

КОЛЕКЦІЇ ТА МЕТОДИ

ТИПИ ДАНИХ

Складні та прості типи

Колекції

Lists

Set

Tuple

Dict

Прості типи даних

Int

Float

Bool

Double

Структури даних

Class

Struct

Enum

Protocol

Бібліотечні типи

NumpyArray

DataFrame

Linkedlist

Etc.



LIST

[1,2,3, 'Text', 5]

Упорядкованні за місцем розташування колекції об'єкти
довільних типів, розмір яких не обмежений.

В системі належать до класу **list**.

Example: [1,2,3,4,'String']

ДОСТУП ДО ЕЛЕМЕНТУ ПО ІНДЕКСУ

`arr[0] = 1, arr[1] = 2 , ...`, доступ до
елемента по індексу

`arr = [1,2,3,4,5]`

Список(list)

Елемент списка

ДОДАТИ ЕЛЕМЕНТИ У СПИСОК

Append — додати один елемент у кінець списку

`arr = [1,2,3]` `arr.append(4) -> [1,2,3,4]`

Extend — додати декілька елементів у кінець списку

`arr = [1,2,3]` `arr.extend([4,5]) -> [1,2,3,4,5]`

ВКЛАДЕНІ СПИСКИ

$a = [1, 2, 3, [4, 5, [2, 3, 4], 6, 7], 6, 7]$

$a[3][2][0]$

Який буде елемент?

$a[3]$

Який буде елемент?

$a[10]$

Який буде елемент?

ВКЛАДЕНІ СПИСКИ

$a = [1, 2, 3, [4, 5, [2, 3, 4], 6, 7], 6, 7]$

$a[3][2][0]$

Який буде елемент?

2

$a[3]$

Який буде елемент?

$a[10]$

Який буде елемент?

ВКЛАДЕНІ СПИСКИ

$a = [1, 2, 3, [4, 5, [2, 3, 4], 6, 7], 6, 7]$

$a[3][2][0]$

Який буде елемент?

2

$a[3]$

Який буде елемент?

$[4, 5, [2, 3, 4], 6, 7][2]$

$a[10]$

Який буде елемент?

ВКЛАДЕНІ СПИСКИ

`a = [1,2,3,[4,5,[2,3,4],6,7],6,7]`

`a[3][2][0]`

Який буде елемент?

2

`a[3]`

Який буде елемент?

`[4,5,[2,3,4],6,7][2]`

`a[10]`

Який буде елемент?

Error

LIST/ARRAY METHODS

Method	Description
<u>append()</u>	Adds an element at the end of the list
<u>clear()</u>	Removes all the elements from the list
<u>copy()</u>	Returns a copy of the list
<u>count()</u>	Returns the number of elements with the specified value
<u>extend()</u>	Add the elements of a list (or any iterable), to the end of the current list
<u>index()</u>	Returns the index of the first element with the specified value
<u>insert()</u>	Adds an element at the specified position
<u>pop()</u>	Removes the element at the specified position
<u>remove()</u>	Removes the first item with the specified value
<u>reverse()</u>	Reverses the order of the list
<u>sort()</u>	Sorts the list

Source: https://www.w3schools.com/python/python_ref_list.asp

**КОРТЕЖ
(TUPLE)**

TUPLE

Упорядкована колекція яка захищена від змін, можете мати довільний розмір у пам'яті

`first_tuple = (1, 2, 3, 4, 5) # Створення tuple` `first_list = [1, 2, 3, 4, 5] # Створення list`

Різниця:

```
a = (1, 2, 3, 4)
b = [1, 2, 3, 4]
print(f'Tuple in memory: {a.__sizeof__()}')
print(f'List in memory: {b.__sizeof__()}')
```

click to scroll output; double click to hide

```
Tuple in memory: 56
List in memory: 72
```

TUPLE METHODS

Python has two built-in methods that you can use on tuples.

Method	Description
<u>count()</u>	Returns the number of times a specified value occurs in a tuple
<u>index()</u>	Searches the tuple for a specified value and returns the position of where it was found

+ Множення на число, додавання, etc

SET

SET (МНОЖИНА)

Не упорядкована та неіндексована колекція унікальних елементів

1

Не упорядкована колекція

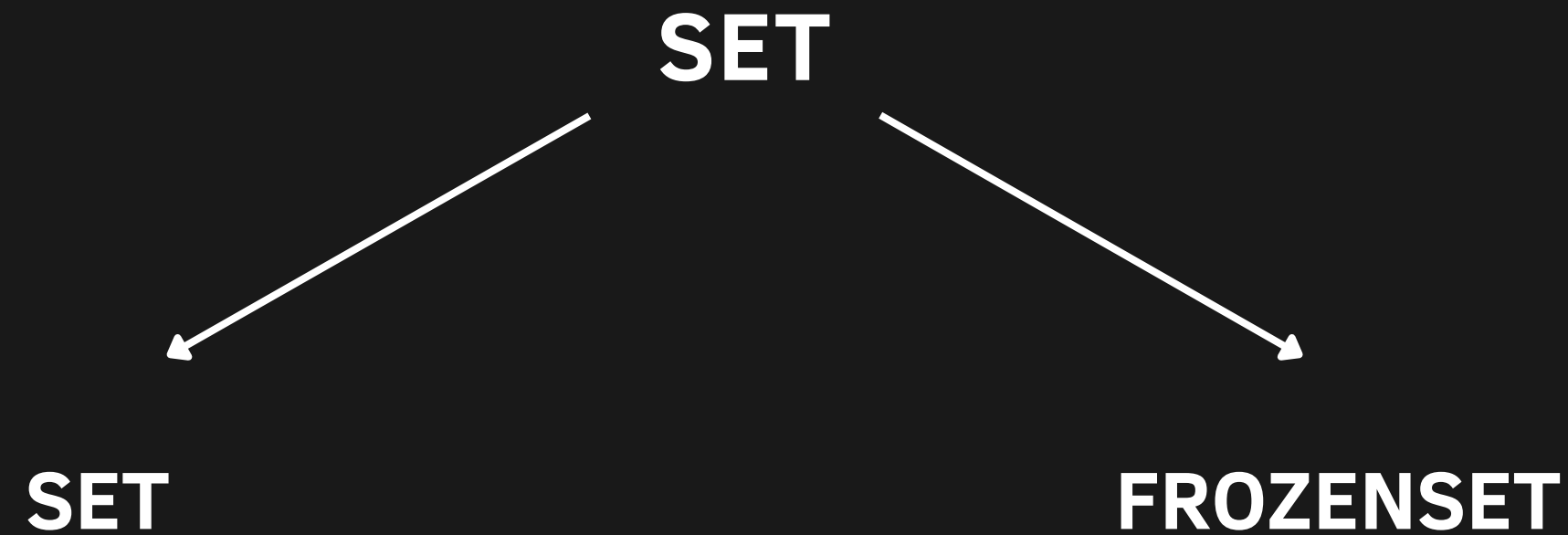
2

Тільки унікальні значення

2

Діапазон операцій: математична теорія множин

TYPES OF SET



Змінна колекція

Не може бути ключом для
Dict

```
some_dict = {some_set: 1}
TypeError: unhashable type: 'set'
```

Незмінна колекція

Може бути ключом для dict

```
Dict: {frozenset({1, 2, 3, 4}): 1}
Type of data: <class 'dict'>
Key: dict_keys([frozenset({1, 2, 3, 4})])
```

ВІДНОСИТЬСЯ ДО КЛАСУ SET

```
some_set = {1,2,3, 3,-1,1,1}
```

```
Set = {1, 2, 3, -1}
```

```
Type: <class 'set'>
```

```
Size: 200
```

OPERATION WITH SET

UNION OPERATION: ADD DISTINCT ELEMENT A AND B

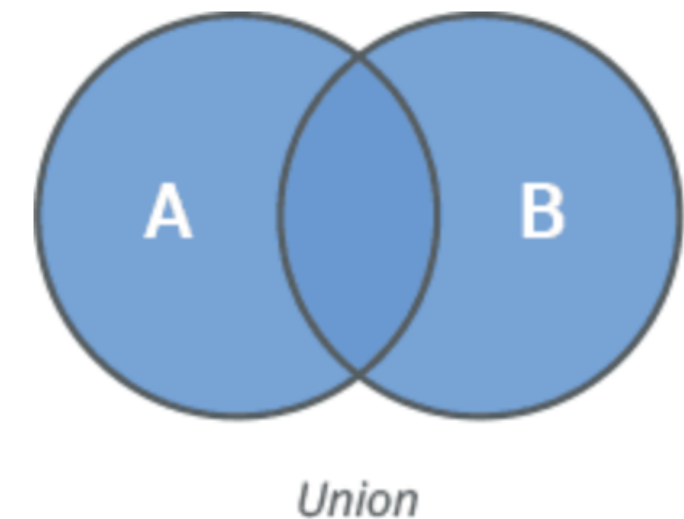
Key word: union or |

Set 1: {1, 2, 3, 4, 5}

Set 2: {1, 2, 6, 7, 8}

Operation union: {1, 2, 3, 4, 5, 6, 7, 8}

Union with | : {1, 2, 3, 4, 5, 6, 7, 8}



OPERATION WITH SET

INTERSECTION OPERATION: ADD ELEMENT IN A AND B

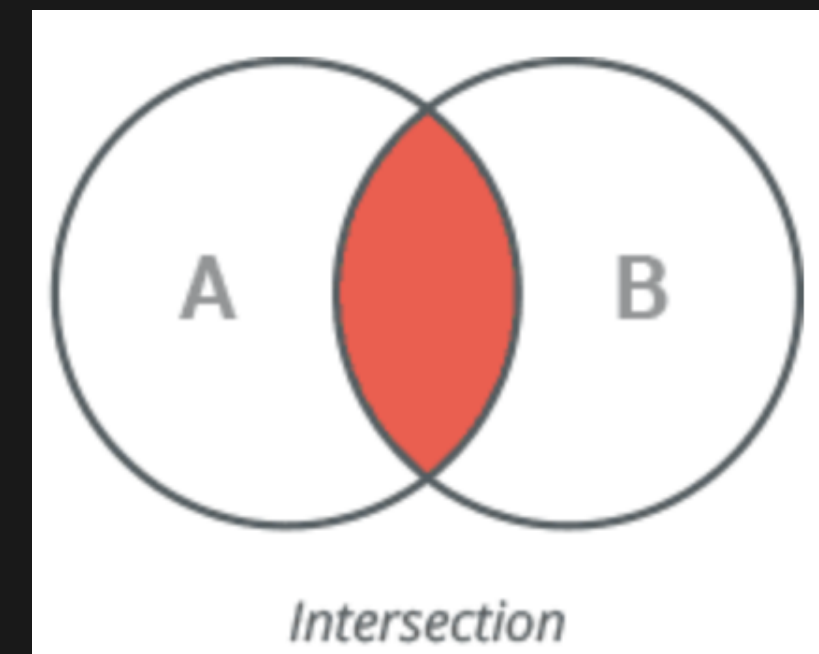
Key word: intersection or &

Set 1: {1, 2, 3, 4, 5}

Set 2: {1, 2, 6, 7, 8}

Operation intersection: {1, 2}

Intersection with & : {1, 2}



OPERATION WITH SET

DIFFERENCE OPERATION: DIFF ELEMENTS A AND B

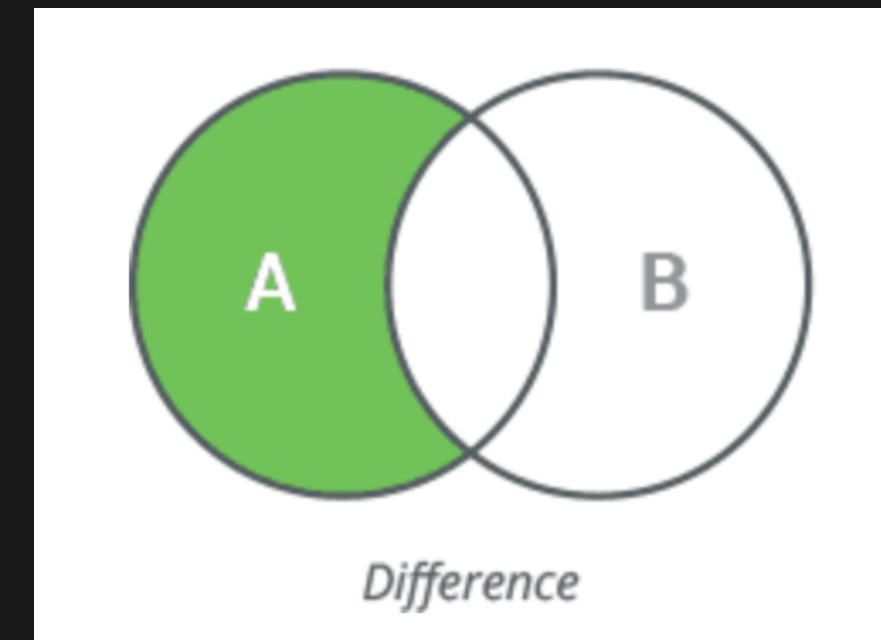
Key word: diffence or - (minus)

Set 1: {1, 2, 3, 4, 5}

Set 2: {1, 2, 6, 7, 8}

Operation difference: {3, 4, 5}

Union with - (minus) : {3, 4, 5}



OPERATION WITH SET

SYMM DIFF OPERATION: ELEMENT IN A AND B

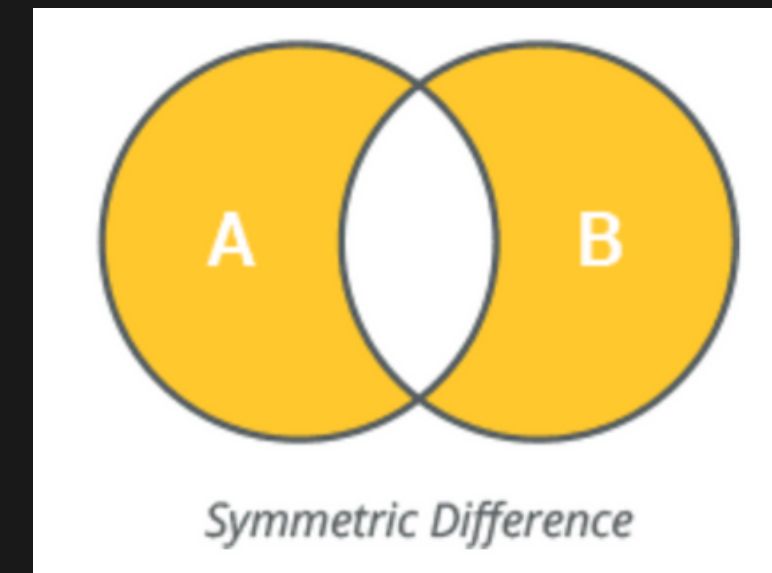
Key word: symmetric_difference or ^

Set 1: {1, 2, 3, 4, 5}

Set 2: {1, 2, 6, 7, 8}

Operation Symmetric difference: {3, 4, 5, 6, 7, 8}

Symm_diff with ^ {3, 4, 5, 6, 7, 8}



ISSUBSET

Метод `issubset()` повертає значення `True`, якщо всі елементи множини присутні в заданій множині, інакше `FALSE`.

Key word: `issubset`, `<=`

```
Set 1: {1, 2, 3, 4, 5}
```

```
Set 2: {1, 2, 6, 7, 8}
```

```
Operation issubset: False
```

```
Union with <= : False
```

MORE SET METHODS

Method	Description
<u>add()</u>	Adds an element to the set
<u>clear()</u>	Removes all the elements from the set
<u>copy()</u>	Returns a copy of the set
<u>difference()</u>	Returns a set containing the difference between two or more sets
<u>difference_update()</u>	Removes the items in this set that are also included in another, specified set
<u>discard()</u>	Remove the specified item
<u>intersection()</u>	Returns a set, that is the intersection of two other sets
<u>intersection_update()</u>	Removes the items in this set that are not present in other, specified set(s)
<u>isdisjoint()</u>	Returns whether two sets have a intersection or not
<u>issubset()</u>	Returns whether another set contains this set or not
<u>issuperset()</u>	Returns whether this set contains another set or not
<u>pop()</u>	Removes an element from the set
<u>remove()</u>	Removes the specified element
<u>symmetric_difference()</u>	Returns a set with the symmetric differences of two sets
<u>symmetric_difference_update()</u>	inserts the symmetric differences from this set and another
<u>union()</u>	Return a set containing the union of sets
<u>update()</u>	Update the set with the union of this set and others

DICTIONARY
DEFAULT DICT

DICTIONARY

Словники в мові Python не є послідовностями, а відображеннями. Відображення — це колекції об'єктів, але доступ до них здійснюється не за індексами, а за ключами.

```
Dictionary: {1: 'one', 2: 'two', 3: 'three', 4: 'four'}  
Type: <class 'dict'>  
Size: 216
```

Значення('one','two' ,3) — елементи словника (values)



```
example_dict = {1: 'one', 2: 'two', 3: 'three'}
```



Ключі(1,2 ,3) - елементи по яким отримуємо доступ
(keys)

MORE DICTIONARY METHODS

Method	Description
<u>clear()</u>	Removes all the elements from the dictionary
<u>copy()</u>	Returns a copy of the dictionary
<u>fromkeys()</u>	Returns a dictionary with the specified keys and value
<u>get()</u>	Returns the value of the specified key
<u>items()</u>	Returns a list containing a tuple for each key value pair
<u>keys()</u>	Returns a list containing the dictionary's keys
<u>pop()</u>	Removes the element with the specified key
<u>popitem()</u>	Removes the last inserted key-value pair
<u>setdefault()</u>	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
<u>update()</u>	Updates the dictionary with the specified key-value pairs
<u>values()</u>	Returns a list of all the values in the dictionary

DEFAULT DICT

FOR USE

```
from collections import defaultdict # need to import
```

Defaultdict - handle errors

```
data = defaultdict() # Create dd
```

```
defaultdict(None, {}) <class 'collections.defaultdict'>
```

NAMEDTUPLE

For use:

```
from collections import namedtuple
```

Namedtuple - покращений варіант tuple, доступ до значень відбувається як за ім'ям та і по індексу

```
Car = namedtuple('Car', ['type_of_engines', 'mark', 'color', 'fuel'])
```



Значення('type', 'mark', 'color', 'fuel') — іменовані елементи tuple

Problems:

1

Не можемо вказати анотацію типів

2

Не змінна структура даних

Solving:

1

Використати NamedTuple з бібліотеки typing

2

Робити перетворення типів

For use:
`from collections import namedtuple`

Namedtuple - покращений варіант `namedtuple`, доступ до значень відбувається як за ім'ям та за індексом. Дозволяє робити анотацію типів.

```
from typing import NamedTuple
```

```
class Car_Named(NamedTuple):  
    mark : str  
    color : str  
    size_of_engines: float  
    type_of_fuel: str  
    year: int
```



**KEEP
CALM
AND
HAPPY
CODING**