In [1]:
```
# Relational Operator
a=20
b=30
print(a>b)
print(a<b)
print(a==b)
print(a!=b)
print(a>=b)
print(a<=b)
```

```
False
True
False
True
False
True
```

In [2]:
```
#Logical Operator
a=True
b=False
print(a and b)
print(a or b)
print( not a)
print( not b)
```

```
False
True
False
True
```

In [3]:
```
#String, list, tuple , set, dictionary
# sequence data type which can hold data more then one value
str1="Hello"
print(str1)
print(type(str1))
```

```
Hello
<class 'str'>
```

In [4]:
```
# forward indexing   0,1,2,3,4
print(str1[3])
#backward indexing   -5,-4,-3,-2,-1
print(str1[-1])
```

```
l
o
```

In [5]:
```
# slicing - extract a sub string from the string
# range- operator : colon
str2="Hello python language"
print(str2[2:5]) # range in exclude the higher index -stop-1
```

```
llo
```

In [6]:
```
len(str2)
print(str2[13:len(str2)])
print(str2[13:21])
```

```
language
language
```

In [7]:
```
print(str2[:5])
print(str2[13:])
```

```
Hello
language
```

In [8]:
```
# reverse the string
print(str2[::])
print(str2[0:21:1])
print(str2[::-1])
```

```
Hello python language
Hello python language
egaugnal nohtyp olleH
```

In [9]:
```
# concatination= adding two strings
str1 = "hello"
str2 = "world"
str3 = str1 + str2
print(str3)
str3 = str1 +' '+ str2
print(str3)
```

```
helloworld
hello world
```

In [10]:
```
print(str1*5)
print((str1+" ")*3)
```

```
hellohellohellohellohello
hello hello hello
```

In [11]:
```
# functions of string- inbuild function
str1="Hello Python"
print(str1.upper())
print(str1.lower())
print(str1.title())# convert every first char into upper case
print(str1.capitalize()) # first char into upper case
```

```
HELLO PYTHON
hello python
Hello Python
Hello python
```

In [12]:
```
print(str1.endswith("n"))
```

```
True
```

In [13]:
```
# split method into list
print(str1.split())
print(str1.count('H'))
```

```
['Hello', 'Python']
1
```

In [14]:
```
str1="Good morning India@"
print(str1.count('o'))
print(str1.count('g'))
```

```
3
1
```

In [15]:
```
# replace method
print(str1.replace('@',""))
```

```
Good morning India
```

In [16]:
```
# LIST
# list created in []
list1=[1,2,3,4.5,9,8,8,"orange","banana"]
print(list1)
print(type(list1))
```

```
[1, 2, 3, 4.5, 9, 8, 8, 'orange', 'banana']
<class 'list'>
```

In [17]:
```
# mutable
list1[0]=20
list1
```

Out[17]:
```
[20, 2, 3, 4.5, 9, 8, 8, 'orange', 'banana']
```

In [18]:
```
# access third elements
list1[2]
```

Out[18]:
```
3
```

In [19]:
```
list1[-1]
```

Out[19]:
```
'banana'
```

In [20]:
```
#slicing
#access first three elements
print(list1[:3])
```

```
[20, 2, 3]
```

In [21]:
```
# len of list
len(list1)
```

Out[21]:
```
9
```

In [22]:
```
# access last two elements
print(list1[-2:])
```

```
['orange', 'banana']
```

In [23]:
```
# reverse order
print(list1[::-1])
```

```
['banana', 'orange', 8, 8, 9, 4.5, 3, 2, 20]
```

In [24]:
```
# create [], contrcutor method list()
list2=list((5,6,7,8))
list2
```

Out[24]:
```
[5, 6, 7, 8]
```

In [25]:
```
new=[1,4,6,8,6.5,7]
```

In [26]:
```
# remove the data
new.pop(1)
new
```

Out[26]:
```
[1, 6, 8, 6.5, 7]
```

In [27]:
```
# remove the data
new.pop() #it always remove last item
new
```

Out[27]:
```
[1, 6, 8, 6.5]
```

In [28]:
```
#remove
new=[1,4,6,8,6.5,7]
new.remove (6.5) #it remove value
new
```

Out[28]:
```
[1, 4, 6, 8, 7]
```

In [29]:
```
# empty the list
new.clear()
new
```

Out[29]:
```
[]
```

In [30]:
```
new=[3,4,9,87,5,6,7]
new
```

Out[30]:
```
[3, 4, 9, 87, 5, 6, 7]
```

In [31]:
```
new.sort() # sorting
new
```

Out[31]:
```
[3, 4, 5, 6, 7, 9, 87]
```

In [32]:
```
new.sort(reverse=True) # decending
new
```

Out[32]:
```
[87, 9, 7, 6, 5, 4, 3]
```

In [33]:
```
#join two list
list1=[5,6,7]
list2=[9,8,1]
list3=list1+list2
list3
```

Out[33]:
```
[5, 6, 7, 9, 8, 1]
```

In [34]:
```
#join two list
list1=[5,6,7]
list2=[9,8,1]
list1.extend(list2)
list1
```

Out[34]:
```
[5, 6, 7, 9, 8, 1]
```

In [35]:
```
# list membership operator # it checks value is present or not
list1=[5,6,7]
print(5 in list1)
```

```
True
```

In [36]:
```
# list membership operator
list1=[5,6,7]
print(10 not in list1)
```

```
True
```

tuple-

1. One dimentional
2. Ordered, indexed
3. Immutable cant change
4. more fast
5. duplicates are allowed

In [38]:
```
# create a tuple ()
tuple1=(4,5,6,2.3,"apple","blue")
```

In [39]:
```
print(tuple1)
print(type(tuple1))
```

```
(4, 5, 6, 2.3, 'apple', 'blue')
<class 'tuple'>
```

In [40]:
```
# tuple using constructor
new1=tuple((6,7,8,4))
new1
```

Out[40]:
```
(6, 7, 8, 4)
```

In [41]:
```
# convert tuple into list
list1=list(new1)
list1
```

Out[41]:
```
[6, 7, 8, 4]
```

In [42]:
```
list1[1]=10
list1
```

Out[42]:
```
[6, 10, 8, 4]
```

In [43]:
```
new2=tuple(list1)
new2
```

Out[43]:
```
(6, 10, 8, 4)
```

In [44]:
```
x=new2.count(4)
x
```

Out[44]:
```
1
```

In [45]:
```
new2.count(4)
```

Out[45]:
```
1
```

In [46]:
```
new2.index(6)
```

Out[46]:
```
0
```

In [47]:
```
jaya="Beaytifull"
print(jaya)
```

```
Beaytifull
```

In [48]:
```
jaya[4]
```

Out[48]:
```
't'
```

In [ ]:
```

```