

Typy, zmienne, operacje logiczne

W Pythonie dostępne są następujące typy danych:

- - Liczby – zmiennoprzecinkowe (*float*) i całkowite (*int*).
- - Teksty (*string*)- Tekst można zapisywać przy użyciu pojedynczego i podwójnego cudzysłowu. Wszystkie znaki są wyświetlane dokładnie tak jak zostały zapisane (w tym białe znaki).
- - Typ logiczny (*bool*)- przechowuje wartości prawda/fałsz
- - Listy (*list*)- złożony typ danych przechowujący elementy oddzielone przecinkiem. Wartości nie muszą być tego samego typu.

```
>>> List = ["mama", 5, True, 3]
>>> List
['mama', 5, True, 3]
>>> _
```

Elementy w liście numerowane są od 0.

```
>>> List[0] = 'tata'
>>> List
['tata', 5, True, 3]
>>>
```

- - *Tuple*- listy które nie są modyfikowalne. Stosuje się okrągłe nawiasy do zapisu

```
>>> Krotka = ('mama', 5, True, 3)
>>> Krotka[0] = 'tata'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
```

Nazwy zmiennych w Pythonie muszą rozpoczynać się od litery. Nazwy są wrażliwe na wielkość liter, zmienna *Dom* i *dom* są różnymi obiektami.

Python pozwala na porównania:

- < - mniejsze niż
- > - większe niż
- <= - mniejsze bądź równe niż
- >= - większe niż równe niż
- == - równe
- != - różne niż

```
>>> 1 < 2
True
>>> 1 > 2
False
>>> 1 <= 1
True
>>> 1 >= 1
True
>>> 1 == 2
False
>>> 1 != 2
True
>>>
```

Do łączenia warunków wykorzystuje się:

- and – koniunkcja, oba warunki muszą być prawdziwe
- or – suma, jeden z warunków musi być spełniony

```
>>> 1 == 1 and 1 == 2
False
>>> 1 == 1 or 1 == 2
True
>>> (1 == 1 and 1 == 2) or 2 == 2
True
>>> _
```

Dodatkowo do porównań można użyć

- is – sprawdza czy wartości są identyczne
- in – sprawdza czy zmienna jest zawarta w innym obiekcie
- not – dodaje zaprzeczenie

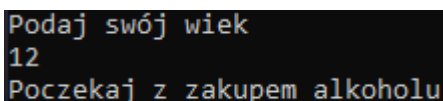
```
>>> List = ["mama", 5, True, 3]
>>> "tata" in List
False
>>> "mama" in List
True
>>> "aaa" is "aaa"
True
>>>
```

Aby sterować algorytmem wykorzystuje się instrukcję warunkową *if*. Sprawdza ona czy dane wyrażenie warunkowe jest prawdziwe i wykonuje odpowiedni fragment kodu. Zapis w funkcji wygląda następująco:

```
if (warunek):  
    polecenie1  
else:  
    polecenie2
```

Poniższy kod da następujący efekt:

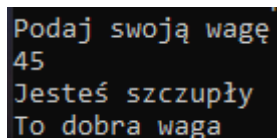
```
print("Podaj swój wiek")  
wiek = int(input())  
if(wiek >= 18):  
    print("Jesteś pełnoletni")  
else:  
    print("Poczekaj z zakupem alkoholu")
```



```
Podaj swój wiek  
12  
Poczekaj z zakupem alkoholu
```

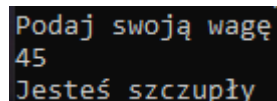
Jeśli program powinien sprawdzać więcej niż jeden blok kodu można użyć instrukcji ponownie słowa *if* lub zastosować *elif*. Przy zastosowaniu *elif* gdy któryś warunek zostanie spełniony to następne nie są sprawdzane.

```
print("Podaj swoją wagę")  
waga = int(input())  
if(waga <= 50):  
    print("Jesteś szczupły")  
if(waga <= 100):  
    print("To dobra waga")  
else:  
    print("Chyba się pomyliłeś")
```



```
Podaj swoją wagę  
45  
Jesteś szczupły  
To dobra waga
```

```
print("Podaj swoją wagę")  
waga = int(input())  
if(waga <= 50):  
    print("Jesteś szczupły")  
elif(waga <= 100):  
    print("To dobra waga")  
else:  
    print("Chyba się pomyliłeś")
```



```
Podaj swoją wagę  
45  
Jesteś szczupły
```

Poniższy przykładowy program pobiera od użytkownika 3 liczby i sortuje je od największej do najmniejszej. Proszę go odtworzyć i sprawdzić czy wykonuje się poprawnie.

```
#pobieram wartości od użytkownika i przypisuje je do zmiennych
print("Podaj pierwszą liczbę:")
a = float(input())

print("Podaj drugą liczbę:")
b = float(input())

print("Podaj trzecią liczbę:")
c = float(input())

# wykonuje szereg porównań aby znaleźć kolejność
if (a > b):
    if(c > a):
        lista = [c, a, b]
    else:
        if(c > b):
            lista = [a, c, b]
        else:
            lista = [a, b, c]
else:
    if(c > b):
        lista = [c, b, a]
    else:
        if(c > a):
            lista = [b, c, a]
        else:
            lista = [b, a, c]

#Wyświetlam wynik wpisany do listy
print("Liczby posortowane w kolejności malejącej:")
print(lista[0])
print(lista[1])
print(lista[2])
```

Zadanie 1

Napisz program który pobiera od użytkownika 3 liczby które są docelowymi długościami boków trójkąta. Sprawdź czy z tych liczb można zbudować trójkąt. Wyświetl stosowny komunikat w zależności od długości boków wprowadzonych przez użytkownika.

Zadanie 2

Napisz program, kalkulator BMI. Program pobiera od użytkownika dane i wyświetla wynik końcowy.

Wzór na BMI:

$$\text{BMI} = \frac{\text{waga}}{\text{wzrost}^2}$$

Zadanie 3

Napisz program, który posortuje liczby 1,5,3 w kolejności rosnącej. Wyświetl wynik.

Zadanie 4

Napisz program, który pobierze od użytkownika 3 cyfry, posortuje je rosnąco i wyświetli wynik.