Set in Python

Set Creation

Set in Python

- In Python, a set is an **unordered collection of unique elements**.
- A set is mutable, meaning elements can be added or removed after creation, but **duplicate values are not allowed**.
- A set is written as a collection of comma-separated values enclosed in curly brackets { }.

Syntax: var = {value1, value2, value3,....} **or**

var = set(value1, value2, value3,....)

Example: "tupledemo.py"

```
s1 = set()
s2 = {123,"python", 3.7}
s3 = {1, 2, 3, 4, 5, 6}
s4 = {"C",}
print(t1)
print(t2)
print(t3)
print(t4)
```

Output:

```
python tupledemo.py
Set()
{123, 'python', 3.7}
{1, 2, 3, 4, 5, 6}
{'C',}
```

set Indexing

set Indexing in Python

❌ Sets do NOT support indexing
❌ You cannot access elements using index numbers like [0], [1]
❌ You cannot use slicing like [1:3]
✓ Because sets are unordered.

Example:

mytuple=('banana','apple','mango','tomato','berry')

mytuple[0]="banana"
mytuple[2]="mango"

mytuple[1:3]=["apple","mango"]

👉This gives error:

TypeError: 'set' object is not subscriptable

Set Operators

Set Operators in Python

- In Python, we can perform various mathematical operations on python sets like union, intersection, difference, etc

Operator	Description
|	Union Operator
&	Intersection Operator
-	Difference Operator

Set Operators in Python

cont...

☞ Union (|) Operator

The union of two sets are calculated by using the or (|) operator. The union of the two sets contains the all the items that are present in both the sets.

Example:

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
print(Days1 | Days2)
```

Output:

```
python setdemo.py

{'Thr', 'Fri', 'Sun', 'Tue', 'Wed', 'Mon',
'Sat'}
```

Set Operators in Python

cont...

☞ Intersection (&) Operator

The & (intersection) operator is used to calculate the intersection of the two sets in python. The intersection of the two sets are given as the set of the elements that common in both sets.

Example:

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
print(Days1 & Days2)
```

Output:

```
python setdemo.py

{'Mon', 'Sat'}
```

Set Operators in Python

cont...

☞ Difference (-) Operator

The & (intersection) operator is used to calculate the intersection of the two sets in python. The intersection of the two sets are given as the set of the elements that common in both sets.

Example:

```
Days1={"Mon","Tue","Wed","Sat"}
Days2={"Thr","Fri","Sat","Sun","Mon"}
print(Days1 & Days2)
```

Output:

```
python setdemo.py

{'Mon', 'Sat'}
```

Set Functions & Methods

Set Functions & Methods in Python

- Python provides various in-built functions and methods which can be used with set. Those are

•len(set)
•max(set)
•min(set)
•sum(set)

•sorted(set)
•set()
•add()
•update()

•discard()
•remove()
•pop()
•clear()

•union()
•intersection()
•difference()
•issubset()
•issuperset()

☞ len():

- In Python, **len()** function is used to find the length of set,i.e it returns the number of items in the set.

Syntax: `len(set)`

Example: **lendemo.py**

```
num={1,2,3,4,5,6}
print("length of Set :",len(num))
```

Output:

python lendemo.py
length of Set : 6

Set Functions & Methods in Python

Cont..

☞ max ():

- In Python, max() function is used to find maximum value in the set.

Syntax: `max(set)`

Example: **maxdemo.py**

```
t1={1,2,3,4,5,6}
t2={'java','c','python','cpp'}
print("Max of set t1 :",max(t1))
print("Max of set t2 :",max(t2))
```

Output:

python maxdemo.py
Max of set t1 : 6
Max of set t2 : python

Set Functions & Methods in Python

Cont..

☞ min ():

- In Python, max() function is used to find minimum value in the set.

Syntax: `max(set)`

Example: **mindemo.py**

```
t1={1,2,3,4,5,6}
t2={'java','c','python','cpp'}
print("Max of set t1 :",min(t1))
print("Max of set t2 :",min(t2))
```

Output:

python mindemo.py
Max of set t1 : 1
Max of set t2 : c

Set Functions & Methods in Python

Cont..

☞ sum ():

- In python, sum(set) function returns sum of all values in the set. Set values must in number type.
- **Syntax:** `sum(set)`

Example: **sumdemo.py**

```
t1={1,2,3,4,5,6}
print("Sum of set t1 :",sum(t1))
```

Output:

python sumdemo.py
Sum of set t1 : 21

Set Functions & Methods in Python

Cont..

☞ sorted ():

- In python, sorted (set) function is used to sort all items of set in an ascending order. It also sorts the items into descending and ascending order. It takes an optional parameter 'reverse' which sorts the set into descending order.
- **Syntax:** `sorted(set)`

Example: **sortedemo.py**

```
t1={1,3,2,4,6,5}
print(" result of sorted :",sorted(t1))
```

Output:

python sortedemo.py
result of sorted : {1, 2, 3, 4, 5, 6}

Set Functions & Methods in Python

Cont..

☞ set ():

- The set() method takes sequence types and converts them to sets. This is used to convert a given string or list or tuple into set.
- **Syntax:** `set(sequence)`

Example: **sortedemo.py**

```
t1={1,3,2,4,6,5}
t2="PYTHON"
print(" result of set :",set(t1))
print(" result of set :",set(t2))
```

Output:

python sortedemo.py
result of set : {1, 2, 3, 4, 5, 6}
result of set : {'N', 'O', 'T', 'H', 'P', 'Y'}

Set Functions & Methods in Python

Cont..

☞ add ():

- In python, the add() method used to add some particular item to the set.
- **Syntax:** `add(item)`

Example: **addemo.py**

```
t1={1,3,2,4,6,5}

t1.add(7)

print(" result of set :",t1))
```

Output:

python addemo.py

result of set : {1, 2, 3, 4, 5, 6,7}

Set Functions & Methods in Python

Cont..

☞ update ():

- Python provides the update () method to add more than one item in the set.
- **Syntax:** `update([item1,item2,.... itemn)`

Example: **updatedemo.py**

```
t1={1,3,2,4,6,5}

t1.update([7,8,9])

print(" result of set :",t1)
```

Output:

python updatedemo.py
result of set : {1, 2, 3, 4, 5, 6,7,8,9}

Set Functions & Methods in Python

Cont..

☞ discard ():

- Python provides discard () method which can be used to remove the items from the set. If item doesn't exist in the set, the python will not give the error. The program maintains its control flow.
- **Syntax:Syntax:** `discard(item)`

Example: **discardemo.py**

```
t1={1,3,2,4,6,5}

t1.discard(4)

print(" result of set :",set(t1))
```

Output:

python discardemo.py
result of set : {1, 2, 3, 5, 6,}

Set Functions & Methods in Python

Cont..

☞ remove ():

- Python provides remove () method which can be used to remove the items from the set. If item doesn't exist in the set, the python will give the error.
- **Syntax:** `remove(item)`

Example: **removedemo.py**

```
t1=[1,3,2,4,6,5]

t1.remove(5)

print(" result of set :",set(t2))
```

Output:

python removedemo.py
result of set : {1, 2, 3, 4, 5, 6}

Set Functions & Methods in Python

Cont..

☞ pop ():

- In Python, pop () method is used to remove the item. However, this method will always remove the last item.
- **Syntax:** `set_name.pop()`

Example: **popdemo.py**

```
t1={1,3,2,4,6,5}
print(" set t1 :",t1)
t1.pop()
print("POP :",t1)
```

Output:

python popdemo.py
set t1 : {1,3,2,4,6,5}
POP : {2, 1, 5,6, 4}

Set Functions & Methods in Python

Cont..

☞ clear ():

- In Python, clear () method is used to remove the all items in set.
- **Syntax:** `set_name.clear()`

Example: **cleardemo.py**

```
t1=[1,3,2,4,6,5]
t1.clear()
print(" result of set :",t1)
```

Output:

python cleardemo.py
result of set : set()

Set Functions & Methods in Python

☞ union ():

- In Python, the union () method is used to perform union of two sets. The union of the two sets contains the all the items that are present in both the sets.
- **Syntax:** `set1.union(set2)`

Example: **udemo.py**

```
t1={1,3,2,4,6,5}

t2={7,3,4,5,2}

print(" union :",t1.union(t2))
```

Output:

python udemo.py
union : {2,3,1,4,5,6,7}

Set Functions & Methods in Python

☞ intersection ():

- In Python, the intersection () is used to calculate the intersection of the two sets in python. The intersection of the two sets is given as the set of the elements that common in both sets.
- **Syntax:** `set1.intersection(set2)`

Example: **idemo.py**

```
t1={1,3,2,4,6,5}

t2={7,3,4,5,2}

print("intersection :",t1.intersect(t2))
```

Output:

python idemo.py
intersection : {3,2,5,4}

Set Functions & Methods in Python

Cont..

☞ difference ():

- The difference of two sets can be calculated by using the difference () method. The resulting set will be obtained by removing all the elements from set 1 that are present in set 2.
- **Syntax:** `set1.difference(set2)`

Example: **difdemo.py**

```
t1={1,3,2,4,6,5}

t2={7,3,4,5,2}

print(" result of set :",t1.difference(t2))
```

Output:

python difdemo.py

union : {1,6}

Set Functions & Methods in Python

Cont..

☞ issubset ():

- The difference of two sets can be calculated by using the difference () method. The resulting set will be obtained by removing all the elements from set 1 that are present in set 2.
- **Syntax:** `set1.issubset(set2)`

Example: **subsetdemo.py**

```
t1={1,3,2,4,6,5}
t2={3,4,5,2}
print(" issubset :",t2.issubset(t1))
```

Output:

python subsetdemo.py
issubset : True

Set Functions & Methods in Python

Cont..

☞ issuperset ():

- The issubset() method returns True if all elements of a set are present in another set (passed as an argument). If not, it returns False.
- **Syntax:** `set1.issuperset(set2)`

Example: **supersetdemo.py**

```
t1={1,3,2,4,6,5}

t2={3,4,5,2}

print(" issuperset :",t1.issuperset(t2))
```

Output:

python supersetdemo.py

issuperset : True