UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

Szkriptnyelvek - 2. gyakorlat

Cservenák Bence

Python szkriptek és futtatásuk

- A Python kódunkat ezentúl fájlokba fogjuk kiszervezni (szkriptek)
- A Python szkriptek kiterjesztése: **`.py`**
- Hozzunk létre egy `test.py` állományt, majd írjuk bele a következő kódot:

```
print("Hello World!")
print("Ez az első Python szkriptem")
```

- Parancssorban navigáljunk el abba a mappába, ahol a fájlt létrehoztuk, majd a **`python test.py`** parancs kiadásával futtassuk a megírt programot
- Az output:

```
Hello World!
Ez az első Python szkriptem
```

A Python blokkszintaxisa

- Pythonban az indentálást (beljebb igazítást) használjuk blokkok megadására
  - C-ben és Javában { … } között adtuk meg a blokkokat
  - az indentálásnak konzisztensnek kell lennie (pl. mindenhol 4 darab szóköz)

```
lottoszamok = [56, 42, 81, 5, 12]
paros_szamok = 0

for szam in lottoszamok:
    if szam % 2 == 0:          # 1. szint: a for-hoz tartozó utasítások
        paros_szamok += 1      # 2. szint: az if-hez tartozó utasítások
        print(szam)

print("Összesen", paros_szamok, "darab páros lottószám volt.")
```

Vezérlési szerkezetek

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Szelekciós vezérlés
  - `if`
  - `else`
  - `elif` (ez a C-ből ismerős `else if` Pythonos megfelelője)
- Ismétléses vezérlés (Ciklusok)
  - `while`
  - `for`

Megjegyzés

Pythonban **nincs** `switch` és `do... while` szerkezet.

- **Példa:** Egy program, amely beolvas egy egész számot a standard inputról, és kiírja, hogy az pozitív, nulla vagy negatív

```
szam = int(input("Adj meg egy számot: "))

if szam < 0:                                    # "ha" ág
   print("A beírt szám negatív")
   print("Abszolútértéke: " + str(abs(szam)))
elif szam == 0:                                 # "egyébként ha" ág
   print("A beírt szám 0")
else:                                           # "egyébként" ág
   print("A beírt szám pozitív")
```

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Fontosabb operátorok:

Operátor	Leírás	Példa
==	Igaz, ha a két operandus értéke megegyezik	`a == b`
!=	Igaz, ha a két operandus értéke nem egyezik meg	`a != b`
<	Igaz, ha az első operandus értéke kisebb, mint a másodiké	`a < b`
<=	Igaz, ha az első operandus értéke kisebb vagy egyenlő, mint a másodiké	`a <= b`
>	Igaz, ha az első operandus értéke nagyobb, mint a másodiké	`a > b`
>=	Igaz, ha az első operandus értéke nagyobb vagy egyenlő, mint a másodiké	`a >= b`
`and`	Igaz, ha az első és második operandus értéke is igaz	`a and b`
`or`	Igaz, ha az első vagy második operandus értéke igaz	`a or b`
`not`	Igaz, ha az operandus értéke hamis	`not a`

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Néhány példa összetettebb feltételekre:

```
if eletkor < 0 or eletkor > 120:
   print("Valós életkort adj meg!")
```

```
if eletkor >= 18 and nem == "férfi":
   print("Nagykorú férfi vagy")
```

```
if nev == "Márk" and not reactok_szama < 10:
   print("Itt van Márk!")
```

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- **Példa:** Írassuk ki az egész számokat 1-től 10-ig `while` utasítás segítségével!

```
i = 1

while i <= 10:
      print(i)
      i += 1        # emlékeztető: Pythonban nincs i++
```

Ismétléses vezérlés

while

- A **break** és **continue** utasítások Pythonban is használhatók
  - **break**: kilép a ciklusból
  - **continue**: a következő iterációval folytatja

```
count = 0

while True:                 # végtelen ciklust indít
    count += 1

    if count > 10:          # megállási feltétel
        break

    if count == 4:          # a 4-es értéket átugorjuk
        continue

    print(count, "Python for the win")
```

```
1 Python for the win
2 Python for the win
3 Python for the win
5 Python for the win
6 Python for the win
7 Python for the win
8 Python for the win
9 Python for the win
10 Python for the win
```

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Pythonban úgynevezett listaszerű **`for`**-ciklus van
  - elemsorozat bejárására használjuk, a ciklusváltozó rendre felveszi a bejárni kívánt elemsorozat minden értékét
  - `range([start], stop, [step])`: számokból álló elemsorozat generálása
    - "hagyományos" `for`-ciklus szimulálása
    - kiírja a számokat `start`-tól `end`-ig (az `end` már nincs benne), `step` lépésközzel
    - `step` értékeként megadható negatív szám is (visszafelé lépkedünk)

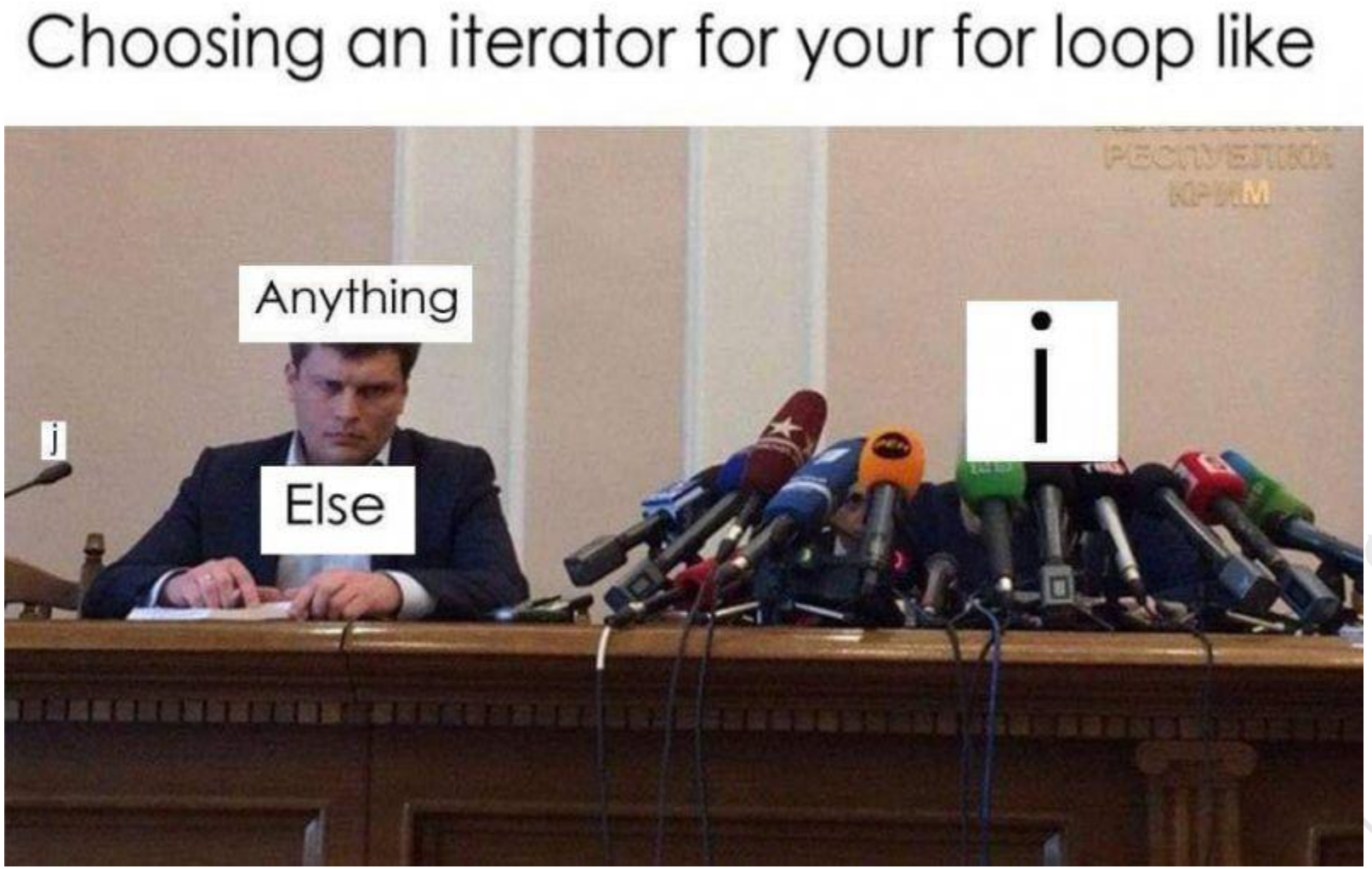
```
# egész számok kiíratása 1-től 10-ig

for i in range(1, 10):
    print(i)
```

```
# 1-100 közötti egész számok kiíratása, 5-ös lépésközzel

for i in range(1, 100, 5):
    print(i)
```

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering



UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Szövegek karaktereinek bejárása `for`-ciklussal:

```
for betu in "almáspite":
    print(betu)
```

```
a
l
m
á
s
p
i
t
e
```

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
SZEGEDI TUDOMÁNYEGYETEM
Department of Software Engineering

- Lista elemeinek bejárása `for`-ciklussal
  - a listákról a következő gyakorlaton fogunk tanulni

```
for szam in [12, 23, 34, 45]:
    print(szam)
```

```
12
23
34
45
```

- Természetesen itt is használhatók a **break** és **continue** utasítások

```
for i in range(1, 100):
    print(i)
    if i == 42:
        print("Megtaláltuk az élet értelmét")
        break
```

```
print("A legjobb pizzafeltétek:")

for feltet in ["sajt", "bacon", "ananász", "pepperoni"]:
    if feltet == "ananász":
        continue
    print(feltet)
```