Logical operators

AND

OR

NOT

In [1]:

```
True and True
```

Out[1]:

```
True
```

In [2]:

```
True and False
```

Out[2]:

```
False
```

In [3]:

```
True or False
```

Out[3]:

```
True
```

In [4]:

```
not True
```

Out[4]:

```
False
```

In [5]:

```
not bool(0)
```

Out[5]:

```
True
```

In [6]:

```
int(bool(0))
```

Out[6]:

```
0
```

In [7]:
```
not(0)
```
Out[7]:
```
True
```
In [8]:
```
not(-1)
```
Out[8]:
```
False
```
In [9]:
```
bool(192)
```
Out[9]:
```
True
```
In [10]:
```
True + True
```
Out[10]:
```
2
```

Presedance rule - BODMAS RULE

In [11]:
```
not True * False
```
Out[11]:
```
True
```
In [12]:
```
(not True)* False
```
Out[12]:
```
0
```

COMPARISION OPEATORS

IS - if a==b , then true, i.e a and b are same objects.

IS NOT - if a not equal to b, then true, i.e a nd b are different objects.

== - if a=b, then true

!= - if a not equal to b , then true

In [13]:

```
lst1=[1,2,3,4]
lst2=[1,2,3,4]
print(id(lst1),id(lst2))
```

```
2253081153280 2253081153024
```

In [14]:

```
# In the above block of code we got different id's because List is mutable so the two lists
```

In [15]:

```
lst1[0]
```

Out[15]:

```
1
```

In [16]:

```
lst1[0]=11
```

In [17]:

```
lst1
```

Out[17]:

```
[11, 2, 3, 4]
```

In [18]:

```
lst1[0]=1
lst1
```

Out[18]:

```
[1, 2, 3, 4]
```

In [19]:

```
lst1 is lst2
```

Out[19]:

```
False
```

In [20]:

```
lst1 is not lst2
```

Out[20]:

```
True
```

In [21]:

```
lst1==lst2
```

Out[21]:

```
True
```

In [22]:

```
lst1!=lst2
```

Out[22]:

```
False
```

Checking Mutability of int and string

In [23]:

```
a=11
b=11
id(a),id(b)
```

Out[23]:

```
(2253000895088, 2253000895088)
```

In [24]:

```
a="Harshith"
b="Harshith"
id(a),id(b)
```

Out[24]:

```
(2253081296240, 2253081296240)
```

In [25]:

```
a[0]
```

Out[25]:

```
'H'
```

In [26]:

```
a[0]="K"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20220\2102382105.py in <module>
----> 1 a[0]="K"

TypeError: 'str' object does not support item assignment
```

Arthmetic Opertions

+ - Addition

- - Substraction

* - Multiply

/ - Division

// - Division without decimal values

% - Remainder

In [27]:

```
a=10
b=12
```

In [28]:

```
a+b
```

Out[28]:

```
22
```

In [29]:

```
a-b
```

Out[29]:

```
-2
```

In [30]:

```
a*b
```

Out[30]:

```
120
```

In [31]:

```
a/b
```

Out[31]:

```
0.8333333333333334
```

In [32]:

```
a//b
```

Out[32]:

```
0
```

In [33]:

```
a%b
```

Out[33]:

```
10
```

In Python, the following bitwise operators are supported:

& (and) | (or) ^ (xor) ~ (not) << (left shift) (right shift)

In [34]:

```
var=10
```

In [35]:

```
var >>1
```

Out[35]:

```
5
```

In [36]:

```
var<<3
```

Out[36]:

```
80
```

In [37]:

```
True & False
```

Out[37]:

```
False
```

In [38]:

```
True | False
```

Out[38]:

```
True
```

In [39]:

```
~var
```

Out[39]:

```
-11
```

In [40]:

```
True ^ False
```

Out[40]:

```
True
```

STRINGS

In [41]:

```

"HARSHITH"
```

Out[41]:

```
'HARSHITH'
```

In [42]:

```
str1="Data Science"
type(str1)
```

Out[42]:

```
str
```

In [43]:

```
# Slicing
str1[4]
```

Out[43]:

```
' '
```

In [44]:

```
str1[4:]
```

Out[44]:

```
' Science'
```

In [45]:
```
str1[2:9]
```
Out[45]:
```
'ta Scie'
```
In [46]:
```
str1[-3:]
```
Out[46]:
```
'nce'
```
In [47]:
```
str1[:-1]
```
Out[47]:
```
'Data Scienc'
```
In [48]:
```
str1[::-1]
```
Out[48]:
```
'ecneicS ataD'
```
In [49]:
```
str1
```
Out[49]:
```
'Data Science'
```
In [50]:
```
str1[3::-1]
```
Out[50]:
```
'ataD'
```
In [51]:
```
name="Data Science Masters"
```
In [52]:
```
name[5:12]
```
Out[52]:
```
'Science'
```

In [53]:

```
name[12:4:-1]
```

Out[53]:

```
' ecneicS'
```

In [54]:

```
# Concatenation
"I am Learning "+name
```

Out[54]:

```
'I am Learning Data Science Masters'
```

In [55]:

```
name*5
```

Out[55]:

```
'Data Science MastersData Science MastersData Science MastersData Science Mast
ersData Science Masters'
```

In [56]:

```
len(name)
```

Out[56]:

```
20
```

In [57]:

```
name.find("en")
```

Out[57]:

```
8
```

In [58]:

```
name.find("a")
```

Out[58]:

```
1
```

In [59]:

```
# Find after 2nd index
name.find("a",2)
```

Out[59]:

```
3
```

In [60]:

```
# Find between 2nd and 9th index
name.find("c",2,10)
```

Out[60]:

```
6
```

In [61]:

```
name.count("a")
```

Out[61]:

```
3
```

In [62]:

```
name.count("a",2)
```

Out[62]:

```
2
```

In [63]:

```
name.count("ci")
```

Out[63]:

```
1
```

In [64]:

```
name.count("")
```

Out[64]:

```
21
```

In [65]:

```
name.split()
```

Out[65]:

```
['Data', 'Science', 'Masters']
```

In [66]:

```
name.split("a")
```

Out[66]:

```
['D', 't', ' Science M', 'sters']
```

In [67]:

```
name.split(" ")
```

Out[67]:

```
['Data', 'Science', 'Masters']
```

In [68]:

```
name.split("a")
```

Out[68]:

```
['D', 't', ' Science M', 'sters']
```

In [69]:

```
name.partition("a")
```

Out[69]:

```
('D', 'a', 'ta Science Masters')
```

In [70]:

```
# String Upper and Lower case
name.upper()
```

Out[70]:

```
'DATA SCIENCE MASTERS'
```

In [71]:

```
name.lower()
```

Out[71]:

```
'data science masters'
```

In [72]:

```
name
```

Out[72]:

```
'Data Science Masters'
```

In [73]:

```
name.swapcase()
```

Out[73]:

```
'dATA sCIENCE mASTERS'
```

In [74]:

```
name.title()
```

Out[74]:

```
'Data Science Masters'
```

In [75]:

```
name="wnjdjne ncbjwc bwj wcjncjb cw jbwcb  cdnwcnj c jc c wc c wdkncbj"
name.title()
```

Out[75]:

```
'Wnjdjne Ncbjwc Bwj Wcjncjb Cw Jbwcb  Cdnwcnj C Jc C Wc C Wdkncbj'
```