Python Sets

Overviews:

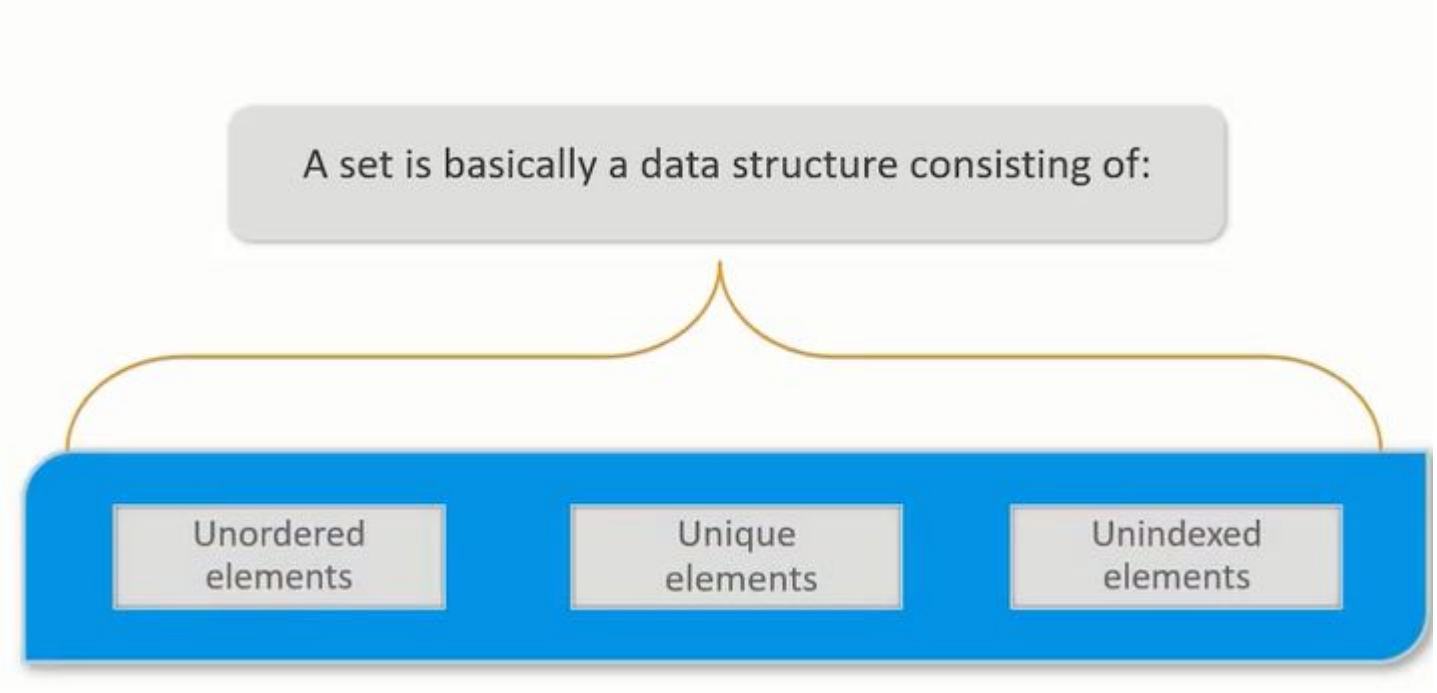
- What is a python sets?
- Where to use sets?
- How to create sets in python?
- Sets operations:

1) Adding / Updating elements
2) Removing/ Deleting elements
3) Union
4) Intersection
5) Difference

- Mutable set
- Sets mutation operation

What is a python sets?

- A set is an unordered collection of items. Every element is unique and must be immutable.
- However, the set itself is mutable. We can add or remove items from it.
- Sets can be used to perform mathematical set operations like union, intersection, symmetric difference etc.

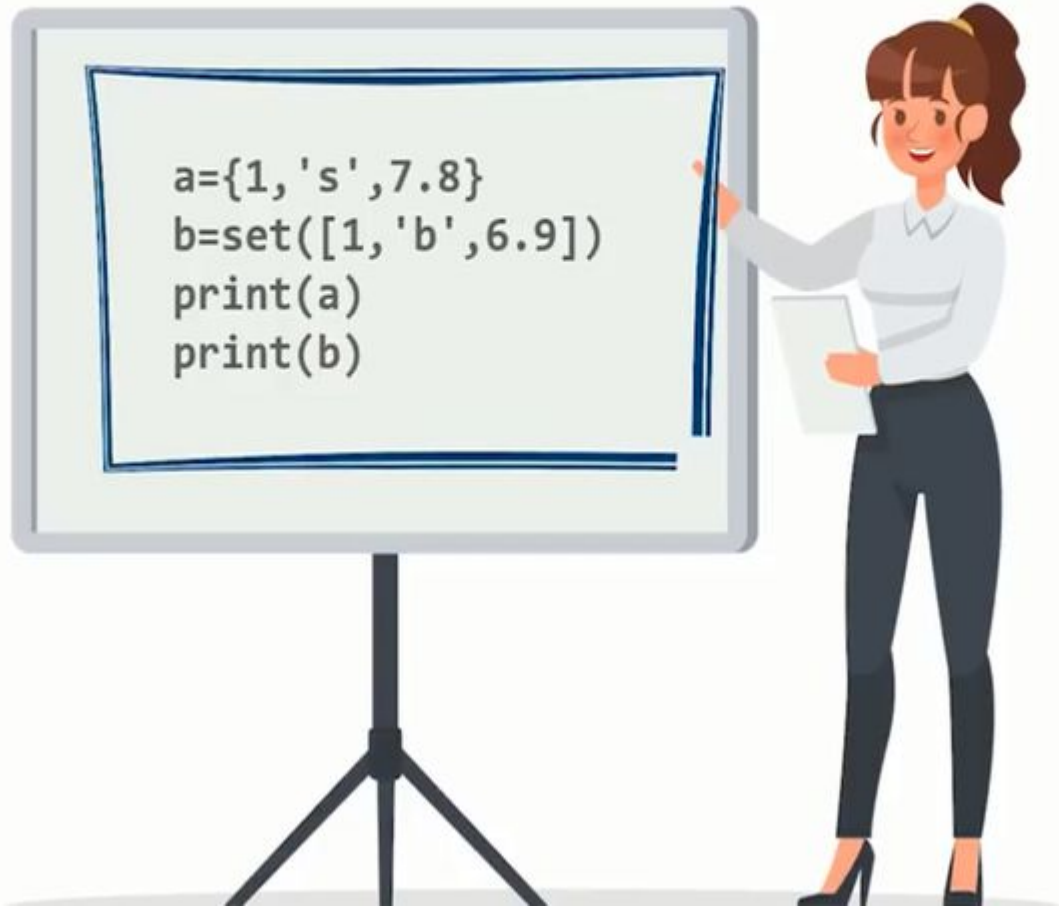


Where to use sets?

- A set is used where order of data is does not matter.

- Unique data element are needed.

How to create Sets in Python?



Two method to create a sets in python

1.Using curly braces {}

A set is created by placing all the elements inside curly braces {} separated with comma e.g:

```
#creating a set
numberSet = {1,2,3,4,3,2}
print(numberSet)
```

```
{1, 2, 3, 4}
```

2.Using set constructor set()

A set is created by using the built-in function set() for e.g:

```
numberset2=set([1,2,3,4,3,2])
print(type(numberset2))
Output:
<class 'set'>
```

How to create an empty set?

```
#creating an empty set
emptySet = {}   #This creates a dictionary
print(type(emptySet))
```

```
<class 'dict'>
```

This will create a dictionary instead of sets

So empty set is create by set method

```
emptySet = set()   #This creates a empty set
print(type(emptySet))
```

```
<class 'set'>
```

A set can contain elements of different type

```
# set of mixed datatypes
my_set = {1.0, "Hello", (1, 2, 3)}
print(my_set)
```

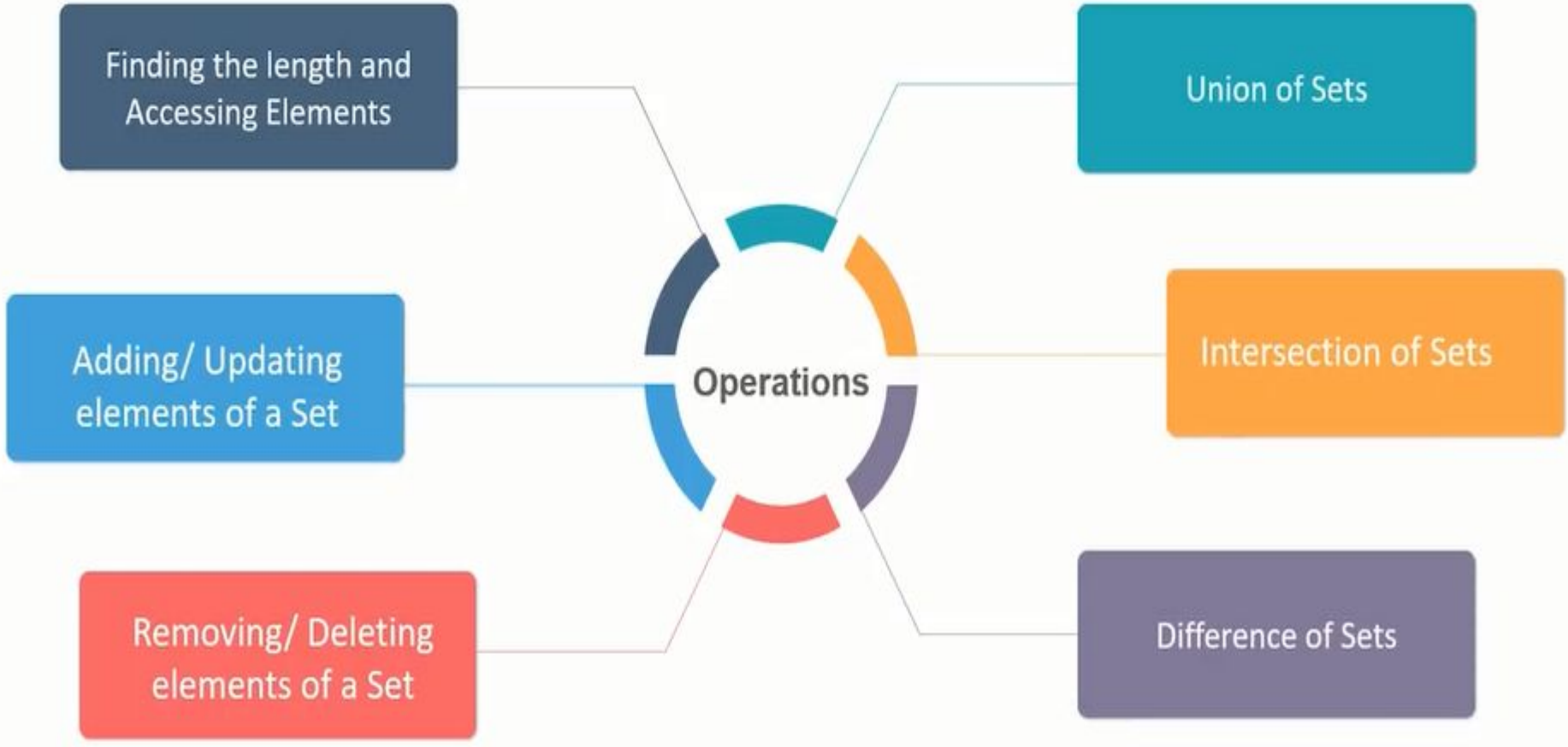
```
{1.0, 'Hello', (1, 2, 3)}
```

We can convert a list to set using set function

```
set_with_lists = set([1,2,3])
print(type(set_with_lists))
print(set_with_lists)
```

```
<class 'set'>
{1, 2, 3}
```

Python Set Operations



Finding the length of a Set

- Length of a set is the number of elements that are actually present in that set.
- You can make use of **len()** function to achieve this.

Example-

```
My_Set={1,'s',7.8}
len(My_Set)
```

Accessing Elements of a Set

- Set elements cannot be accessed using the index numbers
- You can access the elements of a set by looping through it

Example-

```
My_Set={1,'s',7.8}
for x in My_Set:
    print(x)
```

Finding length of set

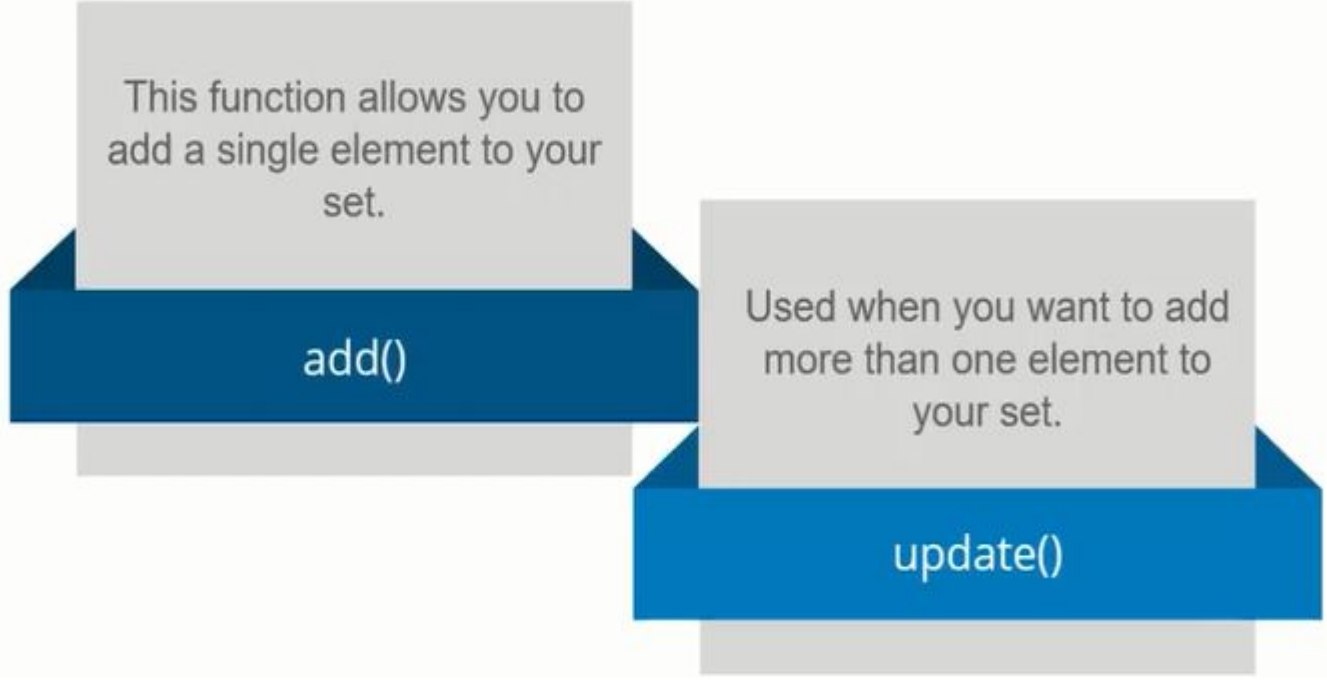
```
set1={1,2.3,"student"}
print(len(set1))
3
```

Accessing element of set

```
set1={1,2.3,"student"}
for i in set1:
    print(i)
Output:
2.3
1
student
```

Adding Elements to a Set

Elements can be added to a set using these two functions-



Add method:

We can add single element using the add() method

```
my_set=set()
my_set.add(2)
print(my_set)
Output:
{2}
```

Update method:

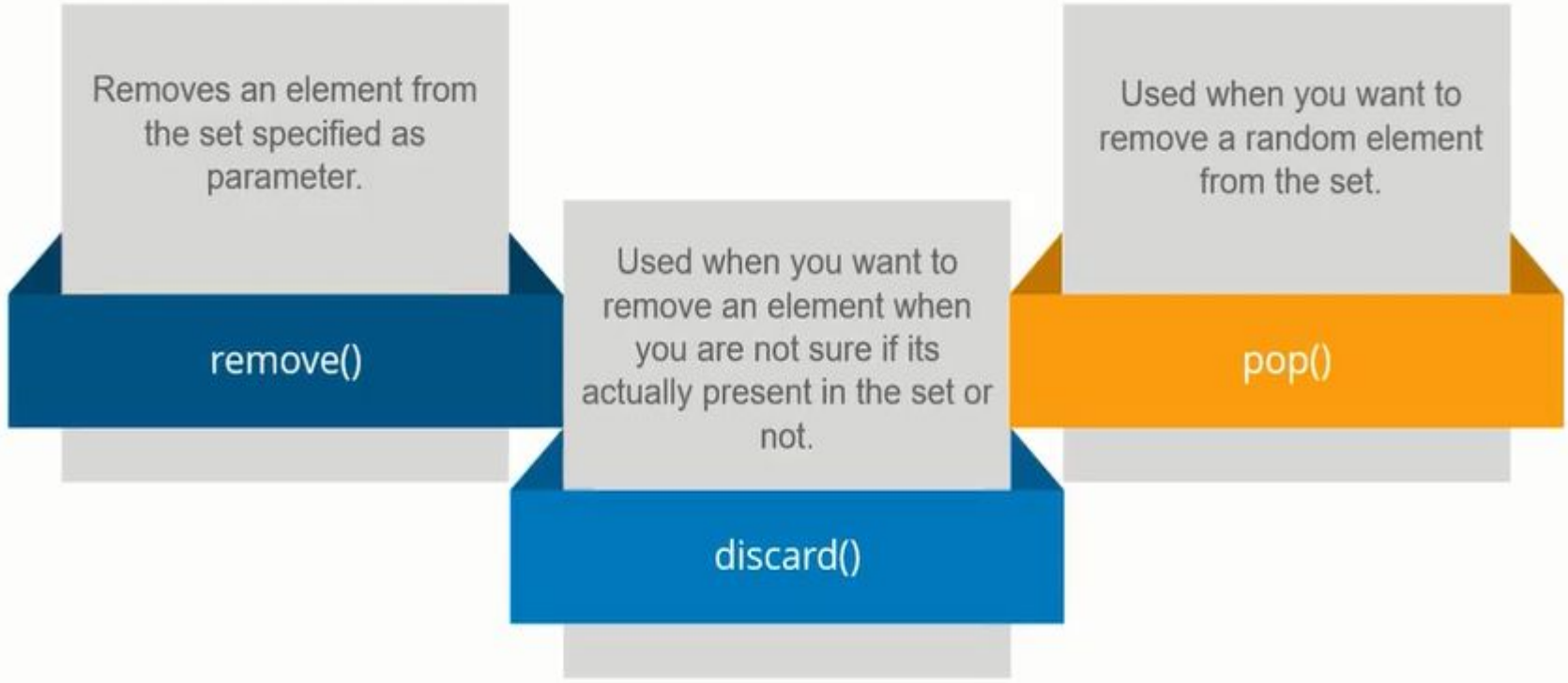
multiple elements using the update() method , The update() method can take tuples, lists, strings or other sets as its argument.

```
my_set = set()   #empty set
my_set.update([9 , 12])
my_set.update((3,5))
my_set.update("SIKANDER")

print(my_set)
```

```
{3, 5, 9, 'A', 12, 'N', 'E', 'S', 'D', 'R', 'I', 'K'}
```

Elements can be removed from a set using these functions-



Removing Elements from a Set

remove method:

A particular item can be removed from set

```
my_set={3,5,9,12,"INDIA","BHARAT"}
my_set.remove(9)
print(my_set)
Output:
{'INDIA', 3, 5, 12, 'BHARAT'}
```

discard method:

A particular item can be removed from set using discard() if the item does not exist in the set, it remains unchanged. But remove() will raise an error in such condition.

```
my_set={3,5,9,12,"INDIA","BHARAT"}
my_set.discard(3)
print(my_set)
Output:
{5, 9, 12, 'INDIA', 'BHARAT'}
```

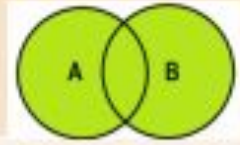
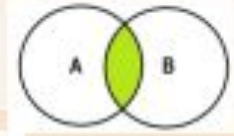
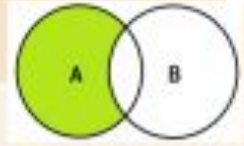
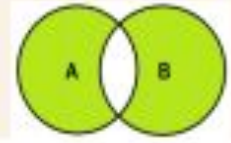
pop method:

A random item can be removed from set

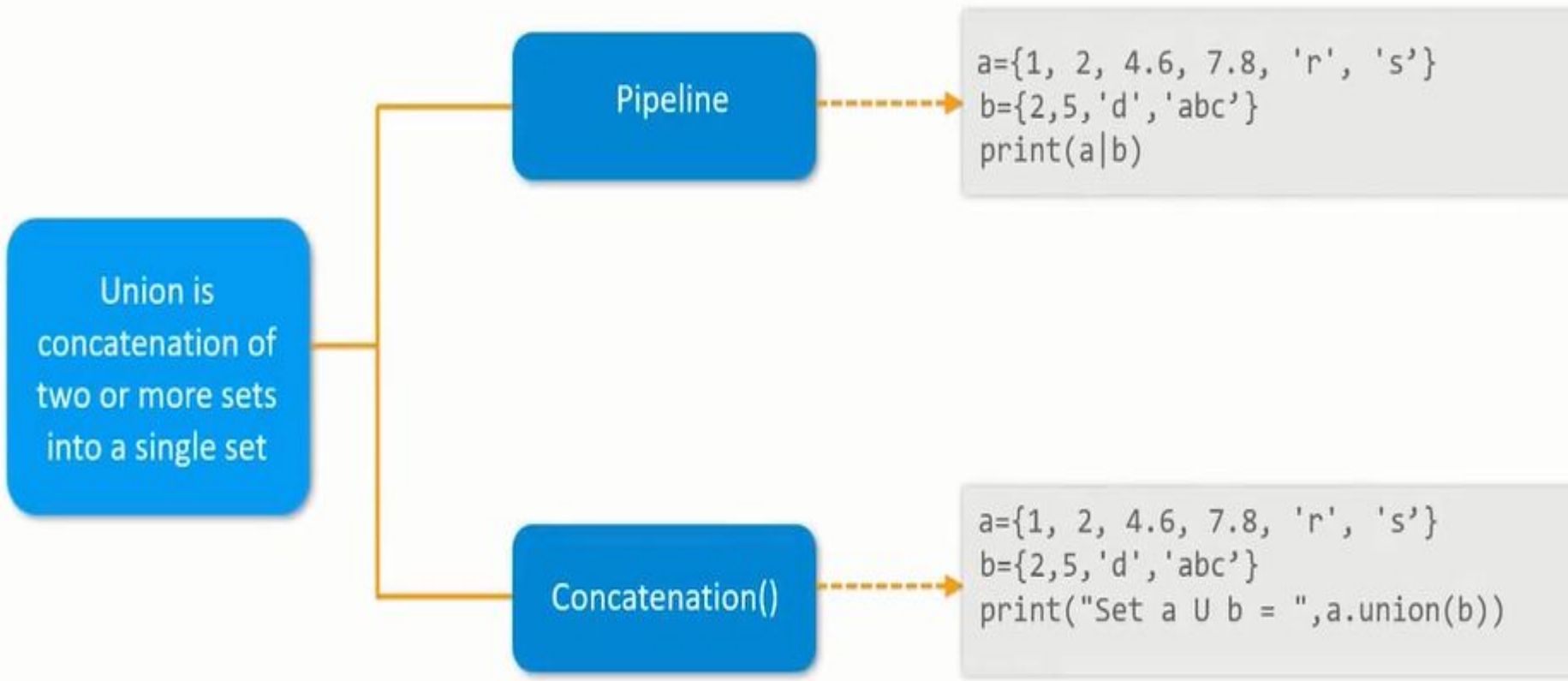
```
my_set={3,5,9,12,"INDIA","BHARAT"}
my_set.pop()
print(my_set)
Output:
{'INDIA', 3, 5, 9, 12}
```

Python Set Operations

- Sets can be used to carry out mathematical set operations like union, intersection, difference and symmetric difference.
- We can do this with operators or methods.

Method	Operator	
union	|	
intersection	&	
difference	-	
symmetric_difference	^	

Union of Sets



Using | operator:

Concatenation of set A and set B without duplicate elements

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(A|B)
Output:
{1, 2, 3, 4, 5, 6, 7, 8}
```

Using union method:

Concatenation of set A and set B without duplicate elements

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(A.union(B))
Output:
{1, 2, 3, 4, 5, 6, 7, 8}
```

Intersection of two or more sets forms a new set consisting of only the common elements present in those sets.

- Using '&' symbol
- Using intersection()

Example-

```
a={1, 2,5, 4.6, 7.8, 'r', 's'}
b={2,5,'d','abc'}
print(a&b)
print("Set a intersection b =    ",a.intersection(b))
```

Intersection of Sets

Using & operator:

Concatenation of common elements of set A and set B

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(A&B)
Output:
{4, 5}
```

Using intersection method:

Concatenation of common elements of set A and set B

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(A.intersection(B))
Output:
{4, 5}
```

Difference of Sets

The difference of sets produces a new set consisting of elements that are present only in one of those sets.

- ✓ Using '-' symbol
- ✓ Using difference()

Example-

```
a={1, 2,5, 4.6, 7.8, 'r', 's’}
b={2,5,'d','abc’}
Print(a-b)
print("Set a - b = ",a.difference(b))
```

Using - operator:

Difference Produce a set which content elements that are only in one of those set

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(A-B)
Output:
{1, 2, 3}
```

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(B-A)
Output:
{8, 6, 7}
```

Using difference method:

Difference Produce a set which content elements that are only in one of those set

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(A.difference(B))
Output:
{1, 2, 3}
```

```
A={1,2,3,4,5}
B={4,5,6,7,8}
print(B.difference(A))
Output:
{8, 6, 7}
```

Frozen Sets

- A frozen set in Python is a set whose values cannot be modified
- Frozen sets can be created using the frozenset() method

Example-

```
a={1, 2,5, 4.6, 7.8, 'r', 's’}
b=frozenset(a)
print(b)
```

frozenset method:

frozenset produce a set which is immutable i.e value cannot be modified

```
A={1,2,3,4,5}
B=frozenset(A)
print(type(B))
Output:
<class 'frozenset'>
print(B.add(3))
Output:
AttributeError: 'frozenset' object has no attribute 'add'
```

THANK YOU