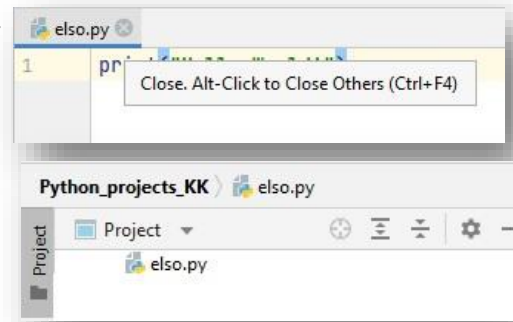


Ha az éppen megnyitott programot/kódot **be akarjuk zárni**, akkor fent a megnyitott "fülek" közül bezárjuk az „X” gombbal.



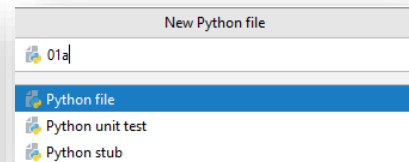
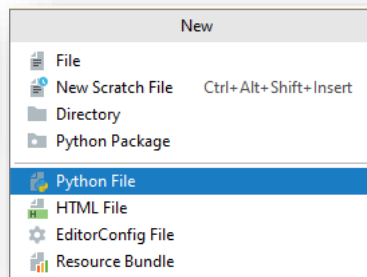
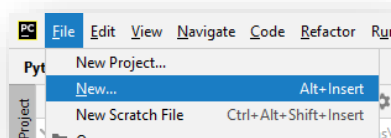
Ha **meg akarunk nyitni**, egy programot/kódot, akkor a bal oldali listából kiválasztjuk.

01. ALAPMŰVELETEK

Nézzük meg, hogy hogyan használjuk az **egyszerű matematikai műveleteket** a python programozáskor. A parancsértelmező nagyon könnyen használható, csak be kell írni egyszerűen a „print()” utasításba a megszokott műveleteket.

(01a.py)

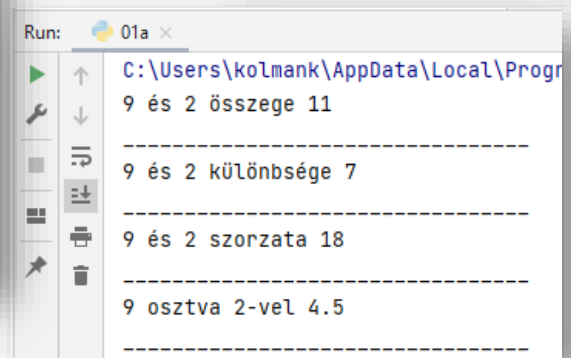
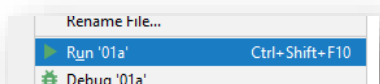
Hozzunk létre egy új python fájlt, a neve legyen 01a!



Ebben a feladatban az **összeadás, kivonás, szorzás és osztás műveleteit** próbáljuk ki!

- Gépeled be a képen látható utasításokat, majd futtasd a programot!
- A kiíratáshoz a "print()" parancsot úgy használjuk, hogy **ha szöveget akarunk megjeleníteni, akkor a zárójelben idézőjelek („szöveg”), vagy aposztrófok ('szöveg') közé írjuk a karaktereket!**
- **Ha szöveget szeretnénk kiírni és mellette még számolandó műveletet is elhelyezni akkor vesszővel kell elválasztani egymástól őket!**
- **A képletekben a matematikában használt módon alkalmazzuk a műveleti jeleket! (+, -, *, /)**
- A sorok között vonalakat húzunk, melyet elég egyszer begépelni, utána csak másolunk! Használhatjuk a kijelölés után a Ctrl+C; Ctrl+V billentyűkombinációkat!
- A jobb oldali képen láthatod a futtatott program eredményét.

```
01a.py x
1 print("9 és 2 összege",9+2)
2 print("-----")
3 print("9 és 2 különbsége",9-2)
4 print("-----")
5 print("9 és 2 szorzata",9*2)
6 print("-----")
7 print("9 osztva 2-vel",9 / 2)
8 print("-----")
```



(01b.py)

Hozzunk létre egy új python fájlt, a neve legyen 01b!

Ebben a feladatban az osztásnál használatos **egész rész meghatározása (div)** és a **maradék meghatározására (mod)** fogunk programot készíteni.

Írassuk ki a 75 egyeseinek helyén található számjegyet, majd a tízesek helyén található is! Gépeljük be a programot a mint alapján!

- Ha két számot elosztunk egymással, akkor a várható módon tizedesre pontosan kiírja a program.
- $(76/10)=7,6$
- Majd a szokott módon húzzunk egy vonalat!
- A 76 egyeseinek helyiértékén 6 áll. Ezt a számot úgy kapjuk meg képlettel, hogy a 76-ot elosztjuk maradékosan 10-el, és így megkapjuk a maradékot. Ebben az esetben a 6-ot!
- **Az osztási maradék műveleti jele a %.** Tehát a két szám közé írunk egy % karaktert a mint alapján!
- A parancsértelmező soraiban látjuk az eredményt!
- Ha az **egész részre van szükségünk** akkor az „integer” (egész) szóból származó **int()** utasítást fogjuk használni! (Az Excel programban is találkoztunk az int() függvénykénnel. Itt is ugyanúgy működik az int(művelet)). Tehát ebben az esetben a futtatott programban a 7-es szám jelenik meg. (Vagy az int() helyett írhatunk // karaktereket is. Így is működik. $75 // 10$)

```

1 print("76 osztva 10-el:", 76 / 10)
2 print("-----")
3 print("Az egyesek helyiértéke:", 76 % 10)
4 print("-----")
5 print("Egész rész: ", int((76/10)))

```



```

Run: 01b
E:\00_MM\12_Python_programozas\prog
76 osztva 10-el: 7.6
-----
Az egyesek helyiértéke: 6
-----
Egész rész: 7

```



(01c.py)

Hozzunk létre egy új python fájlt, a neve legyen 01c!

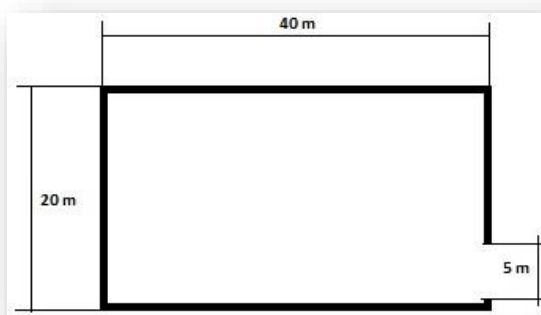
Vásároltál egy telket, amely 20 méter széles, 40 méter hosszú és van rajta egy 5 méteres kapu.

Szeretnéd körbe keríteni. Számold ki, hogy mekkora, hány méter kerítésre lesz szükségünk!

Szeretnéd fűmaggal bevetni az egész kertet. Mekkora területre kell fűmagot vásárolnod?

A feladatot úgy készítsd el, ahogy a képen látod!

- A print() utasítás végén szerepeljenek a mértékegységek! Tehát egy újabb vesszővel válaszd el az idézőjelek közötti szöveget!
- A kerület képlete: $2*(a*b)$
- Ebben az esetben a kapu 5 méterét levonjuk. $2*(20*40)-5$
- A terület képlete: $a*b$
- Ebben az esetben: $20*40$



```

01a.py x 01b.py x 01c.py x
1 print("A kerítés mérete:", 2*(20+40)-5, "m")
2 print("A szükséges fű négyzetmétere:", 40*20, "m2")
3
Run: 01c
C:\Users\kolmank\AppData\Local\Programs\Python\Py
A kerítés mérete: 115 m
A szükséges fű négyzetmétere: 800 m2
Process finished with exit code 0

```

02. VÁLTOZÓK HASZNÁLATA**Változók használata:**

A programozásban nagyon fontos szerepük van a változóknak. Vannak olyan **értékek, amelyeket többször felhasználunk** a programunk során, vagy **mi adunk értékeket egyes esetekben**. Ezeknek a **változóknak „neveket” adunk és a programban ezekkel „hivatkozunk” rájuk**. Tetszőleges nevet adhatunk a változóknak, (lehetőleg olyat, amit könnyen megjegyzünk) majd = jel segítségével értéket adhatunk nekik. Ezt úgy képzeljük el, hogy van egy kamrai polc rendszer, amit felcímkézünk, majd a polcokra különböző dolgokat teszünk. **Egyszerű változók fajtái: egész szám, valós szám, karakterlánc, logikai**

(02a.py)

Hozzunk létre egy új python fájlt, a neve legyen 02a!

Az előző feladatban lévő „telek”-hez visszatérve, számoljuk ki a kerületét és a területét!

- A program elején adjuk értéket az „a” és a „b” oldalnak a minta szerint! $a=38$, $b=52$
- Majd egy „k” változóba számoljuk ki képlettel a kerületet!
- Aztán egy „t” változóba képlettel számoljuk ki a területet, úgy, hogy a program elején definiált a és b változókat használjuk!
- A változókat csak egyszerűen begépeljük a felhasználni kívánt helyen! (Pl.: `print(k); k=2*(a*b)`)
- Arra mindig figyelni kell, hogy csak olyan változót használhatunk fel képletben, amit előbbre definiáltunk!
- A k változóba és a t változóba belekerül a kiszámított érték, amit a következő két sorban a minta szerint kiíratunk!

```

1 a = 38
2 b = 52
3 k = 2*(a+b)
4 t = a*b
5 print("A telek kerülete: ",k,"m")
6 print("A telek területe: ",t,"m2")
7

```

Run: 02a x

```

C:\Users\kolmank\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.9.0.0_q-bcsj8qd9e0j6: Python 3.9.0
A telek kerülete: 180 m
A telek területe: 1976 m2
Process finished with exit code 0

```

Zárjuk be a megnyitott 01-es feladatokat! Csak a 02a.py maradjon megnyitva!

(02b.py)

Hozzunk létre egy új python fájlt, a neve legyen 02b!

Ebben a feladatban egy **változó értékének növelését illetve csökkentését** fogjuk kipróbálni. Gépeljük be az utasításokat!

- Az $a = a+5$ utasítás kicsit megzavarhatja a kezdő programozót, mert ez matematikailag nem helyes. De ebben az esetben ez egy új értéket ad a változónak. Azt mondjuk, hogy az egyenlőség jel az azt jelenti, hogy „**legyen egyenlő**”! Tehát az a értéke legyen egyenlő az eredeti a érték plusz 5-el! Ebben az esetben az értéke 10, akkor e művelet után 15 lesz.
- Tehát nem a bal és a jobb oldal összehasonlítására kell gondolnunk. Erre majd a későbbiekben az $==$ jel fog szolgálni.
- Egy utasításban több értéket is adhatunk, ha a változókat és az értékéül kapott kifejezéseket vesszővel elválasztjuk egymástól.

```

1 a = 10
2 b = 22
3 print(a)
4 print(b)
5 print("-----")
6 a+=5
7 print(a)
8 print("-----")
9 a +=5
10 print(a)
11 print("-----")
12 b -= 8
13 print(b)
14 print("-----")
15 k,t = 2*(a+b),a*b
16 print(k,t)
17

```

Run: 02b x

```

10
22
-----
15
-----
20
-----
14
-----
68 280

```

(02c.py)

Hozzunk létre egy új python fájlt, a neve legyen 02c!

A feladatban számoljuk ki egy **derékszögű háromszög átfogójának méretét**, ha meg van adva a két befogó!

```
02a.py x 02b.py x 02c.py x
1 from math import *
2 #Számoljuk ki Pitagorasz tétellel\
3 #a derékszögű háromszög átfogójának méretét
4 a=5
5 b=6
6 #a^2+b^2=c^2
7 c=sqrt(pow(a,2)+pow(b,2))
8 print(c)
```

Run: 02c x
C:\Users\kolmank\AppData\Local\Microsoft\Windows\Apps\SoftwareLauncherApp\ProgramListings\Python37-64\python.exe
7.810249675906654

- A feladatot Pitagorasz tétellel oldjuk meg. $a^2+b^2=c^2$
- Ahhoz, hogy a négyzetre emelés és a gyökvonás függvényt tudjuk használni egy úgynevezett függvénykönyvtárt kell betöltenünk.
- Ehhez be kell írni az első sorba a: **from math import *** utasítást.
- Ha **megjegyzéseket** szeretnénk írni az utasítások mellé akkor a „**#**” karakter beírása után megtehetjük!
- Mint tudjuk egy **utasításnak** egy sorban kell lennie. Ha mégis **több sorba** akarnánk írni, akkor a „****” karaktert kell használni.
- Ennek a feladatnak a megoldásánaál használnunk kell a **gyökvonás sqrt()** és a **hatványozás pow(alap,kitevő)** függvényeket!
- Gépeljük be és értelmezzük az utasításokat!



Áttekintés és fontos kiegészítések az eddig tanultakhoz:

- A parancsokat parancssorokba írjuk, melyek egymás után hajtódnak végre a futtatás során.
- Egymás alá írt utasítások egységét programnak hívjuk.
- Parancssorok begépelésekor minden sor végén enter kell nyomi. Ezzel jelezzük, hogy befejeztük a parancs begépelését, jöhet a végrehajtás.
- Minden utasítást új sorban kell kezdeni.
- Ha egy utasítás nagyon hosszú több sorban szeretnénk megjeleníteni, akkor a „****” jelet kell használni. (Nem per (/) jel!
- A sorok elején jelentősége van a szóközöknek. Nem mindegy, hogy hol kezdődik az utasítás! Későbbiekben az összetartozó részeket beljebb fogjuk kezdeni.
- Az utasításokon belül szabadon bánhatunk a szóközök használatával!
- A karakterláncokat, (például a print parancsnál) írhatjuk idézőjelek közé vagy aposztrófok közé! De persze nem lehet keverni őket!
- A változó a memória egy darabja, amire a nevével hivatkozhatunk. A nevének az angol abc egyik betűjével kell kezdődnie. A nevének további részében lehet az angol abc betűi, számok és aláhúzás. Nem egyezhet meg a neve egy lefoglalt utasítás nevével (pl.: nem lehet „print” a változó neve)!
- A változókon végrehajtott legfontosabb művelet az értékadás. A x=50 azt jelenti, hogy „az x legyen egyenlő 50-el”.
- Több változón egyszerre is adhatunk értéket: x,y,z=10,20,30
- Amikor értéket adunk egy változónak, azzal eldöntjük a típusát. (Úgy mind az Access adatbázis-kezelésnél.)
- A változó típusai lehetnek: karakterlánc (string); egész számok, valós számok, valamint logikai értékek (True, False)
- A szám változóknál az érték növelésére, csökkentésére, változtatására működnek a +=, -=, *= parancsok. Tehát a bal oldali változó értéket növelik, csökkentik, vagy éppen szorozzák a jobb oldalon megadott értékkel.
- Ha hibaüzenet jelenik meg piros színnel, ez figyelmeztet, hogy valami el lett gépelve, vagy kimaradt valami. A parancsértelmező megpróbálja jelezni, hogy van a hiba. Ahol villog a kurzor ott kell keresni a hibát. Illetve szövegesen is kiírja, hogy mit hibázhattunk el.
- Ha **megjegyzéseket** szeretnénk hozzáfűzni egyes sorokhoz a python programozásakor, akkor a „**#**” karaktert használjuk.

