21 Oct 2025

Python Basic& python Operator

In [1]:
```
1 + 1
```

Out[1]: 2

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

Out[2]: 1

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

Out[3]: 12

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

Out[4]: 2.0

In [5]:
```
8 / 5
```

Out[5]: 1.6

In [6]:
```
8/4
```

Out[6]: 2.0

In [7]:
```
8 // 4
```

Out[7]: 2

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

Out[8]: 10

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

```
  Cell In[9], line 1
    8 + 8 -
          ^
SyntaxError: invalid syntax
```

In [10]:
```
5 + 5 * 5
```

Out[10]: 30

In [11]:
```
(5 + 5) * 5
```

Out[11]: 50

In [12]:
```
2 * 2 * 2 * 2 * 2
```

Out[12]: 32

In [13]:
```
2 * 5
```

Out[13]: 10

In [14]:
```
2 ** 5
```

Out[14]: 32

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

Out[15]: 5.0

In [16]:
```
10 // 3
```

Out[16]: 3

In [17]:
```
15 % 2
```

Out[17]: 1

In [18]:
```
10 % 2
```

Out[18]: 0

In [19]:
```
15 %% 2
```

```
  Cell In[19], line 1
    15 %% 2
       ^
SyntaxError: invalid syntax
```

In [20]:
```
3 + 'nit'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 1
----> 1 3 + 'nit'

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

In [21]:
```
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 [22]:
```
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 [23]:
```
type(c)
```

Out[23]: str

In [24]:
```
'Naresh IT'
```

Out[24]: 'Naresh IT'

In [25]:
```
print('Max it')
```

```
Max it
```

In [26]:
```
"max it technology"
```

Out[26]: 'max it technology'

In [27]:
```
s1 = 'max it technology'
s1
```

Out[27]: 'max it technology'

In [28]:
```
a = 2
b = 3
a + b
```

Out[28]: 5

In [29]:
```
c = a + b
c
```

Out[29]: 5

In [30]:
```
a = 3
b = 'hi'
type(b)
```

Out[30]: str

In [31]:
```
print('max it's"Technology"') # \ has some special meaning to ignore the error
```

```
  Cell In[31], line 1
    print('max it's"Technology"') # \ has some special meaning to ignore the error
                                  ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [32]:
```
print('max it\'s"Technology"') #\ has some special meaning to ignore the error
```

```
max it's"Technology"
```

In [33]:
```
print('max it',  'Technology')
```

```
max it Technology
```

In [34]:
```
print("max it', 'Technology")
```

```
max it', 'Technology
```

In [35]:
```
'nit' + 'nit'
```

Out[35]: 'nitnit'

In [36]:
```
'nit ' nit'
```

```
  Cell In[36], line 1
    'nit ' nit'
           ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [37]:
```
5 * 'nit'
```

Out[37]: 'nitnitnitnitnit'

In [38]:
```
5*' nit'
```

Out[38]: ' nit nit nit nit nit'

In [39]:
```
print('c:\nit')
```

```
c:
it
```

In [40]:
```
print(r'c:\nit')
```

```
c:\nit
```

variable || identifier || object

In [41]:
```
2
```

Out[41]: 2

In [42]:
```
x = 2
x
```

Out[42]: 2

In [43]:
```
x + 3
```

Out[43]: 5

In [44]:
```
y = 3
y
```

Out[44]: 3

In [45]:
```
x + y
```

Out[45]: 5

In [46]:
```
x = 9
x
```

Out[46]: 9

In [47]:
```
x + y
```

Out[47]: 12

In [48]:
```
x + 10
```

Out[48]: 19

In [49]:
```
_ + y
```

Out[49]: 22

In [50]:
```
_ + y
```

Out[50]: 25

In [51]:
```
_ + y
```

Out[51]: 28

In [52]:
```
_ + y
```

Out[52]: 31

In [53]:
```
y
```

Out[53]: 3

In [54]:
```
_ + y
```

Out[54]: 6

In [55]:
```
_ + y
```

Out[55]: 9

In [56]:
```
_ + y
```

Out[56]: 12

In [57]:
```
name = 'mit'
```

In [58]:
```
name
```

Out[58]: 'mit'

In [59]:
```
name + 'technology'
```

Out[59]: 'mittechnology'

In [60]:
```
name + ' technology'
```

Out[60]: 'mit technology'

In [61]:
```
name 'technology'
```

```
  Cell In[61], line 1
    name 'technology'
         ^
SyntaxError: invalid syntax
```

In [62]:
```
name
```

Out[62]: 'mit'

In [63]:
```
len(name)
```

Out[63]: 3

In [64]:
```
name[0]
```

Out[64]: 'm'

In [65]:
```
name[5]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[65], line 1
----> 1 name[5]

IndexError: string index out of range
```

In [66]:
```
name[7]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[66], line 1
----> 1 name[7]

IndexError: string index out of range
```

In [67]:
```
name[-1]
```

Out[67]: 't'

In [68]:
```
name[-2]
```

Out[68]: 'i'

In [69]:
```
name[-6]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[69], line 1
----> 1 name[-6]

IndexError: string index out of range
```

Slicing

In [70]:
```
name
```

Out[70]: 'mit'

In [71]:
```
name[0:1]
```

Out[71]: 'm'

In [72]:
```
name[0:2]
```

Out[72]: 'mi'

In [73]:
```
name[1:4]
```

Out[73]: 'it'

In [74]:
```
name[1:]
```

Out[74]: 'it'

In [75]:
```
name[:4]
```

Out[75]: 'mit'

In [76]:
```
name[3:9]
```

Out[76]: ''

In [77]:
```
name
```

Out[77]: 'mit'

In [78]:
```
name1 = 'fine'
name1
```

Out[78]: 'fine'

In [79]:
```
name1[0:1]
```

Out[79]: 'f'

In [80]:
```
name1[0:1] = 'd'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[80], line 1
----> 1 name1[0:1] = 'd'

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

In [81]:
```
name1[0] = 'd'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[81], line 1
----> 1 name1[0] = 'd'

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

In [82]:
```
name1
```

Out[82]: 'fine'

In [83]:
```
name1[1:]
```

Out[83]: 'ine'

In [84]:
```
'd' + name1[1:]
```

Out[84]: 'dine'

In [85]:
```
num1.insert(2,'nit')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[85], line 1
----> 1 num1.insert(2,'nit')

NameError: name 'num1' is not defined
```

introduce to ID()

In [86]:
```
num = 5
id(num)
```

Out[86]: 140706000385064

In [87]:
```
name = 'nit'
id(name)
```

Out[87]: 2735463164992

In [88]:
```
a = 10
id(a)
```

Out[88]: 140706000385224

In [89]:
```
b = a
```

In [90]:
```
id(b)
```

Out[90]: 140706000385224

In [91]:
```
id(10)
```

Out[91]: 140706000385224

In [92]:
```
k = 10
id(k)
```

Out[92]: 140706000385224

In [93]:
```
a = 20
id(a)
```

Out[93]: 140706000385544

In [94]:
```
id(b)
```

Out[94]: 140706000385224

In [95]:
```
PI = 3.14
PI
```

Out[95]: 3.14

In [96]:
```
PI = 3.15
PI
```

Out[96]: 3.15

In [97]:
```
type(PI)
```

Out[97]: float

1-NUMERIC :- INT || FLOAT || COMPLEX || BOOL

Arithmetic operator

In [98]:
```
x1, y1 = 10, 5
```

In [99]:
```
#x1 ^ y1
```

In [100…
```
x1 + y1
```

Out[100…
```
15
```

In [101…
```
x1 - y1
```

Out[101…
```
5
```

In [102…
```
x1 * y1
```

Out[102…
```
50
```

In [104…
```
x1 / y1
```

Out[104…
```
2.0
```

In [105…
```
x1 // y1
```

Out[105…
```
2
```

In [106…
```
x1 % y1
```

Out[106…
```
0
```

In [107…
```
x1 ** y1
```

Out[107…
```
100000
```

In [108…
```
3 ** 2
```

Out[108…
```
9
```

In [109…
```
x2 = 3
y2 = 3

x2 ** y2
```

Out[109…
```
27
```

Assignment operator

In [110…
```
x = 2
```

In [111…
```
x = x + 2
```

In [112…
```
x
```

Out[112… 4

In [113…
```
x += 2
x
```

Out[113… 6

In [114…
```
x += 2
x
```

Out[114… 8

In [115…
```
x *=2
```

In [116…
```
x
```

Out[116… 16

In [117…
```
x -= 2
```

In [118…
```
x
```

Out[118… 14

In [119…
```
x /= 2
x
```

Out[119… 7.0

In [120…
```
x //=2
x
```

Out[120… 3.0

In [121…
```
a, b = 5,6
print(a)
print(b)
```

```
5
6
```

In [122…
```
a = 5
b = 6
print(a)
print(b)
```

```
5
6
```

```
a
```

In [124…
```
b
```
Out[124… 6

unary operator

In [125…
```
n = 7
n
```
Out[125… 7

In [126…
```
m = -(n)
m
```
Out[126… -7

In [127…
```
n
```
Out[127… 7

In [128…
```
-n
```
Out[128… -7

Relational operator

In [129…
```
a = 5
b = 6
```
In [130…
```
a<b
```
Out[130… True

In [131…
```
a>b
```
Out[131… False

In [132…
```
a == b
```
Out[132… False

In [133…
```
a != b
```
Out[133… True

In [134…
```
b = 5
```
In [135…
```
a == b
```
Out[135… True

In [136…
```
a
```
Out[136… 5

In [137…
```
b
```
Out[137… 5

In [138…
```
a > b
```
Out[138… False

In [139…
```
a >= b
```
Out[139… True

In [140…
```
a <= b
```
Out[140… True

In [141…
```
a < b
```
Out[141… False

In [142…
```
a>b
```
Out[142… False

In [143…
```
b = 7
```
In [144…
```
a != b
```
Out[144… True

Logical Operator .And .Or .Not

In [145…
```
a = 5
b = 4
```
In [146…
```
a < 8 and b < 5
```
Out[146… True

In [147…
```
a < 8 and b < 2
```
Out[147… False

In [148…
```
a < 8 or b < 2
```
Out[148… True

In [149…
```
a>8 or b<2
```

Out[149… False

In [150…
```
x = False
x
```

Out[150… False

In [151…
```
not x
```

Out[151… True

In [152…
```
x = not x
x
```

Out[152… True

In [153…
```
x
```

Out[153… True

In [154…
```
not x
```

Out[154… False

In [155…
```
25
```

Out[155… 25

In [156…
```
bin(25)
```

Out[156… '0b11001'

In [157…
```
int(0b11001)
```

Out[157… 25

In [158…
```
bin(30)
```

Out[158… '0b11110'

In [159…
```
int(0b11110)
```

Out[159… 30

In [160…
```
int(0b11001)
```

Out[160… 25

In [161…
```
oct(25)
```

Out[161… '0o31'

In [162…
```
int(0o31)
```
Out[162… 25

In [163…
```
int(0b11110)
```
Out[163… 30

In [164…
```
0o31
```
Out[164… 25

In [165…
```
0b11001
```
Out[165… 25

In [166…
```
int(0b11001)
```
Out[166… 25

In [167…
```
bin(7)
```
Out[167… '0b111'

In [168…
```
oct(25)
```
Out[168… '0o31'

In [169…
```
0o31
```
Out[169… 25

In [170…
```
int(0o31)
```
Out[170… 25

In [171…
```
hex(25)
```
Out[171… '0x19'

In [172…
```
0x19
```
Out[172… 25

In [173…
```
hex(16)
```
Out[173… '0x10'

In [174…
```
0xa
```
Out[174… 10

In [175…
```
0xb
```

Out[175… 11

In [176…
```
hex(1)
```

Out[176… '0x1'

In [177…
```
hex(2)
```

Out[177… '0x2'

In [178…
```
hex(8)
```

Out[178… '0x8'

In [179…
```
hex(10)
```

Out[179… '0xa'

In [180…
```
hex(11)
```

Out[180… '0xb'

In [181…
```
hex(256)
```

Out[181… '0x100'

In [182…
```
0x19
```

Out[182… 25

In [183…
```
0x15
```

Out[183… 21

In [184…
```
a = 5
b = 6
```

In [185…
```
a = b
b = a
```

In [186…
```
print(a)
print(b)
```

```
6
6
```

In [187…
```
a1 = 7
b1 = 8
```

In [188…
```
temp = a1
a1 = b1
b1 = temp
```

In [189…
```
print(a1)
print(b1)
```
```
8
7
```

THIRD VARIABLE

In [190…
```
a2 = 5
b2 = 6
```

In [191…
```
a2 = a2 + b2
b2 = a2 - b2
a2 = a2 - b2
```

In [192…
```
print(a2)
print(b2)
```
```
6
5
```

In [ ]:

In [193…
```
0b110
```

Out[193… 6

In [194…
```
0b101
```

Out[194… 5

In [195…
```
print(0b110)
print(0b101)
```
```
6
5
```

In [196…
```
print(0b101)
print(0b110)
```
```
5
6
```

In [197…
```
print(bin(11))
```
```
0b1011
```

In [198…
```
print(0b1011)
```
```
11
```

XOR

In [199…
```
print(a2)
print(b2)
```

```
6
5
```

In [200…
```
a2 = a2 ^ b2
b2 = a2 ^ b2
a2 = a2 ^ b2
```

In [201…
```
print(a2)
print(b2)
```

```
5
6
```

In [202…
```
a2, b2
```

Out[202… (5, 6)

In [203…
```
a2, b2 = b2, a2
```

In [204…
```
print(a2)
print(b2)
```

```
6
5
```

In [205…
```
print(bin(12))
print(bin(13))
```

```
0b1100
0b1101
```

In [206…
```
0b1101
```

Out[206… 13

In [207…
```
0b1100
```

Out[207… 12

In [208…
```
~12
```

Out[208… -13

In [209…
```
~46
```

Out[209… -47

In [210…
```
~54
```

Out[210… -55

In [211…
```
~10
```

Out[211… -11

bit wise and operator

In [212…
```
12 & 13
```
Out[212… 12

In [213…
```
12 | 13
```
Out[213… 13

In [214…
```
1 & 0
```
Out[214… 0

In [215…
```
1 | 0
```
Out[215… 1

In [216…
```
bin(13)
```
Out[216… '0b1101'

In [217…
```
print(bin(35))
print(bin(40))
```
```
0b100011
0b101000
```

In [218…
```
35 & 40
```
Out[218… 32

In [219…
```
35 | 40
```
Out[219… 43

In [220…
```
12 ^ 13
```
Out[220… 1

In [221…
```
print(bin(25))
print(bin(30))
```
```
0b11001
0b11110
```

In [222…
```
25^30
```
Out[222… 7

In [223…
```
bin(7)
```
Out[223… '0b111'

In [224…
```
bin(25)
```
Out[224… '0b11001'

In [225…
```
bin(30)
```
Out[225… '0b11110'

In [226…
```
0b00111
```
Out[226… 7

In [227…
```
bin(10)
```
Out[227… '0b1010'

In [228…
```
10<<1
```
Out[228… 20

In [229…
```
10<<2
```
Out[229… 40

In [230…
```
bin(10)
```
Out[230… '0b1010'

In [231…
```
10<<1
```
Out[231… 20

In [232…
```
10<<2
```
Out[232… 40

In [233…
```
10<<3
```
Out[233… 80

In [234…
```
bin(20)
```
Out[234… '0b10100'

In [235…
```
20<<4
```
Out[235… 320

PYTHON PRINY STATEMENT TASK

In [236…
```
a=10
b=20
a
```

```
b
```

Out[236… 20

In [237…
```
a=10
b=20
print(a)
print(b)
```

```
10
20
```

In [238…
```
print(10)
print(10,20)
print('python')
print(10,20,'python')
```

```
10
10 20
python
10 20 python
```

In [239…
```
num1=20
num2=30
add=num1+num2
print(add)
```

```
50
```

print result with string

In [240…
```
num1=20
num2=30
add=num1+num2
print('The addition of',num1,'and',num2,'is=',add)
```

```
The addition of 20 and 30 is= 50
```

In [241…
```
name='Python'
age=20
city='hyd'
```

In [242…
```
print('My name is',name,'and i am',age,'years old form',city)
```

```
My name is Python and i am 20 years old form hyd
```

print Format method

In [243…
```
num1=20
num2=30
add=num1+num2
print('The addition of {} and {} is= {}'.format(num1,num2,add))
```

```
The addition of 20 and 30 is= 50
```

In [244…
```
name='python'
age=20
city='hyd'
```

In [245…
```
print('hello my name is {}, and i am {} years old from {}'.format(name,age,city))
```

```
hello my name is python, and i am 20 years old from hyd
```

In [246…
```
num1=100
num2=25
num3=333
avg=(num1+num2+num3)/3
avg1=round((num1+num2+num3)/3,2)

print('The avrage of {}, {}, and {} is= {} or {}'.format(num1,num2,num3,
avg,avg1))
```

```
The avrage of 100, 25, and 333 is= 152.66666666666666 or 152.67
```

In [247…
```
round(avg,2)
```

Out[247… 152.67

In [248…
```
num1=20
num2=30
```

In [249…
```
add=num1+num2
print(f'The addition of {num1} and {num2} is= {add}')
```

```
The addition of 20 and 30 is= 50
```

In [250…
```
name='Python'
age=20
city='hyd'
```

In [251…
```
print(f'hello my name is {name}, and i am {age} year old, from {city}.')
```

```
hello my name is Python, and i am 20 year old, from hyd.
```

In [252…
```
num1=100
num2=25
num3=333
avg=round((num1+num2+num3)/3,2)
```

In [253…
```
print(f'The avrage of {num1}, {num2} and {num3} is = {avg}')
```

```
The avrage of 100, 25 and 333 is = 152.67
```

In [254…
```
num1=10
num2=20
add = num1+ num2
print('The addition of',num1,'and',num2,'is=',add)
print('The addition of {} and {} is= {}'.format(num1,num2,add))
print(f'The addition of {num1} and {num2} is= {add}')
```

```
The addition of 10 and 20 is= 30
The addition of 10 and 20 is= 30
The addition of 10 and 20 is= 30
```

end statement

In [255…
```
print('hello')
print('good moorning')
```

```
hello
good moorning
```

In [256…
```
print('hello',end='')
print('world good day')
```

```
helloworld good day
```

seprator

In [257…
```
print('hello','hai','how are you',sep='--->')
```

```
hello--->hai--->how are you
```

In [258…
```
print('hello','hai','how are you',sep='&')
```

```
hello&hai&how are you
```

In [259…
```
print('hello','hai','how are you',sep='@')
```

```
hello@hai@how are you
```

In [261…
```
print('hello','hai','how are you',sep=' ')
```

```
hello hai how are you
```

In [262…
```
print(3,'.')
```

```
3 .
```

In [263…
```
print(3,'.',sep='')
```

```
3.
```

In [264…
```
print(1,2,end=' ')
print(3,'.',sep='')
```

```
1 2 3.
```

In [ ]: