

# Sets

A Set is an unordered collection data type that is iterable, mutable, and has no duplicate elements. Python's set class represents the mathematical notion of a set. This is based on a data structure known as hash table.

Contains diff data types,  
Mutable,  
No duplicates are allowed,  
No indexing

In [1]: *# Defining an empty set*

```
set_var=set()
print(set_var)
print(type(set_var))
```

```
set()
<class 'set'>
```

In [2]: set\_var={1,2,3,4,3}

In [3]: set\_var *#No duplicate element in set*

Out[3]: {1, 2, 3, 4}

In [4]: set\_var={"Avengers", "IronMan", "Hitman"}  
print(set\_var)  
print(type(set\_var))

```
{'IronMan', 'Avengers', 'Hitman'}
<class 'set'>
```

In [5]: *#Indexing*  
set\_var[0] *# Set doesn't support indexing*

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20940\2684890553.py in <module>
      1 #Indexing
----> 2 set_var[0]          # Set doesn't support indexing

TypeError: 'set' object is not subscriptable
```

In [6]: set\_var["HitMan"] *# Set doesn't support indexing*

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20940\2596609646.py in <module>
----> 1 set_var["HitMan"]    # Set doesn't support indexing

TypeError: 'set' object is not subscriptable
```

In [7]: *# Inbuilt functions in sets*

```
set_var.add("Hulk")  
print(set_var)
```

```
{'IronMan', 'Avengers', 'Hitman', 'Hulk'}
```

In [8]: 

```
set1= {"Avengers","IronMan","Hitman"}  
set2= {"Avengers","IronMan","HitMan","Hulk2"}
```

In [9]: *#set difference*  

```
set2.difference(set1)           #Just like sets in Maths
```

Out[9]: {'HitMan', 'Hulk2'}

In [10]: 

```
set2           #No change in set2 even after set diff operation
```

Out[10]: {'Avengers', 'HitMan', 'Hulk2', 'IronMan'}

In [11]: *# Difference Update*  

```
set2.difference_update(set1)
```

In [12]: 

```
set2           #set2 got updated
```

Out[12]: {'HitMan', 'Hulk2'}

In [19]: 

```
set1= {"Avengers","IronMan","HitMan"}  
set2= {"Avengers","IronMan","HitMan","Hulk2"}
```

In [16]: 

```
set2.intersection(set1)           # returns common element in set1 and set2
```

Out[16]: {'Avengers', 'HitMan', 'IronMan'}

In [17]: 

```
set2.intersection_update(set1)
```

In [18]: 

```
set2
```

Out[18]: {'Avengers', 'HitMan', 'IronMan'}

In [20]: 

```
set2.union(set1)           # returns every elements of set1 and set2
```

Out[20]: {'Avengers', 'HitMan', 'Hulk2', 'IronMan'}