

Python

02

Змінні

Оголошення змінної:

a = 5

b = 4.8

c = 'test string'

d = True

Імена змінних та ключові слова

Добре

summ
summ_of_variables
_new_sum
is_payable
__another_variable

Погано

Summ
summOfVariables
New_sum
isPayable
__anotherVariable

Не можна

Sum%m
\$summOfVariable\$
42New_sum
isP*yable
__anotherV@riable

Правила для іменування змінних

Правила для іменування змінних:

- Ім'я змінної повинне починатися з літери або нижнього підкреслення
- Ім'я не може починатися з цифри
- Ім'я змінної може містити лише літери англійського алфавіту, цифри та нижні підкреслення (A-z, 0-9, and _)
- Імена змінних чутливі до регістру (age, Age та AGE три різні змінні)

Угода для іменування змінних:

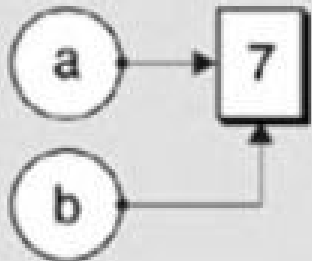
- За наявності кількох слів у змінній як роздільник використовувати нижнє підкреслення `sum_of_cats`
- Не використовувати в змінних літери верхнього регістру: ~~`FirstName = 'Jason'`~~ `first_name = Jason`
- Використовувати усвідомлені імена для змінних: ~~`a = 'Jason Statham'`~~ `fullname = 'Jason Statham'`

Подання змінних у пам'яті

`a = 7`



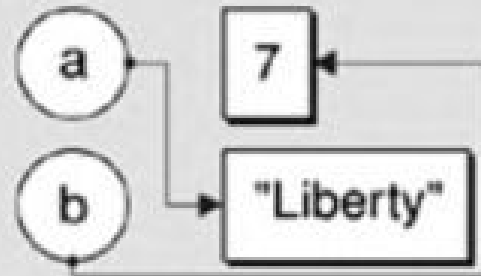
`a = 7`
`b = a`



`a = 7`

`b = a`

`a = "Liberty"`



Основні типи даних

int - цілі числа

float - числа з плаваючою точкою

str - рядки

bool - логічні змінні True чи False

Всі ці типи даних відносяться до незмінних типів даних.

```
>>> print(4)
```

```
>>> type(4)
```

```
>>> print(5.5)
```

```
>>> type(5.5)
```

```
>>> print('Jason Statham')
```

```
>>> type('Jason Statham')
```

```
>>> print(True)
```

```
>>> type(True)
```

Оператори та операнди

Оператори (operators) – спеціальні символи, які представляють обчислення, на кшталт складання та множення. Значення, яких застосовується оператор, називаються операндами (operands).

Оператори та операнди

АРИФМЕТИЧНІ ОПЕРАТОРИ У PYTHON

+ додавання	Складає значення змінних зліва та праворуч від оператора	$10 + 20 = 30$ $20 + -5 = 15$ "hello" + "world" = "helloworld"
- віднімання	Віднімає з лівого операнда правий	$10 - 20 = -10$ $20 - -10 = -30$
* множення	Перемножує лівий та правий операнди	$10 * 20 = 200$ $10 * -20 = -200$ $10 * 5.1 = 51.0$ "hello" * 2 = "hellohello"
/ ділення	Ділить лівий операнд на правий	$20 / 10 = 2.0$ $3 / -10 = -0.3$
% залишок від ділення	Ділить лівий операнд на правий і повертає залишок від розподілу	$20 \% 10 = 0$ $42 \% 10 = 2$
** зведення в ступінь	Зводить лівий операнд у ступінь рівний правому оператору	$5^{**2} = 25$ $-2^{**3^{**2}} = -512$
// цілісний поділ	Ділить лівий операнд на правий і залишає цілу частину.	$9 // 2 = 4$ and $9.0 // 2.0 = 4.0$, $-11 // 3 = -4$ and $-11.0 // 3 = -4.0$

Порядок операцій

Якщо у виразі зустрічається більше, ніж один оператор, то порядок обчислень залежить від правил старшинства (rules of precedence). Для математичних операцій, Python слідує математичним угодам. Абревіатура PEMDAS є простим способом для запам'ятовування правил:

- P**arentheses - Дужки
- E**xponentiation - Зведення в ступінь
- M**ultiplication - Множення
- D**ivision - Ділення
- A**ddition - Додавання
- S**ubtraction - Віднімання

Порядок операцій

Дужки (Parentheses) мають найвищий пріоритет і можуть використовуватися для примусового визначення порядку обчислень у виразі. Таким чином, результат виразу $2*(3-1)$ дорівнюватиме 4, $(1+1)**(5-2)$ дорівнюватиме 8. Ви також можете використовувати дужки для спрощення читання виразів, наприклад, $(minute*100) / 60$, якщо це не вплине на результат.

Зведення в ступінь (Exponentiation) має найбільший пріоритет, так $2**1+1$ дорівнює 3, а не 4 і $3*1**3$ дорівнює 3, а не 27.

Множення та поділ (Multiplication and Division) мають однаковий пріоритет, який вищий за додавання та віднімання (Addition and Subtraction), які також мають однаковий пріоритет. Таким чином, $2*3-1$ дорівнює 5, а не 4 і $6+4/2$ дорівнює 8, а не 5.

Оператори з однаковим пріоритетом обчислюються зліва направо. Таким чином, $5-3-1$ дорівнює 1, а не 3. За винятком зведення в ступінь, у такому випадку вирази обчислюються праворуч наліво:
 $3**2**3 = 3**(2**3)$

Оновлення змінних ($x=x+1$)

1) змінні можуть бути використані для встановлення значень інших змінних

```
my_age = 17
```

```
alisa_age = my_age + 4
```

```
fred_age = alisa_age + 1
```

2) "оновлення самої себе"

```
my_age = 10
```

```
my_age = my_age + 5
```

```
my_age = my_age + 1
```

```
my_age += 1
```