In [97]:

```
#variables declaration

a=10
b=2.5
c=23.50
d="Mindtree"
x,y,z=22,34,67
print("a :",a)
print("b :",b)
print("c :",c)
print("d :",d)
print("z :",z)
```

```
a : 10
b : 2.5
c : 23.5
d : Mindtree
z : 67
```

In [62]:

```
# data types in python

a=10
b=2.5
c=23.50
d="Mindtree"
e=12j+3
f=True
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
```

```
<class 'int'>
<class 'float'>
<class 'float'>
<class 'str'>
<class 'complex'>
<class 'bool'>
```

In [64]:

```
# Type casting in python

a=10
b=2.5
c=23.50
d="50"
print(int(c))
print(float(a))
print(int(d)+30)
print(hex(a))
print(oct(a))
```

```
23
10.0
80
0xa
0o12
```

In [12]:

```
# Strings in Python
```

```
a = 'Python'
c = a[2:4]
print(type(a))
print (c)
```

```
<class 'str'>
th
```

In [96]:

```
# String Operations

a="mindtree minds"

#capitalize first char
print(a.capitalize())

#count no of occurences
print(a.count('m'))

#return index of char
print(a.index('r'))

#to lower case
print(a.lower())

#to upper case
print(a.upper())

print(a.title())

#length of String
print(len(a))

#reverse of String
print(a[::-1])
```

```
Mindtree minds
2
5
mindtree minds
MINDTREE MINDS
Mindtree Minds
14
sdnim eertdnim
```

In [47]:

```
# Lists in python

a=[4,"Hi",50,34.5]

#type of a
print(type(a))
print(a)

#modifying values
a[1]="bye"

#inserting
a.insert(4,"No ")
print(a)

#length of list
print(len(a))

#slicing
print (a[2:4])

#printing index value
print(a[len(a)-1])
```

```
<class 'list'>
[4, 'Hi', 50, 34.5]
[4, 'bye', 50, 34.5, 'No ']
5
[50, 34.5]
No
```

In [49]:

```
# Tuples in Python

a=(1,"no",3,"Hi",5,7,8.6)
b=4,"mind",5.45
print(type(a))

#type
print(type(b))
print(a)

#length of tuple
print(len(a))
print(b)

#printing index
print(b[1])
```

```
<class 'tuple'>
<class 'tuple'>
(1, 'no', 3, 'Hi', 5, 7, 8.6)
7
(4, 'mind', 5.45)
mind
```

In [50]:

```
# Sets in Python

a={1,2,3,4,5,3,2,1}
b={3,4,6,9}
print(type(a))

#adding values to set
a.add(11)
print(a)
print(type(b))
print(b)

#length of set
print(len(a))

#union operation
print(a | b)

#intersection
print(a & b)
```

```
<class 'set'>
{1, 2, 3, 4, 5, 11}
<class 'set'>
{9, 3, 4, 6}
6
{1, 2, 3, 4, 5, 6, 9, 11}
{3, 4}
```

In [56]:

```
# Dictionaries in Python

a={'name':'praveen',
   'age' :20 ,
   'gender':'male'}
```

```
#type
print(type(a))
print(a)

#modifing dict
a['age']=21
print(a['age'])

#using get
print(a.get('name'))
```

```
<class 'dict'>
{'name': 'praveen', 'age': 20, 'gender': 'male'}
21
praveen
```

In [72]:

```
# user input

a=int(input("type int :"))
b=str(input("type string :"))
c=float(input("type float"))
d=input("type anything :")
print(d)
print(c)
print(b)
print(a)
```

```
type int :34
type string :Praveen
type float23.56
type anything :mind
mind
23.56
Praveen
34
```

In [77]:

```
# ARITHMETIC OPERATORS

print("ARITHMETIC OPERATIONS")
a=78
b=54

#addition
print(a+b)

#substraction
print(a-b)

#multi
print(a*b)

#div
print(a/b)
print(a//b)
print(a%b)
```

```
ARITHMETIC OPERATIONS
132
24
4212
1.4444444444444444
1
24
```

In [80]:

```
# ASSIGNMENT OPERATOR
```

```
a=10
print(a)
a+=2
print(a)
a-=5
print(a)
a*=2
print(a)
a%=2
print(a)
```

```
10
12
7
14
0
```

In [81]:

```
# RELATIONAL OPERATOR

a = 5
b = 9
c = 8
print (a>b)
print (b>c)
print ( (a>b) or (b>c))
```

```
False
True
True
```

In [89]:

```
# CONDITIONAL STATEMENTS IN PYTHON

a=int(input("enter your age :"))

if a<18:
    print("your not eligible")
elif(a>=18 and a < 50):
    print("your eligible")
elif a > 50 :
    print("your not eligible")
else:
    print("your not eligible")
```

```
enter your age :12
your not eligible
```

In [91]:

```
# Loops in Python

a=[2,3,4,6,7,8]
b=0
for i in range(len(a)):
    b+=a[i]
    print(a[i])
print("Sum",b)
```

```
2
3
4
6
7
8
Sum 30
```

In [95]:

```
# Nested loops
```

```
for i in range(1,10,1):
    for j in range(1,i):
        print(j,end=" ")
    print('\n')
```

```
1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1 2 3 4 5 6

1 2 3 4 5 6 7

1 2 3 4 5 6 7 8
```

In [ ]:

```

```