#### Praca ze zmiennymi

Możemy przypisać wartość do zmiennej za pomocą pojedynczego znaku = i nie musimy (ani nie możemy) określić typu zmiennej.

>>> my\_str = "This is a simple string"

Drukowanie zmiennej za pomocą print()

>>> print(my\_str)

This is a simple string

String jest nie zmienny ale możemy utworzyć nowy string

>>> my\_str += " testing"

>>> my\_str

'This is a simple string testing'

To nie zmieniło ciągu; ponownie przypisano zmienną. Oryginalny ciąg "To jest prosty ciąg" pozostał niezmieniony.

Ważne jest, aby zrozumieć, że zawartość zmiennej można zmienić i nie musimy utrzymywać tego samego typu:

>>> my\_str = 1

>>> print(my\_str)

1

Najlepiej byłoby, gdybyśmy nie zmienili zawartości zmiennej o nazwie my\_str jako int, ale jest to coś, co pyton mógłby użyć.

Ostatnią rzeczą do zapamiętania jest to, że jeśli przypiszemy zmienną z inną zmienną, zostanie ona przypisana do wyniku zmiennej, a nie do tego, co zmienna wskazuje później.

>>> my\_str = 1

>>> my\_int = my\_str

>>> my\_str = "testing"

>>> print(my\_int)

1

>>> print(my\_str)

testing

**Strings**

Pracowaliśmy już z ciągiem znaków, kiedy stworzyliśmy program "Hello, World!". Tworzymy łańcuchy za pomocą pojedynczych cudzysłowów ('), podwójnych cudzysłowów (") lub potrójnych pojedynczych lub podwójnych cudzysłowów dla ciągu wielowierszowego:

>>> 'single quoted string'

'single quoted string'

>>> "double quoted string"

'double quoted string'

>>> '''

... this is a triple

... quoted string

... '''

'\nthis is a triple\nquoted string\n'

Strings działają również z niektórymi operatorami arytmetycznymi.

Możemy łączyć ciągi używając operatora + i mnożyć łańcuch przez liczbę używając operatora \*:

>>> "pass" + "tor"

'passtor'

>>> "Ha" \* 4

'HaHaHaHa'

Ciąg to sekwencja zgrupowanych razem znaków. Zanim przejdziemy dalej, musimy objąć koncepcję "obiektu" w programowaniu obiektowym.

"Obiekt" zawiera dwie rzeczy:

1. Stan

2. Zachowanie

W przypadku wbudowanych typów stan ma sens, ponieważ jest to cała zawartość obiektu. Aspekt zachowania oznacza, że istnieją funkcje, które możemy wywołać w instancjach obiektów, które mamy. Funkcja związana z obiektem nazywana jest "metodą". Oto kilka przykładowych metod, które możemy wywołać na ciągach:

find lokalizuje pierwsze wystąpienie znaku (lub łańcucha) w ciągu znaków. Ta funkcja zwraca indeks znaku lub łańcucha znaków:

>>> "double".find('s')

-1

>>> "double".find('u')

2

>>> "double".find('bl')

3

lower konwertuje wszystkie znaki w łańcuchu na ich małe litery (jeśli je posiadają). Ta funkcja zwraca nowy ciąg bez zmiany oryginału, a to staje się ważne później:

>>> "TeStInG".lower() # "testing"

'testing'

>>> "another".lower()

'another'

>>> "PassWord123".lower()

'password123'

Wreszcie, jeśli potrzebujemy użyć cudzysłowów lub znaków specjalnych w ciągu znaków,

>>> print("Tab\tDelimited")

Tab Delimited

>>> print("New\nLine")

New

Line

>>> print("Slash\\Character")

Slash\Character

>>> print("'Single' in Double")

'Single' in Double

>>> print('"Double" in Single')

"Double" in Single

>>> print("\"Double\" in Double")

"Double" in Double