

PY		
Megjegyzések, kommentek		
<p># ez egy megjegyzés</p> <p>A Pythonban a megjegyzések kettőskeresztrel ('#') kezdődnek és a sor végéig tartanak. Egy megjegyzés lehet sor elején, vagy követhet szóközt, tabulátor-karaktert, de ha egy karakterlánc (string) belsejébe tesszük, az nem lesz megjegyzés. A kettőskereszt karakter egy karakterláncon belül csak egy kettőskeresztet jelent.</p> <p>” valami # valami”</p>		
Utasítások lezárása		
nincs		
Kiírás		
<pre>print("alma") alma print("alma", "körte") alma körte print("a"+"b") ab</pre>		
Változók		
<p>nem típusos</p> <p>str, int, float, bool</p> <p>típus lekérdezése:</p> <pre>type(változónév) a=2 print(type(a)) int</pre>		
Beolvasás		
<pre>input() input("Kérem... ")</pre>		
Konvertálás		
<pre>a=int(input("Kérem... "))</pre>		
Értékadás		
=		
Aritmetikai műveletek		
<pre>+ - * / % (maradék) ** (hatványozás) // (egész osztás) += -= *= /=</pre>		
Logikai műveletek		
<pre>=, &lt;, &lt;=, &gt;, &gt;=, !=, not, and, or, is, is not</pre>		
A Math osztály használata		
importálni kell:	import math	from math import *
Pi	math.pi	pi
felfelé kerekítés	math.ceil(szám)	ceil(szám)
lefelé kerekítés	math.floor(szám)	floor(szám)
kerekítés	round(szám)	round(szám)
hatványozás	pow(alap, kitevő)	pow(alap, kitevő)
gyökvonás	math.sqrt(szám)	sqrt(szám)
minimum	min(szám)	min(szám)
maximum	max(szám)	max(szám)
abszolútérték	abs(szám)	abs(szám)
rad → fok	math.degrees(radián)	degrees(radián)
fok → rad	math.radians(fok)	radians(fok)
szögfüggvények	math.sin(rad), math.cos, math.tan, math.asin(valós), math.acos, math.atan	sin(rad), cos, tan, asin(valós), acos, atan

Véletlenszám		
importálni kell:	<i>import round</i>	<i>from round import *</i>
[tól; ig]	<i>random.randint(tól, ig)</i>	<i>randint(tól, ig)</i>
[tól; ig)	<i>random.randrange(tól, ig)</i>	<i>randrange(tól, ig)</i>
[0; 1)	<i>random.random()</i>	<i>random()</i>
Sztringek		
'vagy aposztrófok között' "vagy macskakörmök között" "" 'Igen', válaszolta" ' "Igen", válaszolta' \-t használunk, ha nem fér el egy sorban a parancs de: \, , \", " \\ \\ \n New Line, sortörés \t Tab \b BackSpace """" Háromszoros idézőjelben a speciális karakterekhez nem kell \ """"		
A sztring egy szekvencia, nem lehet karaktert kicserélni benne. A karakterre az indexével hivatkozunk (0-tól kezdődik az indexelés)		
szo='Programozás'		
mit csinál	utasítás	eredmény
3-as indexű elem	<i>szo[3]</i>	<i>g</i>
3-5 indexű elem	<i>szo[3:6]</i>	<i>gra</i>
0-2 indexű elem	<i>szo[:3]</i>	<i>Pro</i>
3-tól a végéig	<i>szo[3:]</i>	<i>gramozás</i>
Negatív index (a végétől számol)		
-5, -4, -3	<i>szo[-5: -2]</i>	<i>moz</i>
utolsó 5	<i>szo[-5:]</i>	<i>mozás</i>
utolsó 2 nélküli	<i>szo[:-2]</i>	<i>Programoz</i>
hossza, karakterek száma	<i>len(szo)</i>	<i>11</i>
levágja a szóközt	<i>szo.strip()</i> <i>szo.lstrip()</i> <i>szo.rstrip()</i> <i>szo.strip('a')</i> <i>szo.strip('az')</i>	<i>elejétől és a végétől balról</i> <i>jobbról</i> <i>levágja az 'a'-t</i> <i>levágja az 'az'-t</i>
kisbetűssé alakít	<i>szo.lower()</i>	<i>programozás</i>
nagybetűssé alakít	<i>szo.upper()</i>	<i>PROGRAMOZÁS</i>
karaktercsere akár több szomszédos karaktert is lehet	<i>szo.replace('ás', 'ó')</i>	<i>Programozó</i>
tartalmazza-e	<i>mit in szo</i> <i>'a' in szo</i> <i>'i' in szo</i>	<i>true</i> <i>false</i>
nem tartalmazza	<i>mit not in szo</i> <i>'a' not in szo</i> <i>'i' not in szo</i>	<i>false</i> <i>true</i>
Mondatkezdő	<i>szo.capitalize</i>	<i>Programozás</i>
előfordulás száma	<i>szo.count('o')</i> <i>szo.count('oz')</i>	<i>2</i> <i>1</i>
megkeres valamit, kezdőindexet adja, ha nincs akkor -1	<i>szo.find('o')</i> <i>szo.find('oz')</i> <i>szo.find('i')</i>	<i>2</i> <i>7</i> <i>-1</i>

megkeres valamit, utolsó előfordulás indexét adja, ha nincs akkor -1	<i>szo.rfind('o')</i> <i>szo.rfind('i')</i>	7 -1
x karakter helyet foglal	<i>szo.center(15)</i>	Programozás
. e az utolsó	<i>szo.endswith('.')</i>	false/true
H-val kezdődik-e	<i>szo.startswith('H')</i>	false/true
csak számokból áll-e	<i>szo.isalnum()</i>	true/false
csak betűkből áll-e	<i>szo.isalpha()</i>	true/false
csupa kicsi-e	<i>szo.islower()</i>	true/false
csupa nagy-e	<i>szo.isupper()</i>	true/false
kicsiből nagyot, nagyból kicsit csinál	<i>szo.swapcase()</i>	pROGRAMOZÁS
feldarabol szóköz mentén feldarabol 'o' mentén a darabokat listába teszi	<i>szo.split()</i> <i>szo.split('o')</i>	Programozás Pr gram zás
minden szó nagy kezdőbetűvel	<i>szo.title()</i>	Programozás

### Adatszerkezetek

Tömbök: nincs

Listák: rendezett és változtatható gyűjtemény

A listát [ ] zárójelekkel jelezzük.

```

nev=[ ]           üres lista létrehozása
listam=[1, 'alma', 'citrom'] lista létrehozása elemekkel
print(listam)     kiírja a lista elemet
print(listam[0])  1
print(listam[-1]) citrom           utolsó elem
listam[0:2]       1, alma
listam[1:]        alma, citrom
listam[:2]        1, alma
listam[0]='ribizli'  átírható
listam[
listaelemek kiírása 'szépen':
for item in listam:
    print(item)

```

mit csinál	utasítás	eredmény
lista elemszáma	<i>len(listam)</i>	3
listaelem hozzáadása	<i>listam.append('narancs')</i>	1, alma, citrom, narancs
elem beszúrása adott pozícióba	<i>nev.insert(index, érték)</i> <i>listam.insert(1, 'eper')</i>	1, eper, alma, citrom, narancs
adott elem indexének lekérdezése	<i>listam.index(elem)</i>	
elem törlése	<i>listam.remove('alma')</i>	
utolsó elem eltávolítása	<i>listam.pop()</i>	
az adott indexű elem eltávolítása	<i>listam.pop(index)</i> <i>listam.pop(1)</i> <i>del listam[0]</i>	
lista törlése	<i>del listam</i>	
a lista ürítése(elemek törlődnek, lista nem)	<i>listam.clear()</i>	
lista rendezése	<i>listam.sort()</i>	
lista elemei fordított sorrendben	<i>listam.reverse()</i>	
lista másolása, nem csak hivatkozás lesz	<i>ujlista=regilista.copy()</i> <i>ujlista=list(regilista)</i>	
két lista összefűzése	<i>lista1.extend(lista2)</i> <i>lista3=lista1+lista2</i>	

<p>Tuple: hasonló mint a lista, de nem lehet módosítani  az elemei vesszővel vannak elválasztva  nev=('a', 'b', 2)  Elsősorban ott használjuk, ahol biztosak akarunk lenni, hogy tévedésből nem módosítjuk a programban.  Kevesebb memóriát is foglal.</p>
Set: nem tanítjuk
Dictionary: nem tanítjuk
Vezérlési szerkezetek
Elágazások
<p><i>if feltétel:</i>  utasítások, ha igaz</p> <p><i>if feltétel:</i>  utasítások, ha igaz  <i>else:</i>  utasítások, ha igaz, el is maradhat</p> <p><i>if feltétel:</i>  utasítások, ha igaz  <i>elif feltétel:</i>  utasítások, ha igaz  <i>else:</i>  utasítások, ha igaz, el is maradhat</p> <p>itt nincs switch-case szerkezet</p>
Ciklusok
<p>lista elemeinek elérése  <i>for i in listanév:</i>  utasítások</p> <p>sztring karaktereinek elérése:  <i>for i in sztring:</i>  utasítások</p> <p>”számlálós” ciklus:  <i>for i in range(6):</i> ez a ciklus 6x fut le, i=0-5, lépésköz 1  utasítások</p> <p><i>for i in range(2,6):</i> ez a ciklus 4x fut le, i=2-5, lépésköz 1  utasítások</p> <p><i>for i in range(1,30, 4):</i> ez a ciklus, i=1-29 között fut le, lépésköz 4  utasítások</p> <p><i>while feltétel:</i>  utasítások</p> <p>A while ciklusnak, ha van ciklusváltozója, akkor azt a ciklus előtt kell megadni, és a feltételben szerepelnie kell, az utasítások között pedig gondoskodnunk kell a változtatásáról</p> <p>kiugrási lehetőség: <i>break</i>  folytatás: <i>continue</i></p> <p>pl:  <i>i = 0</i>  <i>while i &lt; 6:</i>  <i>  i += 1</i>  <i>  if i == 3:</i>  <i>    continue/break</i>  <i>  print(i)</i></p> <p>itt nincs hátultesztelő ciklus</p>

## Függvények, eljárások

Általában a programkód elején adjuk meg a függvényeket.

utasítás	meghívása:
<i>def fvnev(paraméter/ek):</i> <i>utasítások</i>	<i>fvnev( érték/ek)</i>
<i>paraméter nélkül:</i> <i>def fvnev():</i> <i>utasítások</i>	<i>fvnev()</i>
<i>def fvnev(p1, p2...):</i> <i>utasítások</i>	<i>fvnev( é1, é2...)</i> pl: <i>print(fvnev(é1))</i> <i>x=fvnev(é1)</i>

## Objektumok

### Osztály:

A programkód elején helyezzük el.

<i>class Nev:</i> <i>def __init__(self, egysor):</i>  <i>darabok=egysor.strip('\n').split('\t')</i>  <i>self.m1=darabok[0]</i>  <i>self.m2=int(darabok[1])</i>  <i>self.m3=darabok[2]</i>	<i>#osztály létrehozása Nev néven</i> <i>#a konstruktor(metódus) létrehozása, a</i> <i>self kötelező paramétere, az egysor az</i> <i>a paraméter, amit a főprogram átad</i> <i>feldolgozásra</i> <i>#a kapott forrást feldaraboljuk, az</i> <i>elválasztókarakter (most tabulátor)</i> <i>mentén</i> <i>#a megfelelő daraboknak létrehozuk a</i> <i>mezőket(változók, m1, m2, m3... a self</i> <i>kötelező előtte</i> <i>#ha kell, akkor átalakítjuk pl szám</i> <i>típusúvá</i>
--	--

```
nevek=[ ]
```

```
for egysor in beolvas:
```

```
    nevek.append(Nev(egysor)) #itt hívjuk meg a konstruktort, töltjük fel a nevek nevű tömböt az adatokkal
```

```
#hivatkozás az egyes elemekre:
```

```
nevek[0].m1 #A nevek lista első elemének az m1 mezőjének értékét adja
```

## Fájlkezelés

Fájlkezelőt kell használni:

```
f=open('filenév.kiterjesztés', 'megnyitás módja')
```

```
'r'      olvasásra
```

```
'a'      ha nincs létrehozva, ha van, hozzáfűz
```

```
'w'      írásra, ha nincs, létrehozva, ha van, felülírja
```

```
'x'      újat hoz létre, ha létezik, hibát ad
```

Ellenőrzés, hogy létezik, e a fájl:

```
def létezikE(filenév):
```

```
    try:
```

```
        f=open(filenév, 'r')
```

```
        f.close()
```

```
        return 1
```

```
    except:
```

```
        return 0
```

```
if létezikE(filenév):
```

```
    utasítások
```

```
else:
```

```
    print('A fájl nem létezik')
```

**Beolvasás:**

soronként:

```
f=open(filenév, 'r')
while True:
    sor=f.readline()
    if len(sor)==0:
        break
    else:
        print(sor)
f.close()
```

soronként, listába:

```
f=open(filenév, 'r'):
listam=f.readlines()
f.close()
```

teljes tartalmat egy sztringbe:

```
f=open(filenév, 'r'):
tartalom=f.read()
f.close()
```

első 5 karakter beolvasása:

```
tartalomrészlet=f.read[5]
```

**Írás:**

```
f=open(filenév, 'w'/'a')
f.write('első')
f.write('második')
f.close()
```

**Fájl törlése:**

Fájl törléséhez importálnia kell az os modult és futtatnia kell az os.remove() függvényét:

```
import os
if os.path.exists("file.txt"):
    os.remove("file.txt")
else:
    print("A fájl nem létezik")
```

**Mappa törlése:**

```
import os
os.rmdir("mappa")
```

Mappa váltása:

```
import os
os.chdir("home/mappa/almappa")
```