



Python

Множества и словари

Урок 7



Проверка знаний

2

1. Что такое функция?
2. Что делает команда return?
3. Как использовать функцию?

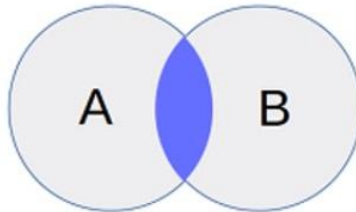
```
def add(x, y):  
    return x + y
```



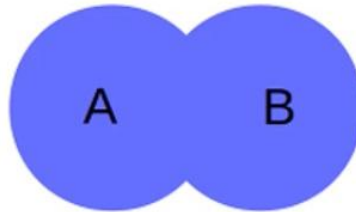
Множества

2

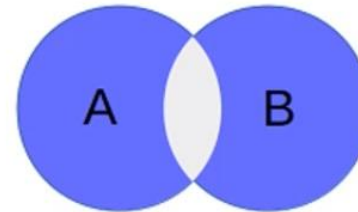
Пересечение $A \cap B$



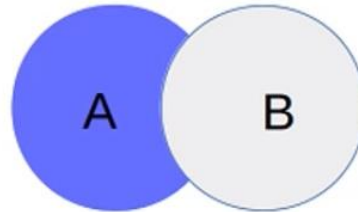
Объединение $A \cup B$



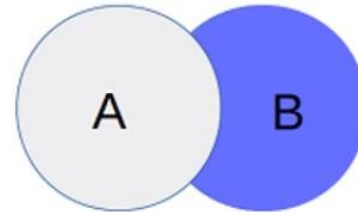
Симметричная разность $A \oplus B$



Разница $A - B$



Разница $B - A$





Задание множеств

3

```
A = {1, 2, 3}
```

```
A = set('qwerty')  
print(A)
```

```
A = {1, 2, 3}  
B = {3, 2, 3, 1}  
print(A == B)
```



Задание множеств

3

```
A = {1, 2, 3}
```

```
A = set('qwerty')  
print(A)
```

```
A = {1, 2, 3}  
B = {3, 2, 3, 1}  
print(A == B)
```



Перебор элементов множеств

5

```
Primes = {2, 3, 5, 7, 11}
for num in Primes:
    print(num)
```

untitled-3.py (pid 160 ▾)

Debug I/O (stdin, stdout, stderr) appears below

```
2
3
5
7
11
```



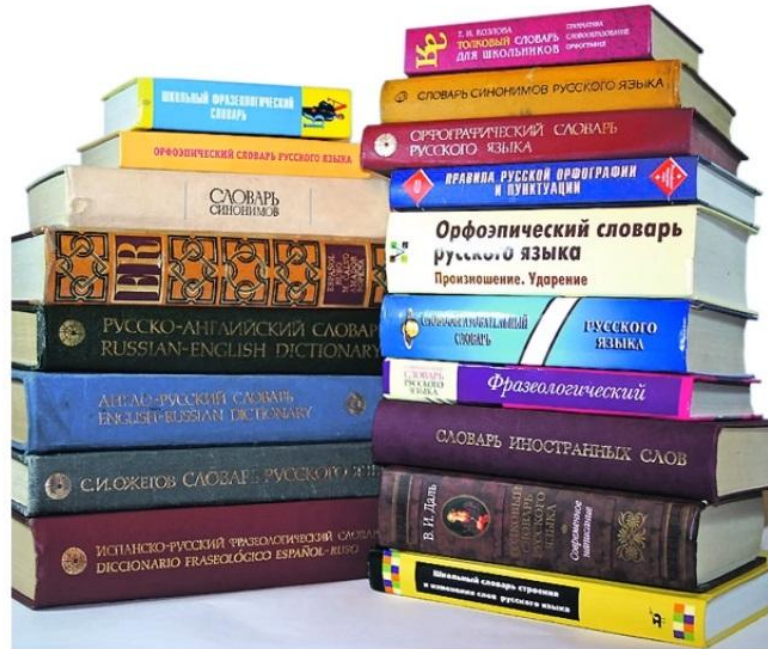
Операции над множествами

6

Операция	Описание
$A \mid B$ - <code>A.union(B)</code>	Возвращает множество, являющееся объединением множеств A и B
$A \mid= B$ - <code>A.update(B)</code>	Добавляет в множество A все элементы из множества B
$A \& B$ - <code>A.intersection(B)</code>	Возвращает множество, являющееся пересечением множеств A и B
$A \&= B$ - <code>A.intersection_update(B)</code>	Оставляет в множестве A только те элементы, которые есть в множестве B
$A - B$ - <code>A.difference(B)</code>	Возвращает разность множеств A и B (элементы, входящие в A, но не входящие в B)
$A -= B$ - <code>A.difference_update(B)</code>	Удаляет из множества A все элементы, не входящие в множество B
$A \wedge B$ - <code>A.symmetric_difference(B)</code>	Возвращает симметрическую разность множеств A и B (элементы, входящие в A или в B, но не в оба из них одновременно)
$A \wedge= B$ - <code>A.symmetric_difference_update(B)</code>	Записывает в A симметрическую разность множеств A и B
$A \leq B$ - <code>A.issubset(B)</code>	Возвращает True, если A является подмножеством B
$A \geq B$ - <code>A.issuperset(B)</code>	Возвращает True, если A является надмножеством B



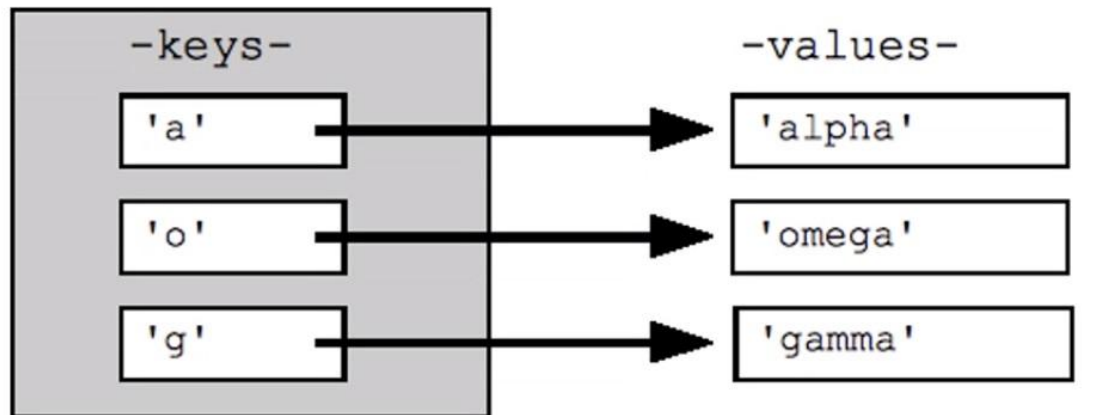
Словари



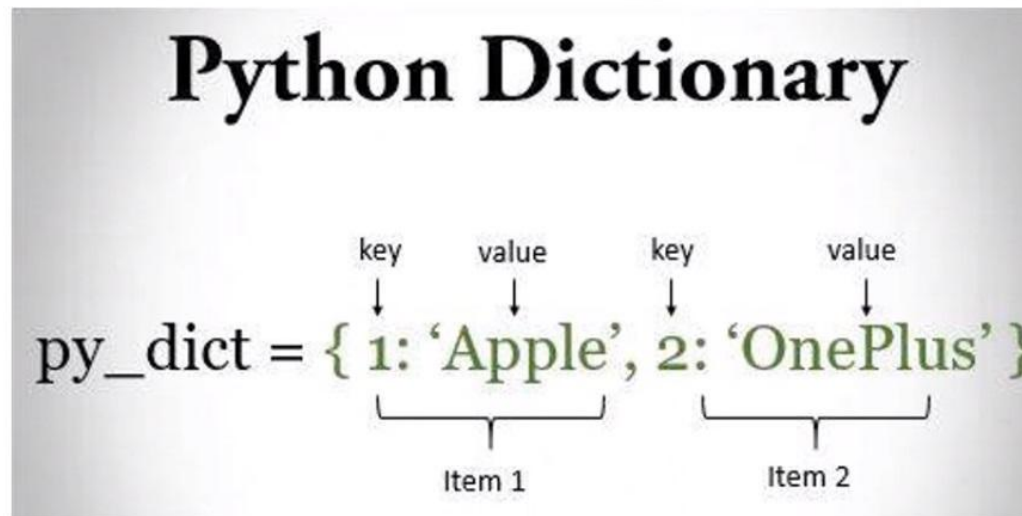


Словари в Python

8



`dict` Создание словаря в Python





Создания словаря

10

```
Capitals = {'Russia': 'Moscow', 'Ukraine': 'Kiev', 'USA':  
'Washington'}  
Capitals = dict(Russia = 'Moscow', Ukraine = 'Kiev', USA =  
'Washington')  
Capitals = dict([("Russia", "Moscow"), ("Ukraine", "Kiev"),  
("USA", "Washington")])  
Capitals = dict(zip(["Russia", "Ukraine", "USA"], ["Moscow",  
"Kiev", "Washington"]))
```



Работа с элементами словаря

11

```
A = {'ab' : 'ba', 'aa' : 'aa', 'bb' : 'bb', 'ba' : 'ab'}
key = 'ac'
# 1 способ
if key in A:
    del A[key]
# 2 способ
try:
    del A[key]except
KeyError:
    print('There is no element with key "' + key + '"
in dict')print(A)
```



Перебор элементов словаря

12

```
A = dict(zip('abcdef', list(range(6))))  
for key in A:  
    print(key, A[key])
```

```
A = dict(zip('abcdef', list(range(6))))  
for key, val in A.items():  
    print(key, val)
```



ПРИМЕР ХРАНЕНИЯ СПИСКОВ В СЛОВАРЕ

```
>>> SUFFIXES = {1000: ['KB', 'MB', 'GB', 'TB', 'PB', 'EB',  
'ZB', 'YB'], 1024: ['KiB', 'MiB', 'GiB', 'TiB', 'PiB',  
'EiB', 'ZiB', 'YiB']}  
>>> len(SUFFIXES) #1  
2  
>>> 1000 in SUFFIXES #2  
True  
>>> SUFFIXES[1000] #3  
['KB', 'MB', 'GB', 'TB', 'PB', 'EB', 'ZB', 'YB']  
>>> SUFFIXES[1024] #4  
['KiB', 'MiB', 'GiB', 'TiB', 'PiB', 'EiB', 'ZiB', 'YiB']  
>>> SUFFIXES[1000][3] #5  
'TB'
```



Проект «Столицы стран»

14

```
# Создадим пустой словарь Capitals
Capitals = dict()
# Заполним его несколькими значениями
Capitals['Russia'] = 'Moscow'
Capitals['Ukraine'] = 'Kiev'
Capitals['USA'] = 'Washington'
# Считаем название страны
print('В какой стране вы живете?')
country = input()
# Проверим, есть ли такая страна в словаре Capitals
if country in Capitals:
    # Если есть - выведем ее столицу
    print('Столица вашей страны', Capitals[country])
else:
    # Запросим название столицы и добавив его в словарь
    print('Как называется столица вашей страны?')
    city = input()
    Capitals[country] = city
```



Задача «Номер появления слова»

15

Условие

В единственной строке записан текст. Для каждого слова из данного текста подсчитайте, сколько раз оно встречалось в этом тексте ранее.

Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.



Задача «Номер появления слова»

16

```
A = {}  
for w in input().split():  
    A[w] = A.get(w, 0) + 1  
print(A[w] - 1, end=' ')
```



Задача «Выборы в США»

17

Условие

Как известно, в США президент выбирается не прямым голосованием, а путем двухуровневого голосования. Сначала проводятся выборы в каждом штате и определяется победитель выборов в данном штате. Затем проводятся государственные выборы: на этих выборах каждый штат имеет определенное число голосов — число выборщиков от этого штата. На практике, все выборщики от штата голосуют в соответствии с результатами голосования внутри штата, то есть на заключительной стадии выборов в голосовании участвуют штаты, имеющие различное число голосов.

В первой строке дано количество записей. Далее, каждая запись содержит фамилию кандидата и число голосов, отданных за него в одном из штатов. Подведите итоги выборов: для каждого из участника голосования определите число отданных за него голосов. Участников нужно выводить в алфавитном порядке.



Задача «Выборы в США»

18

```
n = int(input())
d = {}
def func(key, val):
    if key in d:
        d[key] += int(val)
    else:
        d[key] = int(val)
for j in range(n):
    key, val = input().split()
    func(key, val)
l = d.keys() # получаем ключи
l = list(l) # превращаем его в обычный список
l.sort() # сортируем список
for i in l: # вывод элементов словаря по алфавиту
    print(i, d[i])
```



Домашнее задание

Задача «Словарь синонимов»

19

Условие

Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны.

Для слова из словаря, записанного в последней строке, определите его синоним.



Молодцы!

