

# Függvények

A repository-dban hozz létre egy „fuggvenyek” nevű mappát és ebbe dolgozz! A feladatokat az alábbiak alapján nevezd el: *feladat[sorszám].py*, például: *feladat1.py*.

1. Nyissuk meg az óra anyagai között található *feladat1.py* nevű fájlt. A fájl elején szerepel három függvény. Ezeket felhasználva számítsuk ki, majd írjuk ki a következőket a konzolra:
  - a. Mennyi 52 triplája?
  - b. Mennyi egy olyan téglalap területe, aminek az oldalai 5 és 14 egység hosszúak?
  - c. Mennyi egy olyan téglalap területe, aminek az oldalai 3.2 és 8.5 egység hosszúak?
  - d. A „szék” szót írassuk ki 20 'k' betűvel a végén!
2. Készítsünk programot, ami bekér a felhasználótól egy szöveget, majd kiírja annak hosszát a `len()` függvény meghívásával!
3. Készítsünk programot, ami bekér a felhasználótól egy valós számot, majd kiírja egész számra kerekítve! A feladat megoldásához használjuk a `round()` függvényt!
4. Írjunk függvényt, amely...
  - a. ...egy valós számot kap paraméterül, majd az abszolút értékével tér vissza! (Most ne használjuk az `abs()` függvényt!)
  - b. ...egy valós számot kap paraméterül, majd a köbével tér vissza!Írjunk programot, ami a fenti függvényeket felhasználva kiírja minden -10 és 10 közötti számhoz: a számot, annak abszolút értékét és köbét! Például a -3-hoz: -3; 3; -27.
5. Készítsünk függvényt, ami eldönti egy évről, hogy szökőév-e! (Szökőév minden negyedik, nem szökőév minden századik, mégis az minden 400-adik év. Ezért volt szökőév például 2000). Írjunk programot, ami a felhasználótól évszámokat kér és mindegyikhez kiírja, hogy szökőév-e!
6. Írjunk olyan függvényt, ami paraméterül kap egy sztringet és egy indexet. A függvény vágja ketté a sztringet az adott indexnél, és térjen vissza a második darabbal! Például: "python"; 3 → "on". Ha az adott index nagyobb, egyenlő, vagy egyel kisebb, mint a szöveg hossza, akkor térjünk vissza üres szöveggel! Például: "program"; 6 → "".
7. Készítsünk egyszerű menüvezérelt programot! A program tároljon el egy számot, melynek kezdőértéke  $x = 1$ . Ezt követően a program jelenítse meg a képernyőn  $x$  értékét, és az alább látható menüt. A megfelelő menüpont számának megadása után hajtsa végre  $x$ -en a kiválasztott műveletet, írja ki újból  $x$  új értékét és a menüt! A

menüből mindaddig lehessen újból választani, míg a kilépést nem választja a felhasználó!

```
print("0. Alapertek visszaallitasa (a = 1)",  
      "1. Hozzaad 1-et",  
      "2. Megforditja az elojelet",  
      "3. Szorozza 2-vel",  
      "9. Kilepes",  
      sep="\n")
```

Minden egyes tevékenységet (műveletet) egy pici függvény valósítson meg, amelynek bemenő paramétere az a változó tartalma, visszatérési értéke pedig a megváltozott szám! A program ezen függvények hívásával végezze el a feladatát!

8. Írjunk olyan függvényt, amely paraméterként sztringek listáját kapja. Meg kell vizsgálnia a listában található sztringeket, és megválaszolni ezt a kérdést: van-e olyan sztring, amelyik „a” betűvel kezdődik! Ennek is legyen logikai típusú a visszatérési értéke.

Ügyelj arra, hogy a függvény helyesen működjön abban az esetben is, ha üres sztring van a listában!

Példák:

- **["körte", "alma", "barack"]** – van „a” betűvel kezdődő szó.
- **["dinnye", "papaja", "", "zeller"]** – nincs „a” betűvel kezdődő szó.
- **[]** – nincs „a” betűvel kezdődő szó.

Teszteljük a függvényt ezekkel a példákkal, kiírva a listákat és a függvény visszatérési értékét is a főprogramban!

9. Írjunk egy függvényt, amely egy egész számokat tartalmazó listát kap bemenetként, és két listát ad vissza. Az egyik lista tartalmazza a bemenet páros, a másik a bemenet páratlan számait. Példakód a függvény használatára:

```
bemenet = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
kimenet = szetvalogat(bemenet)  
parosak = kimenet[0]  
paratlanok = kimenet[1]  
print("Páros számok:", parosak)  
print("Páratlan számok:", paratlanok)
```

A feladat megoldásához készítsünk egy segédfüggvényt, ami képes megállapítani egy adott számról, hogy páros-e, vagy sem!

10. Készítsünk függvényt, mely egy 1 és 99 közötti természetes számot képes szöveggé alakítani!

11. A python standard könyvtárában, megtalálhatóak a `min()` és a `max()` függvények. Ezeknek 2 számot adva paraméterül visszaadják a kisebbet (`min`) vagy a nagyobbat (`max`).
- Készítsünk egy függvényt ezen segédfüggvények felhasználásával, amely két oldalról korlátoz egy értéket: `korlatoz(szam, min, max)` adja vissza a számot, ha `min` és `max` közé esik, amúgy pedig `min`-t vagy `max`-ot attól függően, hogy merre haladta meg a tartományt.
- Segítség: a korlátozásnál figyelni kell arra, hogy az alulról limitáláshoz a `max()` függvényt kell használni, a felülről limitáláshoz pedig a `min()` függvényt.