Python - Set

What is Set? Full explanation.

--> set is a data structure which is called collection of items, in which we can represent a group of unique value as a single entity.

syntax:

```
    set-name = {item1, item2, item3,......,item n}
```

Note:

1. We write the items of set inside the curly braces "{}"
2. insertion order is not preserved.
3. Indexing & Slicing not work.
4. Heterogeneous elements are allowed.
5. mutable in nature.

```
# Empty Set
var = {}
print(type(var))
<class 'dict'>
```

```
# perfact Empty set
var = set()
print(type(var))
<class 'set'>
```

```
# value assign in set variable
var = {10,'ani',56.7,True}
print(type(var))
print(var)
{56.7, 'ani', 10, True}
Note: - value not print in order wise.
```

```
# indexing is not Allowed in Set.
var = {10,'ani',56.7,True}
print(type(var))
print(var[1])
TypeError: 'set' object is not subscriptable
```

```
# Traversing a Set.
s = {1,2.5, “Hello”, (1,2)}
for item in s:
      print(item)
```

```
# clear() method
var = {10,'ani',56.7,True,'ani'}
var.clear()
print(var)
```

```
# Unique Value assign a Set.
var = {10,'ani',56.7,True,'ani'}
print(type(var))
print(var)
<class 'set'>
{56.7, True, 10, 'ani'}
```

```
# Mutable operation:
#  add() method
var = {10,'ani',56.7,True,'ani'}
var.add("Code")
print(var)
{'Code', True, 10, 'ani', 56.7}
```

Note: - set.add() takes exactly one argument.

```
#  update() method
var = {10,'ani',56.7,True,'ani'}
var.update("Code")
print(var)
{True, 'C', 'e', 'ani', 10, 'o', 'd', 56.7}
```

Note: - update() method used to insert more than one parameter value.

```
# Update() method Example
var = {10,'ani',56.7,True,'ani'}
var.update(["Code","block"])
print(var)
{True, 'block', 'Code', 10, 56.7, 'ani'}
```

```
#  pop() method
var = {10,'ani',56.7,True,'ani',56.7}
print(var.pop())
56.7
```

```
#  remove() method
var = {10,'ani',56.7,True,'ani',56.7}
var.remove("ani")
print(var)
{56.7, True, 10}
```

```
#  discard() method
var = {1,2,3}
var.discard(4)
print(var)



#  copy() method
var1 = {1,2,3}
var2 = var1.copy()
print(var2)
```

```
#  //   mathametical Method of python Set

# union() method
var = {10,'ani',45.3,True}
var2 = {10,'ani',45.3,True,'Cody',12,False}
print(var.union(var2))
{False, True, 10, 12, 45.3, 'ani', 'Cody'}
```

```
# union() method achieve by operator
var = {10,'ani',45.3,True}
var2 = {10,'ani',45.3,True,'Cody',12,False}
print(var | var2)
{'ani', True, False, 10, 12, 45.3, 'Cody'}
```

```
# intersection() method
var = {10,'ani',45.3,True}
var2 = {10,'ani',45.3,True,'Cody',12,False}
print(var.intersection(var2))
{'ani', True, 10, 45.3}
```

```
# intersection() method achieve by operator
var = {10,'ani',45.3,True}
var2 = {10,'ani',45.3,True,'Cody',12,False}
print(var & var2)
{True, 10, 'ani', 45.3}
```

```
# difference() method
var = {10,'ani',45.3,True,'Cody',12,False}
var2 = {10,'ani',45.3,True}
print(var.difference(var2))
{False, 'Cody', 12}
```

```
# difference() method achieve by operator
var = {10,'ani',45.3,True,'Cody',12,False}
var2 = {10,'ani',45.3,True}
print(var - var2)
{False, 'Cody', 12}
```

```
# symmetric difference() method
var = {10,'ani',45.3,True,'Cody',12,False}
var2 = {10,'ani',45.3,True,'india',17.3}
print(var.symmetric_difference(var2))
{False, 12, 'india', 17.3, 'Cody'}
```

```
# issubset() method
a = {1, 2}
b = {1, 2, 3}
print(a.issubset(b))


# issuperset() method
a = {1, 2, 3}
b = {1, 2}
print(a.issuperset(b))


# isdisjoint() method
a = {1, 2, 5}
b = {3, 4, 6}
print(a.isdisjoint(b)) # True (no common element)



# Set Comprehension
squares  = { x*x for x in range(1, 6)}
print(squares)
```