In [2]:
```
name ='Python'
#strings are in red colour : quotes
#variables are in black colcour
#keywords are in green colour
```

In [3]:
```
type(name)
```

Out[3]:
```
str
```

In [4]:
```
name1="pyton"
type(name1)
```

Out[4]:
```
str
```

In [5]:
```
name
```

Out[5]:
```
'Python'
```

In [ ]:
```
```

In [6]:
```
name1
```

Out[6]:
```
'pyton'
```

In [7]:
```
print(name1) #careful with print statements
```

```
pyton
```

In [8]:
```
print(name)  #careful with print statements
```

```
Python
```

In [11]:
```
name2='i "like" python'
name2
#entire string in single quotes
#python word is in double quotes
```

Out[11]:
```
'i "like" python'
```

triplequotes

- triple quotes meaning is doc string
- if you want to convey the information to user
- in jupiter notebook we have the option called markdown option
- but in another platforms like vs code pycharms we dont have this mark down option
- at that place to convey the information we use triple quotes and that is called docstring.

In [ ]:
```
import random
random.randint()
```

- type
- max
- min
- reversed
- sorted

In [ ]:
```
name="python"
max(name)
#keyword(<variable_name>)
#print(name)
#type(name)
```

$ASCII - AmericanStandardCodeforInformationInterchange$

In [ ]:
```
- A:65
- a:97
```

$Ord\text{-}chr$

In [12]:
```
ord('p'),ord('y'),ord('t'),ord('h'),ord('n')
```

Out[12]: (112, 121, 116, 104, 110)

In [13]:
```
max('python')
```

Out[13]: 'y'

In [14]:
```
min('python')
```

Out[14]: 'h'

In [17]:
```
chr(112),ord('p')
```

Out[17]: ('p', 112)

In [22]:
```
str1='python123'
max(str1)
```

Out[22]: 'y'

In [23]:
```
min(str1)
```

Out[23]: '1'

In [24]:
```
str1="123-100"
max(str1)
```

Out[24]: '3'

In [26]:
```
ord("-"),ord("3")
```

Out[26]: (45, 51)

In [29]:
```
#How many ASCII values existing
for i in range(100):
    print(i,chr(i),end=' ')
```

```
0  1  2  3  4  5  6  7  8 9       10
 14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29
30  31  32   33 ! 34 " 35 # 36 $ 37 % 38 & 39 ' 40 ( 41 ) 42 * 43 + 44 , 45 -
46 . 47 / 48 0 49 1 50 2 51 3 52 4 53 5 54 6 55 7 56 8 57 9 58 : 59 ; 60 < 61 =
62 > 63 ? 64 @ 65 A 66 B 67 C 68 D 69 E 70 F 71 G 72 H 73 I 74 J 75 K 76 L 77 M
78 N 79 O 80 P 81 Q 82 R 83 S 84 T 85 U 86 V 87 W 88 X 89 Y 90 Z 91 [ 92 \ 93 ]
94 ^ 95 _ 96 ` 97 a 98 b 99 c
```

In [30]:
```
'Banana'>"banana"  #'B':66 "b":99 66>99
```

Out[30]: False

In [31]:
```
'Banana'>"BAnana"  # a='97' 'A':65
```

Out[31]: True

Len

In [32]:
```
str1='python'
len(str1)
```

Out[32]: 6

In [34]:
```
str1='i love python'
len(str1)
```

Out[34]: 13

In [35]:
```
str2 ='             ' #empty string
len(str2)
```

Out[35]: 13

In [36]:
```
str3=' ' # it has one space
len(str3)
```

Out[36]:
```
1
```

In [37]:
```
reversed('python')

#<output is stored inherently>
# we use the for loop
```

Out[37]:
```
<reversed at 0x27a75b70dc0>
```

in- operator

In [44]:
```
'p' in 'python'
'y' in 'python'
't' in 'python'
'h' in 'python'
'o' in 'python'
'n' in 'python'
nm='manchester united'
for i in 'python':

    print(i)#genralized
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[44], line 8
      6 'n' in 'python'
      7 nm='manchester united'
----> 8 for i in len(nm):
     10     print(i)

TypeError: 'int' object is not iterable
```

In [40]:
```

'Y' in 'python'
```

Out[40]:
```
False
```

In [46]:
```
for i in reversed('python'):

    print(i,end=' ')#genralized
```

```
n o h t y p
```

In [47]:
```
s1='apple'
for i in s1:
    print(i)
```

```
a
p
p
l
e
```

In [51]:
```
s2='french toast'
for i in s2:
    print(i,end=' ')
```

```
f r e n c h   t o a s t
```

In [52]:
```
sorted('python')
```

Out[52]:
```
['h', 'n', 'o', 'p', 't', 'y']
```

In [ ]:
```
#sorted keyowrd sort the letters
#[104,]
```

In [55]:
```
ord('p'),ord('y'),ord('t'),ord('h'),ord('n')
```

Out[55]:
```
(112, 121, 116, 104, 110)
```

In [58]:
```
# sorted gives the value in asending order
str1= "hello"
str2 = "python"
#concatenation
print(str1+str2)#'hellopyton'
print(str1-str2)#error
print(str1*str2)#error
print(str1/str2)#error
```

```
hellopython

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[58], line 6
      3 str2 = "python"
      5 print(str1+str2)
----> 6 print(str1-str2)
      7 print(str1*str2)
      8 print(str1/str2)

TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

In [59]:
```
str1+ ' '+ str2
```

Out[59]:
```
'hello python'
```

In [ ]:
```
type
max
min
 ord
 chr
reversed
sorted
concatenation
```

In [62]:
```
str1 ='python'
#-6 -5 -4-3-2-1 last value will be minus 1 always
#p y t h o n
#0 1 2 3 4 5
str1[0]
```

Out[62]:
```
'p'
```

In [63]:
```
str1[0],str1[1],str1[2],str1[3],str1[4],str1[5]
```

Out[63]:
```
('p', 'y', 't', 'h', 'o', 'n')
```

In [65]:
```
 for i in range (6):
        print(str1[i])
#i wnt to iterate this
#in : i means direct letter
#range : i means number
```

```
p
y
t
h
o
n
```

In [70]:
```
for i in range(6): #in range means number henace the out put will be number in ora
       print(str1[i],end=' ')#str1 of sqare bracket of itterator to get the value
```

```
p y t h o n
```

In [68]:
```
for i in str1:
    print(i, end=" ")
```

```
p y t h o n
```

In [ ]:
```

```