In [3]:
```
v=5  #vaiable declaration
v
```

Out[3]: 5

In [4]:
```
id(v)  #address of the memory location
```

Out[4]: 140727510116920

In [5]:
```
5=v # invalid
```

```
  Cell In[5], line 1
    5=v
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='?
```

In [6]:
```
1v=9
```

```
  Cell In[6], line 1
    1v=9
    ^
SyntaxError: invalid decimal literal
```

In [7]:
```
v1=9
v1
```

Out[7]: 9

In [8]:
```
v2=10
V2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[8], line 2
      1 v2=10
----> 2 V2

NameError: name 'V2' is not defined
```

In [9]:
```
a2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 1
----> 1 a2

NameError: name 'a2' is not defined
```

In [10]:
```
a2=10
a2
```

Out[10]: 10

In [11]:
```
v@=78
v
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[11], line 1
----> 1 v@=78
      2 v

TypeError: unsupported operand type(s) for @=: 'int' and 'int'
```

In [12]:
```
v*=22
v*
```

```
  Cell In[12], line 2
    v*
      ^
SyntaxError: invalid syntax
```

In [13]:
```
v_=10
v_
```

Out[13]: 10

In [14]:
```
if=78
```

```
  Cell In[14], line 1
    if=78
      ^
SyntaxError: invalid syntax
```

In [15]:
```
else=9
```

```
  Cell In[15], line 1
    else=9
    ^
SyntaxError: invalid syntax
```

In [16]:
```
for=9
```

```
  Cell In[16], line 1
    for=9
        ^
SyntaxError: invalid syntax
```

In [17]:
```
import keyword
keyword.kwlist
```

Out[17]:
```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [19]:
```
len(keyword.kwlist)
```

Out[19]:
```
35
```

In [20]:
```
else=6
```

```
  Cell In[20], line 1
    else=6
    ^
SyntaxError: invalid syntax
```

In [22]:
```
ELSE=6
ELSE
```

Out[22]:
```
6
```

In [23]:
```
for=78
for
```

```
  Cell In[23], line 1
    for=78
        ^
SyntaxError: invalid syntax
```

In [24]:
```
FOR=78
FOR
```

Out[24]: 78

In [25]:
```
prefix='py'
prefix='thon'
prefix
```

Out[25]: 'thon'

In [26]:
```
'py'+prefix
```

Out[26]: 'python'

In [27]:
```
continue=1
continue
```

```
  Cell In[27], line 1
    continue=1
             ^
SyntaxError: invalid syntax
```

In [28]:
```
CONTINUE=1
CONTINUE
```

Out[28]: 1

In [29]:
```
ASSERT=1
ASSERT
```

Out[29]: 1

In [30]:
```
assert=1
assert
```

```
  Cell In[30], line 1
    assert=1
           ^
SyntaxError: invalid syntax
```

In [31]:
```
IMPORT=1
IMPORT
```

Out[31]: 1

In [32]:
```
import=1
import
```

```
  Cell In[32], line 1
    import=1
           ^
SyntaxError: invalid syntax
```

In [33]:
```
spam=1
spam
```

Out[33]: 1

In [34]:
```
SPAM=1
SPAM
```

Out[34]: 1

In [ ]:
```
TASK-1
```

In [1]:
```
2+2
```

Out[1]: 4

In [2]:
```
6-9
```

Out[2]: -3

In [3]:
```
6+5-9*7
```

Out[3]: -52

In [4]:
```
(4+5*6)/6
```

Out[4]: 5.666666666666667

In [5]:
```
34/6     #division always returns with float point number
```

Out[5]: 5.666666666666667

In [7]:
```
8/6
```

Out[7]: 1.3333333333333333

In [8]:
```
8//6 #the // operator discards the float
```

Out[8]: 1

In [9]:
```
8%6  #the % operator gives remainder value
```

Out[9]: 2

In [12]:
```
1*6+2 #floored quotient*divisor+remainder
```

Out[12]: 8

In [14]:
```
4**2  #the ** opeator to calculate power
```

Out[14]: 16

In [16]:
```
8**3 #8 to the power of 3
```

Out[16]: 512

In [18]:
```
length=34
breadth=2*2
length*breadth
```

Out[18]: 136

In [19]:
```
n
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[19], line 1
----> 1 n

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

In [10]:
```
tax=12.5/100
price=100.5
price*tax
```

Out[10]: 12.5625

In [3]:
```
price + _
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 price + _

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

In [30]:
```
s='firstline\nsecondline'
s # with out print()
```

Out[30]: 'firstline\nsecondline'

In [31]:
```
s='firstline\nsecondline'
print(s)  #with print() \n produces the new line
```

```
firstline
secondline
```

In [38]:
```
print("c;some\name")  #here\n new line

```

```
c;some
ame
```

In [35]:
```
print(r'c;some\name')  #r is used before the first quote for not interpriting \ cha
```

```
c;some\name
```

In [43]:
```
print('''
hai      hellow[]

welcome to python
ok bi
''')
```

```
hai      hellow[]

welcome to python
ok bi
```

In [48]:
```
2*'python+lib'  # 2 times python+lib coz 'sigle is at end of quote
```

Out[48]:
```
'python+libpython+lib'
```

In [51]:
```
2*'python'+"lib "  # 2 times python followed by lib
```

Out[51]:
```
'pythonpythonlib '
```

In [61]:
```
3*"num"+3*"py"
```

Out[61]:
```
'numnumnumpypypy'
```

In [71]:
```
"he llo"  #strings r not next to each other
```

Out[71]:
```
'he llo'
```

In [73]:
```
"he""llo"  #strings are next to each other
```

Out[73]:
```
'hello'
```

In [83]:
```
string=('hello''world''goodmorning')
string
```

Out[83]:
```
'helloworldgoodmorning'
```

In [93]:
```
prefix= 'py'  #variable and string can't concanete
prefix 'thon'
```

```
  Cell In[93], line 2
    prefix 'thon'
           ^
SyntaxError: invalid syntax
```

In [100…
```
prefix='py'
prefix='thon'
prefix+'thon' #takes the previous prefix in output+thon
```

Out[100… 'thonthon'

In [101…
```
prefix='py'
prefix='thon'
'py'+prefix #py+ takes previous prefix
```

Out[101… 'python'

In [103…
```
prefix='thon'
prefix='py'
prefix+'thon'
```

Out[103… 'python'

In [6]:
```
word='python'
word[0] #character in position 0
```

Out[6]: 'p'

In [110…
```
word[2]
```

Out[110… 't'

In [112…
```
word[-1] #here - indicate last value
```

Out[112… 'n'

In [7]:
```
print(word[0])
print(word[1])
print(word[2])
```

```
p
y
t
```

In [124…
```
word[2]+word[4]
```

Out[124… 'to'

In [125…
```
word[0:4] # characters from 0 included 4 is excluded
```

Out[125… 'pyth'

In [126…
```
word[:3]
```

Out[126… 'pyt'

In [130…
```
word[-2:2]
```

Out[130… ''

In [128…
```
word[-2:-5]
```

Out[128… ''

In [136…
```
word[0:3]+word[3:6]
```

Out[136… 'python'

In [140…
```
word[-1]+'oht'
```

Out[140… 'noht'

In [141…
```
word[:4]+'n'
```

Out[141… 'pythn'

In [17]:
```
a = "ramya"
```

In [20]:
```
a[-5:-4]
```

Out[20]: 'r'

In [45]:
```
add=[1,2,3,4]
add
```

Out[45]: [1, 2, 3, 4]

In [170…
```
letter='a,b,c,d'
letters
```

Out[170… 'a,b,c,d'

In [53]:
```
letters[0:2]='e','f'
letters
```

Out[53]: ['e', 'f', 'B', ',', 'C']

In [172…
```
a=1,2,3
b=4,5,6
x=[a,b]
x
```

Out[172… [(1, 2, 3), (4, 5, 6)]

In [174…
```
x[1]
```

Out[174… (4, 5, 6)

In [176…
```
x[1][1]
```

Out[176… 5

In [2]:
```
a='does\'t'
a
```

Out[2]:
```
"does't"
```

In [4]:
```
a='didn\'t'
a
```

Out[4]:
```
"didn't"
```

In [42]:
```
word='python'
word[-3:-2]
```

Out[42]:
```
'h'
```

In [1]:
```
price=10.234
tax=2.34/100
price*tax
```

Out[1]:
```
0.23947559999999998
```

In [6]:
```
price + _
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[6], line 1
----> 1 price + _

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

In [ ]:
```

```