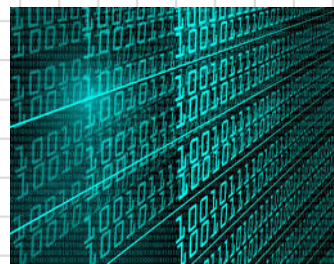




ÓRATERVEK



PROGRAMOZÁS ÉS KÓDOLÁS A SZAKOKTATÁSBAN

2023-2-SK01-KA210-VET-000184496 2024. 01. 01. – 31. 12. 2025

CNC

Az Európai Unió által finanszírozott. A kinyilvánított nézetek és vélemények azonban csak a szerző(k)é, és nem feltétlenül tükrözik az Európai Unió vagy az Európai Oktatási és Szlovák Akadémiai Szövetség a Nemzetközi Együttműködésért (SAAIC) véleményét. Sem az Európai Unió, sem az SAAIC nem tehető felelőssé értük.



Funded by
the European Union



ÓRATERV



TÁRGY: PROGRAMOZÁS

ÓRA TÉMA: PROGRAMOZÁS PYTHONBAN – GEOMETRIAI ALAKZATOK RAJZOLÁSA.

TANULÁSI CÉLKITÜZÉSEK: 1. SAJÁTÍTSA EL A PYTHON PROGRAMOZÁSI ALAPISMERETEKET A TKINTER GRAFIKUS KÖNYVTÁR HASZNÁLATÁVAL.
2. MEGTANULNI DOLGOZNI GRAFIKUS VÁSZONNAL. RAJZOLJ RÁ EGYSZERŰ FORMÁKAT.
3. HOZZON LÉTRE KÉPEKET VÉLETLENSZERŰ HELYEN A TKINTER KÖNYVTÁR SEGÍTSÉGÉVEL.

ESZKÖZÖK:
EGY SZÁMÍTÓGÉP PYTHONNAL ÉS A TKINTER KÖNYVTÁRRAL.
PYTHON MINTA TANKÖNYV MUNKALAP GYAKORLATOKKAL

HOOK/BEVEZETÉS: MEGMUTATJUK, HOGYAN KELL ELKÉSZÍTENI ÉS FUTTATNI AZ ELSŐ PROGRAMOT, KÉSZÍTENI EGY GRAFIKUS VÁSZNAT, AMELYRE GRAFIKUS ALAKZATOKAT FOGUNK RAJZOLNI. VÁLTOZÓT HASZNÁLUNK, VÉLETLENSZERŰ ÉRTÉKEKET GENERÁLUNK, ÉS KÉPEKET RAJZOLUNK VÉLETLENSZERŰ HELYRE A TKINTER KÖNYVTÁR SEGÍTSÉGÉVEL.

ÓRA FELÉPÍTÉSE: ELKÖTELEZŐDÉS (4 PERC) – ELEMI KÉP RAJZOLÁSA DERÉKSZÖGŰ KOORDINÁTÁKKAL INTERJÚ TANULÓKKAL A KOORDINÁTARENDSZERRŐL A TKINTER MODULBAN (1. FELADAT)

- **KUTATÁS (12 PERC) – ÚJ PARANCSSOR KERESÉSE (2-4. FELADAT)**
- **PREZENTÁCIÓ (15 PERC) – ISMERETEK ÖSSZEFOGLALÁSA ÉS MAGYARÁZATA A GRAFIKUS FELHASZNÁLÓI FELÜLETTEL RENDELKEZŐ PROGRAMOK KÉSZÍTÉSÉNEK KÉRDÉSKÖRÉNEK KUTATÁSA CÍMŰ RÉSZBŐL**
- **FEJLESZTÉS (9 PERC) – IGÉNYESEBB FELADATOK ÖNÁLLÓ PROGRAMOZÁSA (5-6. FELADATOK)**
- **ÉRTÉKELÉS (5 PERC) – AZ ÖNÉRTÉKELŐ TESZT MEGOLDÁSA, A VÁLASZOK MEGBESZÉLÉSE.**

A PYTHON PROGRAMOZÁS RÖVID ÁTTEKINTÉSE (KEZDŐKNEK).

A TKINTER KÖNYVTÁR TELEPÍTÉSE (HA NINCS TELEPÍTVE).

A TKINTER KÖNYVTÁR BEMUTATÁSA.

A VÁSZONRA RAJZOLÁS ELVÉNEK MAGYARÁZATA.

FELADATLAP A DIÁKNAK

A PROGRAMOZÁS PYTHONBAN ELEKTRONIKUS TANKÖNYVÜNK SZERINT JÁRUNK EL, AMELY KÖZÉPISKOLÁK SZÁMÁRA KÉSZÜLT. TANKÖNYVMINTÁT TALÁLHAT A KÖVETKEZŐ CÍMEN:

[HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf) A MINTA EGYSZERŰ RAJZPÉLDÁKAT TARTALMAZ:

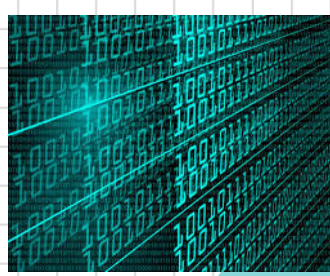
1. ALAPFORMÁK RAJZOLÁSA:

KÖR, TÉGLALAP, NÉGYZET ÉS HÁROMSZÖG RAJZOLÁSA KÜLÖNBÖZŐ SZÍNEKKEL ÉS VONALVASTAGSÁGGAL
KÓDPÉLDA AZ EGYES ALAKZATOKHOZ EXPOZÍCIÓS FÁZIS – ÚJ OBJEKTUM EXPOZÍCIÓJA

– AZ ALAPFOGALMAKAT RÖVIDEN MEGJEGYZEM A TÁBLÁN

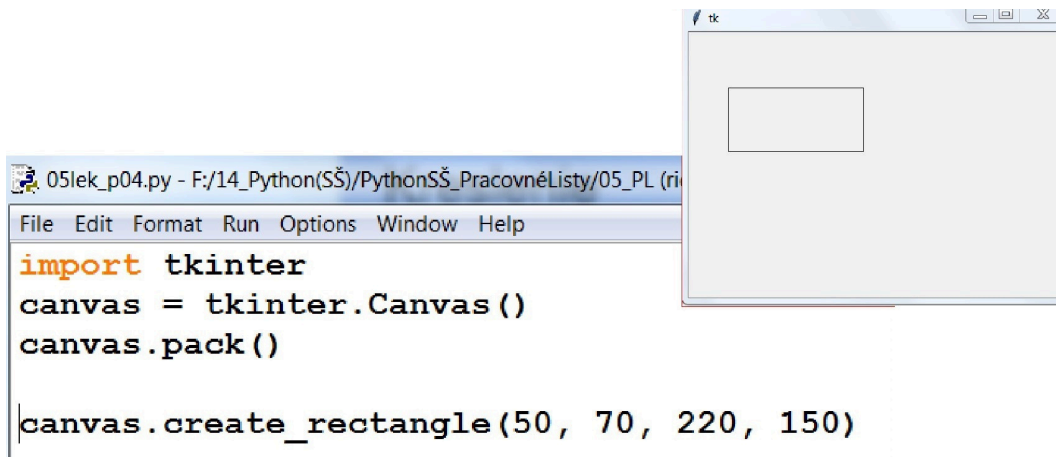


ÓRATERV



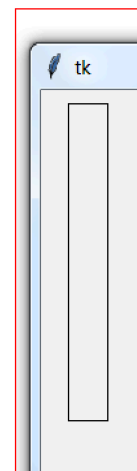
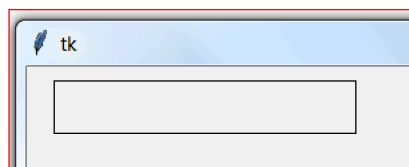
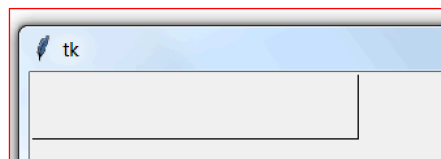
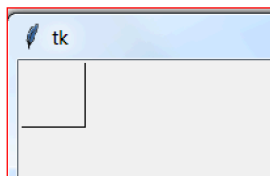
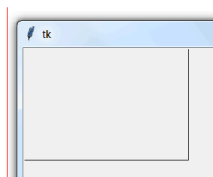
FELADATLAP A DIÁKNAK

1. Feladat: Hozza létre a 05lek_p04.py fájlt, a benne felsorolt parancsokat, és futtassa az xxxxx programot:



2. Feladat: A `canvas.create_rectangle(,,,)` parancs zárójelében 4 szám található. Próbálja meg fokozatosan módosítani őket a 05lek_p05.py programban (az előző bejegyzéshez képesti változtatások sárga színnel vannak kiemelve). Futtassa a programot minden alkalommal, hogy megnézze, mit rajzol!

- a) `canvas.create_rectangle(0, 0, 220, 150)`
- b) `canvas.create_rectangle(0, 0, 50, 50)`
- c) `canvas.create_rectangle(0, 0, 250, 50)`
- d) `canvas.create_rectangle(20, 10, 250, 50)`
- e) `canvas.create_rectangle(20, 10, 50, 250)`





ÓRATERV



1.

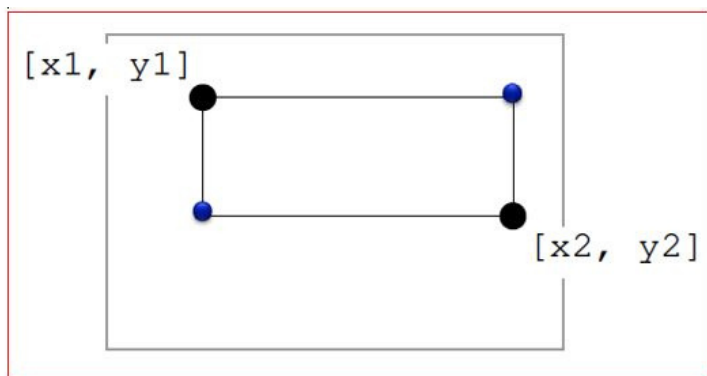
FELADATLAP A DIÁKNAK

1. 3. Feladat: A számítógépen a bal felső sarokban van egy $[0, 0]$ koordinátájú pont. Az x tengely balról jobbra halad. Az y tengely a matematikához képest fordított - fentről lefelé halad (minél nagyobb a szám, annál alacsonyabb).



1.
2.

4. Feladat: A `canvas.create_rectangle(x1, y1, x2, y2)` parancsban zárójelbe írjuk a téglalap ellentétes csúcsainak koordinátáit:



1. 5. Feladat: Változtassa meg a `05lek_p06.py` programját úgy, hogy rajzoljon egy téglalapot, amelynek az $[50, 30]$ és $[300, 200]$ ellentétes csúcsainak koordinátái vannak.

```
*05lek_p05.py - F:/14_Python(SŠ)/PythonSŠ_PracovnéListy/05_PL (riešenia)/05lek_p05.py
File Edit Format Run Options Window Help
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

canvas.create_rectangle(50, 30, 300, 200)
```

Šírka: $300 - 50 = 250$

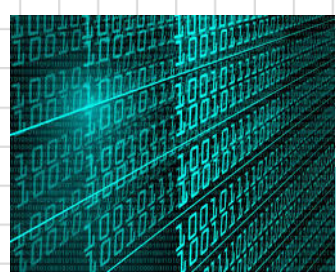
Výška: $200 - 30 = 170$

1. 6. Feladat: Számítsa ki a téglalap szélességét és magasságát az előző példából számítógép használata nélkül.

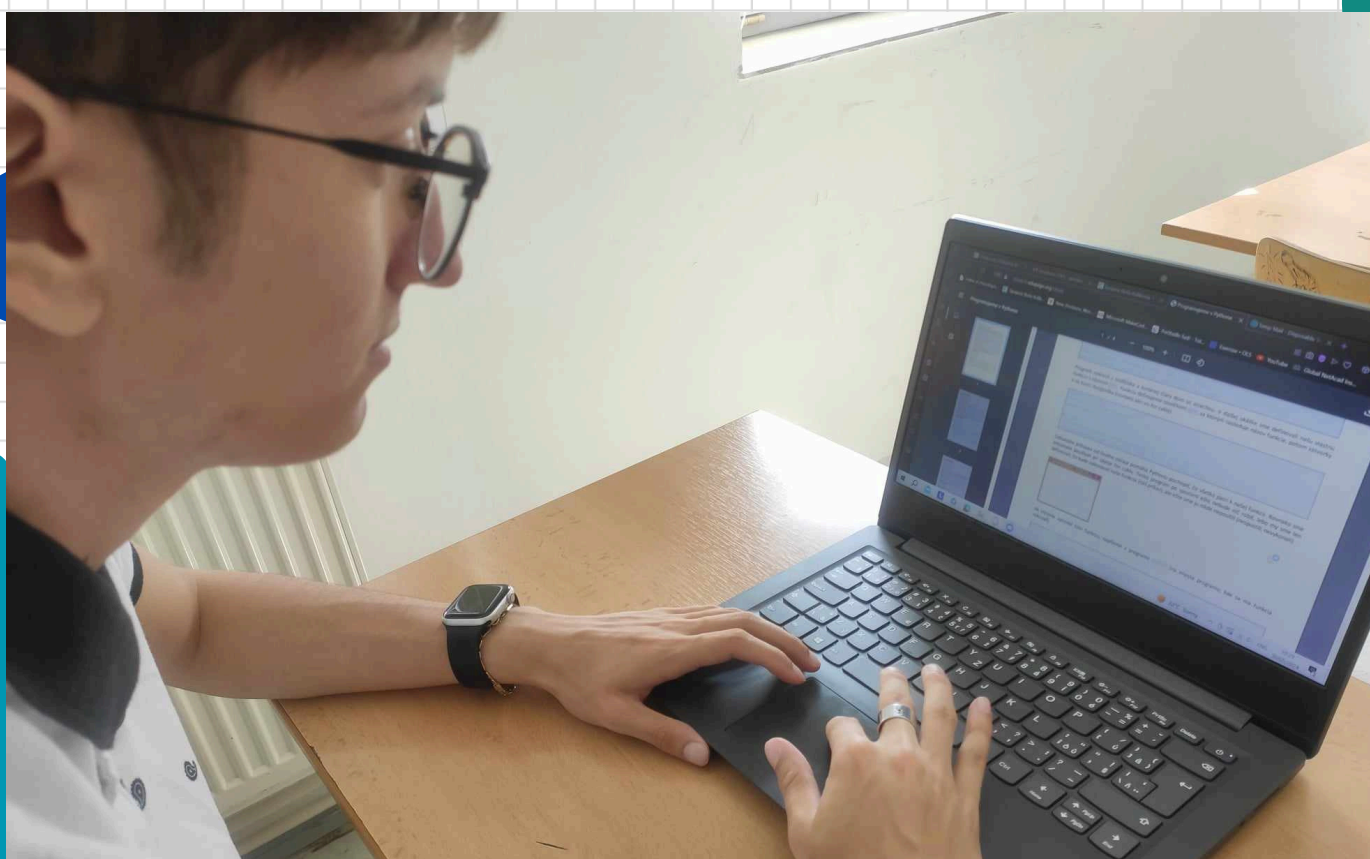
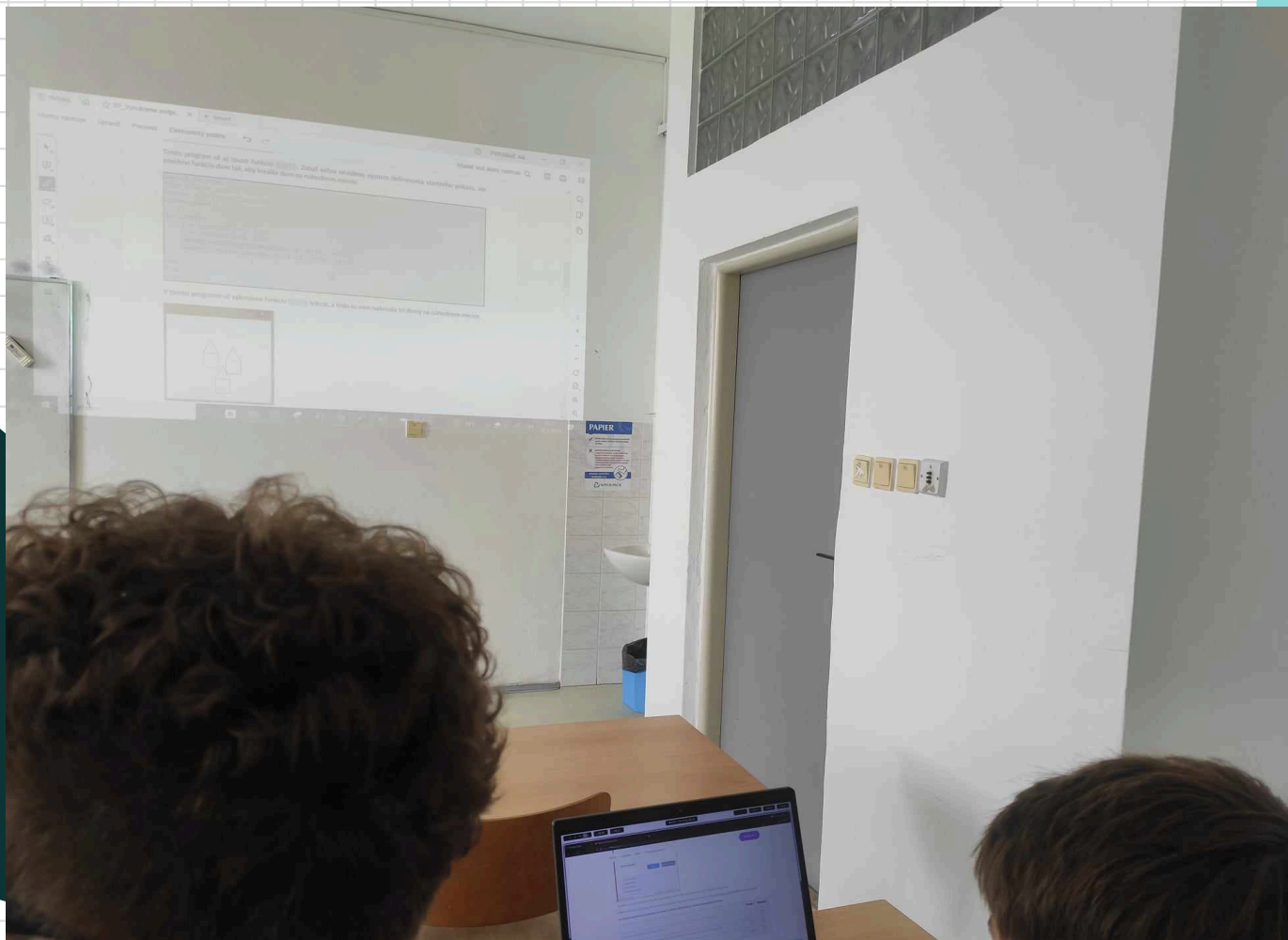
1. 7. Feladat: Hozzon létre egy új `05lek_p08.py` programot, és rajzoljon egy téglalapot, amelynek egy csúcsa van a $[200, 100]$ koordinátákon, szélessége 60, magassága 140.



MELLÉKLET

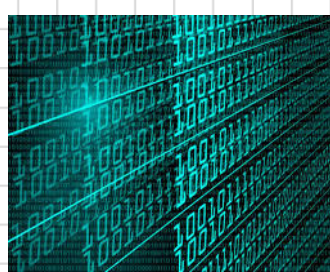


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL

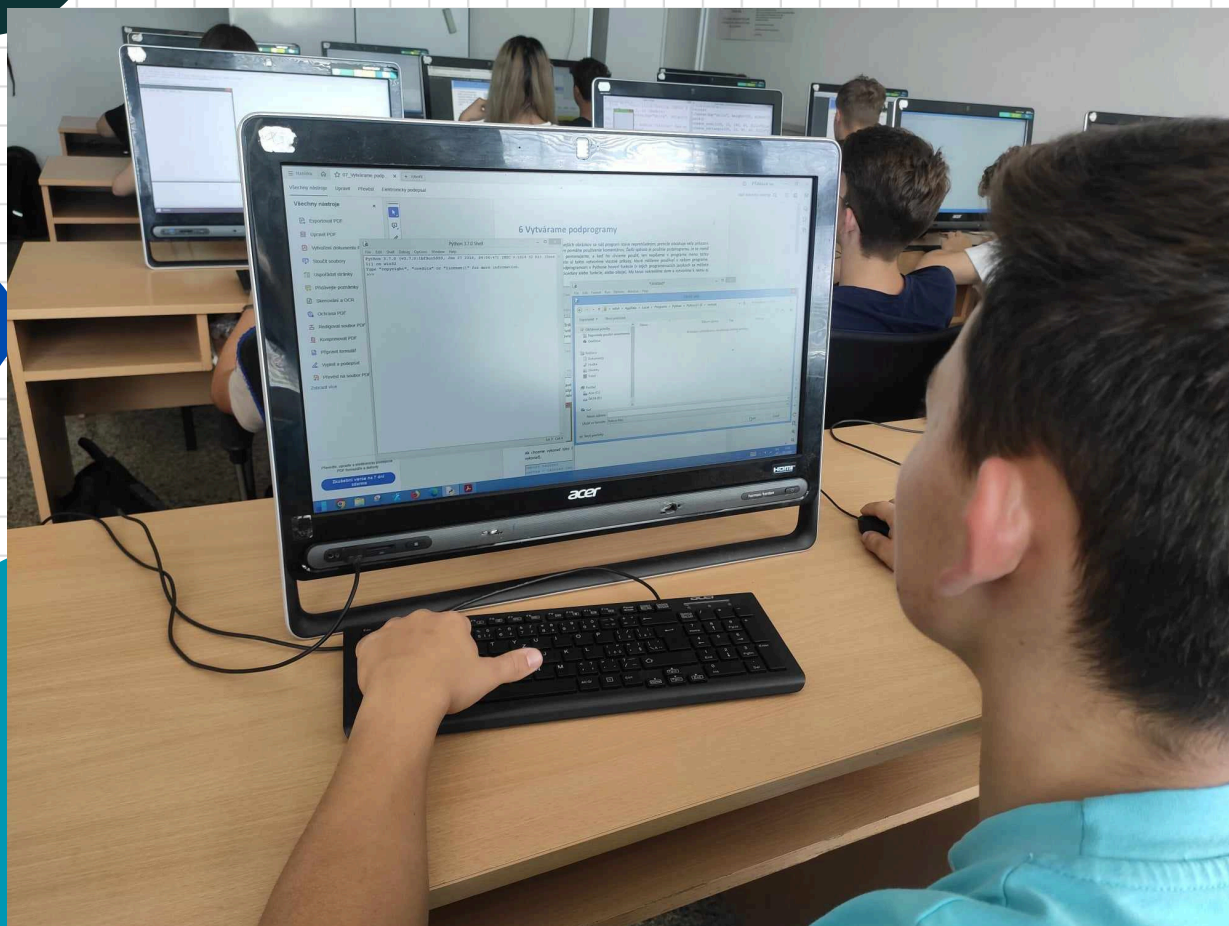
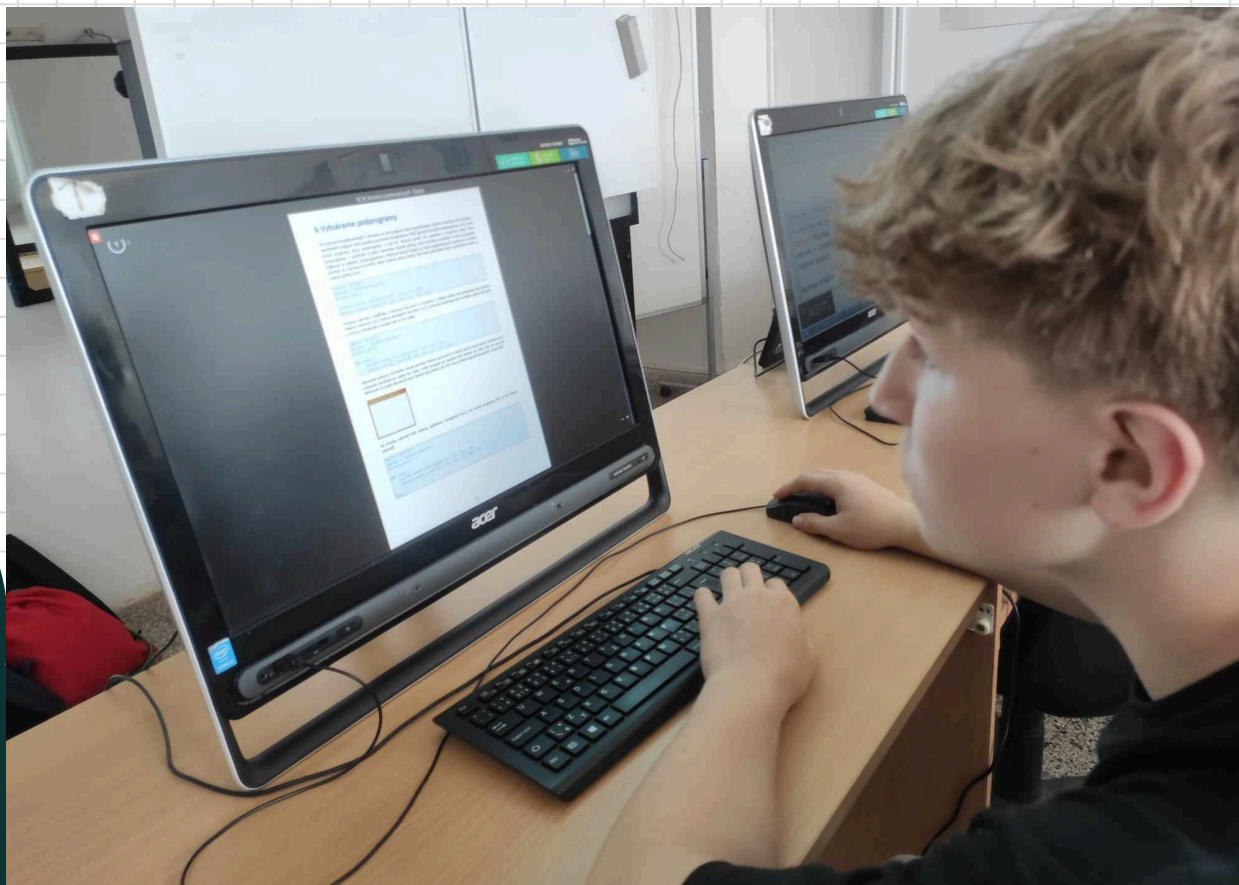




MELLÉKLET



FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL

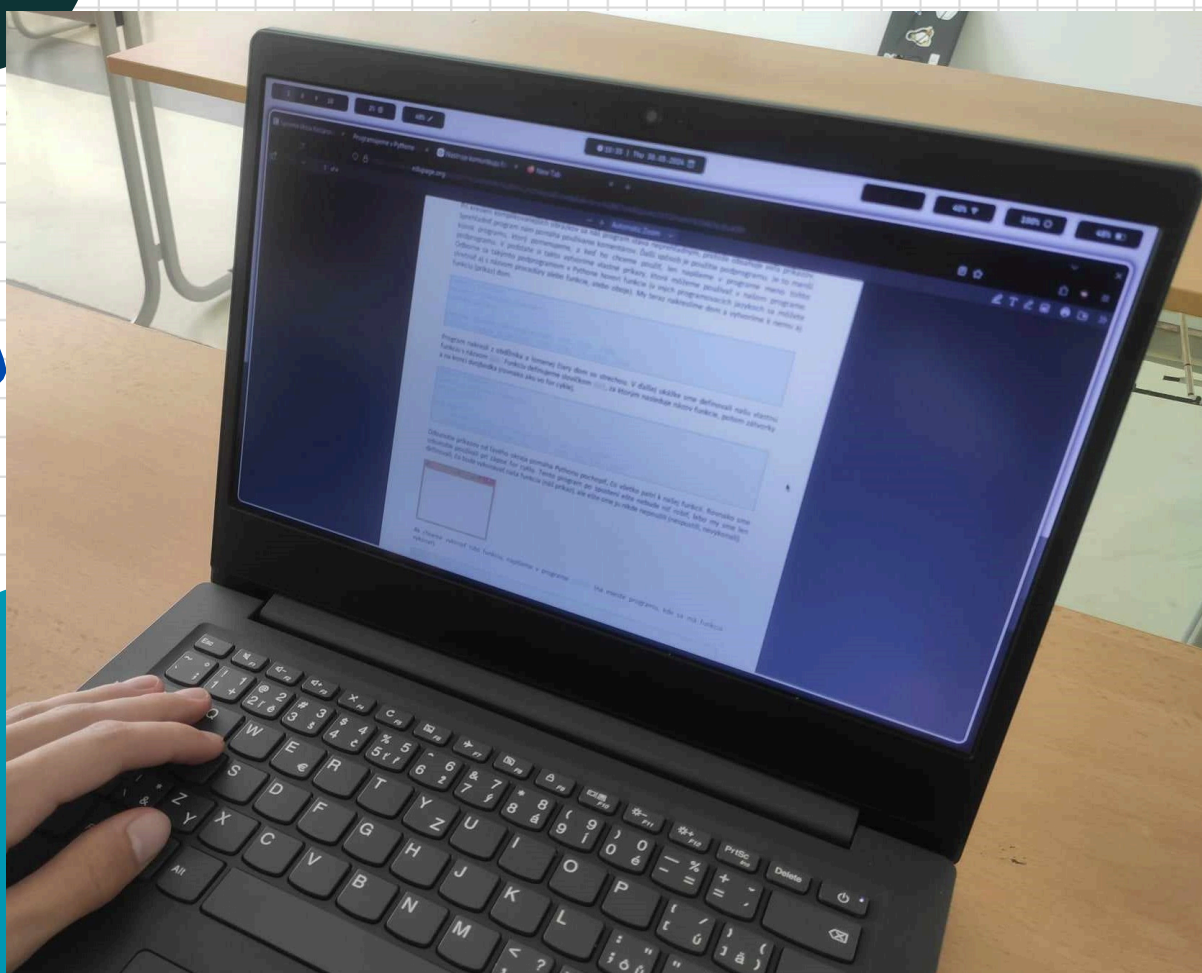
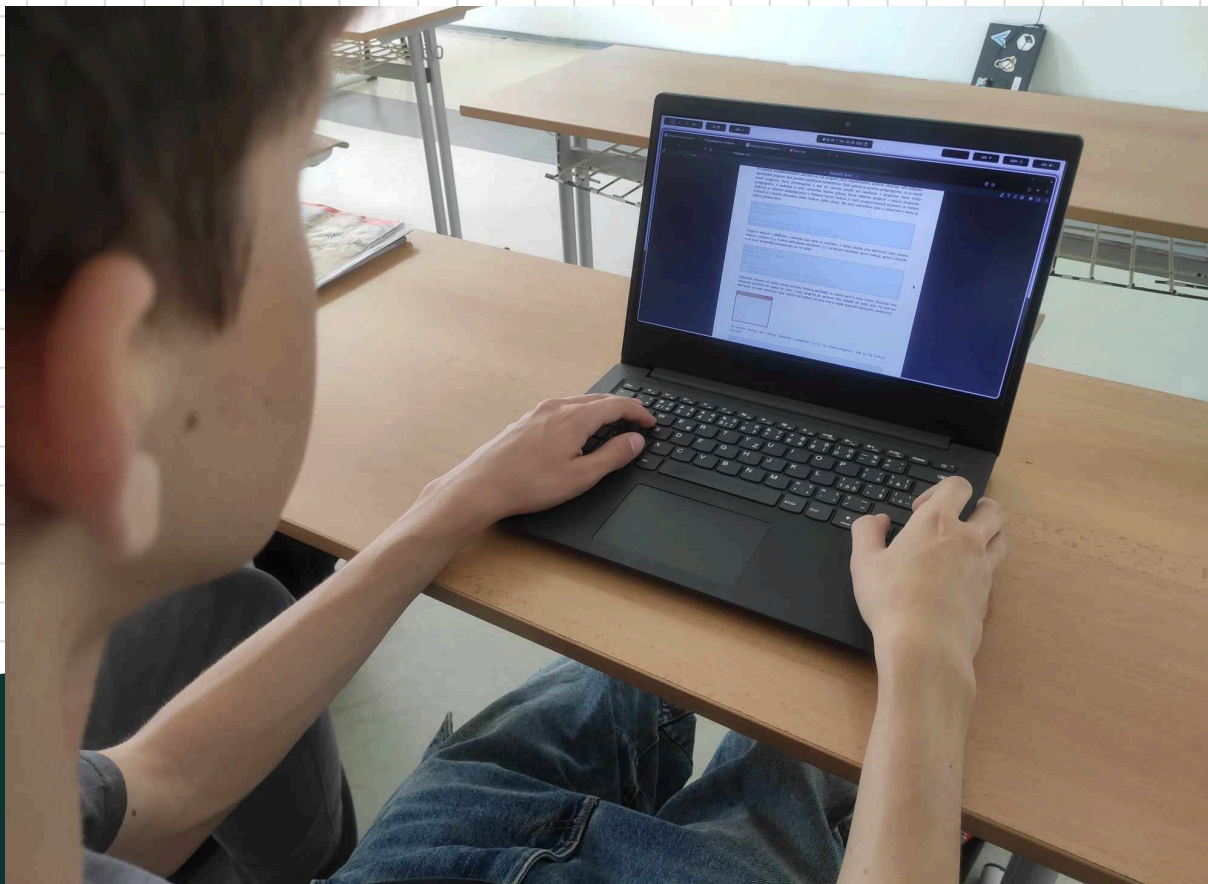




MELLÉKLET

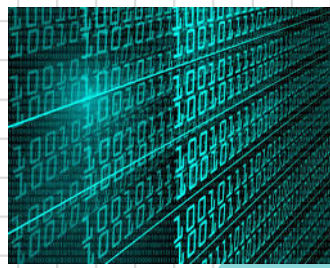


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL





MELLÉKLET



DIÁKOK KIMENETE

```
FROM TKINTER IMPORT * TKINTER
ABLAK LÉTREHOZÁSA ROOT = TK()
VÁSZON LÉTREHOZÁSA VÁSZONRA
RAJZOLÁSHOZ = CANVAS(GYÖKÉR,
SZÉLESSÉG=300, MAGASSÁG=300)
CANVAS.PACK() NÉGYZET ALAKÚ
VÁSZON
LÉTREHOZÁSA.CREATE_RECTANGLE(
50, 50, 80 , 80, FILL="BLUE")
TÉGLALAP LÉTREHOZÁSA
CANVAS.CREATE_RECTANGLE(100,
100, 160, 140, FILL="RED") FUTTASSA
A TKINTER FŐHURKOT
ROOT.MAINLOOP()
```




ÓRATERV



TÁRGY: PROGRAMOZÁS

ÓRA TÉMA: PROGRAMOZÁS PYTHONBAN – RAJZZÁS ÉS SZÍNEK.

TANULÁSI CÉLKITÜZÉSEK:

- 1. ISMERJE MEG A PYTHONBAN VALÓ RAJZOLÁS ALAPELVEIT A TKINTER KÖNYVTÁR HASZNÁLATÁVAL
- 2. TANULJA MEG A KÜLÖNBÖZŐ SZÍNEK ÉS EFFEKTUSOK HASZNÁLATÁT PYTHONBAN VALÓ RAJZOLÁS SORÁN.

ESZKÖZÖK:

**EGY SZÁMÍTÓGÉP PYTHONNAL ÉS A TKINTER KÖNYVTÁRRAL.
PYTHON MINTA TANKÖNYV MUNKALAP
GYAKORLATOKKAL**

**HOOK/BEVEZETÉS: KÜLÖNBÖZŐ SZÍNEK
HASZNÁLATA RAJZOLÁSKOR:**

**SZÍN MEGHATÁROZÁSA RGB ÉRTÉKEK
SEGÍTSÉGÉVEL**

**KÜLÖNBÖZŐ SZÍNEK HASZNÁLATA AZ EGYES
RAJZOBJEKTUMOKHOZ**

- HATÁSOK LÉTREHOZÁSA
SZÍNKOMBINÁCIÓK SEGÍTSÉGÉVEL

**AZ ÓRA FELÉPÍTÉSE: ELKÖTELEZŐDÉS (4 PERC) – EGYSZERŰ KÉP LÉTREHOZÁSA PYTHON RAJZ
SEGÍTSÉGÉVEL SZÍNEK ÉS EFFEKTUSOK SEGÍTSÉGÉVEL
BESZÉLGETÉS TANULÓKKAL A PYTHONBAN A TKINTER KÖNYVTÁR SEGÍTSÉGÉVEL TÖRTÉNŐ
RAJZOLÁS ÉS SZÍNEKKEL VALÓ MUNKA ALAPELVEIRŐL**

- KUTATÁS (12 PERC) – AZ ÚJ KÉPKITÖLTÉS ÉS OUTLINE PARANCSONK Felfedezése
- PREZENTÁCIÓ (15 PERC) – ISMERETEK ÖSSZEFOGLALÁSA ÉS MAGYARÁZATA A PYTHON
ELABORATION RAJZOLÁS ÉS SZÍNEKKEL VALÓ MUNKA ALAPELVEIBŐL PROGRAMKÉSZÍTÉS
KÉRDÉSKÖRÉNEK VIZSGÁLATA CÍMŰ RÉSZBŐL (9 PERC) – A TANULÓK TÁMOGATÁSA A TOVÁBBI
ÖNÁLLÓ RAJZOLÁSBAN ÉS A PROGRAM LEHETŐSÉGEINEK FELFEDEZÉSÉBEN. PROGRAMOZÁS
PYTHONBAN SZÍNEK HASZNÁLATÁVAL
- ÉRTÉKELÉS (5 PERC) – AZ ÖNÉRTÉKELŐ TESZT MEGOLDÁSA, A VÁLASZOK MEGBESZÉLÉSE.

FELADATLAP A DIÁKNAK

**1. A PROGRAMOZÁS PYTHONBAN ELEKTRONIKUS TANKÖNYVÜNK SZERINT JÁRUNK EL, AMELY
KÖZÉPISKOLÁK SZÁMÁRA KÉSZÜLT. TANKÖNYVMINTÁT TALÁLHAT A KÖVETKEZŐ CÍMEN:
[HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf) A MINTA EGYSZERŰ PÉLDÁKAT
TARTALMAZ A SZÍNES RAJZOLÁSRA:**

1. FORMA TÖLTTELÉK:

**KÖR, TÉGLALAP, NÉGYZET ÉS HÁROMSZÖG OUTLINE KÜLÖNBÖZŐ SZÍNEKKEL ÉS
VONALVASTAGSÁGGAL KÓDMINTA KITÖLTÉSHEZ ÉS OUTLINE EXPOZÍCIÓS FÁZIS – ÚJ PARANCSONK
EXPOZÍCIÓJA**

– AZ ALAPFOGALMAKAT RÖVIDEN MEGJEGYZEM A TÁBLÁN



ÓRATERV

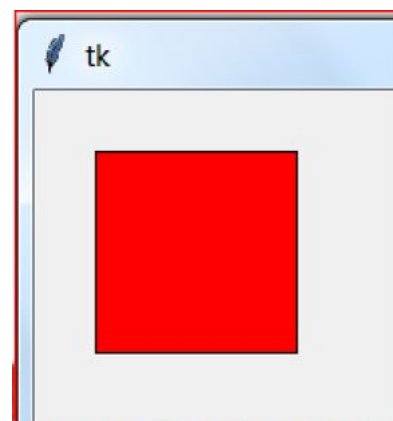


2. TANULÓI FELADATLAP

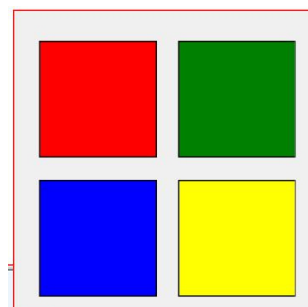
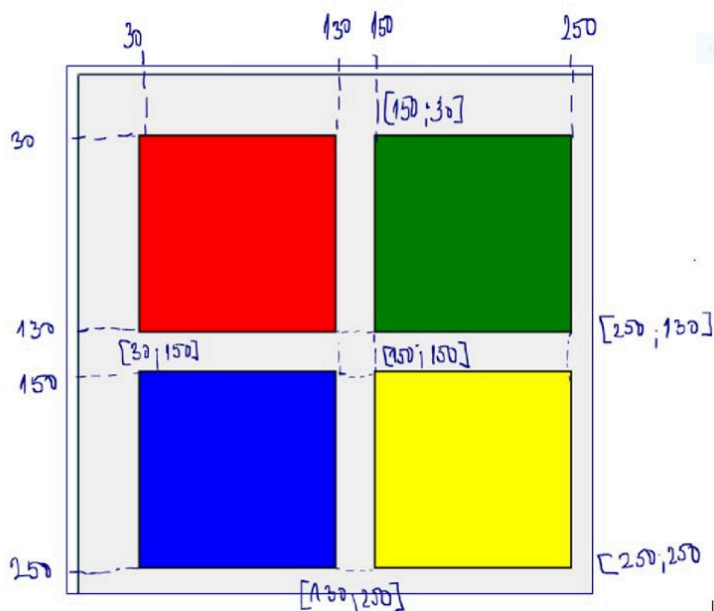
1. Feladat: Hozzon létre egy új 06lek_p02.py programot, rajzoljon egy színes téglalapot a következő kóddal:

```
*06lek_p02.py - F:/14_Python(SŠ)/PythonSŠ_PracovnéListy/06_PL (riešenia)/06lek_p02.py (3.6.2rc1)*  
File Edit Format Run Options Window Help  
import tkinter  
canvas = tkinter.Canvas()  
canvas.pack()  
  
canvas.create_rectangle(30, 30, 130, 130, fill='red')
```

A négyzet szélessége (és magassága) 100

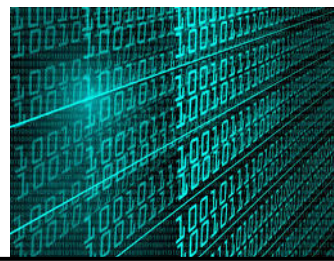


2. Feladat: Adjon hozzá további 3 parancsot téglalapok rajzolásához a 06lek_p03.py programhoz, hogy a következő képet kapja:
A négyzet szélessége (és magassága) 100. A négyzetek közötti távolság 20





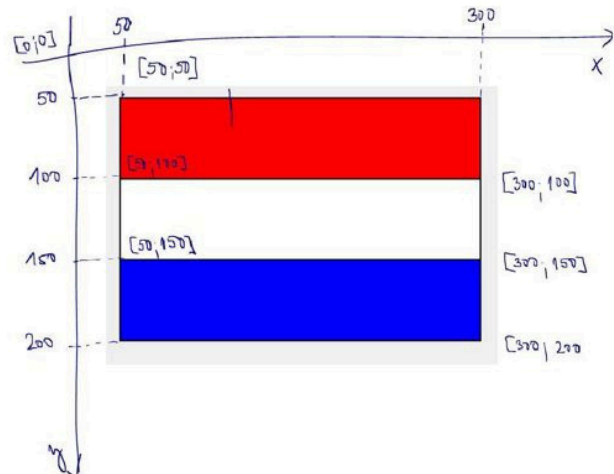
ÓRATERV



```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

canvas.create_rectangle(30, 30, 130, 130, fill='red')
canvas.create_rectangle(150, 30, 250, 130, fill='green')
canvas.create_rectangle(30, 150, 130, 250, fill='blue')
canvas.create_rectangle(150, 150, 250, 250, fill='yellow')
```

1. 3. Feladat: Hozzon létre egy új 06lek_p04.py programot, amely megrajzolja a holland zászlót:



```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
```

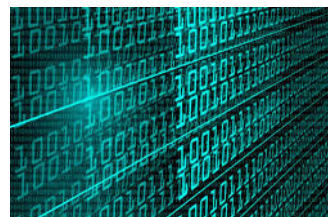
```
canvas.create_rectangle(50, 50, 300, 100, fill='red')
canvas.create_rectangle(50, 100, 300, 150, fill='white')
canvas.create_rectangle(50, 150, 300, 200, fill='blue')
```

1. 4. Feladat: Hozzon létre egy 06lek_p05.py programot, amely megrajzolja az ír zászlót "narancssárga" színnel:



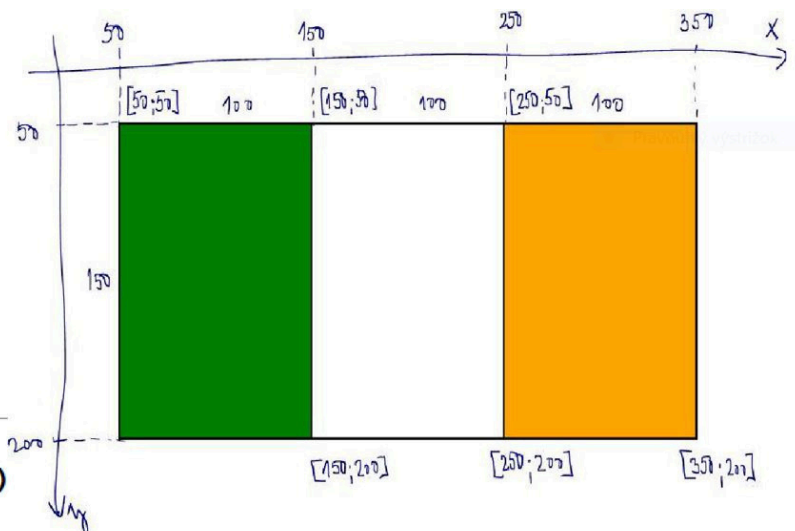


ÓRATERV



1.

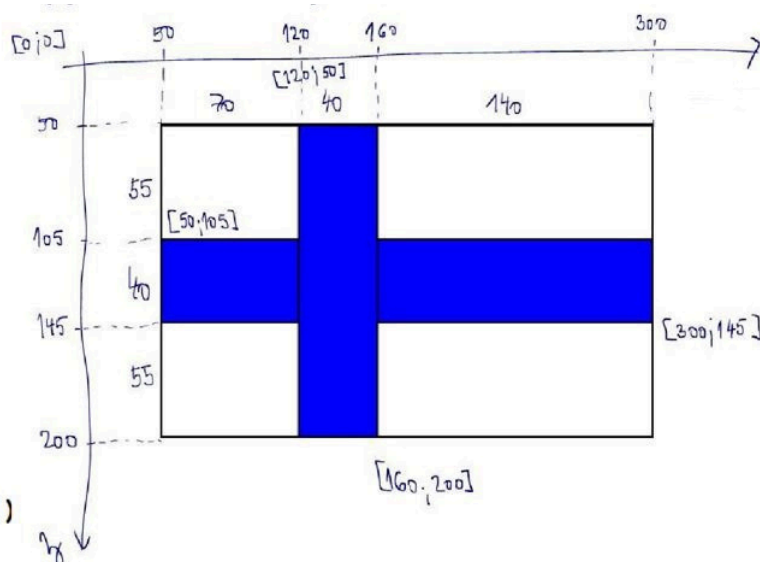
FELADATLAP A DIÁKNAK



```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
```

```
canvas.create_rectangle(50, 50, 150, 200, fill='green')
canvas.create_rectangle(150, 50, 250, 200, fill='white')
canvas.create_rectangle(250, 50, 350, 200, fill='orange')
```

1. 5. Feladat: Hozz létre egy új 06lek_p06.py programot, és rajzolj egy ilyen képet: Melyik ország zászlajára hasonlít a kép?

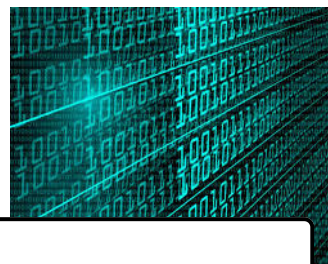


```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
```

```
canvas.create_rectangle(50, 50, 300, 200, fill='white')
canvas.create_rectangle(50, 105, 300, 145, fill='blue')
canvas.create_rectangle(120, 50, 160, 200, fill='blue')
```

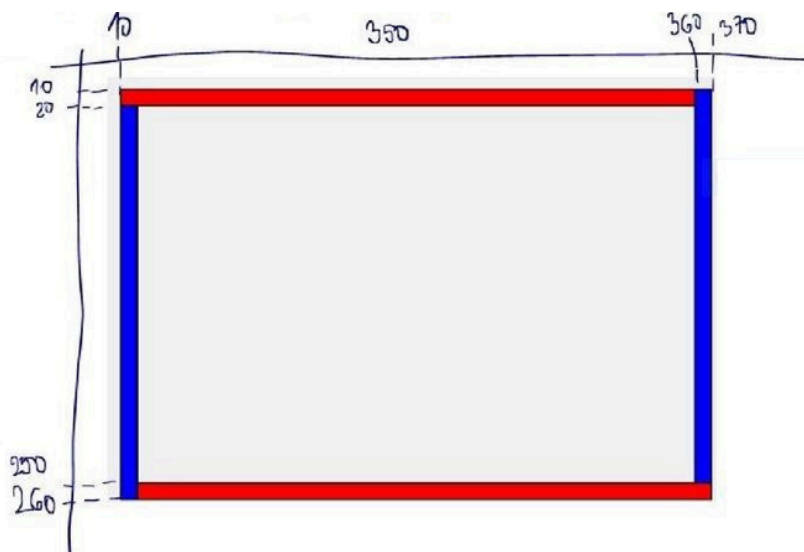


ÓRATERV



1. FELADATLAP A DIÁKNAK

1. 6. Feladat: • Hozzon létre egy új 06lek_p07.py programot, amelyben négy keskeny téglalapról rajzoljon egy ilyen keretet:



```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

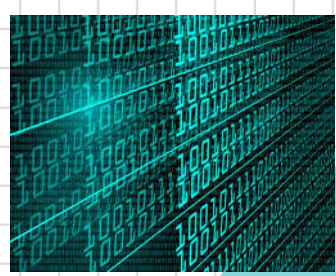
canvas.create_rectangle(10, 10, 360, 20, fill='red')
canvas.create_rectangle(10, 20, 20, 260, fill='blue')
canvas.create_rectangle(20, 250, 370, 260, fill='red')
canvas.create_rectangle(360, 10, 370, 250, fill='blue')
```

1. PYTHON RAJZOLÁSKOR SZÁMOS SZÍN HASZNÁLÁSÁRA VAN LEHETŐSÉGÜNK. ÍME VÁLOGATÁS NÉHÁNY KÖZÜL:

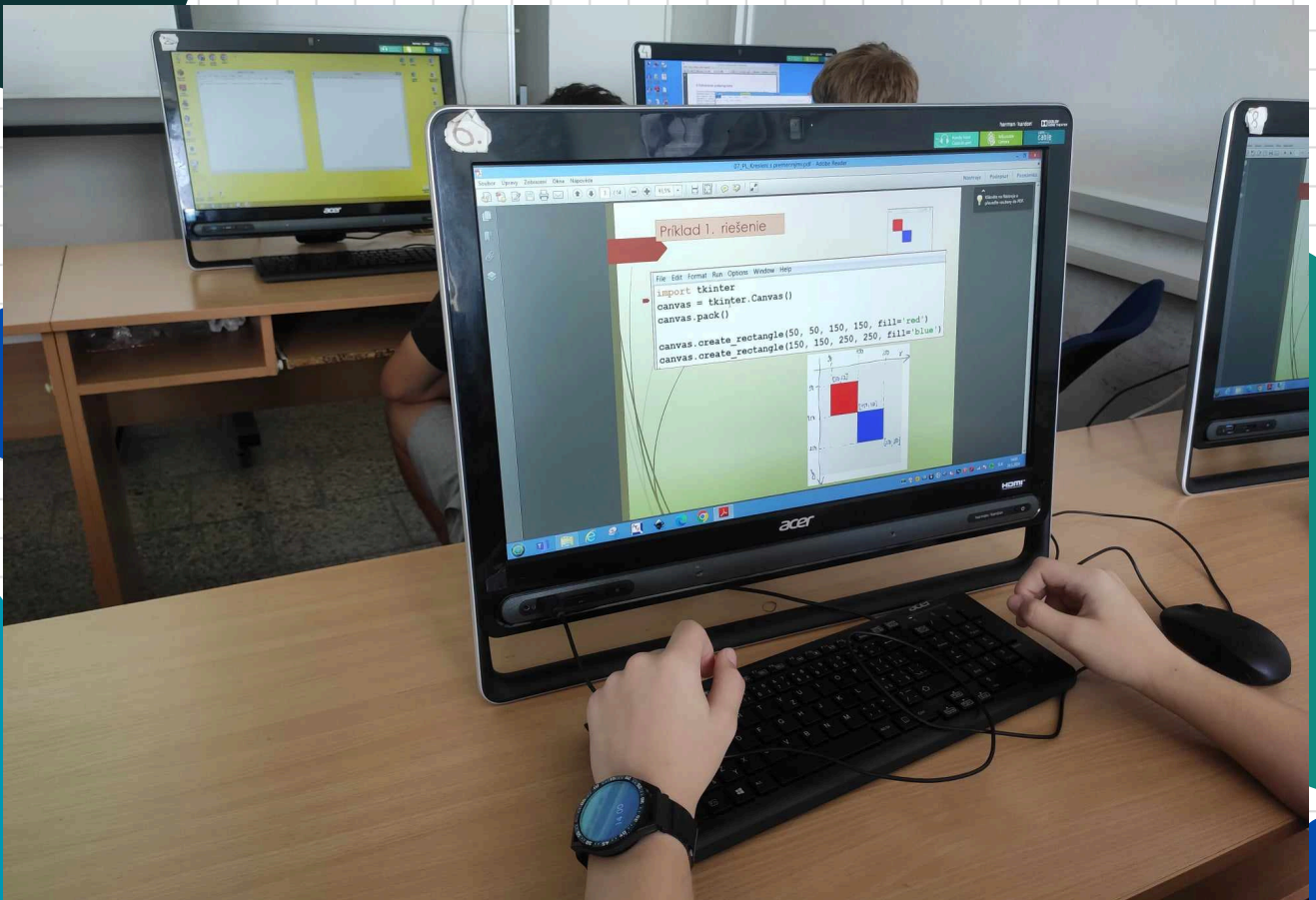
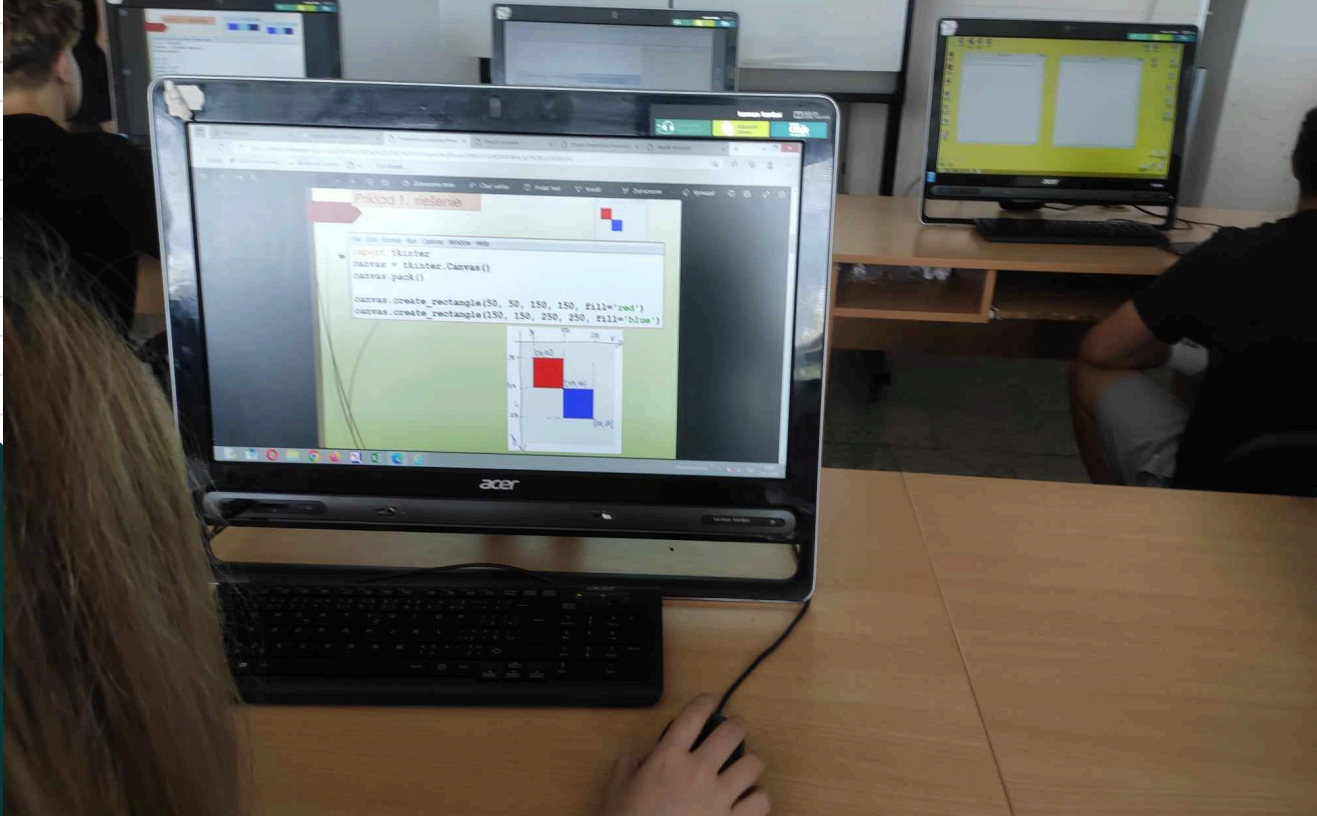
Blue	LightBlue	Cyan	SkyBlue	CornFlowerBlue	DeepSkyBlue	DodgerBlue
RoyalBlue	SlateBlue	SteelBlue	MediumBlue	Navy	Red	SandyBrown
Salmon	Coral	Tomato	Orange	DarkOrange	OrangeRed	IndianRed
Chocolate	Tan	Maroon	Sienna	Brown	SaddleBrown	Pink
Plum	Violet	Orchid	Magenta	Purple	DarkMagenta	Green
PaleGreen	YellowGreen	MediumSeaGreen	LawnGreen	LimeGreen	ForestGreen	DarkGreen
Yellow	Khaki	Gold	Gray	LightGray	Black	White



MELLÉKLET



FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL

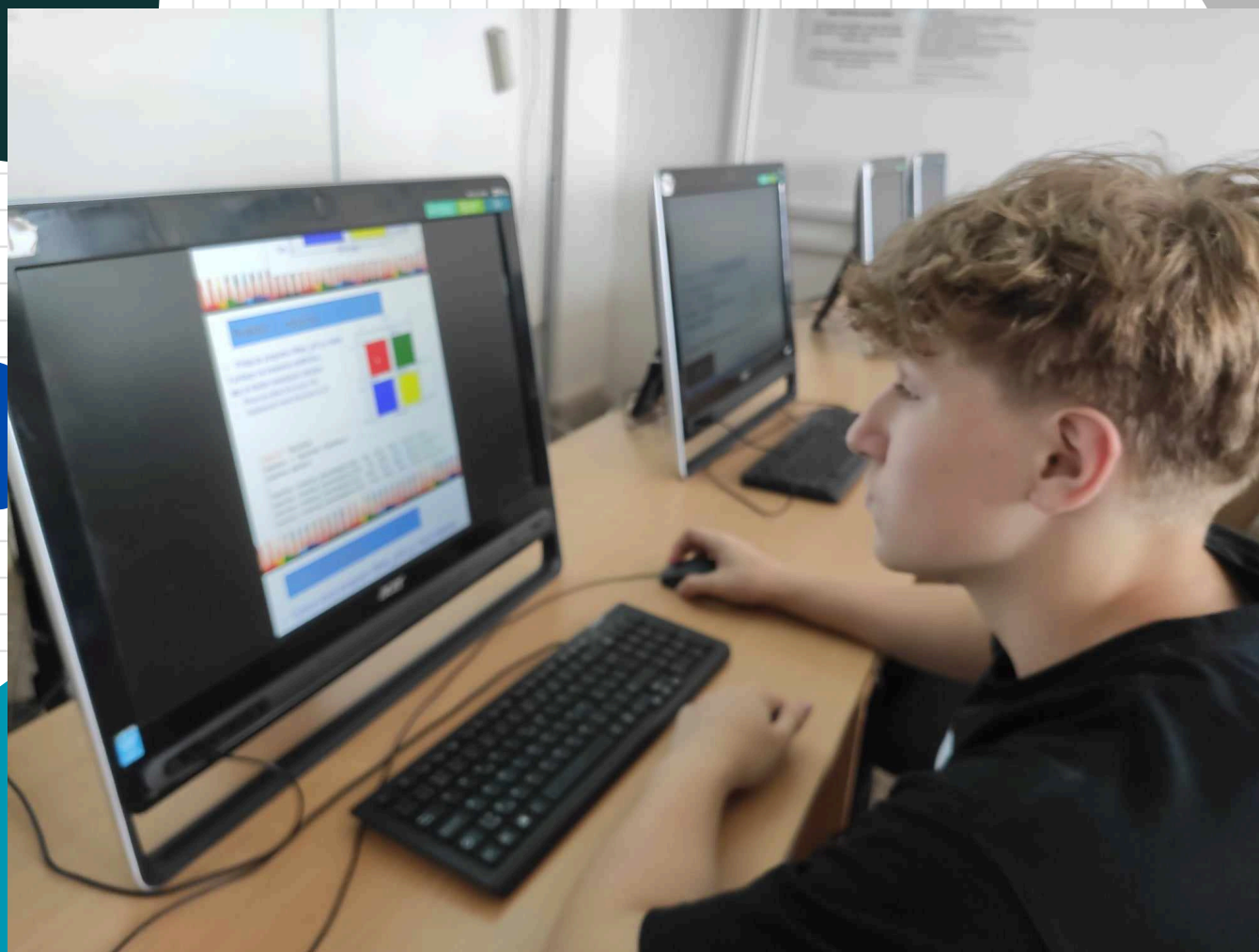
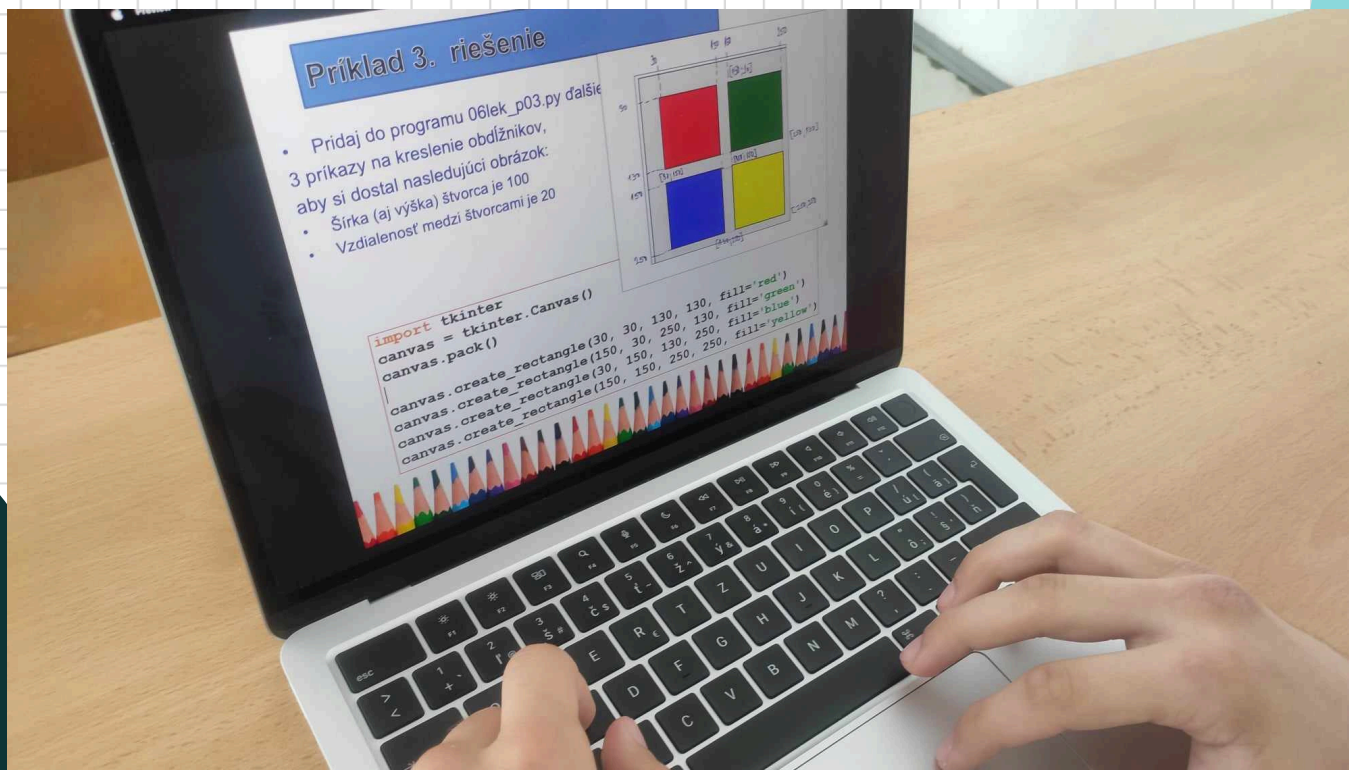




MELLÉKLET

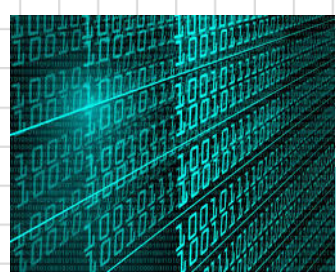


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL

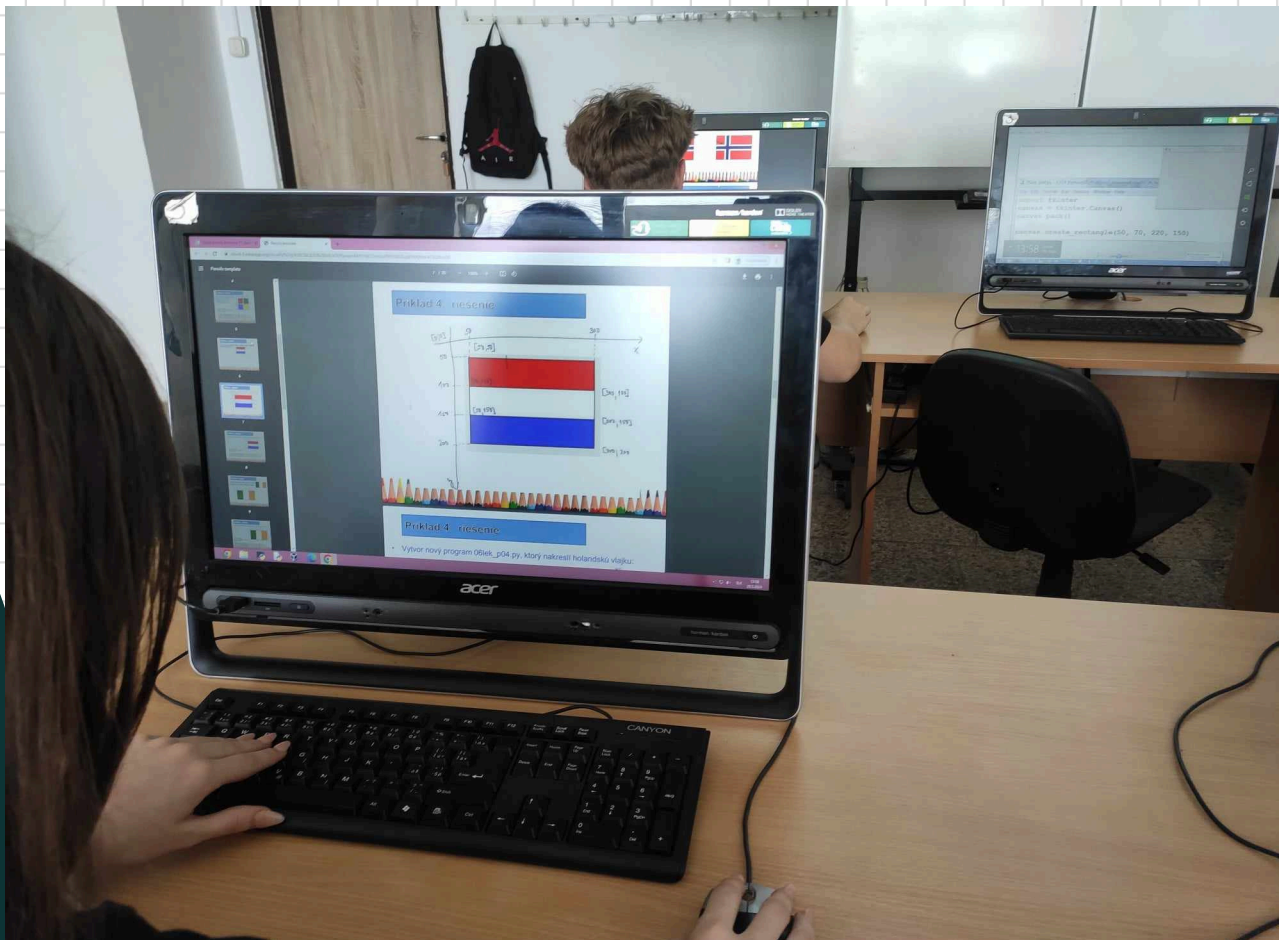




MELLÉKLET

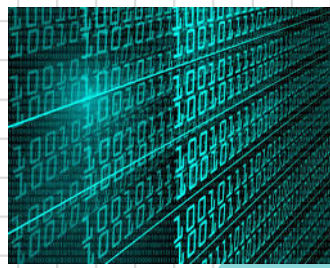


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL





MELLÉKLET



DIÁKOK KIMENETE

IMPORT TKINTER

**# LÉTREHOZUNK EGY ABLAK ABLAKOT =
TKINTER.TK() WINDOW.TITLE("HÁROM TÉGLALAP")**

**# BEÁLLÍTJUK AZ ABLAK MÉRETÉT: SZÉLESSÉG =
400 MAGASSÁG = 200 WINDOW.GEOMETRY(""
{WIDTH}X{HEIGHT}")**

**# LÉTREHOZUNK EGY VÁSZONVÁSZONT =
TKINTER.CANVAS(WINDOW, WIDTH=WIDTH,
HEIGHT=HEIGHT) CANVAS.PACK()**

**# ELSŐ TÉGLALAP X1 = 50 Y1 = 20 X2 = 150 Y2 =
80 COLOR1 = "PIROS"
CANVAS.CREATE_RECTANGLE(X1, Y1, X2, Y2,
FILL=COLOR1)**

**# MÁSODIK TÉGLALAP X2 = 200 Y2 = 50 COLOR2 =
"KÉK" CANVAS.CREATE_RECTANGLE(X1, Y1, X2, Y2,
FILL=COLOR2)**

**# HARMADIK TÉGLALAP X1 = 100 Y1 = 100 X2 = 250
Y2 = 150 COLOR3 = "ZÖLD"
VÁSZON.CREATE_RECTANGLE(X1, Y1, X2, Y2,
FILL=COLOR3)**



ÓRATERV



TÁRGY: PROGRAMOZÁS

ÓRA TÉMA: PROGRAMOZÁS PYTHONBAN – RAJZÁS VÁLTOZÓKKAL.

TANULÁSI CÉLKITÜZÉSEK:

- **1. ISMERJE MEG A PYTHONBAN A VÁLTOZÓK SEGÍTSÉGÉVEL TÖRTÉNŐ RAJZOLÁS ALAPELVEIT.**
- **2. TANULJA MEG A VÁLTOZÓK HASZNÁLATÁT PYTHONBAN VALÓ RAJZOLÁS SORÁN.**

ESZKÖZÖK:
EGY SZÁMÍTÓGÉP PYTHONNAL ÉS A TKINTER KÖNYVTÁRRAL.
PYTHON MINTA TANKÖNYV MUNKALAP GYAKORLATOKKAL

HOOK/INTRO: VÁLTOZÓK HASZNÁLATA RAJZOLÁSHOZ:
VÁLTOZÓK MEGHATÁROZÁSA ÉS HASZNÁLATA PYTHONBAN
HASZNÁLJON VÁLTOZÓKAT ALAKZATOK RAJZOLÁSÁHOZ.
ÁLLÍTSA BE AZ ALAKZATOK SZÍNÉT ÉS MÉRETÉT.

ÓRA FELÉPÍTÉSE: ELKÖTELEZŐDÉS (4 PERC) – KÉP KÉSZÍTÉSE RAJZ SEGÍTSÉGÉVEL

- **PYTHONBAN VÁLTOZÓ HASZNÁLATÁVAL BESZÉLGETÉS A TANULÓKKAL A RAJZOLÁS ÉS A SZÍNEKKEL VALÓ MUNKA ALAPELVEIRŐL PYTHONBAN A TKINTER KÖNYVTÁR HASZNÁLATÁVAL**
- **KUTATÁS (12 PERC) – ÚJ VÁLTOZÓPARANCSONK FELFEDEZÉSE A KÉPKITÖLTÉSHEZ ÉS AZ OUTLINE-HOZ**
- **ÉRTELMEZÉS (15 PERC) – EGYSZERŰ PROGRAM ÖSSZEFoglalása ÉS ÍRÁSA PYTHON NYELVEN, AMELY KÜLÖNBÖZŐ SZÍNEKKEL ÉS MÉRETEKKEL RAJZOL KÜLÖNBÖZŐ ALAKZATOKAT KIDOLGOZÁS (9 PERC) – A TANULÓK TÁMOGATÁSA A PYTHON PROGRAMOZÁSI LEHETŐSÉGEK TOVÁBBI ÖNÁLLÓ MEGRAJZOLÁSÁBAN VÁLTOZÓK SEGÍTSÉGÉVEL**
- **ÉRTÉKELÉS (5 PERC) – AZ ÖNÉRTÉKELŐ TESZT MEGOLDÁSA, A VÁLASZOK MEGBESZÉLÉSE.**

FELADATLAP A DIÁKNAK

1. A PROGRAMOZÁS PYTHONBAN ELEKTRONIKUS TANKÖNYVÜNK SZERINT JÁRUNK EL, AMELY KÖZÉPISKOLÁK SZÁMÁRA KÉSZÜLT. MINTATANKÖNYVET TALÁLHAT A KÖVETKEZŐ CÍMEN: [HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf) A MINTA EGYSZERŰ PÉLDÁKAT TARTALMAZ VÁLTOZÓS RAJZOLÁSRA:

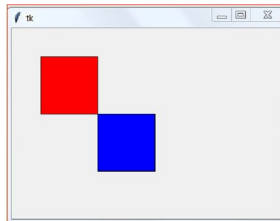
1. NÉGYZETET, TÉGLALAPOT, HÁROMSZÖGET ÉS KÖRT RAJZOLÓ PROGRAM. AZ OBJEKTUM BAL FELSŐ SARKÁNAK KOORDINÁTÁIT AZ X, Y VÁLTOZÓK TÁROLJÁK.

2. MINTAKÓD VÁLTOZÓKKAL A KITÖLTÉSHEZ ÉS A OUTLINE KÜLÖNBÖZŐ SZÍNEKHEZ ÉS VONALVASTAGSÁGOKHOZ EXPOSE FÁZIS – ÚJ PARANCSONK EXPONÁLÁSA VÁLTOZÓKKAL – A TÁBLÁN RÖVIDEN MEGJEGYZEM AZ ALAPFOGALMAK TERM TÉRKÉPÉT

ÓRATERV

3. TANULÓI FELADATLAP

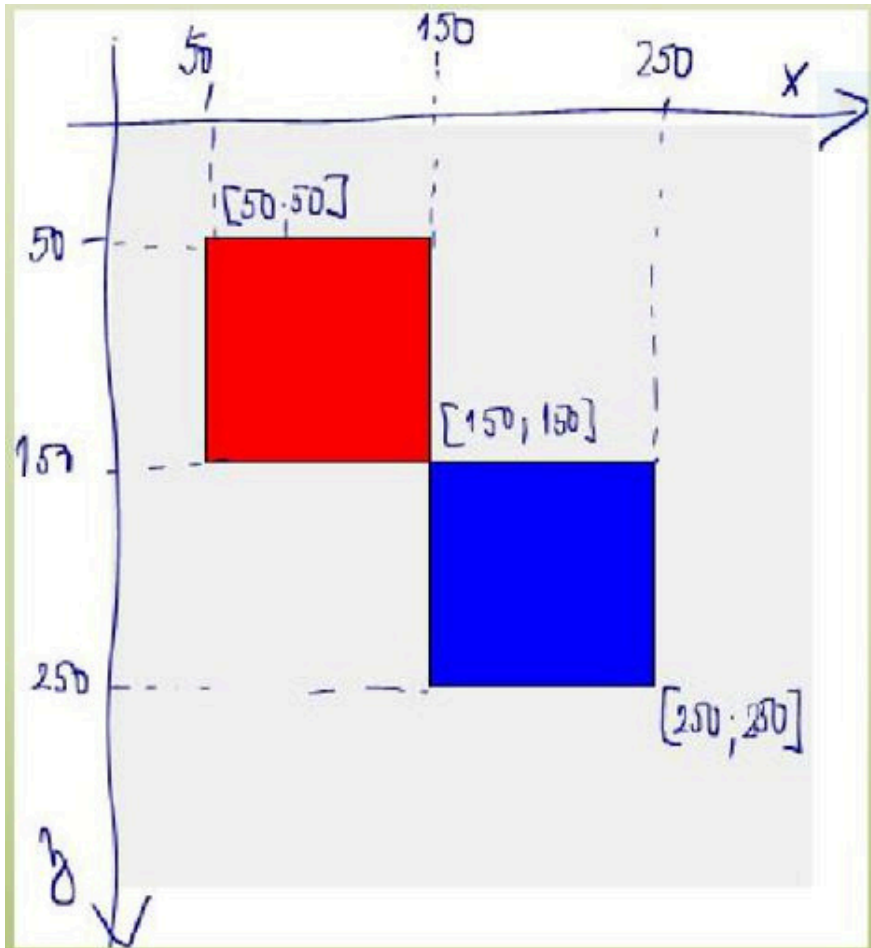
1.Feladat: Hozzon létre egy új 07lek_p01.py programot, amely a képen látható módon két egymáshoz kapcsolódó négyzetet rajzol. Mindkettő oldalhossza 100, a pirosnak a bal felső sarka az [50, 50] pontnál van.



File Edit Format Run Options Window Help

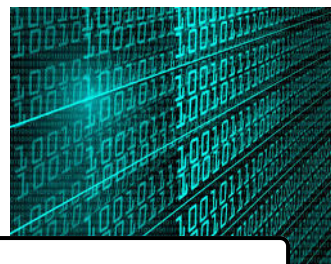
```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

canvas.create_rectangle(50, 50, 150, 150, fill='red')
canvas.create_rectangle(150, 150, 250, 250, fill='blue')
```





ÓRATERV



1.

3. TANULÓI FELADATLAP

1.

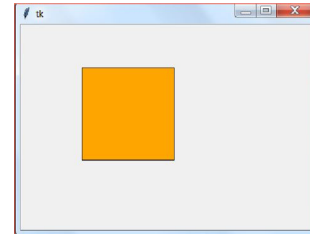
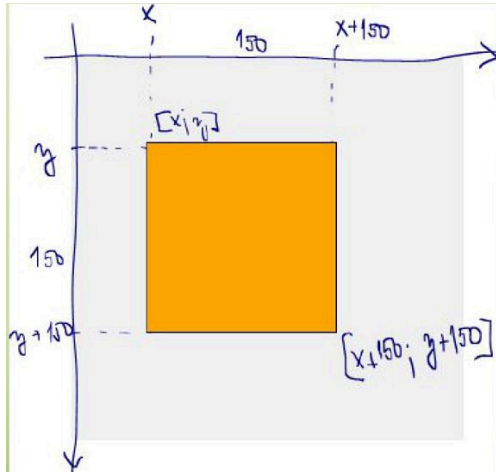
2. Feladat: Hozzon létre egy új 07lek_p02.py programot, amely 150 oldalhosszúságú négyzetet rajzol.

2.

Az x , y változók a négyzet bal felső sarkának koordinátáit tárolják.

3.

A bal felső sarok koordinátái $[x;y] = [100;70]$ változóként kerülnek megadásra.

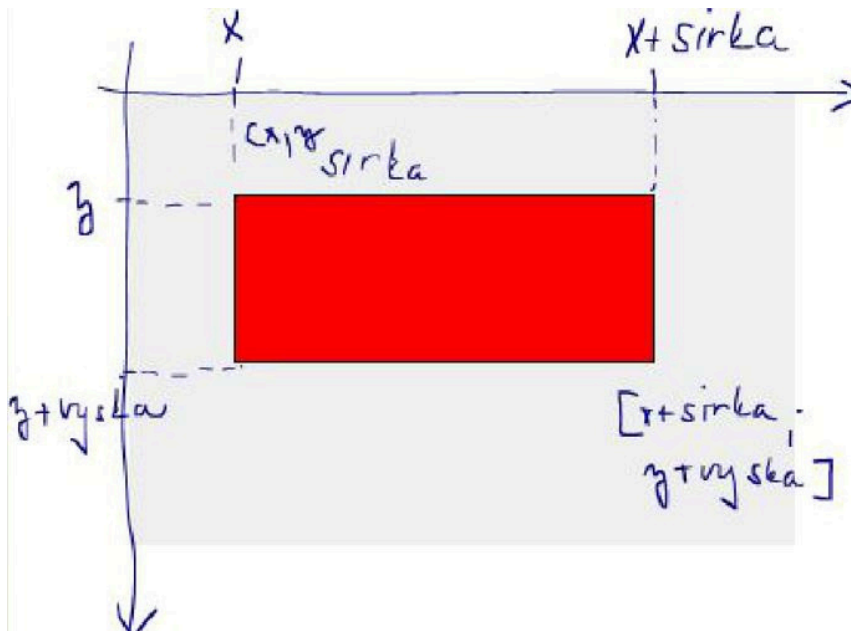


```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

x = 100
y = 70
canvas.create_rectangle(x,y,x + 150,y + 150, fill='orange')
```

1. 3. Feladat: Készítsen 07lek_p03.py programot, amely x , y koordinátákra, adott szélességgel és magassággal egy téglalapot rajzol a bal felső sarkával. Válassza ki a kívánt színt.

2. Például, ha a program a következőket tartalmazza: $x = 100$, $y = 70$, szélesség = 200, magasság = 80, akkor a képen látható téglalap rajzolódik ki:





ÓRATERV



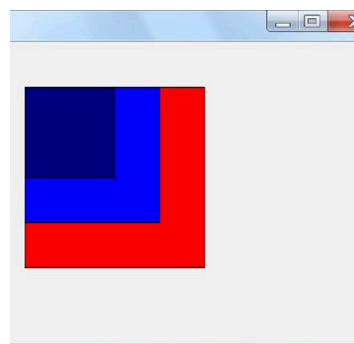
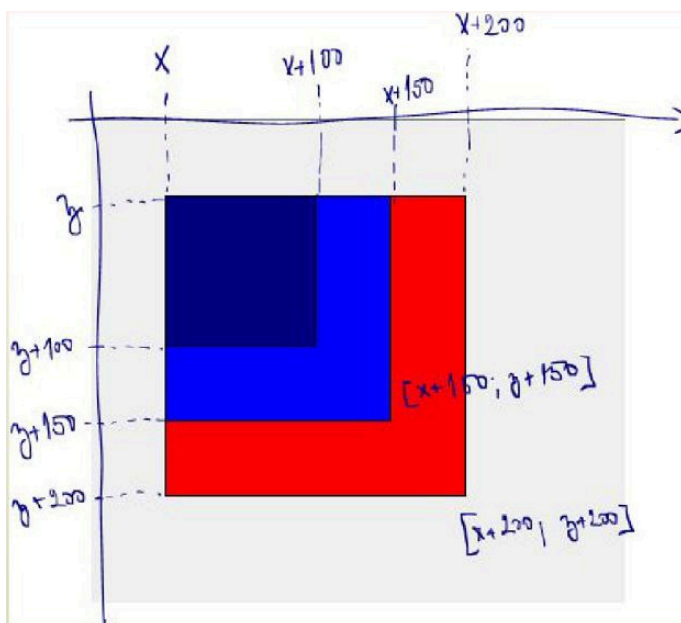
1. 3. TANULÓI FELADATLAP

File Edit Format Run Options Window Help

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

x = 100
y = 70
sirka = 200
vyska = 80
canvas.create_rectangle(x, y, x + sirka, y + vyska, fill='red')
```

4. Feladat: Hozzon létre egy 07lek_p04.py programot, amely négyzeteket rajzol a képen látható módon. Ezeknek a négyzeteknek van egy közös bal felső sarkuk, melynek koordinátái az x, y változókból vannak.
- A négyzetek fokozatosan csökkennek, így a piros oldalhossza 200, a kék 150 és a sötétkék oldalhossza 100.
- A középpont koordinátái lehetnek például: [x;y] = [80;50]



File Edit Format Run Options Window Help

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

x = 80
y = 50
canvas.create_rectangle(x, y, x+200, y+200, fill='red')
canvas.create_rectangle(x, y, x+150, y+150, fill='blue')
canvas.create_rectangle(x, y, x+100, y+100, fill='navy')
```



ÓRATERV



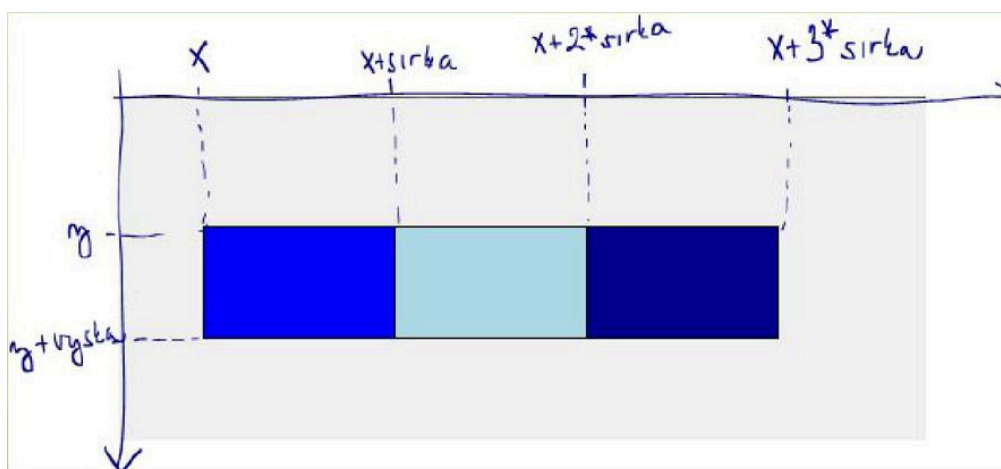
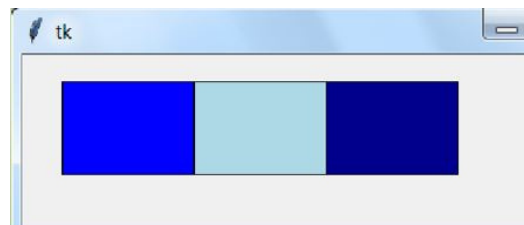
1.3. TANULÓI FELADATLAP

1.5. Feladat: Hozzon létre egy 07lek_p05.py programot, amely három egymást érő téglalapot rajzol:

2. Az első téglalap bal felső sarkának koordinátáit az x , y változók tárolják.

Mindhárom téglalapnak azonos a szélessége és magassága – ezek a méretek a szélességi és magassági változókban tárolódnak.

Akkor is helyesen fog működni a program, ha a változó szélességét 20-al, az y változó értékét 10-el növeljük? Ha nem, akkor a javítóprogram.



File Edit Format Run Options Window Help

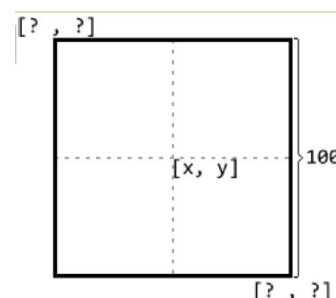
```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

x = 30
y = 20
sirka = 100
vyska = 70
canvas.create_rectangle(x, y, x+sirka, y+vyska, fill='blue')
canvas.create_rectangle(x+sirka, y, x+2*sirka, y+vyska, fill='light blue')
canvas.create_rectangle(x+2*sirka, y, x+3*sirka, y+vyska, fill='dark blue')
```

Feladat: Hozzon létre egy új 07-lek_p06.py programot, amely megrajzol egy négyzetet, amelynek középpontja x , y koordinátaival, oldalainak hossza pedig 100.

Az x , y koordinátákat azonos nevű változókban tároljuk. Ennek a négyzetnek a megrajzolásához ki kell számítanunk a bal felső és jobb alsó sarkának koordinátáit.

Egy új programban írjon egy zöld négyzetet, amelynek középpontja $[x, y]$ és oldalhossza 100.





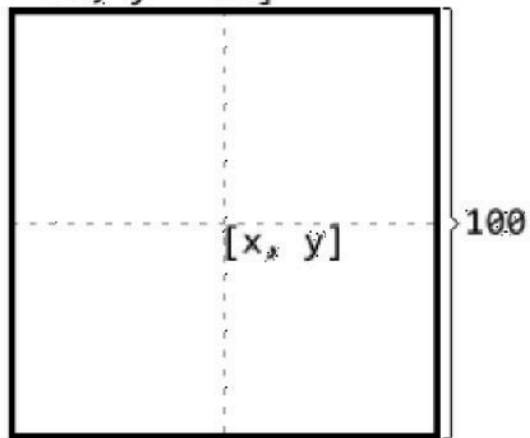
ÓRATERV



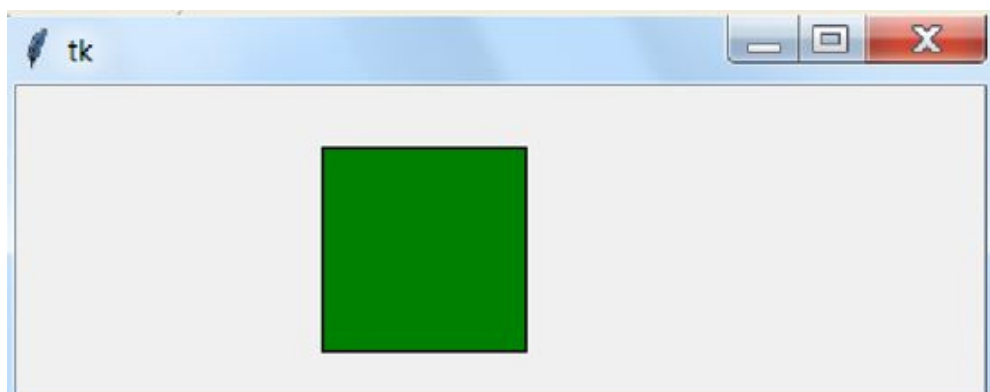
1.

3. TANULÓI FELADATLAP

$[x - 50, y - 50]$



$[x + 50, y + 50]$



File Edit Format Run Options Window Help

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

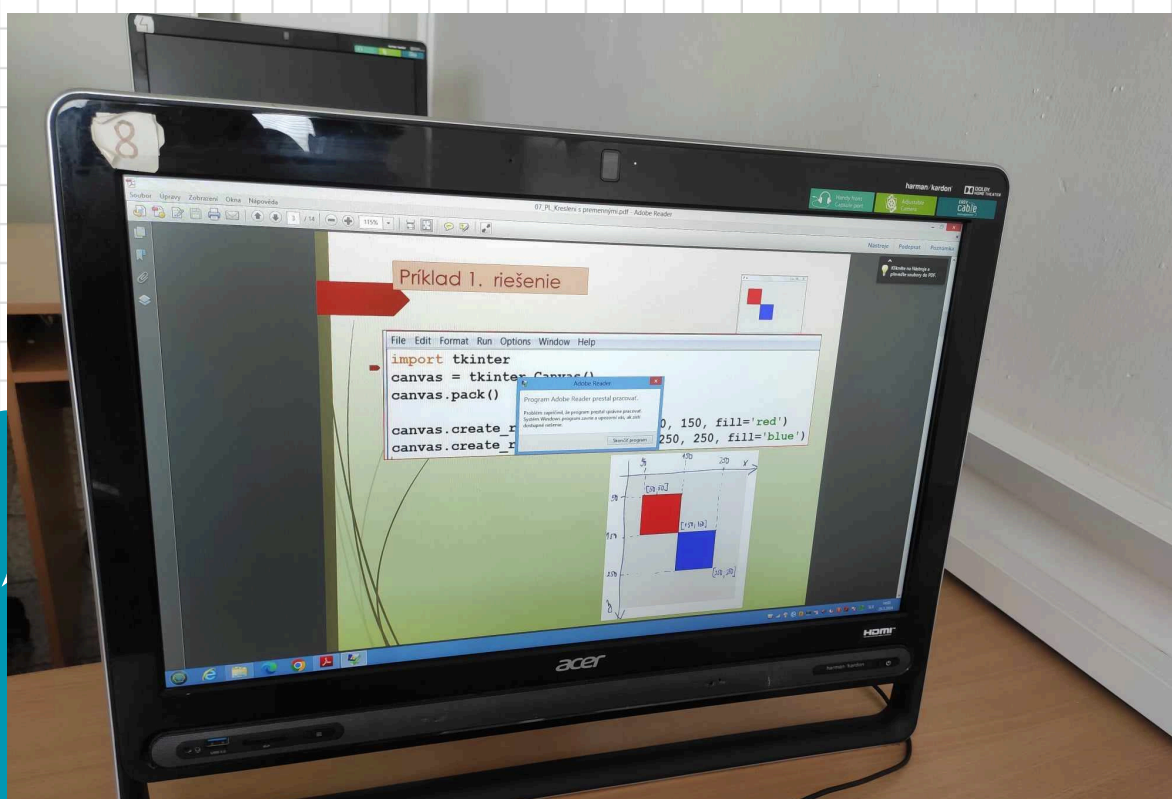
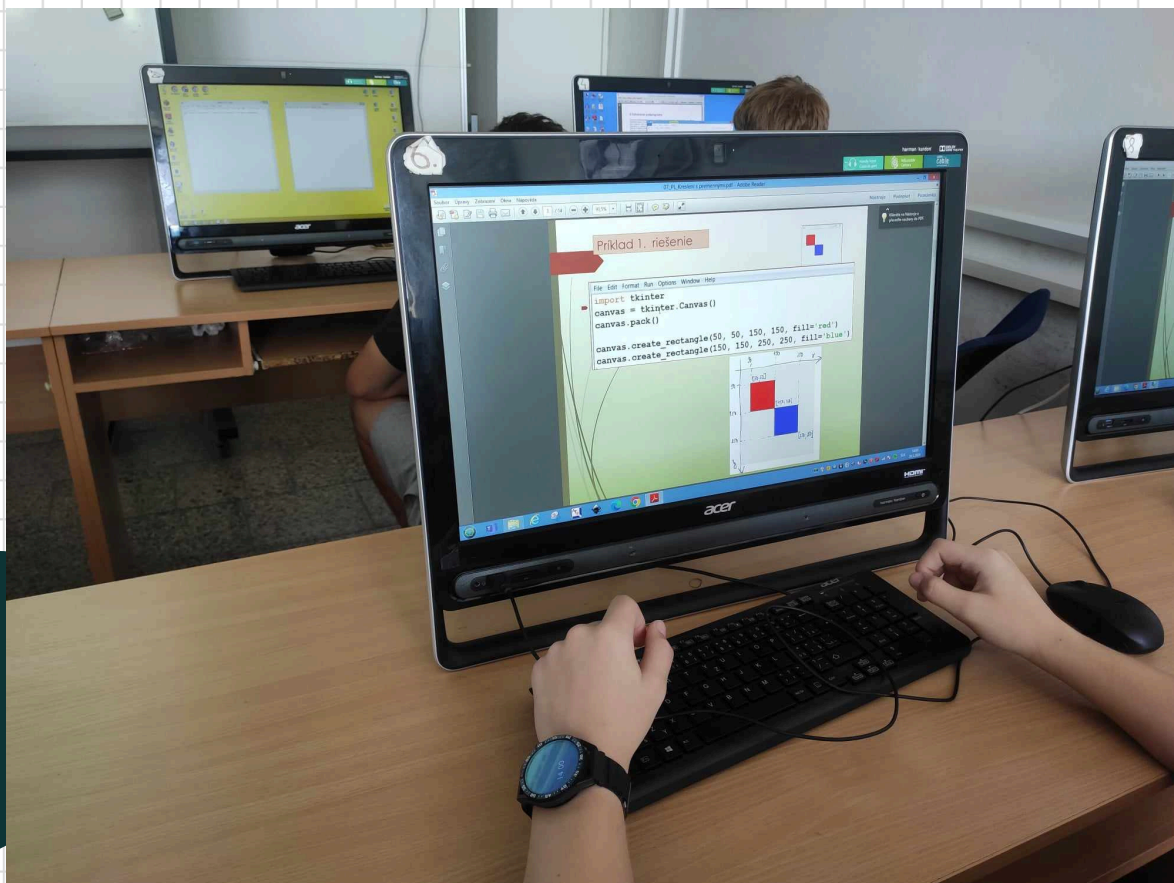
x = 200
y = 80
canvas.create_rectangle(x-50,y-50,x+50,y+50,fill='green')
```




MELLÉKLET

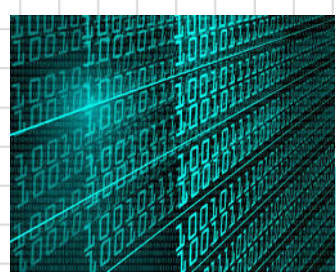


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL

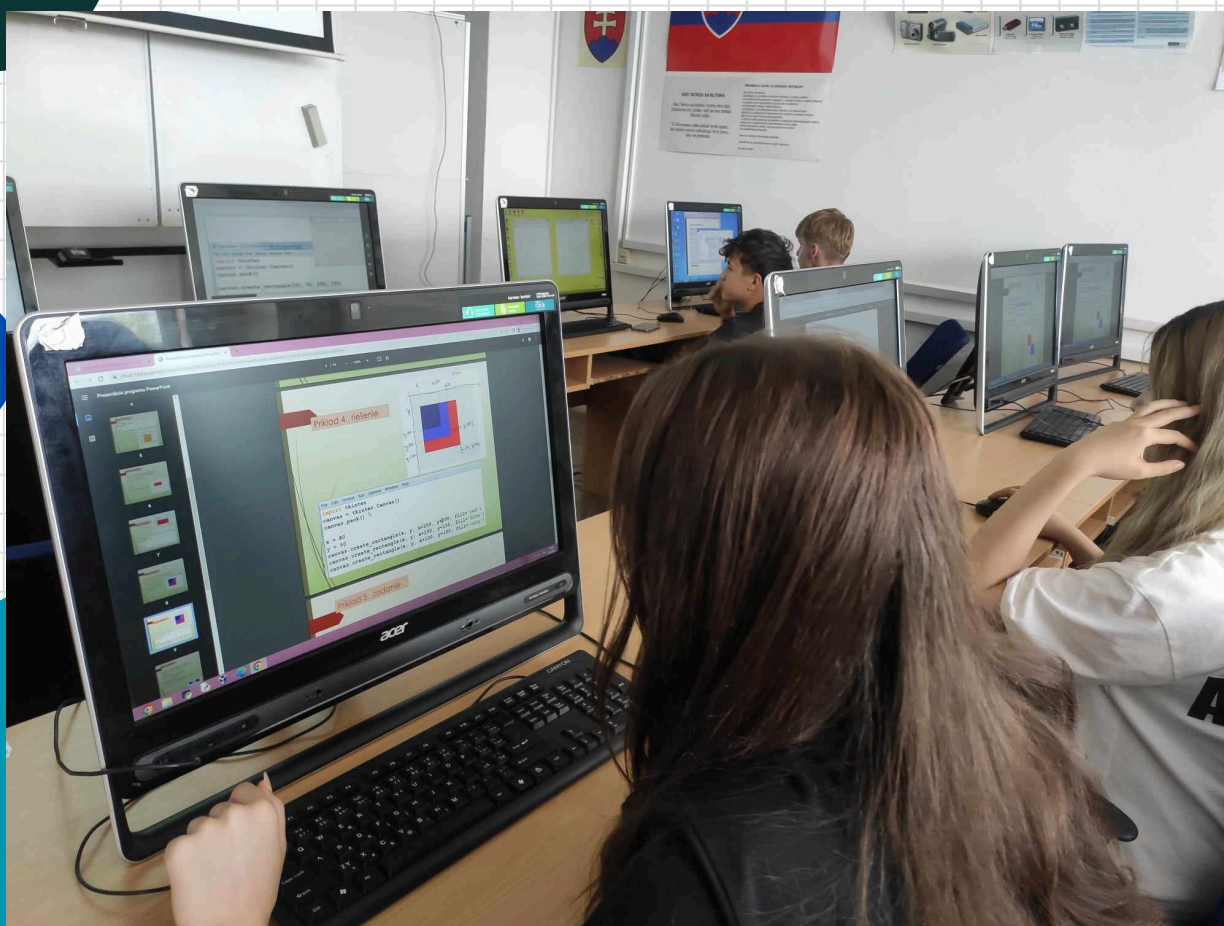
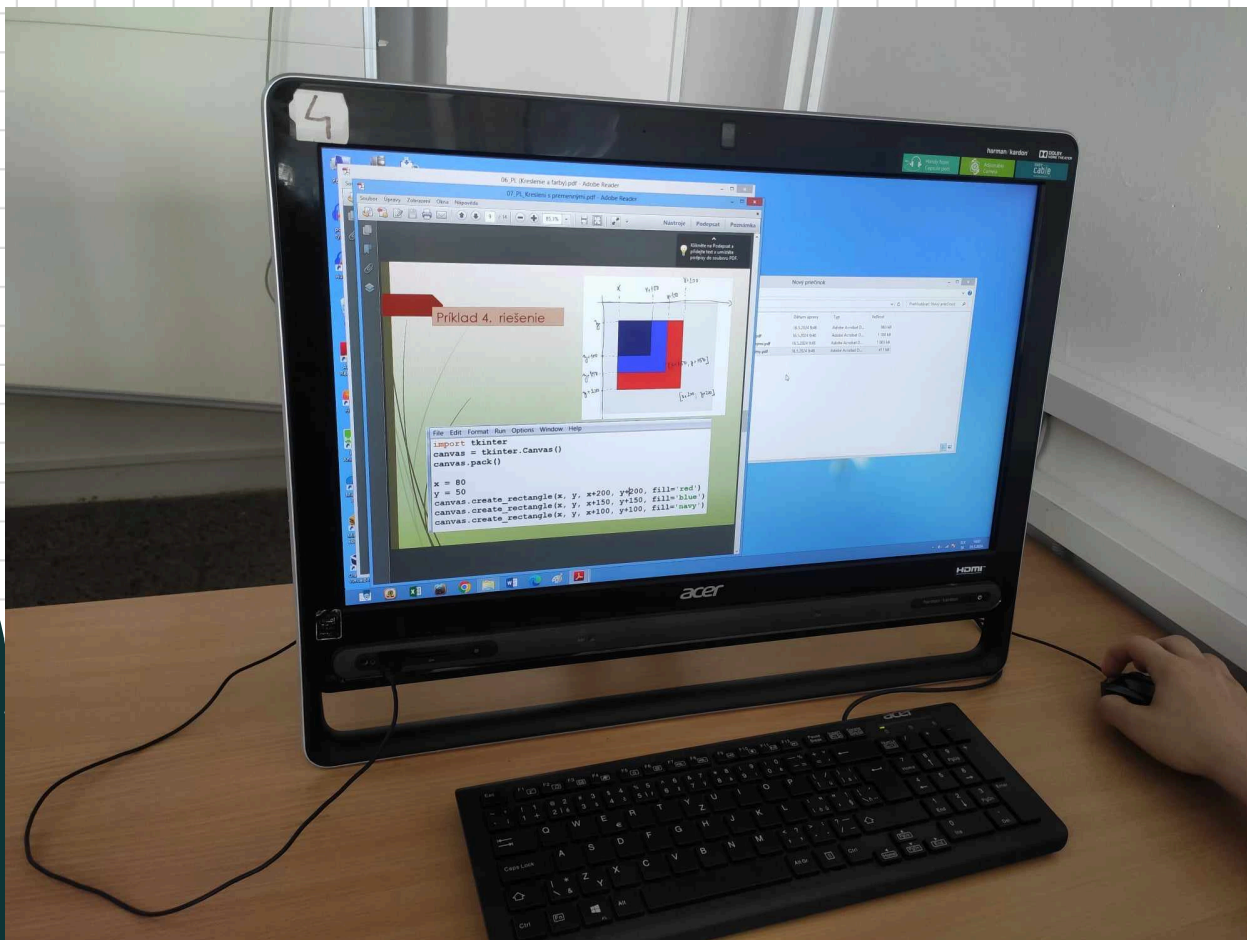




MELLÉKLET

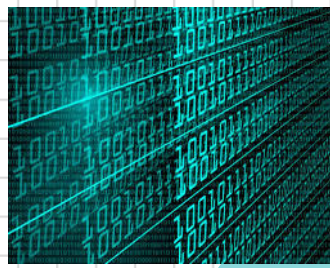


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL





MELLÉKLET



DIÁKOK KIMENETE:

**ÍRJ EGY PROGRAMOT, AMELY
BEOLVAS HÁROM SZÓT, ÉS KIÍRJA
MIND A 6 KÜLÖNBÖZŐ PERMUTÁCIÓT.
INDÍTÁS UTÁN MEGKAPHATJA:**

ÍRJA BE AZ 1. SZÓT: FEHÉR

ÍRJA BE A 2. SZÓT: KÉK

ÍRJA BE A 3. SZÓT: PIROS

FEHÉR KÉK PIROS

FEHÉR PIROS KÉK

KÉK FEHÉR PIROS

KÉK PIROS FEHÉR

PIROS FEHÉR KÉK

PIROS KÉK FEHÉR

VAGY HASZNÁLJON VÁLTOZÓKAT:

MÖGÖTT 1. BÚCSÚ: X

HÁTUL 2. VISZLÁT: ***

MÖGÖTT 3. BÚCSÚ: :-)

X * :-)**

X :-) ***

***** X :-)**

***** :-) X**

:-) X ***

:-) * X**



ÓRATERV



TÁRGY: PROGRAMOZÁS

ÓRA TÉMA: PROGRAMOZÁS PYTHONBAN – SZUBRUTINOK.

TANULÁSI CÉLKITÜZÉSEK:

- 1. HASZNÁLJON SZUBRUTINOKAT A PYTHON RAJZOLÁSI PARANCSAINAK TISZTÁZÁSÁRA
- 2. TANULJON MEG FUNKCIÓK MEGHATÁROZÁSA PYTHONBAN VALÓ RAJZOLÁS KÖZBEN.

ESZKÖZÖK:
EGY SZÁMÍTÓGÉP PYTHONNAL ÉS A
TKINTER KÖNYVTÁRRAL.
PYTHON MINTA TANKÖNYV MUNKALAP
GYAKORLATOKKAL

HOOK/BEVEZETÉS: KÜLÖNBÖZŐ SZÍNEK
HASZNÁLATA RAJZOLÁSKOR:
FÜGGVÉNYEK MEGHATÁROZÁSA

- ALPROGRAM HASZNÁLATA EGYEDI RAJZOBJEKTUMOKHOZ
- MÓDOSÍTSA A FÜGGVÉNYT ÚGY, HOGY A HASZNÁLAT SORÁN HELYET ADHASSUNK
- ISMÉTELJE MEG A PARANCSONKAT CIKLUSBAN

AZ ÓRA FELÉPÍTÉSE: KAPCSOLÓDÁS (4 PERC) – A SUBROUTINE EGY KISEBB RÉSE A PROGRAMNAK, AMIT ELNEVEZÜNK, ÉS AMIKOR HASZNÁLNI AKARJUK, CSAK BEÍRJUK ENNEK AZ ALPROGRAMNAK A NEVÉT A PROGRAMBA.

BESZÉLGETÉS TANULÓKKAL AZ ALPROGRAMOK PYTHONBAN VALÓ HASZNÁLATÁNAK ELŐNYEIRŐL: –

ALPROGRAMOK HASZNÁLATA BONYOLULTABB KÉPEK RAJZOLÁSAKOR ALKALMAS ARRÁ, HOGY A PROGRAM NE LEGYEN ZAVARÓ, MERT SOK PARANCSONKAT TARTALMAZ. A MEGJEGYZÉSEK HASZNÁLATA SEGÍT ÁTLÁTHATÓBBÁ TENNI A PROGRAMOT. JOBB MÓDJA EGY SZUBRUTIN HASZNÁLATA.

• **KUTATÁS (12 PERC) – ÚJ PARANCSONKAT FELFEDEZÉSE – A FÜGGVÉNY EGY ELNEVEZETT PARANCSONBLOKK (NÉHA SZUBRUTINNAK IS NEVEZIK). SPECIÁLIS KONSTRUKCIÓVAL ÍRJUK LE (DEFINIÁLJUK).**

• **PREZENTÁCIÓ (15 PERC) – ISMERETEK ÖSSZEFOGLALÁSA ÉS MAGYARÁZATA A PYTHON FEJLESZTÉSI**

ALPROGRAMOK LÉTREHOZÁSÁNAK KÉRDÉSKÖRÉNEK VIZSGÁLATA (9 PERC) – A HALLGATÓK TÁMOGATÁSA PYTHON ALPROGRAMOK TOVÁBBI ÖNÁLLÓ LÉTREHOZÁSÁBAN.

• **ÉRTÉKELÉS (5 PERC) – AZ ÖNÉRTÉKELŐ TESZT MEGOLDÁSA, A VÁLASZOK MEGBESZÉLÉSE.**

FELADATLAP A DIÁKNAK

A PROGRAMOZÁS PYTHONBAN ELEKTRONIKUS TANKÖNYVÜNK SZERINT JÁRUNK EL, AMELY KÖZÉPISKOLÁK SZÁMÁRA KÉSZÜLT. TANKÖNYVMINTÁT TALÁLHAT A KÖVETKEZŐ CÍMEN:

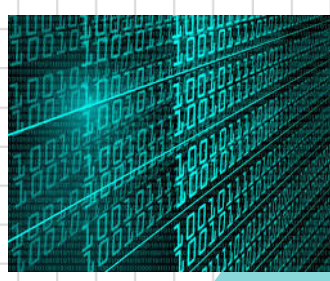
[HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf) A MINTA EGYSZERŰ PÉLDÁKAT TARTALMAZ ALPROGRAMOK LÉTREHOZÁSÁRA PYTHONBAN:

1. ELKÉSZÍJTJÜK SAJÁT PARANCSONKAINKAT, AMELYEKEKET A PROGRAMUNKBAN HASZNÁLHATUNK. SZAKMAILAG AZ ILYEN SZUBRUTINOKAT A PYTHONBAN FÜGGVÉNYNEK HÍVJÁK (MÁS PROGRAMOZÁSI NYELVEKBEN ELŐFORDULHAT, HOGY ELJÁRÁS VAGY FÜGGVÉNY NEVE VAGY MINDKETTŐ).

FELFEDEZÉSI FÁZIS – AZ ÁLLÍTÁSONKAT BAL MARGÓRÓL VALÓ ELTOLÁSA SEGÍT A PYTHONNAK MEGÉRTENI, HOGY MI MINDEN MEGY A FUNKCIÓNKBA. – AZ ALAPFOGALMAKAT RÖVIDEN MEGJEGYZEM A TÁBLÁN



ÓRATERV



4. TANULÓI FELADATLAP

1. Feladat: Rajzolj egy házat, és alkoss hozzá egy funkció- (parancs)házat!

```
import tkinter
```

```
vászon = tkinter.Canvas() canvas.pack() canvas.create_rectangle(100, 100, 150, 150) canvas.create_line(100, 100, 125, 50, 150, 100)
```

A program téglalapból és szaggatott vonalból tetős házat rajzol. A következő példában definiáltuk a ház nevű saját függvényünket. A függvényt a `def` szóval határozzuk meg, ezt követi a függvény neve, majd zárójelek és a végén kettőspont (mint a `for` ciklusban).

```
import tkinter canvas = tkinter.Canvas() canvas.pack() def dom():  
    canvas.create_rectangle(100, 100, 150, 150) canvas.create_line(100, 100, 125, 50, 150, 100)
```

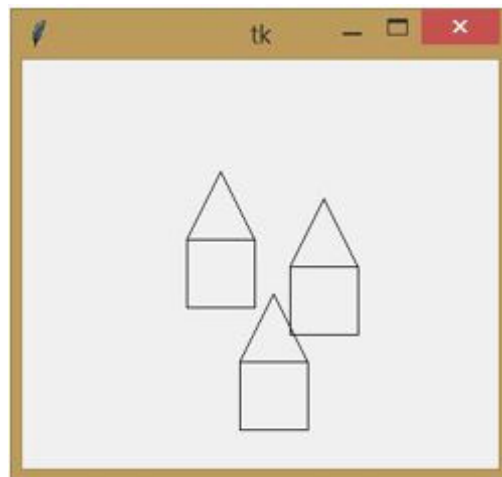
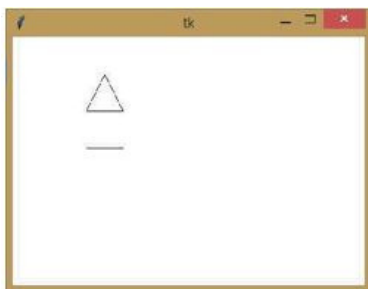
Ha ezt a függvényt szeretnénk végrehajtani, akkor a `dom()`-t írjuk a programba (a program azon helyére, ahol a függvényt végre kell hajtani).

```
import tkinter canvas = tkinter.Canvas() canvas.pack() def dom():  
canvas.create_rectangle(100, 100, 150, 150) canvas.create_line(100, 100, 125, 50, 150, 100) dom()
```

Ez a program elindítja a `dom()` függvényt is. Jobban átlátjuk az egyéni parancs meghatározásának fontosságát, ha a ház függvényt úgy változtatjuk meg, hogy egy házat véletlenszerű helyre rajzoljunk.

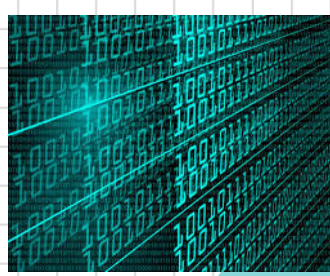
```
tkinter importálása véletlenszerű importból * canvas = tkinter.Canvas() canvas.pack() def dom():  
x = randint(100, 300) y = randint(100, 300) canvas.create_rectangle(x, y, x+50, y+50) canvas.create_line(x, y, x+25, y-50, x+ 50) , y)  
    dom() dom() dom()
```

Ebben a programban már háromszor végrehajtjuk a `house()` függvényt, így véletlenszerűen három ház kerül kisorsolásra.





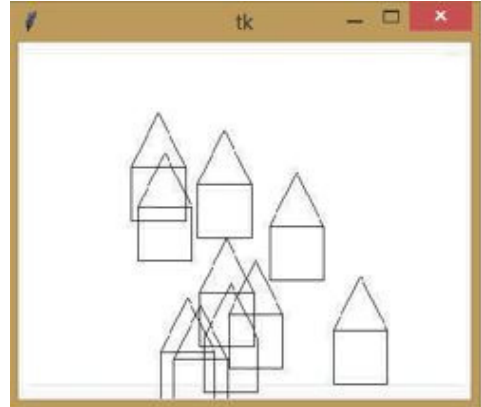
ÓRATERV



TANULÓI FELADATLAP

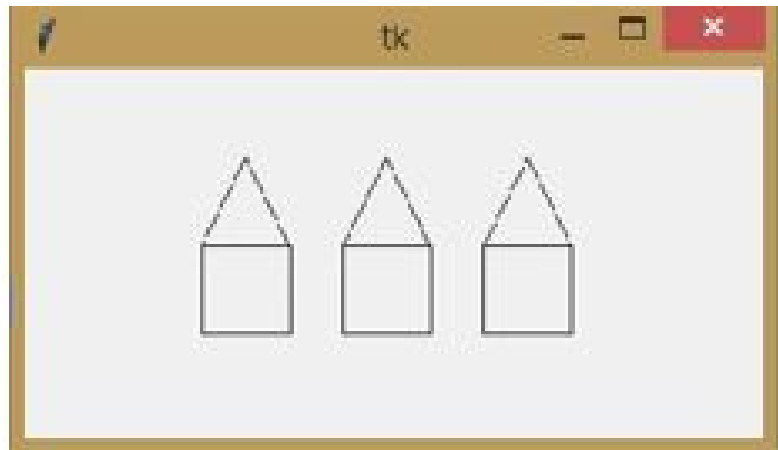
Feladat: Használd a for ciklust, így könnyedén rajzolhatsz 10 házat.

```
import tkinter
véletlenszerű importból *
canvas = tkinter.Canvas() canvas.pack() def dom():
x = randint(100, 300) y = randint(100, 300)
canvas.create_rectangle(x, y, x+50, y+50)
canvas.create_line(x, y, x+25, y-50, x+50, y)
i esetén az (1, 11)
tartományban:
dom()
```

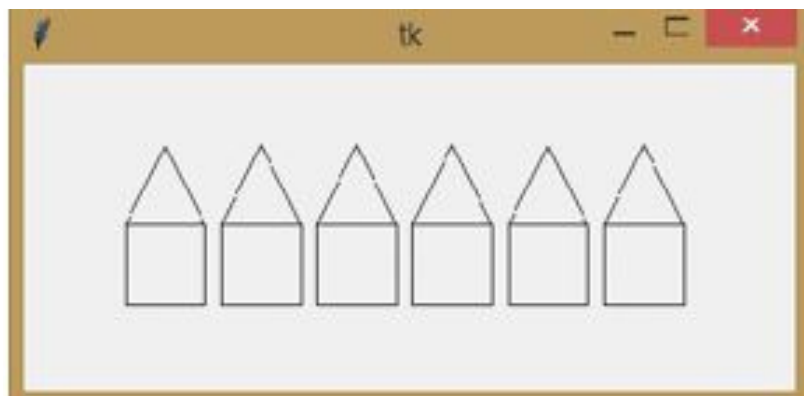


A funkció úgy is módosítható, hogy használatakor beírjuk azt a helyet, ahová a házat meg szeretnénk rajzolni. A `create_rectangle` függvény használatához hasonlóan a zárójelben lévő számok befolyásolják a rajzolás helyét. Az ilyen számokat (vagy egyéb adatokat), amelyek a függvényt befolyásolják, paramétereknek nevezzük. Ha egy függvényt paraméterrel definiálunk, akkor az egyes paraméterek nevét zárójelbe tesszük, vesszővel elválasztva (ha többet akarunk). Ezeket a neveket használhatjuk a függvényünkben (alapvetően olyan változókról van szó, amelyek csak ebben a függvényben használhatók), és ugyanaz az értékük, amit a függvény használatakor adtunk meg.

```
import tkinter
véletlenszerű importból *
canvas = tkinter.Canvas() canvas.pack()
def dom(x, y):
canvas.create_rectangle(x, y, x+50, y+50)
canvas.create_line(x, y, x+25, y-50, x+50, y)
dom(100, 100) dom(180, 100) dom(260, 100)
```

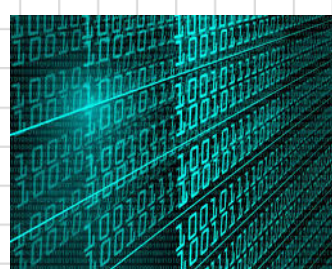


```
tkinter importálása véletlenszerű importból *
canvas = tkinter.Canvas() canvas.pack() def dom(x, y):
canvas.create_rectangle(x, y, x+50, y+50)
canvas.create_line(x, y, x+25, y-50, x+50, y) for i tartományban (1, 7):
dom(i*60, 100)
```

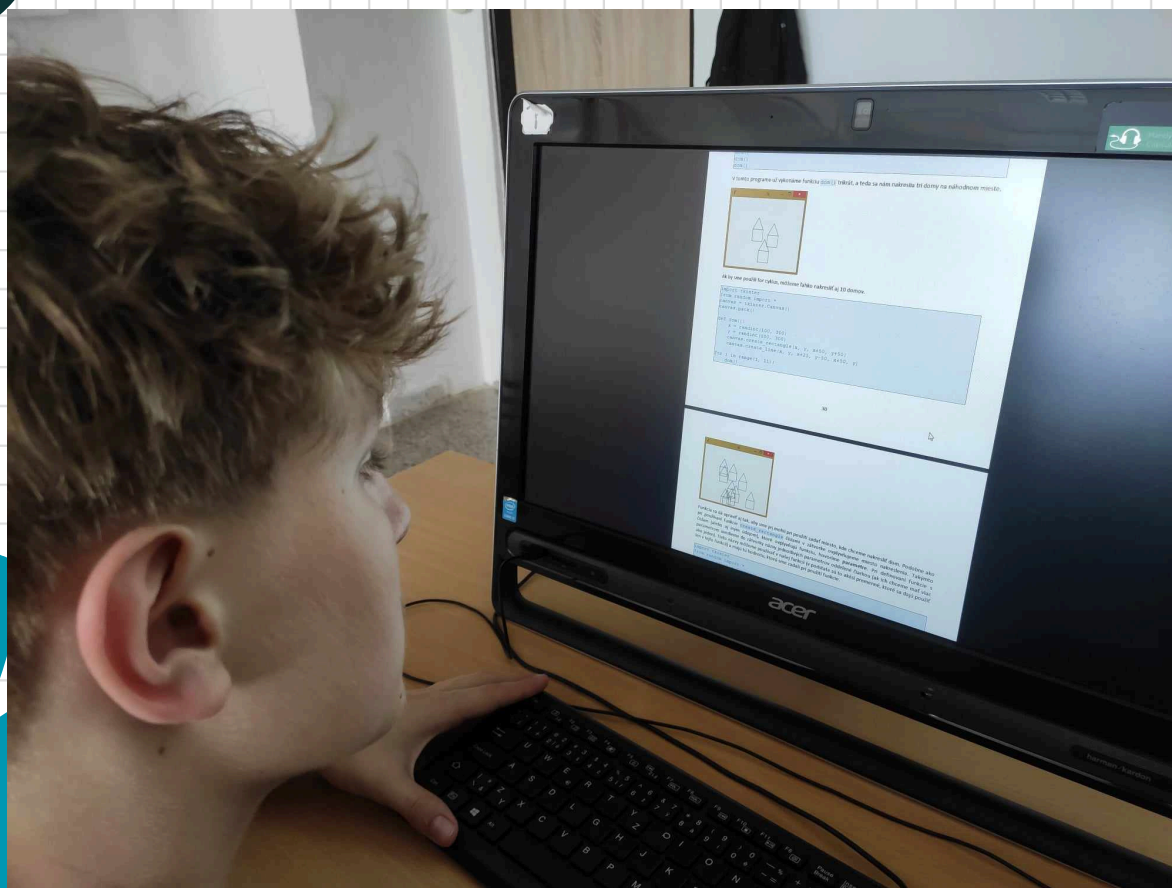
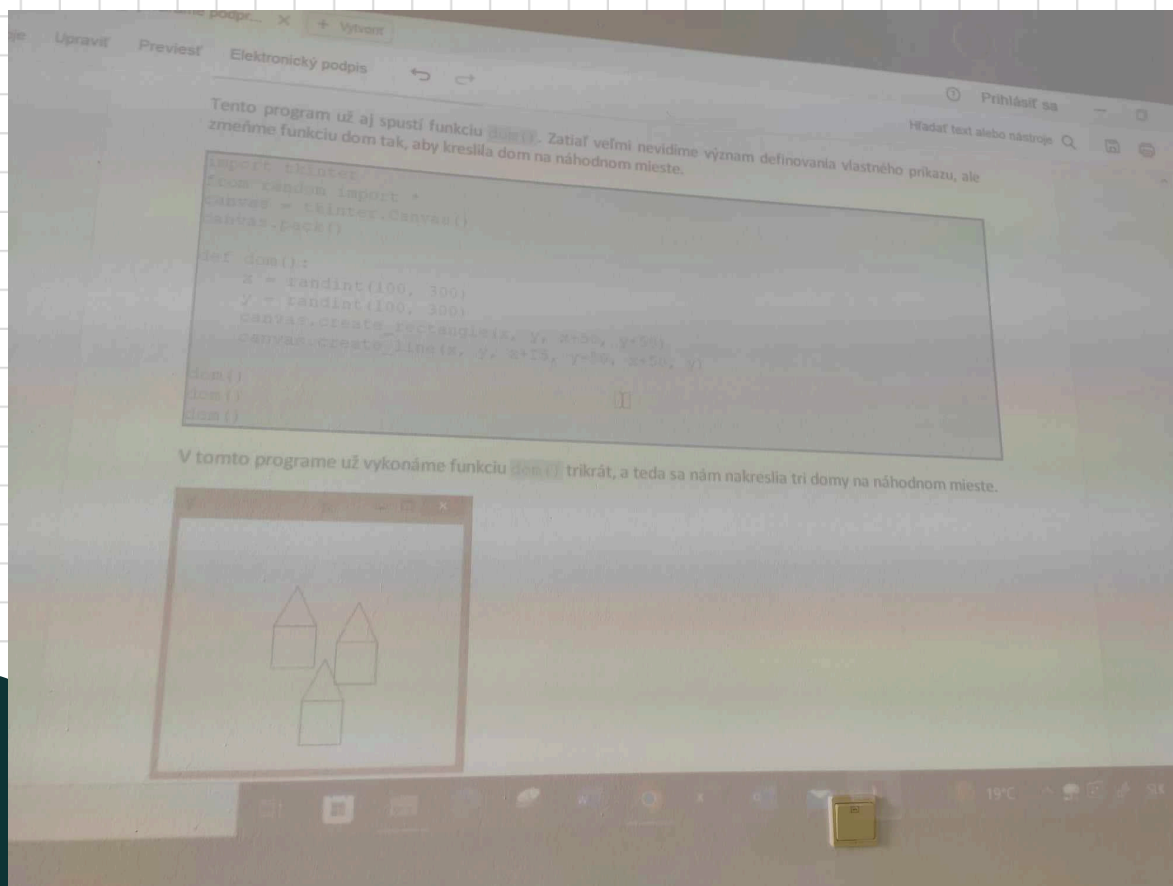




MELLÉKLET

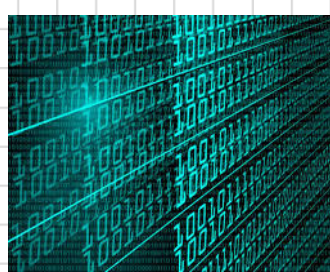


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL

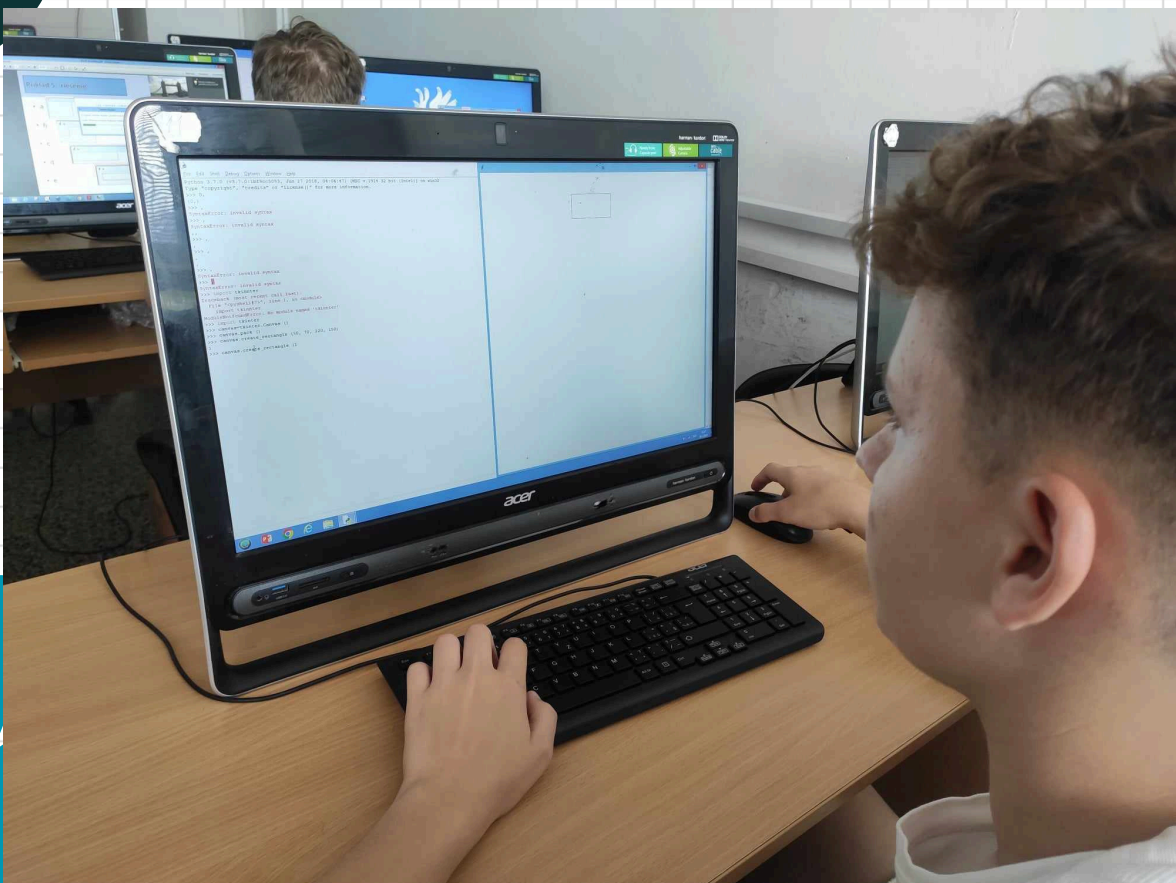
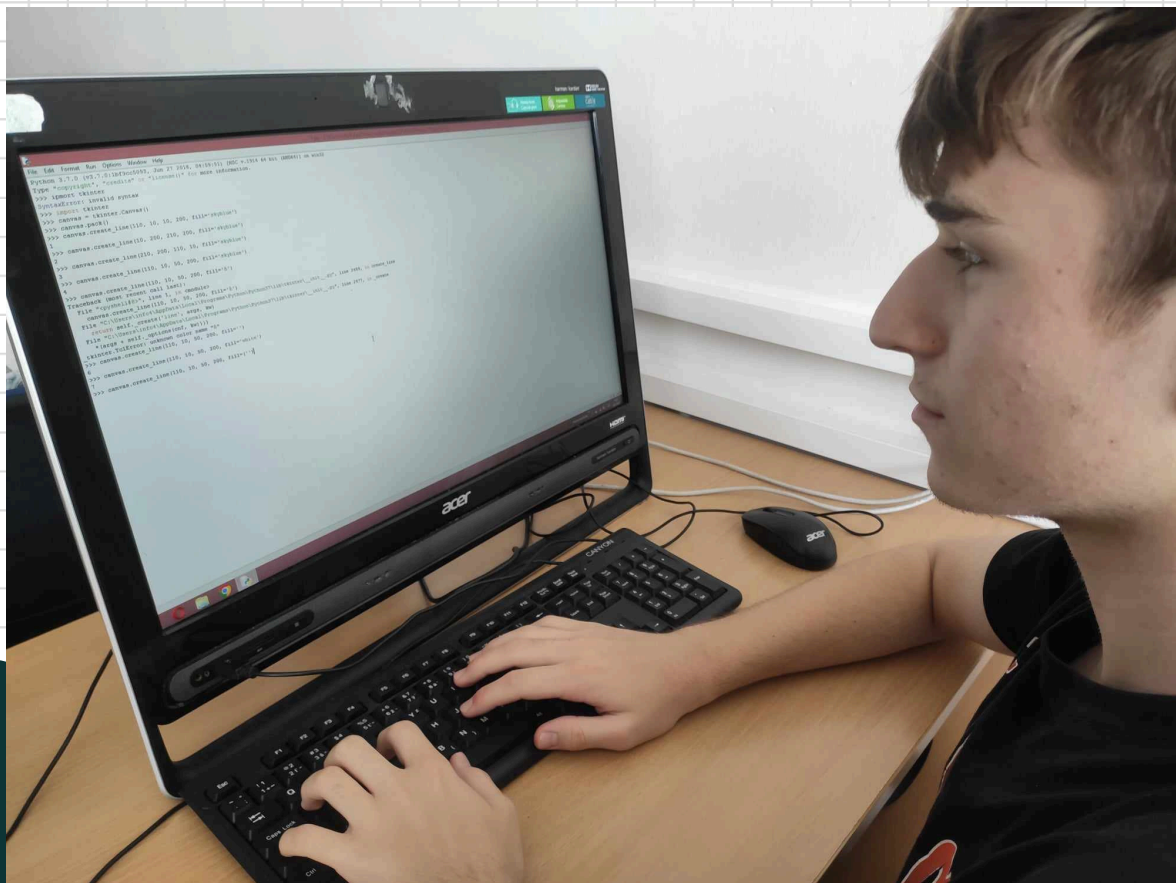




MELLÉKLET



FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL





MELLÉKLET



DIÁKOK KIMENETE

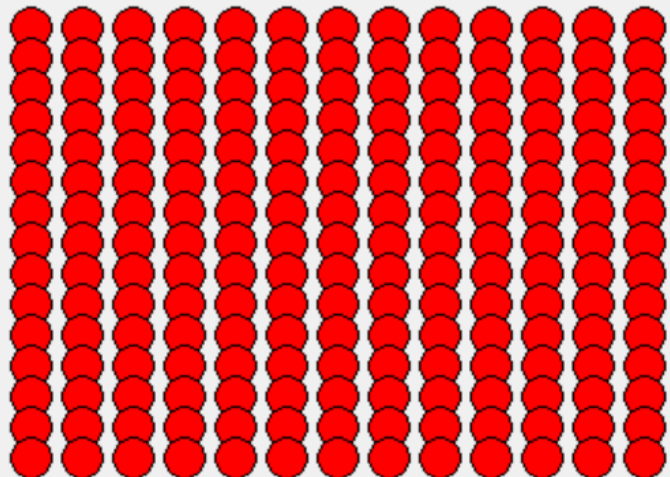
A HÍVÁSI MECHANIZMUST A ZOPIS_HVIEZDIČEK(30) HÍVÁS SZEMLÉLTETI:
A HÍVÁS VISSZATÉRÉSI CÍME MEGJEGYZI
LÉTREJÖN EGY ÚJ VÁLTOZÓSZÁM (FORMÁLIS PARAMÉTER), ÉS
HOZZÁRENDELŐDIK A TÉNYLEGES 30-AS PARAMÉTER ÉRTÉKE
A FÜGGVÉNYDEFINÍCIÓBAN (FÜGGVÉNYTÖRZSBEN) SZEREPLŐ ÖSSZES
UTASÍTÁS VÉGREHAJTÁSRA KERÜL
A FÜGGVÉNY VÉGREHAJTÁSA SORÁN LÉTREHOZOTT ÖSSZES VÁLTOZÓ
TÖRLŐDIK
A VEZÉRLÉS VISSZAKERÜL ODA, AHOL A FÜGGVÉNYT MEGHÍVTÁK
ÍRJUNK EGY ÚJ RED_CIRCLE() FÜGGVÉNYT, AMINEK KÉT PARAMÉTERE LESZ: A
KÖR KÖZÉPPONTJÁNAK KOORDINÁTÁI. A FÜGGVÉNY 10 SUGARÚ KÖRT RAJZOL
A MEGADOTT KÖZÉPPONTTAL:
DEF CERVENY_KRUH(X, Y):
CANVAS.CREATE_OVAL(X - 10, Y - 10, X + 10, Y + 10, FILL='RED'), ÉS
NEVEZHETJÜK PÉLDÁUL ÍGY:

```
IMPORT TKINTER DEF CERVENY_KRUH(X, Y):  
    CANVAS.CREATE_OVAL(X - 10, Y - 10, X + 10, Y + 10, FILL='PIROS')
```

```
CANVAS = TKINTER.CANVAS() CANVAS.PACK()
```

```
X TARTOMÁNYBAN (30, 350, 25):  
Y TARTOMÁNYBAN (20, 240, 15):  
PIROS_KÖR(X, Y)
```

```
TKINTER.MAINLOOP()
```





ÓRATERV



TÁRGY: PROGRAMOZÁS

ÓRA TÉMA: PROGRAMOZÁS PYTHONBAN – ALPROGRAMOKAT KÉSZÍTÜNK.

TANULÁSI CÉLKITÜZÉSEK:

- **1. ISMERJE MEG A SZUBRUTINOK LÉTREHOZÁSÁNAK ALAPELVEIT PYTHONBAN A TKINTER KÖNYVTÁR HASZNÁLATÁVAL**
- **2. MEGTANULNI A FÜGGVÉNYEK PARAMÉTEREKKEK VALÓ HASZNÁLATÁT PYTHONBAN.**

ESZKÖZÖK:
EGY SZÁMÍTÓGÉP PYTHONNAL ÉS A TKINTER KÖNYVTÁRRAL.
PYTHON MINTA TANKÖNYV MUNKALAP GYAKORLATOKKAL

HOOK/BEVEZETÉS: KÜLÖNBÖZŐ SZÍNEK HASZNÁLATA RAJZOLÁSKOR:
A FÜGGVÉNY EGY ELNEVEZETT PARANCSBLOKK (SZUBRUTIN). EGY SPECIÁLIS KONSTRUKCIÓVAL ÍRJUK LE (DEFINIÁLJUK):

- **CSAK AKKOR HÍVHATJUK MEG A FÜGGVÉNYT, HA A PYTHON MÁR ISMERI A DEFINÍCIÓJÁT**

ÓRA FELÉPÍTÉSE: CSATLAKOZÁS (4 PERC)

MEGTANULTA A SZINTAKTIKAI SZABÁLYOKAT

- **AZ ALPROGRAM DEFINÍCIÓ ÍRÁSÁNAK MÓDJA: FEJLÉC ÉS TÖRZS AZ ALPROGRAM TÖRZSÉBEN LÉVŐ ÖSSZES PARANCS ELTOLÁSA A BAL MARGÓTÓL ALPROGRAMHÍVÁS ÍRÁSÁNAK MÓDJA (ÜRES ZÁRÓJELEK AZ ALPROGRAM NEVE UTÁN)**

BESZÉLGETÉS A TANULÓKKAL A SZUBRUTINOK HASZNÁLATÁNAK ELŐNYEIRŐL PYTHONBAN: – AZ ALPROGRAMOK HASZNÁLHATJÁK A CANVAS GLOBÁLIS VÁLTOZÓT

AZ ALPROGRAM BEJEGYZÉS FORMÁZÁSÁKOR A <TAB> ÉS AZ <ENTER> BILLENTYŰKET HASZNÁLJUK

- **KUTATÁS (12 PERC) – ÚJ PARANCSON KUTATÁSA – ALPROGRAMOK SEGÍTSÉGÉVEL A GRAFIKAI TERÜLETEN RAJZOLÁS TÖRTÉNIK. MOST OLYAN SZUBRUTINOKAT FOG LÉTREHOZNI, AMELYEK A GRAFIKUS ABLAKBAN RAJZOLNAK.**

PREZENTÁCIÓ (15 PERC) – A PROBLÉMA VIZSGÁLATA CÍMŰ RÉSZ ISMERETEINEK ÖSSZEFoglalása ÉS MAGYARÁZATA HOZZON LÉTRE EGY ÚJ KRESBA_PODPROGRAM.PY PROGRAMOT, ÉS PRÓBÁLJA MEG:

- **IMPORT TKINTER**
- **VÁSZON = TKINTER.CANVAS()**
- **CANVAS.PACK()**
- **DEF SZÉKEK ():**
- **CANVAS.CREATE_RECTANGLE(10, 20, 30, 40, FILL='PIROS')**
- **SZÉK()**
- **EBBEN A FELADATBAN AZ ALPROGRAM TÖRZSE EGYETLEN PARANCSOT TARTALMAZ – TÉGLALAP RAJZOLÁSÁRA.**
- **KIDOLGOZÁS (9 PERC) – A HALLGATÓK TÁMOGATÁSA PYTHON ALPROGRAMOK TOVÁBBI ÖNÁLLÓ LÉTREHOZÁSÁBAN**
- **ÉRTÉKELÉS (5 PERC) – AZ ÖNÉRTÉKELŐ TESZT MEGOLDÁSA, A VÁLASZOK MEGBESZÉLÉSE.**

EXPOZÍCIÓS FÁZIS – ÚJ MEGRENDELÉSEK BEMUTATÁSA

– AZ ALAPFOGALMAKAT RÖVIDEN MEGJEGYZEM A TÁBLÁN

ÓRATERV



5. TANULÓI FELADATLAP

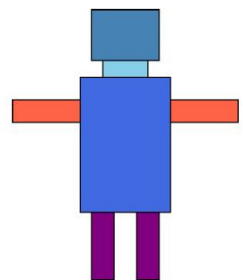
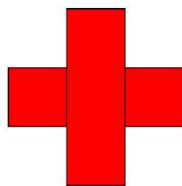
1. Feladat: Hozzon létre egy új `kriz.py` programot és definiáljon benne egy `kriz` alprogramot, amely meghívásakor piros keresztet húz:

```
import tkinter
canvas = tkinter.Canvas() canvas.pack() def kriz():
canvas.create_rectangle(150 - 90, 100 - 30, 150 + 90, 100 + 30, fill='piros') canvas.create_rectangle(150 - 30,
100 - 90, 150 + 30, 100 - 30, fill='90 ') kriz()
```

Ebben a feladatban az alprogram törzse két utasításból álló sorozatot tartalmaz. Minden alkalommal, amikor ezt az alprogramot hívják, először az első téglalapot, majd a második téglalapot rajzolják át rajta.

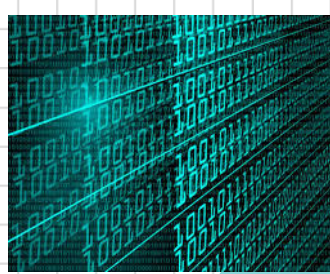
2. Feladat: Hozzon létre egy új `robot.py` programot, amely képes lesz robotot rajzolni. A programban négy szubrutin lesz – fej, karok, lábak, test –, és mindegyik képes lesz rajzolni a robot egy-egy részét. Ha a következő sorrendben hívja őket: `head()` `arms()` `legs()` `body()` az egész robot kirajzolódik.

```
import tkinter
canvas = tkinter.Canvas() canvas.pack() def hlava():
canvas.create_rectangle(160, 40, 200, 100, fill='sky blue') canvas.create_rectangle(150, 10, 210, 55,
fill='acélkék') def telo():
canvas.create_rectangle(140, 70, 220, 190, fill='királykék') def ruce():
canvas.create_rectangle(80, 90, 280, 110, fill='paradicsom') def nohy():
canvas.create_rectangle(150, 160, 170, 250, fill='lila') canvas.create_rectangle(190, 160, 210, 250, fill='lila')
hlava() ruce() nohy() telo()
```





ÓRATERV



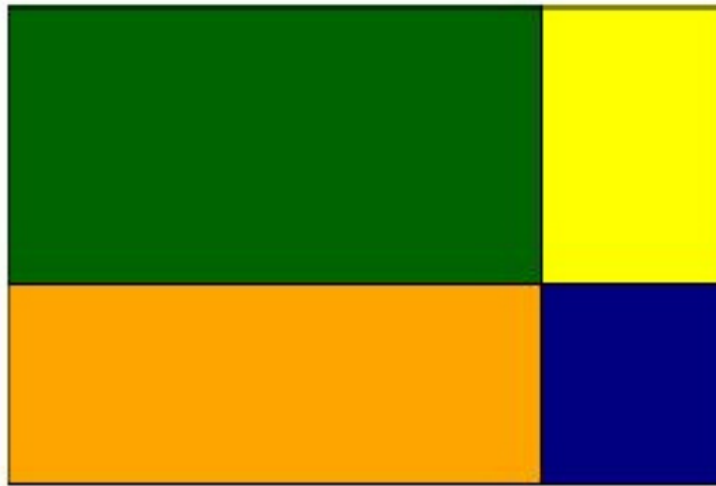
5. TANULÓI FELADATLAP

3. Feladat: Készíts egy új sportová_vlajka.py programot, amely kirajzolja osztályod sportzászlóját.

A zászló négy színes téglalapról áll, amelyek egyetlen ponton érintik egymást, mint az alábbi képen:

A program véletlenszerűen választja ki az x, y koordinátákat, amelyek mind a négy téglalap érintkezési pontját jelentik.

A zászló külső mérete 300x200 legyen; saját belátása szerint határozza meg a zászló színeit.



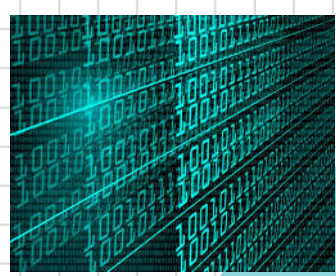
```
import tkinter
import random
canvas = tkinter.Canvas()
canvas.pack()
x = random.randint(50, 250)
y = random.randint(50, 150)
canvas.create_rectangle(10, 10, x, y, fill='sötétzöld')
canvas.create_rectangle(x, 10, 310, y, fill='sárga')
canvas.create_rectangle(10, y, x, 210, fill='narancs')
canvas.create_rectangle(x, y, 310, 210, fill='navy')
```

A téglalap külső csúcsai tetszőlegesen választhatók. Megoldásunkban a bal felső csúcsnak vannak koordinátái [10, 10] és mivel a téglalap mérete 300x200 legyen, a szemközti csúcsnak (jobbra lent) vannak koordinátái [310, 210].

A véletlenszerűen kiválasztott belső pontot úgy generáltuk, hogy az legalább 50-re legyen a téglalap éleitől, így a négy terület mindegyikének mérete legalább 50x50 lesz.



MELLÉKLET

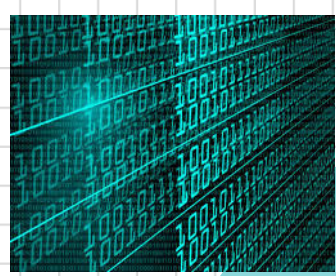


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL

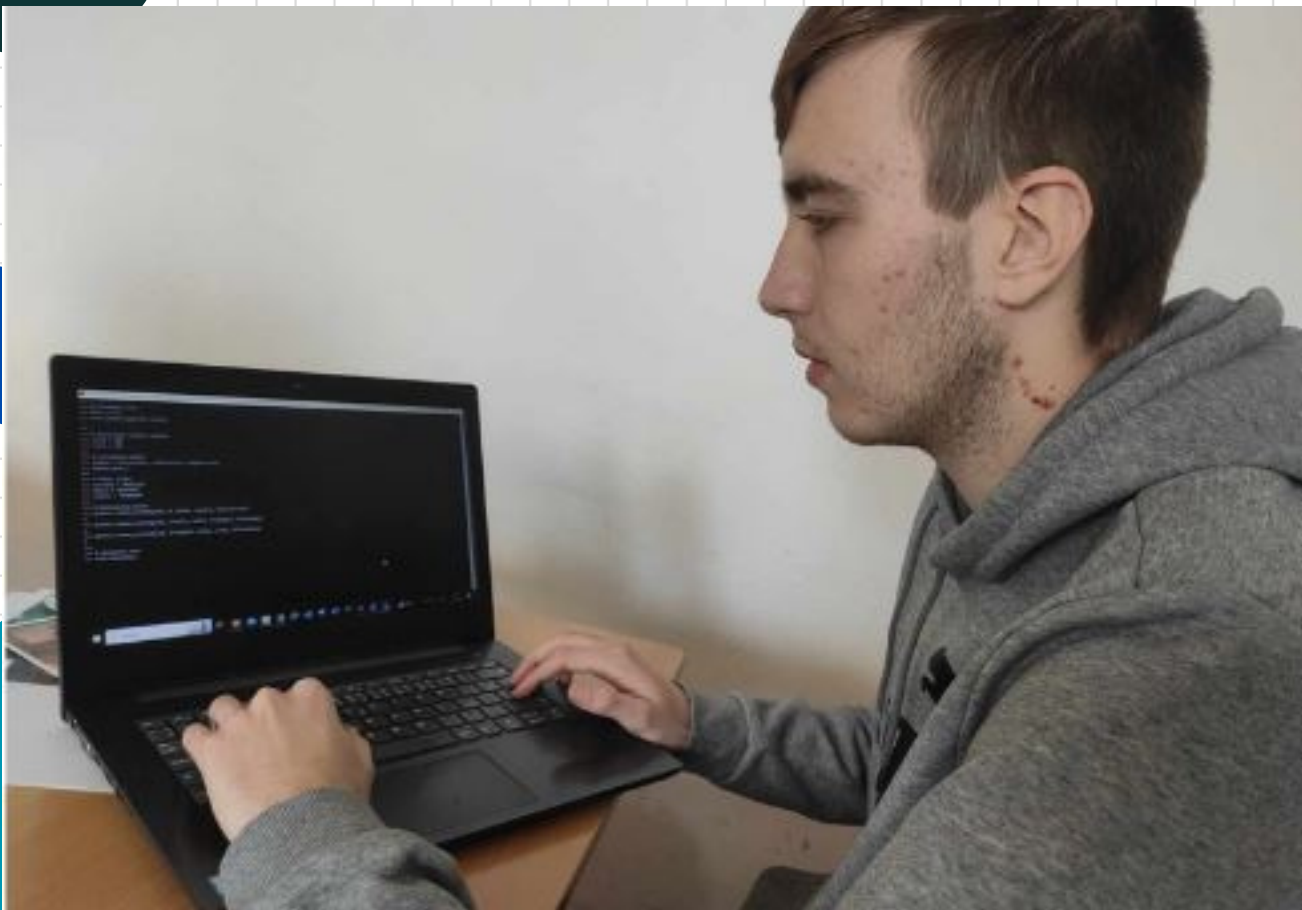




MELLÉKLET

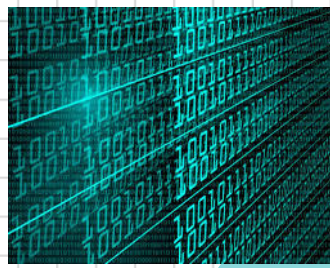


FOTÓK AZ ÓRA MEGVALÓSÍTÁSAIBÓL





MELLÉKLET



DIÁKOK KIMENETE

MÓDOSÍTSA A RANDOM_SQUARES.PY PROGRAMOT ÚGY, HOGY A NÉGYZETEK NE CSAK VÉLETLENSZERŰ HELYEKRE LEGYENEK RAJZOLVA, HANEM ÚGY IS, HOGY MINDEN NÉGYZET VÉLETLENSZERŰ MÉRETŰ LEGYEN 10 ÉS 100 KÖZÖTT:

```
DEF RANDOM_SQUARE():
```

```
X = RANDOM.RANDINT(10, 300) Y = RANDOM.RANDINT(10, 200) A =  
RANDOM.RANDINT(10, 100) CANVAS.CREATE_RECTANGLE(X, Y, X + A, Y + A,  
FILL='NARANCS ')
```

```
IMPORT TKINTER IMPORT RANDOM CANVAS = TKINTER.CANVAS()
```

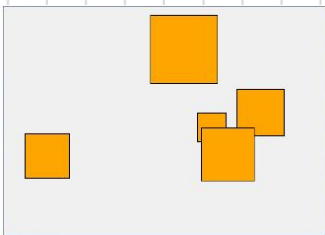
```
CANVAS.PACK() DEF RANDOM_THURSDAY():
```

```
X = RANDOM.RANDINT(10, 300) Y = RANDOM.RANDINT(10, 200) A =  
RANDOM.RANDINT(10, 100) CANVAS.CREATE_RECTANGLE(X, Y, X + A, Y + A,  
FILL='NARANCS ') DEF RANDOM_OBDLŽNIK():
```

```
X = RANDOM.RANDINT(10, 300) Y = RANDOM.RANDINT(10, 200) A =  
RANDOM.RANDINT(10, 100) B = RANDOM.RANDINT(10, 100)
```

```
CANVAS.CREATE_RECTANGLE(X, Y, X + A, Y + B, FILL='LIME GREEN')
```

```
RANDOM_OBD() RANDOM_OBD() RANDOM_OBD() RANDOM_OBD() RANDOM_OBD()  
RANDOM_OBD() RANDOM_OBD() RANDOM_OBD() RANDOM_OBD() RANDOM_OBD()
```



Óraterv

Bevezető leckék sorozat a programozásról C nyelven

Téma: Kezdő lépések a C-vel

1. lecke

A lecke fókusza és céljai:

Ez a lecke bemutatja a tanulókat a C programozási nyelvvel, annak történetével és alapvető szintaxisával. A lecke végére a tanulóknak képesnek kell lenniük egy egyszerű "Hello, World!" program

Szükséges anyagok:

C-fordítóval rendelkező számítógépek (pl. GCC)
Projektor és képernyő
Tábla és markerek
Az alapvető C szintaxis és szerkezet kiosztásai

Tanulási célok:

Mutasd be C
Mutassa be a Jegyzetömböt és a GCC-t. Írja:
„Hello, World!” program.

Szerkezet/tevékenység:

Bevezetés a C-be (15 perc)

Beszélgetés (5 perc): Kezdje a C programozási nyelv rövid történetével. Beszéljétek meg fontosságát, és azt, hogyan fektette le számos modern programozási nyelv alapjait. Emelje ki a kulcsfontosságú alkalmazásokat, például az operációs rendszereket (például UNIX, Linux) és a beágyazott rendszereket.

Prezentáció (10 perc): Használjon kivetítőt, hogy mutassa meg, hol használják ma a C-t, és beszélje meg, miért elengedhetetlen a C tanulása. Adjon példákat népszerű C nyelven írt szoftverekre és rendszerekre.

Alapszintaxis és szerkezet (20 perc)

Magyarázat (10 perc): Ismertesse meg egy alap C program felépítését! Magyarázza el az #include, a main() függvény és az alapvető szintaktikai szabályok szerepét. Adjon példákat a szintaktikai hibákra és azok elkerülésére.

Példa (10 perc): Írjon egy egyszerű "Hello, World!" program a táblán vagy a projektor képernyőjén. Bontsa fel a kód minden sorát, magyarázza el a célját. Bátorítsa a tanulókat, hogy tegyenek fel kérdéseket a kód minden egyes részével kapcsolatban.

Gyakorlati tevékenység (20 perc)

Tevékenység (15 perc): Kérd meg a tanulókat, hogy írják meg saját "Hello, World!" programot a számítógépükön. Vezesse őket a fejlesztői környezet beállításán (szükség esetén telepítsen egy C-fordítót, például a GCC-t). Győződjön meg arról, hogy minden tanuló sikeresen tudja összeállítani és futtatni programját.

Peer Review (5 perc): Párosítsd a tanulókat, hogy átnézzék egymás kódját. Bátorítsa őket, hogy beszéljenek meg minden felmerülő problémát vagy kérdést.

Kérdések és válaszok és összefoglaló (5 perc)

Kérdések és válaszok (3 perc): Nyissa meg a teret, ha bármilyen kérdése van a leckével kapcsolatban. Vegye figyelembe a gyakori problémákat vagy zavaró pontokat, amelyek a tanulóknál felmerülhetnek.

Összefoglaló és házi feladat (2 perc): Foglalja össze a lecke legfontosabb pontjait. Házi feladathoz kérje meg a tanulókat, hogy keressenek online egy egyszerű C-programot, módosítsák kissé (például módosítsák a kimeneti üzenetet), és futtassák azt.

Óraterv – szóróanyag

Bevezető leckék sorozat a programozásról C nyelven

Téma: Kezdő lépések a C-vel

1. tájékoztató

1. Bevezetés a C-be

Főbb pontok:

A C egy alapvető programozási nyelv, amelyet számos alkalmazásban használnak, beleértve az operációs rendszereket és a beágyazott rendszereket.

A C megértése segít más programozási nyelvek, például C++, Java és Python tanulásában.

Példa programok:

„Hello, World!": A legegyszerűbb C program.

```
#include <stdio.h>

int main() { printf("Szervusz világ!\n");
    visszatérés 0;
}
```

2. Alapvető szintaxis és szerkezet

Főbb pontok:

Minden C programnak tartalmaznia kell az `#include <stdio.h>`-t és egy `main()` függvényt.

A `main()` függvény bármely C program kezdőpontja.

Tevékenység:

Írja meg és futtassa a "Hello, World!" program.

Határozza meg a program minden sorának célját.

3. Házi feladat:

Keressen egy egyszerű C programot online, módosítsa kissé (például módosítsa a kimeneti üzenetet), és futtassa.

Óraterv

Bevezető leckék sorozat a programozásról C nyelven

Témakör: Változók és adattípusok

2. lecke

A lecke fókusza és céljai:

Tanítsa meg a tanulókat a változókra, adattípusokra és alapvető bemeneti/kimeneti műveletekre C nyelven.

Szükséges anyagok:

C-fordítóval rendelkező számítógépek (pl. GCC)
Projektor és képernyő
Tábla és markerek
Kiosztók változódeklaráció példáival és
általános bemeneti/kimeneti műveletekkel

Tanulási célok:

Ismerje meg a változókat/adattípusokat írjon programot a printf segítségével

Szerkezet/tevékenység:

Az előző lecke áttekintése (5 perc)

Beszélgetés (5 perc): Foglalja össze a C program alapvető felépítését és a "Hello, World!" program az előző leckéből. Hangsúlyozza a szintaxis megértésének fontosságát, mielőtt bonyolultabb fogalmakra térne át.

Bevezetés a változókba és adattípusokba (15 perc)

Magyarázat (10 perc): Mutassa be a változók fogalmát az adatok helyőrzőjeként. Magyarázza el a különböző adattípusok (pl. int, float, char, double) közötti különbséget és tipikus felhasználásukat.

Példa (5 perc): Írjon egy egyszerű programot, amely különféle típusú változókat deklarál, és a printf segítségével adja ki az értékeket. Mutassa be, hogyan rendelhet értékeket a változókhoz, és hogyan hajthat végre alapvető aritmetikai műveleteket.

1. gyakorlati tevékenység: Változó nyilatkozat (15 perc)

Tevékenység (10 perc): A tanulók deklarálják a különböző adattípusú változókat egy új programban. Kérje meg őket, hogy hajtsák végre az alapvető műveleteket, például az összeadást és a szorzást, majd nyomtassák ki az eredményeket.

Peer Review (5 perc): A tanulók kicserélik a programjaikat, és ellenőrzik, hogy a változók deklarációi és műveletei helyesek-e. Bátorítsd őket, hogy beszéljék meg a különbségeket vagy hibákat.

Bevezetés az alapvető bemenet/kimenetbe (10 perc)

Magyarázat (5 perc): Mutassa be a scanf funkciót a felhasználói bevitel fogadására. Magyarázza el, hogyan működik a printf-vel együtt interaktív programok létrehozása.

Példa (5 perc): Írjon egy programot, amely megkérdezi a felhasználótól az életkorát, majd a scanf és a printf segítségével visszaírja.

2. gyakorlati tevékenység: Interaktív program (15 perc)

Tevékenység (10 perc): A tanulók módosítják korábbi programjukat, hogy felhasználói bevitelt is tartalmazzanak. Például kérjen a felhasználótól két számot, hajtsa végre egy műveletet (például összeadás), és nyomtassa ki az eredményt.

Támogatás (5 perc): Körbejárja a termet, hogy segítsen a tanulóknak az esetleges szintaktikai vagy logikai hibákban.

Kérdések és válaszok és összefoglaló (5 perc)

Kérdések és válaszok (3 perc): A tanulók kérdéseket tehetnek fel a változókkal, adattípusokkal vagy bemeneti/kimeneti műveletekkel kapcsolatban.

Összefoglaló és házi feladat (2 perc): Foglalja össze a leckét. Rendeljen házi feladatot, amelyben a tanulók olyan programot készítenek, amely három különböző adattípushoz veszi a felhasználói bevitelt, és egy formázott karakterláncban adja ki azokat.

Óraterv – szóróanyag

Bevezető leckék sorozat a programozásról C nyelven

Témakör: Változók és adattípusok

2. tájékoztató

1. Bevezetés a változókba

Főbb pontok:

A változók az adatok helyőrzői. C-ben minden változónak van egy adott adattípusa.

Gyakori adattípusok: int (egész), float (lebegőpontos szám), char (karakter), double (kettős pontosságú lebegőpontos).

Példa kód:

```
int életkor = 25;  
úszómagasság = 5,9;  
char kezdeti = 'A';  
dupla távolság = 12345,67;
```

2. Bemenet és kimenet

Főbb pontok:

Használja a printf-et az adatok kiadásához és a scanf-et a bemenethez.

Példa kód:

```
int kor;  
printf("Adja meg életkorát: ");  
scanf("%d", &age);  
printf("Beírtad: %d\n", életkor);
```

3. Házi feladat:

Hozzon létre egy programot, amely három különböző adattípushoz veszi a felhasználói bevitelt, és formázott karakterláncban adja ki azokat.

Óraterv

Bevezető leckék sorozat a programozásról C nyelven

Téma: Vezérlési struktúrák

3. lecke

A lecke fókuszja és céljai:

Ez a lecke bevezeti a tanulókat a C nyelv struktúráinak vezérlésébe, beleértve az if-else utasításokat és ciklusokat. Az óra végére a tanulónak képesnek kell lenniük arra, hogy ezeket a struktúrákat használják programjaik áramlásának szabályozására.

Szükséges anyagok:

C-fordítóval rendelkező számítógépek (pl. GCC)
Projektor és képernyő
Tábla és markerek
Kiosztóanyagok példákkal if-else utasításokra és ciklusokra

Tanulási célok:

Ismerje meg az alapvető bemeneti/kimeneti adatokat. Ismerje meg az if/if-else utasításokat

Szerkezet/tevékenység:

Az előző lecke áttekintése (5 perc)

Beszélgetés (5 perc): Foglalja össze a változók, az adattípusok és az alapvető input/output fogalmát. Emelje ki, hogyan fogják ezeket az elemeket használni a vezérlőszervezetekkel együtt.

Az If-Else nyilatkozatok bemutatása (15 perc)

Magyarázat (10 perc): Ismertesse az if-else utasítások szintaxisát és használatát. Beszéljétek meg a feltételes logika fogalmát, és azt, hogyan teszi lehetővé a programok számára a döntéshozatalt.

Példa (5 perc): Írjon programot, amely a felhasználótól kér egy számot, és kiírja, hogy az páros vagy páratlan.

Magyarázza el, hogyan használható az if-else utasítás a feltételek ellenőrzésére.

1. gyakorlati tevékenység: Feltételes logika (15 perc)

Tevékenység (10 perc): A tanulók olyan programot írnak, amely a felhasználói bevitelt veszi, és if-else utasításokat használ a döntések meghozatalához (pl. ellenőrzi, hogy egy szám pozitív, negatív vagy nulla).

Peer Review (5 perc): Kérd meg a tanulókat, hogy cseréljék ki programjaikat, és nézzék át a használt logikát. Bátorítsa őket, hogy teszteljék egymás programjait különböző bemenetekkel.

Bevezetés a Loopsba (15 perc)

Magyarázat (10 perc): For és while ciklusok bevezetése. Magyarázza el, hogyan használják a ciklusokat a műveletek megismétlésére egy programban, és beszélje meg azokat a forgatókönyveket, ahol az egyes huroktípusok megfelelőek.

Példa (5 perc): Írjon programot, amely 1-től 10-ig számokat ír ki a for ciklus segítségével. Ezután írja át egy while ciklus segítségével.

2. gyakorlati tevékenység: Szervezetek hurkolása (15 perc)

Tevékenység (10 perc): A tanulók olyan programot írnak, amely ciklus segítségével számítja ki egy szám faktoriálisát, vagy szorzótáblát állít elő.

Támogatás (5 perc): Segítségnyújtás a tanulónak ciklusai hibakeresésében és a logika helyességének biztosításában.

Kérdések és válaszok és összefoglaló (5 perc)

Kérdések és válaszok (3 perc): A vezérlési struktúrákkal kapcsolatos kérdések megválaszolása. Beszéljétek meg a gyakori hibákat, például a végtelen ciklusokat vagy az if-else nem megfelelő használatát.

Összefoglaló és házi feladat (2 perc): Foglalja össze a leckét. Rendeljen házi feladatot, amelyben a tanulók egy programot írnak if-else utasítások és ciklusok felhasználásával egy egyszerű számológép létrehozásához.

Óraterv – szóróanyag

Bevezető leckék sorozat a programozásról C nyelven

Téma: Vezérlési struktúrák

3. tájékoztató

1. If-Else nyilatkozatok

Főbb pontok:

Az if-else utasítások lehetővé teszik a program számára, hogy feltételek alapján döntsön.

Példakód:

```
int szám;  
printf("Írjon be egy számot: ");  
scanf("%d", &szám);  
  
if (szám % 2 == 0) { printf("A szám páros.\n");  
} else { printf("A szám páratlan.\n");  
}
```

2. Hurok

Főbb pontok:

A hurok lehetővé teszi a program számára a műveletek megismétlését. A közös hurok for and while.

Példakód:

```
for (int i = 1; i <= 10; i++) {  
    printf("%d\n", i);  
}
```

3. Házi feladat:

Írjon egy programot if-else utasítások és ciklusok felhasználásával egy egyszerű számológép létrehozásához.
A bekezdés szövege

Óraterv

Bevezető leckék sorozat a programozásról C nyelven

Téma: Funkciók

4. lecke

A lecke fókuszja és céljai:

Ez a lecke megtanítja a tanulókat a C-beli függvényekről, azok fontosságáról, valamint létrehozásukról és használatukról. Az óra végére a tanulóknak képesnek kell lenniük az alapvető funkciók meghatározására és meghívására a programjukban.

Szükséges anyagok:

C-fordítóval rendelkező számítógépek (pl. GCC)
Projektor és képernyő
Tábla és markerek

- Kiosztók függvénydefiníciókra és függvényhívásokra vonatkozó példákkal

Tanulási célok:

Ismerje meg az alapvető funkciók meghatározását a C gyakorlati gyakorlatban

Szerkezet/tevékenység:

Az előző lecke áttekintése (5 perc)

Megbeszélés (5 perc): Gyorsan áttekintheti a vezérlési struktúrákat, hangsúlyozva, hogy a funkciók hogyan segítik a kódok rendszerezését és újrafelhasználását bonyolultabb programokban.

Bevezetés a funkciókba (15 perc)

Magyarázat (10 perc): Határozzon meg funkciókat, magyarázza el a kód kisebb, újrafelhasználható blokkokra való felosztásának célját. Beszélje meg egy függvény szintaxisát, beleértve a visszatérési típusokat, a függvényneveket és a paramétereket.

Példa (5 perc): Írjon egy egyszerű függvényt, amely kiszámítja egy szám négyzetét. Magyarázza el, hogyan definiálható, hívható meg a függvény, és hogyan ad vissza értéket.

1. gyakorlati tevékenység: írási funkciók (15 perc)

Tevékenység (10 perc): A tanulók írják meg saját függvényüket, amely egy adott feladatot hajt végre, például a Celsius-fokozatot Fahrenheitre konvertálja. Győződjön meg arról, hogy értik a paraméterek átadását és az értékek visszaadását.

Peer Review (5 perc): A tanulók kicserélik a programjaikat és tesztelik egymás funkcióit.

A funkciók bővítése (15 perc)

Magyarázat (10 perc): Beszélje meg, hogyan lehet több funkciót használni egyetlen programban. Magyarázza el, hogyan hívhatnak a függvények más függvényeket, és hogyan kezelhetik a függvényfüggőségeket.

Példa (5 perc): Írjon programot, amely két funkciót tartalmaz: az egyiket a bemenet érvényesítésére, a másikat pedig a számítás elvégzésére. Mutassa be, hogyan működnek együtt ezek a funkciók.

2. gyakorlati tevékenység: Funkciók interakciója (15 perc)

Tevékenység (10 perc): A tanulók olyan programot készítenek, amely legalább két funkciót együtt használ, például egy egyszerű számológépet, amely a műveletek végrehajtása előtt ellenőrzi a bevitelt.

Támogatás (5 perc): Útmutatást ad a hibakereséshez és a funkciók hatóköreinek megértéséhez.

Kérdések és válaszok és összefoglaló (5 perc)

Kérdések és válaszok (3 perc): válaszoljon a funkciókkal kapcsolatos kérdésekre. Beszélje meg a gyakori problémákat, például a funkció túlterhelését és az elnevezési ütközéseket.

Összefoglaló és házi feladat (2 perc): Foglalja össze a leckét. Házi feladathoz kérje meg a tanulókat, hogy módosítsák a programot úgy, hogy az tartalmazza a bemeneti, feldolgozási és kimeneti funkciókat.

Óraterv – szóróanyag

Bevezető leckék sorozat a programozásról C nyelven

Téma: Funkciók

4. tájékoztató

1. Bevezetés a funkciókba

Főbb pontok:

A függvények olyan újrafelhasználható kódblokkok, amelyek meghatározott feladatokat hajtanak végre.

A függvényeknek van visszatérési típusa, neve és paraméterei.

Példakód:

```
int square(int num) { return num * num;
}

int main() { int eredmény = négyzet(5);
    printf("Az 5-ös négyzet %d\n", eredmény);
    visszatérés 0;
}
```

2. Írási és hívási funkciók

Tevékenység:

Írjon egy függvényt, amely a Celsius-t Fahrenheitre konvertálja, és meghívja a `main()`-ban.

3. Házi feladat:

Módosítsa a programot úgy, hogy tartalmazza a bemeneti, feldolgozási és kimeneti funkciókat.

Óraterv

Bevezető leckék sorozat a programozásról C nyelven

Téma: Tömbök és karakterláncok

5. lecke

A lecke fókuszja és céljai:

Ez a lecke bemutatja a tanulóknak a tömböket és karakterláncokat C nyelven, elmagyarázva, hogyan deklarálják, inicializálják és használják őket a programokban. Az óra végére a tanulóknak képesnek kell lenniük a tömbök és karakterláncok manipulálására az alapvető C programokban.

Szükséges anyagok:

C-fordítóval rendelkező számítógépek (pl. GCC)
Projektor és képernyő
Tábla és markerek

- Kiosztók tömb- és karakterlánc-műveletek példáival

Tanulási célok:

Tömbök és karakterláncok bemutatása
Gyakorlati gyakorlat

Szerkezet/tevékenység:

Az előző lecke áttekintése (5 perc)

Beszélgetés (5 perc): Ismételve meg a függvények fogalmát és használatát a C-ben. Emelje ki, hogyan egyszerűsíthetik le a függvények a tömbökkel és karakterláncokkal való munkát.

Bevezetés a tömbökbe (15 perc)

Magyarázat (10 perc): Magyarázza el, mik azok a tömbök, beleértve a szintaxisukat és a deklarálásuk és inicializálásuk módját. Beszélje meg a különböző típusú tömböket, például az egydimenziós és többdimenziós tömböket.

Példa (5 perc): Írjon programot, amely egy tömb segítségével tárolja és összegzi a számok listáját. Magyarázza el a program minden lépését, a deklarációtól az iterációig és az összegzésig.

1. gyakorlati tevékenység: tömbmanipuláció (15 perc)

Tevékenység (10 perc): A tanulók írnak egy programot, amely egy tömb segítségével tárolja a felhasználói adatokat (pl. osztályzatok listáját), majd kiszámítja az átlagot.

Támogatás (5 perc): Segítsen a tanulóknak olyan problémák elhárításában, mint például a határokon túli tömbhibák és a helytelen indexelés.

Bevezetés a vonósokhoz (15 perc)

Magyarázat (10 perc): Beszélje meg, hogyan jelennek meg a karakterláncok C-ben karaktertömbök használatával. Ismertesse a karakterláncok deklarálását, inicializálását és kezelését.

Példa (5 perc): Írjon programot, amely a felhasználótól veszi a karakterlánc bevitelét, és fordítva kiírja. Járassd végig a tanulókat a karakterlánc-manipuláció logikáján.

2. gyakorlati tevékenység: Húrműveletek (15 perc)

Tevékenység (10 perc): A tanulók írnak egy programot, amely bekéri a felhasználó nevét, majd különféle műveleteket hajt végre, például megszámolja a karakterek számát, vagy megkeresi az első magánhangzót.

Támogatás (5 perc): Segítségnyújtás a karakterlánc-lezárással és a karakterlánc-manipuláció gyakori hibáival kapcsolatban.

Kérdések és válaszok és összefoglaló (5 perc)

Kérdések és válaszok (3 perc): A tanulók kérdéseket tehetnek fel a tömbökről és karakterláncokról. Beszéljétek meg ezen fogalmak megértésének fontosságát a fejlettebb programozáshoz.

Összefoglaló és házi feladat (2 perc): Foglalja össze a leckét. Rendeljen házi feladatot, amelyben a tanulók olyan programot írnak, amely tömböket és karakterláncokat is használ egy egyszerű névjegyzék kezeléséhez.

Óraterv – szóróanyag

Bevezető leckék sorozat a programozásról C nyelven

Téma: Tömbök és karakterláncok

5. számú tájékoztató

1. Bevezetés a tömbökbe

Főbb pontok:

A tömbök azonos típusú változók gyűjteményei.

A tömbök nulla indexeltek; az első elem a 0 indexnél van.

Példakód:

```
int számok[5] = {1, 2, 3, 4, 5};  
printf("Első szám: %d\n", számok[0]);
```

2. Bevezetés a hűrokba

Főbb pontok:

A C karakterláncai olyan karaktertömbök, amelyek null karakterrel végződnek `'\0'`.

Példakód:

```
char name[20];  
printf("Írja be a nevét: ");  
scanf("%s", név);  
printf("Helló, %s!\n", név);
```

3. Házi feladat:

Írjon egy programot, amely tömböket és karakterláncokat is használ egy egyszerű névjegyzék kezeléséhez.



ÓRATERV



SUBJECT: PROGRAMMING AND CODING

TÉMA: BEVEZETŐ LECKÉK A PYTHON PROGRAMOZÁSI NYELVRŐL – 1. LECKE

LECKE FÓKUSZA ÉS CÉLJAI:

1. EZ A LECKE BEVEZETI A TANULÓKAT A PYTHON PROGRAMOZÁSI NYELV ALAPJAIBA, BELEÉRTVE A TÖRTÉNETÉT, A NYELV SAJÁTOSságAIT ÉS AZ ELSŐ PYTHON PROGRAM MEGÍRÁSÁT.

2. A LECKE VÉGÉRE A TANULÓKNAK KÉPESNEK KELL LENNIÜK MEGÉRTENI A PYTHON NYELV ALAPVETŐ SZINTAXISÁT, VALAMINT ELKÉSZÍTENI ÉS FUTTATNI EGY EGYSZERŰ "HELLO, WORLD!" PROGRAMOT.

SZÜKSÉGES ANYAGOK:

- SZÁMÍTÓGÉPEK PYTHON TELEPÍTÉSSEL (PL. VS CODE, PYCHARM, VAGY MÁS FEJLESZTŐI KÖRNYEZET)
- PROJEKTOR ÉS KÉPERNYŐ
- TÁBLA ÉS MARKEREK
- AZ ALAPVETŐ PYTHON SZINTAXIS ÉS SZERKEZET KIOSZTÁSAI

1. BEVEZETÉS A PYTHONBA (15 PERC)

BESZÉLGETÉS (5 PERC): KEZDJE A PYTHON PROGRAMOZÁSI NYELV RÖVID TÖRTÉNETÉVEL, AMELYET GUIDO VAN ROSSUM FEJLESZTETT KI AZ 1980-AS ÉVEK VÉGÉN. BESZÉLJEN A NYELV KIALAKULÁSÁRÓL ÉS ARRÓL, HOGYAN VÁLT NÉPSZERŰVÉ A KÖNNYŰ TANULHATÓSÁGÁNAK ÉS SZÉLES KÖRŰ ALKALMAZHATÓSÁGÁNAK KÖSZÖNHETŐEN. EMELJE KI, HOGY A PYTHON HOGYAN FEKTETTE LE A MODERN PROGRAMOZÁSI NYELVEK ALAPJAIT, ÉS HOGY KÜLÖNÖSEN NÉPSZERŰ A TUDOMÁNYOS SZÁMÍTÁSOK, A WEBFEJLESZTÉS ÉS AZ ADATFELDOLGOZÁS TERÜLETÉN.

PREZENTÁCIÓ (10 PERC): VETÍTSE KI A PYTHON JELENLEGI HASZNÁLATÁT KÜLÖNBÖZŐ IPARÁGAKBAN. BESZÉLJE MEG, MIÉRT ÉRDEMES MEGTANULNI PYTHONT, ÉS MUTASSON PÉLDÁKAT NÉPSZERŰ PYTHON-ALAPÚ ALKALMAZÁSOKRA, MINT PÉLDÁUL A YOUTUBE, INSTAGRAM, ÉS A NASA ADATELEMZÉSEI.

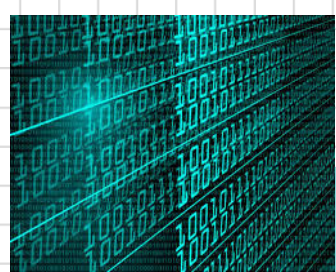
2. ALAPSZINTAXIS ÉS SZERKEZET (20 PERC)

MAGYARÁZAT (10 PERC): MUTASSA BE EGY EGYSZERŰ PYTHON PROGRAM SZERKEZETÉT. ISMERTESSE AZ IMPORT HASZNÁLATÁT, A MAIN() FÜGGVÉNY SZEREPÉT, ÉS A PYTHON SZINTAKTIKAI SAJÁTOSságAIT, PÉLDÁUL A BEHÚZÁSOK FONTOSságÁT. MAGYARÁZZA EL, HOGYAN KEZELI A PYTHON A KIFEJEZÉSEKET, ÉS MILYEN HIBÁKAT LEHET ELKÖVETNI. MUTASSON BE EGYSZERŰ HIBÁKAT, PÉLDÁUL AZ INDENTÁLÁS ELMULASZTÁSÁT, ÉS MAGYARÁZZA EL, HOGYAN LEHET EZEKET ELKERÜLNI.

PÉLDA (10 PERC): ÍRJON EGY "HELLO, WORLD!" PROGRAMOT A TÁBLÁN VAGY A PROJEKTOR KÉPERNYŐJÉN. RÉSZLETESEN MAGYARÁZZA EL A KÓD MINDEN SORÁNAK SZEREPÉT, BELEÉRTVE A PRINT() FÜGGVÉNY MŰKÖDÉSÉT, A SZTRINGEK KEZELÉSÉT ÉS A KIMENETI EREDMÉNY MEGÉRTÉSÉT. BÁTORÍTSA A TANULÓKAT, HOGY TEGYENEK FEL KÉRDÉSEKET A KÓD MINDEN RÉSZÉVEL KAPCSOLATBAN.



ÓRATERV



3. GYAKORLATI TEVÉKENYSÉG (20 PERC)

TEVÉKENYSÉG (15 PERC): KÉRJE MEG A TANULÓKAT, HOGY ÍRJÁK MEG SAJÁT "HELLO, WORLD!" PROGRAMJUKAT PYTHONBAN. SEGÍTSE ŐKET A FEJLESZTŐI KÖRNYEZET BEÁLLÍTÁSÁBAN (PÉLDÁUL AZ IDLE VAGY A PYCHARM HASZNÁLATÁVAL), ÉS ELLENŐRIZZE, HOGY MINDENKI SIKERESEN TUDJA FUTTATNI A PROGRAMJÁT. GYŐZŐDJÖN MEG ARRÓL, HOGY MINDEN TANULÓ MEGÉRTI A KÓD MŰKÖDÉSÉT ÉS A PYTHON KÖRNYEZETÉBEN VALÓ NAVIGÁCIÓT.

PEER REVIEW (5 PERC): PÁROSÍTSA A TANULÓKAT, HOGY ÁTNÉZZÉK EGYMÁS KÓDJÁT. BÁTORÍTSA ŐKET, HOGY BESZÉLJENEK MEG MINDEN FELMERÜLŐ KÉRDÉST VAGY PROBLÉMÁT, ÉS SEGÍTSENEK EGYMÁSNAK A KÓD JAVÍTÁSÁBAN.

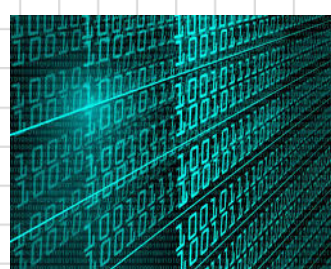
4. KÉRDÉSEK ÉS VÁLASZOK ÉS ÖSSZEFOGLALÓ (5 PERC)

KÉRDÉSEK ÉS VÁLASZOK (3 PERC): NYISSA MEG A TERET A TANULÓK KÉRDÉSEI SZÁMÁRA. BESZÉLJEN MEG GYAKORI PROBLÉMÁKAT VAGY ZAVAROS PONTOKAT, AMELYEK FELMERÜLHETTEK A LECKE SORÁN. ADJ LEHETŐSÉGET ARRA, HOGY A TANULÓK VISSZAJELZÉST ADJANAK A LECKE MEGÉRTÉSÉRŐL.

ÖSSZEFOGLALÓ ÉS HÁZI FELADAT (2 PERC): FOGLALJA ÖSSZE A LECKE LEGFONTOSABB PONTJAIT, KIEMELVE A PYTHON ALAPVETŐ SZINTAXISÁT ÉS A "HELLO, WORLD!" PROGRAM FONTOSSÁGÁT. HÁZI FELADATKÉNT KÉRJE MEG A TANULÓKAT, HOGY KERESSENEK EGY EGYSZERŰ PYTHON PROGRAMOT ONLINE, MÓDOSÍTSÁK KIMENETI ÜZENETÉT, ÉS FUTTASSÁK AZT. KÉRJE ŐKET, HOGY HOZZÁK EL A KÖVETKEZŐ ÓRÁRA A MÓDOSÍTOTT KÓDOT, ÉS KÉSZÜLJENEK FEL ANNAK BEMUTATÁSÁRA.



ÓRATERV – SZÓRÓANYAG



SUBJECT: PROGRAMMING AND CODING

TÉMA: BEVEZETŐ LECKÉK A PYTHON PROGRAMOZÁSI NYELVRŐL – 1. LECKE

1. BEVEZETÉS A PYTHONBA

FŐBB PONTOK:

A PYTHON EGY KÖNNYEN TANULHATÓ, MAGAS SZINTŰ PROGRAMOZÁSI NYELV, AMELYET SZÁMOS TERÜLETEN ALKALMAZNAK, MINT PÉLDÁUL WEBFEJLESZTÉS, ADATFELDOLGOZÁS, TUDOMÁNYOS SZÁMÍTÁSOK, ÉS MESTERSÉGES INTELLIGENCIA.

A PYTHON KIALAKULÁSA AZ 1980-AS ÉVEK VÉGÉRE NYÚLIK VISSZA, AMIKOR GUIDO VAN ROSSUM MEGALKOTTA A NYELVET A KÖNNYŰ TANULHATÓSÁG ÉS OLVASHATÓSÁG JEGYÉBEN.

A PYTHON ELTERJEDT ALKALMAZÁSAI KÖZÖTT MEGTALÁLHATÓK OLYAN NÉPSZERŰ PLATFORMOK, MINT A YOUTUBE, INSTAGRAM, ÉS NASA ADATELEMZÉSEI.

PÉLDA PROGRAM:

"HELLO, WORLD!" – AZ EGYIK LEGEREGYSZERŰBB PYTHON PROGRAM:

```
PRINT("HELLO, WORLD!")
```

2. ALAPVETŐ SZINTAXIS ÉS SZERKEZET

FŐBB PONTOK:

A PYTHONBAN A KÓD STRUKTÚRÁJA NAGYMÉRTÉKBEN FÜGG AZ INDENTÁLÁSTÓL, AZAZ A BEHÚZÁSOKTÓL. AZ INDENTÁLÁS HATÁROZZA MEG A KÓD BLOKKJAIT, ÍGY ELENEDHETETLEN A HELYES HASZNÁLATA.

A PRINT() FÜGGVÉNY AZ EGYIK LEGALAPVETŐBB ESZKÖZ A PYTHONBAN, AMELY LEHETŐVÉ TESZI, HOGY SZÖVEGEKET ÉS MÁS ADATOKAT JELENÍTSÜNK MEG A KÉPERNYŐN.

A PYTHONBAN NEM SZÜKSÉGES A SOROK VÉGÉN PONTOSVESSZŐ HASZNÁLATA, ELLENTÉT BEN SOK MÁS PROGRAMOZÁSI NYELVVEL.

TEVÉKENYSÉG:

ÍRJA MEG ÉS FUTTASSA A "HELLO, WORLD!" PROGRAMOT A SAJÁT FEJLESZTŐI KÖRNYEZETÉBEN.

HATÁROZZA MEG A PROGRAM MINDEN SORÁNAK CÉLJÁT, ÉS GYŐZŐDJÖN MEG ARRÓL, HOGY MEGÉRTI, HOGYAN MŰKÖDIK A PROGRAM.

3. HÁZI FELADAT

KÉSZÍTSEN EGY PYTHON PROGRAMOT, AMELY A "HELLO, WORLD!" ÜZENET HELYETT EGY SZEMÉLYES ÜZENETET JELENÍT MEG (PÉLDÁUL A SAJÁT NEVÉT VAGY EGY KEDVENC IDÉZETET). PRÓBÁLJON MEG KÍSÉRLETEZNI A PRINT() FÜGGVÉNNYEL, PÉLDÁUL HASZNÁLJON KÜLÖNBÖZŐ IDÉZŐJELEKET, VAGY ADJON HOZZÁ EGY ÚJ SORT A PROGRAMHOZ, AMELY TOVÁBBI SZÖVEGET JELENÍT MEG. MENTSE EL A MÓDOSÍTOTT PROGRAMOT, ÉS HOZZA EL A KÖVETKEZŐ ÓRÁRA, HOGY BEMUTATHASSA AZ OSZTÁLY ELŐTT.



ÓRATERV



SUBJECT: PROGRAMMING AND CODING

TÉMA: BEVEZETŐ LECKÉK A PYTHON PROGRAMOZÁSI NYELVRŐL – 1. LECKE

LECKE FÓKUSZA ÉS CÉLJAI:

1. EZ A LECKE BEVEZETI A TANULÓKAT A PYTHON PROGRAMOZÁSI NYELV ALAPJAIBA, BELEÉRTVE A TÖRTÉNETÉT, A NYELV SAJÁTOSságAIT ÉS AZ ELSŐ PYTHON PROGRAM MEGÍRÁSÁT.

2. A LECKE VÉGÉRE A TANULÓKNAK KÉPESNEK KELL LENNIÜK MEGÉRTENI A PYTHON NYELV ALAPVETŐ SZINTAXISÁT, VALAMINT ELKÉSZÍTENI ÉS FUTTATNI EGY EGYSZERŰ "HELLO, WORLD!" PROGRAMOT.

SZÜKSÉGES ANYAGOK:

- SZÁMÍTÓGÉPEK PYTHON TELEPÍTÉSSEL (PL. VS CODE, PYCHARM, VAGY MÁS FEJLESZTŐI KÖRNYEZET)
- PROJEKTOR ÉS KÉPERNYŐ
- TÁBLA ÉS MARKEREK
- AZ ALAPVETŐ PYTHON SZINTAXIS ÉS SZERKEZET KIOSZTÁSAI

1. BEVEZETÉS A PYTHONBA (15 PERC)

BESZÉLGETÉS (5 PERC): KEZDJE A PYTHON PROGRAMOZÁSI NYELV RÖVID TÖRTÉNETÉVEL, AMELYET GUIDO VAN ROSSUM FEJLESZTETT KI AZ 1980-AS ÉVEK VÉGÉN. BESZÉLJEN A NYELV KIALAKULÁSÁRÓL ÉS ARRÓL, HOGYAN VÁLT NÉPSZERŰVÉ A KÖNNYŰ TANULHATÓSÁGÁNAK ÉS SZÉLES KÖRŰ ALKALMAZHATÓSÁGÁNAK KÖSZÖNHETŐEN. EMELJE KI, HOGY A PYTHON HOGYAN FEKTETTE LE A MODERN PROGRAMOZÁSI NYELVEK ALAPJAIT, ÉS HOGY KÜLÖNÖSEN NÉPSZERŰ A TUDOMÁNYOS SZÁMÍTÁSOK, A WEBFEJLESZTÉS ÉS AZ ADATFELDOLGOZÁS TERÜLETÉN.

PREZENTÁCIÓ (10 PERC): VETÍTSE KI A PYTHON JELENLEGI HASZNÁLATÁT KÜLÖNBÖZŐ IPARÁGAKBAN. BESZÉLJE MEG, MIÉRT ÉRDEMES MEGTANULNI PYTHONT, ÉS MUTASSON PÉLDÁKAT NÉPSZERŰ PYTHON-ALAPÚ ALKALMAZÁSOKRA, MINT PÉLDÁUL A YOUTUBE, INSTAGRAM, ÉS A NASA ADATELEMZÉSEI.

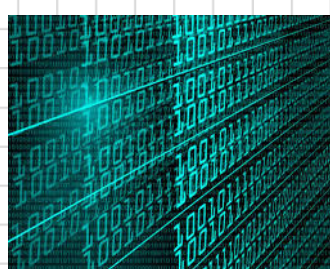
2. ALAPSZINTAXIS ÉS SZERKEZET (20 PERC)

MAGYARÁZAT (10 PERC): MUTASSA BE EGY EGYSZERŰ PYTHON PROGRAM SZERKEZETÉT. ISMERTESSE AZ IMPORT HASZNÁLATÁT, A MAIN() FÜGGVÉNY SZEREPÉT, ÉS A PYTHON SZINTAKTIKAI SAJÁTOSságAIT, PÉLDÁUL A BEHÚZÁSOK FONTOSságÁT. MAGYARÁZZA EL, HOGYAN KEZELI A PYTHON A KIFEJEZÉSEKET, ÉS MILYEN HIBÁKAT LEHET ELKÖVETNI. MUTASSON BE EGYSZERŰ HIBÁKAT, PÉLDÁUL AZ INDENTÁLÁS ELMULASZTÁSÁT, ÉS MAGYARÁZZA EL, HOGYAN LEHET EZEKET ELKERÜLNI.

PÉLDA (10 PERC): ÍRJON EGY "HELLO, WORLD!" PROGRAMOT A TÁBLÁN VAGY A PROJEKTOR KÉPERNYŐJÉN. RÉSZLETESEN MAGYARÁZZA EL A KÓD MINDEN SORÁNAK SZEREPÉT, BELEÉRTVE A PRINT() FÜGGVÉNY MŰKÖDÉSÉT, A SZTRINGEK KEZELÉSÉT ÉS A KIMENETI EREDMÉNY MEGÉRTÉSÉT. BÁTORÍTSA A TANULÓKAT, HOGY TEGYENEK FEL KÉRDÉSEKET A KÓD MINDEN RÉSZÉVEL KAPCSOLATBAN.



ÓRATERV



3. GYAKORLATI TEVÉKENYSÉG (20 PERC)

TEVÉKENYSÉG (15 PERC): KÉRJE MEG A TANULÓKAT, HOGY ÍRJÁK MEG SAJÁT "HELLO, WORLD!" PROGRAMJUKAT PYTHONBAN. SEGÍTSE ŐKET A FEJLESZTŐI KÖRNYEZET BEÁLLÍTÁSÁBAN (PÉLDÁUL AZ IDLE VAGY A PYCHARM HASZNÁLATÁVAL), ÉS ELLENŐRIZZE, HOGY MINDENKI SIKERESEN TUDJA FUTTATNI A PROGRAMJÁT. GYŐZŐDJÖN MEG ARRÓL, HOGY MINDEN TANULÓ MEGÉRTI A KÓD MŰKÖDÉSÉT ÉS A PYTHON KÖRNYEZETÉBEN VALÓ NAVIGÁCIÓT.

PEER REVIEW (5 PERC): PÁROSÍTSA A TANULÓKAT, HOGY ÁTNÉZZÉK EGYMÁS KÓDJÁT. BÁTORÍTSA ŐKET, HOGY BESZÉLJENEK MEG MINDEN FELMERÜLŐ KÉRDÉST VAGY PROBLÉMÁT, ÉS SEGÍTSENEK EGYMÁSNAK A KÓD JAVÍTÁSÁBAN.

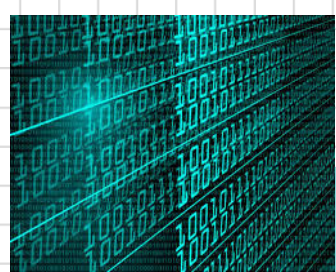
4. KÉRDÉSEK ÉS VÁLASZOK ÉS ÖSSZEFOGLALÓ (5 PERC)

KÉRDÉSEK ÉS VÁLASZOK (3 PERC): NYISSA MEG A TERET A TANULÓK KÉRDÉSEI SZÁMÁRA. BESZÉLJEN MEG GYAKORI PROBLÉMÁKAT VAGY ZAVAROS PONTOKAT, AMELYEK FELMERÜLHETTEK A LECKE SORÁN. ADJ LEHETŐSÉGET ARRA, HOGY A TANULÓK VISSZAJELZÉST ADJANAK A LECKE MEGÉRTÉSÉRŐL.

ÖSSZEFOGLALÓ ÉS HÁZI FELADAT (2 PERC): FOGLALJA ÖSSZE A LECKE LEGFONTOSABB PONTJAIT, KIEMELVE A PYTHON ALAPVETŐ SZINTAXISÁT ÉS A "HELLO, WORLD!" PROGRAM FONTOSSÁGÁT. HÁZI FELADATKÉNT KÉRJE MEG A TANULÓKAT, HOGY KERESSENEK EGY EGYSZERŰ PYTHON PROGRAMOT ONLINE, MÓDOSÍTSÁK KIMENETI ÜZENETÉT, ÉS FUTTASSÁK AZT. KÉRJE ŐKET, HOGY HOZZÁK EL A KÖVETKEZŐ ÓRÁRA A MÓDOSÍTOTT KÓDOT, ÉS KÉSZÜLJENEK FEL ANNAK BEMUTATÁSÁRA.



ÓRATERV – SZÓRÓANYAG



SUBJECT: PROGRAMMING AND CODING

TÉMA: BEVEZETŐ LECKÉK A PYTHON PROGRAMOZÁSI NYELVRŐL – 1. LECKE

1. BEVEZETÉS A PYTHONBA

FŐBB PONTOK:

A PYTHON EGY KÖNNYEN TANULHATÓ, MAGAS SZINTŰ PROGRAMOZÁSI NYELV, AMELYET SZÁMOS TERÜLETEN ALKALMAZNAK, MINT PÉLDÁUL WEBFEJLESZTÉS, ADATFELDOLGOZÁS, TUDOMÁNYOS SZÁMÍTÁSOK, ÉS MESTERSÉGES INTELLIGENCIA.

A PYTHON KIALAKULÁSA AZ 1980-AS ÉVEK VÉGÉRE NYÚLIK VISSZA, AMIKOR GUIDO VAN ROSSUM MEGALKOTTA A NYELVET A KÖNNYŰ TANULHATÓSÁG ÉS OLVASHATÓSÁG JEGYÉBEN. A PYTHON ELTERJEDT ALKALMAZÁSAI KÖZÖTT MEGTALÁLHATÓK OLYAN NÉPSZERŰ PLATFORMOK, MINT A YOUTUBE, INSTAGRAM, ÉS NASA ADATELEMZÉSEI.

PÉLDA PROGRAM:

"HELLO, WORLD!" – AZ EGYIK LEGEGYSZERŰBB PYTHON PROGRAM:

```
PRINT("HELLO, WORLD!")
```

2. ALAPVETŐ SZINTAXIS ÉS SZERKEZET

FŐBB PONTOK:

A PYTHONBAN A KÓD STRUKTÚRÁJA NAGYMÉRTÉKBEN FÜGG AZ INDENTÁLÁSTÓL, AZAZ A BEHÚZÁSOKTÓL. AZ INDENTÁLÁS HATÁROZZA MEG A KÓD BLOKKJAIT, ÍGY ELENEDHETETLEN A HELYES HASZNÁLATA.

A PRINT() FÜGGVÉNY AZ EGYIK LEGALAPVETŐBB ESZKÖZ A PYTHONBAN, AMELY LEHETŐVÉ TESZI, HOGY SZÖVEGEKET ÉS MÁS ADATOKAT JELENÍTSÜNK MEG A KÉPERNYŐN.

A PYTHONBAN NEM SZÜKSÉGES A SOROK VÉGÉN PONTOSVESSZŐ HASZNÁLATA, ELLENTÉTBEN SOK MÁS PROGRAMOZÁSI NYELVVEL.

TEVÉKENYSÉG:

ÍRJA MEG ÉS FUTTASSA A "HELLO, WORLD!" PROGRAMOT A SAJÁT FEJLESZTŐI KÖRNYEZETÉBEN.

HATÁROZZA MEG A PROGRAM MINDEN SORÁNAK CÉLJÁT, ÉS GYŐZŐDJÖN MEG ARRÓL, HOGY MEGÉRTI, HOGYAN MŰKÖDIK A PROGRAM.

3. HÁZI FELADAT

KÉSZÍTSEN EGY PYTHON PROGRAMOT, AMELY A "HELLO, WORLD!" ÜZENET HEYETT EGY SZEMÉLYES ÜZENETET JELENÍT MEG (PÉLDÁUL A SAJÁT NEVÉT VAGY EGY KEDVENC IDÉZETET). PRÓBÁLJON MEG KÍSÉRLETEZNI A PRINT() FÜGGVÉNNYEL, PÉLDÁUL HASZNÁLJON KÜLÖNBÖZŐ IDÉZŐJELEKET, VAGY ADJON HOZZÁ EGY ÚJ SORT A PROGRAMHOZ, AMELY TOVÁBBI SZÖVEGET JELENÍT MEG. MENTSE EL A MÓDOSÍTOTT PROGRAMOT, ÉS HOZZA EL A KÖVETKEZŐ ÓRÁRA, HOGY BEMUTATHASSA AZ OSZTÁLY ELŐTT.



ÓRATERV



SUBJECT: PROGRAMMING AND CODING

TÉMA: VÁLTOZÓK A PYTHON PROGRAMOZÁSI NYELVBEN – 3. LECKE

LECKE FÓKUSZA ÉS CÉLJAI:

1. BEVEZETÉS A VÁLTOZÓK FOGALMÁBA, AZOK SZEREPÉBE ÉS HASZNÁLATÁBA A PYTHON PROGRAMOZÁS SORÁN.

2. A LECKE VÉGÉRE A TANULÓKNAK KÉPESNEK KELL LENNIÜK VÁLTOZÓKAT LÉTREHOZNI, ÉRTÉKEKET HOZZÁRENDELNI, ÉS MEGÉRTENI, HOGYAN HASZNÁLHATÓK A VÁLTOZÓK KÜLÖNBÖZŐ PROGRAMRÉSZEKBE.

SZÜKSÉGES ANYAGOK:

- SZÁMÍTÓGÉPEK PYTHON TELEPÍTÉSSEL (PL. VS CODE, PYCHARM, VAGY MÁS FEJLESZTŐI KÖRNYEZET)
- PROJEKTOR VAGY KÉPERNYŐ
- TÁBLA ÉS MARKEREK
- AZ ADATTÍPUSOKAT BEMUTATÓ MELLÉKLETEK

1. BEVEZETÉS A VÁLTOZÓKBA (15 PERC)

BESZÉLGETÉS (5 PERC): KEZDJE AZZAL, HOGY ELMAGYARÁZZA, MI AZ A VÁLTOZÓ, ÉS MIÉRT FONTOSAK A PROGRAMOZÁSBAN. BESZÉLJEN ARRÓL, HOGYAN HASZNÁLHATÓK A VÁLTOZÓK AZ ADATOK TÁROLÁSÁRA ÉS MANIPULÁLÁSÁRA A PROGRAMOKBAN.

PREZENTÁCIÓ (10 PERC): VETÍTSE KI A KÜLÖNBÖZŐ VÁLTOZÓTÍPUSOKAT, ÉS MAGYARÁZZA EL, HOGYAN MŰKÖDNEK. PÉLDÁUL MUTASSA BE, HOGYAN LEHET EGY VÁLTOZÓBA SZÁMOKAT, SZÖVEGEKET, ILLETVE LOGIKAI ÉRTÉKEKET TÁROLNI. HANGSÚLYOZZA A VÁLTOZÓK NÉVADÁSI SZABÁLYAIT ÉS LEGJOBB GYAKORLATAIT.

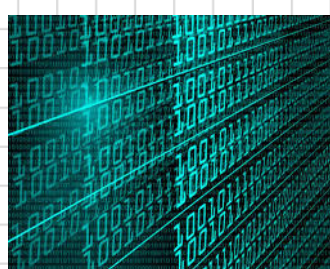
2. VÁLTOZÓK HASZNÁLATA PYTHONBAN (20 PERC)

MAGYARÁZAT (10 PERC): MUTASSA BE, HOGYAN LEHET VÁLTOZÓKAT LÉTREHOZNI ÉS HASZNÁLNI PYTHONBAN. MAGYARÁZZA EL A VÁLTOZÓK ÉRTÉKHOZZÁRENDELÉSÉT, A VÁLTOZÓK ÚJRAÉRTÉKEZÉSÉT (ÚJ ÉRTÉK HOZZÁRENDELÉSÉT), VALAMINT A KÜLÖNBÖZŐ VÁLTOZÓ TÍPUSOK KÖZÖTTI ÁTVÁLTÁST (TÍPUSKONVERZIÓ).

PÉLDA (10 PERC): ÍRJON NÉHÁNY EGYSZERŰ PÉLDAKÓDOT A TÁBLÁN VAGY A PROJEKTOR KÉPERNYŐJÉN, AMELY BEMUTATJA A VÁLTOZÓK LÉTREHOZÁSÁT, ÉRTÉKHOZZÁRENDELÉSÉT ÉS HASZNÁLATÁT EGY PROGRAMBAN. MAGYARÁZZA EL, HOGYAN HASZNÁLHATÓ EGY VÁLTOZÓ TÖBB MŰVELETBEN, ÉS HOGY A VÁLTOZÓK HOGYAN TESZIK HATÉKONYABBÁ A KÓDOT.



ÓRATERV



3. GYAKORLATI TEVÉKENYSÉG (20 PERC)

TEVÉKENYSÉG (15 PERC): KÉRJE MEG A TANULÓKAT, HOGY ÍRJANAK EGY EGYSZERŰ PYTHON PROGRAMOT, AMELY TÖBB VÁLTOZÓT IS HASZNÁL. PÉLDÁUL HOZZANAK LÉTRE NÉHÁNY VÁLTOZÓT KÜLÖNBÖZŐ TÍPUSÚ ÉRTÉKEKKEL, MAJD VÉGEZZENEK MŰVELETEKET EZEKKEL A VÁLTOZÓKKAL. SEGÍTSE ŐKET A FEJLESZTŐI KÖRNYEZET BEÁLLÍTÁSÁBAN, ÉS GYŐZŐDJÖN MEG ARRÓL, HOGY MINDENKI SIKERESEN TUDJA FUTTATNI A PROGRAMJÁT.

PÁROS MUNKA (5 PERC): PÁROSÍTSA A TANULÓKAT, HOGY ÁTNÉZZÉK EGYMÁS KÓDJÁT. BÁTORÍTSA ŐKET, HOGY BESZÉLJENEK MEG MINDEN FELMERÜLŐ KÉRDÉST VAGY PROBLÉMÁT, ÉS SEGÍTSENEK EGYMÁSNAK A KÓD JAVÍTÁSÁBAN.

4. KÉRDÉSEK ÉS VÁLASZOK ÉS ÖSSZEFOGLALÓ (5 PERC)

KÉRDÉSEK ÉS VÁLASZOK (3 PERC): NYISSA MEG A TERET A TANULÓK KÉRDÉSEI SZÁMÁRA. BESZÉLJEN MEG GYAKORI PROBLÉMÁKAT VAGY ZAVAROS PONTOKAT, AMELYEK FELMERÜLHETTEK A LECKE SORÁN. ADJ LEHETŐSÉGET ARRA, HOGY A TANULÓK VISSZAJELZÉST ADJANAK A LECKE MEGÉRTÉSÉRŐL.

ÖSSZEFOGLALÓ ÉS HÁZI FELADAT (2 PERC): FOGLALJA ÖSSZE A LECKE LEGFONTOSABB PONTJAIT, KIEMELVE A VÁLTOZÓK HASZNÁLATÁNAK ALAPJAIT ÉS AZOK FONTOSSÁGÁT. HÁZI FELADATKÉNT KÉRJE MEG A TANULÓKAT, HOGY ÍRJANAK EGY PYTHON PROGRAMOT, AMELY KÜLÖNBÖZŐ TÍPUSÚ VÁLTOZÓKAT HASZNÁL, ÉS VÉGEZZENEK EL VELÜK KÜLÖNBÖZŐ MŰVELETEKET.



ÓRATERV – SZÓRÓANYAG



SUBJECT: PROGRAMMING AND CODING

TÉMA: VÁLTOZÓK A PYTHON PROGRAMOZÁSI NYELVBEN – 3. LECKE

1. VÁLTOZÓK

FŐBB PONTOK:

A VÁLTOZÓK OLYAN TÁROLÓK, AMELYEK ADATOKAT TARTANAK A PROGRAM FUTÁSA SORÁN. A VÁLTOZÓK LÉTREHOZÁSOKOR NEVET ADUNK NEKIK, ÉS HOZZÁRENDELJÜK A KÍVÁNT ÉRTÉKET.

A VÁLTOZÓK NEVEI KÖVETIK A PYTHON SZINTAKTIKAI SZABÁLYAIT, ÉS EZEK A NEVEK SEGÍTENEK AZ ADATOK AZONOSÍTÁSÁBAN ÉS FELHASZNÁLÁSÁBAN A PROGRAMBAN.

PÉLDA PROGRAM:

```
variables_example.py > ...
1  # Example of using variables
2  x = 10                        # integer variable
3  y = 5.5                      # floating-point variable
4  name = "Alice"              # string variable
5  is_happy = True              # boolean variable
6
7  # Performing operations with variables
8  sum_result = x + y
9  greeting = "Hello, " + name
10 status = is_happy and (x > y)
11
12 print(sum_result)            # Output the result of the addition
13 print(greeting)              # Output the greeting message
14 print(status)                # Output the logical operation result
```

2. GYAKORLATI FELADAT

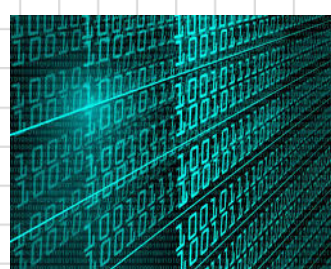
FELADAT: ÍRJON EGY PYTHON PROGRAMOT, AMELY KÜLÖNBÖZŐ TÍPUSÚ ÉRTÉKEKET HASZNÁL. PÉLDÁUL ADJON ÖSSZE KÉT SZÁMOT, JELENÍTSEN MEG EGY ÜZENETET A KÉPERNYŐN, ÉS HASZNÁLJON LOGIKAI OPERÁTOROT EGY FELTÉTEL ELLENŐRZÉSÉRE.

3. HÁZI FELADAT

KÉSZÍTSEN EGY PROGRAMOT, AMELY KISZÁMÍTTJA A KÖR TERÜLETÉT ÉS KERÜLETÉT EGY ADOTT SUGÁR ALAPJÁN, ÉS HASZNÁLJON VÁLTOZÓKAT AZ ÉRTÉKEK TÁROLÁSÁRA. MENTSE EL A PROGRAMOT, ÉS HOZZA EL A KÖVETKEZŐ ÓRÁRA, HOGY BEMUTATHASSA AZ OSZTÁLY ELŐTT.



ÓRATERV



SUBJECT: PROGRAMMING AND CODING

TÉMA: MŰVELETI JELEK ÉS OPERANDUSOK A PYTHON PROGRAMOZÁSI NYELVBEN – 4. LECKE

LECKE FÓKUSZA ÉS CÉLJAI:

1. BEVEZETÉS A KÜLÖNBÖZŐ MŰVELETI JELEKBE (ARITMETIKAI, ÖSSZEHASONLÍTÓ, LOGIKAI, STB.) ÉS AZOK HASZNÁLATÁBA PYTHONBAN.

2. A LECKE VÉGÉRE A TANULÓKNAK KÉPESNEK KELL LENNIÜK EGYSZERŰ KIFEJEZÉSEKET LÉTREHOZNI ÉS ÉRTÉKELNI, VALAMINT MEGÉRTENI A KÜLÖNBÖZŐ MŰVELETEK ÉS AZ OPERANDUSOK KÖZÖTTI KAPCSOLATOT

SZÜKSÉGES ANYAGOK:

- SZÁMÍTÓGÉPEK PYTHON TELEPÍTÉSSEL (PL. VS CODE, PYCHARM, VAGY MÁS FEJLESZTŐI KÖRNYEZET)
- PROJEKTOR VAGY KÉPERNYŐ
- TÁBLA ÉS MARKEREK
- MŰVELETI JELEKET ÉS OPERANDUSOKAT BEMUTATÓ MELLÉKLET

1. BEVEZETÉS A MŰVELETI JELEKBE ÉS OPERANDUSOKBA (15 PERC)

BESZÉLGETÉS (5 PERC): KEZDJE AZZAL, HOGY ELMAGYARÁZZA, MI AZ A MŰVELETI JEL ÉS OPERANDUS, ÉS HOGYAN HASZNÁLATOSAK EZEK A PROGRAMOZÁSBAN. PÉLDÁKAT MUTATHAT A MINDENNAPI ÉLETBŐL, MINT PÉLDÁUL A MATEMATIKAI MŰVELETEK ELVÉGZÉSE.

PREZENTÁCIÓ (10 PERC): VETÍTSE KI A KÜLÖNBÖZŐ MŰVELETI JELEKET (PL. ÖSSZEADÁS, KIVONÁS, SZORZÁS, OSZTÁS, STB.) ÉS AZOK SZINTAXISÁT PYTHONBAN. BESZÉLJE MEG, HOGYAN MŰKÖDNEK EZEK A JELEK AZ OPERANDUSOKKAL EGYÜTT, ÉS MUTASSA BE AZ ARITMETIKAI ÉS LOGIKAI OPERÁTOROK KÖZÖTTI KÜLÖNBBSÉGEKET.

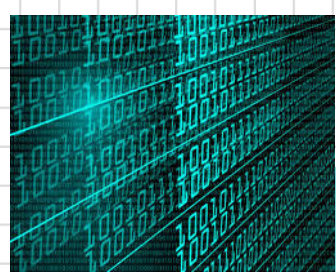
2. MŰVELETI JELEK ÉS OPERANDUSOK HASZNÁLATA PYTHONBAN (20 PERC)

MAGYARÁZAT (10 PERC): MUTASSA BE, HOGYAN LEHET EGYSZERŰ KIFEJEZÉSEKET LÉTREHOZNI ÉS HASZNÁLNI PYTHONBAN A MŰVELETI JELEKKEL ÉS OPERANDUSOKKAL. MAGYARÁZZA EL A KIFEJEZÉSEK KIÉRTÉKELÉSÉNEK FOLYAMATÁT, ÉS HOGYAN ADNAK VISSZA EREDMÉNYEKET EZEK A MŰVELETEK.

PÉLDA (10 PERC): ÍRJON NÉHÁNY EGYSZERŰ PÉLDAKÓDOT A TÁBLÁN VAGY A PROJEKTOR KÉPERNYŐJÉN, AMELY BEMUTATJA AZ ALAPVETŐ ARITMETIKAI ÉS LOGIKAI OPERÁTOROK HASZNÁLATÁT PYTHONBAN. MAGYARÁZZA EL, HOGYAN MŰKÖDNEK EZEK A MŰVELETEK KÜLÖNBÖZŐ TÍPUSÚ OPERANDUSOKKAL, PÉLDÁUL SZÁMOKKAL ÉS LOGIKAI ÉRTÉKEKKEL.



ÓRATERV



3. GYAKORLATI TEVÉKENYSÉG (20 PERC)

TEVÉKENYSÉG (15 PERC): KÉRJE MEG A TANULÓKAT, HOGY ÍRJANAK EGY EGYSZERŰ PYTHON PROGRAMOT, AMELY KÜLÖNBÖZŐ MŰVELETI JELEKET ÉS OPERANDUSOKAT HASZNÁL. PÉLDÁUL SZÁMOLJANAK KI EGYSZERŰ ARITMETIKAI MŰVELETEKET, ÉS KÉSZÍTSENEK OLYAN KIFEJEZÉSEKET, AMELYEK LOGIKAI OPERÁTOROKAT HASZNÁLNAK. SEGÍTSE ŐKET A FEJLESZTŐI KÖRNYEZET BEÁLLÍTÁSÁBAN, ÉS GYŐZŐDJÖN MEG ARRÓL, HOGY MINDENKI SIKERESEN TUDJA FUTTATNI A PROGRAMJÁT.

PÁROS MUNKA (5 PERC): PÁROSÍTSA A TANULÓKAT, HOGY ÁTNÉZZÉK EGYMÁS KÓDJÁT. BÁTORÍTSA ŐKET, HOGY BESZÉLJENEK MEG MINDEN FELMERÜLŐ KÉRDÉST VAGY PROBLÉMÁT, ÉS SEGÍTSENEK EGYMÁSNAK A KÓD JAVÍTÁSÁBAN.

4. KÉRDÉSEK ÉS VÁLASZOK ÉS ÖSSZEFOGLALÓ (5 PERC)

KÉRDÉSEK ÉS VÁLASZOK (3 PERC): NYISSA MEG A TERET A TANULÓK KÉRDÉSEI SZÁMÁRA. BESZÉLJEN MEG GYAKORI PROBLÉMÁKAT VAGY ZAVAROS PONTOKAT, AMELYEK FELMERÜLHETTEK A LECKE SORÁN. ADJ LEHETŐSÉGET ARRA, HOGY A TANULÓK VISSZAJELZÉST ADJANAK A LECKE MEGÉRTÉSÉRŐL.

ÖSSZEFOGLALÓ ÉS HÁZI FELADAT (2 PERC): FOGLALJA ÖSSZE A LECKE LEGFONTOSABB PONTJAIT, KIEMELVE A MŰVELETI JELEK ÉS OPERANDUSOK HASZNÁLATÁT PYTHONBAN. HÁZI FELADATKÉNT KÉRJE MEG A TANULÓKAT, HOGY ÍRJANAK EGY PYTHON PROGRAMOT, AMELY KÜLÖNBÖZŐ ARITMETIKAI ÉS LOGIKAI KIFEJEZÉSEKET TARTALMAZ.



ÓRATERV – SZÓRÓANYAG



SUBJECT: PROGRAMMING AND CODING

TÉMA: MŰVELETI JELEK ÉS OPERANDUSOK A PYTHON PROGRAMOZÁSI NYELVBEN – 4. LECKE

1. MŰVELETI JELEK ÉS OPERANDUSOK

FŐBB PONTOK:

A MŰVELETI JELEK SEGÍTSÉGÉVEL KÜLÖNBÖZŐ MŰVELETEKET VÉGEZHETÜNK EL AZ OPERANDUSOKKAL, AMELYEK LEHETNEK SZÁMOK, SZÖVEGEK VAGY LOGIKAI ÉRTÉKEK. AZ ARITMETIKAI MŰVELETEK (PL. ÖSSZEADÁS, KIVONÁS) ÉS A LOGIKAI OPERÁTOROK (PL. AND, OR, NOT) ALAPVETŐ ESZKÖZÖK A PROGRAMOZÁSBAN.

FONTOS MEGÉRTENI, HOGYAN VÁLTOZTATJÁK MEG EZEK A MŰVELETEK AZ OPERANDUSOK ÉRTÉKÉT ÉS TÍPUSÁT.

PÉLDA PROGRAM:

 variables_example.py > ...

```
1  # Example of using variables
2  x = 10                        # integer variable
3  y = 5.5                      # floating-point variable
4  name = "Alice"              # string variable
5  is_happy = True              # boolean variable
6
7  # Performing operations with variables
8  sum_result = x + y
9  greeting = "Hello, " + name
10 status = is_happy and (x > y)
11
12 print(sum_result)            # Output the result of the addition
13 print(greeting)              # Output the greeting message
14 print(status)                # Output the logical operation result
```

2. GYAKORLATI FELADAT

FELADAT: ÍRJON EGY PYTHON PROGRAMOT, AMELY KÜLÖNBÖZŐ MŰVELETI JELEKET ÉS OPERANDUSOKAT HASZNÁL. PÉLDÁUL VÉGEZZEN ÖSSZEADÁST, OSZTÁST ÉS LOGIKAI MŰVELETEKET, MAJD ÉRTÉKELJE KI A KIFEJEZÉSEKET.

3. HÁZI FELADAT

KÉSZÍTSEN EGY PROGRAMOT, AMELY KISZÁMÍTJA KÉT SZÁM ÁTLAGÁT ÉS MEGJELENÍTI AZ EREDMÉNYT, MAJD VÉGEZZEN LOGIKAI ELLENŐRZÉST, HOGY AZ ÁTLAG NAGYOBB-E EGY MEGADOTT ÉRTÉKNÉL. MENTSE EL A PROGRAMOT, ÉS HOZZA EL A KÖVETKEZŐ ÓRÁRA, HOGY BEMUTATHASSA AZ OSZTÁLY ELŐTT.



ÓRATERV



SUBJECT: PROGRAMMING AND CODING

TÉMA: FELTÉTELES UTASÍTÁSOK (IF-ELSE) A PYTHON PROGRAMOZÁSI NYELVBEN – 5. LECKE

LECKE FÓKUSZA ÉS CÉLJAI:

1. EZ A LECKE BEVEZETI A TANULÓKAT A FELTÉTELES UTASÍTÁSOK HASZNÁLATÁBA A PYTHON PROGRAMOZÁSI NYELVBEN, BELEÉRTVE AZ IF, ELIF, ÉS ELSE KULCSSZAVAKAT.

2. A LECKE VÉGÉRE A TANULÓKNAK KÉPESNEK KELL LENNIÜK EGYSZERŰ FELTÉTELES LOGIKÁT IMPLEMENTÁLNI PROGRAMJAIKBAN, ÉS MEGÉRTENI A DÖNTÉSHOZATAL FOLYAMATÁT A KÓDOKBAN.

SZÜKSÉGES ANYAGOK:

- SZÁMÍTÓGÉPEK PYTHON TELEPÍTÉSSEL (PL. VS CODE, PYCHARM, VAGY MÁS FEJLESZTŐI KÖRNYEZET)
- PROJEKTOR VAGY KÉPERNYŐ
- TÁBLA ÉS MARKEREK
- FELTÉTELES UTASÍTÁSOKAT BEMUTATÓ MELLÉKLET

1. BEVEZETÉS A FELTÉTELES UTASÍTÁSOKBA (15 PERC)

BESZÉLGETÉS (5 PERC): KEZDJE AZZAL, HOGY ELMAGYARÁZZA, MIÉRT VAN SZÜKSÉG FELTÉTELES UTASÍTÁSOKRA A PROGRAMOZÁSBAN. BESZÉLJEN A DÖNTÉSHOZATAL FONTOSSÁGÁRÓL, ÉS ARRÓL, HOGYAN HASZNÁLNAK A PROGRAMOK LOGIKAI KIFEJEZÉSEKET ANNAK ELDÖNTÉSÉRE, HOGY MILYEN MŰVELETEKET HAJTSANAK VÉGRE.

PREZENTÁCIÓ (10 PERC): VETÍTSE KI A FELTÉTELES UTASÍTÁSOK SZERKEZETÉT PYTHONBAN. MAGYARÁZZA EL AZ IF, ELIF, ÉS ELSE KULCSSZAVAK SZEREPÉT, ÉS MUTASSON PÉLDÁKAT MINDENNAPI ÉLETBŐL, AHOL DÖNTÉSEKET KELL HOZNI, MAJD EZEKRE PÉLDA KÓDOT MUTASSON BE PYTHONBAN.

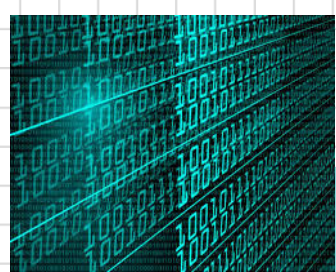
2. FELTÉTELES UTASÍTÁSOK HASZNÁLATA PYTHONBAN (20 PERC)

MAGYARÁZAT (10 PERC): MUTASSA BE, HOGYAN HASZNÁLHATÓK A FELTÉTELES UTASÍTÁSOK PYTHONBAN. MAGYARÁZZA EL, HOGYAN MŰKÖDIK AZ IF-ELSE SZERKEZET, ÉS HOGYAN LEHET LOGIKAI KIFEJEZÉSEKET HASZNÁLNI A FELTÉTELEK ELLENŐRZÉSÉRE. MUTASSON PÉLDÁKAT KÜLÖNBÖZŐ SZITUÁCIÓKRA, MINT PÉLDÁUL AZ IDŐ ALAPJÁN TÖRTÉNŐ NAPSZAK MEGHATÁROZÁSA, VAGY EGY PONTSZÁM ALAPJÁN ELDÖNTENI, HOGY VALAKI ÁTMENT-E EGY VIZSGÁN.

PÉLDA (10 PERC): ÍRJON EGYSZERŰ PÉLDAKÓDOT A TÁBLÁN VAGY A PROJEKTOR KÉPERNYŐJÉN, AMELY BEMUTATJA AZ IF-ELSE SZERKEZET HASZNÁLATÁT. PÉLDÁUL EGY PROGRAM, AMELY ELDÖNTI, HOGY VALAKI JOGOSULT-E EGY KEDVEZTMÉNYRE, VAGY HOGY EGY SZÁM POZITÍV, NEGATÍV, VAGY NULLA.



ÓRATERV



3. GYAKORLATI TEVÉKENYSÉG (20 PERC)

TEVÉKENYSÉG (15 PERC): KÉRJE MEG A TANULÓKAT, HOGY ÍRJANAK EGY EGYSZERŰ PYTHON PROGRAMOT, AMELY TARTALMAZ EGY IF-ELSE STRUKTÚRÁT. PÉLDÁUL ÍRJANAK EGY PROGRAMOT, AMELY ELLENŐRZI A FELHASZNÁLÓ ÉLETKORÁT, ÉS ELDÖNTI, HOGY A FELHASZNÁLÓ BELÉPHET-E EGY ADOTT ESEMÉNYRE. SEGÍTSE ŐKET A FEJLESZTŐI KÖRNYEZET BEÁLLÍTÁSÁBAN, ÉS GYŐZŐDJÖN MEG ARRÓL, HOGY MINDENKI SIKERESEN TUDJA FUTTATNI A PROGRAMJÁT.

PÁROS MUNKA (5 PERC): PÁROSÍTSA A TANULÓKAT, HOGY ÁTNÉZZÉK EGYMÁS KÓDJÁT. BÁTORÍTSA ŐKET, HOGY BESZÉLJENEK MEG MINDEN FELMERÜLŐ KÉRDÉST VAGY PROBLÉMÁT, ÉS SEGÍTSENEK EGYMÁSNAK A KÓD JAVÍTÁSÁBAN.

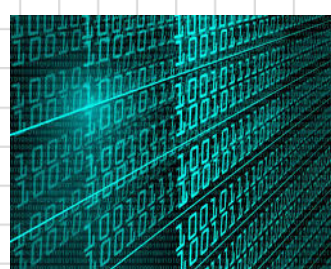
4. KÉRDÉSEK ÉS VÁLASZOK ÉS ÖSSZEFOGLALÓ (5 PERC)

KÉRDÉSEK ÉS VÁLASZOK (3 PERC): NYISSA MEG A TERET A TANULÓK KÉRDÉSEI SZÁMÁRA. BESZÉLJEN MEG GYAKORI PROBLÉMÁKAT VAGY ZAVAROS PONTOKAT, AMELYEK FELMERÜLHETTEK A LECKE SORÁN. ADJ LEHETŐSÉGET ARRA, HOGY A TANULÓK VISSZAJELZÉST ADJANAK A LECKE MEGÉRTÉSÉRŐL.

ÖSSZEFOGLALÓ ÉS HÁZI FELADAT (2 PERC): FOGLALJA ÖSSZE A LECKE LEGFONTOSABB PONTJAIT, KIEMELVE A FELTÉTELES UTASÍTÁSOK FONTOSSÁGÁT. HÁZI FELADATKÉNT KÉRJE MEG A TANULÓKAT, HOGY ÍRJANAK EGY PYTHON PROGRAMOT, AMELY EGY EGYSZERŰ DÖNTÉSI FOLYAMATOT MODELLEZ EGY IF-ELSE SZERKEZETTEL.



ÓRATERV – SZÓRÓANYAG



SUBJECT: PROGRAMMING AND CODING

TÉMA: FELTÉTELES UTASÍTÁSOK (IF-ELSE) A PYTHON PROGRAMOZÁSI NYELVBEN – 5. LECKE

1. FELTÉTELES UTASÍTÁSOK

FŐBB PONTOK:

A FELTÉTELES UTASÍTÁSOK SEGÍTSÉGÉVEL A PROGRAM DÖNTÉSEKET HOZHAT, ATTÓL FÜGGŐEN, HOGY BIZONYOS FELTÉTELEK TELJESÜLNEK-E.

AZ IF, ELIF, ÉS ELSE KULCSSZAVAK KÜLÖNBÖZŐ HELYZETEK KEZELÉSÉRE SZOLGÁLNAK A KÓDBAN.

A LOGIKAI KIFEJEZÉSEK MEGHATÁROZZÁK, HOGY MELYIK KÓDRÉSZLET HAJTÓDIK VÉGRE.

PÉLDA PROGRAM:

```
if_else_examples.py > ...  
1  # Example of using if-else statements to determine the time of day  
2  hour = int(input("Enter the current hour (0-23): "))  
3  
4  if 5 <= hour < 12:  
5      print("Good morning!")  
6  elif 12 <= hour < 18:  
7      print("Good afternoon!")  
8  elif 18 <= hour < 22:  
9      print("Good evening!")  
10 else:  
11     print("Good night!")
```

2. GYAKORLATI FELADAT

FELADAT: ÍRJON EGY PYTHON PROGRAMOT, AMELY EGY IF-ELSE STRUKTÚRÁT HASZNÁL KÜLÖNBÖZŐ FELTÉTELEK ELLENŐRZÉSÉRE. PÉLDÁUL DÖNTSE EL, HOGY EGY SZÁM PÁROS VAGY PÁRATLAN, VAGY ELLENŐRIZZE, HOGY EGY SZEMÉLY JOGOSULT-E EGY KEDVEZTMÉNYRE ÉLETKOR ALAPJÁN.

3. HÁZI FELADAT

KÉSZÍTSEN EGY PROGRAMOT, AMELY EGY HŐMÉRSÉKLETET ELLENŐRIZ, ÉS ELDÖNTI, HOGY AZ IDŐJÁRÁS HIDEG, MELEG VAGY FORRÓ-E. HASZNÁLJON IF-ELSE STRUKTÚRÁT A DÖNTÉSHEZ, ÉS MENTSE EL A PROGRAMOT, HOGY A KÖVETKEZŐ ÓRÁN BEMUTATHASSA.



01. 01. 2024 –31. 12. 2025



**Funded by
the European Union**



**Spojená škola
Kollárova 17, Sečovce**

**Móricz Zsigmond
Református Kollégium,
Gimnázium, Technikum,
Általános Iskola és Óvoda**



**SOLSKI CENTER
SLOVENJ GRADEC**

2023-2-SK01-KA210-VET-000184496