Example:

```
>>> set1={10,20,30,40,50}
>>> print(type(set1))
<class 'set'>
>>> print(set1)
{50, 20, 40, 10, 30}
>>> set2={10,10,20,20,10,30,10,40,50}
>>> print(set2)
{50, 20, 40, 10, 30}
>>> set3={"naresh","suresh","naresh"}
>>> print(set3)
>>> set4={10,1.5,"naresh",1+2j}
>>> print(set4)
{1.5, 10, 'naresh', (1+2j)}
>>> set5=set()
>>> print(set5)
set()
>>> set6=set(range(10,60,10))
>>> print(set6)
{40, 10, 50, 20, 30}
>>> set7=set([10,10,40,50,60,20,60])
>>> print(set7)
{40, 10, 50, 20, 60}
>>> set8=set("NARESH")
>>> print(set8)
{'H', 'S', 'E', 'A', 'N', 'R'}
```

Reading elements from set

1. for loop
2. iterator
3. enumerate

Reading elements from set using for loop

Example:

```
set1={10,20,30,40,50}
for value in set1:
   print(value)
```

Output:

```
50
20
40
10
30
```

Reading elements from set using iterator

Example

```
set1={10,20,30,40,50}
a=iter(set1)
value1=next(a)
value2=next(a)
print(value1,value2)
```

Output:

```
50 20
```

Reading elements from set using enumerate

```
set1={10,20,30,40,50}
e=enumerate(set1)
t1=next(e)
t2=next(e)
print(t1)
print(t2)
```

Output:

```
(0, 50)
(1, 20)
```

Set is mutable collection, after creating set we can add, remove items/elements.

add(*elem*)

Add element *elem* to the set.

Example:

```
# create set of n elements

n=int(input("enter how many elements?"))
set1=set()
for i in range(n):
    value=int(input("enter any value"))
    set1.add(value)


print(set1)
```

Output:

```
enter how many elements?5
enter any value10
enter any value20
enter any value30
enter any value40
enter any value50
{40, 10, 50, 20, 30}
```

Example:

```
>>> set1={1,2,3,4,5}
>>> print(set1)
{1, 2, 3, 4, 5}
>>> set1.add(6)
>>> print(set1)
{1, 2, 3, 4, 5, 6}
```

https://www.hackerrank.com/challenges/py-set-add/problem?isFullScreen=false

```
n=int(input())
set1=set()
for i in range(n):
    country=input()
    set1.add(country)
```

```
print(len(set1))
```

remove(*elem*)
Remove element *elem* from the set. Raises KeyError if *elem* is not contained in the set.

```
>>> set1={10,20,30,40,50}
>>> print(set1)
{50, 20, 40, 10, 30}
>>> set1.remove(40)
>>> print(set1)
{50, 20, 10, 30}
>>> set1.remove(40)
Traceback (most recent call last):
  File "<pyshell#27>", line 1, in <module>
    set1.remove(40)
KeyError: 40
```

Example:

```
# create a set of n elements and remove input element from set

n=int(input("enter how many elements"))
set1=set(map(int,input().split()[:n]))
value=int(input("enter value to remove"))
if value in set1:
    set1.remove(value)
    print("value removed...")
else:
    print("value not found")

print(set1)
```

Output:

```
========= RESTART: F:/python6pmaug/test119.py ========
enter how many elements5
10 20 30 40 50 60 70
enter value to remove10
value removed...
```

```
{40, 50, 20, 30}
>>>
========= RESTART: F:/python6pmaug/test119.py ========
enter how many elements5
10 20 30 40 50
enter value to remove90
value not found
{40, 10, 50, 20, 30}
```

discard(*elem*)

Remove element *elem* from the set if it is present.

```
>>> set1={10,20,30,40,50}
>>> print(set1)
{50, 20, 40, 10, 30}
>>> set1.discard(50)
>>> print(set1)
{20, 40, 10, 30}
>>> set1.discard(50)
>>> print(set1)
{20, 40, 10, 30}
>>> set1.discard(90)
```

pop()

Remove and return an **arbitrary element** from the set. Raises KeyError if the set is empty.

```
>>> set1=set(range(10,110,10))
>>> print(set1)
{100, 70, 40, 10, 80, 50, 20, 90, 60, 30}
>>> value1=set1.pop()
>>> print(value1)
100
>>> print(set1)
{70, 40, 10, 80, 50, 20, 90, 60, 30}
>>> value2=set1.pop()
>>> print(value2)
70
```

clear()

Remove all elements from the set.

```
>>> set1=set(range(10,110,10))
>>> print(set1)
{100, 70, 40, 10, 80, 50, 20, 90, 60, 30}
>>> set1.clear()
>>> print(set1)
set()
```