SET

In [ ]:
```
s = {}
```

In [ ]:
```
s
```

In [55]:
```
s
type(s)
```

Out[55]:
```
set
```

In [ ]:
```
s = set()
print(s)
print(type(s))
```

In [56]:
```
s2 = {1,2,4}
s2
```

Out[56]:
```
{1, 2, 4}
```

In [59]:
```
s3 = {1,4,"Sowji",1+5j,4.0} # Set will not allow duplicate values even though ty
```

In [58]:
```
s3
```

Out[58]:
```
{(1+5j), 1, 4, 'Sowji'}
```

In [60]:
```
print(s1)
print(s2)
print(s3)
```

```
{1, 2, 4, 5}
{1, 2, 4}
{(1+5j), 1, 'Sowji', 4}
```

In [67]:
```
s1 = s3
print(id(s1))
print(id(s3))
print(s1 == s3)
```

```
2158027436352
2158027436352
True
```

In [63]:
```
s3.add(5.5)
```

In [64]:
```
s1
```

Out[64]:
```
{(1+5j), 1, 4, 5.5, 'Sowji'}
```

In [65]:
```
s3
```

Out[65]:
```
{(1+5j), 1, 4, 5.5, 'Sowji'}
```

In [70]:
```
s4 = {1, 2.3, 'nit', 1+2j, [1,2,3], (4,5,6), True} # Set will not allow nested l
s4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[70], line 1
----> 1 s4 = {1, 2.3, 'nit', 1+2j, [1,2,3], (4,5,6), True} # Set will not allow n
ested list or tuples. Whereas we can have set,tuple,dic inside a tuple or list
      2 s4

TypeError: unhashable type: 'list'
```

In [71]:
```
s3
```

Out[71]:
```
{(1+5j), 1, 4, 5.5, 'Sowji'}
```

In [74]:
```
s3.clear() # clears all the elemtns in the data
```

In [73]:
```
s3
```

Out[73]:
```
set()
```

In [75]:
```
del s3 # deleted entire variable
```

In [76]:
```
s3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[76], line 1
----> 1 s3

NameError: name 's3' is not defined
```

In [78]:
```
s2
```

Out[78]:
```
{1, 2, 4}
```

In [82]:
```
s2.remove(4) # removes particular element
```

In [83]:
```
s2
```

Out[83]:
```
{1}
```

In [84]:
```
s2.add(5)
```

In [85]:
```
s2
```

Out[85]:
```
{1, 5}
```

In [86]:
```
s2.add(3)
```

In [87]:
```
s2
```

Out[87]:
```
{1, 3, 5}
```

In [88]:
```
s2.discard(6) # discard function will remove element if it is present otherwise
```

In [92]:
```
s2.remove(6) #if element was not present it will throw error
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[92], line 1
----> 1 s2.remove(6)

KeyError: 6
```

In [90]:
```
s2.discard(1)
```

In [95]:
```
s2.add(6)
```

In [96]:
```
s2
```

Out[96]:
```
{3, 4, 5, 6}
```

In [97]:
```
s2.pop()
```

Out[97]:
```
3
```

In [98]:
```
s2.pop()
```

Out[98]:
```
4
```

In [102…
```
s2.add(4.5)
```

In [103…
```
for i in s2:
    print(i)
```

```
4
5
6
Sowji
4.5
```

In [104…
```
for i in s2:
    print(i)
```

```
4
5
6
Sowji
4.5
```

In [105…
```
s2
```

Out[105…
```
{4, 4.5, 5, 6, 'Sowji'}
```

In [107…
```
for i in enumerate(s2):
    print(i)
```

```
(0, 4)
(1, 5)
(2, 6)
(3, 'Sowji')
(4, 4.5)
```

In [108…
```
"Sowji" in s2
```
Out[108… True

In [109…
```
s2
```
Out[109… {4, 4.5, 5, 6, 'Sowji'}

In [111…
```
s1
```
Out[111… set()

In [116…
```
s1.update(s2) # all elements in s2 will be copied to s1
```

In [113…
```
s1
```
Out[113… {4, 4.5, 5, 6, 'Sowji'}

In [115…
```
s2
```
Out[115… {4, 4.5, 5, 6, 'Sowji'}

SET Operations

- Union |

In [121…
```
s6 = {1,2,3,4,5}
s7 = {4,5,6,7,8}
s8 = {8,9,10}
```

In [122…
```
s6.union(s7)
```
Out[122… {1, 2, 3, 4, 5, 6, 7, 8}

In [123…
```
s6.union(s7,s8)
```
Out[123… {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [124…
```
s6 | s7
```
Out[124… {1, 2, 3, 4, 5, 6, 7, 8}

In [125…
```
s6 | s7 | s8
```
Out[125… {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [126…
```
s6 & s7
```
Out[126… {4, 5}

In [127…
```
s6.intersection(s7)
```
Out[127… {4, 5}

In [128…
```
s7.intersection(s8)
```

Out[128…
```
{8}
```

In [129…
```
print(s6)
print(s7)
print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [130…
```
s6.difference(s7)
```

Out[130…
```
{1, 2, 3}
```

In [131…
```
s6 - s7
```

Out[131…
```
{1, 2, 3}
```

In [132…
```
s7 - s8
```

Out[132…
```
{4, 5, 6, 7}
```

In [133…
```
print(s6)
print(s7)
print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [134…
```
s6.symmetric_difference(s7)
```

Out[134…
```
{1, 2, 3, 6, 7, 8}
```

In [ ]: