In [1]:
```
myset =  {'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
myset
```

Out[1]:
```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

In [3]:
```
myset.add("NINE")
print(myset)
```

```
{'three', 'NINE', 'eight', 'seven', 'five', 'four', 'six', 'two', 'one'}
```

In [5]:
```
myset.add("TEN")
```

In [7]:
```
print(myset)
```

```
{'three', 'NINE', 'eight', 'seven', 'five', 'TEN', 'four', 'six', 'two', '
one'}
```

In [9]:
```
myset.update(["E","M","N"])
print(myset)
```

```
{'three', 'NINE', 'eight', 'N', 'seven', 'E', 'five', 'TEN', 'four', 'si
x', 'two', 'one', 'M'}
```

In [11]:
```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
A.union(B,C)
```

Out[11]:
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [13]:
```
mydict = {1:'one' , 'A':'two' , 3:'three'}
mydict
```

Out[13]:
```
{1: 'one', 'A': 'two', 3: 'three'}
```

In [15]:
```
mydict.keys()
```

Out[15]:
```
dict_keys([1, 'A', 3])
```

In [17]:
```
mydict.values()
```

Out[17]:
```
dict_values(['one', 'two', 'three'])
```

In [19]:
```
mydict.items()
```

Out[19]:
```
dict_items([(1, 'one'), ('A', 'two'), (3, 'three')])
```

In [21]:
```
keys = {'a' , 'b' , 'c' , 'd'}
value = [10,20,30]
mydict = dict.fromkeys(keys,value)
print(mydict)
```

```
{'a': [10, 20, 30], 'b': [10, 20, 30], 'd': [10, 20, 30], 'c': [10, 20, 3
0]}
```

In [23]:
```
value.append(40)
print(mydict)
```

```
{'a': [10, 20, 30, 40], 'b': [10, 20, 30, 40], 'd': [10, 20, 30, 40], 'c':
[10, 20, 30, 40]}
```

In [31]:
```
keys.("f")
value.append(50)
print(keys)
```

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

In [ ]:

In [33]:
```
print(mydict)
```

```
{'a': [10, 20, 30, 40, 50], 'b': [10, 20, 30, 40, 50], 'd': [10, 20, 30, 4
0, 50], 'c': [10, 20, 30, 40, 50]}
```

In [37]:
```
qTuple = (1, 'Jhon', 1+3j)
print(type(qTuple[2:3]))
```

```
<class 'tuple'>
```

In [39]:
```
type(range(5))?
```

```
  Cell In[39], line 1
    type(range(5))?
                  ^
SyntaxError: invalid syntax
```

print result with string

In [ ]:

In [4]:
```
num1 = 10
num2 = 20
add = num1 + num2
print("This addtion of",num1, "and", num2, "is", add)
```

```
This addtion of 10 and 20 is 30
```

print Format method

In [7]:
```
print("This addtion of {} and {} is = {}".format(num1,num2,add))
```

```
This addtion of 10 and 20 is = 30
```

In [11]:

```
num1=100
num2=25
num3=333
avg = (num1+num2+num3)/3
print(avg)
```

```
152.66666666666666
```

In [17]:

```
avg1 = round((num1+num2+num3)/3,2)
print(avg1)
```

```
152.67
```

In [25]:

```
print(f"The sum of {num1} and {num2} and {num3} is = {avg}")
```

```
The sum of 100 and 25 and 333 is = 152.66666666666666
```

end statement

In [ ]:

In [30]:

```
print("hello")
print("Good Morning")
```

```
hello
Good Morning
```

In [36]:

```
print("hello", end=' ')
print("Good Morning!")
```

```
hello Good Morning!
```

seprator

In [39]:

```
print("hello", "Good","Morning!", sep = '---->')
```

```
hello---->Good---->Morning!
```

In [45]:

```
print("hello","Good", "Morning", sep= ' @')
```

```
hello @Good @Morning
```

Binary in Python

In [ ]:

In [48]:

```
bin(19)
```

Out[48]:

```
'0b10011'
```

```
In [50]: bin(24)
```

```
Out[50]: '0b11000'
```

```
In [ ]: 
```

```
In [52]: bin(54)
```

```
Out[52]: '0b110110'
```

Operations with Binary Numbers

```
In [55]: 0b10011 + 0b10011
```

```
Out[55]: 38
```

```
In [57]: 0b10011 * 0b10011
```

```
Out[57]: 361
```

```
In [61]: 0b10011 + 19
```

```
Out[61]: 38
```

```
In [63]: 0b10011/0b10011
```

```
Out[63]: 1.0
```

convert a binary literal to an integer,

```
In [66]: int("0b10011",2)
```

```
Out[66]: 19
```

```
In [68]: int("0b10011",10)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[68], line 1
----> 1 int("0b10011",10)

ValueError: invalid literal for int() with base 10: '0b10011'
```

Octal Representation

```
In [71]: oct(52)

Out[71]: '0o64'

In [77]: oct(123)

Out[77]: '0o173'

In [79]: oct(100010)

Out[79]: '0o303252'

In [ ]:

In [ ]:
```