**Dictionaries**

Słowniki są głównym typem mapowania, którego będziemy używać w Pythonie. Ten obiekt jest porównywalny z hashem lub „tablicą asocjacyjną” w innych językach.

Rzeczy do odnotowania na temat słowników:

1. W przeciwieństwie do słowników Pythona 2, tak jak w Pythonie 3.6, klucze są zamawiane w słownikach. Będziesz potrzebował OrderedDict, jeśli chcesz, aby działało to w innej wersji Pythona.

2. Możesz ustawić klucz dla dowolnego IMMUTABLE TYPE (bez list).

3. Unikaj używania jako kluczy rzeczy innych niż proste przedmioty.

4. Każdy klucz może mieć tylko jedną wartość (więc nie twórz duplikatów podczas tworzenia dyktatu).

Tworzymy literały słownikowe za pomocą nawiasów klamrowych ({i}), oddzielając klucze od wartości za pomocą dwukropków (:) i oddzielając pary klucz / wartość za pomocą przecinków (,). Oto przykładowy słownik:

>>> ages = { 'kevin': 59, 'alex': 29, 'bob': 40 }

>>> ages

{'kevin': 59, 'alex': 29, 'bob': 40}

Możemy odczytać wartość ze słownika przez indeksowanie za pomocą klucza:

>>> ages['kevin']

59

>>> ages['billy']

Traceback (most recent call last):

File "", line 1, in

KeyError: 'billy'

Klucze można dodawać lub zmieniać za pomocą subskrypcji i przypisania:

>>> ages['kayla'] = 21

>>> ages

{'kevin': 59, 'alex': 29, 'bob': 40, 'kayla': 21}

Elementy można usunąć ze słownika za pomocą instrukcji del lub za pomocą metody pop:

>>> del ages['kevin']

>>> ages

{'alex': 29, 'bob': 40, 'kayla': 21}

>>> del ages

>>> ages

Traceback (most recent call last):

File "", line 1, in

NameError: name 'ages' is not defined

>>> ages = { 'kevin': 59, 'alex': 29, 'bob': 40 }

>>> ages.pop('alex')

29

>>> ages

{'kevin': 59, 'bob': 40}

Często zdarza się, że chcemy wiedzieć, jakie klucze lub wartości mamy bez dbania o powiązania. W tej sytuacji mamy metody wartości i kluczy:

>>> ages = {'kevin': 59, 'bob': 40}

>>> ages.keys()

dict\_keys(['kevin', 'bob'])

>>> list(ages.keys())

['kevin', 'bob']

>>> ages.values()

dict\_values([59, 40])

>>> list(ages.values())

[59, 40]

**Alternatywne sposoby tworzenia dyktatu za pomocą argumentów słów kluczowych**

**Istnieje kilka innych sposobów tworzenia słowników, które możemy zobaczyć: te, które używają konstruktora dict z argumentami klucz / wartość oraz lista krotek:**

>>> weights = dict(kevin=160, bob=240, kayla=135)

>>> weights

{'kevin': 160, 'bob': 240, 'kayla': 135}

>>> colors = dict([('kevin', 'blue'), ('bob', 'green'), ('kayla', 'red')])

>>> colors

{'kevin': 'blue', 'bob': 'green', 'kayla': 'red'}