

Słowniki w języku Python 3

24-03-2015

Plan:

- *hashable* (przypomnienie)
- *używanie słowników (API)*
- *zadania*

- *hashable*

An object is *hashable* if it has a hash value which never changes during its lifetime (it needs a `__hash__()` method), and can be compared to other objects (it needs an `__eq__()` method).

Hashable objects which compare equal must have the same hash value.

(<https://docs.python.org/3/glossary.html>)

- „hash value” – liczba, identyfikator (w przybliżeniu: zwykle powinno to być coś jak PESEL)

dict

- słownik (*dictionary*), mapa (*map*), mapowanie (*mapping*), tablica asocjacyjna, tablica skojarzeniowa (*associative array*)
- struktura danych, która przechowuje pary:
 - (unikalny klucz, wartość)
- intuicyjnie: to samo co *set*, tylko każdy element ma do pary „ze sobą” jakiś obiekt
- klucze muszą być *hashowalne* i *unikalne*, wartości są dowolne

dict – przykładowe zastosowania

- tytuł książki -> główny autor

```
{'Structure and Interpretation of Computer Programs' : 'Harold Abelson',  
'Introduction to Algorithms' : 'Thomas H. Cormen',  
'The C Programming Language' : 'Brian W. Kernighan',  
'The Pragmatic Programmer: From Journeyman to Master' : 'Andrew Hunt',  
'Art of Computer Programming' : 'Donald Ervin Knuth',  
'Design Patterns: Elements of Reusable Object-Oriented Software' : 'Erich Gamma',  
'Artificial Intelligence: A Modern Approach' : 'Stuart Russell',  
'Introduction to the Theory of Computation' : 'Michael Sipser',  
'Code Complete' : 'Steve McConnell',  
'The Mythical Man-Month: Essays on Software Engineering' : 'Frederick P. Brooks Jr.'}
```

dict – przykładowe zastosowania

- tytuł książki -> numer regału

```
{ 'Structure and Interpretation of Computer Programs'      : 1,  
  'Introduction to Algorithms'                             : 34,  
  'The C Programming Language'                             : 23,  
  'The Pragmatic Programmer: From Journeyman to Master'    : 12,  
  'Art of Computer Programming'                           : 4,  
  'Design Patterns: Elements of Reusable Object-Oriented Software' : 586,  
  'Artificial Intelligence: A Modern Approach'            : 32,  
  'Introduction to the Theory of Computation'              : 98,  
  'Code Complete'                                          : 77,  
  'The Mythical Man-Month: Essays on Software Engineering' : 3}
```

dict – przykładowe zastosowania

- student -> lista ocen

```
{ 'Adam Nowak'           : [5, 5, 1],  
  'Ada Nowakowska'       : [5, 5, 4],  
  'Barbara Kowalska'     : [3, 3, 4],  
  'Bartek Kowalewski'    : [2, 2, 1],  
  'Jędrzej Iksiński'     : [5, 3, 5],  
  'Janina Iksińska'      : [3, 5, 4] }
```

dict – podstawowe operacje

- Tworzenie
- Modyfikowanie
 - Dodawanie elementów
 - Usuwanie elementów
 - Zmiana istniejących elementów
- Dostęp do elementów
- Porównywanie

dict – tworzenie

```
d1 = {'one' : 1, 'two' : 2}
d2 = dict(one = 1, two = 2)
d3 = dict([('one', 1), ('two', 2)])
print(d1)
print(d2)
print(d3)
print(d1 == d2 == d3)
```

dict – dodawanie elementów

```
d = {}  
print(d)  
d['Adam Nowak'] = [5, 4, 3]  
print(d)
```

dict – usuwanie elementów

```
d = { 'Adam Nowak'      : 5,  
      'Ada Nowakowska' : 4}  
print(d)  
del d['Adam Nowak']  
print(d)
```

dict – modyfikowanie elementów

```
d = { 'Adam Nowak'      : 5,  
      'Ada Nowakowska'  : 4}  
print(d)  
d['Adam Nowak'] = 3  
print(d)
```

dict – dostęp do elementów

```
d = { 'Adam Nowak'      : [5],  
      'Ada Nowakowska'  : [4, 5] }  
print(d)  
print(d['Adam Nowak'])  
d['Adam Nowak'].append(4)  
print(d)
```

dict – porównywanie

```
d1 = {'one' : 1, 'two' : 2}
d2 = dict(one = 1, two = 2)
d3 = dict([('one', 1), ('two', 2)])
print(d1)
print(d2)
print(d3)
print(d1 == d2 == d3)
```

dict – porównywanie

- Co będzie dla $<$, $>$, $<=$, $>=$?

dict – wiele innych przydatnych operacji!

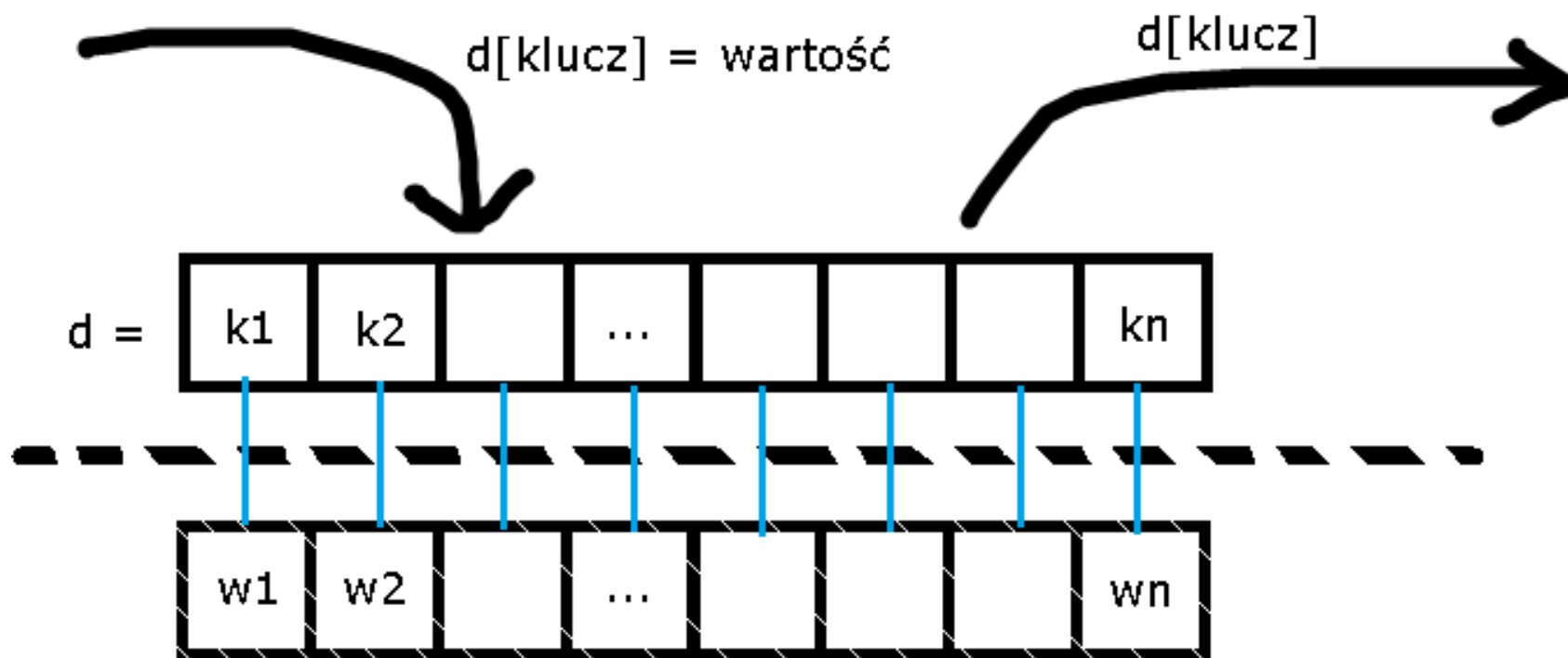
- <https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>

dict – składnia

- Czym się różnią zaznaczone dwukropki?

```
{'Structure and Interpretation of Computer Programs' : 'Harold Abelson',  
'Introduction to Algorithms' : 'Thomas H. Cormen',  
'The C Programming Language' : 'Brian W. Kernighan',  
'The Pragmatic Programmer: From Journeyman to Master' : 'Andrew Hunt',  
'Art of Computer Programming' : 'Donald Ervin Knuth',  
'Design Patterns: Elements of Reusable Object-Oriented Software' : 'Erich Gamma',  
'Artificial Intelligence: A Modern Approach' : 'Stuart Russell',  
'Introduction to the Theory of Computation' : 'Michael Sipser',  
'Code Complete' : 'Steve McConnell',  
'The Mythical Man-Month: Essays on Software Engineering' : 'Frederick P. Brooks Jr.'}
```

dict - podsumowanie



Zadania!

- https://github.com/CodeCarrots/warsztaty_2015