

Słowniki

- W dwóch omówionych dotąd typach sekwencji – listach i krotkach – dostęp do dowolnego elementu możliwy był poprzez podanie jego indeksu.
- W słowniku (ang. dictionary) dostęp do dowolnej wartości przechowywanej w słowniku możliwy jest poprzez podanie klucza do niej.
- Słownik składa się zatem ze zbioru **kluczy** i zbioru **wartości**, gdzie każdemu kluczowi przypisana jest pojedyncza wartość.
- Zależność między kluczem a jego wartością nazywana bywa odwzorowaniem (ang. mapping).
- Klucz nie musi być liczbą, tak jak jest nią indeks, wystarczy, że jest typu niezmiennego.
- Można powiedzieć więc, że o ile lista czy krotka odwzorowuje liczby całkowite (indeksy) na obiekty dowolnego typu, o tyle słownik odwzorowuje obiekty dowolnego typu niezmiennego na obiekty dowolnego typu.

Słowniki

Przykład: tworzenie słownika

```
tel = {"policja":997, "straż":998, "pogotowie":999}  
print(tel)  
#wynik: {'policja': 997, 'pogotowie': 999, 'straż': 998}
```

Kluczami są: policja, straż, pogotowie.

Wartościami są: 997, 998, 999.

Przykład: klucze nie muszą być tekstem, a wartości liczbami

```
bohater = {"hans":"kloss", "james":"bond"}  
print(bohater)  
#wynik: {'hans': 'kloss', 'james': 'bond'}  
ujemne = {7:-7, 3:-3}  
print(ujemne)  
#wynik: {3: -3, 7: -7}
```

Aby otrzymać wartość podanego klucza...

```
print(tel ["policja"])  
#wynik: 997  
print(bohater["james"])  
#wynik: bond  
print(ujemne[3])  
#wynik: -3
```

Zwróćmy uwagę, że `ujemne[3]` nie oznacza trzeciego elementu słownika, lecz taki element słownika, którego klucz stanowi liczba 3.

Przykład: tworzenie słownika przez przypisanie

```
tel = {}  
tel["policja"] = 997  
tel["straż"] = 998  
tel["pogotowie"] = 997  
print(tel)  
#wynik: {'policja': 997, 'pogotowie': 999, 'straż': 998}
```

Słowniki

Aby dodać nowy klucz...

```
tel = {"policja":997, "straż":998, "pogotowie":999}  
tel["taxi"] = 919  
print(tel)  
#wynik: {'taxi': 919, 'policja': 997, 'pogotowie': 999, 'straż': 998}
```

Aby zmodyfikować istniejącą wartość...

```
tel["taxi"] = 9622  
print(["taxi"])  
#wynik: 9622
```

Słowniki

Aby usunąć określoną wartość ze słownika...

```
tel = {"taxi": 919, "policja": 997, "pogotowie": 999, "straż": 998}
del tel["taxi"]
print(tel)
#wynik: {'policja': 997, 'pogotowie': 999, 'straż': 998}
```

Aby odczytać i od razu usunąć określoną wartość ze słownika...

```
tel = {"taxi": 919, "policja": 997, "pogotowie": 999, "straż": 998}
usuniety = tel.pop("taxi")
print(usuniety)
#wynik: 919
print(tel)
#wynik: {'policja': 997, 'pogotowie': 999, 'straż': 998}
```

Słowniki

Aby sprawdzić, czy dany klucz występuje w słowniku...

```
tel = {"policja": 997, "straż": 998, "pogotowie": 999}
print("taxi" in tel)
#wynik: False
print("straż" in tel)
#wynik: True
```

Aby zaktualizować słownik w oparciu o inny słownik...

```
tel = {"policja": 997, "straż": 998}
tel2 = {"pogotowie": 999}
tel.update(tel2)
print(tel)
#wynik: {'policja': 997, 'straż': 998, 'pogotowie': 999}
```

Słowniki

Aby uzyskać listę kluczy występujących w słowniku...

```
tel = {"taxi": 919, "policja": 997, "pogotowie": 999, "straż": 998}
klucze = tel.keys()
print(klucze)
#wynik: dict_keys(['taxi', 'policja', 'pogotowie', 'straż'])

klucze = list(tel.keys())
print(klucze)
#wynik: ['taxi', 'policja', 'pogotowie', 'straż']
```

Aby uzyskać listę wartości występujących w słowniku...

```
tel = {"taxi": 919, "policja": 997, "pogotowie": 999, "straż": 998}
wartosci = list(tel.values())
print(wartosci)
#wynik: [919, 997, 999, 998]
```

Słowniki

Słowniki i pętla for (1)

```
sloownik = {'imie': 'Jan', 'nazwisko': 'Kowalski', 'wiek': 30, 'zawód': 'lekarz'}
```

```
for (klucz, wartosc) in sloownik.items():  
    print(klucz + ": " + str(wartosc))
```

```
#wynik: imie: Jan
```

```
#wynik: nazwisko: Kowalski
```

```
#wynik: wiek: 30
```

```
#wynik: zawód: lekarz
```

Metoda `items()` zwraca kolejne pary klucz-wartość.

Słowniki i pętla for (2)

```
slovník = {'imie': 'Jan', 'nazwisko': 'Kowalski', 'wiek': 30, 'zawód': 'lekarz'}  
for klucz in slovník.keys():  
    print(klucz)  
  
#wynik: imie  
#wynik: nazwisko  
#wynik: wiek  
#wynik: zawód
```

Słowniki i pętla for (3)

```
slovník = {'imie': 'Jan', 'nazwisko': 'Kowalski', 'wiek': 30, 'zawód': 'lekarz'}  
for wartosc in slovník.values():  
    print(wartosc)  
  
#wynik: Jan  
#wynik: Kowalski  
#wynik: 30  
#wynik: lekarz
```

Słowniki

Program zamieniający całkowitą liczbę dziesiętną na odpowiadającą jej liczbę rzymską.

Przeanalizuj i przetestuj:

```
rzym = {1000:"M", 900:"CM", 500:"D", 400:"CD", 100:"C", 90:"XC",
50:"L", 40:"XL", 10:"X", 9:"IX", 5:"V", 4:"IV", 1:"I"}
liczba = int(input("Podaj liczbę całkowitą: "))
r = list(rzym.keys())
r.sort()
r.reverse()
lr = ""
x = liczba
for i in r:
    while i <= x:
        lr = lr + rzym[i]
        x = x - i
print("Liczba " + str(liczba) + " w notacji rzymskiej to: " + lr)
```

Zad. 63. Utwórz słownik *imię:ocena*, zawierający oceny czterech osób, a następnie oblicz średnią arytmetyczną ocen.

Zad. 64. Napisz program, który dla zadanej liczby całkowitej wypisuje słownie jej cyfry.

Przykład:

0: zero,

-147: minus jeden cztery siedem.

Zad. 65. Napisz program, który zamieni wprowadzoną liczbę rzymską na jej postać dziesiętną.