

Operatory arytmetyczne i przypisania w Pythonie

M@я3k Pцđ€£kØ

Programowanie w Pythonie

Spis treści

- Operator przypisania
- Operatory arytmetyczne
 - Dodawanie
 - Odejmowanie
 - Mnożenie
 - Dzielenie
 - Dzielenie całkowitoliczbowe
 - Dzielenie modulo
- Złożone operatory przypisania

Operator przypisania

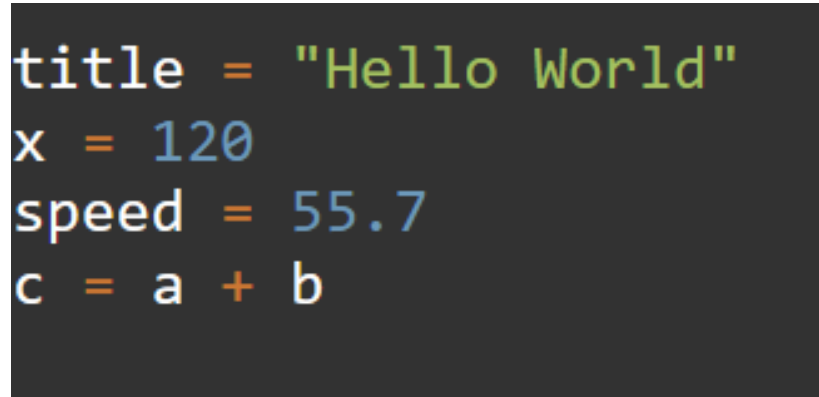
- W Pythonie operator przypisania to pojedynczy znak równości =, służący do przypisywania wartości do zmiennych.

```
title = "Hello World"
```

```
x = 120
```

```
speed = 55.7
```

```
c = a + b
```



```
title = "Hello World"  
x = 120  
speed = 55.7  
c = a + b
```

- Operatorem przypisania = nie należy mylić z operatorem porównania ==.
 - Operator == jest operatorem porównania sprawdzającym, czy dwie wartości są równe (zwraca **True** lub **False**).

Operatory arytmetyczne

- W Pythonie operatory arytmetyczne służą do wykonywania podstawowych obliczeń na liczbach.

Operator	Działanie
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
//	dzielenie całkowitoliczbowe
%	resztę z dzielenia (dzielenie modulo)
**	potęgowanie

Dodawanie

- Operator dodawania + dodaje dwie lub więcej liczb.

```
print (24+46)
x = 15+20+11+45+9
print (x)
a=9
b = 11
c = a+b
print (c)
```

```
print (24+46)
x = 15+20+11+45+9
print (x)
a=9
b = 11
c = a+b
print (c)
```

70

100

20

Odejmowanie

- Operator odejmowania - odejmuje dwie lub więcej liczb.

```
print (64-24)
x = 100-22-38-10
print (x)
a = 25
b = 15
c = a - b
print (c)
```

```
print (64-24)
x = 100-22-38-10
print (x)
a = 25
b = 15
c = a - b
print (c)
```

40

30

10

Mnożenie

- Operator mnożenia * mnoży dwie lub więcej liczb.

```
print (12*24)
x = 1*2*3*4*5*6
print (x)
a = 2
b = 4
c = a * b
print (c)
```

```
print (12*24)
x = 1*2*3*4*5*6
print (x)
a = 2
b = 4
c = a * b
print (c)
```

288

720

8

Potęgowanie

- Operator potęgowania `**` odnosi pierwszą liczbę do potęgi drugiej.
- Na przykład `2 ** 3` to $2^3 = 2 * 2 * 2 = 8$.

```
print (2**8)
```

```
x = 3**4
```

```
print (x)
```

```
a = 5
```

```
b = 2
```

```
c = a ** b
```

```
print (c)
```

```
print (2**8)
```

```
x = 3**4
```

```
print (x)
```

```
a = 5
```

```
b = 2
```

```
c = a ** b
```

```
print (c)
```

256

81

25

Dzielenie

- Operator dzielenia / dzieli dwie liczby.
- Wynik zawsze jest liczbą zmiennoprzecinkową (float), nawet jeśli obie liczby są całkowite.
- Druga liczba nie może być równa zero!

```
x = 30 / 5
print (x)
print (type(x))
a = 2
b = 8
c = a / b
print (c)
```

```
x = 30 / 5
print (x)
print (type(x))
a = 2
b = 8
c = a / b
print (c)
```

6.0
<class 'float'>
0.25

Dzielenie

- Druga liczba nie może być równa zero!

```
a = 2
b = 0
c = a / b
print (c)
```

```
a = 2
b = 0
c = a / b
print (c)
```

```
Traceback (most recent call last):
  File "/home/main.py", line 3, in <module>
    c = a / b
        ~~~^~~
ZeroDivisionError: division by zero
```

Dzielenie całkowitoliczbowe

- Operator dzielenia całkowitoliczbowego `//` dzieli dwie liczby, a następnie obcina część dziesiętną, zwracając wynik w postaci liczby całkowitej (`int`).

```
x = 30 // 7
print (x)
print (type(x))
a = 2
b = 8
c = a // b
print (c)
```

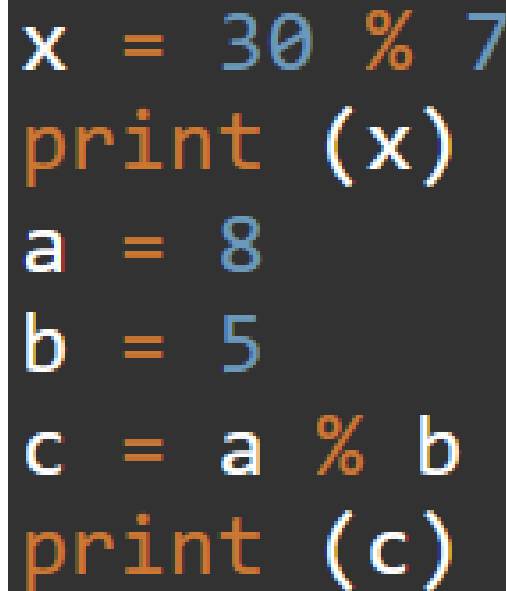
```
x = 30 // 7
print (x)
print (type(x))
a = 2
b = 8
c = a // b
print (c)
```

4
<class 'int'>
0

Dzielenie modulo

- Operator dzielenia modulo % zwraca resztę z dzielenia dwóch liczb całkowitych.

```
x = 30 % 7
print (x)
print (type(x) )
a = 2
b = 8
c = a % b
print (c)
```

A screenshot of a code editor with a dark background. It contains Python code demonstrating the modulo operator. The code is: x = 30 % 7, print (x), a = 8, b = 5, c = a % b, and print (c). The text is color-coded: variables are white, numbers are blue, and the modulo operator and print statements are orange.

```
x = 30 % 7
print (x)
a = 8
b = 5
c = a % b
print (c)
```

Porównanie dzielen

50 dzielone przez 8

Dzielenie

Dzielenie
całkowite

Dzielenie
Modulo
(reszta z
dzielenie)

$$50 / 8 == 6.25$$

$$50 // 8 == 6$$

$$50 \% 8 == 2$$

$$50 == 8 * 6.25$$

$$50 == 6 * 8 + 2$$

Złożone operatory przypisania

$+=$	$x += 2$ $x = x + 2$	dodaj i przypisz	Dodaje wartość po prawej stronie do zmiennej po lewej stronie i przypisuje wynik
$-=$	$x -= 2$ $x = x - 2$	odejmij i przypisz	Odejmuje wartość po prawej stronie od wartości zmiennej po lewej stronie
$*=$	$x *= 2$ $x = x * 2$	pomnóż i przypisz	Mnoży wartość zmiennej po lewej stronie przez wartość po prawej stronie
$/=$	$x /= 2$ $x = x / 2$	podziel i przypisz	Dzieli wartość zmiennej po lewej stronie przez wartość po prawej stronie
$**$ $=$	$x **= 2$ $x = x ** 2$	potęgowanie i przypisanie	Podnosi wartość zmiennej do potęgi
$//$ $=$	$a //= 30$ $a = a // 30$	dzielenie całkowite i przypisanie	Wykonuje dzielenie całkowite i przypisuje wynik
$\%$ $=$	$a \%= 7$ $a = a \% 7$	dzielenie modulo i przypisanie	Wykonuje dzielenie modulo i przypisuje wynik

Złożone operatory przypisania

- Wykonują operację matematyczną na wartości zmiennej, a następnie przypisują wynik z powrotem do tej zmiennej.
- Uproszczenie kodu
 - Złożone operatory eliminują potrzebę dwuetapowego procesu: najpierw wykonania operacji, a potem przypisania jej wyniku do tej samej zmiennej. Łączą operację arytmetyczną z przypisaniem wyniku do zmiennej w jednym kroku.
- Zwiększona czytelność
 - Kod staje się bardziej zwarty i łatwiejszy do zrozumienia, zwłaszcza w sytuacjach, gdy często aktualizuje się wartości zmiennych, np. w pętlach.
- Wydajność
 - W niektórych przypadkach mogą poprawiać wydajność kodu, chociaż głównym celem jest klarowność.
- Unikanie błędów
 - Użycie operatorów złożonych pomaga uniknąć powtarzania nazwy zmiennej, co może prowadzić do błędów.
- Naturalne odzwierciedlenie matematycznych operacji
 - Pozwalają na pisanie kodu, który bardziej naturalnie odzwierciedla matematyczne koncepcje, np. zwiększanie wartości o stałą liczbę (licznik += 1).

Ćwiczenia 1

1. Wczytaj trzy liczby, oblicz ich średnią i wypisz wynik. Użyj operatorów `+=` oraz `/=`.
2. Oblicz pole prostokąta o bokach `a` i `b`. Następnie zwiększ wartość pola o 10% przy użyciu operatora `*=`.
3. Ustaw zmienną `x = 12`. Użyj kolejno operatorów `+=`, `-=`, `*=`, `/=`, by zmodyfikować wartość `x`. Wypisz wartość po każdej operacji.
4. Poproś użytkownika o podanie dwóch liczb. Wypisz wyniki wszystkich działań arytmetycznych (`+`, `-`, `*`, `/`, `//`, `%`). Użyj złożonych operatorów przypisania.

Ćwiczenia 2

1. Pobierz od użytkownika liczbę i oblicz jej kwadrat dwoma sposobami:
 - a) za pomocą operatora `*`,
 - b) za pomocą operatora przypisania `**=`.
2. Pobierz od użytkownika dwie liczby `a` i `b`. Oblicz `a` do potęgi `b`, a następnie resztę z dzielenia wyniku przez 10
3. Wymiana wartości z użyciem zmiennej pomocniczej.
 - Masz dwie zmienne **`a = 5`**, **`b = 8`**. Użyj trzeciej zmiennej, by zamienić ich wartości miejscami, używając operatorów.
4. Wymiana wartości bez zmiennej pomocniczej:
 - Masz dwie zmienne **`a = 5`**, **`b = 8`**. Zamień ich wartości miejscami, używając operatorów przypisania (bez trzeciej zmiennej).

Ćwiczenia 3

1. Punktacja gracza:

- Gracz zaczyna z liczbą punktów **punkty = 100**.
- traci 15 punktów,
- zyskuje 40,
- jego wynik mnoży się przez 1.2 (bonus),
- następnie traci 10% za spóźnienie.
- Użyj operatorów przypisania, aby obliczyć końcowy wynik.

2. Zamiana czasu:

- Pobierz liczbę minut i przelicz ją na godziny oraz minuty (np. 135 minut → 2 godziny i 15 minut).
- Skorzystaj z operatorów // i %.

3. Rabaty i podwyżki:

- Cena produktu to cena = 250.
- Najpierw obniż cenę o 10% (operator *=)
- Potem zwiększ o 10%
- Wypisz cenę końcową po dwóch operacjach. Czy będzie taka sama?

Powtórzenie

1. Wymień znane ci operatory arytmetyczne w Pythonie.
2. Jak uzyskać z potęgowania pierwiastkowanie?
3. Jakiej wartości nie może przyjąć dzielnik?
4. Czym się różni dzielenie od dzielenia całkowitoliczbowego i dzielenia modulo?
5. Czym się różni `=` od `==`?
6. Po co stosuje się złożone operatory przypisania?