# Тема 8. Словники

Словники є структурою даних, яка призначена для збереження сукупності різнотипних елементів. До кожного елемента є доступ за допомогою так званого ключа. 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.

{key1:x1, …, keyn:xn}

Нехай множини *M1, …, Mn* є носіями типів t1, …, tn, до яких належать x1, …, xn. *Mkey* – множина ключів.

Тоді носієм типу словника буде *Mt* - множина відображень *Mkey* у об’єднання *M1, …, Mn*.

*Mt* = { *Mkey* → *M1*U*…*U*Mn*}

Операції для словників

|  |  |
| --- | --- |
| **Операція** | **Опис** |
| {key1:x1, …, keyn:xn} | Створити словник з ключів key1, …, keyn та елементів x1, … , xn |
| {} | Порожній словник |
| dict(x) | Перетворення *x* у словник (x повинно належати типу, що ітерується) |
| d[key] | Елемент *d* з ключем key. Якщо такого ключа у словнику немає, виникає помилка |
| len(d) | Довжина *d* |
| min(d) | Найменший ключ словника *d* |
| max(d) | Найбільший ключ словника *d* |
| d.copy() | Повертає копію словника *d* |
| d.fromkeys(s, v) | Повертає словник типу dict, ключами якого є елементи послідовності s, а значеннями або None, або v, якщо аргумент v визначений |
| d.get(k) | Повертає значення ключа k або None, якщо ключа k немає у словнику |
| d.get(k, v) | Повертає значення ключа k або v, якщо ключа k немає у словнику |
| d.items() | Повертає представлення (послідовність) всіх пар (ключ, значення) в словнику *d* |
| d.keys() | Повертає представлення (послідовність) всіх ключів словника *d* |
| d.values() | Повертає представлення (послідовність) всіх значень в словнику *d* |

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 ≡ 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.

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

|  |  |  |
| --- | --- | --- |
| **Інструкція** | **Опис** | |
| d[k] = e | Присвоїти елементу словника *d* з ключем *k* значення *e* |
| del d[k] | Видаляє елемент словника *d* з ключем *k* або дає помилку, якщо ключа *k* немає у словнику |
| d.clear() | Видаляє всі елементи словника *d* |
| d.pop(k) | Повертає значення ключа *k* і видаляє зі словника елемент з ключем *k* або дає помилку, якщо ключа *k* немає у словнику |
| d.pop(k, v) | Повертає значення ключа *k* і видаляє зі словника елемент з ключем *k* або повертає значення *v*, якщо ключ *k* немає у словнику |
| d.popitem() | Повертає і видаляє довільну пару (ключ, значення) зі словника *d* або дає помилку, якщо словник *d* порожній |
| d.setdefault(k, v) | Те ж, що і dict.get () за винятком того, що, якщо ключ *k* в словнику відсутній, в словник вставляється новий елемент з ключем *k* і зі значенням None або *v*, якщо аргумент v заданий |
| d.update(a) | Додає в словник *d* пари (ключ, значення) з *а*, які відсутні в словнику *d*, а для кожного ключа, який вже присутній в словнику *d*, виконується заміна відповідним значенням з *а*; *а* 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×n перевірити, чи є ця матриця симетричною (версія 1).

Функції all, any та filter

Дві вбудованих функції all () та any () використовують для обчислення бульових 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 *ei*, які входять у *e* та для яких *f*(*ei*) є істинним (True). filter(*f*, *e*), як і раніше розглянута функція map(), застосовує функцію *f* до всіх елементів *e*.

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×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.

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) for x in *t* 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 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. A Byte of Python (Russian) Версия 2.01 Swaroop C H (Translated by Vladimir Smolyar), <http://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.01.pdf>
2. Марк Лутц, Изучаем Python, 4-е издание, 2010, Символ-Плюс
3. Python 3.4.3 documentation
4. Марк Саммерфилд, Программирование на Python 3. Подробное руководство. - Символ-Плюс, 2009.
5. <http://www.python-course.eu/python3_dictionaries.php>