



Materialy Informatyka 4

Opracował: mgr inż. Woźniak Piotr

1. Zmienne języka Python pozwalają na przechowywanie różnego typu danych np. całkowitego, zmiennoprzecinkowego, tekstowego. Rodzaj typu zależy od przypisanej wartości. Możliwe jest sprawdzenie typu funkcją **type**.

```
zmienna1 = 'Podaj liczbę'      #zmienna przechowująca typ tekstowy
zmienna2 = -10                 #zmienna przechowująca typ liczby całkowitej
print(type(zmienna1))          #wyświetlenie typu zmiennej
```

2. Należy uważać przy operowaniu na zmiennych w kwestii typów np.

```
print(5.1 + 5) #dodawanie różnego typu zmiennych
print('tekst' + 5) #dodawanie różnego typu zmiennych
print('tekst' - 'tekst2') #dodawanie różnego typu zmiennych
```

Operowanie na różnych typach może doprowadzić do błędów krytycznych lub logicznych.

3. W celu wymuszenia określonego zachowania należy wymusić potraktowanie zmiennej jako określony typ (rzutowanie) np.

```
zmienna1 = '100'
print(zmienna1 + 20) #błąd dodawanie tekstu i liczby
print(int(zmienna1) + 20) #poprawne dodanie po rzutowaniu
```

Operowanie na różnych typach może doprowadzić do błędów krytycznych lub logicznych.

4. Typ tekstowy (string) zbudowany jest z łańcucha znaków, czyli: 'tata' = 't', 'a', 't', 'a'. Pozwala to operować na typie tekstowym z pomocą operatora nawiasu kwadratowego [], gdzie wewnątrz określa się indeks (pozycję) elementu.

```
zmienna1 = 'abcdefg'
print(zmienna1[0]) #wyświetlenie 1 elementu łańcucha
print(zmienna1[1]) #wyświetlenie 2 elementu łańcucha
print(zmienna1[4:6]) #wyświetlenie elementów od 5 do 6
```

Indeksowanie elementów w języku Python rozpoczyna się od **0**, czyli pierwszy element ma indeks zerowy.

5. Istotnym elementem operowania na łańcuchach tekstowych jest określenie ilości znaków. Używa się do tego funkcji **len** od length (ang. długość).

```
print(len('tata')) #wyświetlenie długości łańcucha
```

6. Operowanie na tekście ułatwia wykorzystanie dodatkowych funkcji języka Python m.in.

```
a = "Hello, World!"  
print(a.lower()) #zamiana wielkich liter na małe  
print(a.upper()) #zamiana małych liter na wielkie  
print(a.replace("H", "W")) #zamiana fragmentów tekstu na inny  
print(a.split(",")) #dzielenie łańcuchów na fragmenty
```