

Az **elektromos giroszkópos érzékelő** segíti az okostelefon képernyőjének elfordulását is.

Ahhoz, hogy a giroszkópos érzékelő értékeit megfigyelhesd, nyisd meg az **Open Roberta** felületen az alábbi programot. Válaszd ki az **EV3** robotot és a *Matematika helyszínt*. Nyomd le az *Enter* gombot a robot megállításához.



Figyelem

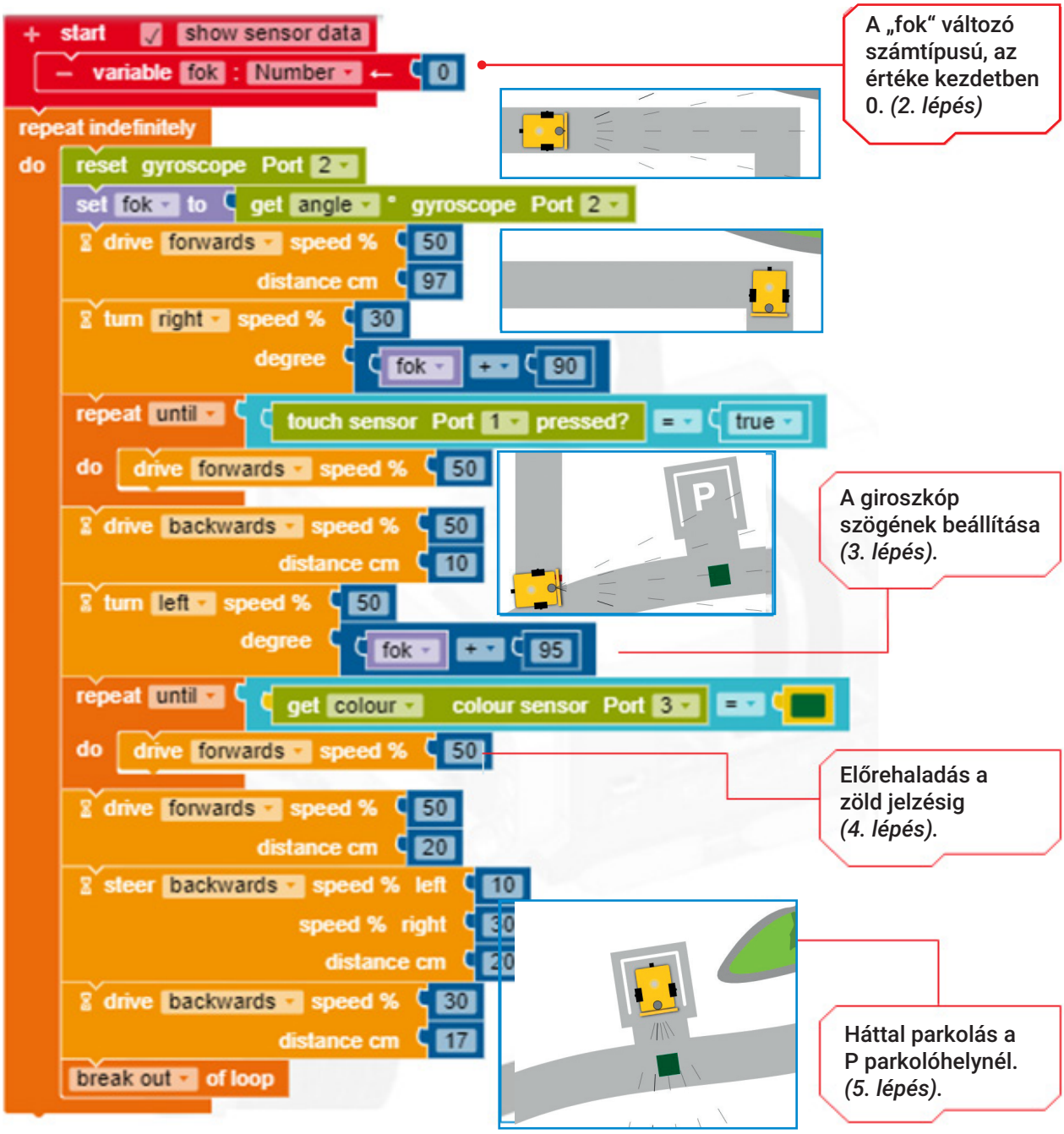
Az **EV3** robot által rajzolt rajzot úgy tudod letörölni, ha Bezárod/Kinyitod a Modellezési felületet, vagy ha mozgatod az egér görgetőjét.

GYAKORLAT // A SZÁLLÍTÓ

Az **Open Roberta Lab** felületről válaszd ki azt az **EV3** robotot, amelynek van színérzékelője, érintésérzékelője és giroszkópos érzékelője is. Nyisd meg a *Roberta Helyszínt*, és programozd a robotot úgy, hogy:

- induljon el a kezdeti helyzetéből, és menjen előre **97 cm**-t;
- forduljon jobbra **90°-t, és folytassa az útját, míg eléri a falat**;
- menjen hátrafelé **10 cm-t, és utána forduljon el balra 95°-t**;
- menjen előre, míg eléri a zöld jelzést a **P-vel jelölt parkolóhely előtt**;
- parkoljon be háttal a **P** parkolóhelyre.

Kezdetben a robot motrai egyformán hatnak a kerekekre. Mentsd a programot a saját mappádba, **Szállító** névvel.



A „fok“ változó számtípusú, az értéke kezdetben 0. *(2. lépés)*

A giroszkóp szögének beállítása *(3. lépés)*.

Előrehaladás a zöld jelzésig *(4. lépés)*.

Háttal parkolás a P parkolóhelynél. *(5. lépés)*.

1. lépés

Válaszd ki az **EV3 (c4ev3)** robotot és a *Roberta Helyszínt*. Nyisd meg a *Beállítási felületet* és ellenőrizd az **érintésérzékelő, a színérzékelő és a giroszkópos érzékelő** portjait.

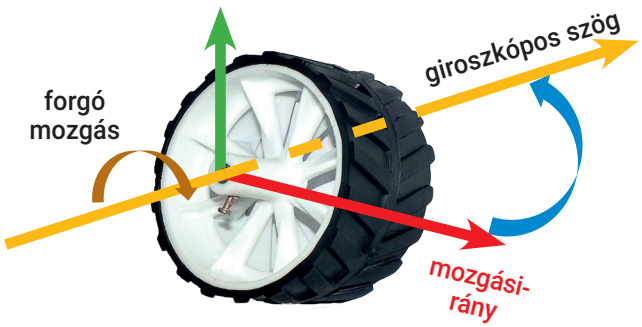
2. lépés

Deklaráld a **fok** nevű, szám típusú változót. A változó a giroszkópos érzékelő tájékozódásához szükséges szögének értékét jelöli. Állíts be a változónak 0 értéket. Húzd a *Programozási felületre* a 2. Zónából a megfelelő grafikus tömböket.

3. lépés

Többszörös próbával állapítsd meg a **giroszkópos érzékelő** értékét és a mozgási sebességet. A giroszkópos érzékelőnek két választási lehetősége van **0°** és **360°** közötti értékekkel.

Az egyik meghatározza a haladási irányhoz mért szög értékét, a másik pedig a kerekek dőlési szögét a függőleges tengelyhez képest.
A pálya elején határozd meg az mozgás érzékelő kezdőértékét, ezt majd tanácsos visszaállítani, mert a haladás során különböző sebességeknél vibrálni kezd a robot.

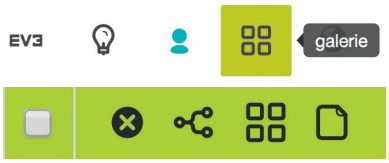


4. lépés

Figyelem! Miután a robot eléri a falat, többszöri próbával állapítsd meg a robot sebességét!
Ahhoz, hogy a zöld jelzéshez érjen, állítsd a giroszkópos érzékelő szögét la 95-ra, a sebességet, a távolságot és a színérzékelőket. A Szállító kis sebességgel kell haladjon ahhoz, hogy háttal parkolni tudjon a P parkolóhely előtt.

5. lépés

Nyomd meg a ***Szerkesztést*** és *exportáld a programot Szállító néven a saját mappádba. A Menüből válaszd ki a* ***Szerkesztést*** *(Edit) és utána az* ***Én programjaimat*** *(My programs) és oszd meg (Share) a programot a társaiddal.*



Gratulálunk! A virtuális szállító bejárta a kijelölt útvonalat, és elérkezett a kezdeti helyzetétől a végpontba!

FEDEZD FEL!!

Bányászrobotok	CoroBotok
A Julius nevű robotok nagy segítséget nyújtanak a bányászatban. Egyenként **180 kg** és **400 kg** közötti ásványt szállítanak, átlagsebességük **1 m/s**.	Dél-afrikai aranybányákban előszeretettel használják a mozgatható és kibontható felületekkel rendelkező robotokat az emberek sebesülési lehetőségének csökkentése érdekében, valamint a nehezen elérhető helyekre való bejutás miatt.

KITŰZÖTT FELADATOK

I. **Fedezd fel**, hogyan járják be a robotok a bányákat a valóságban!

http://y2u.be/K_r2Bc33eyo

Rox - robot a bányában

http://y2u.be/0qryoXI2ZN8

Julius - robot a bányában

II. **Figyeld meg** a giroszkópos érzékelő felhasználását a mindennapokban

A szereplők irányítása

Joystick

A kamera forgása

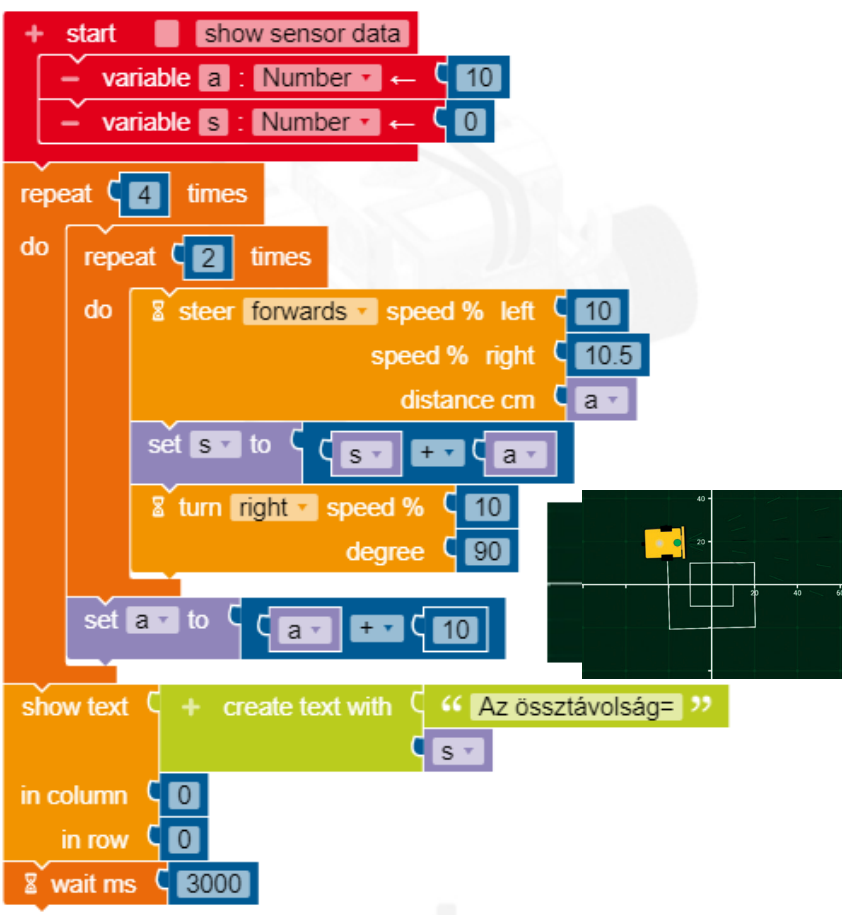
Térfigyelo kamera

A propeller forgása

A helikopter

III. **Fedezd fel!** Az **Open Roberta Lab** programozási felületen válaszd ki az **EV3 didaktikai robotot** és a *Matematika helyszínt*. Programozd a robotot úgy, hogy elvégezze az alábbi feladatokat:

- Induljon el kezdő helyzetből;
- Járja be egyenként a számsorban található távolságokat: 10 10 20 20 30 30 40 40;
- Az adott távolságok mindegyike után forduljon a robot **90°-ot jobbra**,
- **3 másodpercig** írja ki **a sorozatban levő távolságok összegét.**



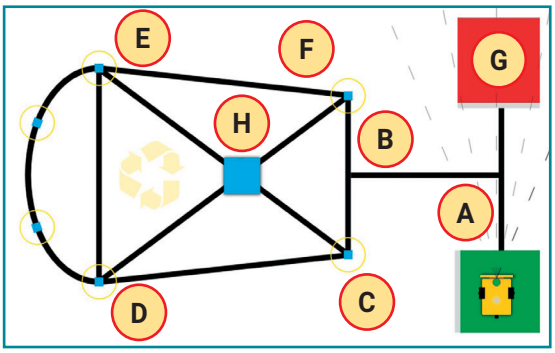
IV. **Alkalmazd! Írd át a fenti programot úgy, hogy az EV3-as robot minden távolság után balra forduljon 90°-ot.**

JEGYEZD MEG!

- **A kijelölt útvonal bejárása** a robot beállításától és a pálya jelzéseitől függ.
- A robot a pályán a **giroszkópos érzékelő** segítségével **tájékozódik**.
- Ha **csoportban dolgozol**, a feladatokat gyorsabban el tudod végezni.
- A virtuális platformokon lehetséges az **online együttműködés**, a programok megoszthatók másokkal.

ISMÉTLÉS /// CSAPATMUNKA!

Mindazt, amit elméletben tanultál a Lokátor, a Rover és a Szállító gyakorlatoknál, kipróbálhatod az **Open RobertaLab felületén** a *World Robot Olympiad Scene helyszínen*. Mivel a pálya hosszú, és sok próbálkozásra van szükség a bejárásához, ajánlott a csoportmunka. Sőt! Jó lenne, ha 3 csoportot alakítanátok ki különböző feladatkörökkel. Így a a teljes pálya bejárásához szükséges program könnyebben és hamarabb elkészül!



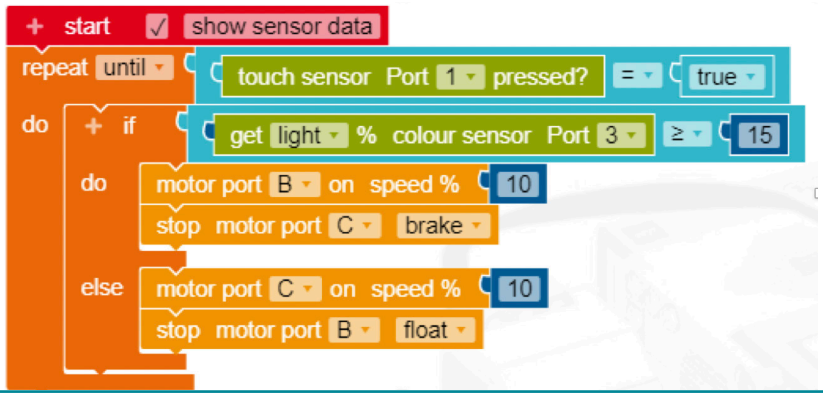
Minden csapat külön programot ír a neki megfelelő feladat alapján! A csapatok a tanár segítségével meghatározzák a programok megírására és tesztelésére szükséges időt.

A csapatok megosztják az elkészült programokat, majd egyesítik egy nagy programmá úgy, hogy az EV3 robot be tudja járni a teljes pályát felhasználva mind a 4 érzékelőt: **érintés, szín, ultrahang** és **giroszkóp.**

1. Csapat – megírja azt a programrészt, amivel az **EV3 robot bejárja az ABCHCD-pályarészt.** A **H pontban** levő akadály **megérinthető**!

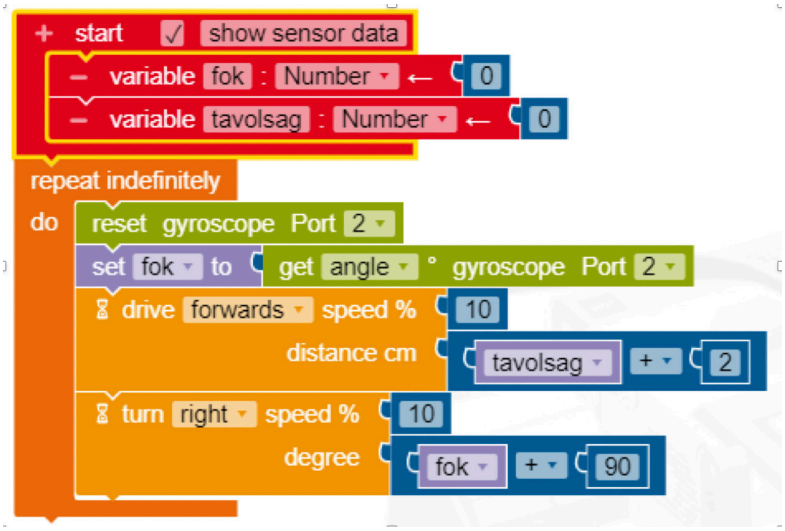
2. Csapat – megírja azt a programrészt, amivel az **EV3 robot bejárja a DE kanyart, majd az EHEF részt.** A **H pontban** levő akadály **megérinthető**!

A bejáráshoz a csapatok felhasználhatják a **Rover** gyakorlatnál készített programot, de tesztelhetik az alábbit is:



3. Csapat – megírja azt a programrészt, amivel az **EV3 robot bejárja az FHFBG pályarészt.** A **H pontban** levő akadály **nem érinthető meg**!

A tájékozódáshoz a csapatok felhasználhatják a **Szállító** gyakorlatnál készített programot, de tesztelhetik az alábbit is:



Gratulálunk! A csapatok sikeresen együttműködtek, és megoldották a feladatot!

Felmérés

1

Kapcsold össze nyilak segítségével az első oszlopból az érzékelőt a második oszlopból a megfelelő tevékenységgel.

2

Segíts az EV3 robotnak **rombuszt** rajzolni **az Open Roberta felületen.**

3

Az **EV3**-asnak van egy ultrahangos érzékelője. Segíts neki bejárni a kijelölt pályát a kezdő helyzetből az akadály eléréséig!

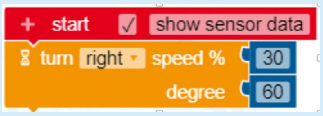
4

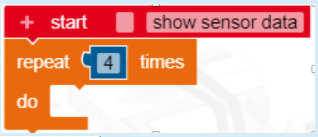
Programozd az **EV3 robotot** úgy, hogy kövesse a pálya vonalát a kezdő helyzettől **a második piros jelzésig.**

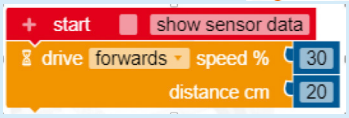
Légy okos, fedezd fel!

Érzékelő	Tevékenység
ultrahangos	Egy megadott fokkal való elfordulást határoz meg.
szín	Érzékeny az érintésre.
érintés	A tárgyaktól való távolságot méri centiméterben.
giroszkópos	Az érzékelő alatti színeket határozza meg.

Válaszd a *Matematika helyszínt* és az alábbi grafikus tömböket:

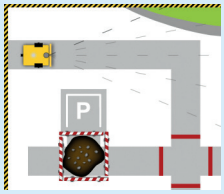






Mentsd a programot a saját mappádba **Rombusz** névvel.

Válaszd a Roberta helyszínt és helyezd el az akadályt a pályán a P-vel jelölt parkolóhely elé. Mentsd a programot a saját mappádba, **Jelzés** névvel.



Válaszd a *Mentés helyszínt*. Az **EV3**-nak van egy színérzékelője. Mentsd a programot a saját mappádba **Szín** névvel.

A felmérő 10 pontot ér. Hivatalból jár egy pont.
A feladat szövegében nem meghatározott elemek választhatók.
Az időt a tanár határozza meg.

Önértékelés

Elem	Javítókulcs	Pont
1	Minden helyes párosítás **0,5p.** **0,5p × 4 párosítás**	**2p.**
2	A tömbök helyes sorrendje **1p.** Helyes rajz **1p.**	**2p.**
3	Az útvonal bejárása **1p.** Megállási feltétel **1p.**	**2p.**
4	A vonal követése **1p.** Az egyensúly megtartása **1p.** Megállási feltétel **1p.**	**3p.**

ÚTMUTATÓ

Elem	Tartalom	Oldal
1	Az érzékelők szerepének ismerete	71, 75, 79
2	A grafikus tömbök ismerete	72, 76, 78
3	A program megírása, a kijelölt pálya feltételeinek kezelése	79–80, 83
4	A program megírása, a vonal követéséhez szükséges feltételek kezelése	75–77, 83

Az elmúlt évtizedekben jelentős éghajlatváltozások figyelhetők meg, amelyek tanulmányozása a fizika törvényein alapuló, összetett számtani modelekkel történik. Az emberek, tetteikkel globálisan befolyásolják ezeket az éghajlatváltozásokat. Most megoldásokat kereshetsz a globális felmelegedés lassítására, interdiszciplináris projektek megvalosításával.

16. Lecke Ismétlés és felmérés

Most, hogy a VIII. osztály végére értél, készíts egy olyan projektet amelynek témája a **bolygónkat veszélyeztető éghajlatváltozás elleni küzdelem!**

Földrajzból megtanultad, hogy az éghajlat nem más mint a légkör több éves állapota egy nagyobb területen. Ezt az időjárási elemek: hőmérséklet, csapadék, szél- átlag értékei mutatják. Ha az időjárás változhat is egyik napról a másikra vagy akár egy napon belül is, az éghajlat hosszabb időn keresztül megőrzi a jellemzőit. Az éghajlatváltozásról rendelkezésünkre álló statisztikai adatok azt mutatják, hogy intézkedésekre van szükség a globális felmelegedés megfékezésére. A számítógépes adatfeldolgozás ezen a területen is hatékony lehet! Ugyancsak a számítógép segítségével pontos előrejelzéseket lehet készíteni és olyan modeleket kidolgozni amelyek szimulálják az éghajlatváltozást. A XXI. századra vonatkozó becslések szerint a globális felmelegedés **1,1°C** és **6,4°C** között lehet.

Dolgozz csapatban az osztálytársaiddal és készítsétek el a **Légy felelős, védd a természetet!** projektet.

TÁBLÁZATKEZELÉS

A Felelősök csapata az alábbi táblázatokkal dolgozik,amelyben az elmúlt három évben mért hőmérsékletre és csapadékmennyiségre vonatkozó adatok vannak. A csapat egy táblázatkezelő segítségével oldja meg a az alábbi követelményeket:

1. A munkafüzetben két munkalap van. A munkalapok neve: **hőmérséklet** és **csapadék**;
2. A **hőmérséklet** munkalapon készítsétek el az alábbi táblázatot **Éves átlag hőmérsékleti értékek, Celsius-fokban**;

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Hónap / Év	Január	Február	Március	Április	Május	Június	Július	Augusztus	Szeptember	Október	November	December
2	**1. év**	-8,9	0,7	5,9	13	17,7	21,6	23,5	26	18,9	9,5	12,4	8,7
3	**2. év**	-8	1	6,1	14,2	20	24,2	25	27	19,1	11	14,1	9
4	**3. év**	-7,2	0,6	5,4	12,7	18,1	22,3	24	25,8	17,8	10,6	13,4	8,3
5	**Legnagyobb**	-7,2	1	6,1	14,2	20	24,2	25	27	19,1	11	14,1	9
6	**Legkisebb**	-8,9	0,6	5,4	12,7	17,7	21,6	23,5	25,8	17,8	9,5	12,4	8,3

Éves átlag hőmérsékleti értékek, Celsius-fokban

3. A csapadék munkalapon készítsétek el az alábbi minta szerint a **Légköri cspadékmennyiség- havonta, milliméterben(mm) kifejezve** táblázatot. Változtassátok a **szegélyeket** és a **kitöltő színt**, a minta szerint.
4. Töltsétek ki a **B5:M5** és **B6:M6** cellatartományokat a **hőmérséklet** munkalapon, a megfelelő függvényeket használva.
5. Töltsétek ki a **B5:M5** cellatartományokat **csapadék** munkalapon, a megfelelő függvényt használva.
6. A **hőmérséklet** munkalapon készítsetek egy diagramot, amin csak a legnagyobb és a legkisebb értékeket ábrázoljátok.
7. A **csapadék** munkalapon, készítsetek egy diagramot ami a táblázat összes adatát ábrázolja.
8. Mentsétek a táblázatot az osztály mappájába **Idojaras.xlsx** néven.

Hónapok / Évek	Január	Február	Március	Április	Május	Június	Július	Augusztus	Szeptember	Október	November	December
Év1	18	1,4	53	44	94	75	4	24	45	152	2	41
Év2	18,7	1,6	43	37	80	34	6	10	32,4	98	10	56
Év3	23	15	58	40,1	89,5	48	5	16	23	74,3	16,8	42,8
Átlag												

Légköri csapadékmennyiség – havonta, milliméterben kifejezve (mm)

Ha ezeket mind elkészítettétek legyetek kreatívak használjátok a megtanultakat. **Legyetek felelősségteljesek!**

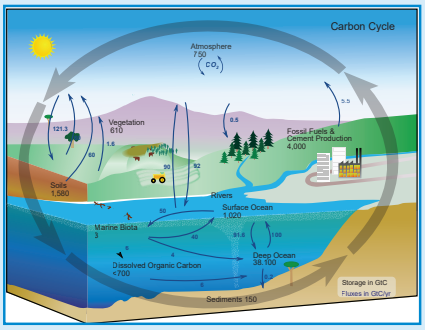
WEBLAPOK

A **Megfontoltak** csapata készít egy weblapot, hogy bemutassa a globális felmelegedés okait és hatásait. A csapat egy weblap szerkesztőt használ az alábbi követelmények megoldására:

LÉGY FELELŐSSÉGTELJES, VÉDD A TERMÉSZETET!

A globális felmelegedés az a jelenség amely a talaj közvetlen közelében, valamint az óceánok vízében mért átlagos hőmérséklet folyamatos emelkedésében nyilvánul meg. A jelenség az utóbbi két évszázadban jelent meg de az elmúlt évtizedekben különösen felerősödött

A gobális felmelegedést kiváltó okok:
♦ üvegház hatás ♦ növekvő gőznyomás ♦ az ózonréteg elvékonyodása ♦ a széndioxid koncentrációjának növekedése



Az évi átlaghőmérséklet növekedése a 2000 évhez képest

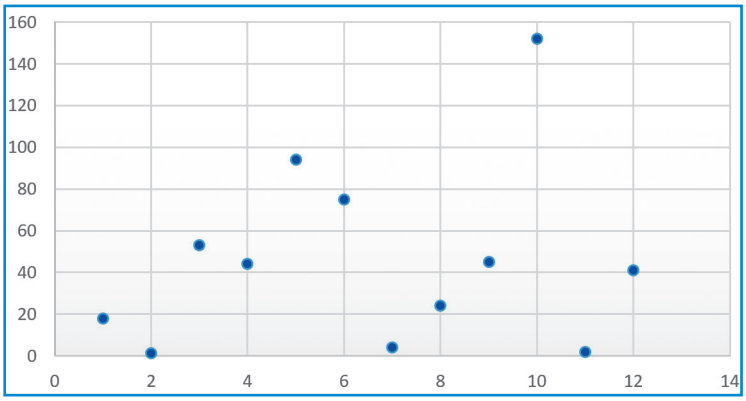
Év	2010	2015	2020
Hőmérséklet °C	0,48	0,52	0,57

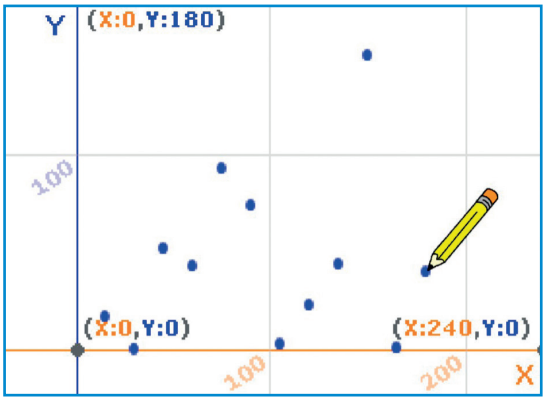
1. A weblap címe **A globális felmelegedés.**
2. A szöveget helyezük el úgy, ahogy a képen látod, két bekezdés különböző betűmérettel.
3. Az oldal bal és jobboldalán helyezzük el a képeket, a képek szélessége **480px** fekete **3px vastagságú keretben**. A képeket készíthetik a csapat tagjai vagy letölthetőek az internetről, azzal a megjegyzéssel, hogy a szerzői jogokat betartsuk.
4. A képeknek vannak címkéi. Ha az egeret az első képen görgetjük a „Szén körforgása a természetben”, ha a második képen görgetjük a „Környezetszennyezés” szövegek jelenjenek meg.
5. A két kép közé szúrjunk be egy 2 soros és 4 oszlopos táblázatot.
6. A táblázat szegélye legyen **kék színű és 3px** vastag.
7. Az oldal egyéb formázásait a csapat szabadon végezheti.
8. Mentsétek a Weblapot az osztály mappájába és **Legyetek megfontoltak**.

ALGORITMUSOK

A **Kíváncsiak** csapata egy hasonló diagramot fog készíteni, mint amilyen a **Felelősök** csapatáé, csak ők azt egy Scratch algoritmus segítségével oldják meg!

Kérjetek segítséget a **Felelősök** csapattól! Szükségetek van a **Idojaras.xlsx** munkafüzet **csapadék** munkalapján az **Év1**-ben tárolt csapadékértékekre. Pontosabban a **Felelősök** csapat tagjai kell elküldjék nektek a **Vremea.csv** állományt, amely egy munkalapot tartalmaz, egyetlen fejléc nélküli oszloppal, amely tartalmazza a 12 bejegyzést az **Év1**-nek megfelelően, vagy letölthetitek a **http://bit.ly/37fPYBq** címről. Íme, egy **Excel** (baloldalt) és egy **Scratch** (jobboldalt) diagram minta:

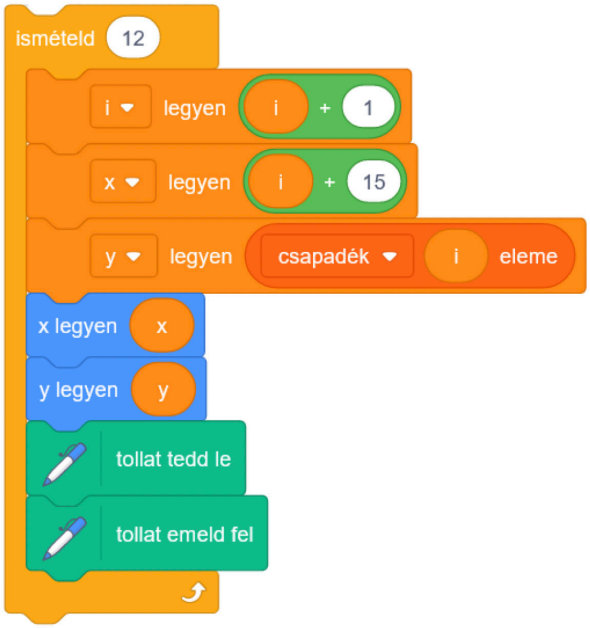




A következő algoritmus elkészíti **Scratch**-ban a grafikont. A **Kíváncsiak** csapata kiegészíti az algoritmust úgy, hogy megjelenítse az alábbiakat:

1. A **legkevesebb** csapadékkal rendelkező hónapot.
2. A **legnagyobb** csapadékértéket.
3. Az **éves átlag** csapadékmennyiséget.
4. Azokat az **egymást követő hónapokat**, ahol a legnagyobb a **csapadékmennyiségek különbsége**.

Ha ezeket megoldottátok, **legyetek kíváncsiak** és fedezzetek fel más algoritmusokat is!



A ROBOT

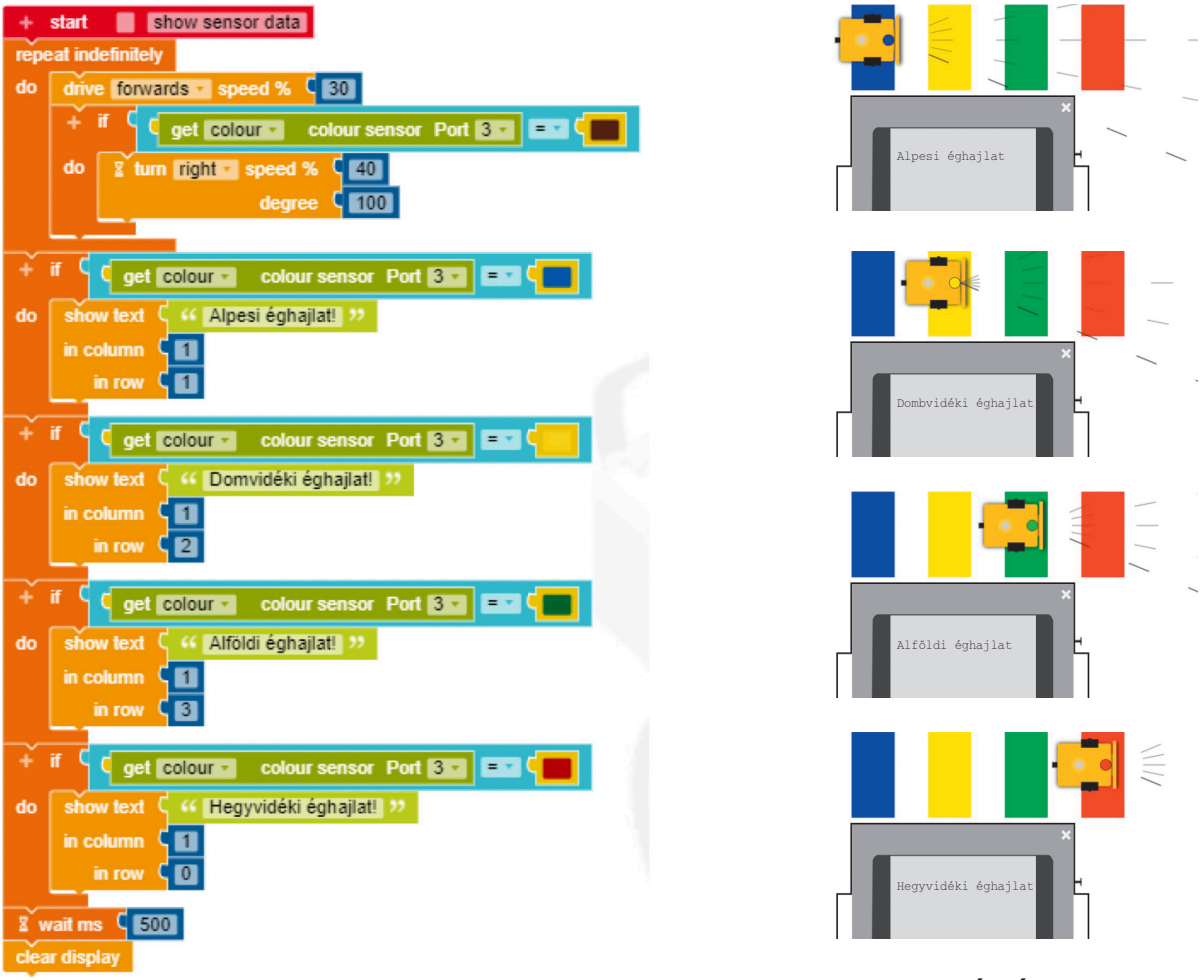
Az **Okosak** csapata kutatásokat végzett és felfedezte, hogy Románia éghajlata mérsékelt kontinentális. A felszíne változatos, lépcsőzetes beállítottságú, és a magasságok 0 m-től (a Fekete-tengernél) 2500 m-ig (Kárpátok) terjednek, ezáltal különböző éghajlati övezeteket határoznak meg.

Példa: T°ma = éves átlaghőmérséklet; csap = csapadék.

ÉGHAJLAT	T°ma (°C)	Csapadék (mm/év)
alpesi (2000 m felett**)**	**0** alatt	**1200** felett
hegyvidéki (**2000 m** alatt)	**0 – 6**	**800 – 1200**
medencei	**6 – 8**	**600 – 700**
dombvidéki	**6 – 10**	**600 – 800**
alföldi	**10 – 11**	**400 – 600**
tengerparti	**11** felett	**400** alatt

Nagyon valószínű, hogy a közeljövőben megjelenik majd egy olyan robot, amely hazánk területén egy bizonyos éghajlati övezetben valós időben méri a hőmérsékletet és a csapadék mennyiségét. Az **Open Roberta** programozási környezetben az Egyszerű helyszínen az EV3-as robot felkészült arra, hogy négy éghajlati övezetet fedezzen fel a pálya területén belül a következő kódolással: **kék** – alpesi éghajlat, **piros** – hegyvidéki éghajlat, **sárga** – dombvidéki éghajlat, **zöld** - alföldi éghajlat. Amikor az **EV3** eléri valamelyik színt, a képernyőre kiírja a megfelelő éghajlati övezetet.

Legyetek okosok, kutassatok, ültessétek gyakorlatba és védjétek a környezetet!



Az *Ismétlés* és a *Felmérés* után végre megérdemelsz egy KELLEMES VAKÁCIÓT!

Válaszok és magyarázatok

1. Lecke
7. oldal

Ismétlés: I.1. A; **2.** F; **3.** A; **4.** A. **II. 1.** elemek; **2.** csoportok; **3.** együttműködő; **4.** közösségi. **IV. 1.** A; **2.** A; **3.** A; **4.** F; **5.** F; **6.** A; **7.** A. **V. a.** Ctrl+S; **b.** Ctrl+N; **c.** F9/F5; **d.** Alt+F4. **VI. 1.** 420; **2.** 9968; **3.** Az algoritmus beolvassa az n természetes számot, meghatározza és kiírja az n páros számjegyeinek eltávolítása után kapott szám tükörszámát.
Értékelés: I.1. d; **2.** e; **3.** b; **4.** f; **5.** c ; **6.** a. **III. 1. a.** F; **b.** A; **c.** A; **d.** A. **2.** 2 a 2. hatványon, 3 az 1. hatványon, 5 a 2. hatványon. **3.** x az 5 nullától különböző hatványa **4.** Az x törzstényezőkre bontása.

2. Lecke
16. oldal

II. 1. c. **2. a.** A; **b.** F; **c.** A; **d.** A.

3. Lecke
20. oldal

II. 1. d. **2.** c. **3. a.** A; **b.** A; **c.** F; **d.** A.

4. Lecke
24. oldal

II. 1. b. **2. a.** A; **b.** F; **c.** F; **d.** F. **3. a.** képlet; **b.** előre meghatározott; **c.** érték; **d.** sorszám.
III. a. =MIN(B2:B9); **b.** =MAX(B10:B17); **c.** =AVERAGE(B18:B25);
d. =IF(B29<AVERAGE(B27:B28);"Da";"Nu")

5. Lecke
28. oldal

II. 1. d. **2.** c. **3. a.** A; **b.** A; **c.** F; **d.** A.

6. Lecke
36. oldal

II. 1. c. **2. a.** F; **b.** A; **c.** F; **d.** A.

7. Lecke
40. oldal

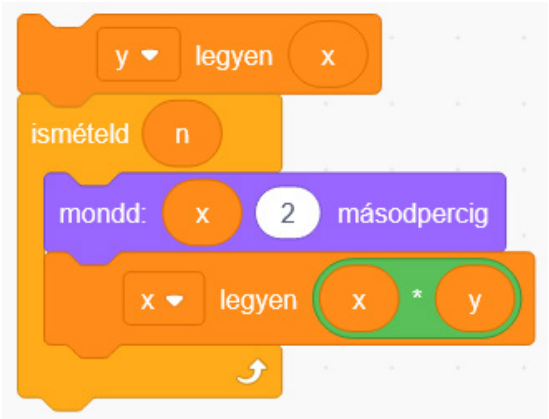
II. 1. d. **2. a.** F; **b.** A; **c.** F; **d.** F.

8. Lecke
44. oldal

II. 1. b. **2. a.** A; **b.** F; **c.** F; **d.** A.

9. Lecke
52. oldal

II.4.



III.2.

```
#include <iostream>
using namespace std;
int main()
{    int n,nr_p=0,nr_i=0;
     cin>>n;
     do
     {    if(n%2==0)
               nr_p++;
          else nr_i++;
          n=n/10;
     }while(n!=0);
     cout<<nr_p<<' '<<nr_i;
     return 0;
}
```

```
n=int(input())
nr_i=0
if n==0:
    nr_p=1
else:
    nr_p=0
while n!=0:
    if n%2==0:
        nr_p=nr_p+1
    else:
        nr_i=nr_i+1
    n=n//10
print(nr_p,nr_i)
```

III.3.

```cpp
#include <iostream>
using namespace std;
int main()
{ int n,i,db=0;
    cin>>n;
    for(i=n/2;i>=2;i--)
    { if(n%i==0)
       {cout<<i<<' ';db++;}
    }
    if(db==0) cout<<"Nem letezik";
    return 0;
}
```

```python
n=int(input())
db=0
for i in range(n//2,1,-1):
    if n%i==0:
        print(i,end=' ')
        db=db+1
if db==0:
    print('Nem letezik')
```

III.4.

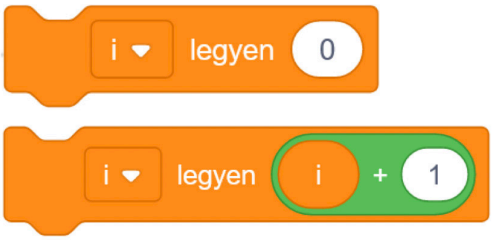
```cpp
#include <iostream>
using namespace std;
int main()
{int i,n,a,b,c;
 cin>>n>>a>>b;
 cout<<a<<' '<<b;
 for(i=3;i<=n;i++)
 {c=a+b;cout<<' '<<c;
  a=b;b=c;}
 return 0;
}
```

```python
n=int(input())
a=int(input())
b=int(input())
print(a,b,end=' ')
for i in range(3,n+1):
    c=a+b
    print(c,end=' ')
    a=b
    b=c
```

10. Lecke
56. oldal

II.3.

II. 3. Az **A minta** szerinti piramis számsorban:
– az **i** változó kezdeti értéke **1**.
– a fő vezérlő szerkezetben az **i** értéke **1**-gyel nő.



II. 3. A **B minta** szerinti piramis számsorban
– az **i** változó kezdeti értéke **n**.
– a fő vezérlő szerkezetben az **i** értéke **1**-gyel csökken.

III.1.

```
#include <iostream>
using namespace std;
int main()
{unsigned int i,n,s=0;
 cin>>n;
 for(i=1;i<=n;i++)
  {cout<<2*i<<' ';
   if((2*i)%6==0)
         s=s+2*i;
  }
 cout<<endl<<s;
 return 0;
}
```

```
n=int(input())
s=0
for i in range(1,n+1):
   print(2*i,end=' ')
   if (2*i)%6==0:
      s=s+2*i
print()
print(s)
```

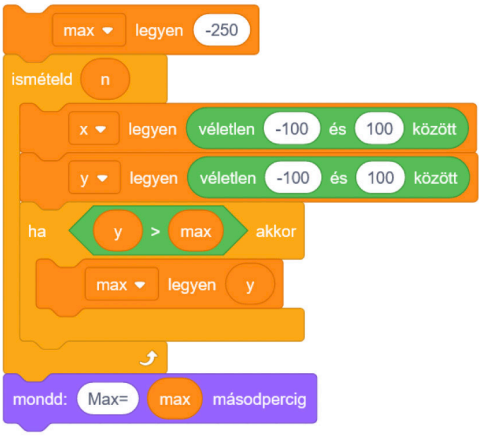
III.2.

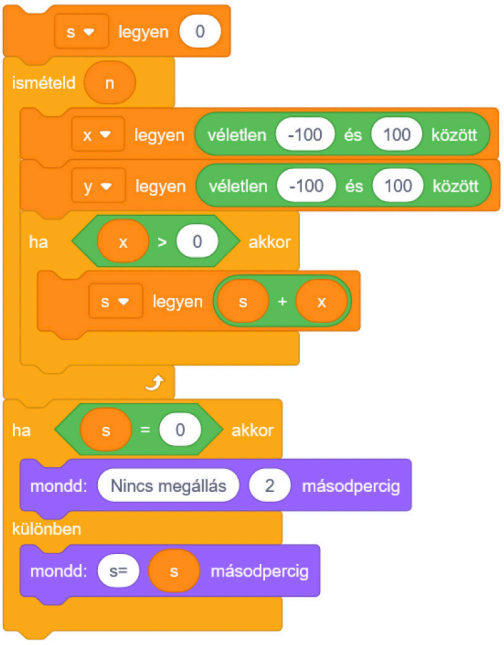
```
 for(i=1;i<=n;i++)
    {for(j=1;j<=i;j++)
      cout<<i<<' ';
     cout<<endl;
    }
 for(i=1;i<=n;i++)
    {for(j=1;j<=i;j++)
      cout<<j<<' ';
     cout<<endl;
    }
 for(i=n;i>=1;i--)
    {for(j=1;j<=i;j++)
      cout<<n-i+1<<' ';
     cout<<endl;
    }
  for(i=n;i>=1;i--)
    {for(j=1;j<=i;j++)
      cout<<i-j+1<<' ';
     cout<<endl;
    }
```

```
n=int(input())
for i in range(1,n+1):
   for j in range(1,i+1):
      print(i,end=' ')
   print()
for i in range(1,n+1):
   for j in range(1,i+1):
      print(j,end=' ')
   print()
for i in range(n,0,-1):
   for j in range(1,i+1):
      print(n-i+1,end=' ')
   print()
for i in range(n,0,-1):
   for j in range(1,i+1):
      print(i-j+1,end=' ')
   print()
```

11. Lecke
60. oldal

II.2.





III.2.

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,i,x,Max=0,db;
  cin>>n;
  for(i=1;i<=n;i++)
  { cin>>x;
    if(x==Max)db++;
    else
     if(x>Max && x%2!=0)
       {Max=x;
        db=1;
       }
  }
  if(Max==0) cout<<"Nem letezik";
  else cout<<Max<<' '<<db;
    return 0;
}
```

```
n=int(input())
Max=0
for i in range(1,n+1):
    x=int(input())
    if x==Max:
        db=db+1
    else:
        if x>Max and x%2!=0:
            Max=x
            db=1
if Max==0:
    print('Nem letezik')
else:
    print(Max,db)
```

12. Lecke
64. oldal

II. fibonacci.csv

	A	B	C
1	0	59	0.125
2	1	60	0.125
3	1	60	0.25
4	2	62	0.125
5	3	64	0.125
6	5	67	0.25
7	8	72	0.25
8	1	60	0.125
9	3	64	0.125
10	2	62	0.125
11	1	60	0.125
12	3	64	0.125

12	3	64	0.125
13	4	65	0.125
14	5	67	0.25
15	5	67	0.125
16	8	72	0.25
17	9	74	0.125
18	1	60	0.125
19	4	65	0.125
20	4	65	0.25
21	2	62	0.125
22	3	64	0.125
23	3	64	0.25

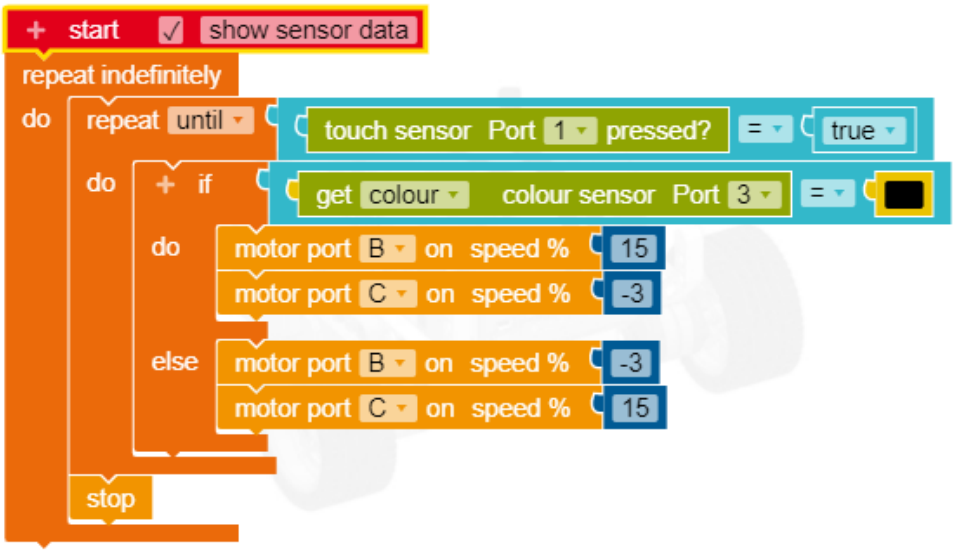
fibo	
1	0
2	1
3	1
4	2
5	3
6	5
7	8
8	1
9	3
10	2
11	1
12	3
+ hosszúság 23 =	

hangjegy	
1	59
2	60
3	60
4	62
5	64
6	67
7	72
8	60
9	64
10	62
11	60
12	64
+ hosszúság 23 =	

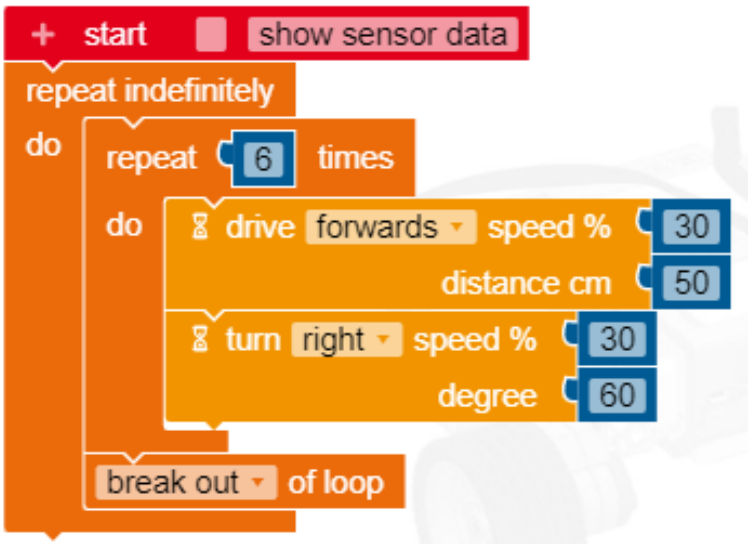
ütés	
1	0.125
2	0.125
3	0.25
4	0.125
5	0.125
6	0.25
7	0.25
8	0.125
9	0.125
10	0.125
11	0.125
12	0.125
+ hosszúság 23 =	

14. Lecke
76. oldal

III.



IV.



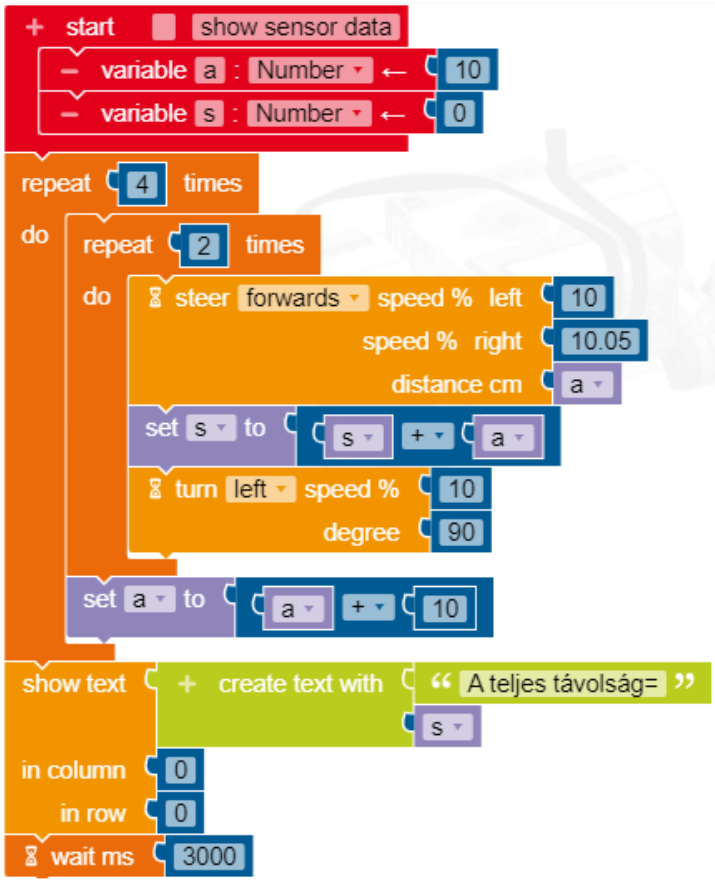
Felmérés
1. Egység
30. oldal

1. **Címsor** – az állomány nevét és az ablakkezelő gombokat tartalmazza.
   Állapotsor – Tartalmazza a dokumentum nézetvezérlő területét.
   Görgetősáv – azért van, hogy görgethessük, balra-jobbra, le-fel az ablak tartalmát.
   Szerkesztőléc – kiírja az aktuális cellában levő képletet, függvényt.
2. **a.** I; **b.** H; **c.** I; **d.** I.

Felmérés
2. Egység
46. oldal

1. **A <a> cimke** – hivatkozások beszurását teszi lehetővé.
   A <h1> </h1> cimke – legnagyobb méretű címsort hoz létre.
   A <th> </th> cimke – a táblázat első sorát fejlécként szerkeszti meg.
   Az <ol> cimke – sorszámozott listát készít.
2. **a.** I; **b.** H; **c.** H; **d.** H.

15. Lecke
80. oldal



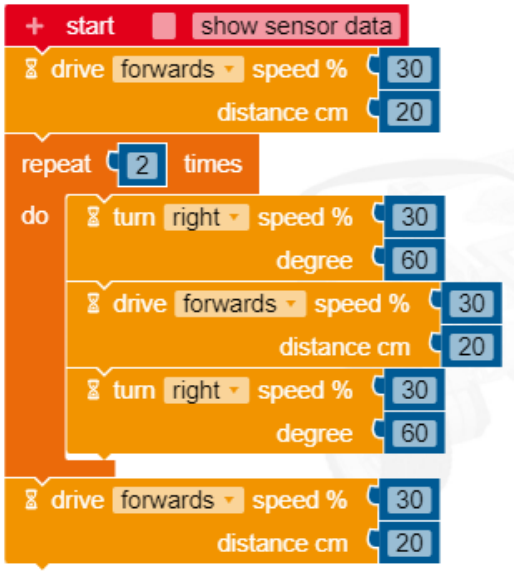
Felmérés
3. Egység
66. oldal

1. **Az összegző algoritmust használod**– amikor összeget akarsz számolni.
   A szorzat algoritmust – amikor szorzatot akarsz számolni.
   Használd a számláló algoritmust – amikor meg akarsz számolni valamit.
   A felcserélő algoritmust – amikor két változó tartalmát akarod felcserélni.
2. **a.** I; **b.** H; **c.** I; **d.** H.

Felmérés
4. Egység
82. oldal

1. **Az ultrahangos érzékelő** – megméri a tárgyakhoz képest a távolságot centiméterben.
   A színérzékelő – az érzékelő alatt található színeket érzékeli.
   Az érintés érzékelő – érzékeny az érintésre.
   A giroszkópos érzékelő – meghatározza egy bizonyos fokkal való elfordulást.

2.



Függelék – HTML szöveg-formázások

HTML	Hatás
formázás nélküli szöveg	formázás nélküli szöveg
félkövér szöveg	**félkövér szöveg**
dőltbetűs szöveg	*dőlt betűs szöveg*
<u>aláhúzott szöveg</u>	aláhúzott szöveg
~~áthúzott szöveg~~	~~áthúzott szöveg~~
^{felsőindex}	felsőindex
_{alsóindex}	alsóindex
<tt>ritkított betűk</tt>	ritkított betűk
<cite>idézet</cite>	*idézet*
kiemelt szöveg	*kiemelt szöveg*
vastag szöveg	**vastag szöveg**
<big>nagy szöveg</big>	nagy szöveg
kis szöveg	kis szöveg
piros betűk	piros betűk
Calibri betűk	Calibri betűk
font size=4	font size=4
középre igazított szöveg	középre igazított szöveg
balra igazított szöveg	balra igazított szöveg
jobbra igazított szöveg	jobbra igazított szöveg
1-es szintű címsor	**1-es szintű címsor**
2-es szintű címsor	**2-es szintű címsor**
3-as szintű címsor	**3-as szintű címsor**
4-es szintű címsor	**4-es szintű címsor**
5-s szintű címsor	5-ös szintű címsor
6-s szintű címsor	6-os szintű címsor

Függelék – HTML színkódok

Szín megnevezése	színkód	szín
White	#FFFFFF	
Silver	#C0C0C0	
Gray	#808080	
Black	#000000	
Maroon	#800000	
Firebrick	#B22222	
Red	#FF0000	
LightCoral	#F08080	
Chocolate	#D2691E	
Orange	#FFA500	
Gold	#FFD700	
Yellow	#FFFF00	
LightGreen	#90EE90	
Lime	#00FF00	
Olive	#808000	
Green	#008000	
SeaGreen	#2E8B57	
Teal	#008080	
SkyBlue	#87CEEB	
Aqua	#00FFFF	
Blue	#0000FF	
Navy	#000080	
Indigo	#4B0082	
BlueViolet	#8A2BE2	
Purple	#800080	
Fuchsia	#FF00FF	
Pink	#FFC0CB	

Az **Informatikában és IKT**-ben használt kifejezések szótára

Angol	Magyar
account	fiók
align	igazítás
back	vissza
background	háttér
body	test
bold	félkövér
border	szegély
bottom	lent
button	nyomógomb
caption	felirat
cell	cella
center	közép
color	szín
column	oszlop
copy	másolás
cut	kivágás
date	dátum
delete	törlés
display	kijelző
draft	piszkozat
file	állomány
fill	kitöltés
filter	szűrés
font	betű
for	minden
form	űrlap
formula	képlet
forward	tovább
height	magasság
hyperlink	hivatkozás
image	kép
insert	beszúrás
italic	dőlt
layout	elrendezés
left	bal

Angol	Magyar
main	fő
menu	menü
open	megnyit
page layout	oldal beállítás
paragraph	bekezdés
paste	beillesztés
picture	kép
previous	előző
print	nyomtatás
random	véletlenszerű
replace	helyettesítés
review	felülvizsgálat
ribbon	szalag
right	jobb
row	sor
save	mentés
save as	mentés másként
search	keresés
selected	kiválasztott
sheet	munkalap
show	mutasd
size	méret
size	méret
sort	rendez
standard	standard
style	stílus
tag	címke
title	cím
top	fent
type	típus
underline	aláhúzott
until	ameddig
while	amíg
width	szélesség
window	ablak

Programa școlară poate fi accesată la adresa:
http://programe.ise.ro.

A tankönyvnek nyomtatott és digitális változata van.
A digitális változat a nyomtatottal azonos tartalmú.
Továbbá tartalmaz egy sor interaktív, multimédiás tanulási tevékenységet (interaktív gyakorlatokat, oktató játékokat, animációkat, filmeket, szimulációkat).

Tudásunk egy csepp. Amit nem tudunk, az egy egész óceán.

Isaac Newton

Tradiție din 1989
www.litera.ro
ISBN 978-606-33-5476-2

LITERA