

PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 6





AGENDA DAY 6

- Definiowanie funkcji
- Argumenty funkcji
- return
- •docstring





1. Definiowanie funkcji

Tworzymy re-używalny kod



tworzenie funkcji

```
def do_nothing():
      pass
>>> do nothing()
                                <- nic nie zrobił (dokładnie to co chcieliśmy)</pre>
>>> x = do nothing
                             <- funkcja jest obiektem możemy więc zapisać do zmiennej
>>> x()
                             <- zmienna z funkcją można wywołać
```



tworzenie funkcji

```
give_square(35)

def give_square(x):
    print(x**2)
```

```
give_square(35)
NameError: name 'give square' is not defined
```



argumenty funkcji

```
def do_nothing():
    pass

def do_nothing(x):
    pass

def do_nothing(x, y, z):
    wiele argumentów
    pass
```



argumenty domyślne

```
def do nothing(x, y=10):
    pass
def do nothing(x, y, z=12, w =,,0la"):
    pass
def do nothing(y=10):
    pass
argumenty domyślne muszą być po argumentach wymaganych
argument domyślny jest sprawdzany tylko przy pierwszym wywołaniu
funkcji - uwaga na typy referencyjne!
```



argumenty domyślne wywołanie funkcji

```
def do something(x, y, z=12, w =,,0la"):
    pass
>>> do something(1) <- błąd - wszystkie arg. pozycyjne muszą być podane
>>> do something(1, 23)
>>> do_something(1, 2, "trzy")
>>> do_something(1, 2, 34, "ola")
>>> do_something(1, 33, w="ola")
```



return

```
funkcja może robić coś wewnątrz siebie (nawet nie trzeba print)
def print square(x)
    print(x**2)
funkcja może oddać jakiś wynik/obiekt – używamy return
def give square(x)
    return x**2
aby użyć funkcję zwracającą obiekt należy ten obiekt zapisać w zmiennej
>>> wynik = give square(3)
>>> print(wynik)
>>>
```



argumenty domyślne – typy referencyjne

```
def dodaj_imie(imie, imiona=[]):
    imiona.append(imie)
    return imiona

>>> print(dodaj_imie("Ala"))
["Ala"]
>>> print(dodaj_imie("Ola"))
["Ala", "Ola"]
>>> print(dodaj_imie("Ewa"))
["Ala", "Ola", "Ewa"]
```

```
def dodaj_imie(imie, imiona=None):
    if imiona == None:
        imiona = []
    imiona.append(imie)
    return imiona

>>> print(dodaj_imie("Ala"))
["Ala"]
>>> print(dodaj_imie("Ola"))
["Ola"]
>>> print(dodaj_imie("Ewa"))
["Ewa"]
```



2. Zakres zmiennych

Jak daleko sięgać mogę



ZAKRES ZMIENNYCH zmienne lokalne i globalne

Zmienne lokalne funkcji są do wykorzystania tylko w tej funkcji (i głębiej)

Co jeśli mamy zmienną **imie** na poziomie głównego programu, oraz zmienną lokalną **imie** na poziomie funkcji

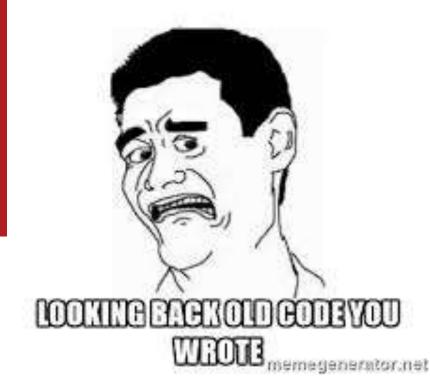
```
imie = "jola"

def drukuj imiona(imie 2):
   imie = "ania"
```



3. docstring

dokumentuj kod





docstring – PEP257 documentation string

```
def do_nothing(x, y=10):
    """Does absolutely nothing"""
    pass
def give square(x):
    """Returns square of given numer
    (numer) -> numer
    נננננ
    return x**2
```

https://www.python.org/dev/peps/pep-0257/





Thanks!!