

# Változók

Készítette: Vastag Attila

2020

Amikor programot írunk, akkor szükség lehet tárolókra, ahová az adatainkat ideiglenesen eltároljuk. Ezeket a tárolókat változóknak nevezzük.

A változók a memória egy (vagy több) cellájára hivatkozó leírók. Egy változót a következő módon hozhatunk létre *Python* nyelven:

*változónév;*

A változónév első karaktere csak betű vagy alulvonás jel (  ) **lehet, a többi karakter szám is.** Lehetőleg kerüljük az ékezetes karakterek használatát.

Konvenció szerint a változónevek kisbetűvel kezdődnek. Amennyiben a változónév több szóból áll, akkor célszerű azokat a szóhatárnál nagybetűvel “elválasztani” (pl. pirosAlma, vanSapkaRajta, stb).

# Python típusok

Két fajtáját különböztetném meg a Python beépített típusainak. Ezek a primitívek és az objektumok. A primitív típusok egy értéket jelölnek. Ezek például a logikai értékek és a számok. Az objektumok összetett adattípusok, amik általában más, primitív típusokat kombinálnak.

## Primitív típusok

| Név   | Típus | Leírás   |
|---|-------|--|
| <b>Integer</b> , vagy egész szám            | int   | Ezek tetszőleges nagyságú pozitív vagy negatív értékek, tizedesjegyek nélkül.  |
| <b>Lebegőpontos szám</b> , vagy tizedestört | float | Az ilyen típusú értékek néhány tizedesjegyig pontosak, és nem tudnak tetszőleges nagy egészrészt tárolni.                              |
| <b>Boolean</b> , vagy logikai érték         | bool  | Ennek a típusnak pontosan két érték felel meg, a logikai igaz ( <b>True</b> ) és a hamis ( <b>False</b> ).                             |
| <b>None</b>                                 | None  | A <i>semmi</i> . Néha bizony egy művelet végeredménye a nagy büdös semmi. Ennek a típusnak egy érték felel meg, ami stílusosan a None. |

# Python típusok

## Objektumok típusok

| Név                                       | Típus | Leírás  |
|---|-------|---|
| <b>String</b> , vagy karakterlánc         | str   | Mindenféle szöveges érték ilyen típusú.   |
| <b>Lista</b>                              | list  | Több különböző típusú adatot lehet benne egymás után felsorolva tárolni.  |
| <b>Dictionary</b> , vagy asszociatív tömb | dict  | Ebben is több különböző típusú adatot lehet tárolni, de az adatokat saját egyedi azonosító kulccsal érhetjük el   |
| <b>Tuples</b>                             | tup   | <i>A tuple</i> egy megváltoztathatatlan lista.  |
| <b>Sets</b>                               | set   | <i>A halmaz</i> egyedi értékek rendezetlen „kupaca”. Egy halmaz tetszőleges megváltoztathatatlan adattípusú értékeket tartalmazhat. Ha van két halmazod, akkor végrehajthatsz rajtuk általános halmazműveleteket, mint az unió, metszet és különbség. |

## Deklaráció és definíció

n: **int** = 7

**n-nek a 7-et adjuk értékül**

msg: **str** = "Mi újság ?"

**A "Mi újság ?" értéket adjuk msg-nek**

pi: **float** = 3.14159

**pi név változóhoz hozzárendeljük a 3.14159 értékét**

sikeres: **bool** = True

**sikeres név változóhoz hozzárendeljük az igaz értékét**

A Pythonban nem szükséges a változók használata előtt a típusuk definiálása érdekében speciális programsorokat írni. Elég ha hozzárendelünk egy értéket egy változónévhez. A Python a változót automatikusan azzal a típussal hozza létre, ami a legjobban megfelel az értéknek. Az előző gyakorlatban például az **n**, **msg**, **pi** és **sikeres** változók mindegyikét automatikusan hozná létre különböző típusokkal (rendre «egész szám», «karakterlánc (string)», «lebegőpontos szám», «logikai» típusokkal).

# Változó típusának ellenőrzése

**type()** függvény segítségével

```
n: int = 7  
msg: str = "Mi újság ?"  
pi: float = 3.14159  
sikeres: bool = True
```

---

|                      |                              |
|----------------------|------------------------------|
| <b>type(n)</b>       | <b>&lt;class 'int'&gt;</b>   |
| <b>type(msg)</b>     | <b>&lt;class 'str'&gt;</b>   |
| <b>type(pi)</b>      | <b>&lt;class 'float'&gt;</b> |
| <b>type(sikeres)</b> | <b>&lt;class 'bool'&gt;</b>  |

---