Atrybuty

Atrybut – „nazwa przywiązana (lub należąca) do obiektu”.
(nazwy nazywają obiekty, więc w szczególności atrybuty nazywają obiekty „przywiązane” do danego obiektu)

Składnia dostępu do atrybutu attribute obiektu obj: . (kropka):

obj.attribute

Przykład: liczby zespolone:

```
z = 2 + 1j      # liczba zespolona 2 + i
print(z.real) # część rzeczywista
print(z.imag) # część urojona
```

Przykład: moduły (obiekty) i ich atrybuty (np. funkcje, stałe):

```
import math

y = math.sin(math.pi / 2)
```

Metody

Metoda – atrybut, funkcja przywiązana do obiektu.

O takiej funkcji można myśleć, że jej ukrytym parametrem jest obiekt, do którego jest przywiązana.

Przykłady:

```
z1 = 2 + 1j              # liczba zespolona
z2 = z1.conjugate()      # sprzężenie z1, bez parametrów
# z2 == 2 - 1j
```

```
lst = [1, 2, 3]
lst.append(4)    # dokłada 1 na koniec listy lst
# lst == [1, 2, 3, 4]
```

Operacje na listach

Atrybuty (w tym metody) obiektu są w znacznej części zdeterminowane przez typ tego obiektu.
Na wykładzie: niektóre metody list.

- append, extend, insert
- pop, remove, clear
- index, count
- sort, reverse
- copy

Oprócz tego, inne operacje na listach (nie przez metody):

- del, len
- min, max, sum
- sorted, reversed
- listy składane
- (extended) slicing

Listy składane

Listy składane (schematy konstrukcji list) – skrótowe wyrażenia konstruujące listy z innych list (lub napisów, etc.).
„Ręczna" konstrukcja listy kwadratów $[0, 1, 4, 9, 16, 25]$:

```
lst = [] # pusta lista
for i in range(6):
    lst.append(i ** 2)
```

Taką listę można stworzyć (i potem nazwać) pojedynczym wyrażeniem:

```
lst = [i ** 2 for i in range(6)]
```

W liście składanej mogą pojawiać się warunki (odrzucające niektóre obiekty) oraz więcej iteracji, np.:

```
lst1 = [1 / x for x in range(-3, 3) if x != 0]
lst2 = [c * i for c in "abc" for i in range(3)]
```