```
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 [18]: 15 % 2

Out[18]: 1

In [19]: 10 % 2

Out[19]: 0

In [20]: 15 %% 2

           Cell In[20], line 1
             15 %% 2
                 ^
         SyntaxError: invalid syntax



In [21]: 3 + 'nit'

         ---------------------------------------------------------------------------
         -
         TypeError                                 Traceback (most recent call las
         t)
         Cell In[21], line 1
         ----> 1 3 + 'nit'

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

In [22]:

```
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 [23]:

```
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 [24]:

```
type(c)
```

Out[24]:

```
str
```

In [25]:

```
'Naresh IT'
```

Out[25]:

```
'Naresh IT'
```

In [26]:

```
print('Max it')
```

```
Max it
```

In [27]:

```
"max it technology"
```

Out[27]:

```
'max it technology'
```

In [28]:

```
s1 = 'max it technology'
s1
```

Out[28]:

```
'max it technology'
```

In [29]:

```
a = 2
b = 3
a + b
```

Out[29]:

```
5
```

In [30]:
```
c = a + b
c
```

Out[30]:
```
5
```

In [31]:
```
a = 3
b = 'hi'
type(b)
```

Out[31]:
```
str
```

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

```
  Cell In[32], line 1
    print('max it's"Technology"') # \ has some special meaning to ignore t
he error
                               ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [33]:
```
print('max it\'s"Technology"') #\ has some special meaning to ignore the er
```

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

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

```
max it Technology
```

In [35]:
```
print("max it',  'Technology")
```

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

In [36]:
```

'nit' + ' nit'
```

Out[36]:
```
'nit nit'
```

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

Out[37]:
```
'nit nit'
```

In [38]:
```

5 * 'nit'
```

Out[38]:
```
'nitnitnitnitnit'
```

In [39]:
```
5*' nit'
```

Out[39]:
```
' nit nit nit nit nit'
```

In [40]:

```
print('c:\nit')
```

```
c:
it
```

In [41]:

```
print(r'c:\nit')
```

```
c:\nit
```

In [42]:

```
2
```

Out[42]:

```
2
```

In [43]:

```
x = 2
x
```

Out[43]:

```
2
```

In [44]:

```
x + 3
```

Out[44]:

```
5
```

In [45]:

```
y = 3
y
```

Out[45]:

```
3
```

In [46]:

```
x + y
```

Out[46]:

```
5
```

In [47]:

```
x = 9
x
```

Out[47]:

```
9
```

In [48]:

```
x + y
```

Out[48]:

```
12
```

In [49]:

```
x + 10
```

Out[49]:

```
19
```

In [50]:

```
_ + y
```

Out[50]:

```
22
```

In [51]:

```
_ + y
```

Out[51]:

```
25
```

```
In [52]: _ + y

Out[52]: 28

In [53]: _ + y

Out[53]: 31

In [54]: y

Out[54]: 3

In [55]: _ + y

Out[55]: 6

In [56]: _ + y

Out[56]: 9

In [57]: _ + y

Out[57]: 12

In [58]: 
         name = 'mit'

In [59]: name

Out[59]: 'mit'

In [60]: name + 'technology'

Out[60]: 'mittechnology'

In [61]: name + ' technology'

Out[61]: 'mit technology'

In [62]: name 'technology'

           Cell In[62], line 1
             name 'technology'
                  ^
         SyntaxError: invalid syntax



In [63]: name

Out[63]: 'mit'
```

```
In [64]: len(name)

Out[64]: 3
```

```
In [65]: name[0]

Out[65]: 'm'
```

```
In [66]: name[5]
```

```
--------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Cell In[66], line 1
----> 1 name[5]

IndexError: string index out of range
```

```
In [67]: name[7]
```

```
--------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Cell In[67], line 1
----> 1 name[7]

IndexError: string index out of range
```

```
In [68]: name[-1]

Out[68]: 't'
```

```
In [69]: name[-2]

Out[69]: 'i'
```

```
In [70]: name[-6]
```

```
--------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
Cell In[70], line 1
----> 1 name[-6]

IndexError: string index out of range
```

```
In [71]: name

Out[71]: 'mit'
```

```
In [72]: name[0:1] #to print 2 character

Out[72]: 'm'
```

```
In [73]: name[0:2]

Out[73]: 'mi'
```

```
In [74]: name[1:4]

Out[74]: 'it'
```

```
In [75]: name[1:]

Out[75]: 'it'
```

```
In [76]: name[:4]

Out[76]: 'mit'
```

```
In [77]: name[3:9]

Out[77]: ''
```

```
In [78]: name

Out[78]: 'mit'
```

```
In [79]: name1 = 'fine'
         name1

Out[79]: 'fine'
```

```
In [80]: name1[0:1]

Out[80]: 'f'
```

```
In [83]: name1[0:1] = 'd

           Cell In[83], line 1
             name1[0:1] = 'd
                          ^
         SyntaxError: unterminated string literal (detected at line 1)
```

```
In [84]: name1[0] = 'd'
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[84], line 1
----> 1 name1[0] = 'd'

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

```
In [85]: name1
```

```
Out[85]: 'fine'
```

```
In [86]: name1[1:]
```

```
Out[86]: 'ine'
```

```
In [87]: 'd' + name1[1:]
```

```
Out[87]: 'dine'
```

```
In [88]: num1.insert(2,'nit')
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[88], line 1
----> 1 num1.insert(2,'nit')

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

```
In [89]: num = 5
         id(num)
```

```
Out[89]: 140711955829672
```

```
In [90]: name = 'nit'
         id(name)
```

```
Out[90]: 1806250763888
```

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

```
Out[91]: 140711955829832
```

```
In [92]:  b = a

In [93]:  id(b)

Out[93]:  140711955829832

In [94]:  id(10)

Out[94]:  140711955829832

In [95]:  k = 10
          id(k)

Out[95]:  140711955829832

In [96]:  a = 20
          id(a)

Out[96]:  140711955830152

In [97]:  id(b)

Out[97]:  140711955829832

In [98]:  PI = 3.14
          PI

Out[98]:  3.14

In [99]:  PI = 3.15
          PI

Out[99]:  3.15

In [100]: type(PI)

Out[100]: float

In [101]: x1, y1 = 10, 5

In [102]: #x1 ^ y1

In [103]: x1  + y1

Out[103]: 15

In [104]: x1 - y1

Out[104]: 5
```

In [105]:

```
x1 * y1
```

Out[105]:

```
50
```

In [106]:

```
x1 / y1
```

Out[106]:

```
2.0
```

In [107]:

```
x1 // y1
```

Out[107]:

```
2
```

In [108]:

```
x1 % y1
```

Out[108]:

```
0
```

In [109]:

```
x1 ** y1
```

Out[109]:

```
100000
```

In [110]:

```
x2 = 3
y2 = 2
x2 ** y2
```

Out[110]:

```
9
```

Assignment operator

In [111]:

```
x = 2
```

In [112]:

```
x = x + 2 # if you want to increment by 2
```

In [113]:

```
x
```

Out[113]:

```
4
```

In [114]:

```
x += 2
x
```

Out[114]:

```
6
```

In [115]:

```
x += 2
x
```

Out[115]:

```
8
```

In [116]:

```
x *= 2
```

```
In [117]: x

Out[117]: 16

In [118]: x -= 2

In [119]: x

Out[119]: 14

In [120]: x /= 2
          x

Out[120]: 7.0

In [121]: x //= 2
          x

Out[121]: 3.0

In [122]: a, b = 5,6 # you can assigned variable in one line as well

In [123]: a

Out[123]: 5

In [124]: b

Out[124]: 6

In [126]: n = 7
          n

Out[126]: 7

In [127]: m = -(n)
          m

Out[127]: -7

In [128]: n

Out[128]: 7

In [129]: -n

Out[129]: -7
```

In [130]:

```
a = 5
b = 6
```

In [131]:

```
a<b
```

Out[131]:

```
True
```

In [132]:

```
a>b
```

Out[132]:

```
False
```

In [133]:

```
# a = b # we cannot use = operatro that means it is assigning
```

In [134]:

```
a == b
```

Out[134]:

```
False
```

In [135]:

```
a != b
```

Out[135]:

```
True
```

In [136]:

```
# hear if i change b = 6
b = 5
```

In [137]:

```
a == b
```

Out[137]:

```
True
```

In [138]:

```
a
```

Out[138]:

```
5
```

In [139]:

```
b
```

Out[139]:

```
5
```

In [140]:

```
a >= b
```

Out[140]:

```
True
```

In [141]:

```
a <= b
```

Out[141]:

```
True
```

In [142]:

```
a < b
```

Out[142]:

```
False
```

```
In [143]: a>b

Out[143]: False

In [144]: b = 7

In [145]: a != b

Out[145]: True

In [146]: a = 5
          b = 4

In [147]: a < 8 and b < 5 #refer to the truth table

Out[147]: True

In [148]: a < 8 and b < 2

Out[148]: False

In [149]: a < 8 or b < 2

Out[149]: True

In [150]: a>8 or b<2

Out[150]: False

In [151]: x = False
          x

Out[151]: False

In [152]: not x  # you can reverse the operation

Out[152]: True

In [ ]:

In [ ]:
```