

Zmienna

Zmienna to podstawowy element języka programowania. Każdą zmienną można opisać za pomocą:

nazwy – określa nazwę, przez którą następuje odwołanie do zmiennej;

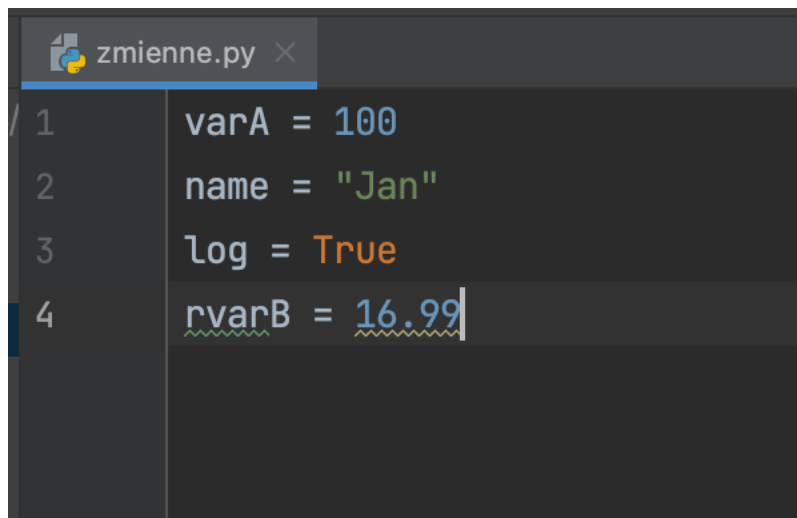
adres – określa bieżący adres zmiennej w pamięci;

wartość – określa wartość przypisaną do zmiennej;

zakres widoczności – określa blok instrukcji, w których zmienna może przyjmować;

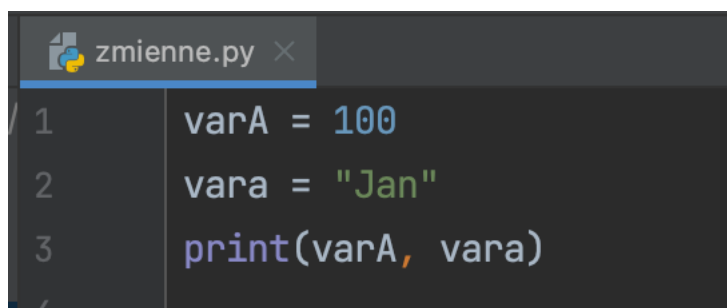
okres życia – określa czas, przez który zmienna istnieje. Przykładowo okres życia zmiennej zadeklarowanej w funkcji zaczyna się od wejścia do i funkcji i kończy się w momencie zakończenia funkcji.

Potocznie mówi się o „zadeklarowaniu zmiennej” oznacza to, że należy wpisać lub podać nazwę oraz opcjonalnie jej wartość początkową. Zmienna ta jest pojemnikiem w pamięci komputera przechowująca wartość pod określoną nazwą. Przykładem takiej zmiennej jest przechowywanie nazwy postaci w grze komputerowej.



```
zmienne.py x
1  varA = 100
2  name = "Jan"
3  log = True
4  rvarB = 16.99
```

W języku Python najlepiej nazywać zmienne w języku angielskim, nie stosując polskich znaków. Nazwa zmiennej ma być tak utworzona, aby dobrze opisywała naszą zmienną, aby wracając do aplikacji za kilka tygodnie nie trzeba było zastanawiać się co ta zmienna przechowuje. Dodatkowo wielkość liter w nazwach zmiennych ma znaczenie. Zmienna varA oraz vara to są zupełnie inne zmienne. Nie jest dopuszczalne używanie spacji w nazwach zmiennych. Nie możemy również nazywać zmiennych rozpoczynając od liczby lub znaku specjalnego.



```
zmienne.py x
1  varA = 100
2  vara = "Jan"
3  print(varA, vara)
4
```

Typy zmiennych

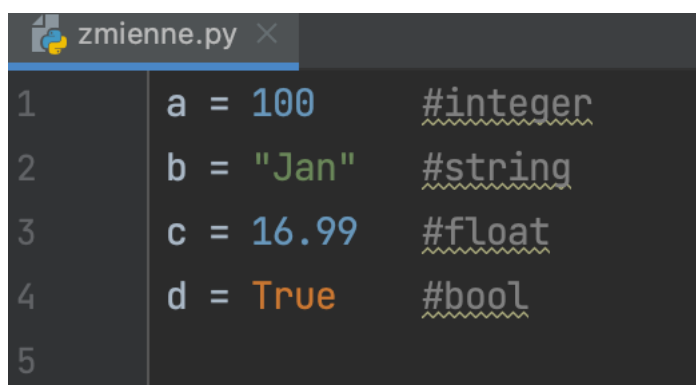
Jeśli już wiemy, czym jest zmienna, to teraz musimy przedstawić typy zmiennych. Typ zmiennej to rodzaje wartości, które chcemy przechować. W zmiennych możemy przechowywać:

liczby całkowite — typ integer

liczby zmiennoprzecinkowe — typ float lub double

ciągi znaków — typ string

wartość logiczna — typ bool

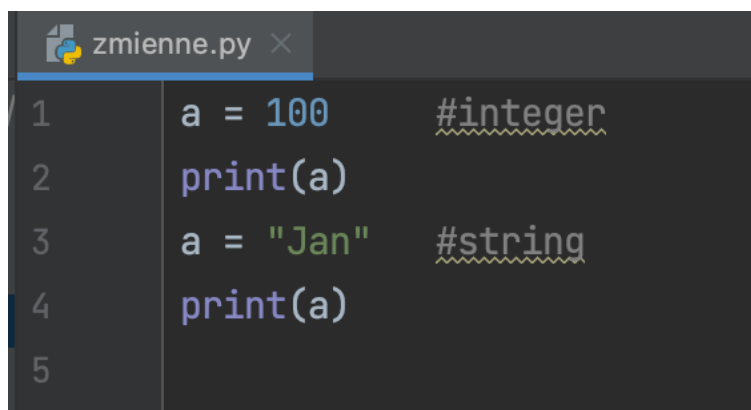


```
zmienne.py x
1 a = 100      #integer
2 b = "Jan"    #string
3 c = 16.99    #float
4 d = True     #bool
5
```

Jak widzimy, musimy uważać na to co przypisujemy do zmiennej.

UWAGA!

W języku Python możemy zadeklarować zmienną znakową (string) ale za chwilę zmienni typ tej zmiennej poprzez przypisanie liczby całkowitej. Nie będzie to błędem, musimy pamiętać poprzez dodawanie komentarzy co zawiera zadeklarowana zmienna.



```
zmienne.py x
1 a = 100      #integer
2 print(a)
3 a = "Jan"    #string
4 print(a)
5
```

Operatory arytmetyczne

Po poznaniu co to jest zmienna i jakie może mieć typy, musimy poznać co to jest operator arytmetyczny.

Czym jest operator arytmetyczny?

Operatorem nazywamy symbole, które w konkretnym wyrażeniu przyczynią się do wykonania jakiegoś działania - zazwyczaj matematycznego. Operatory możemy podzielić na jednoargumentowe (inkrementacja, dekrementacja, zmiana znaku liczby) oraz dwuargumentowe (dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie, modulo, konkatenacja).

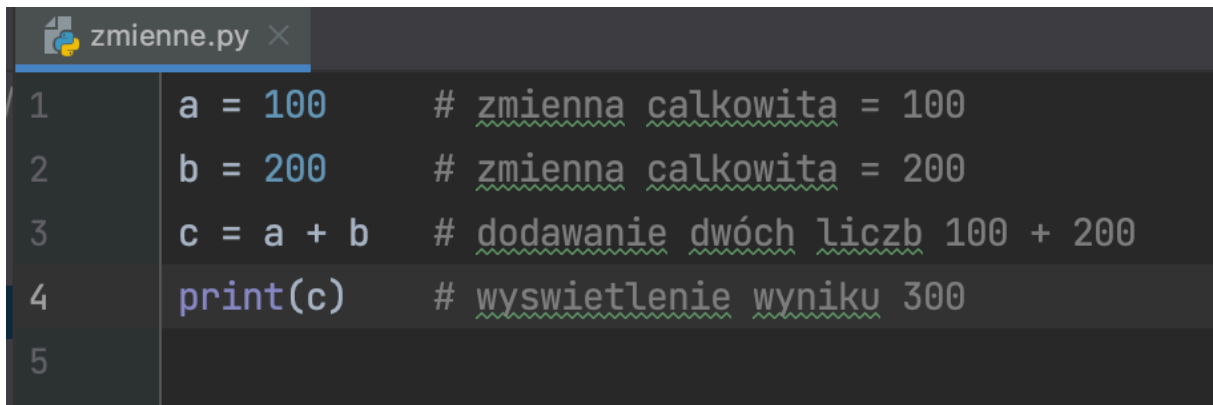
Podstawowymi operatorami arytmetycznymi w Python są:

- + (dodawanie, konkatenacja)
- (odejmowanie, zmiana znaku liczby)
- / (dzielenie)
- * (mnożenie)
- % (modulo)
- ** (potęgowanie)
- // (dzielenie z zaokrągleniem w dół, do liczby całkowitej)

Wyjaśnijmy wszystkie te operatory na przykładzie.

+ (dodawanie, konkatenacja)

Przykład **dodawania** dwóch zmiennych możemy przedstawić:



```
1 a = 100      # zmienna całkowita = 100
2 b = 200      # zmienna całkowita = 200
3 c = a + b    # dodawanie dwóch liczb 100 + 200
4 print(c)     # wyświetlenie wyniku 300
5
```

Ale co to jest **konkatenacja**? Jest łączenie ze sobą dwóch zmiennych znakowych, po prostu łączenie tekstów.

```
zmienne.py x
1 a = "Jan"      # zmienna znakowa = "Jan"
2 b = "Nowak"    # zmienna calkowita = "Nowak"
3 c = a + b      # dodawanie dwóch zmiennych
4 print(c)       # wyswietlenie wyniku "JanNowak"
5
```

– (odejmowanie, zmiana znaku liczby)

Przykład **odejmowania** dwóch zmiennych możemy przedstawić:

```
zmienne.py x
1 a = 100        # zmienna znakowa = 100
2 b = 200        # zmienna calkowita = 200
3 c = a - b      # odejmowanie dwóch liczb 100 / 200
4 print(c)       # wyswietlenie wyniku -100
5
6
```

Mamy tu jeszcze operator **zmiany znaku liczby**. Można to wyjaśnić tak, że jest operator, który przy definiowaniu wartości zmiennej lub podawaniu wartości zmiennej wpisujemy przed liczbą tworząc zmienną o wartości ujemnej. Na przykład *zmienna* = -2.

/ (dzielenie)

Operator dzielenia to jest taki operator, który wykonuje dzielenie rzeczywiste, oznacza to, że wynikiem takiego działania będzie liczba rzeczywista (float).

```
zmienne.py x
1 a = 100        # zmienna znakowa = 100
2 b = 230        # zmienna calkowita = 230
3 c = a / b      # dzielenie dwóch liczb 100 / 230
4 print(c)       # wyswietlenie wyniku 0.43478260869565216
5
```

* (mnożenie)

Operator mnożenia ... co można tu napisać, podam tylko przykład:

```
zmienne.py x
1 a = 23      # zmienna znakowa = 23
2 b = 58      # zmienna całkowita = 58
3 c = a * b   # iloczyn dwóch liczb 23 * 58
4 print(c)    # wyświetlenie wyniku 1334
5
```

% (modulo)

Po pierwsze jak działa **modulo**? Jest to inaczej reszta z dzielenia. W naszym przypadku $a = 123$ $b = 58$ a wynikiem modulo jest 7. Dlaczego? Wystarczy odpowiedzieć sobie na pytanie, ile liczb 58 mieści się w 123, a reszta, która pozostała jest wynikiem działania modulo. Gdy już wiemy, ile razy mieści się dana liczba w drugiej, w naszym przypadku wystarczy przemnożyć 58 przez ilość występowania w 123, czyli 2, a wynik odjąć od 123, to co pozostało jest właśnie wynikiem modulo.

```
zmienne.py x
1 a = 123     # zmienna znakowa = 123
2 b = 58      # zmienna całkowita = 58
3 c = a % b   # reszta z dzielenia dwóch liczb 123 % 58
4 print(c)    # wyświetlenie wyniku 7
5
```

** (potęgowanie)

Używając dwóch znaków mnożenia otrzymujemy symbol potęgowania.

```
zmienne.py x
1 a = 3       # zmienna całkowita = 3
2 b = 4       # zmienna całkowita = 4
3 c = a ** b  # potęgowanie liczb 3 ** 4
4 print(c)    # wyświetlenie wyniku 81
5
```

// (dzielenie z zaokrągleniem w dół, do liczby całkowitej)

Operator dzielenia całkowitego lub jak powinno się nazywać operator dzielenia z zaokrągleniem w dół służy do zwracanie wyniku jako liczby zaokrąglonej w dół do całości. Przykład poniżej pokazuje działanie tego operatora.

```
zmienne.py x
1  a = 90      # zmienna całkowita = 90
2  b = 25      # zmienna całkowita = 25
3  c = a // b  # dzielenie z zaokrągleniem do liczby całkowitej 25 i 90
4  print(c)   # wyświetlenie wyniku 3
5
```