

Języki Skryptowe - Python

Wykład 3

Artur Barasiński



Podstawowe operatory porównania to

- operator równości (**==**)
- operator nierówności (**!=**)
- operator ostrej większości i mniejszości (**>**, **<**)
- operator nieostrej większości i mniejszości (**>=**, **<=**)
- operator "identyfikatora" (**is**)
- operator negacji "identyfikatora" (**is not**)
- operator zawierania (**in**)
- operator niezawierania (**not in**)

Operatory porównania zwracają wartość **True** lub **False**.

Przykład 1

```
>>> 'a' < 'z'           #kolejność leksykograficzna
True
>>> 'Z' < 'a'
True
>>> '9' < 'A'           #cyfry < litery
True
>>> '123' < '14'        #porównujemy znak po znaku
False
>>> 2 == 2 > 1.1 > -1 > -1000 #porównanie wielokrotne
True
```

Przykład 2

```
>>> a = 500
>>> b = 500
>>> a == b      #porównuje wartość
True
>>> a is b      #porównuje referencje
False
>>> b = a
>>> a is b      #porównuje referencje
True
```

Przykład 3

```
>>> "in" in "indigo"          #dla napisów
True
>>> "id" in "indigo"
False
>>> "in" in ["in", "out"]     #dla list
True
>>> "in" in ["indigo", "violet"]
False
>>> b = a
>>> "in" not in ["indigo", "violet"]
True
```

- Typ danych, który zwraca wartość **True** lub **False** nazywamy typem logicznym (boolowskim, boolean).
- W Pythonie następujące wartości są utożsamiane z fałszem:
 - ▶ False
 - ▶ None
 - ▶ wartość zero (0,0.0,0j)
 - ▶ pusta sekwencja "", (), []
- Pozostałe wartości uznawane są za prawdę

- Suma logiczna (**or**) np. *True or False* \rightarrow *True*
- Iloczyn logiczny (**and**) np. *True and False* \rightarrow *False*
- Negacja logiczna (**not**) np. *not (True and False)* \rightarrow *True*

W pierwszej kolejności wykonywane są operatory porównania, później operator negacji, następnie iloczynu logicznego, a na końcu sumy logicznej (w kolejności od lewej do prawej).

Np. w języku C

```
if (x > y) {  
    x = 1;  
    y = 2;  
}
```

Python

```
if x > y:  
    x = 1  
    y = 2
```


Np. w języku C

```
if (x > y){  
    x = 1;  
    y = 2;  
}
```

Python

```
if x > y:  
    → x = 1  
    → y = 2
```

Python

```
if warunek1:  
    instrukcja1  
    instrukcja2  
    ...  
elif warunek2:  
    instrukcja3  
    ...  
.  
.  
.  
else:  
    instrukcja4  
    instrukcja5
```

Przykład 4

```
if 2 > 3:  
    print("2 jest większe od 3")  
elif 2 == 3:  
    print("2 jest równe 3")  
else:  
    print("2 jest mniejsze od 3")
```

Dziękuję za uwagę!