

**04. FORMÁZOTT KIÍRATÁS**

Az eddigi tananyag során nem alkalmaztunk **sortörés**, nem használtunk **azonos sorban kiíratást** különböző utasításoknál! Egyébként ezeket hívjuk **tagolásnak**.

Eddig amikor kiíratunk a képernyőre eredményeket, nem **formáztuk**, nem adtuk meg, hogy milyen formátumban jelenítse meg az adott számot! Bonyolultabb esetekben a formázott adat kiíratást célszerű jelölőkkel (markarekkel) elkészíteni. Ezek lesznek a „%”-jelek.

Ezek használatát legegyszerűbben példákon keresztül lehet megérteni!

**(04a.py)**

Nyisd meg a 03b.py programot és mentsd el másként 04a.py néven!

Első lépésként alakítsd át a minta alapján, hogy ez legyen a kiindulási helyzet!

```
03b.py x
1 a=int(input("Kérem az a oldal méretét: "))
2 b=int(input("Kérem a b oldal mértét: "))
3 k,t=2*(a+b),a*b
4 print("A téglalap kerülete: ",k)
5 print("A téglalap területe: ",t)
```

```
Run: 03b x
C:\Users\kolmank\AppData\Local
Kérem az a oldal méretét: 6
Kérem a b oldal mértét: 9
A téglalap kerülete: 30
A téglalap területe: 54
```

A célunk az, hogy úgy módosítsuk a programunkat, hogy a jobb oldali képen látható módon jelenjen meg az eredmény! Úgy változtassuk meg, hogy külön print utasítás legyen „A téglalap kerülete: „, valamint a 30! Viszont egy utasítás legyen, de külön sorban „A téglalap területe: „ és az 54!

```
Run: 04a x
C:\Users\kolmank\AppData\Local
Kérem az a oldal méretét: 6
Kérem a b oldal mértét: 9
A téglalap kerülete: 30
A téglalap területe:
54
```

**Tagolt kiíratás**

Tehát, ha két külön utasítást szeretnénk végrehajtani, amit **külön sorba írtunk**, de azt szeretnénk, hogy **egy sorban legyenek megjelenítve**, akkor az első sor utasításának végére vesszővel elválasztva írjuk be az **end=' '** utasítást!

Ha viszont, **szeretném külön sorba tagolni** az utasítás kiíratását, akkor szintén vesszővel elválasztva írjuk az utasítás végére a **sep='\n'** utasítást!

- Ha még jobban kibővítjük a programunkat, akkor biztosan megértjük, hogy hogyan működnek ezek a tagolások!
- Az end=' ' utasítást mindig be kell írni az utasítás végére!
- A sep='\n' -nél pedig minden vesszővel elválasztott tagot külön sorba ír!

```
04a.py x
1 a=int(input("Kérem az a oldal méretét: "))
2 b=int(input("Kérem a b oldal mértét: "))
3 k,t=2*(a+b),a*b
4 print("A téglalap kerülete: ",end=' ')
5 print(k)
6 print("A téglalap területe: ",t, sep='\n')
```

```
04a.py x
1 a=int(input("Kérem az a oldal méretét: "))
2 b=int(input("Kérem a b oldal mértét: "))
3 k,t=2*(a+b),a*b
4 print("A téglalap kerülete: ",end=' ')
5 print(k,end=' ')
6 print(k)
7 print("A téglalap területe: ",t,t,t,t, sep='\n')
```

```
Run: 04a x
C:\Users\kolmank\AppData\Local
Kérem az a oldal méretét: 6
Kérem a b oldal mértét: 4
A téglalap kerülete: 20 20
A téglalap területe:
24
24
24
24
```

## Jelölők használata

A **formázott adatkiírást** célszerű jelölőkkel megoldani. A jelölők százalék jellel kezdődnek, beírhatók a szövegbe, és a print utasítás egy későbbi részén elegendő megadni az értéküket.

Általános jelölő: %s, amelyeket **karakterláncok** helyettesítésére használunk; az **egész számoknál %d**; a **valós számoknál %f**-et adunk meg. Ezek megértéséhez szintén nézzünk példákat!



### (04b.py)

Hozzunk létre egy új python fájlt, a neve legyen 04b.py!

Készítsünk programot, melyben bekérünk két közönséges törtet, majd megjelenítjük egyszerűsítés nélkül a szorzatukat, a következő formátumban:  $15/11 * 5/3 = 75/33$

A változók bekérése a már ismert módon történik!

- A jelölők használatára példát az utolsó sorban látunk.
- A print utasítás zárójelben lévő dolgait három részre bonthatjuk.
- Az első harmadban az idézőjelek között, ahova az egész számok fognak kerülni, ott a %d jelölőt használjuk.
- A második harmadba csak simán beírunk egy % karaktert!
- Az utolsó harmadba pedig zárójelek közé beírjuk a változókat sorban egymás után, ahogy egymás után következnek az első, idézőjeles részben!
- Vizsgáld meg alaposan a program képernyőképét és értelmezd!

```

04b.py x
1  sz1=int(input("Kérem az első tört számlálóját: "))
2  n1=int(input("Kérem az első tört nevezőjét: "))
3  sz2=int(input("Kérem a második tört számlálóját: "))
4  n2=int(input("Kérem a második tört nevezőjét: "))
5  print("%d/%d * %d/%d = %d/%d" % (sz1,n1,sz2,n2,sz1*sz2,n1*n2))

Run: 04b x
C:\Users\kolmank\AppData\Local\Programs\Python\Python38-64\python.exe
Kérem az első tört számlálóját: 16
Kérem az első tört nevezőjét: 17
Kérem a második tört számlálóját: 9
Kérem a második tört nevezőjét: 8
16/17 * 9/8 = 144/136
    
```

### Tizedes törtek megjelenítése adott pontossággal

A valós számoknál a tizedes helyeknek megfelelő megjelenítést a következő módon készítjük el. A **%.2f** azt jelenti, hogy két tizedes pontossággal jelenítjük meg a valós számot.

### (04c.py)

Hozzunk létre egy új python fájlt, a neve legyen 04c.py!

A feladat az, hogy olvasson be a program egy alapszámot és egy értéket! A kérdés az az, hogy az alapnak hány százaléka az érték?

- Gépeljük be a programot!
- Majd értelmezzük a beírtakat!
- A print parancsban az egész számok helyén %d, a valós szám helyén a %.2f jelölő áll!
- Szintén három részre tagolhatjuk a print utasításban lévő parancsokat!
- Figyeljünk, hogy a % karakter ne maradjon ki!

```

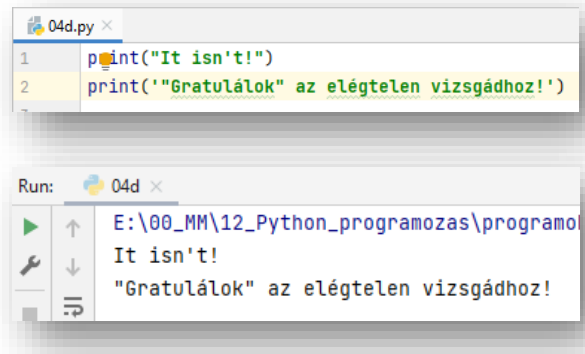
04c.py x
1  a = int(input("Kérem az alapot: "))
2  e = int(input("Kérem az értéket: "))
3  print("alap: %d, érték: %d , százalékláb: %.2f" % (a,e,e/a*100))

Run: 04c x
C:\Users\kolmank\AppData\Local\Programs\Python\Python38-64\python.exe
Kérem az alapot: 654
Kérem az értéket: 123
alap: 654, érték: 123 , százalékláb: 18.81
    
```

### (04d.py)

Hozzunk létre egy új python fájlt, a neve legyen 04d.py!

Előfordulhat, amikor a kiírandó szövegünkben aposztrófokat vagy idézőjeleket kell használnunk. Az idézőjelek által közrefogott szöveg aposztróft is tartalmazhat ("It isn't."), az aposztrófok által határolt szöveg pedig tartalmazhat idézőjelet is ("Gratulálok" az elégtelen vizsgához!).



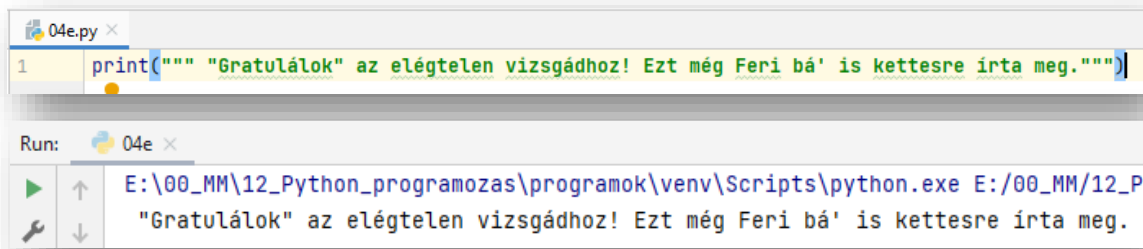
```
04d.py x
1 print('It isn't!')
2 print('\"Gratulálok\" az elégtelen vizsgához!')

Run: 04d x
E:\00_MM\12_Python_programozas\programo
It isn't!
\"Gratulálok\" az elégtelen vizsgához!
```

### (04e.py)

Hozzunk létre egy új python fájlt, a neve legyen 04e.py!

Ha véletlenül olyan szöveget kell kiíratnunk, amelyben egyszerre szerepel aposztróf és idézőjel is, akkor a szövegünket tripla aposztróf, vagy idézőjelek közé kell írni!



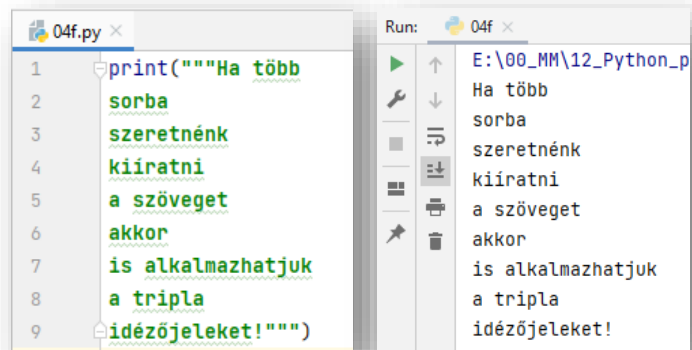
```
04e.py x
1 print('\"\"\" \"Gratulálok\" az elégtelen vizsgához! Ezt még Feri bá' is kettesre írta meg.\"\"\"')

Run: 04e x
E:\00_MM\12_Python_programozas\programok\env\Scripts\python.exe E:/00_MM/12_P
\"Gratulálok\" az elégtelen vizsgához! Ezt még Feri bá' is kettesre írta meg.
```

### (04f.py)

Hozzunk létre egy új python fájlt, a neve legyen 04f.py!

A tripla idézőjelek és aposztrófok alkalmazásakor több sorba írhatjuk ki a szövegünket!



```
04f.py x
1 print('\"\"\"Ha több
2 sorba
3 szeretnénk
4 kiíratni
5 a szöveget
6 akkor
7 is alkalmazhatjuk
8 a tripla
9 idézőjeleket!\"\"\"')

Run: 04f x
E:\00_MM\12_Python_p
Ha több
sorba
szeretnénk
kiíratni
a szöveget
akkor
is alkalmazhatjuk
a tripla
idézőjeleket!
```

A Pythonban vannak úgynevezett **lefoglalt szavak**, amelyeket nem lehet felhasználni egyéb esetben, például nem adhatjuk változó neveknek. Vannak köztük olyanok, amelyeket már ismerünk és vannak olyanok, amelyekkel később fogunk megismerkedni. **Ezek a következők:**

and	as	assert	break	class	continue
def	del	elif	else	except	exec
finally	for	from	global	if	import
in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with
yield	True	False	None		

