# Mit kell tudni Pythonból (alapórán)?

## Típusok

A Pythonban minden adat objektum, és minden objektumnak van típusa (osztály).

| Alaptípusok: **int**, **float**, **str**, **bool** | 15, 15.0, 6.02e+23, "medve", True, False |
| --- | --- |
| Sorozat típusok: **list**, **range**, str (mint karaktersorozat) | [2,3,5,7], range(1,101) |

Az osztályoknak van konstruktorfüggvénye, amely előállít egy adott osztályú objektumot; sokszor konverziós függvényként is működik.

| **int**() | int("79")-->79 |
| --- | --- |
| **float**() | float("3.14")-->3.14 |
| **str**() | str(51)-->"51" |
| **bool**() | bool(0)-->False, bool("nem")-->True |
| **list**() | list()-->[], list("ló")-->["l","ó"] |
| **range**() egész számok sorozata | range(10,20) |

## Speciális szövegek

A Pythonban a szövegek " vagy ' jelek között vannak. A szövegben előforduló \ speciális karaktert jelent.

"**\n**" újsor-jel

"**\t**" tabulátor

"**\"**" idézőjel

"**\\**" backslash karakter

## Objektum létrehozása/módosítása

| név**=**érték létrehoz egy objektumot és egy hivatkozást a név változóban | a=5 |
| --- | --- |
| név1=név2 létező objektumhoz újabb nevet hoz létre | b=a |
| név**+=**érték, név**-=**érték, név**\*=**érték szám típusú objektum értékét növeli/csökkenti | a+=1  ugyanaz, mint a=a+1 |

## Műveletek

| Számok között: **+**, **-**, **\***, **/**, **%**, **//**, **\*\*** | 15/6→2.5, 15//6→2, 15%6→3 |
| --- | --- |
| Listák és szövegek között: +, \* | [1,2]+[5,6]→[1,2,5,6]  "ba"\*2→"baba" |
| Relációk: **<**, **>**, **==**, **<=**, **>=**, **!=** | 5<3→False |
| Logikai műveletek: **and**, **or**, **not**, **in**, **not in** | "ed" in "medve"→True |
| Indexelés: sorozat**[**index**]** valahányadik elem, 0-val kezdődik, negatív index hátulról megy | range(1,5)[2]-->3  [1,2,3,4,5][-2]-->4 |
| Részsorozat: **[**mettől**:**meddig**]**, ahol a meddig már nincs benne; negatív érték hátulról megy; a kezdőelem mindig 0 | "medve"[-1]-->"e"  range(1,10)[2:5]-->range(3,6)  [2,4,6,8,10][2:]-->[6,8,10] |

## Alapfüggvények

A függvénynek van valahány kötelező paramétere (akár nulla is lehet), és nevesített paraméterei alapértelmezett értékkel. Ha ezeket nem adjuk meg, az alapértelmezett érték lép érvénybe.

A függvénynek lehet valamilyen hatása (a print() kiír valamilyen értéket) és visszatérési értéke (a round() nem ír ki semmit, viszont ad eredményt).

| **print**(...,...,sep=" ",end="\n",file=...) | print(1,2,3)-->1 2 3  print(1,2,3,sep="\*",end="!")-->1\*2\*3! |
| --- | --- |
| **round**(...,ndigits=...) | round(1.294,1)-->1.3 |
| **min**(sorozat) csak olyan sorozatra, melynek tagjai összehasonlíthatók a < művelettel!  **max**(sorozat)  sorted(sorozat) | min("medve")-->"e"  min([2,1,3,4)-->1]  sorted([5,3,4])-->[3,4,5] |
| **sum**(sorozat) csak számokat tartalmazó sorozatra | sum(range(1,101))-->5050 |
| **len**(sorozat) | len("medve")-->5 |
| **input**(str) str-t kiírja, majd a begépelt szöveget adja eredményül |  |

## Metódusok

A metódus (tagfüggvény) egy adott osztályú objektumra vonatkozó függvény. Ezek némelyike eredményt ad (str metódusai), más pedig módosítja az objektumot (list metódusai).

Szövegek metódusai:

| str.**isupper**()  str.**islower**()  str.**isalpha**() | "MEDVE".isupper()-->True  "Medve".islower()-->False  "2 medve".isalpha()-->False |
| --- | --- |
| str.**upper**()  str.**lower**() | "ló".upper()-->"LÓ" |
| str.**strip**() levágja a "whitespace" karaktereket a szöveg széléről, ezek: szóköz, \n, \t | " medve ".strip()-->"medve" |
| str.**split**(elválasztó) szövegből listát csinál | "3 L-es tojás".split(" ")-->["3","L-es","tojás"] |
| str.**count**(szövegrészlet) | "kerge kecske".count("ke")-->3 |
| str.replace(mit,mire,hányat=-1) | "kecske".replace("e","a")-->"kacska"  "kecske".replace("k","f",1)-->"fecske" |

Listák metódusai

| list.**append**(elem) | l=[8,1]  l.append(2)  print(l)-->[8,1,2] |
| --- | --- |
| list.sort() | l.sort()  print(l)-->[1,2,8] |
| list.**index**(elem) az első olyan elem indexe | [1,2,8,2].index(2)-->1 |
| list.**count**(elem) hányszor szerepel | [1,2,8,2].count(2)-->2 |

## Vezérlési szerkezetek

A beljebb kezdés struktúráknál Pythonban kötelező!

| **if** bool:  utasítások  **elif** bool:  utasítások  **else**:  utasítások  az elif és else rész kihagyható |  |
| --- | --- |
| **while** bool:  utasítások | a=1  while a<1000:  print(a)  a\*=2 |
| **for** név **in** sorozat:  utasítások | for x in [1,2,3]:  print(x)  →1 2 3 |
| **break** megszakítja a while és for ciklust | while True:  if input("?")=="\*":  break |
| pass nem csinál semmit |  |
| **def** függvény(paraméter,...,paraméter=érték,...):  utasítások  **return** érték | def növel(a,b=0):  return a+b  print(növel(8))-->8  print(növel(8,2))-->10 |

## Modulok

Külső modul importálása:

**import** modulnév

### A random modul elemei:

| random.**randrange**(mettől,meddig) | random.randrange(1,5)-->3 (vagy más) |
| --- | --- |
| random.random() [0,1[ intervallumból float érték  random.seed(kezdőérték) enélkül a véletlenszámok mindig mások lesznek, ezzel a véletlen sorozatok reprodukálhatóak |  |

### A turtle modul elemei

| **Turtle**() konstruktor | teknőc=turtle.Turtle() |
| --- | --- |
| **Screen**() konstruktor | képernyő=turtle.Screen() |
| turtle objektum metódusai:  forward(n) / fd  back(n) / bk  left(n) / lt  right(n) / rt  penup() /pu  pendown() / pd  color(r,g,b)  width(n)  clear() | teknőc.fd(100) |
| screen objektum metódusai:  colormode(1.0/255)  bgcolor(r,g,b) | képernyő.colormode(255)  képernyő.bgcolor(10,100,200) |

## Fájlkezelés

fájl=**open**(fájlnév,"r/w/a",encoding=kódtábla) a továbbiakban a fájl változóval hivatkozunk a fájlra

r=olvasás, w=új fájl létrehozása írásra, a=létező fájl megnyitása hozzáfűzésre

szöveges fájl, a beolvasás soronként történik, a beolvasott sorok végén megmarad a sorvégjel

a sorokat csak egymás után lehet beolvasni

a kódtábla pl. "utf-8", vagy "cp1250"

Beolvasás és kiírás:

f.**readline**() a következő beolvasott sor vagy üres, ha vége a fájlnak

f.**write**(str) szöveg kiírás a fájlba sorvégjel nélkül

print(...,**file**=fájl) képernyő helyett fájlba ír, sorvégjelet is

for sor in file: a fájl sorait egyesével olvassa be a sor változóba (úgy megy végig rajta, mint egy sorokat tartalmazó listán)

Példa soronkénti beolvasásra kétféleképpen:

f=open("adat.txt","r","encoding="utf-8")

while True:

sor=f.readline()

if sor=="":

break

sor=sor.strip()

print(sor)

f=open("adat.txt","r","encoding="utf-8")

for sor in f:

sor=sor.strip()

print(sor)

## Technikák

### Összetett listák táblázatok kezelésére:

Egy lista elemei listák is lehetnek.

lista=[ ["ló",4] , ["csirke",2] , ["pók",8] , ["ember",2] ]

for adat in lista:

if adat[1]==2:

print(adat[0])

### Tagolt sorokat tartalmazó fájl kezelése:

Minden sorát listaként olvassuk be, összetett listában tárolhatjuk; a fejlécet külön kezeljük. (A \ karaktert az útvonalban speciális karakterként kell kezelni!)

f=open("c:\\feladat\\adat.txt","r","encoding="utf-8")

f.readline()

adatok=[]

for sor in f:

sor=sor.strip()

adat=sor.split("\t")

adatok.append(adat)