

Funkcje

```
def funkcja_1():  
    print('A')
```

funkcja1()

```
def funkcja_2():  
    return('A')
```

wynik = funkcja_2()
print(wynik)

```
def funkcja_3(arg1):  
    print(arg1)
```

funkcja_3('ABC')

```
def funkcja_4(arg1=1)  
    print(arg1)
```

funkcja_4() *argument z wartością domyślną*
lub
funkcja_4(777)

```
def funkcja_5(*args):  
    for arg in args:  
        print(arg)
```

argumenty nienazwane

funkcja_5(4, 22, 't', 6)

```
def funkcja_6(**kwargs):  
    for key in kwargs:  
        print(key, kwargs[key])
```

argumenty nazwane

funkcja_6(ala=4, kot='Kicia', t=2)

Słowniki

```
my_dictionary = {'Anna': 25, 'Jan': 45}  
my_dictionary.keys()  
my_dictionary['Anna']
```

zwraca klucze słownika
wybiera wartość dla klucza 'Anna'

Zbiory

```
my_set = {5, 6, 7}  
my_set = set([5, 6, 7])  
stworzenie listy tylko z unikalnymi wpisami:  
my_list = list(set(original_list))
```

Wyjątki

```
try:  
    do_something  
except Exception as error:  
    print(error)
```

```
try:  
    do_something  
finally:  
    finish_something
```

```
try:  
    do_something  
except Exception as error:  
    print(error)  
finally:  
    finish_something
```

Klasy

```
class MyClass:  
    """ Description """  
  
    def __init__(self, *args, **kwargs):  
        print('Konstruktor')
```

```
    def method_a(self):  
        return 'A'
```

```
class MyOtherClass(MyClass):  
    """ Description """
```

```
    def method_b(self):  
        return 'B'
```

```
class_instance = MyOtherClass()  
print(class_instance.method_a())  
print(issubclass(MyOtherClass, MyClass))
```

Listy

```
my_list = ['el1', 'el2', 5, 6.7, (2,3)]  
wybranie elementu o indeksie 0:  
my_list[0]  
wybranie elementów od 1 do -2:  
my_list[1:-2]  
wybranie wszystkich elementów (kopiuje listę):  
my_list[:]  
ile razy występuje dany element:  
my_list.count(element)  
jaki indeks ma pierwszy znaleziony element:  
my_list.index(element)  
dodawanie nowego elementu na koniec listy:  
my_list.append(new_element)  
wstawianie nowego elementu pod zadany indeks:  
my_list.insert(index, new_element)  
łączenie dwóch list:  
my_list.extend(other_list)  
usuwanie elementu o zadany indeksie:  
del my_list[index]  
usuwanie i zwracanie ostatniego elementu:  
my_list.pop()  
usuwanie pierwszego znalezione elementu:  
my_list.remove(element)  
odwracanie kolejności listy:  
my_list.reverse()  
sortowanie listy (dwa sposoby):  
my_list.sort()  
sorted_list = sorted(my_list)  
sprawdzanie długości listy:  
len(my_list)
```

Krotki

```
my_tuple = (1, 4, 5, 1)  
my_tuple = tuple([4, 5, 6])
```