



Become QA Auto

Множини

Бутенко Сергій



```
le.py > ...
x = "PiMyLifeUp" #First assign the 'x' variable a string
type(x) #Output the data type of our 'x' variable
# 43 #Now assign it a integer value
type(x) #Output its type again

days = 5
print("There are " + str(days) + " days r

a, b, c = "Apple", "Banana", "Cherry"

b = c = "Lime"

print("Hello World")
# 453
int(x)
int(y)

print("Life Up")
```



План лекції



Поняття множини



Вилучення елементів



Додавання елементів



Операції над множинами



Доступ до елементів



Множини



Множина countries
Ukraine
Poland
Great Britain
...
China

Множина – це **НЕ** впорядкований набір різнорідних даних, який не містить дублікатів.

Множина **НЕ** може містити дублікати
Множина - динамічна структура

Основні етапи роботи з множинами:

- Оголошення та ініціалізація
- Опрацювання
- Вивід результату

Оголошення множини:

`<ім'я> = {<елемент1>, <елемент2>}` **або**

`<ім'я> = set((<елемент1>, <елемент2>))`

Приклад:

```
countries = {'Ukraine', 'Poland'...} або
```

```
countries = set(('Ukraine', 'Poland'...))
```

Додавання елементів до множини



Множина countries
Ukraine
Poland
Great Britain
...
China

Загальний вигляд додавання елемента до множини:

`<ім'я множини>.add(<елемент>)`

Наприклад:

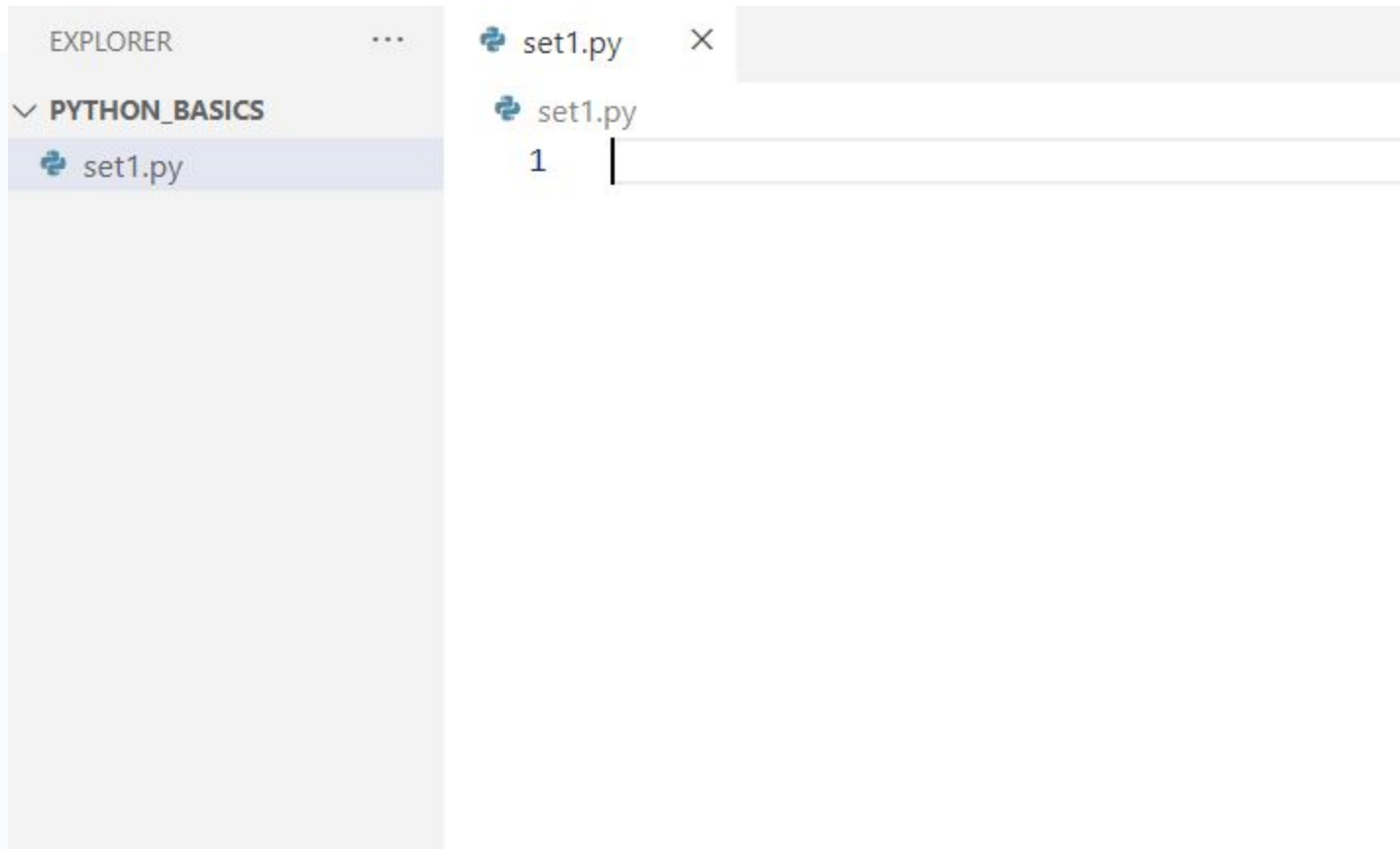
```
countries = {'Ukraine', 'Poland'}  
print(countries)  
>> {'Poland', 'Ukraine'}
```

```
uk = "Great Britain"  
countries.add(uk)  
print(countries)  
>> {'Poland', 'Ukraine', 'Great Britain'}
```

```
countries.add(uk)  
print(countries)  
>> {'Poland', 'Ukraine', 'Great Britain'}
```



Практика





Практика



set1.py X

set1.py > countries

```
1 countries = {'Ukraine', 'Poland'}
2 print(countries)
3
4 uk = 'Great Britain'
5 countries.add(uk)
6 print(countries)
7
8 countries.add(uk)
9 print(countries)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/sergii.butenko/Desktop/python_basics/set1.py
{'Poland', 'Ukraine'}
{'Great Britain', 'Poland', 'Ukraine'}
{'Great Britain', 'Poland', 'Ukraine'}
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```



Доступ до елемента множини



Множина countries
Ukraine
Poland
Great Britain
...
China

Отримати доступ до певного елемента множини неможливо.

Для перевірки чи елемент присутній у множині використовують оператор - in

Наприклад:

```
countries = {'Ukraine', 'Poland'}  
print(countries)  
>> {'Ukraine', 'Poland'}  
print('Ukraine' in countries)  
>> True  
print('US' in countries)  
>> False
```



Практика



```
set2.py X
8.4 Множини > set2.py > ...
1 countries = {'Ukraine', 'Poland'}
2 print(countries)
3 print('Ukraine' in countries)
4 print('US' in countries)
5
6 if 'Ukraine' in countries:
7     print("Україна є в переліку країн")
8 else:
9     print("України немає в переліку країн")
10
11 if 'US' in countries:
12     print("Сполучені Штати є в переліку країн")
13 else:
14     print("Сполучених Штатів немає в переліку країн")
15
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/sergii.buter
sktop/python_basics/8.4 Множини/set2.py"
{'Poland', 'Ukraine'}
True
False
Україна є в переліку країн
Сполучених Штатів немає в переліку країн
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```




Вилучення елементів множини



Множина countries
Ukraine
Poland
China

Можливе вилучення лише за значенням елемента

Наприклад:

```
countries = { 'Ukraine', 'Poland', 'China' }
```

```
print(countries)
```

```
>> { 'Ukraine', 'China', 'Poland' }
```

```
countries.remove( 'China' )
```

```
countries.discard( 'Austria' )
```

```
print(countries)
```

```
>> { 'Ukraine', 'Poland' }
```



Практика



set3.py X

8.4 Множини > set3.py > ...

```
1 countries = {'Ukraine', 'Poland', 'China'}
2 print(countries)
3
4 uk = 'Great Britain'
5 countries.add(uk)
6 print(countries)
7
8 countries.remove('China')
9 countries.discard('Austria')
10 print(countries)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

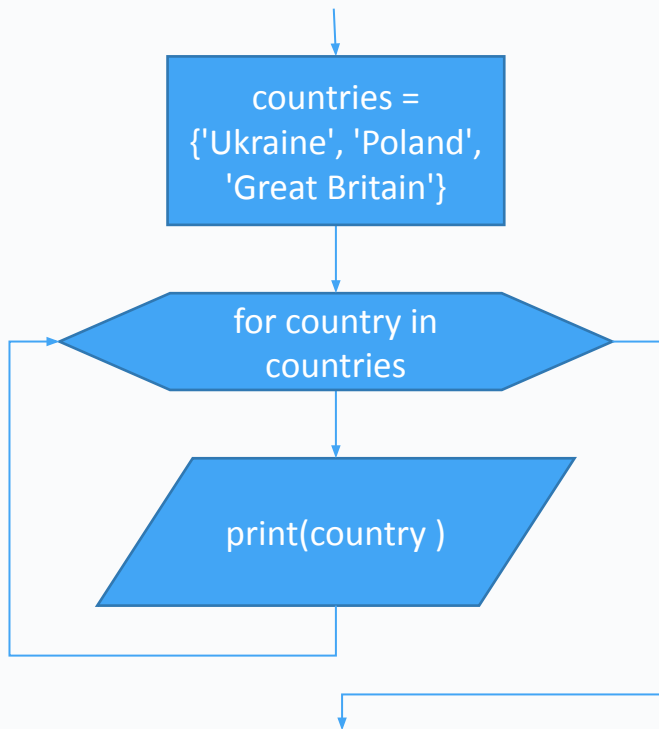
TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/s
/python_basics/8.4 Множини/set3.py"
{'Poland', 'Ukraine', 'China'}
{'Poland', 'Ukraine', 'China', 'Great Britain'}
{'Poland', 'Ukraine', 'Great Britain'}
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```



Автоматизація опрацювання множин за допомогою циклів



Задача:

Вивести назви усіх зареєстрованих країн

Розв'язок задачі:

В циклі для змінної `country`, що набуває значення кожного елементу множини, по черзі:

вивести значення `country` на екран



Практика



set_for_loop.py X

8.4 Множини > set_for_loop.py > ...

```
1 countries = {'Ukraine', 'Poland', 'Great Britain'}
2
3 for countrie in countries:
4     print(countrie)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/sergii.butenko/Desktop/python_basics/8.4 Множини/set_for_loop.py"
```

```
Poland
```

```
Great Britain
```

```
Ukraine
```

```
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```

◎ **Опрацювання множин**



- об'єднання ($|$)
- перетин ($\&$)
- різниця ($-$)

Опрацювання множин - об'єднання



Множина countries_friends	Множина countries_new_friends
Ukraine	Great Britain
Poland	USA

Оператор |

об'єднує дві множини і повертає нову множину, яка включає всі елементи обидвох множин.

```
countries_friends = { 'Ukraine',  
                      'Poland' }  
countries_new_friends = { 'Great  
Britain', 'USA' }
```

```
print(countries_friends |  
      countries_new_friends)  
>> {'USA', 'Great Britain',  
     'Ukraine', 'Poland'}
```

Опрацювання множин - об'єднання



set4.py X

set4.py > ...

```
1 countries_friends = {'Ukraine', 'Poland'}
2 countries_new_friends = {'Great Britain', 'USA'}
3
4 print(countries_friends | countries_new_friends)
5
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/sergii.butenko/Desktop/python_basics/set4.py
{'Poland', 'Ukraine', 'USA', 'Great Britain'}
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```



Опрацювання множин - перетин



Множина countries_friends	Множина countries_new_friends
Ukraine	Poland
Poland	Great Britain
	USA

Оператор &

повертає нову множину із значеннями, які належать одночасно обом множинам.

```
countries_friends = { 'Ukraine',  
                      'Poland' }  
countries_new_friends = { 'Poland',  
                          'Great Britain', 'USA' }  
  
print(countries_friends &  
      countries_new_friends )  
>> { 'Poland' }
```




Опрацювання множин - перетин



set5.py ×

set5.py > countries_friends

```
1 countries_friends = {'Ukraine', 'Poland'}
2 countries_new_friends = {'Poland', 'Great Britain', 'USA'}
3
4 print(countries_friends & countries_new_friends)
5
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/sergii.butenko/python_basics/set5.py
{'Poland'}
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```



Опрацювання множин - різниця



Множина countries_friends	Множина countries_new_friends
Ukraine	Poland
Poland	Great Britain
Austria	USA

Оператор -

повертає множину із елементами, які є в першій множині, але немає в іншій.

```
countries_friends = { 'Ukraine',  
                      'Poland', 'Austria' }  
countries_new_friends = { 'Poland',  
                          'Great Britain', 'USA' }
```

```
print(countries_friends -  
      countries_new_friends )  
>> { 'Ukraine', 'Austria' }
```



Опрацювання множин - різниця



set6.py X

set6.py > ...

```
1 countries_friends = {'Ukraine', 'Poland', 'Austria'}
2 countries_new_friends = {'Poland', 'Great Britain', 'USA'}
3
4 print(countries_friends - countries_new_friends)
5 |
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/sergii.butenko/
python_basics/set6.py
{'Austria', 'Ukraine'}
PS C:\Users\sergii.butenko\Desktop\python_basics> □
```



Підсумки



Множина – це НЕ впорядкований набір різнорідних даних, який не містить дублікатів



Множина - динамічна структура.



Основні етапи роботи з множинами:

- Оголошення та ініціалізація
- Опрацювання
- Вивід результату



Операції над множинами

- Об'єднання (|)
- Перетин (&)
- Різниця (-)