```
In [9]: 9
Out[9]: 9

In [11]: 9 + 9
Out[11]: 18

In [13]: 9 - 9
Out[13]: 0

In [16]: 9 * 9
Out[16]: 81

In [18]: 9 / 9 ##float division
Out[18]: 1.0

In [20]: 9 // 9 ## int division
Out[20]: 1

In [22]: 2 ** 3
Out[22]: 8

In [24]: (3 * 6)-8+3
Out[24]: 13

In [26]: 3 * 6-8+3 ##bodmas
Out[26]: 13

In [28]: 9 % 9
Out[28]: 0

In [30]: 3 > 5
Out[30]: False

In [32]: 3 < 5
Out[32]: True

In [36]: 3 == 5
```

```
Out[36]:  False

In [38]:  3 != 5

Out[38]:  True

In [ ]:   WORKING NUMBER PYTHON VTH PYTHON & PYTHON OPERATOR
          ## ARTHEMATIC OPERATOR[ --,-,+,/,//,*]

In [40]:  WELCOME TO NARESHIT

            Cell In[40], line 1
              WELCOME TO NARESHIT
                      ^
          SyntaxError: invalid syntax

In [42]:  'WELCOME TO NARESHIT'

Out[42]:  'WELCOME TO NARESHIT'

In [44]:  "WELCOME TO NARESHIT"

Out[44]:  'WELCOME TO NARESHIT'

In [46]:  '''WELCOME TO NARESHIT'''

Out[46]:  'WELCOME TO NARESHIT'

In [48]:  'WELCOME
          TO
          NARESHIT'

            Cell In[48], line 1
              'WELCOME
              ^
          SyntaxError: unterminated string literal (detected at line 1)

In [50]:  "WELCOME
          TO
          NARESHIT"

            Cell In[50], line 1
              "WELCOME
              ^
          SyntaxError: unterminated string literal (detected at line 1)

In [52]:  '''WELCOME
          TO
          NARESHIT'''

Out[52]:  'WELCOME\nTO\nNARESHIT'

In [ ]:   3- python(variable=object=identifier)
In [2]: v = 5
        v

Out[2]: 5

In [4]: type(v)

Out[4]: int

In [6]: 5 = v

  Cell In[6], line 1
    5 = v
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='?

In [8]: 1v = 5

  Cell In[8], line 1
    1v = 5
    ^
SyntaxError: invalid decimal literal

In [16]: v1 = 10
         v1

Out[16]: 10

In [18]: id(v1)

Out[18]: 140734387727064
```

8 MAY PYTHON VARIABLES

```
In [7]: v1 = 5.5
        v1

Out[7]: 5.5

In [14]: type(v1)
         float

Out[14]: float

In [16]: v_ = 9
         v_

Out[16]: 9

In [18]: if = 67
```

```
  Cell In[18], line 1
    if = 67
       ^
SyntaxError: invalid syntax
```

In [20]:
```
import keyword
keyword.kwlist
```

Out[20]:
```
['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 [22]:
```
len(keyword.kwlist)
```

Out[22]:
```
35
```

In [24]:
```
while = 9
```

```
  Cell In[24], line 1
    while = 9
          ^
SyntaxError: invalid syntax
```

```
In [26]: nit = 8
         NIT
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[26], line 2
      1 nit = 8
----> 2 NIT

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

```
In [28]: Nit = 8
         nit
```

```
Out[28]: 8
```

```
In [ ]: RULES TO IDENTIFY PYTHON VARIABLES
        -case sensitive
        -cannot start with digit
        -special symbol is not allowed
        -only_ is allowed
        -keywords or reserve can define as variable
```

```
In [ ]: VARIABLE NAME = VALUE VALUES ALSO CALLED
        data types
        -int
        -float
        -string
        -complex
        -boolen
```

```
In [ ]: INTEGER
```

```
In [32]: i = 7
         i
```

```
Out[32]: 7
```

```
In [34]: type(i)
         i
```

```
Out[34]: 7
```

```
In [ ]: FLOAT
```

```
In [38]: a, b = 10, 20
```

```
In [40]: c = a+b
         d = a-b
         c
         d
```

```
Out[40]:  -10
```

```
In [42]:  c = a+b
          d = a-b
          e = a * b
          f = a / b

          print(c)
          print(d)
          print(e)
          print(f)
```

```
30
-10
200
0.5
```

```
In [ ]:
```

10 MAY

```
In [19]:  s = 'nit'
          s
```

```
Out[19]:  'nit'
```

```
In [21]:  s1 = 'hello'
          s1
```

```
Out[21]:  'hello'
```

```
In [23]:  print(s)
          print(s1)
```

```
nit
hello
```

```
In [25]:  s1[:]
```

```
Out[25]:  'hello'
```

```
In [27]:  s1[1]
```

```
Out[27]:  'e'
```

```
In [29]:  s1[10]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[29], line 1
----> 1 s1[10]

IndexError: string index out of range
```

PYTHON DATA TYPES

```
In [34]: i = 5 (variable = value)
         type(i)

Out[34]: int

In [36]: f = 110.4
         f

Out[36]: 110.4

In [38]: f = 110.4
         f

Out[38]: 110.4

In [44]: type(f)

Out[44]: float

In [50]: c = 10 + 20j#real & imaginary part
         c

Out[50]: (10+20j)

In [52]: type(c)

Out[52]: complex

In [54]: c.real

Out[54]: 10.0

In [56]: c.imag

Out[56]: 20.0

In [62]: d = 5 + 3j
         d

Out[62]: (5+3j)

In [64]: print(c)
         print(d)

        (10+20j)
        (5+3j)

In [66]: c + d

Out[66]: (15+23j)
```

BOOLEN( TRUE OR FALSE)

```
In [75]: True
```

```
Out[75]: True
```

```
In [77]: False
```

```
Out[77]: False
```

```
In [79]: True + False
```

```
Out[79]: 1
```

```
In [81]: True - False
```

```
Out[81]: 1
```

```
In [83]: False  * False
```

```
Out[83]: 0
```

```
In [87]: b = True
         b1 = False
```

```
In [89]: print(b+b1)
         print(b-b1)
         print(b*b1)
         print(b1/b)
         print(b1//b)
```

```
1
1
0
0.0
0
```

```
In [ ]: FLOAT
```

```
In [70]: z = 4
         type(z)
```

```
Out[70]: int
```

```
In [72]: z = '4.4'
         type(z)
```

```
Out[72]: str
```

```
In [ ]:
```

```
In [ ]:
```

14 MAY PYTHON TYPECASTING

```
In [7]: v = 12 #v is variable& 12 is value
        v
```

```
Out[7]: 12
```

```
In [9]: ni$ = 78
```

```
  Cell In[9], line 1
    ni$ = 78
      ^
SyntaxError: invalid syntax
```

```
In [11]: if = 67 # if is a keyword
```

```
  Cell In[11], line 1
    if = 67
       ^
SyntaxError: invalid syntax
```

```
In [15]: import keyword
         keyword.kwlist
```

```
Out[15]: ['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 [23]: nit_ = 56
         nit_
```

```
Out[23]: 56
```

```
In [ ]: DATA TYPES
        i = 34 #integer
        f = 3.4 #float
        s = 'nit' #string
        b = True #boolen
        c = 1+2j #complex
```

```
In [27]: i, f, s, b, c = 34, 3.4, 'nit', True, 1+2j
In [29]: print(i)
         print(f)
         print(s)
         print(b)
         print(c)
```

```
34
3.4
nit
True
(1+2j)
```

```
In [31]: print(type(i))
         print(type(f))
         print(type(s))
         print(type(b))
         print(type(c))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'complex'>
```

# PYTHON TYPECASTING

```
In [34]: int(3.4) #flaat argument to integer
```

```
Out[34]: 3
```

```
In [36]: int(3.4, 4.5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[36], line 1
----> 1 int(3.4, 4.5)

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

```
In [38]: int(True)
```

```
Out[38]: 1
```

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

```
Out[40]: 4
```

```
In [42]: int(1+2j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[42], line 1
----> 1 int(1+2j)

TypeError: int() argument must be a string, a bytes-like object or a real number, no
t 'complex'
```

```
In [44]: int('10')
```

```
Out[44]: 10
```

```
In [46]: int('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[46], line 1
----> 1 int('ten')

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

```
In [1]: float(False)
```

```
Out[1]: 0.0
```

```
In [3]: complex(10)
```

```
Out[3]: (10+0j)
```

```
In [5]: complex(10,20)
```

```
Out[5]: (10+20j)
```

```
In [7]: complex(10,20,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[7], line 1
----> 1 complex(10,20,30)

TypeError: complex() takes at most 2 arguments (3 given)
```

```
In [11]: print(complex(3.4))
         print(complex(33.4, 34))
         print(complex('10'))
         print(complex(True,False))
```

```
(3.4+0j)
(33.4+34j)
(10+0j)
(1+0j)
```

```
In [13]: bool(1)
```

```
Out[13]: True
```

```
In [15]: bool(0)

Out[15]: False

In [17]: bool()

Out[17]: False

In [19]: bool(1+2j)

Out[19]: True

In [23]: bool(0+0j)

Out[23]: False

In [25]: bool('10')

Out[25]: True

In [27]: bool('ten')

Out[27]: True
```

15 MAY DATA STRUCTURE

```
In [46]: a = 5
         a

Out[46]: 5

In [48]: l=[]
         l

Out[48]: []

In [50]: type(l)

Out[50]: list

In [52]: print(len(l))

        0

In [54]: l.append(10)

In [56]: l

Out[56]: [10]

In [58]: l.append(20)
In [60]: l.append(30)
         l.append(40)
         l.append(50)
         l.append(50)

In [62]: l

Out[62]: [10, 20, 30, 40, 50, 50]

In [64]: l.append(60,70,80,90,100)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[64], line 1
----> 1 l.append(60,70,80,90,100)

TypeError: list.append() takes exactly one argument (5 given)

In [73]: l1=[]
         l1

Out[73]: []

In [75]: l1.append('nit')
         l1.append(True)
         l1.append(2.3)
         l1.append(1+2j)
         l1.append([1,2,3])

In [77]: print(l)
         print(l1)

[10, 20, 30, 40, 50, 50]
['nit', True, 2.3, (1+2j), [1, 2, 3]]

In [ ]: l1.append('nit)

In [81]: l1

Out[81]: ['nit', True, 2.3, (1+2j), [1, 2, 3]]

In [83]: type(l1)

Out[83]: list

In [85]: for i in l1:
             print(i)

nit
True
2.3
(1+2j)
[1, 2, 3]
```

In [87]:
```
for i in enumerate(l1):
    print(i)
```
```
(0, 'nit')
(1, True)
(2, 2.3)
(3, (1+2j))
(4, [1, 2, 3])
```

In [89]:
```
l[:]
```
Out[89]:
```
[10, 20, 30, 40, 50, 50]
```

In [91]:
```
l[0]
```
Out[91]:
```
10
```

In [93]:
```
l[-1]
```
Out[93]:
```
50
```

In [95]:
```
l[-3]
```
Out[95]:
```
40
```

In [97]:
```
l[-4]
```
Out[97]:
```
30
```

In [99]:
```
l[10]
```
```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[99], line 1
----> 1 l[10]

IndexError: list index out of range
```

In [101…
```
l[-3]
```
Out[101…
```
40
```

In [103…
```
l[1:4]
```
Out[103…
```
[20, 30, 40]
```

In [105…
```
l[0:6]
```
Out[105…
```
[10, 20, 30, 40, 50, 50]
```

In [107…
```
l[0:4]
```
Out[107…
```
[10, 20, 30, 40]
```

In [113…
```
l[0]=100
```

In [116…
```
l
```

Out[116…
```
[100, 20, 30, 40, 50, 50]
```

In [ ]: