



## 6. hét: függvények

 Czirkos Zoltán, Frey Balázs ·  2022.10.06.  
Függvények, paraméterek és visszatérési értékek. Kód refaktorálása, kódrészletek kiemelése különálló függvényekbe.

### Tartalom

- [Második Kis ZH](#)
- [main\(\)](#)
- [Függvények – alapok](#)
- [Szökőév](#)
- [Az év napja](#)
- [„a” betűvel kezdődő sztring](#)
- [Betű cseréje](#)
- [Átlagnál kisebbek](#)
- [Menüvezérelt program](#)
- [Madárnyelv](#)
- [Alapértelmezett paraméterek](#)

### 1. Második Kis ZH

A hétfői alkalmakon lesz a második kis ZH.

[Jelentkeztél NZH-ra?](#) Nézted a [nagy házik listáját](#)? Ne feledd, ezek határidős feladatokat!

### 2. main()

Az alábbi feladatok programjai a legtöbb helyen nem tartalmaznak `main()` függvényt. Ahol nem kéri külön a feladat, ott is add hozzá! A következő héttől kezdve minden kódban, minden ZH-ban kérjük és pontozzuk a `main()` függvény használatát.

### 3. Függvények – alapok

Írj [függvényeket](#), amelyek valós számot vesznek át, és visszatérnek a:

- `kob()` – harmadik hatványával,
- `abszolot()` – abszolút értékével (van beépített `abs()` függvény, de most ne használd)!

Írj programot, amelyik a = -1-től +1-ig, tizedenként lépve, kiírja egymás mellé a, a<sup>3</sup>, |a| és sin(a) értékét, mindig *négy tizedesjegy* pontossággal!

### 4. Szökőév

Készíts függvényt, amelyik adott évszámról eldönti, hogy az szökőév-e. Szökőév minden negyedik, nem szökőév minden századik, mégis az minden 400-adik. (2000-ben ezért volt szökőév.) A függvény visszatérési értéke legyen logikai típusú!

Írj programot, amelyik a felhasználótól évszámokat kér, és mindegyikre kiírja, hogy szökőév-e! Például:

```
? 2005
Nem szökőév.

? 2000
Szökőév.

? 1980
Szökőév.

? 1900
Nem szökőév.
```

► [Megoldás](#)

### 5. Az év napja

Írj programot, amely megkérdezi a felhasználótól egy hónap számát (pl. 3 = március), és utána kiírja, hány napos az a hónap!

► [Megoldás](#)

Írj programot, amelyik egy adott dátumról (év, hónap, nap) kiírja, hogy az év hányadik napja! Az év paraméterre a szökőévek miatt van szükség. Használd fel az előző feladatban megírt függvényed!

Végül tedd be az év napját kiszámító algoritmusod is egy függvénybe! Vagyis az `evnapja(ev, honap, nap)` függvényhívás visszatérési értékében adja meg a nap sorszámát 1 és 365 (szökőév esetén 366) között.

► [Megoldás](#)

### 6. „a” betűvel kezdődő sztring

Egy olyan függvényt kell írnod, 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.
- `["dínnye", "papaja", "", "zeller"]` – nincs „a” betűvel kezdődő szó.
- `[]` – nincs „a” betűvel kezdődő szó.

Teszteld a függvényed ezekkel a példákkal, kiírva a listákat és a függvény visszatérési értékét is a főprogramban!

### 7. Betű cseréje

A programodban egy szót fogsz kapni, továbbá egy pozíciót (indexet), és egy betűt, amire az adott helyen a karaktert cserélni kell. Például a „papa” szó 1-es indexű betűjét „i”-re cserélve a „pipa” szót kapjuk.

Írj függvényt, aminek paraméterei a szó, a pozíció és a betű! Végezze el a függvény a cserét! Hívd meg a főprogramból a függvényt (például a fenti adatokkal), és írasd ki a főprogramból a megváltoztatott sztringet!

► [Tipp – ha elakadtál](#)

► [Tipp – ha nincs ötleted](#)

Egészsds ki úgy a függvényed, hogy figyelembe vegye az érvénytelen indexeket, és azokra dobjon kivételt! Tehát pl. „papa→pipa”, ha az 1-es indexű karaktert kell cserélni, de dobj kivételt „Érvénytelen pozíció: X” üzenettel, ha mondjuk a 9-es vagy a -5-ös karakter cseréjét kérné a hívó! Az X helyére mindig íródjon a helytelen index. Kapd is el ezt a kivételt a főprogramban!

► [Megoldás](#)

### 8. Átlagnál kisebbek

Az alábbi program a megadott számsorból az átlagnál kisebbeket gyűjti egy listába:

```
szamok = [24, 31, 22, 43, 10, 84, 38, 44, 84, 56, 67, 51, 56, 84, 31, 65, 69, 83, 39]
szum = 0
db = 0
for x in szamok:
    szum += x
    db += 1
atlag = szum / db
szurt = []
for x in szamok:
    if x < atlag:
        szurt.append(x)
print(szurt)
```

Értsd meg a működését, utána pedig darabold fel függvényekre! Válaszd külön az átlagszámítás és a szűrés műveletét! Ahol lehet, használd a Python már tanult függvényeit! Például a lista hosszát nem úgy szoktuk meghatározni, hogy megszámoljuk az elemeit egyesével, de más részfeladatokra is találsz beépített függvényeket.

► [Megoldás](#)

### 9. Menüvezérelt program

Készíts egyszerű menüvezérelt programot! A program tároljon el egy számot, melynek kezdőértéke a = 1. Ezt követően a program jelenítse meg a képernyőn a értékét, és az alább látható menüt. A megfelelő menüpont számának megadása után hajtsa végre a-n a kiválasztott műveletet, írja ki újból a 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 kicsi függvény valósítson meg, amelynek bemenő paramétere az a változó tartalma, visszatérési értéke pedig a megváltozott szám! A `main()` ezen függvények hívásával végezze el a feladatát!

**Miért olyan lényeges ez a feladat?**

Figyeld meg a kapott főprogramot! Ez irányítja a többi függvény működését: meghívja az egyes részfeladatokhoz tartozó alprogramokat, amelyek dolgukat végezve visszatérnek, újra a főprogram kezébe adva az irányítást. A főprogram és az alprogramok a paramétereken és a visszatérési értékeken keresztül kommunikálnak.

### 10. Madárnyelv

Adott az alábbi program, amely madárnyelven (mavadárnyelveven) írja ki a beírt szöveget.

```
szoveg = input()
for c in szoveg:
    if c == 'a' or c == 'e' or c == 'i' or c == 'o' or c == 'u':
        print(c, 'v', c, sep="", end="")
    else:
        print(c, end="")
```

Írj függvényt, amelyik megmondja egy betűről, hogy magánhangzó-e! Alakítsd át úgy a programot, hogy a megírt függvényt használd a `main()`-ben! A magánhangzó-e függvény megírható egyetlen egy sorból is (a fejlécén túl, természetesen).

Végezz el egy további átalakítást is: a madárnyelvesítés is legyen függvény! Vagyis írs egy olyan függvényt, amely paraméterként egy sztringet kap, visszatérési értéke is egy sztring: például `madarnyelv("alma")` == `"avalnava"`. (A sztring összeállításához a `+=` operátort tudod használni.) Használd fel ezt a függvényt a `main()`-ben az eredeti feladat megoldására!

Hogyan lehetne megoldani azt, hogy a nagybetűvel kezdődő szavakat is helyesen kezelje a program? Pl. az „Alma” szóra azt kell kiírnia, hogy „Aválmava”. Ehhez fel kell tudnia ismerni a nagybetűvel írt magánhangzókat is. Ugyanakkor az ismételt magánhangzó már a szó közepén van, tehát kisbetűsíteni kell.

► [Megoldás](#)

### 11. Alapértelmezett paraméterek

A `range()` függvény három különféle paraméterezéssel hívható:

- `range(min, max)`: például a `range(1, 5)`-ből az 1, 2, 3, 4 számok potyognak ki.
- `range(min, max, step)`: adott lépésközzel, például `range(0, 5, 2)` paraméterek esetén 0, 2, 4.
- `range(max)`: egy paraméter esetén a kezdeti érték nulla lesz. Például `range(3)` esetén 0, 1, 2.

Írj függvényt `samtani()` néven, amely egy számtani sorozatot ír ki a képernyőre! A paraméterezése legyen pontosan ugyanaz, mint a `range()` függvényé, vagyis pont ugyanazokat a számokat írj ki.

A függvény belsejében ne használd a `range()`-et, és ne is építs listát! A feladat lényege az, hogy lásd, hogyan kell eltérő paraméterszámmal hívható függvényt listni.

► [Tipp](#)