In [16]:
```
#1+2+3+4+5
```

In [17]:
```
def myAdd(a,b):
    return a+b

#myAdd(21.5,3)
```

In [18]:
```
def mySub(bigNo,smallNo):
    return bigNo-smallNo
#mySub(21,50)
```

In [ ]:
```
def myMul(a,b):
    return a*b
```

In [23]:
```
def myExp(base,power):
    return base**power
#myExp(2,3)
```

In [24]:
```
def myDiv(num,divider):
    return num/divider # 12/5 =2.4 exact
#myDiv(12,5)
```

In [25]:
```
def myDivInteger(num,divider):
    return num//divider # 12//5 =2( just the quotent)
#myDivInteger(12,5)
```

In [27]:
```
def myRemainder(num,divider):
    return num%divider #14%5 =4
#myRemainder(14,5)
```

In [32]:
```
def concat(a,b):
    return a+b

print(concat("hello",'baby'))
print(concat("hello",'      1'))
print(concat("hello",1))#Error #concatination between different data types not po
```

```
hellobaby
hello1
```

In [44]:
```
len("abc")#3
len((2,3))#2
len([111,"ansh",22.3,0,'a'])#5
#will show only the last len value
```

Out[44]:
```
5
```

In [42]:
```
chr(244)
```

Out[42]:
```
'ô'
```

In [43]:
```
ord('@')
```

Out[43]:
```
64
```

In [56]:
```
print(type(123)) #int
print(type(12999999999999999999999999993)) #still int , their is no long or big
print(type(True)) #boolean
print(type(2.41))#float
print(type(24.444444444441234884444444444)) #still float , ther is nodouble
print(type("ansh"))#string
print(type((1,"abc",24.6)))#tuple
print(type(123))#List
print(type({"a":1})) #dictionary
```

```
<class 'int'>
<class 'int'>
<class 'bool'>
<class 'float'>
<class 'float'>
<class 'str'>
<class 'tuple'>
<class 'int'>
<class 'dict'>
```

In [65]:
```
True==1 and False==0
```

Out[65]:
```
True
```

In [71]:
```
print("Value is ", round(3.91919001 ,0),"Rounded to",0,"places")
print("Value is ", round(3.91919001 ,1),"Rounded to",1,"places")
print("Value is ", round(3.91919001 ,2),"Rounded to",2,"places")
print("Value is ", round(3.91919001 ,3),"Rounded to",3,"places")
print("Value is ", round(3.91919001 ,4),"Rounded to",4,"places")
```

```
Value is  4.0 Rounded to 0 places
Value is  3.9 Rounded to 1 places
Value is  3.92 Rounded to 2 places
Value is  3.919 Rounded to 3 places
Value is  3.9192 Rounded to 4 places
```

In [72]:
```
"stone"*3
```

Out[72]:
```
'stonestonestone'
```

In [73]:
```
3*"Story"
```

Out[73]:
```
'StoryStoryStory'
```

In [74]:
```
3*'stone'+'age'
```

Out[74]:
```
'stonestonestoneage'
```

In [ ]: