

Set in Python

- ⇒ Distinct Element
- ⇒ Unordered
- ⇒ No Indexing
- ⇒ Union, Intersection, Set Difference, etc. are fast
- ⇒ Uses hashing internally

```
s1 = {10, 20, 30}
print(s1)
s2 = set([20, 30, 40])
print(s2)
s3 = {}
print(type(s3))
s4 = set()
print(type(s4))
print(s4)
```

O/p: {10, 20, 30}
{40, 20, 30}
<class, 'dict'>
<class, 'set'>
set()

```
s = {10, 20}
s.add(30)
print(s)
s.add(30)
print(s)
s.update([40, 50])
print(s)
s.update({60, 70}, [80, 90])
print(s)
```

O/p: {10, 20, 30}
{10, 20, 30}
{40, 10, 50, 20, 30}
{70, 40, 10, 30, 50, 60, 80, 90, 20}

```
s = {10, 20, 30, 40}
print(len(s))
print(20 in s)
print(50 in s)
```

O/p: 4
True
False

```
s = {10, 30, 20, 40}
s.discard(30)
print(s)
s.remove(20)
print(s)
s.clear()
print(s)
s.add(50)
del s
```

O/p: {40, 10, 20}
{40, 10}
set()

```
s1 = {2, 4, 6, 8}
s2 = {3, 6, 9}
print(s1 | s2) # Union
print(s1 & s2) # Intersection
print(s1 - s2) # Difference
print(s1 ^ s2) # Symmetric Difference
```

O/p: {2, 3, 4, 6, 9, 8}
{6}
{8, 2, 4}
{2, 4, 3, 9, 8}

```
s1 = {2, 4, 6, 8}
s2 = {4, 8}
print(s1.isdisjoint(s2))
print(s1 <= s2) # issubset()
print(s1 < s2) # propersub...
print(s1 >= s2) # issuperset()
print(s1 > s2) # propersup...
```

O/p: False
False
False
True
True