In [1]:
```
# Python Tokens:-
```

In [2]:
```
# A. Keywords :-
# => Keywords are reserve words in pythons. E.g. True, False, If, elif,
as, while, with, def, class, continue etc.
# => When keywords are written then they turn in to green color.
# => Keywords cannot be used for the name of identifiers such as variab
les, funtions  and objects.
```

In [3]:
```
# B. Identifiers:-
#=> Identifiers are the names given to variables, functions and object
s.
```

In [4]:
```
# Rules of defining an identifiers:-
# 1. Identifiers accept only underscore(_). They donot accept any speci
al characters except underscore(_).
# Examples:-
```

In [5]:
```
_x = 10
```

In [6]:
```
_x
```

Out[6]:
```
10
```

In [7]:
```
_y_ = 200
```

In [8]:
```
_y_
```

Out[8]:
```
200
```

In [9]:
```
z_ = " Sharon"
```

```
In [10]: z_
Out[10]: ' Sharon'

In [11]: # 2. Identifiers are case sensitive i.e. Uppercase and lowercase

In [12]: Mango = 900
         mango = " Kelly"

In [13]: Mango
Out[13]: 900

In [14]: mango
Out[14]: ' Kelly'

In [15]: # 3. Fisrt letter of identifiers cannot be a digit(0,1,2,3,4,5,6,7,8,
         9,), a number (0...9,10....99,100....999,1000....9999etc.)
         # and any special characters except underscore(_).

In [16]: # C. Literals:-
         # => Literals are the values assigned to identifiers. They donot change
         because they are constants.

In [17]: n1 = 980
         n1
Out[17]: 980

In [18]: v1 = " Molly"
         v1
Out[18]: ' Molly'

In [19]: # In the above example, n1 & v1 is are identifiers and 980 & 'Molly' ar
```

```
e  literals.
```

In [20]:
```
# D. Operators:-
# 1. Arithmetic Operators, 2. Relational Operators, and 3. Logical OPer
ators
```

In [21]:
```
#a. Arithmetic Operators:-
#Example:
```

In [22]:
```
m = 50
n = 30
m,n
```

Out[22]:
```
(50, 30)
```

In [23]:
```
m + n
```

Out[23]:
```
80
```

In [24]:
```
m - n
```

Out[24]:
```
20
```

In [25]:
```
n - m
```

Out[25]:
```
-20
```

In [26]:
```
m * n
```

Out[26]:
```
1500
```

In [27]:
```
n * m
```

Out[27]:
```
1500
```

In [28]:
```
m / n
```

```
Out[28]: 1.6666666666666667

In [29]: n / m

Out[29]: 0.6

In [30]: #b. Relational Operators:-
         #Example:

In [31]: p = -60
         q = 80
         p,q

Out[31]: (-60, 80)

In [32]: p > q

Out[32]: False

In [33]: p < q

Out[33]: True

In [34]: q > p

Out[34]: True

In [35]: q < p

Out[35]: False

In [36]: p == p

Out[36]: True

In [37]: p == q
```

```
Out[37]: False

In [38]: q == q

Out[38]: True

In [39]: q == p

Out[39]: False

In [40]: q != p

Out[40]: True

In [41]: p != q

Out[41]: True

In [42]: #c. Logical Operators:-
         #Example:

In [43]: x = True
         y = False
         x,y

Out[43]: (True, False)

In [44]: x & x

Out[44]: True

In [45]: x & y

Out[45]: False

In [46]: y & x
```

```
Out[46]: False

In [47]: y & y

Out[47]: False

In [48]: x | x

Out[48]: True

In [49]: x | y

Out[49]: True

In [50]: y | x

Out[50]: True

In [51]: y | y

Out[51]: False
```