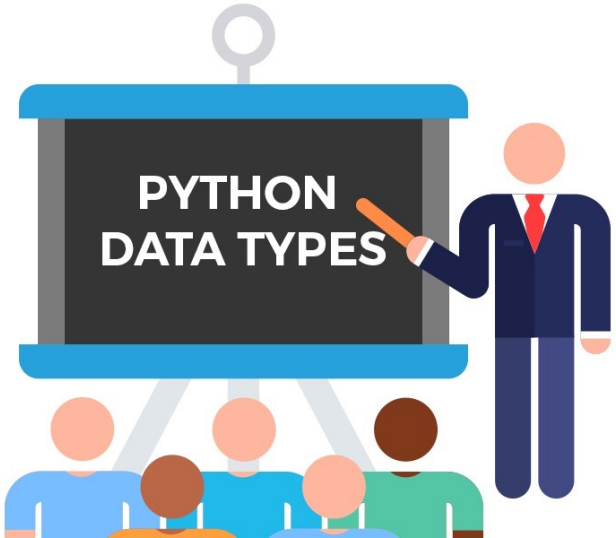
Literały

Podstawy programowania w języku Python



Co to są literały?

- zapisy reprezentujące pewne ustalone wartości w kodzie
- występują w różnych typach

```
1    print("Hello World!")
2    PI = 3.14
```

Literał to wartość zapisana bezpośrednio w kodzie. Z wartości tej można wywnioskować jakiego jest typu.

Podstawowe typy literałów

- liczby całkowite (`int`)
- liczby zmiennoprzecinkowe (`float`)
- ciągi znaków (`str`)
- wartości boolowskie (`bool`)

```
1    print("Hello World!")
2    x = 4
3    condition = True
```

Liczby całkowite

- ciąg cyfr, które składają się na liczbę
- liczby zapisane bez elementu ułamkowego
- liczby ujemne zapisujemy korzystając z operatora ”–”
- w celu poprawy czytelności można stosować podkreślenia

```
1    123
2    23423423423
3    1_000_000
4    -2
5    +99
```

Literały a systemy liczbowe

Liczbę całkowitą możemy przedstawić za pomocą różnych systemów liczbowych

- dwójkowy – prefiks `0b` lub `0B`
- ósemkowy – prefiks `0o` lub `0O`
- szesnastkowy – prefiks `0x` lub `0X`

```
1    0b1101
2    0o177
3    0xFF
4    print(0xFF)
```

```
1    print(0b1101) #literał w systemie dwójkowym
```

Liczby zmiennoprzecinkowe

```
1    4.0
2    print(123.345)
3    x = .567
4    y = 2.
5    3.1954e-5
```

- mają (lub mogą mieć) ułamkową część po kropce dziesiętnej
- do oddzielenia części ułamkowej stosujemy znak kropki ”.”
- w zapisie możemy pomijać 0, gdy jest to jedyna cyfra przed kropką dziesiętną lub po niej
- dla poprawy czytelności bardzo dużych lub bardzo małych liczb stosuję się notację wykładniczą (naukową)

Ciągi znaków

```
1  print("Ala ma kota, a kot ma Alę.")
2  'To także jest ciąg znaków'
3  "tekst podzielony\nznakiem nowe linii"
4  print("I'm Monty Python.")
```

- są używane do przetwarzania tekstu
- ciągi znaków zapisujemy korzystając z cudzysłowów lub apostrofów
- ciąg znaków może być pusty
- do wprowadzania znaków o szczególnym znaczeniu używamy znaku ukośnika wstecznego (backslash) ”\”

Wartości boolowskie

- reprezentują tylko dwie wartości: prawda lub fałsz
- zapisujemy je jako `True` lub `False`
- większość wartości nie pustych jest traktowana jako `True`
- `0` jest traktowane jako `False`

```
1    x = True
2    print(False)
3    bool(1)
4    bool(2 > 4)
```

Pytanie

Poniżej zapisano literały reprezentujące:

```
1    "123.0", '0.123e5', "0.1"
```

a) liczby całkowite
b) liczby zmiennoprzecinkowe
c) ciągi znaków
d) wartości logiczne

Odpowiedź: c)

Pytanie

Wskaż niepoprany zapis ciągu znaków:

a) `"I'm Monty Python"`
b) `'I'm Monty Python'`
c) `'\\'`
d) `"'\n'"`

Odpowiedź: b)

Pytanie

Wskaż typ oraz wartość poniższego literału:

```
1    0e-9
```

a) ciąg znaków 0e-9
b) liczba całkowita -9
c) liczba całkowita 0
d) liczba zmiennoprzecinkowa 0.0

Odpowiedź: d)