# Python!

## Lekcja 4 – pętle

# Plan

- Pętla – czyli właściwie co?
- Pętla while
- Pętla for
- break , continue oraz pass
- Iteracja po elementach tablicy
- Iteracja po indeksach tablicy
- Iteracja po słownikach

# Pętla – czyli właściwie co?

Pętla – sposób na wielokrotne wykonanie tej samej funkcji lub zestawu procedór. Pozwala zredukować czas potrzebny na napisanie kodu oraz zwiększa jego elastyczność

Pętle będą wykonywać zadany blok kodu (pamiętamy o wcięciach!)

W Pythonie są dwa rodzaje pętli:

- while
- for

# Pętla while

Pętla while będzie wykonywana tak długo jak warunek jest prawdziwy. Sprawdzanie prawdziwości warunku ma miejsce przed każdym obiegiem pętli.

```
>>> i = 0
>>> while i < 3:
...     print(i)
...     i += 1
...
0
1
2
>>>
```

Warunek jest wyrażeniem, którego wynik może mieć jedną z dwóch wartości: True i False

# Pętla for

Pętla *for* – wykonuje operacje na zadanym zestawie obiektów.

Najczęsciej jest to lista albo słownik. Można też iterować po krotkach, stringach lub podać zestaw obiektów bezpośrednio.

```
>>> for i in 1,2,3:
...     print(i)
```

```
>>> a = (1,2,3)
>>> for i in a:
...     print(i)
```

```
>>> a = [1,2,3]
>>> for i in a:
...     print(i)
```

```
>>> for i in "123":
...     print(i)
```

# break , continue oraz pass

Słowa kliczowe używane w pętlach to:

- break – przerywa wykonywanie ‘najmniejszej’ pętli
- continue – przerywa bieżący obieg ‘najmniejszej’ pętli
- pass – nic nie robi. Dosłownie.

```
>>> for i in 1,2,3:
...     if i == 2: break
...     print(i)
...
1
>>>
```

```
>>> for i in 1,2,3:
...     if i == 2: continue
...     print(i)
...
1
3
>>>
```

# Iteracja po elementach tablicy

Tablica (lista) może zawierać obiekty dowolnego typu.

Jednak warto tak projektować algorytmy, żeby wszytskie były podobne – w znacznym stopniu upraszcza to kod.

Tablice mogą być zagnieżdżone, ale to nie problem:

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = [7, 8, 9]
>>> d = [a, b, c]
>>> for i in d:
...     for j in i:
...             print(j)
...
```

# Iteracja po indeksach tablicy

Funkcja range jest niezwykle przydatna przy iteracji po indeksach w połączeniu z funkcją len.

W Python3 range nie jest funkcją, choć tak wygląda. Pomijając techniczne aspekty, range zwraca sekwencję liczb całkowitych.

class range(stop)

class range(start, stop[, step])

```
>>> a = [1, 2, 3]
>>> for i in range(len(a)):
...     print(a[i])
...
1
2
3
>>>
```

https://docs.python.org/3/library/stdtypes.html#typesseq-range

# Iteracja po słownikach

Słowniki można iterować w sposób podbny do list, jednak nie pozwala to w pełni wykorzystać właściwiści tego typu danych.

Dlatego należy iterować je po parach klucz:wartość.

Oczywiście można iterować po samych kluczach i samych wartościach.

```
>>> a = {'a': 1, 'b': 2, 'c': 3}
>>> for k, v in a.items():
...     print("klucz: {}, wartość: {}".format(k, v))
...
klucz: a, wartość: 1
klucz: c, wartość: 3
klucz: b, wartość: 2
>>>
```

Jak to będzie z:
a.keys()
a.values()
???

# Dziękuję za uwagę!