In [1]:

```
set1 = {"apple", "banana", "cherry"}
set2 = {1, 5, 7, 9, 3}
set3 = {True, False, False}
```

In [3]:

```
type(set3)
```

Out[3]:

```
set
```

In [5]:

```
thisset = set("apple", "banana", "cherry") # note the double round-brackets
print(thisset)
```

```
-------------------------
TypeErrorTraceback (most recent call last)
Input In [5], in <cell line: 1>()
----> 1 thisset = set("apple", "banana", "cherry") # note the double round-b
rackets
      2 print(thisset)

TypeError: set expected at most 1 argument, got 3
```

In [6]:

```
thisset = set(("apple", "banana", "cherry")) # note the double round-brackets
print(thisset)
```

```
{'apple', 'cherry', 'banana'}
```

In [7]:

```
thisset = {"apple", "banana", "cherry"}
print("banana" in thisset)
```

```
True
```

In [8]:

```
thisset = {"apple", "banana", "cherry"}
thisset.add("orange")
print(thisset)
```

```
{'apple', 'cherry', 'banana', 'orange'}
```

In [6]:

```
thisset = {"apple", "banana", "cherry"}
tropical = {"pineapple", "mango", "papaya"}

thisset.update(tropical)

print(thisset)
thisset
```

```
{'banana', 'papaya', 'mango', 'apple', 'pineapple', 'cherry'}
```

Out[6]:

```
{'apple', 'banana', 'cherry', 'mango', 'papaya', 'pineapple'}
```

In [27]:

```
s={'apple', 'banana', 'pineapple', 'mango', 'papaya', 'cherry'}
s
```

Out[27]:

```
{'apple', 'banana', 'cherry', 'mango', 'papaya', 'pineapple'}
```

In [8]:

```
thisset = {"apple", "banana", "cherry"}
mylist = ("kiwi", "orange")
thisset.update("apple")
print(thisset)
thisset
```

```
{'a', 'p', 'banana', 'apple', 'l', 'e', 'cherry'}
```

Out[8]:

```
{'a', 'apple', 'banana', 'cherry', 'e', 'l', 'p'}
```

In [40]:

```
thisset = {"dapple", "banana", "cherry"}
thisset.remove("banana")
print(thisset)
```

```
{'cherry', 'dapple'}
```

In [37]:

```
thisset = {"apple", "", "cherry"}

thisset.remove("banana")

print(thisset)
```

```
-------------------------
KeyErrorTraceback (most recent call last)
Input In [37], in <cell line: 3>()
      1 thisset = {"apple", "", "cherry"}
----> 3 thisset.remove("banana")
      5 print(thisset)

KeyError: 'banana'
```

In [38]:

```
thisset = {"apple", "banana", "cherry"}

thisset.discard("banana")

print(thisset)
```

```
{'apple', 'cherry'}
```

In [42]:

```
thisset = {"apple", "", "cherry"}

thisset.discard("banana")

print(thisset)
```

```
{'apple', '', 'cherry'}
```

In [11]:

```
thisset = { "banana","apple", "cherry"}
x = thisset.pop()
print(x)
print(thisset)
```

```
banana
{'apple', 'cherry'}
```

In [12]:

```
thisset = { "banana","apple", "cherry"}
x = thisset.pop()
print(x)
print(thisset)
```

```
banana
{'apple', 'cherry'}
```

In [76]:

```
thisset = {"apple", "banana", "cherry"}

thisset.clear()

print(thisset)
```

```
set()
```

In [77]:

```
thisset = {"apple", "banana", "cherry"}

del thisset

print(thisset)
```

```
-------------------------
NameErrorTraceback (most recent call last)
Input In [77], in <cell line: 5>()
      1 thisset = {"apple", "banana", "cherry"}
      3 del thisset
----> 5 print(thisset)

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

In [13]:

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

set3 = set1.union(set2)
print(set3)
```

```
{1, 2, 3, 'b', 'c', 'a'}
```

In [14]:

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}
set1.union(set2)
print(set1)
```

```
{'a', 'c', 'b'}
```

In [15]:

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

x.intersection_update(y)

print(x)
```

```
{'apple'}
```

In [90]:

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.intersection(y)

print(z)
```

```
{'apple'}
```

In [91]:

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

x.symmetric_difference_update(y)

print(x)
```

```
{'banana', 'microsoft', 'cherry', 'google'}
```

In [92]:

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.symmetric_difference(y)

print(z)
```

```
{'banana', 'microsoft', 'cherry', 'google'}
```

In [95]:

```
x = {"apple", "banana", "cherry", True}
y = {"google", 1, "apple", 2}

z = x.symmetric_difference(y)
z
```

Out[95]:

```
{2, 'banana', 'cherry', 'google'}
```

In [5]:

```
d={"apple":[1,2,3]}
d['apple'].append(4)
d
```

Out[5]:

```
{'apple': [1, 2, 3, 4]}
```

In [4]:

```
fruits = {"apple", "banana", "cherry"}

x = fruits.copy()

print(x)
x.pop()
print(fruits)
x
```

```
{'apple', 'banana', 'cherry'}
{'apple', 'banana', 'cherry'}
```

Out[4]:

```
{'banana', 'cherry'}
```

In [6]:

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

z = x.difference(y)

print(z)
```

```
{'banana', 'cherry'}
```

In [7]:

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}

x.difference_update(y)

print(x)
```

```
{'banana', 'cherry'}
```

In [8]:

```
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "facebook"}

z = x.isdisjoint(y)

print(z)
```

```
True
```

In [9]:

```
x = {"a", "b", "c"}
y = {"f", "e", "d", "c", "b", "a"}

z = x.issubset(y)

print(z)
```

```
True
```

In [10]:

```
x = {"f", "e", "d", "c", "b", "a"}
y = {"a", "b", "c"}

z = x.issuperset(y)

print(z)
```

```
True
```

In [11]:

```
x = {"a", "b", "c"}
y = {"f", "d", "a"}
z = {"c", "d", "e"}

result = x.union(y, z)

print(result)
```

```
{'d', 'a', 'b', 'f', 'e', 'c'}
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: