

Operatory

Matematyczne

+, -, *, /,
np. 3 + 4
+=, -=, *=, /=
np. x += 3
(wartość x zwiększona o 3)
** potęgowanie
np. 3**2
// - dzielenie całkowite
% - reszta z dzielenia

Porównania

<, >, <=, >=
==, !=, is, is not

Logiczne

and np. 1 and True
or np. 0 or 5
not np. not False

Typy zmiennych

int - liczba całkowita np. 88
float - liczba zmiennoprzecinkowa np. 54.221
complex - liczba zespolona np. 4 + 5i
bool - typ logiczny True/False
str - łańcuch znaków np. 'Ala ma kota'
list - lista elementów (mutowalna) np. [1, 3, 'a']
tuple - krotka (niemutowalna) np. (3, 4, 'b')
dict - słownik np. {'wiek': 34, 'imie': 'Tomasz'}
set - zbiory - zawierają elementy bez powtórzeń
np. {3, 4, 7}
None - wartość specjalna

Konwersja typów

float('5.45') int('444')
str(2) bool('False')

Sprawdzanie typu:

type('a')

Sprawdzanie czy jest instancją danej klasy:

isinstance('a', str)

Zakresy

range(5) -> [0, 1, 2, 3, 4] range(1, 5) -> [1, 2, 3, 4]
range(1, 9, 2) -> [1, 3, 5, 7] range(9, 3, -2) -> [9, 7, 5]

Pętle i instrukcje sterujące

While

licznik = 1
while licznik < 7:
 print(licznik)
 licznik += 1

For

for element in (1, 2, 3, 5):
 print(element)

Sterowanie

conuter = 0
while True:
 counter += 1
 if counter % 2 == 0:
 continue
 print(counter)
 if counter > 30:
 break

Stringi

Formatowanie

x = 'Ala ma kota'
x = 'Ala ma %s kotów' % 3
x = 'Ala ma %s kotów i %.2f zabawek' % (4, 1.5)
x = 'Ala ma {} kotów i {} psów'.format(1, 2)
x = 'Ala ma {1} kotów i {0} psów'.format(2, 1)
data = {'pies': 2, 'kot': 1}
x = 'Ala ma {kot} kotów i {pies} psów'.format(**data)
pies=2; kot=1
x = f'Ala ma {pies} psów i {kot} kotów'

Zarządzanie wielkością liter

str.capitalize()
str.lower()
str.upper()
str.swapcase()
str.title()

Pozycjonowanie tekstu

str.ljust(width[, fillchar])
str.center(width[, fillchar])
str.rjust(width[, fillchar])
str.zfill(width)

Znaki specjalne, surowy tekst

\t - znak tabulacji
\n - przejście do następnej linii - Unix / Linux
\r\n - przejście do następnej linii - Windows
surowy tekst, np. adres url:
r'c:\programy\program1.exe'

Wyszukiwanie

str.find(sub[, start[, end]])
str.rfind(sub[, start[, end]])
str.index(sub[, start[, end]])
str.rindex(sub[, start[, end]])

Usuwanie znaków z początku/końca

str.lstrip([chars])
str.strip([chars])
str.rstrip([chars])

Podział

str.split(sep=None, maxsplit=-1)
str.rsplit(sep=None, maxsplit=-1)
str.splitlines([keepends])
str.partition(sep)
str.rpartition(sep)

Łączenie

str.join(iterable)

Zamiana

str.replace(old, new[, count])
str.translate(table)

Metody informacyjne

str.isalnum() str.isalpha() str.isdecimal()
str.isdigit() str.isnumeric() str.isidentifier()
str.islower() str.isupper() str.isprintable()
str.isspace() str.istitle()
str.startswith(prefix[, start[, end]])
str.endswith(suffix[, start[, end]])