PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 5

AGENDA
DAY 5

- Listy, Listy zagnieżdżone
- kopiowanie list
- tuple
- dict
- import

1.
Listy

list()
[]

```
lista = [1, 2, 3]
lista2 = ["kwiatek", "doniczka", "ziemia", "woda"]
lista3 = []
lista4 = [1, "dwa", 3, 4]
lista5 = list(range(2,5))
```

Możemy indeksować, slice'ować
Do elementu odwołujemy się przez indeks

listy zagnieżdżone

```
lista = [[1,2,3],[4,5,6],[7,8,9]]

lista = [[1,2,3],
      [4,5,6],
      [7,8,9]]

lista[1][2] => 6
```

2.
break, continue

modyfikacja pętli

continue, break

oba słowa kluczowe używane w instrukcji warunkowej (if-elif-else) wewnątrz pętli., modyfikują działanie pętli:

continue – program pomija pozostałe instrukcje w bloku i wraca do sprawdzenia warunku (while) lub do kolejnego elementu (for)

break – działanie pętli jest przerywane, program przechodzi do kolejnej instrukcji po całym bloku pętli

kopiowanie list

czyli początek zabawy z obiektami

kopiowanie list

Listy są typami referencyjnymi.

jeśli przypiszemy listę do innej zmiennej to tak naprawdę przypiszemy adres w pamięci do listy

możemy użyć kopiowania list:

```
nowa_lista = lista.copy()
nowa_lista = list(stara_lista)
nowa_lista = stara_lista[:]
```

ale czy to zawsze działa???

kopiowanie list

Listy są typami referencyjnymi.

do głębokiego kopiowania (kopiowanie wszystkiego jako wartość) używamy modułu copy i metody deepcopy()

```
import copy

nowy = copy.deepcopy(stary)
```

Tuple

tuple
krotka ()

Tuple jest typem niezmiennym – raz zdefiniowanego nie można zmienić

```
tuple1 = ("raz", "dwa", "trzy")

tuple1[0] = "jeden" – spowoduje błąd

x = "raz",
y = "raz", dwa"
```

rozpakowanie tupli

```
tuple1 = ("raz", "dwa", "trzy")

x, y, z = tuple1

print(x)
>>> "raz"
print(y)
>>> "dwa"
print(z)
>>> "trzy"
```

for **indeks**, **element** in enumerate(kolekcja):

Dict

dict
słownik {}

Zawiera pary klucz-wartość

klucz – musi być typem niezmiennym (string, tuple, liczba), musi być unikalny (tylko jeden w słowniku)

wartość – mogą być powtórzone

Odwołujemy się poprzez klucz a nie indeks!!!
x = {"nazwisko":"kowalski", "pesel":88120134567}
x['pesel']

dict
słownik {}

```
osoby = {"studenci":["Ala", "Jan", "Ania"], "wykladowcy":["doktor", "profesor"]}

print(osoby["studenci"][1])

osoby["wykladowcy"].append("magister")
osoby["administracja"] = ["pani Basia z dziekanatu"]

osoby.update({"ochrona":"Impel"})

print(osoby.keys)
print(osoby.values)

for key, item in osoby.items():
    print(key, item)
```

4.
import

ktoś już wykonał za nas pracę

import

```
import moduł
from moduł import funkcja
from moduł import *
```

```
string, datetime, copy, math, decimal,
random, os, csv, antigravity
```

Thanks!!