

## Python – operatory działań

Operator	Działanie	Przykład
+	dodawanie	4 + 7 → 11
-	odejmowanie	21 - 4 → 17
*	mnożenie	4 * 3 → 12
/	dzielenie rzeczywiste	5 / 2 → 2.5
//	dzielenie całkowite	5 // 2 → 2
%	reszta z dzielenia całkowitego	5 % 2 → 1
**	potęgowanie	4 ** 2 → 16

## Python – operatory porównania

Operator	Działanie	Przykłady	
==	równe	4 == 4 → True	4 == 5 → False
!=	różne	4 != 7 → True	4 != 4 → False
<	mniejsze	3 < 9 → True	9 < 3 → False
<=	mniejsze równe	3 <= 3 → True	5 <= 4 → False
>	większe	4 > 1 → True	1 > 4 → False
>=	większe równe	4 >= 4 → True	3 >= 4 → False

## Python – polecenia

Instrukcja <code>print()</code> – wypisywanie		
<code>print("las")</code> ↓ las	<code>print(3, 24)</code> ↓ 3 24	<code>print()</code> ↓ pusty wiersz
Funkcje		
<code>def &lt;nazwa funkcji&gt;(parametry):</code> <instrukcje>		
<code>def godziny(t):</code> g = t // 60 return g	<code>def droga(v, t):</code> return v * t	
<code>print(godziny(325))</code> ↓ 5	<code>print(droga(53, 2))</code> ↓ 106	
Instrukcja przypisania		
<code>&lt;nazwa&gt; = &lt;wartość&gt;</code>		
<code>x = 45</code>		

### Instrukcja warunkowa – prosta

```
if <wyrażenie>:
    <wykonaj instrukcje>
```

```
if t == 4:
    print("dobry")
```

### Instrukcja warunkowa – złożona

```
if <wyrażenie>:
    <wykonaj instrukcje1>
elif <wyrażenie>:
    <wykonaj instrukcje2>
else:
    <wykonaj instrukcje3>
```

```
if x == 3:
    print("dodatni")
elif x == -3:
    print("ujemny")
else:
    print("nieoznaczony")
```

### Pętla for

```
for <nazwa_zmiennej> in <sekwencja>:
    <instrukcje>
```

```
for i in range(6):
    print(i + 1)
```

### Pętla while

```
while <wyrażenie>:
    <instrukcje>
```

```
while x > 100:
    print(x)
    x = x - 20
```

### Listy

```
lista = [2, 4, 5, 6]
```

```
print(lista[0]) → 2
print(lista[3]) → 6
```

```
lista = lista + [7]
print(lista) → [2, 4, 5, 6, 7]
```

```
lista = [1] + lista
print(lista) → [1, 2, 4, 5, 6, 7]
```

```
for x in lista:
    print(x)
for i in range(len(lista)):
    print(lista[i])
```