

```
In [77]: 1+1 # ADDITION
```

```
Out[77]: 2
```

```
In [78]: 2-1
```

```
Out[78]: 1
```

```
In [79]: 3*4
```

```
Out[79]: 12
```

```
In [80]: 8/4 # Division
```

```
Out[80]: 2.0
```

```
In [81]: 8/5 # Float division
```

```
Out[81]: 1.6
```

```
In [82]: 8/4
```

```
Out[82]: 2.0
```

```
In [83]: 8 // 4 # integer division
```

```
Out[83]: 2
```

```
In [84]: 8 + 9 - 7
```

```
Out[84]: 10
```

```
In [85]: 8 + 8 -
```

```
Cell In[85], line 1
      8 + 8 -
           ^
SyntaxError: invalid syntax
```

```
In [86]: 5 + 5 * 5 # BODMAS (Bracket || Orders || Divide || Multiply || Add || Substact)
```

```
Out[86]: 30
```

```
In [87]: 2 * 2 * 2 * 2 * 2 * 2 * 2 # exponentation
```

```
Out[87]: 64
```

```
In [88]: 2 * 5
```

```
Out[88]: 10
```

```
In [89]: 2 ** 5
```

```
Out[89]: 32
```

```
In [90]: 15 / 3
```

```
Out[90]: 5.0
```

```
In [91]: 10 // 3
```

```
Out[91]: 3
```

```
In [92]: 15 % 2 # Modulus
```

```
Out[92]: 1
```

```
In [93]: 10 % 2
```

```
Out[93]: 0
```

```
In [94]: 15 %% 2
```

```
Cell In[94], line 1
      15 %% 2
      ^
SyntaxError: invalid syntax
```

```
In [95]: 3 + 'nit'
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[95], line 1
----> 1 3 + 'nit'

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
In [96]: a,b,c,d,e = 15,7.8,'nit',8+9j,True
          print(a)
          print(b)
          print(c)
          print(d)
          print(e)
```

```
15
7.8
nit
(8+9j)
True
```

```
In [97]: print(type(a))
          print(type(b))
          print(type(c))
          print(type(d))
          print(type(e))
```

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

```
In [98]: type(c)
```

Out[98]: str

string

In [99]: 'Naresh IT'

Out[99]: 'Naresh IT'

In [100]: print('Max it')

Max it

In [101]: "max it technology"

Out[101]: 'max it technology'

In [102]: s1 = 'max it technology'
s1

Out[102]: 'max it technology'

In [103]: a = 2
b = 3
a + b

Out[103]: 5

In [104]: c = a + b
c

Out[104]: 5

In [105]: a = 3
b = 'hi'
type(b)

Out[105]: str

In [106]: print('max it's"Technology"') # \ has some special meaning to ignore the error

```
Cell In[106], line 1
    print('max it's"Technology"') # \ has some special meaning to ignore the error
or
    ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [107]: print('max it\'s"Technology"') #\ has some special meaning to ignore the error
max it's"Technology"

In [108]: print('max it', 'Technology')
max it Technology

In [109]: 'nit' + ' nit'

Out[109... 'nit nit'

In [110... 'nit' ' nit'

Out[110... 'nit nit'

In [111... 5 * 'nit'

Out[111... 'nitnitnitnitnit'

In [112... 5*' nit' # space between words

Out[112... ' nit nit nit nit nit'

In [113... print('c:\nit') #\n -- new line

c:

it

In [114... print(r'c:\nit') #raw string

c:\nit

variable

In [115... 2

Out[115... 2

In [116... x = 2
x

Out[116... 2

In [117... x + 3

Out[117... 5

In [118... y = 3
y

Out[118... 3

In [119... x + y

Out[119... 5

In [120... x = 9
x

Out[120... 9

In [121... x + y

Out[121... 12

In [122... `x + 10`

Out[122... 19

In [123... `_ + y`

Out[123... 22

In [124... `_ + y`

Out[124... 25

In [125... `_ + y`

Out[125... 28

In [126... `y`

Out[126... 3

In [127... `# string variable`
`name = 'mit'`
`name`

Out[127... 'mit'

In [128... `name + 'technology'`

Out[128... 'mittechnology'

In [129... `name 'technology'`

Cell In[129], line 1
 name 'technology'
 ^
SyntaxError: invalid syntax

In [130... `name`

Out[130... 'mit'

In [131... `len(name)`

Out[131... 3

In [132... `name[0]`

Out[132... 'm'

In [133... `name[5]`

```
-----  
IndexError                                Traceback (most recent call last)  
Cell In[133], line 1  
----> 1 name[5]  
  
IndexError: string index out of range
```

In [134... name[7]

```
-----  
IndexError                                Traceback (most recent call last)  
Cell In[134], line 1  
----> 1 name[7]  
  
IndexError: string index out of range
```

In [135... name[-1]

Out[135... 't'

In [136... name[-2]

Out[136... 'i'

In [137... name[-6]

```
-----  
IndexError                                Traceback (most recent call last)  
Cell In[137], line 1  
----> 1 name[-6]  
  
IndexError: string index out of range
```

slicing

In [138... name

Out[138... 'mit'

In [139... name[0:1]

Out[139... 'm'

In [140... name[1:4]

Out[140... 'it'

In [141... name[1:]

Out[141... 'it'

In [142... name[:4]

Out[142... 'mit'

```
In [143... name[3:9]
```

```
Out[143... ''
```

```
In [144... name
```

```
Out[144... 'mit'
```

```
In [145... name1 = 'fine'
name1
```

```
Out[145... 'fine'
```

```
In [146... name1[0:1]
```

```
Out[146... 'f'
```

```
In [147... name1[0:1] = 'd'
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[147], line 1
----> 1 name1[0:1] = 'd'

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

```
In [148... name1[0] = 'd'
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[148], line 1
----> 1 name1[0] = 'd'

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

```
In [149... name1
```

```
Out[149... 'fine'
```

```
In [150... name1[1:]
```

```
Out[150... 'ine'
```

```
In [151... 'd' + name1[1:] #i want to change fine to dine
```

```
Out[151... 'dine'
```

```
In [152... num1.insert(2,'nit')
```

```
Cell In[152], line 1
    num1.insert(2,'nit')
               ^
SyntaxError: unterminated string literal (detected at line 1)
```

Introduce to ID()

```
In [154... num = 5 # variable address  
id(num)
```

```
Out[154... 140729894906424
```

```
In [155... name = 'nit'  
id(name) # Address will be different for both
```

```
Out[155... 1841196533344
```

```
In [156... a = 10  
id(a)
```

```
Out[156... 140729894906584
```

```
In [157... b = a # thats why python is more efficient
```

```
In [158... id(b)
```

```
Out[158... 140729894906584
```

```
In [159... id(10)
```

```
Out[159... 140729894906584
```

```
In [160... a = 20  
id(a) # as we change the value of then address will also change
```

```
Out[160... 140729894906904
```

```
In [161... id(b)
```

```
Out[161... 140729894906584
```

```
In [162... PI = 3.14 # In maths pi value is constant but in python we change pi value  
PI
```

```
Out[162... 3.14
```

```
In [163... PI = 3.15  
PI
```

```
Out[163... 3.15
```

```
In [164... type(PI)
```

```
Out[164... float
```

Arithmetic Operator

```
In [165... x1, y1 = 10,5
```

```
In [167... x1 ^ y1
```


Out[167... 15

In [168... `x1 + y1`

Out[168... 15

In [169... `x1 - y1`

Out[169... 5

In [170... `x1 * y1`

Out[170... 50

In [171... `x1 / y1`

Out[171... 2.0

In [172... `x1 // y1`

Out[172... 2

In [173... `x1 % y1`

Out[173... 0

In [174... `x1 ** y1`

Out[174... 100000

Assignment operator

In [175... `x = 2`

In [176... `x = x+2`

In [177... `x`

Out[177... 4

In [179... `x += 2`
`x`

```
Cell In[179], line 1
  x += 2
    ^
SyntaxError: invalid syntax
```

In [180... `x *= 2`
`x`

```
Cell In[180], line 1
  x *= 2
    ^
SyntaxError: invalid syntax
```

```
In [181... x - = 2
x
```

```
Cell In[181], line 1
    x - = 2
      ^
SyntaxError: invalid syntax
```

```
In [182... x / = 2
x
```

```
Cell In[182], line 1
    x / = 2
      ^
SyntaxError: invalid syntax
```

```
In [183... a,b = 5,6
```

```
In [184... a
```

```
Out[184... 5
```

```
In [185... b
```

```
Out[185... 6
```

unary operator

- unary means 1 || binary means 2
- Here we are applying unary minus operator(-) on the operand n; the value of m becomes -7, which indicates it as a negative value.

```
In [186... n = 7
n
```

```
Out[186... 7
```

```
In [187... m = -(n)
m
```

```
Out[187... -7
```

```
In [188... n
```

```
Out[188... 7
```

```
In [189... -n
```

```
Out[189... -7
```

Relational operator

we are using this operator for comparing

```
In [190... a = 5
          b = 6

In [191... a < b

Out[191... True

In [192... a > b

Out[192... False

In [193... # a = b # we cannot use = operatro that means it is assigning

In [194... a == b

Out[194... False

In [196... a != b

Out[196... True

In [197... b = 5

In [198... b

Out[198... 5

In [199... a == b

Out[199... True

In [200... a >= b

Out[200... True

In [201... a <=b

Out[201... True

In [202... a < b

Out[202... False

In [203... a > b

Out[203... False

In [204... b = 7
```

In [205... `a != b`

Out[205... `True`

Logical operator

- AND,OR,NOT

In [206... `a = 5`
`b = 4`

In [207... `a < 8 and b < 5`

Out[207... `True`

In [208... `a < 8 and b < 2`

Out[208... `False`

In []: `a > 8 and o`