

Python 2

Zmienne:

W Pythonie dane są obiektami. Dane takie jak liczby całkowite, zmiennoprzecinkowe, tekstowe, czy funkcje – również są obiektami.

Obiekt w Pythonie ma właściwości takie jak:

typ – definiuje operacje jakie można wykonać na obiekcie.

Identyfikator – odróżnia go od innych obiektów. Identyfikator jest unikatowy. Jest to adres w pamięci.

Wartość – odpowiednią do jego typu

licznik odwołań – ile razy obiekt jest użyty.

Podstawowe typy danych:

Nazwa	Typ	Mutowalny?	Przykłady
Wartość logiczna	bool	Nie	True, False
Liczba całkowita	int	Nie	47, 25000, 25_000
Liczba zmiennoprzecinkowa	float	Nie	3.14, 2.7e5
Liczba zespolona	complex	Nie	3j, 5 + 9j
Tekst	str	Nie	'raz', 'dwa', 'trzy i cztery'
Lista	list	Tak	['alfa', 'beta', 'gamma']
Krotka	tuple	Nie	(2, 4, 8)
Bajty	bytes	Nie	b'ab\xff'
Macierz bajtów	bytearray	Tak	bytearray(...)
Zbiór	set	Tak	set([3, 5, 7])
Zamrożony zbiór	frozenset	Nie	frozenset(['Adam', 'Ewa'])
Słownik	dict	Tak	{ 'imię': 'Jan', 'nazwisko': 'Nowak', 'obywatelstwo': 'polskie' }

Figura 1: Podstawowe typy danych

Przypisanie zmiennych:

Przypisanie do zmiennej oraz jej wyświetlenie prezentuje przykład:

```
pageurl = 'https://github.com'
print(pageurl)
calkowita, zmiennoprzecinkowa, logiczna = 10, 10.0, False
```

Nazwy zmiennych:

- mogą zawierać litery, cyfry, i znaki podkreślenia – nie mogą zaczynać się od cyfry.
- nie mogą zawierać spacji
- nie mogą być słowami kluczowymi
- powinny być krótkie, ale czytelne

```
>>> help('keywords')

Here is a list of the Python keywords. Enter any keyword to get more help.

False      class      from       or
None       continue  global     pass
True       def       if         raise
and        del       import     return
as         elif      in         try
assert     else      is         while
async      except    lambda     with
await      finally  nonlocal   yield
break      for       not
```

Figura 2: Słowa kluczowe

Zmienne w językach typowanych statycznie takich jak java zmienną można było traktować jako pudełko – miejsce w pamięci do przechowywania wartości.

Zmienna w Pythonie to **etykieta** – nazwa, a nie miejsce w pamięci. Można ją sobie wyobrazić jako etykietę przyczepioną do pudełka (obiektu)

W językach typowanych statycznie zmienna ma określony typ, oraz jest związana z miejscem w pamięci. Wartość tej zmiennej może być modyfikowana o wartości tylko jeżeli są takiego samego typu.

W Pythonie – zmienna może odnosić się do wszystkiego – można to sobie wyobrazić jako odwiązanie etykiety i przyczepienie jej do innego pudełka.

Funkcja `id()` pobiera obiekt i zwraca jego tożsamość, czyli adres pamięci wyrażony jako liczba całkowita

```
>>> a = 300
>>> id(a)
2962110325456
>>> a = "tekst"
>>> id(a)
2962109954544
>>> a = 300
>>> id(a)
2962110325456
```

Utworzenie obiektu z wartością 300 i przypisanie do niego etykiety a.

Utworzenie obiektu z wartością „tekst” oraz przypisanie etykiety a.

Przypisanie z powrotem do wartości 300 etykiety a.

Ciekawostka – tworząc 2 zmienne o wartości 300 mamy 2 obiekty z taką wartością.

Dla małych liczby Interpreter tworzy pojedyncze obiekty i różne zmienne o tych samych wartościach wskazują na ten sam obiekt (-5,256)

```
>>> a = 300
>>> b = 300
>>> id(a)
2962110325456
>>> id(b)
2962110329168
>>> c = 25
>>> d = 25
>>> id(c)
140735446447656
>>> id(d)
140735446447656
```

Aby policzyć liczbę referencji do danego obiektu możemy użyć:

```

>>> import sys
>>> import ctypes
>>> a=b=c="tekst"
>>> sys.getrefcount(a)
4
>>> ctypes.c_long.from_address(id(a)).value
3

```

Uwaga! W pierwszym przypadku zostaje utworzona dodatkowa referencja aby policzyć liczbę referencji do obiektu(+1).

Aby sprawdzić z jakiego typu zmiennymi mamy do czynienia możemy użyć funkcji **type**:

```

>>> a = "czy a to liczba?"
>>> b = 123
>>> c = 123.5
>>> d = True
>>> type(a)
<class 'str'>
>>> type(b)
<class 'int'>
>>> type(c)
<class 'float'>
>>> type(d)
<class 'bool'>

```

Typ logiczny może przyjmować 2 wartości : True i False

Typ całkowity liczby takie jak -100, 200, 1000

Typ zmiennoprzecinkowy – np. 100.0, zawiera separator dziesiętny, lub wykładnik (np. 10e2)

Literał jest dowolnym ciągiem cyfr, można też rozdzielić je znakiem podkreślenia:

```

>>> 0
0
>>> 1000
1000
>>> 1_000_000
1000000
>>> #ale nie:
>>> 05
File "<stdin>", line 1
    05
    ^

```

Wartości logiczne mogą zostać uzyskane z innych typów zmiennych:

```
>>> bool(1)
True
>>> bool("sdgf")
True
>>> bool(0)
False
>>> bool(0.0)
False
>>> bool(0.5)
True
```

Za pomocą funkcji int oraz float można dokonać też konwersję na typy całkowitoliczbowe i zmiennoprzecinkowe:

```
>>> a="32.6"
>>> a="36.6"
>>> b=float(a)
>>> b
36.6
>>> a=int(b)
>>> a
36
>>> a=round(b)
>>> a
37
>>> round(b,3)
36.6
```

Operacje jakie możemy wykonywać na liczbach prezentuje tabela:

Operator	Opis	Przykład	Wynik
+	Dodawanie	5 + 8	13
-	Odejmowanie	90 - 10	80
*	Mnożenie	4 * 7	28
/	Dzielenie	7/2	3.5
	zmiennoprzecinkowe		
//	Dzielenie całkowite	7//2	3
%	Dzielenie z reszta	7%3	1
**	Potęgowanie	3 ** 4	81

```
>>> 5 + 8
13
>>> 90 - 10
80
>>> 4 * 7
28
>>> 7 / 2
3.5
>>> 7 // 2
3
>>> 7 % 3
1
>>> 3 ** 4
81
```

Ciąg tekstowy może być umieszczony w apostrofach jak i cudzysłowach:

```
>>> napis = "lore ipsum..."
>>> napis
'lore ipsum...'
>>> napis = 'lore ipsum...'
>>> 'cytat:"lore ipsum"'
'cytat:"lore ipsum"'
>>> napis = 'cytat:"lore ipsum"'
>>> napis
'cytat:"lore ipsum"'
```

Wartości zmiennych możemy przyrównywać za pomocą operatorów porównania:

równe	==
różne	!=
mniej niż	<
mniej lub równe niż	<=
więcej niż	>
więcej lub równe niż	>=

```
>>> 5 > 6
False
>>> 5 >= 6
False
>>> 5 <= 6
True
>>> 5 == 6
False
>>> 5 == 5
True
>>> "napis" == "napis"
True
>>> "napis" != "napis"
False
```

Wyniki porównania zmiennych mogą zostać użyte w testach warunkowych. Do podejmowania decyzji użyjemy słów kluczowych **if**, **else**, oraz **elif**:

```
>>> a = 1
>>> b = 100
>>> if a < b :
...     print('a mniejsze od b')
... else :
...     print('a większe bądź równe b')
...
a mniejsze od b
```

```
>>> a = 1
>>> b = 1
>>> if a < b :
...     print("a mniejsze od b")
... elif a > b :
...     print("a większe od b")
... else :
...     print("a równe b")
...
a równe b
```

Należy pamiętać, że blok kodu jest wyznaczany przez wcięcia!!!! (użyłem 4 spacje)

Wielkość znaków przy porównaniu ma znaczenie:

```
>>> a = "napis"
>>> b = "NAPIS"
>>> a == b
False
```

Czy można użyć innych operatorów do porównania łańcuchów znaków?

```
>>> ord("n")
110
>>> ord("N")
78
>>> chr(110)
'n'
>>> chr(78)
'N'
>>> "n">"N"
True
>>> "napis" > "NAPIS"
True
```

Złączanie łańcuchów tekstowych:

```
>>> napis = input("podaj marke telefonu")
podaj marke telefonugoogle
>>> model = input("podaj model telefonu ")
podaj model telefonu pixel7
>>> napis+model
'googlepixel7'
>>> napis + " " + model
'google pixel7'
```

Za pomocą ciągu tekstowego f-string można wstawić wartości zmiennych w łańcuch znaków:

```
>>> f'Marka twojego telefonu to {napis}, model {model}, wygląda na to że to najnowszy model od producenta {napis}.'
'Marka twojego telefonu to google, model pixel7, wygląda na to że to najnowszy model od producenta google.'
```

Wielkość liter może zostać zmieniona za pomocą metod **upper()** lub **lower()**, jeżeli chcemy zamienić pierwszą literę na wielką użyjemy metody **title()**:

```
>>> telefon = napis + " " + model
>>> telefon.upper()
'GOOGLE PIXEL7'
>>> telefon.lower()
'google pixel7'
>>> telefon.title()
'Google Pixel7'
>>> telefon.upper().lower()
'google pixel7'
```

Za pomocą znaków `\n` oraz `\t` można dodać koniec wiersza oraz tabulator:

```
>>> print(f"{napis}\n{model}")
google
pixel7
```

Zadania:

1. Utwórz 3 zmienne wskazujące na ten sam obiekt – napis „Python 2023”.
 - a. Wyświetl wartość logiczną zwróconą przez porównanie napisów ze zmiennej pierwszej i drugiej oraz drugiej i trzeciej.
 - b. Wyświetl typy tych zmiennych oraz ich adres w pamięci (w postaci szesnastkowej – funkcja `hex()`)Pod trzecią zmienną przypisz napis „Java 11”
Ponownie wykonaj podpunkt a i b
2. Napisz skrypt kalkulator, który pobierze od użytkownika 2 liczby a następnie po podaniu odpowiedniego operatora wykona adekwatną operację i wyświetli wynik.
3. Utwórz skrypt, która pobierze od użytkownika odpowiedzi na pytania – ankieta z pytaniami jak w przykładzie (pytania 1-7) oraz pytaniem o imię i nazwisko:
<https://www.webankieta.pl/wzor-ankiety/ankieta-czytelnicza/>
Wyświetli pytania wraz z odpowiedziami w formacie
pytanie: Jak masz na imię oraz nazwisko?
odpowiedź: Adam Adamski
pytanie: W jakich okolicznościach czytasz książki najczęściej?
odpowiedź: Podczas pracy/nauki (to ich element)