

# Програмування

---

## ТЕМА 8. СЛОВНИКИ

# СЛОВНИКИ

---

Словники є структурою даних, яка призначена для збереження сукупності різнотипних елементів.

До цих елементів є доступ за допомогою так званих ключів.

Ключ повинен належати типу, що не змінюється (immutable).  
Наприклад: число, рядок, кортеж.

У деяких мовах програмування словники інколи називають асоціативними масивами або колекціями.

# Носій типу словник

---

Словник позначається включенням пар <ключ>:<елемент> у фігурні дужки через кому.

$$\{\text{key}_1:x_1, \dots, \text{key}_n:x_n\}$$

Нехай множини  $M_1, \dots, M_n$  є носіями типів  $t_1, \dots, t_n$ , до яких належать  $x_1, \dots, x_n$ .  $M_{key}$  – множина ключів.

Тоді носієм типу словника буде  $M_t$  - множина відображень  $M_{key}$  у об'єднання  $M_1, \dots, M_n$ .

$$M_t = \{ M_{key} \rightarrow M_1 \cup \dots \cup M_n \}$$

# Операції для словників

Операція	Опис
<code>{key<sub>1</sub>:x<sub>1</sub>, ..., key<sub>n</sub>:x<sub>n</sub>}</code>	Створити словник з ключів <code>key<sub>1</sub>, ..., key<sub>n</sub></code> та елементів <code>x<sub>1</sub>, ..., x<sub>n</sub></code>
<code>{}</code>	Порожній словник
<code>dict(x)</code>	Перетворення <code>x</code> у словник ( <code>x</code> повинно належати типу, що ітерується, та складатися з пар значень)
<code>d[key]</code>	Елемент <code>d</code> з ключем <code>key</code> . Якщо такого ключа у словнику немає, виникає помилка
<code>len(d)</code>	Довжина <code>d</code>
<code>min(d)</code>	Найменший ключ словника <code>d</code>
<code>max(d)</code>	Найбільший ключ словника <code>d</code>
<code>d.copy()</code>	Повертає копію словника <code>d</code>
<code>d.fromkeys(s, v)</code>	Повертає словник типу <code>dict</code> , ключами якого є елементи послідовності <code>s</code> , а значеннями або <code>None</code> , або <code>v</code> , якщо аргумент <code>v</code> визначений

# Операції для словників.2

---

Операція	Опис
<code>d.get(k)</code>	Повертає елемент з ключем <code>k</code> або <code>None</code> , якщо ключа <code>k</code> немає у словнику
<code>d.get(k, v)</code>	Повертає елемент з ключем <code>k</code> або <code>v</code> , якщо ключа <code>k</code> немає у словнику
<code>d.items()</code>	Повертає представлення (послідовність) всіх пар (ключ, значення) в словнику <code>d</code>
<code>d.keys()</code>	Повертає представлення (послідовність) всіх ключів словника <code>d</code>
<code>d.values()</code>	Повертає представлення (послідовність) всіх значень в словнику <code>d</code>

# Відношення для словників

---

Для словників визначено відношення

$==$ ,  $!=$ ,  $in$ ,  $not\ in$ .

Відношення  $d == b$  означає попарну рівність всіх елементів двох словників  $d$ ,  $b$ .

Відношення  $d != b \equiv not (d == b)$

$x\ in\ d == True$ , коли  $x$  входить у  $d$

$x\ not\ in\ d == True$ , коли  $x$  не входить у  $d$

# Інструкції для словників

---

Для словників визначено присвоєння та виведення.

```
d = e, print(d)
```

Введення не визначено, тому треба вводити словник поелементно.

# Інструкції для словників.2

---

Визначено також цикли по всіх елементах словника

```
for key in d:
```

*P*

або

```
for key in d.keys():
```

*P*

або

```
for key, v in d.items():
```

*P*

Окрім цього, визначено ще ряд інструкцій



# Інструкції для словників.3

Інструкція	Опис
<code>d[k] = e</code>	Присвоїти елементу словника <code>d</code> з ключем <code>k</code> значення <code>e</code>
<code>del d[k]</code>	Видаляє елемент словника <code>d</code> з ключем <code>k</code> або дає помилку, якщо ключа <code>k</code> немає у словнику
<code>d.clear()</code>	Видаляє всі елементи словника <code>d</code>
<code>d.pop(k)</code>	Повертає елемент з ключем <code>k</code> і видаляє зі словника елемент з ключем <code>k</code> або дає помилку, якщо ключа <code>k</code> немає у словнику
<code>d.pop(k, v)</code>	Повертає елемент з ключем <code>k</code> і видаляє зі словника елемент з ключем <code>k</code> або повертає значення <code>v</code> , якщо ключ <code>k</code> немає у словнику
<code>d.popitem()</code>	Повертає і видаляє довільну пару (ключ, значення) зі словника <code>d</code> або дає помилку, якщо словник <code>d</code> порожній

# Інструкції для словників.4

Інструкція	Опис
<code>d.setdefault(k, v)</code>	Те ж, що і <code>dict.get ()</code> за винятком того, що, якщо ключ <code>k</code> в словнику відсутній, в словник вставляється новий елемент з ключем <code>k</code> і зі значенням <code>None</code> або <code>v</code> , якщо аргумент <code>v</code> заданий
<code>d.update(a)</code>	Додає в словник <code>d</code> пари (ключ, значення) з <code>a</code> , які відсутні в словнику <code>d</code> , а для кожного ключа, який вже присутній в словнику <code>d</code> , виконується заміна відповідним значенням з <code>a</code> ; <code>a</code> може бути словником, належати типу, що ітерується, з парами (ключ, значення) або іменованими аргументами

# Приклади

---

Слова у рядку розділяються одним або декількома пропусками. Визначити кількість входжень кожного слова до рядка та слово, яке входить найбільшу кількість разів (версія 1)

Ненульові елементи розрідженої матриці зберігаються у словнику. Ключами словника є кортежі, що складаються з індексів рядка та стовпчика, а значеннями словника, - значення елементів матриці. Для квадратної розрідженої матриці  $n \times n$  перевірити, чи є ця матриця симетричною (версія 1).

# Функції all, any та filter

---

Дві вбудованих функції `all ()` та `any ()` використовують для обчислення булевих значень від складених типів.

Вбудована функція `filter()` дозволяє відібрати зі складеного типу ті елементи, що задовольняють умову. Умова, в свою чергу, задана деякою функцією.

Якщо  $e$  – вираз типу, що ітерується, то

- `all (e)` повертає значення `True` тоді і тільки тоді, коли всі елементи  $e$  є істинними (`True`).
- `any (e)` повертає значення `True` тоді і тільки тоді, коли хоча б один елемент  $e$  є істинним (`True`).
- `filter(f, e)` повертає послідовність, що складається з усіх таких елементів  $e_i$ , які входять у  $e$  та для яких  $f(e_i)$  є істинним (`True`). `filter(f, e)`, як і раніше розглянута функція `map()`, застосовує функцію  $f$  до всіх елементів  $e$ .

# Функції all, any та filter.2

---

Ми вже визначали, коли бульові та числові вирази є істинними.

Для складених типів даних правила є такими:

- будь-який непорожній рядок є істинним
- будь-який непорожній список є істинним
- будь-який непорожній кортеж є істинним
- будь-який непорожній словник є істинним

# Приклад

---

Ненульові елементи розрідженої матриці зберігаються у словнику. Ключами словника є кортежі, що складаються з індексів рядка та стовпчика, а значеннями словника, - значення елементів матриці. Для квадратної розрідженої матриці  $n \times n$  перевірити, чи є ця матриця симетричною (версія 2).

# Словникоутворення

---

Словникоутворення (dictionary comprehension) – це вираз, результатом якого є словник.

Словникоутворення схоже на спискоутворення за виключенням того, що у виразі треба вказувати пари <ключ>:<елемент>.

Вираз має такий синтаксис:

`{key:value for key, value in tt if F}`

- де *key*, *value* – вирази, *tt* – вираз типу, що ітерується та містить пари елементів, *F* – умова.

# Словникоутворення.2

---

Python вибирає всі *key*, *value* з *tt*, які задовольняють умову *F*, додає у словник пари *key:value* та повертає отриманий словник.

Або

$\{k(x):v(x) \text{ for } x \text{ in } t \text{ if } F\}$

- де  $k(x)$ ,  $v(x)$  – вирази, які залежать від  $x$ ,  $t$  – вираз типу, що ітерується,  $F$  – умова.

Python вибирає всі  $x$  з  $t$ , які задовольняють умову  $F$ , застосовує до кожного  $x$  вирази  $k(x)$ ,  $v(x)$  та повертає отриманий словник.

Якщо умова  $F$  відсутня, то  $\text{if } F$  опускають.



# Приклад

---

Слова у рядку розділяються одним або декількома пропусками.  
Визначити кількість входжень кожного слова до рядка та слово, яке  
входить найбільшу кількість разів (версія 2)

# Резюме

---

Ми розглянули:

1. Словники. Носій для словників.
2. Операції, відношення та інструкції для словників.
3. Функції `all`, `any` та `filter`
4. Словникоутворення.

# Де прочитати

---

1. Обвінцев О.В. Інформатика та програмування. Курс на основі Python. Матеріали лекцій. – К., Основа, 2017
2. A Byte of Python (Russian) Версія 2.01 Swaroop С Н (Translated by Vladimir Smolyar),  
<http://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.01.pdf>
3. Марк Лутц, Изучаем Python, 4-е издание, 2010, Символ-Плюс
4. Python 3.4.3 documentation
5. [http://www.python-course.eu/python3\\_dictionaries.php](http://www.python-course.eu/python3_dictionaries.php)