In [1]:
```
v=8
v
```

Out[1]: 8

In [2]:
```
var=8
VAR
```

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

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

In [3]:
```
v_=20
v_
```

Out[3]: 20

In [4]:
```
1var=20
1var
```

```
  Cell In[4], line 1
    1var=20
    ^
SyntaxError: invalid decimal literal
```

In [5]:
```
var1=20
var1
```

Out[5]: 20

In [9]:
```
def=79
def
```

```
  Cell In[9], line 1
    def=79
        ^
SyntaxError: invalid syntax
```

In [10]:
```
DEF=79
DEF
```

Out[10]: 79

In [11]:
```
false=26
false
```

Out[11]: 26

In [12]:
```
False=26
False
```

```
  Cell In[12], line 1
    False=26
    ^
SyntaxError: cannot assign to False
```

In [13]:
```
true=8
true
```

Out[13]: 8

In [14]:
```
True=8
True
```

```
  Cell In[14], line 1
    True=8
    ^
SyntaxError: cannot assign to True
```

In [15]:
```
i=5
type(i)
```

Out[15]: int

In [16]:
```
f=100.46
type(f)
```

Out[16]: float

In [23]:
```
import keyword
```

In [24]:
```
keyword.kwlist
```

Out[24]:
```
['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 [25]:
```
print(keyword.kwlist)
```

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

In [26]:
```
i=5
f=4.5
i+f
```

Out[26]:
```
9.5
```

In [27]:
```
i-f
```

Out[27]:
```
0.5
```

In [28]:
```
i*f
```

Out[28]: 22.5

In [29]:
```
i/f
```

Out[29]: 1.1111111111111112

In [30]:
```
i//f
```

Out[30]: 1.0

In [31]:
```
True
```

Out[31]: True

In [32]:
```
int(True)
```

Out[32]: 1

In [33]:
```
int(False)
```

Out[33]: 0

In [34]:
```
True+False
```

Out[34]: 1

In [35]:
```
True+True
```

Out[35]: 2

In [36]:
```
False+False
```

Out[36]: 0

In [37]:
```
False*True
```

Out[37]: 0

True/True

In [38]:
```
True/True
```

Out[38]: 1.0

In [39]:
```
True//True
```

Out[39]: 1

In [40]:
```
True/False
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[40], line 1
----> 1 True/False

ZeroDivisionError: division by zero
```

In [41]:
```
False/True
```

Out[41]: 0.0

In [42]:
```
False//True
```

Out[42]: 0

In [43]:
```
s='hello'
s
```

Out[43]: 'hello'

In [44]:
```
s="hello"
s
```

Out[44]: 'hello'

In [46]:
```
s1='hello team'
```

In [47]:
```
s1
```

Out[47]: 'hello team'

In [48]:
```
c=10+20j
c
```

Out[48]: (10+20j)

In [49]:
```
c.real
```

Out[49]: 10.0

In [50]:
```
int(c.real)
```

Out[50]: 10

In [51]:
```
bool(c.real)
```

Out[51]: True

In [52]:
```
d=20+30j
print(c+d)
print(c-d)
```

```
(30+50j)
(-10-10j)
```

In [53]:
```
print(c*d)
```

```
(-400+700j)
```

In [54]:
```
print(c/d)
```

```
(0.6153846153846154+0.0769230769230769j)
```

In [55]:
```
int(2.4)
```

Out[55]: 2

In [56]:
```
int(2.4,3.5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[56], line 1
----> 1 int(2.4,3.5)

TypeError: 'float' object cannot be interpreted as an integer
```

In [57]:
```
float(2.1,3.4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[57], line 1
----> 1 float(2.1,3.4)

TypeError: float expected at most 1 argument, got 2
```

In [58]:
```
 int(True)
```

Out[58]: 1

In [59]:
```
int(False)
```

Out[59]: 0

In [60]:
```
int(True,False)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[60], line 1
----> 1 int(True,False)

TypeError: int() can't convert non-string with explicit base
```

In [61]:
```
int('10')
```

Out[61]: 10

In [62]:
```
int('Ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[62], line 1
----> 1 int('Ten')

ValueError: invalid literal for int() with base 10: 'Ten'
```

In [63]:
```
float(20000)
```

Out[63]: 20000.0

In [64]:
```
float(444.0)
```

Out[64]: 444.0

In [65]:
```
str(8)
```

Out[65]: '8'

In [66]:
```
str(1+2j)
```

Out[66]: '(1+2j)'

In [67]:
```
str()
```

Out[67]: ''

In [68]:
```
bool(7)
```

Out[68]: True

In [69]:
```
bool()
```

Out[69]: False

In [70]:
```
bool(0)
```

Out[70]: False

In [71]:
```
bool('Ten')
```

Out[71]: True

In [72]:
```
bool(ten)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[72], line 1
----> 1 bool(ten)

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

In [73]:
```
 complex(18,19)
```

Out[73]: (18+19j)

In [74]:
```
complex(17,-17)
```

Out[74]: (17-17j)

In [75]:
```
complex(True)
```

Out[75]: (1+0j)

In [76]:
```
complex(False)
```

Out[76]: 0j

In [77]:
```
complex('10')
```

Out[77]: (10+0j)

In [78]:
```
complex(True,False)
```

Out[78]: (1+0j)

In [80]:
```
f=1e0
f
```

Out[80]: 1.0

In [81]:
```
f1=2e1
```

In [82]:
```
f1
```

Out[82]: 20.0

In [84]:
```
f2=2.4e2
```

In [85]:
```
f2
```

Out[85]: 240.0

In [86]:
```
type(f2)
```

Out[86]: float

In [87]:
```
import numpy as np
```

In [88]:
```
a=np.nan
```

In [90]:
```
type(a)
```

Out[90]: float

In [91]:
```
i=34
id(i)
```

Out[91]:
```
140723641432008
```

In [92]:
```
i=87
p=87
q=87
print(id(i))
print(id(p))
print(id(q))
```

```
140723641433704
140723641433704
140723641433704
```

In [93]:
```
print(id(i,p,q))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[93], line 1
----> 1 print(id(i,p,q))

TypeError: id() takes exactly one argument (3 given)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: