

18th & 19th

```
v = 5 #variable declaration
```

```
v
```

```
5
```

```
id(v) # address of the memory location
```

```
140704131459640
```

```
5 = v
```

```
Cell In[3], line 1
```

```
5 = v
^
```

```
SyntaxError: cannot assign to literal here. Maybe you meant '=='
instead of '='?
```

```
1v = 9
```

```
Cell In[4], line 1
```

```
1v = 9
```

```
^
```

```
SyntaxError: invalid decimal literal
```

```
v1 = 9
```

```
v1
```

```
9
```

```
a2 = 10
```

```
A2
```

```
-----
-----
```

```
NameError                                Traceback (most recent call
last)
```

```
Cell In[4], line 2
```

```
1 a2 = 10
```

```
----> 2 A2
```

NameError: name 'A2' is not defined

a2

10

v@ = 78

v@

Cell In[6], line 1

v@ = 78

^

SyntaxError: invalid syntax

v* = 78

v*

Cell In[7], line 1

v* = 78

^

SyntaxError: invalid syntax

v_ = 10

v_

10

if = 78

Cell In[9], line 1

if = 78

^

SyntaxError: invalid syntax

else = 80

Cell In[10], line 1

else = 80

^

SyntaxError: invalid syntax

```
for = 89
```

Cell In[11], line 1

```
for = 89
```

^

SyntaxError: invalid syntax

```
import keyword
keyword.kwlist
```

```
['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']
```

```
len(keyword.kwlist)
```

```
35
```

```
else = 6
```

```
Cell In[14], line 1
```

```
    else = 6
```

```
    ^
```

```
SyntaxError: invalid syntax
```

```
ELSE = 6
```

```
ELSE
```

```
6
```

```
for = 78
```

```
for
```

```
Cell In[16], line 1
```

```
    for = 78
```

```
    ^
```

```
SyntaxError: invalid syntax
```

```
For = 90
```

```
FOR
```

```
-----  
-----
```

```
NameError                                Traceback (most recent call  
last)
```

```
Cell In[17], line 2
```

```
    1 For = 90
```

```
----> 2 FOR
```

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

```
prefix = 'py'  
prefix = 'thon'
```

```
prefix
```

```
'thon'
```

```
'py' + prefix
```

```
'python'
```

19th

```
i = 32
```

```
i
```

```
32
```

```
type(i)
```

```
int
```

```
f = 110.32
```

```
f
```

```
110.32
```

```
type(f)
```

```
float
```

```
f1 = 1e0
```

```
f1
```

```
1.0
```

```
f2 = 1e1
```

```
f2
```

```
10.0
```

```
f3 = 1e2
```

```
f3
```

```
100.0
```

```
f4 = 1e3
```

```
f4
```

```
1000.0
```

```
f5 = 1E3  
f5
```

```
1000.0
```

```
a = 10  
b = 20
```

```
a + b  
a - b  
a * b  
a / b
```

```
0.5
```

```
print(a+b)  
print(a-b)  
print(a*b)  
print(a/b)
```

```
30  
-10  
200  
0.5
```

```
num1=20  
num2=30  
add=num1+num2  
print('The addition of two number',num1,'and',num2,'is=',add)
```

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

```
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
```

```
num1=20  
num2=30  
num3 = 40
```

```
add=num1+num2+num3
```

```
print('The addition of {} and {} and {} is=  
{}`.format(num1,num2,num3,add))
```

```
The addition of 20 and 30 and 40 is= 90
```

```

c = 1 + 2j
c
(1+2j)
type(c)
complex
c
(1+2j)
c.real
1.0
c.imag
2.0
c = 5 + 10j
d = 10 + 20j
print(c+d)
print(c-d)
(15+30j)
(-5-10j)
def team():
    print('hello')
c2 = 5 + 10j
c2
(5+10j)
b = true
b

```

```

-----
NameError                                Traceback (most recent call
last)

```

```

Cell In[44], line 1

```

```

----> 1 b = true

```

```

      2 b

```

NameError: name 'true' is not defined

```
import keyword
keyword.kwlist
```

```
['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']
```

```
b = True
b
```

```
True
```

```
b1 = False
b1
```



```
False
int(True)
1
int(False)
0
True + True
2
True-False
1
False-True
-1
True-True*False+False
1
```

Practice

```
a=10
b=20
a
b

20

a=10
b=20
print(a)
print(b)

10
20

print(10)
print(10,20)
print('python')
print(10,20,'python')

10
10 20
```

```
python
10 20 python
```

```
num1=20
num2=30
add=num1+num2
print(add)
```

```
50
```

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

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

```
name='Python'
age=20
city='hyd'

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
```

```
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
```

```
name='Python'
age=20
city='hyd'

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
```

```
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
```

```
round(avg,2)
```

152.67

```
num1=20
num2=30
add=num1+num2
print(f'The addition of {num1} and {num2} is= {add}') #
```

The addition of 20 and 30 is= 50

```
name='Python'
age=20
city='hyd'
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.

```
num1=100
num2=25
num3=333
avg=round((num1+num2+num3)/3,2)
print(f'The avrage of {num1}, {num2} and {num3} is = {avg}')
```

The avrage of 100, 25 and 333 is = 152.67

```
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

```
print('hello')
print('good moorning')
```

hello
good moorning

```
print('hello', end=' ' )
print('world good day')
```

hello world good day

```
print('hello','hai','how are you',sep='--->')
```

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

```
print('hello','hai','how are you',sep='&')
```

hello&hai&how are you

```
print('hello','hai','how are you',sep='@')
```

hello@hai@how are you

```
print('hello','hai','how are you',sep=' ')
```

hello hai how are you

```
print(3,'.')
```

3 .

```
print(3,'.',sep='')
```

3.

```
print(1,2,end=' ')
```

```
print(3,'.',sep='')
```

1 2 3.