Takács Edit
edit.takacs@gmail.com
GitHub:
https://github.com/gtakacse/CS_class_scripts

Emberi Erőforrások Minisztériuma

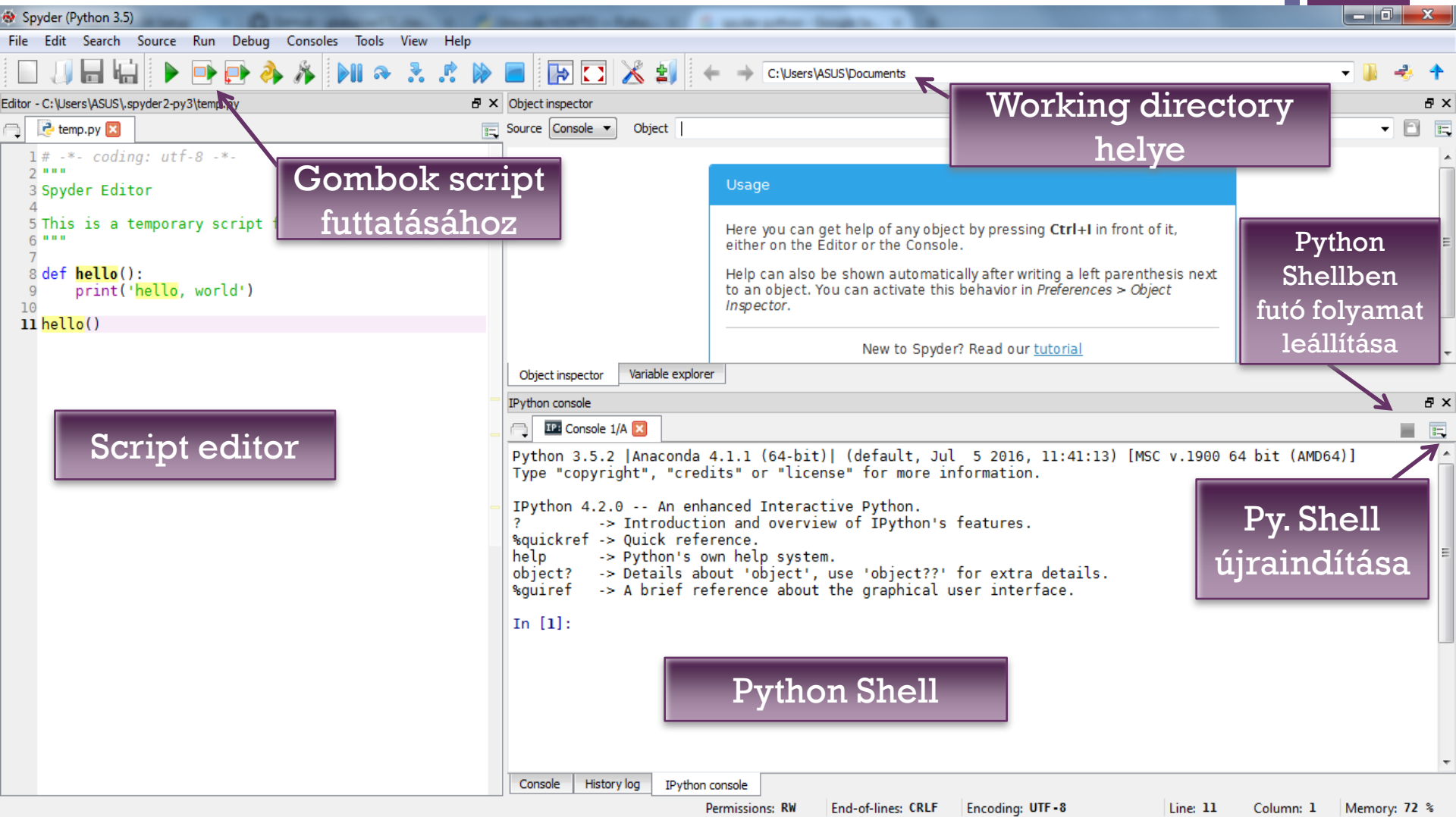
Az Emberi Erőforrások Minisztériuma Új Nemzeti Kiválóság Programjának támogatásával készült

Programozás nyelvészeknek (2. óra)

Ismerkedés a Spyder keretprogrammal és a Pythonnal

Python objektumok

Spyder


Spyder (Python 3.5)
File Edit Search Source Run Debug Consoles Tools View Help
C:\Users\ASUS\Documents
Working directory helye
Editor - C:\Users\ASUS\.spyder2-py3\temp.py
temp.py
-*- coding: utf-8 -*-
"""
Spyder Editor
This is a temporary script
"""
def hello():
print('hello, world')
hello()
Gombok script futtatásához
Script editor
Object inspector
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Object Inspector.
New to Spyder? Read our tutorial
Object inspector
Variable explorer
Python Shellben futó folyamat leállítása
IPython console
Console 1/A
Python 3.5.2 |Anaconda 4.1.1 (64-bit)| (default, Jul 5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 4.2.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%guiref -> A brief reference about the graphical user interface.
In [1]:
Py. Shell újraindítása
Python Shell
Console
History log
IPython console
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 11
Column: 1
Memory: 72 %

Egyszerű Python objektumok

- Integer (egész számok)
- Float (lebegőpontos számok)
- Változók létrehozása
- String (karaktersor)

Integer (Egész szám)

- Például
  - 7
  - 1
  - -2
  - 0
- type(2)

Float (Lebegőpontos szám)

- 3.14159265
- Vigyázz a szám nem egész részét PONT jelöli nem pedig vessző!
- CSAK közelíti a valós számokat:
  - 10/3
  - print('%.50f' % (10/3))
- egy végtelen hosszú számot nem lehet pontosan tárolni a számítógép véges memóriájában

Python mint számológép

- Egyszerű aritmetikai operátorok (+, -, /, *)
- 2 + 2 # kifejezés, amelyet Python képes értékelni
- Egyéb operátorok
  - ** # hatvány, 2^2 Pythonban 2**2
  - // # int osztás (9/2) [vigyázz Python 2-ben nem így működik!!]
  - % # maradék visszaadás (7%5)
- Zárójel ( ) ugyanúgy működnek, mint a matekban. A számítások sorrendjét adja meg
- (2 + 3) * 5 ≠ 2 + 3 * 5

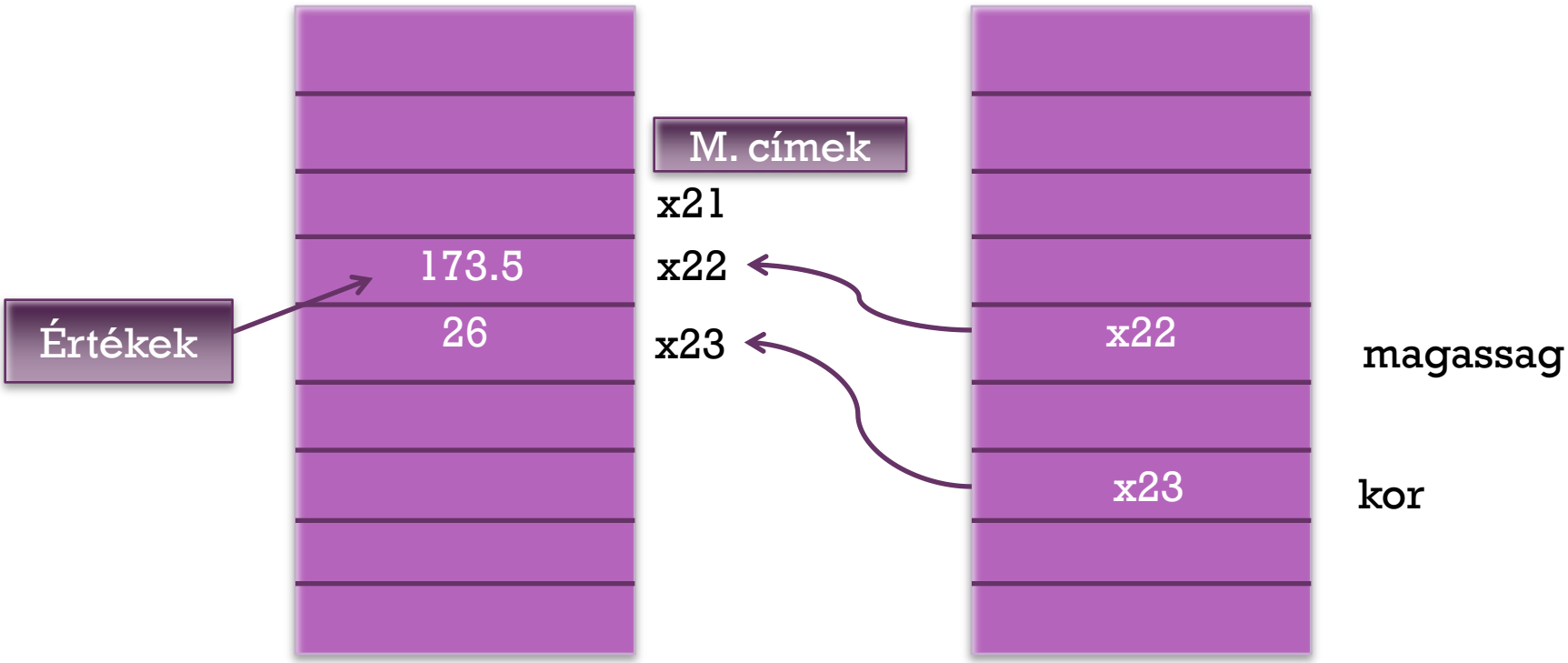
Feladat - Mi lesz az eredménye a következő kifejezéseknek?

1. 17 - 23 * 4
2. Mi az első eredményének típusa?
3. 54/67
4. Mi a 3. kifejezés típusa?
5. 2.3 + 7
6. Mi az 5. eredményének típusa?
7. (45 + 23.5) * 2 - 7
8. 5 ** 3
9. 6 // 5
10. int(6/5)
11. 13 % 9

Változók elnevezése és tárolása

- Memória

Változó = elnevezett hely a memóriában



Változók

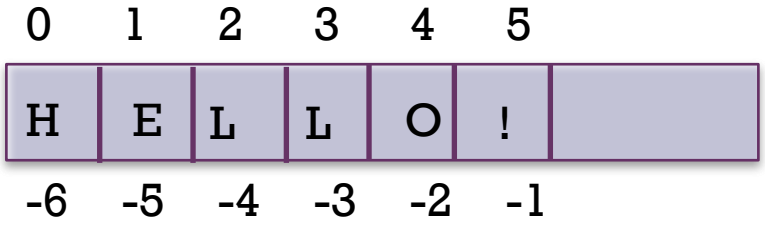
- változó létrehozása a ‘=‘ *hozzárendelő operátorral*
  - `x = 4`
- hex(id(x))
- felülírhatók
  - `x = 7`
  - `x = x + 4  # shorthand x += 4`
  - `y = 3.0`
  - type(y)

String (karaktersor)

- szövegként kezelt változó
- idézőjelek között szerepel
  - ‘szöveg’
  - “és ez is szöveg”

String tulajdonságai

- van hosszuk >> len()
- indexelhetők
  - x = ‘hello!’
  - x[0]
- / a menekülő karakter (pl. “\”” >> a string ‘)
- speciális kombinációk: \n >> új sor, \t >> tab stb.
- print()



You know you're a programmer when..

you count 3 apples

Feladat - Mi a következő kifejezések értéke?

1. `x = 'Have a good day!'`
2. `x[:]`
3. `x[1:]`
4. `x[::-1]`
5. `x[18]`
6. `x[:18]`
7. `x[::2]`

Mi a bug?

- Hiba a számítógépes programban, ami hatására, az nem úgy működik, mint elvárt.

Modern Bug

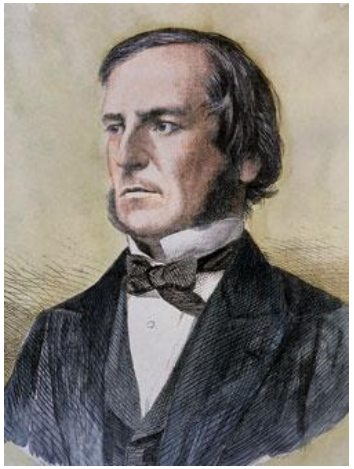
```
In [51]: s[18]
Traceback (most recent call last):

  File "<ipython-input-51-1c65de0552f6>", line 1, in <module>
    s[18]

IndexError: string index out of range
```

Feladat - Írj Python kefejezést, ami a következőket hajtja végre!

- x = ‘hello’
- city = ‘Budapest’

1. Add vissza az x változó utolsó két karakterét!
2. Add vissza a city változó első és utolsó karakterét!
3. Add vissza az x változó első, második és harmadik karakterét!
4. Add vissza a city változó 3-tól 6. karakterét.
5. Add vissza a city változó minden harmadik karakterét!
6. Hozzd létre az x és city változók felhasználásával a következő string-et: ‘hello Budapest’

Boolean



George Boole

- Logikai művelet
- True/False (Igaz / Hamis)
- a kifejezés igazságértéke (Esik az eső? >> Igaz vagy Hamis)
- a feltételes utasításokban hasznalt struktúra (if, while)

```
1. ha esik az eső:
2.       vigyél esernyőt
```

```
>>> 3 < 2
False
```

Operátorok:

<	kisebb
>	nagyobb
==	egyenlő
!=	nem egyenlő
>=	nagyobb vagy egyenlő
<=	kisebb vagy egyenlő
and	logikai és
or	logikai vagy
not	logikai nem

Feladat - Milyen értéket adnak a következő kifejezések?

- x = hello
- num = 5

1. len(hello) == num
2. num**2 >= 100
3. True == False
4. 3 == 3.0
5. type(3) != type(3.0)
6. ‘b’ <= ‘a’
7. not(True) == False
8. len(x*2) == num + 4
9. type(False) == bool
10. type(x) != type(‘77’)
11. x == num
12. x <= num

Feladat - Írj olyan Python kifejezéseket, amik a következő eredményt adják!

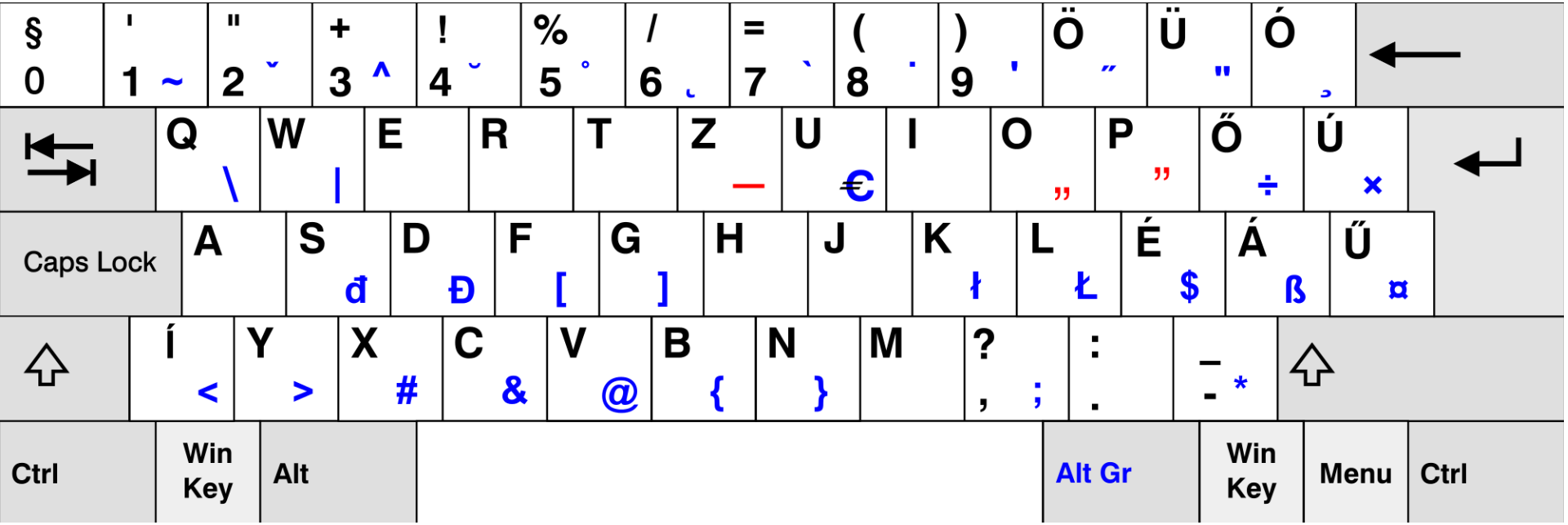
- x = 4.5
- y = ‘ELTE’
- z = False

1. Az Y változó hosszabb, mint az X változó értéke?
2. Az Y változó típusa string?
3. Ha az Y változó 2. elemét megszorozzuk X egész szám részével, a kifejezés értéke ‘EEEE’ lesz?
4. *Az X típusa megegyezik Y típusával* kifejezés értéke megegyezik Z változó értékével?
5. Nem Z változó értéke megegyezik Z értékével?

None

- Nonetype
- az érték hiányát jelzi

Speciális karakterek a magyar billentyűzeten



Otthoni Feladat

1. A https://github.com/gtakacse/CS_class_scripts/tree/master/homework címről töltsd le a hw1_basic_data_types.py fájlt.
2. Minden "Ide írd a kódodat!" helyre illesztd be a saját megoldásodat!
3. Miután végeztél a fájlt mentsd el úgy, hogy a neve a 'hw1_VEZETÉKNEVED_KERESZTNEVED.py' legyen!
4. A fájlt küld el az email címemre!

Képek forrása

- BedBugs.org
- https://en.wikipedia.org/wiki/George_Boole
- https://news.digitalmediaacademy.org/tag/where-was-the-first-computer-bug-found/
- http://www.idgconnect.com/blog-abstract/10357/this-tech-history-september-first-bug-literally
- https://www.python.org/
- plus.google.com
- https://hu.wikipedia.org/wiki/AltGr