**SET DATA STRUCTURE**

* A Python Set is the unordered and unique collection of the data.
* Sets can contain elements of different Data Types Like int, string, Tuple, etc but can not contain List, sets, dictionaries.
* Set data structure is used when you want each element to be unique in a large collection of data.

**Properties of set datatype**:

 1.unordered

 2.immutable

 3.unique

**1.unordered**

Set will store the element in an unordered manner i.e items in the set are not having any defined order such that no direct access to the element is possible. We use an iterator to iterate over all elements of the set.

**2.immutable**

set elements are immutable i.e can’t be changed but the set is mutable which means we can insert and delete elements from the set.

**3.unique**

The frequency of each element in the set is always one.that means no duplicates are allowed to add to a set.even though you add the duplicates to set it will not be entered into the set created.

**INITIALIZATION OF SET :**

 Set can be initialized in two ways

 1.using set function

 2.using curly braces { }

**Using set function:**

**Initialization** : using the inbuilt set function

**Syntax:** set name = set(iterable element)

Ex: s = set( )

INPUT

### #creating a list for iterable items

### l=[10,20,30,40,50,56,60]

### #creating a set from list l

### s=set(l)

### print(type(s))

### print(type(l))

### print(s)

### OUTPUT :<class 'set'>

### <class 'list'>

### {40, 10, 50, 20, 56, 60, 30}

=============================================

 **Using curly braces:**

**Syntax: variable={element1,element2,........element n}**

Example :

 INPUT

### #creating a set

### s={10,20,30,40,50}

### print(type(s))

### print(s)

OUTPUT

### <class 'set'>

### {50, 20, 40, 10, 30}

While using this type of initialisation even if you give the duplicate elements it will not be taken into the set elements.

Example :

 INPUT

### s={10,20,30,40,50,20}

### m={12,14,16,18,20,12,18}

### print("The Elements in set s are:",s)

### print("The Elements in set s are:",m)

OUTPUT:

The Elements in set s are: {50, 20, 40, 10, 30}

The Elements in set s are: {16, 18, 20, 12, 14}

======================================================

**Some important points**:

1.observe the elements order in set s and set m in both the input and output as we can clearly see that there is no order i.e unordered one of the properties of set .

2. Observe set m we have given the duplicate values but after the printing we get only unique elements in the set i.e unique elements one of the properties of set

**Functions in set**

**1. add:** This isused toadd a single element to the set

**syntax: setname.add(value)**

Example:

INPUT

### s={10,20,30,40,50}

### print("Set before adding the elements:",s)

### s.add(90)

### s.add(98)

### print("Set after adding the elements:",s)

OUTPUT

### Set before adding the elements: {50, 20, 40, 10, 30}

### Set after adding the elements: {50, 98, 20, 90, 40, 10, 30}

===================================================

2. **Update** : This is used to add multiple elements to a set at a time i.e iterable elements like list elements to the set .

**Syntax: (setname.update(iterable elements ) )**

INPUT

### s={10,20,30,40,50}

### l=[90,80,60,70,50]

### #updating the set using a iterable data (list)

### s.update(l)

### print("Set after updating using the list:",s)

### #updating the list using range

### s.update(range(1,10,4))

### print("Set after updating using the range:",s)

OUTPUT

#### Set after updating using the list: {70, 10, 80, 20, 90, 30, 40, 50, 60}

#### Set after updating using the range: {1, 5, 70, 9, 10, 80, 20, 90, 30, 40, 50, 60}

============================================================

3. **pop**: used to remove and return a random element from the set

 **Syntax: (setname.pop( ) )**

Example:

INPUT

### s={10,2,3,6,8,5,9,18}

### print(s.pop())

### print(s.pop())

### print("Set after removing the elements:",s)

OUTPUT

### 2

### 3

### Set after removing the elements: {5, 6, 8, 9, 10, 18}

=======================================================

4. **Remove:** This is used to remove a specific element from the set .If the specified element is not present in the set we get a key error.

**Syntax: (setname.remove(value to remove))**

Example:

INPUT

### s={10,2,3,6,8,5,9,18}

### s.remove(6)

### s.remove(5)

### print("Set after removing the elements:",s)

OUTPUT

### Set after removing the elements: {2, 3, 8, 9, 10, 18}

===================================================

5.**discard** :This is used to remove the specified element from the set.if the specified element is not present we don't get any key value error

**Syntax: (setname.discard(value to remove))**

Example:

INPUT

## s={10,2,3,6,8,5,9,18}

## s.discard(6)

## #below element that we want to remove is not present in set s

## s.discard(17)

## print("Set after removing the elements:",s)

OUTPUT

## Set after removing the elements: {2, 3, 5, 8, 9, 10, 18}

==================================================

6. **Clear**: This is used to remove all the items from the set

**Syntax: setname.clear()**

Example:

INPUT

## s={10,2,3,6,8,5,9,18}

## s.clear()

## print("The items in set after using clear :",s)

OUTPUT

## The items in set after using clear : set()

=================================================

7. **Copy**: This is used to create a clone object or create a cop[y of the set

**Syntax: variable=setname.copy()**

Example:

INPUT

## s={10,2,3,6,8,9,5,18}

## m=s.copy()

## print("copied set is :",m)

OUTPUT

## copied set is : {2, 3, 5, 6, 8, 9, 10, 18}