

Python Tréning "2"

(Mostly) advanced

II. Nyelvi elemek, adatszerkezetek (gyakorlati alkalom)

Story so far

Melegítés (pótlás):

- típusok és konverzió, gyors áttekintés
- hasznos `__builtin__` nyelvi eszközök (`dir()`, `help()`, `type(t)`...)
- import szintaxis, (példa complex számokkal)
- adatszerkezet alapok, konverziók, tuple csomagolás és swap
- logikai kifejezések és lehetőségek (egysoros `if else`), `any()`, `all()`, `in`...)

Story so far

További gyakorlati témák:

- mutable és immutable típusok
- string, string formázás, file írás és olvasás
- szekvenciák (`list`, `tuple`, `array`), felhasználásuk, indexelések
- halmazok, dictionary, előnyök és hátrányok
- iterálás collection típusokon, `iter`, `enumerate`, `zip`, `len`, `max`, `min`...
- függvények, scope, paraméter referálás, `*args` és `**kwargs`

**But first, something completely
different...**

Alap típusok

- int, float
- str, (unicode, bytearray...)
- long, (complex)
- bool (`True`, `False`), NoneType (`None`)
- tuple, list, set, dict, (array...)
- ajánlott összefoglaló:
<https://docs.python.org/2/library/stdtypes.html>

Indexelés, szeletelés

- általános szintaxis:

[*ettől/kezdve* : *ez előttig* : *lépésközzel*]

- elejétől vagy végéig: *üres* :
- indexelés hátulról: *negatív* index
 - legelső: **0**, legutolsó: **-1**
- változatok: **range(to)**, **range(from, to)**,
range(from, to, step)

Operátorok, literálok

- klasszikusok: `=`, `==`, `!=`, `+`, `-`, `*`, `/`, `**`, `%`, `//`, `,`
 - multifunkciós: `+`, `*`, `%`, `>`, `<`, `>=`, `<=`
 - klasszikus bitwise: `&`, `|`, `^`
 - logikai (!!!): `and`, `or`, `not`, `in`
- long: `L`, complex: `j`, számrendszerek: `0b`, `0`, `0x`
- str: `" "`, `' '`, `"""`, repr: `r`, unicode: `u`
- tuple: `(a,)`, list: `[]`, dict: `{ key : value, }`, set: `{ a, }`

Beolvasás, file

- bemenetről: `input`, `raw_input()`

File buffer

```
f = open("file.name", "r+") # "r", "w", "a", "w+", "rb", "w+b"
# f.read(), f.readline(), f.readlines(), f.seek() ...
# f.write(), f.writelines()
# f.flush()
f.close()
```


Beolvasás, file

Biztonságosabb módszer:

```
lines = []  
with open("c:\\path\\to\\file.txt", "r") as file_read:  
    # work with file_read object  
    for curr_line in file_read:  
        # like: curr_line = file_read.readline()  
        if "sample string" in curr_line:  
            lines.append(curr_line)  
  
# file_read object closed automatically!
```

String feldolgozás, formázás

- concat, `bytearray`... (numpy str array)
- formázó karakterek:

```
"int: %03d f1: %.3f str: %s\n" % (42, math.pi, "abc")
```

```
rec = {"name": u"Kovács Dezső", "phone": "+3611234567"}  
s = "név: {} \t tel.: {}\n".format(rec["name"], rec["phone"])
```

```
data = ("some string", 123, 456)  
s2 = "n1: {1}\tn2: {2}\ts: {0}"  
s2.format(*data)
```

- metódusok: `.center()`, `.title()`, `.capitalize()`, `.strip()`,
`.replace()`, `.split()`, ...

dictionary, set

collection bejárás

Kérdések?

Folyt. köv.