```
In [2]: nit = 15 # variable are case sensitive.
        NIT
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[2], line 2
      1 nit = 15
----> 2 NIT

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

```
In [3]: nit
```

```
Out[3]: 15
```

```
In [4]: 1a = 67 # Variavle never starts with digits.
        1a
```

```
  Cell In[4], line 1
    1a = 67
    ^
SyntaxError: invalid decimal literal
```

```
In [5]: a1 = 67 # Variable never starts with digits but ends with digits.
        a1
```

```
Out[5]: 67
```

```
In [6]: nit$ = 89 # Special keywords are not allowed to define a variables. Except Unc
        nit$
```

```
  Cell In[6], line 1
    nit$ = 89
       ^
SyntaxError: invalid syntax
```

```
In [7]: x_train, x_test, y_train, y_test = 80, 20, 70, 30
```

```
In [8]: x_train
        x_test
        y_train
        y_test
```

```
Out[8]: 30
```

If we have to get all values we have to use print function print()

```
In [9]: print(x_train)
        print(x_test)
print(y_train)
print(y_test)
```

```
80
20
70
30
```

# In python print function are always ends with():

```
In [10]: import keyword
         keyword.kwlist
```

```
Out[10]: ['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 [13]: len(keyword.kwlist)
```

```
Out[13]: 35
```

```
In [27]: a10 = 78
         a9 = 89
```

```
In [15]: print(a10)
         print(a9)
```

```
78
89
```

```
In [28]: del a10 # del function is used for delete values.
```

```
In [29]: a10 # Deleted
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[29], line 1
----> 1 a10

NameError: name 'a10' is not defined
```

```
In [18]: for = 90  # keywords
```

```
  Cell In[18], line 1
    for = 90
        ^
SyntaxError: invalid syntax
```

```
In [20]: For = 90
         For
```

```
Out[20]: 90
```

DATA TYPES

Boolean

The Boolean (bool) type has two values: True and False.

```
In [21]: a = true # Case sensitive
         a
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[21], line 1
----> 1 a = true
      2 a

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

```
In [22]: a = True
         a
```

```
Out[22]: True
```

```
In [32]: b = False
         b
```

```
Out[32]: False
```

```
In [30]: type(a)
```

```
Out[30]: bool
```

```
In [33]: type(b)
```

```
Out[33]: bool
```

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

```
Out[40]: 1
```

```
In [41]: True - True
```

```
Out[41]: 0
```

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

```
Out[42]: 0
```

```
In [43]: False / True
```

```
Out[43]: 0.0
```

```
In [44]: False // True
```

```
Out[44]: 0
```

```
In [45]: True/False
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[45], line 1
----> 1 True/False

ZeroDivisionError: division by zero
```

```
In [34]: i = 25 # Value without decimal called integar
         i
```

```
Out[34]: 25
```

```
In [35]: type(i)
```

```
Out[35]: int

In [36]: print(type(i))

         <class 'int'>

In [38]: petrol = 109.50 # value with decimal called float data types.
         petrol

Out[38]: 109.5

In [39]: type(petrol)

Out[39]: float

In [46]: c1 = 10 + 20j
         c1

Out[46]: (10+20j)

In [47]: type(c1)

Out[47]: complex

In [48]: c1.real

Out[48]: 10.0

In [50]: c1.imaginary

         ---------------------------------------------------------------------------
         AttributeError                            Traceback (most recent call last)
         Cell In[50], line 1
         ----> 1 c1.imaginary

         AttributeError: 'complex' object has no attribute 'imaginary'

In [51]: c1.imag

Out[51]: 20.0

In [52]: c1

Out[52]: (10+20j)

In [53]: c2 = 20 + 30j

In [54]: print(c1)
         print(c2)

         (10+20j)
         (20+30j)
```

In [55]:
```
c1 + c2
```
Out[55]:
```
(30+50j)
```
In [56]:
```
c1 - c2
```
Out[56]:
```
(-10-10j)
```
In [57]:
```
c2 - c1
```
Out[57]:
```
(10+10j)
```
In [58]:
```
c1 * c2
```
Out[58]:
```
(-400+700j)
```

String

In [60]:
```
s = 'nareshit'
s
```
Out[60]:
```
'nareshit'
```
In [ ]:
```
s1 = "nareshit"
s1
```
In [62]:
```
s2 = '''naresh
      it'''
s2
```
Out[62]:
```
'naresh\n      it'
```

string slicing [:]

In [63]:
```
s
```
Out[63]:
```
'nareshit'
```
In [64]:
```
s[:]
```
Out[64]:
```
'nareshit'
```
In [65]:
```
s[3]
```
Out[65]:
```
'e'
```
In [66]:
```
s[4] #Forward Indexing
```

```
Out[66]: 's'

In [67]: s[-4] # Backward Indexing

Out[67]: 's'

In [68]: s[1:7]

Out[68]: 'areshi'

In [69]: s[10] #Index error

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[69], line 1
----> 1 s[10]

IndexError: string index out of range

In [ ]:
```