UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

Szkriptnyelvek - 1. gyakorlat

Cservenák Bence

Közérdekű információk

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Gyakorlatvezető: Cservenák Bence
  - e-mail: `Cservenak.Bence@stud.u-szeged.hu`
    - ha nem válaszolok, csekkold, hogy jól írtad-e
  - CooSpace üzenetek are also welcome
- Tematika, követelmények: CooSpace
- Közérdekű infók, slide-ok, órai kódok: CooSpace
- Gyakorlathoz tartozó szkriptek: `/pub/Szkriptnyelvek/gyakorlat`

Követelmények

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- A gyakorlat látogatása kötelező, kettőnél több hiányzás esetén a kurzus nem teljesített
  - kivéve: első 2 hét (tárgyfelvételi időszak)
- Tematika: Python és JavaScript
- Számonkérések
  - Python ZH (50 pont)
    - október 7. hét, saját gyakorlat helyén és időpontjában
    - min. teljesítendő 25 pont
  - JavaScript ZH (50 pont)
    - november 25. hét, saját gyakorlat helyén és időpontjában
    - min. teljesítendő 25 pont

Követelmények

- A gyakorlat látogatása kötelező, kettőnél több hiányzás esetén a kurzus nem teljesített
  - kivéve: első 2 hét (tárgyfelvételi időszak)
- Tematika: Python és JavaScript
- Számonkérések
  - Python ZH (50 pont)
    - október 7. hét, saját gyakorlat helyén és időpontjában
    - min. teljesítendő 25 pont
  - JavaScript ZH (50 pont)
    - november 25. hét, saját gyakorlat helyén és időpontjában
    - min. teljesítendő 25 pont

BÍRÓ
Feladatbeadó oldal

Követelmények

- Órai munkára legfeljebb 10 plusz pont szerezhető
  - elégséges érdemjegy elérése után…
- Javító ZH a szorgalmi időszak utolsó hetén az elégséges érdemjegyért
- Ponthatárok:

89 - 100 pont	Jeles (5)
76 - 88 pont	Jó (4)
63 - 75 pont	Közepes (3)
50 - 62 pont	Elégséges (2)
0 - 49 pont	Elégtelen (1)

Bevezetés - Fordított nyelvek, szkriptnyelvek

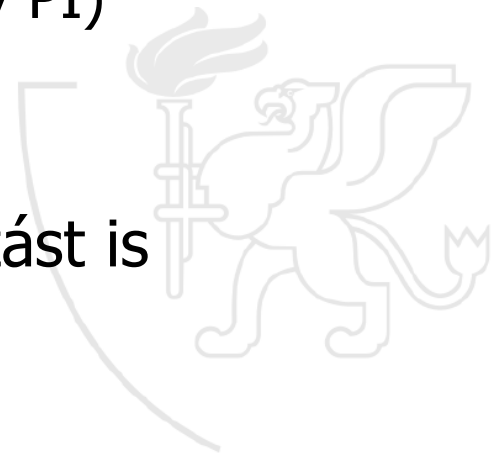
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Fordított nyelvek
  - a szöveges fájlban lévő kód gépi kódra fordul le
  - gyors, viszont platformfüggő
  - pl. C, C++, Java (részben), …
- Szkriptnyelvek (interpretált nyelvek)
  - nincs fordítás, nincs gépi kód
  - a szöveges fájl értelmezéséhez kell egy értelmező (interpreter)
  - platformfüggetlen, viszont lassabb
  - pl. Python, JavaScript, PHP, BASH, AWK, Java (részben), Matlab, …

Bevezetés - A Python főbb jellemzői

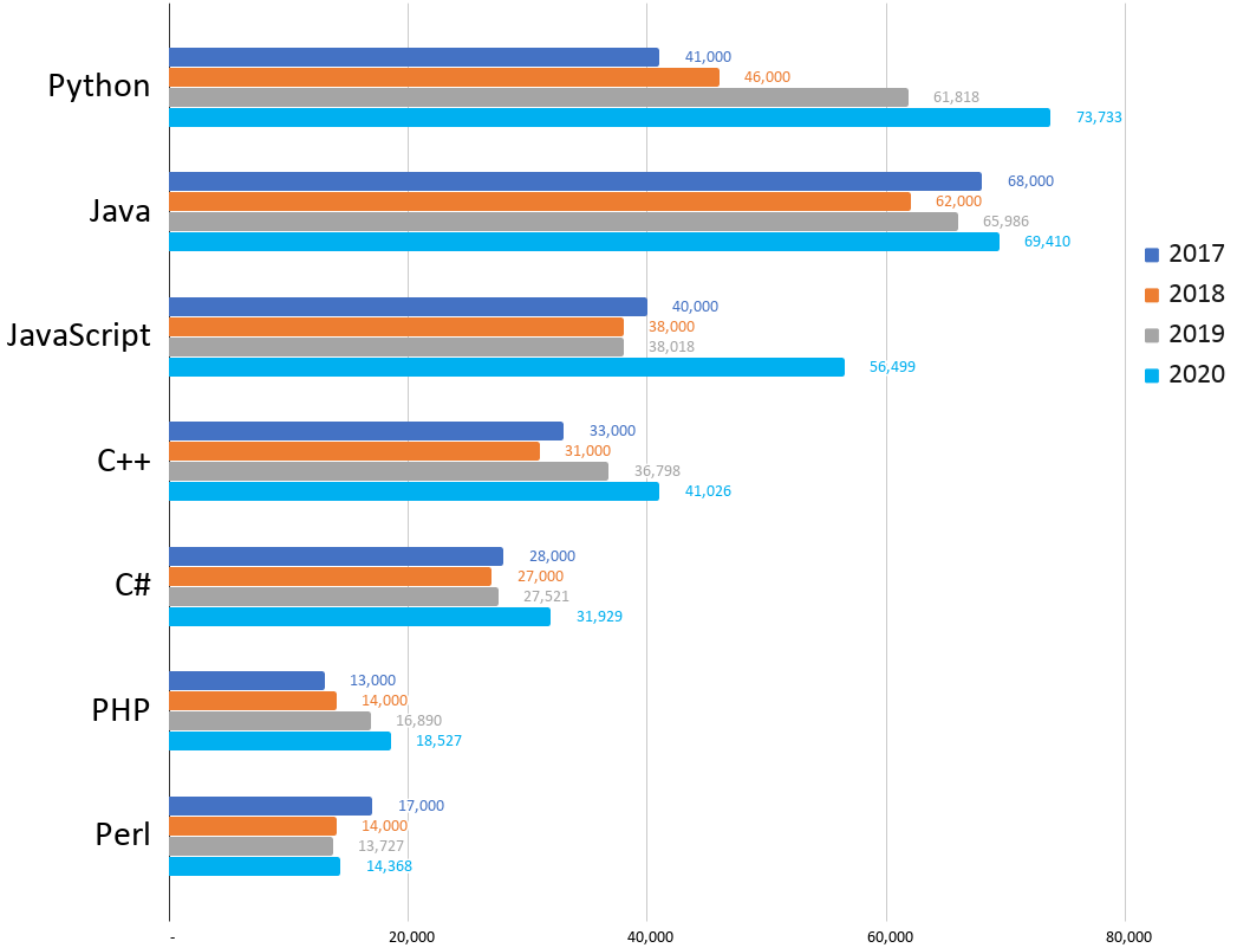
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Open-source
- Operációs rendszer független
- Széleskörű alkalmazási lehetőségek
  - parancssori, asztali alkalmazások
  - webes alkalmazások (backend)
  - beágyazott rendszerekre írt alkalmazások (pl. Raspberry PI)
- Intuitív, könnyen tanulható szintaxis
- Objektumorientált, de támogatja a funkcionális programozást is
- Részletes, "kezdőbarát" hibaüzenetek



Bevezetés - Egy kis érdekesség

- Programozási nyelvek keresettsége a munkaerő-piacon (`indeed.com`):



A Python telepítése

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Windows, Mac OS X letöltési link: https://www.python.org/downloads/
  - válasszuk a Python 3-at
- Linux: `sudo apt-get update && sudo apt-get install python3.7`
- Telepítés után adjuk hozzá a Pythont a **`PATH`** környezeti változóhoz
- Tesztelés parancssorban:

```
C:\Users\cservZ> python --version
Python 3.7.4
```

A parancssori Python értelmező

- A parancssorban a **`python`** paranccsal indíthatjuk el a Python értelmezőt

Megjegyzés

Mivel a Kabinetben az alapértelmezett Python értelmező a Python 2, ezért a gyakorlaton mindig a **`python3`** parancsot fogjuk használni.

- Innentől kezdve a beírt parancsokat folyamatosan értelmezi kilépésig
- A kilépés EOF karakter segítségével történik
  - Windows alatt: `Ctrl+Z`
  - UNIX alatt: `Ctrl+D`
  - használhatók még az `exit()` és `quit()` parancsok is

Kommentek

- Pythonban is lehetőségünk van a kódba kommenteket (megjegyzéseket) írni

```
# ez egy egysoros komment

"""
   ez egy több
   soros komment
"""
```


Kiíratás

- **`print`**`(…)`: kiírja a paraméterül kapott kifejezést
  - tetszőleges számú paraméterrel hívható (vesszővel elválasztva)
- A **`python`** (**`python3`**) paranccsal indítsuk el a parancssori Python értelmezőt, majd gépeljük be az alábbiakat:

```
>>> print("Hello World!")
Hello World!

>>> print("Never", "gonna", "give", "you", "up")
Never gonna give you up
```

Alapvető adattípusok

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- `bool`: logikai adattípus
  - **`True`** vagy **`False`** értéket vehet fel (kis- és nagybetűérzékenyek!)
- `int`: egész szám
- `float`: lebegőpontos (valós) szám
- `str`: szöveg (string)
  - megadás: aposztrófok (`'…'`) vagy idézőjelek (`"…"`) között
  - összefűzés: `+` operátorral

```
>>> print('Hello ' + "Python")
Hello Python
```

  - egy kis érdekesség:

```
>>> print("NA" * 10 + " BATMAN")
NANANANANANANANANANA BATMAN
```

Változók

- Létrehozás, értékadás: `valtozonev = ertek`
  - a változónév csak betűket, számokat és alulvonás karaktert (_) tartalmazhat, és mindig betűvel vagy alulvonással kell kezdődnie
  - a változónév tartalmazhat ékezeteket, azonban ezt célszerű kerülni
  - a változónév érzékeny a kis- és nagybetűkre

```
>>> nev = "Béla"
>>> Nev = "Sanyi"     # nem ugyanaz, mint a kisbetűs nev

>>> print(nev)
Béla

>>> print(Nev)
Sanyi
```

Változók

- A Python a dinamikusan típusos nyelvek közé tartozik
  - a változók típusa futásidőben derül ki
  - ugyanaz a változó eltérő típusú értékeket is tárolhat

```
>>> val = 42          # val típusa: int
>>> print(val)
42

>>> val = "Béla"      # val típusa: str
>>> print(val)
Béla
```

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

Fontosabb operátorok

- Aritmetikai operátorok: `+` (összeadás), `-` (kivonás), `*` (szorzás), `/` (valós osztás), `//` (egész osztás), `%` (maradékos osztás), `**` (hatványozás)

```
>>> 5 / 2          # valós osztás
2.5

>>> 5 // 2         # egész osztás
2

>>> 5 ** 2         # hatványozás
25
```

- Hozzárendelő operátorok: `=`, `+=`, `-=`, `*=`, `/=` (ugyanúgy, mint C-ben)
- Összehasonlító operátorok: `==`, `!=`, `<`, `<=`, `>`, `>=` (ugyanúgy, mint C-ben)
- Logikai operátorok: **`and`** (logikai ÉS), **`or`** (logikai VAGY), **`not`** (logikai NEM)

Fontosabb operátorok

- Pythonban a `++` és `--` operátorok **nem** szerepelnek!
  - helyettük használjuk a `+= 1`, illetve `-= 1` utasításokat az érték 1-gyel történő növeléséhez/csökkentéséhez

```
>>> eletkor = 20
>>> eletkor += 1
>>> print(eletkor)
21

>>> eletkor++                          # nope :(
File "<stdin>", line 1
    eletkor++
            ^
SyntaxError: invalid syntax
```


Típuskonverzió

- Szövegből egész számot az `int()` segítségével készíthetünk

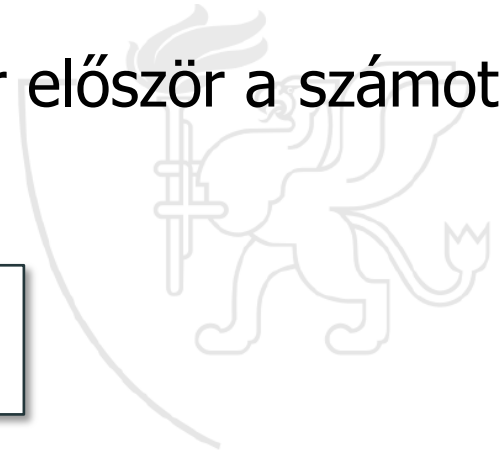
```
>>> int("42")
42
```

- Lebegőpontos számnál ugyanígy járunk el, csak ekkor a `float()` használatos

```
>>> float("3.14")
3.14
```

- Ha egy számot össze szeretnénk fűzni egy stringgel, akkor először a számot stringgé kell konvertálnunk az `str()` használatával

```
>>> str(3.14)
'3.14'
```



Beolvasás a konzolról

- **`input`**`(prompt)`: beolvasás a standard inputról
  - `prompt` értékeként megadható egy szöveg, ami az input bekérésekor jelenik meg
  - mindig stringet olvas be, szükség esetén konvertálni kell

```
>>> nev = input("Hogy hívnak? ")
Hogy hívnak? Béla

>>> print("Hello", nev)
Hello Béla
```


Beolvasás a konzolról

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Olvass be két egész számot a standard inputról, és írasd ki az összegüket!

```
>>> a = input("Első szám: ")          # beolvasás
Első szám: 5
>>> b = input("Második szám: ")
Második szám: 2

>>> a = int(a)                        # konvertálás egész számmá
>>> b = int(b)
>>> sum = a + b

>>> print("Az összeg: " + str(sum)) # konvertálás stringgé
Az összeg: 7
```