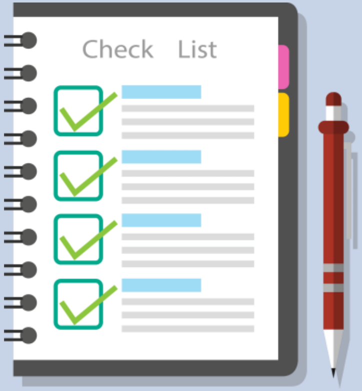
# Python – Basics

Digital Lync

INNOVATION - EDUCATION - INCUBATION

Shah Ayub Quadri
aquadri@digital-lync.com

# Index



Python Basics

- Sets
  - Introduction
  - Accessing Sets
  - Operations
  - Methods

# Sets

- A set is an unordered collection of items where every element is unique.
- A set is created by placing all the items inside curly braces {}, separated by comma or by using the built-in function set().
- Empty curly braces {} will make an empty dictionary in Python. To make a set without any elements we use the set() function without any argument.

**Examples of sets :**

- items = {10,20,30,40,50}
- names = {"arrow", "spear", "arrow", "arrow", "rock"}
- student = {"arrow", 99, "arrow", 50, "rock"}
- program =set(“pythonprogramming”)

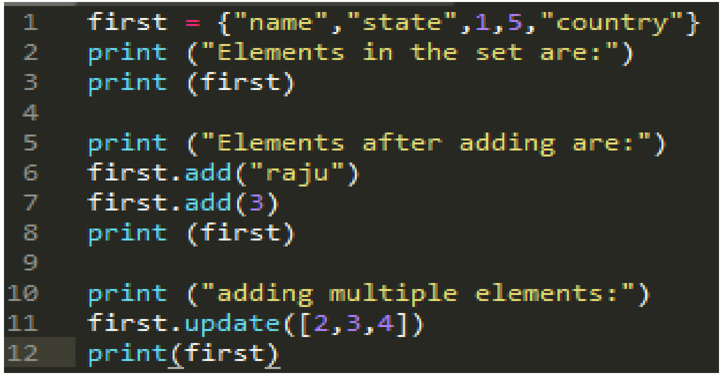
- colors = {"arrow", 99, "arrow", 50, "rock"}
- print (len(colors))  ## 5
- print (program)   ## {'r', 'h', 'a', 'i', 'p', 'g', 'o', 't', 'n', 'm', 'y'}

# Sets

## Updating Sets:

- We can add single element using the add() method and multiple elements using the update() method. The update() method can take tuples, lists, strings or other sets as its argument.

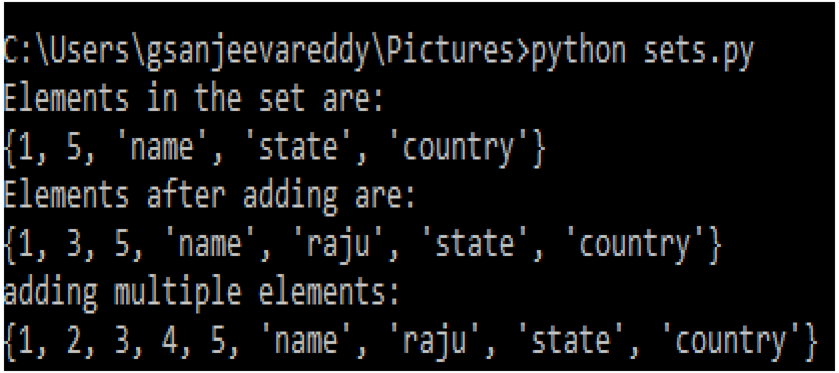
- **For example:**



```
first = {"name","state",1,5,"country"}
print ("Elements in the set are:")
print (first)

print ("Elements after adding are:")
first.add("raju")
first.add(3)
print (first)

print ("adding multiple elements:")
first.update([2,3,4])
print(first)
```



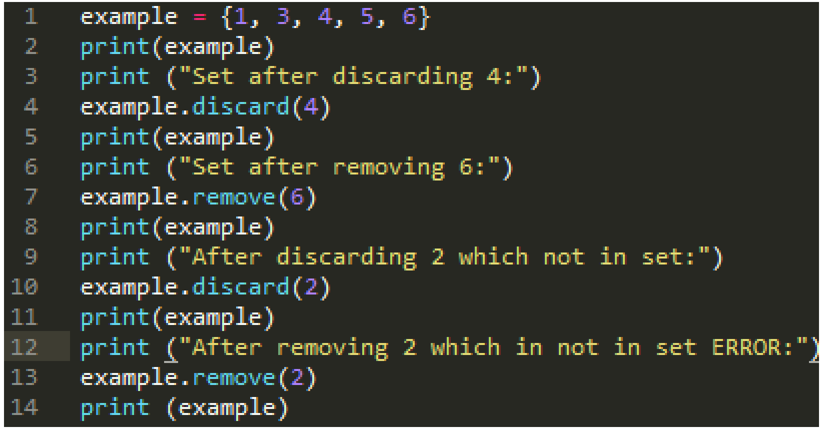
```
C:\Users\gsanjeevareddy\Pictures>python sets.py
Elements in the set are:
{1, 5, 'name', 'state', 'country'}
Elements after adding are:
{1, 3, 5, 'name', 'raju', 'state', 'country'}
adding multiple elements:
{1, 2, 3, 4, 5, 'name', 'raju', 'state', 'country'}
```

# Sets

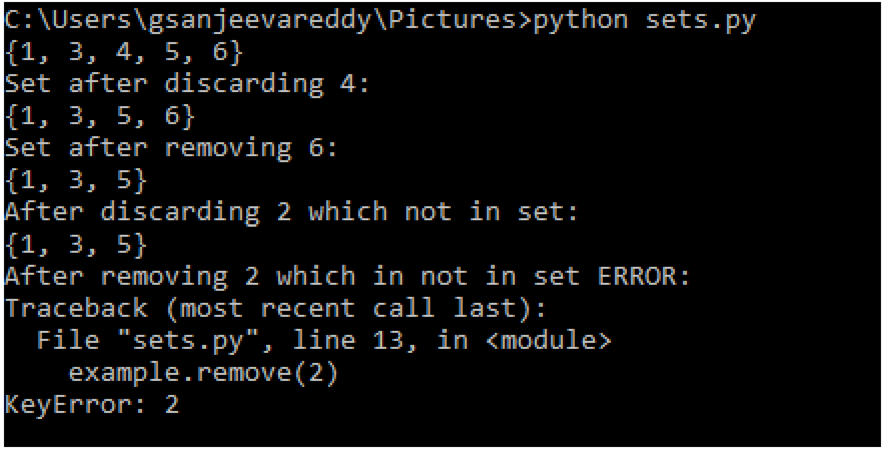
## Delete Set Elements:

A particular item can be removed from set using methods, discard() and remove().
The only difference between the two is that, while using discard() if the item does not exist in the set, it remains unchanged. But remove() will raise an error in such condition.

- **For example :**



```python
example = {1, 3, 4, 5, 6}
print(example)
print ("Set after discarding 4:")
example.discard(4)
print(example)
print ("Set after removing 6:")
example.remove(6)
print(example)
print ("After discarding 2 which not in set:")
example.discard(2)
print(example)
print ("After removing 2 which in not in set ERROR:")
example.remove(2)
print (example)
```



```
C:\Users\gsanjeevareddy\Pictures>python sets.py
{1, 3, 4, 5, 6}
Set after discarding 4:
{1, 3, 5, 6}
Set after removing 6:
{1, 3, 5}
After discarding 2 which not in set:
{1, 3, 5}
After removing 2 which in not in set ERROR:
Traceback (most recent call last):
  File "sets.py", line 13, in <module>
    example.remove(2)
KeyError: 2
```

# Sets

## Basic Set Operations:

| Python Expression | Results | Description |
|---|---|---|
| len({1,2,'arrow','hyderabad',7} | 5 | Length |
| {1,2,”arrow”} \| {"spear","rock"} | {1,2,”arrow”,”spear”,”rock”} | Union |
| {1,2,”arrow”}&{2,”arrow”,"rock”} | {2,”arrow”} | Intersection |
| {1, 2, 3, 4, 5} - {4, 5, 6, 7, 8} | {1,2,3} | Difference |
| {1, 2, 3, 4, 5}^{4, 5, 6, 7, 8} | {1,2,3,6,7,8} | Symmetric difference |

# Sets

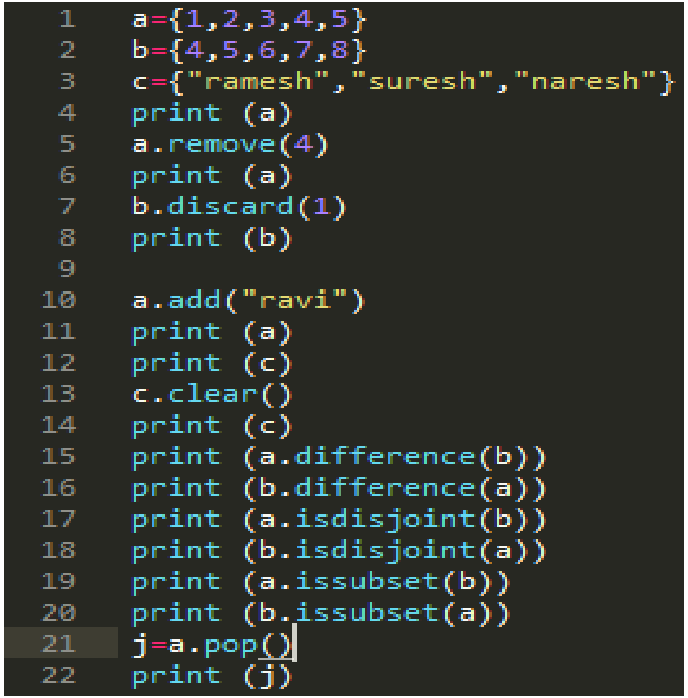
**Set Methods:**

Here are some other common set methods.

- **set.remove(element) :** remove an specified element.
- **set.add(elem)** :add one (or) multiple elements to the set by using list,tuple.
- **set.clear() :** It completely clear elements in set and display empty set.
- **set.difference(set1) :** remove elements that are common in set with set1 and display set elements.
- **seta.isdisjoint(setb) :** check both set has same elements (or) not and give boolean values.
- **a.issubset(b) :** check a is subset of b or not and give boolean values.
- **set.pop():** Remove and return an arbitary set element. Raise KeyError if the set is empty

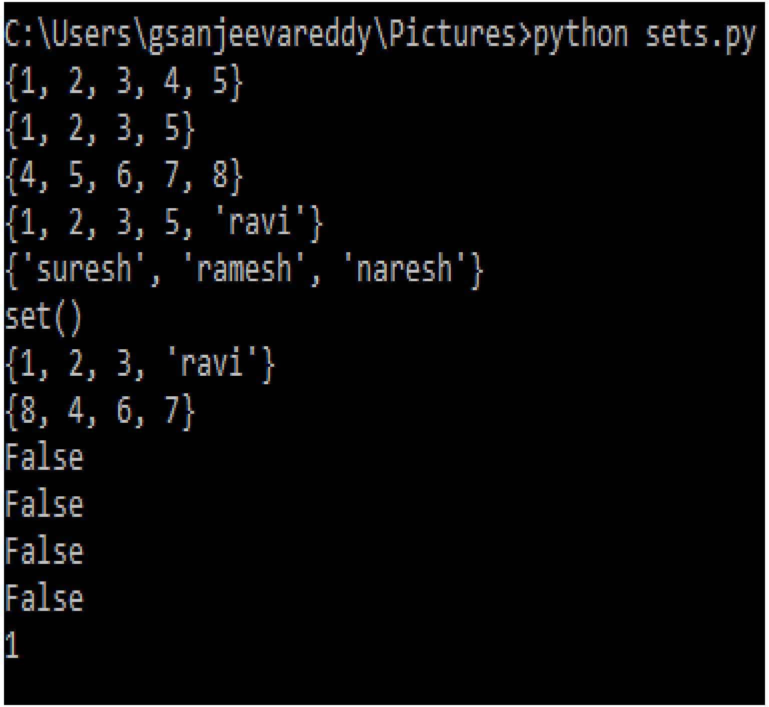
# Sets

Sets Methods:
EXAMPLES:



```python
a={1,2,3,4,5}
b={4,5,6,7,8}
c={"ramesh","suresh","naresh"}
print (a)
a.remove(4)
print (a)
b.discard(1)
print (b)

a.add("ravi")
print (a)
print (c)
c.clear()
print (c)
print (a.difference(b))
print (b.difference(a))
print (a.isdisjoint(b))
print (b.isdisjoint(a))
print (a.issubset(b))
print (b.issubset(a))
j=a.pop()
print (j)
```



```
C:\Users\gsanjeevareddy\Pictures>python sets.py
{1, 2, 3, 4, 5}
{1, 2, 3, 5}
{4, 5, 6, 7, 8}
{1, 2, 3, 5, 'ravi'}
{'suresh', 'ramesh', 'naresh'}
set()
{1, 2, 3, 'ravi'}
{8, 4, 6, 7}
False
False
False
False
1
```

# Assignment - 5

1. Write a Python program to create a set.

2. Write a Python program to create an intersection of sets.

3. Write a Python program to create a union of sets.

**4.** Write a Python program to create set difference.

5. Write a Python program to create a symmetric