Mentor Klub

Python tanfolyam

6. alkalom

Kód struktúrálása

Mentor Klub

- A programkódot a könnyebb olvashatóság és egyszerűbb karbantarthatóság érdekében struktúrálni szoktuk.
- Ha egy forrásfájl (.py fájl) 400 sornál hosszabb, akkor sokkal nehezebb karbantartani.
- Ha egy egybefüggő algoritmus 15-20 sornál hosszabb, akkor sokkal nehezebb karbantartani.
- A forrásfájl tartalmát függvényekre bontjuk.
- A forrásfájlokat pedig feldaraboljuk és modulokra bontjuk

Függvény – Mire használjuk?

Mentor Klub

- Nagyon “pongyola” megfogalmazás, mert nem matematikusnak készülünk:
  - A függvény egy olyan leképezés, ami valamennyi bemeneti paraméterhez egy kimeneti értéket rendel.
  - Pl.: $f(x) = x^2$ vagy $T_{\text{téglalap}}(a,b) = a*b$
- A függvények segitségével a forráskódunkat “emészthető” méretű darabkákra vághatjuk.
- Egy algoritmust vagy annak egy részét általánositjuk, újrafelhasználhatóvá tesszük.
- Egy hosszú kódrészlet egy jól behatárolt részét kiemeljük.
- Ökölszabály:
  - Egy függvény lehetőleg egy valamit csináljon.

Függvény definiálása

```
def fuggveny_nev(valtozo_1,…, valtozo_n):
    utasitas1
    utasitas2
    return visszateresi_ertek

x = fuggveny_nev(1,2, "c")

#Az x változó értéke a függvény által visszaadott visszatértési érték lesz.
```

Függvény felépítése

- Függvény neve:
  - Egy egyedi név, ami az adott modulon belül azonosítja a függvényt.
- Szignatúra:
  - A bemeneti paraméterek (argumentumok) száma.
  - Más programozási nyelvekben a bemeneti paraméterek típusa, sorrendje is fontos.
- Törzs:
  - Az a programrész, ami a függvény hívásakor végrehajtódik. A beneti paramétereket változókként használhatjuk.
  - Lehetnek benne függvényhívások.
- Visszatérési érték:
  - Ezt számolja ki a függvény. Nem kötelező elem. Olyan függvény is létezik, ami nem tér vissza semmivel.

Gyakorló feladat

- Készítsük el a faktoriális számitás algoritmusát iteratív módszerrel.
- A bemeneti paraméter egy természetes szám (>= 0)
- A kimeneti paraméter egy természetes szám, ami a bemeneti paraméternél kisebb egyenlő nemnegatív számok szorzata.

- Példa:
  - faktorialis(4) = 1*2*3*4

Konténer típusok

Mentor Klub

- A Pythonban vannak olyan adattípusok, melyek nem egyetlen értéket tárolnak, hanem tetszőleges mennyiségűt.
- Két **fontos** tipus létezik:
  - Tuple *(ejtsd: tápöl):*
    - Jelölése: (1,2,3) #Kerek zárójelek között vesszővel felsorolt elemek.
    - Létrehozás után nem lehet megváltoztatni
  - Lista:
    - Jelölése: [1,2,3] #Szögletes zárójelek között felsorolt elemek
    - Tartalma módosítható
- Több másfajta konténer típus is létezik, de azokról majd máskor.

Lista

- Létrehozása:
  - `x = [1,2,3]`
  - `y = list()`
  - `z = []`
- Elemek hozzáadása:
  - `x.append(4) #az eredmény [1,2,3,4]`
  - `y.append([23,24,25]) # az eredmény [[23,24,25]]`
  - `z.extend([5,6,7,8]) # az eredmény [5,6,7,8]`

Elem lekérdezése - Indexelés

- A legtöbb programozási nyelvben a listákat (tömböket) lehet indexelni, azaz tetszőleges eleméhez azonnal hozzáférni.
- Például:
  - x = [1,2,3,4] esetén x[2] = 3, x[0] = 1 és x[4] az hiba.
  - Ha annyiadik elemre hivatkozunk, ami már nincs a listában, akkor azt mondjuk, hogy túlindexeltünk.
  - Az indexek 0-tól indulnak, ezért a legutolsó index mindig a lista elemszáma minusz 1
    - `max_index = len(x) -1 # Ez üres lista esetén -1 lenne, ezért azt mindig meg kell nézzük, hogy egy lista üres-e`
    - `len(x) == 0 # Üres a lista?`

Gyakorló- és házi feladatok

Mentor Klub

- Készitsük el az alapvető algoritmusokat függvényként. A bemeneti paraméter minden esetben egy lista legyen.
- Egy számokat tartalmazó listából keressük meg a legnagyobb hárommal osztható számot.
- Egy számokat tartalmazó listából keressük meg a legkisebb 5-tel osztható számot és adjuk vissza a lista-beli indexét
- Egy szavakat tartalmazó listából válasszuk ki azt a szót, amiben a legtöbb, szintén paraméterként átadott betű szerepel.
  - Például a ['cica', 'kutya’, 'kakas'] szavakat tartalmazó listában mely szóban van a legtöbb ‘a’ betű?