



A Python Programnyelv alapjai

Funk Gábor



Előszó

- A Python egy egyszerűen megtanulható és ajánlott programnyelv.
- Érthető, egyszerű és egyes hibákat is figyelmen kívül tud hagyni.
- Szükséges hozzá:
- Python 3.0+
- Pycharm 2020.3.3 Ajánlott



Tartalomjegyzék

- Print() függvény (4-5)
- Mit lehet még csinálni a kiírással? (6-9)
- Változók és input() (10-12)
- Listák (13-14)
- Mátrixok (15)
- Ciklusok (16-19)
- Extension-ok (20)
- Import (21-22)



Print()

A print() függvény egy paraméteres függvény ami alapértelmezetten be van építve a pythonba.

Paramétere lehet string,int,float,list,matrix és még más változók is.

Egy példa amivel kiíratjuk a console-ra/terminálra, hogy Hello World!:

```
Printable = „Hello World!”  
print(Printable) #vagy:  
print(„Hello World!”)
```





Print()

Ha kicsit komplikált dolgokat akarunk kiírni print() segítségével akkor 3 megoldás lehetséges.

```
Num1=„Gábor”  
Num2=32  
  
1.  
print(„Hello, I am ” + Num1 + „, I ate ” + Num2 + „ Pizzas Today.”)  
2.  
print(f””””Hello, I am {Num1} I ate Num2 Pizzas Today.””””)  
3.  
print(„Hello, I am %s I ate %d Pizzas Today.” %(Num1,Num2))
```

Sortörést \n-el lehet csinálni, ha a szövegbe rakjuk.

```
print(„1.st line,\n2.nd line!”)
```





Mit tudunk a Print()-el?

A print() függvénybe lévő szöveget sok módon tudjuk változtatni, formázni. Még érzékelni is tudunk nagy, kis betűt, számot.

A print függvény hasáiban lényegében más függvényeket is kiértékelhetünk, esetleg össze láncolhatunk.

Ezek a függvények lehetnek:

- Formázások
- Ellenőrzések
- Eldöntések
- Saját függvények, amik mást is csinálnak a háttérben.



Formázások



A formázásokkal egy meglévő szöveget módosíthatunk úgy, hogy az csak a kiírásakor módosuljon.

```
Print(„Alap szöveg”.upper()) #Nagy betűt csinál az összes kis betűből
```

```
Print(„Alap szöveg”.lower()) #Kis betűt csinál az összes nagy betűből
```

```
Print(„Alap szöveg”.title()) #A szavak 1. betűjét nagy betűvé alakítja
```

```
Print(„Alap szöveg”.capitalize()) #A mondat 1. betűjét nagy betűvé alakítja
```

Érzékelések



Ezekkel a függvényekkel érzékelhetjük vagy ,megszámolhatjuk', hogy például egy szövegbe mennyiszor szerepel egy betű. Ezek általában paraméteresek.

```
Print(„Alap szöveg.”.__len__()) #Megszámolja, hogy hány betűből áll a szöveg
```

```
Print(„Alap szöveg”.split(„ ”)) #Felbontja a szöveget 1 Paraméterrel
```

```
Print(„Alap szöveg”.count(„e”)) #Megszámolja, hogy a paraméter hányszor szerepel a szövegbe
```

```
Print(„Alap szöveg”.find(„szöveg”)) #Megnézi, hogy benne van-e a paraméter a szövegbe
```


Érzékelések



```
Print(„Alap szöveg”.startswith(„A”)) #Megnézi, hogy a paraméterrel kezdődik-e a szöveg
```

```
Print(„Alap szöveg”.endswith(„.”)) #Megnézi, hogy a paraméterrel végződik-e a szöveg
```

Példa egy alkalmazásra: Meg szeretnénk nézni, hogy az USA mennyiszer van egy szövegbe. Az a baj, hogy van ahol usa és van ahol Usa.

```
Print(„Welcome to USA.Usa is awesome, isn't it?Let's go to usa.”.upper().count(„USA”)) #Először nagy betűvé alakítjuk aztán legszámoljuk, hogy hányszor van meg az USA szó a szövegben.
```



Változók és Input()

A Változók olyan értékek amik vagy konstansok vagy változnak. Ilyen értékeket az input() segítségével kérhetünk be:

```
Val=0 #input előtt is megadhatunk változót de nincs sok értelme.
```

```
Val=input(print(„Milyen szám legyen?")) #Itt az a baj, hogy nem csak számot tud megadni és mivel az input string-ként vesz mindent ezért például a 7 nem 7 lesz hanem str(7). Ha számmá akarjuk konvertálni az inputot akkor ezt kell beírnunk:
```

```
Val=int(input(print(„Milyen szám legyen?")))) #(int = integer)
```





Változók és Input()

Sok féle változó van. A Python 3-ban szereplő alap változók az str,int,float,list és logikai változó.

```
Teststring=„Szöveg” #A string változó szöveges, lehet benne akármilyen karakter. Megadáshoz szükséges „”.
```

```
Testint=7 #Az integer értéke szám lehet
```

```
Float=8*20/3 #A float egy olyan változó amelynek műveletekkel is meg lehet adni az értékét. (Lehet tört és negatív szám is.)
```

```
isTrue= True #A logikai változónál az érték csak True vagy False lehet.
```





Változók és Input()

Itt van egy példa a változók használatára. Két számot bekérünk és összeadunk:

```
Num1=int(input(print(„1.szám:”)))  
Num2=int(input(print(„2.szám:”)))  
Num3=int(Num1+Num2)  
print(Num3)
```



```
1. szám:  
None  
2. szám:  
None  
2  
  
Process finished with exit code 0
```





Listák

A listák olyan változók amelyek több elemet tudnak eltárolni. Az elemek indexelése 0-val kezdődik. Jelölésük: [] vagy [0].

```
Lista=[] #Üres lista.
```

```
Lista=[0,10,32,56,34] #Számokat tartalmazó lista.
```

```
Lista=[„Ádám”,„Ábel”,„Béla”] #Szöveget tartalmazó lista.
```

```
Lista=[0,„Ádám”,„1”] #Vegyes lista.
```





Listák

Listákat sokféle képpen ki lehet írni.

Tartalmát:

```
Lista=[0,1,2,3,4,86,5,6]  
print(Lista)
```



```
Run: project1_9-1 x  
C:\Users\Gabcı\PycharmProjects\pyth  
[0, 1, 2, 3, 4, 86, 5, 6]  
Process finished with exit code 0
```

Hosszát:

```
Lista=[0,1,2,3,4,86,5,6]  
print(len(Lista))
```

```
8  
Process finished with exit code 0  
Run | TODO | Problems | Terminal | Python Console
```





Mátrixok

A 2 dimenziós listákat mátrixnak nevezzük. (Listában Lista)
Jele: `[] []` vagy `[[]]`. Hasznos 2 Dimenziós játékoknál PL: sakk, aknakereső, labirintus.

```
Matrix=[[0,1,2,3,4,5,6,7,8,9],[9,8,7,6,5,4,3,2,1]]  
print(Matrix) #Ugyanúgy fog kiíródni mint a Lista csak az új  
belső listát új sorba fogja írni.
```





Ciklusok

- A ciklusok megadott parancsokat ismételnék és végrehajtanak megadási értékkel vagy anélkül.
- Főbb ciklusok a For(), While(), Do Until...

<https://www.youtube.com/watch?v=6iF8Xb7Z3wQ>





For()

A for egy előtesztelős egy vagy több megadási értékkel rendelkező függvény. Van kezdőértéke, visszatérési értéke és hatóköre is. Példa:

```
for i in range(10): #Az i a kezdőértéke és 0-10-ig fut le  
    print(i) #Kiírja az i értékét.
```

```
List=[1,2,3,4,5,6,7]  
for i in range(len(List)): #Az i a lista hossza-szor ismétlődik.  
    print(List[i]) #kiírja a lista i-edik értékét.
```





While()

A while egy előtesztelős ciklus aminek 1 vagy több megadási értéke lehet. Példa:

```
Xy=1 #Változó  
while Xy != 10: #Amíg a változó nem egyenlő 10 addig ismétél.  
    print(Xy)  
    Xy+=1 #Megnöveli 1-el a változó értékét.
```





Do Loop

A Do egy hátultesztelős ciklus aminek 1 vagy több paramétert adhatunk meg. (Ilyen pontosan nincs Python-ban, ezt egy while ciklus módosításával lehet elérni.)

```
Num=1
while True:
    print(Num)
    Num += 1
    if i > 5: #Addig ismétli amíg az 5 nem nagyobb 5-nél.
        break #Véget vet az ismétlésnek.
```





Extension-ok

A python-hoz vannak úgy nevezett extension-ok, ezekkel lehet bizonyos dolgokat egyszerűbben elkészíteni.

- ❖ PyQt5 – GUI builder könyvtár a python-hoz
- ❖ Tkinter – GUI megjelenítő a python-hoz
- ❖ Cython – C programnyelvbe átírható a Python kód

Egy példa az extension-ok használatára:

<https://www.youtube.com/watch?v=jE-SpRI3K5g>





Import

Bizonyos könyvtárakat lehet beépíteni a python programba az import segítségével. Példák:

```
import random #véletlen szám.  
import system #rendszerbe lévő adatok lekéréshézes  
használatos.  
from PyQt5.QtCore import * #Specifikált import egy  
könyvtárból vagy extension-ból.
```





Import használata

Vannak bizonyos dolgok amit nem lehet megcsinálni a Python beépített alapfunkciójával. Erre jó az import. Példa egy véletlen szám generálására:

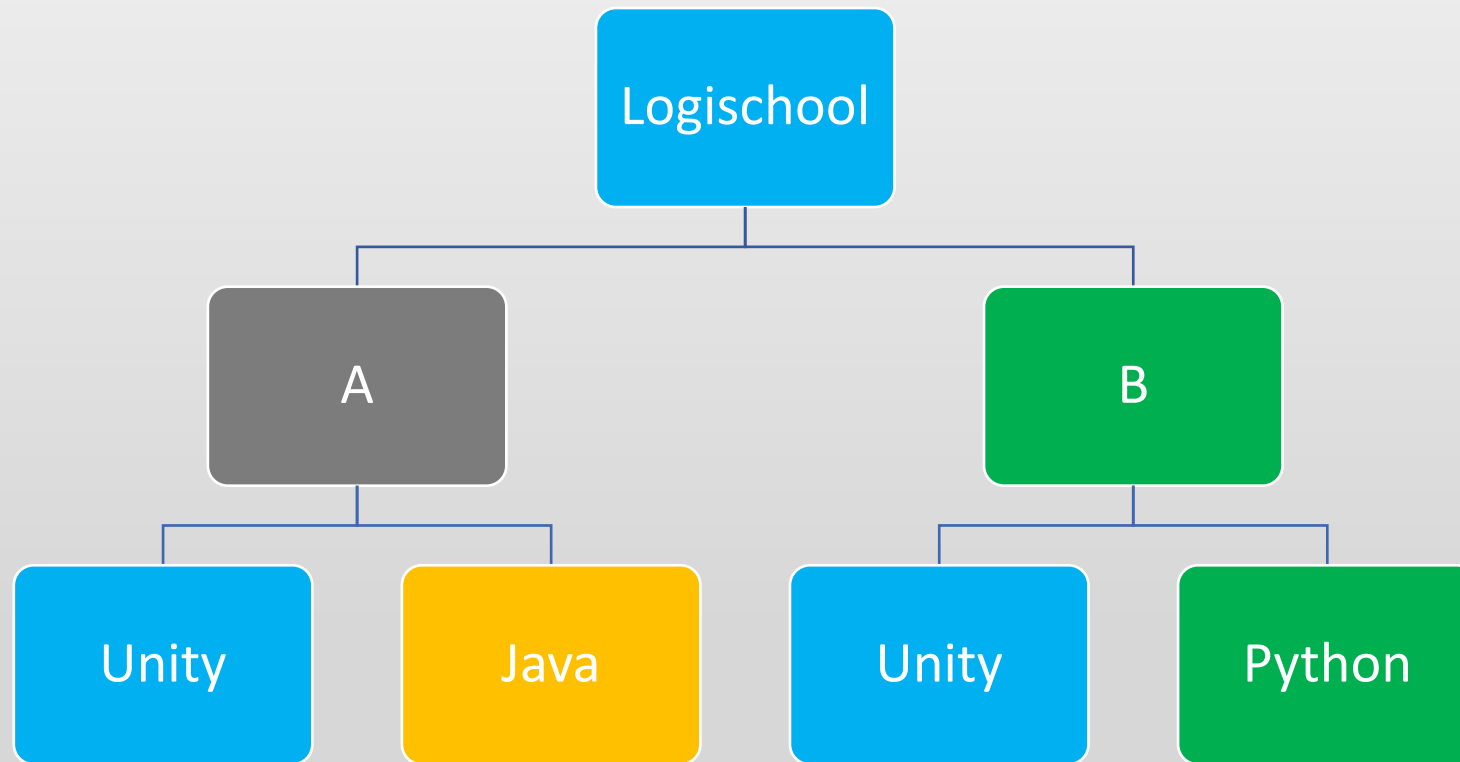
```
import random
Var = random.random() #Megadunk a random segítségével
egy véletlen számot a Var változónak.
print(Var)
```





Hol tanítják?

Ha valaki Python-t akar tanulni akkor a Logischool jó hely neki. Egy féléven belül megtanít annyi anyagot amennyit egy főiskola 1 év alatt.





Végszó

A Python a legjobb programnyelv kezdők számára mivel sokkal egyszerűbb, elsajátíthatóbb és nem olyan bonyolult mint egy C++ vagy Java. Az egyik nagy előnye az, hogy kiírja a hibákat sor és betű pontossággal, nem úgy mint a Java. Nem kell speciális karakter minden egyes utasítás végére és ha nem használjuk az else() ágot akkor még energiatakarékos* is. Ha valaki el akar kezdeni programozni tanulni akkor a Python-t nagyon ajánlom neki.

