

Python tanfolyam

4. alkalom

Vezérlési szerkezetek

- A programozásban a következő vezérlési szerkezeteket különböztetjük meg:
 - Szekvencia: Utasítások végrehajtása egymás után. Igazából ezt sose nevezzük a nevén, a szekvencia általában mást jelent.
 - Elágazás: Valamilyen feltétel teljesülése alapján a program vezérlése 2 vagy több irány közül az egyiken megy tovább.
 - Ciklus: Egy programrész többszöri ($0 \leq x \leq \infty$) lefuttatása valamilyen feltétel teljesülése esetén.
- Ezeknek a vezérlési szerkezeteknek a kombinációjával bármilyen programozási feladatot meg lehet oldani

Logikai kifejezés

- Logikai kifejezésnek nevezzük azokat a kifejezéseket, amiknek az eredménye IGAZ vagy HAMIS. True vagy False.

- Például:

```
x = 3 # Ez egy sima értékadás
```

```
x == 5 # Ez egy logikai kifejezés, mely ebben az esetben HAMIS lesz.
```

```
x == 3 and False # Ez egy logikai kifejezés, mely ebben az esetben HAMIS
```

```
x == 3 or False # Ez egy logikai kifejezés, mely ebben az esetben IGAZ
```

Elágazás – Szintaxis és szemantika

if LOGIKAI_KIFEJEZÉS:

#Utasítások, melyek akkor hajtódnak végre, ha a LOGIKAI_KIFEJEZÉS igaz

elif LOGIKAI_KIFEJEZÉS_2:

#Utasítások, melyek akkor hajtódnak végre, ha LOGIKAI_KIFEJEZÉS hamis

#és LOGIKAI_KIFEJEZÉS_2 igaz.

#Nem kötelező, tetszőleges számú ilyen blokk lehet.

else:

#Utasítások, melyek akkor hajtódnak végre, ha az összes fenti

#logikai kifejezés hamis.

#Ez se kötelező blokk.

While ciklus – Szintaxis és szemantika

while LOGIKAI_KIFEJEZÉS:

#Ezeket az utasítás1-et és utasítás2-t addig ismétli a program,
#amíg a LOGIKAI_KIFEJEZÉS igaz.

utasítás1

utasítás2

utasítás3

#utasítás3 csak akkor lesz végrehajtva, ha LOGIKAI_KIFEJEZÉS hamis

Gyakorlás

- Készítsünk programot,
 - mely bekér a felhasználótól egy számot
 - Ha az páros szám, akkor egyesével megnöveli 10-szer, és kiírja a számokat, soronként egyet.
 - Ha páratlan a szám, akkor, akkor 10-zel csökkenti, kettesével, és kiírja a számokat, soronként egyet.
 - Ha a megadott érték nem szám, akkor kiírja, hogy **Hibás input**.
- Továbbfejlesztés:
 - A program mindaddig kérjen be egy számot, amíg hibás inputot kap.

Elemenkénti feldolgozás

- Sokszor előfordul, hogy valamilyen lista szerű adatszerkezet minen egyes elemét fel kell dolgozni
 - Szakszóval végig kell rajtuk iterálni.
- Ezekre az esetekre a for, vagy más néven for-each ciklusok alkalmasak.
- A for ciklus mindig kiváltható **while** ciklussal, legfeljebb többet kell hozzá kódolni.

For ciklus – Szintaxis és szemantika

for VÁLTOZÓ in ITERÁLHATÓ_KIFEJEZÉS:

```
#Ezeket az utasítás1-et és utasítás2-t addig ismétli a program,  
#amíg van elem az ITERÁLHATÓ_KIFEJEZÉS-ben.
```

```
#A for ciklus hatókörében a VÁLTOZÓ elérhető, és mindig az aktuális  
#elemet jelenti
```

```
utasítás1
```

```
utasítás2
```

```
utasítás3
```

```
#utasítás3 csak akkor lesz végrehajtva, ha az összes elemet  
#feldolgoztuk az ITERÁLHATÓ_KIFEJEZÉSBEN
```


Range

- Számsorozatok generálására a `range()` függvényt használhatjuk.
- Használata:
 - `range(5)`
 - 0,1,2,3,4
 - `range(5, 10)`
 - 5,6,7,8,9
 - `range(0, 10, 2)`
 - 0,2,4,6,8
- Használható for ciklusban is:

```
for v in range(5,10):  
    print(v)
```