

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://biro.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 3.12** 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://biro.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

Leltározás (10 pont)

Írj egy `leltar` nevű függvényt, ami két stringet vár paraméterül. Az első string a raktárban lévő tárgyakat tartalmazza vesszővel elválasztva, a második pedig egy tárgy neve, ami ha nincs megadva, akkor egy üres string. A függvény térjen vissza üres dictionaryvel, ha az első paraméter nem string, ha a második paraméter nem string, akkor a függvény kezelje úgy, mintha a default érték lenne megadva.

A függvény menjen végig a stringen és egy dictionaryba gyűjtse össze, hogy melyik tárgyból hány darab szerepelt a stringben és térjen vissza vele. A második paraméterben szereplő tárgyakat ne gyűjtse össze. A dictionary kulcsa a tárgy neve, értéke a tárgy darabszáma legyen.

Mivel babonások vagyunk, ha valamelyik tárgyból 13 van, akkor azokat helyben megsemmisítjük és ezért nem szerepelnek a dictionaryben.

Példák

Input: "alma,alma,alma","körte" Output: {'alma': 3}

Input: "kenyér,kifli,zsömle","kifli" Output: {'kenyér': 1, 'zsömle': 1}

Input: 31314 Output: {}

Input: "13,13,13,13,13,13,13,13,13,13,13,13,13,13" Output: {}

Kalandorok (30 pont)

Írj egy `Kalandor` osztály, ami, ahogy azt a neve is mutatja egy kalandort reprezentál. Az osztály adattagjai: `nev`, `életerő`, `arany`, `targyak` és `csapat`.

A konstruktor három paramétert várjon, egy string nevet, egy egész szám életerőt és egy tárgyak listát. Mindhárom adattagnak van default paramétere: a névé `John Doe`, az életerőé `20` és a tárgyaké pedig egy `üres lista`. A konstruktor emellett inicializálja a csapat adattagot egy üres listával és az arany adattagot 100-zal. A konstruktor végezzen típusellenőrzést is, és amelyik adattagnál rossz típust érzékel, annál használja a default paramétert.

Az életerő adattagnak legyen gettere és settere, ahol a setter biztosítja, hogy csak egész szám típusú nemnegatív számot lehet beállítani. Ha a szám kisebb mint 0, akkor legyen az életerő 0.

A kalandorokat lehessen kiírni stringként a következő formátumban: `{név}|{életerő}|{arany}|{targyak vesszővel elválasztva}|{csapattagok nevei vesszővel elválasztva}` (A kapcsos zárójeleket nem kell kiírni).

Legyen megvalósítva a `+=` operátor, amivel az operátor két oldalán lévő kalandor neve belekerül egymás csapat adattagjába formában. Ha egy kalandor már szerepel egy listában, akkor még egyszer ne kerüljön bele. Kalandor saját magát ne tehesse a saját csapatába.

Legyen egy `harc` metódus, ami egy másik kalandort vár paraméterül. A függvény ne csináljon semmit, ha valamelyik kalandor élete 0, vagy ha a paraméter nem kalandor. A harc kimenetelét a tárgyak száma dönti el. Akinek több tárgya van, az 2 életpontot veszít, akinek kevesebb az pedig 5-öt. Ha ugyanannyi tárgyuk van, akkor mindketten 3 pontot vesztenek.

Legyen egy `vásárlás` metódus, ami egy stringet vár, ami egy tárgy neve, a lenti táblázatból. A metódus ne csináljon semmit, ha a paraméter nem string, vagy ha a tárgy nem szerepel a listában. A vásárlás során a kalandor megveszi és eltárolja a tárgyat a tárgyak adattagjában, ha van rá elég aranya. Ha nincs, akkor a metódus dobjon egy `ArithmeticError`-t. Az árat természetesen vonjuk le a kalandor aranyából.

Tárgy	Ár
Kard	40
Pajzs	30

Tárgy	Ár
Íj	40
Bájtal	20
Sisak	50
Vért	60

Jó munkát!