

Szkriptnyelvek - Python ismertető

- A programot Python nyelven kell megírni.
- A benyújtandó fájl neve: `feladat.py`
 - Egy Python nyelven írt, szöveges fájl (nem zip, rar, stb.)
 - Ez csak a feladatban kért dolgokat tartalmazza! Amennyiben saját inputtal teszteled a kódot lokálisan, úgy feltöltés előtt a tesztelő kódrészletet kommenteld ki!
- A megoldást Bíró2 webes felületén (<https://biro2.inf.u-szeged.hu>) keresztül kell benyújtani és a megoldást a Bíró fogja kiértékelni.
 - A Feladat beadása felületen a *Feltöltés* gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a Bíró a programot **Python 3.10.1** interpreterrel fogja futtatni, és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 2 másodpercnél és hiba nélkül fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- Ha 3 teszteset futási ideje túllépi a fenti időkorlátot, a tesztelés befejeződik, a pontszám az addig szerzett pontszám lesz.
- A riport.txt megtekinthető az alábbi módon:
 1. Az Eredmények megtekintése felületen a vizsgálandó próba új lapon való megnyitása
 2. A kapott url formátuma:
https://biro2.inf.u-szeged.hu/Hallg/IB370G/FELADAT_SZAMA/hXXXXXX/4/riport.txt
 3. Az url-ből visszatörölve a 4-esig (riport.txt törlése) megkaphatók a 4-es próbálkozás adatai
- A programot 25 alkalommal lehet benyújtani, a megadott határidőig.
- A munkád során figyelj arra, hogy pontosan kövesd a feladatban leírtakat, az elnevezéseket!
- A fájl elejére kommentbe írd be a neved, Neptun és h-s azonosítód az alábbi formában:

```
# Nev: Vezeteknev Keresztnev  
# Neptun: NEP4LF  
# h: h123456
```

Szkriptnyelvek 1. ZH - A csoport

1. feladat: Kétszer (15 pont)

Kriszta készített egy listát azokról az emberekről, akiket el szeretett volna hívni Halloweenezni, azonban néhány embernek a nevét véletlenül kétszer is felírta. Segíts neki megtalálni a duplikált neveket.

Írd meg a `ketszer` függvényt, ami egy sztringet kap paraméterül. A paraméterben kapott szöveget egy listaként kell értelmezni, amelyben a nevek vesszőkkel vannak elválasztva és minden vesszőt egy szóköz követ. A függvény feladata, hogy visszaadjon egy listát azokról a nevekről, amelyek **pontosan kétszer** szerepelnek az eredeti szövegben. Ha egyik név sem szerepel kétszer, téj vissza üres listával.

Példa:

```
Input: 'Albert, Peti, Tibor, Peti, Albert'  
Return: ['Albert', 'Peti']
```

```
Input: 'Peti, Tibor, Peti, David, Peti, Tibor'  
Return: ['Tibor']
```

```
Input: 'Albert, Peti, David'  
Return: []
```

2. feladat: Temeto (30 pont)

- Írj egy `Temeto` nevű osztályt, ami egy szellemek által kísértett temetőt reprezentál! Az osztálynak legyen 3 adattagja: `nev`, `_srok_szama` és egy `szelemek` nevű lista, ami a temetőt kísértő szellemeket tárolja! **(3 pont)**
- A konstruktor a temető nevét és a sírok számát kapja paraméterben (ilyen sorrendben), és ezek alapján inicializálja a `nev` és `_srok_szama` adattagokat! A `szelemek` adattag egy üres listával legyen inicializálva!
 - A `_srok_szama` paraméter értékét ne legyen kötelező megadni, alapértéke legyen 10! **(3 pont)**
- Készíts get és set property-t a `_srok_szama` adattaghoz, `srok_szama` néven! A getter adja vissza az adattag értékét, a setter ellenőrizze, hogy az új érték egy 10 vagy annál nagyobb szám legyen!
 - Ha a setterben a paraméter értéke megfelelő, akkor állítsd be az adattagot a paraméterben kapott értékre!
 - Hibás érték esetén a setter 10-re állítsa be az adattag értékét! **(6 pont)**
- Írd át az osztály konstruktorát úgy, hogy a `_srok_szama` értéke itt is ellenőrizve legyen! **(1 pont)**
- Definiáld felül az objektum szöveggé alakításáért felelős metódust úgy, hogy az a következő formátumú szöveggel térjen vissza: `A {nev} temetoben {_srok_szama} sir van, es {szelemek szama} szellem kíserti` (ahol a `szelemek szama` a `szelemek` lista hossza). Ha egy szellem se kíserti a temetőt, akkor az eredmény a `A {nev} temetoben {_srok_szama} sir van, es nem kíserti szellem` szöveg legyen. **(5 pont)**
- Definiáld felül az osztályban a `+=` operátort (`__iadd__` metódus), amely egy szellemet (azaz a szellem nevét, szöveg) kap paraméterül!
 - Ha a paraméter nem sztring típusú, dobj `ValueError` típusú kivételt, amit a `Nem szellem` szöveggel inicializálj! **(2 pont)**
 - Egyéb esetben a listában keresd meg, hogy szerepel-e már ilyen nevű szellem!
 - Ha nem szerepel, akkor egyszerűen add hozzá a szellemet a listához! **(3 pont)**
 - Ha szerepel, akkor a listában szereplő azonos nevű szellem nevének a végére fűzz egy `1`-est, az paraméterben kapott szellem nevéhez pedig egy `2`-est, és úgy add hozzá a listához. **(4 pont)**
- Definiáld felül az egyenlőséget vizsgáló operátort! Az operátor pontosan akkor adjon vissza igazat, ha az összehasonlítani kívánt temetők neve, illetve sírjainak száma megegyezik. Mással ne törődjön. **(3 pont)**

Példa használat:

```

print(ketszer("")) # []
print(ketszer("Albert")) # []
print(ketszer("Albert, Peti, Tibor")) # []
print(ketszer("Albert, Peti, Tibor, Peti, Albert")) # ["Peti"]
print(ketszer("Albert, Peti, Tibor, Peti, David, Peti")) # []

t1 = Temeto("Belvarosi", 400)
t2 = Temeto("Alsovarosi", 5)

t1.sirok_szama = 3
print(t1.sirok_szama) # 10

t1 += "Albert"
t1 += "Bela"
t1 += "Bela"
print(t1.szellemek) # ['Albert', 'Bela1', 'Bela2']
print(t1) # A Belvarosi temetoben 10 sir van, es 3 szellem kíserti

try:
    t1 += 666
except ValueError as ve:
    print(ve) # Nem szellem

print(t1 == t2) # False

```

Jó munkát!