Introduction

In Call by value, a copy of the variable is passed whereas in Call by reference, a variable itself is passed. In Call by value, actual and formal arguments will be created in different memory locations whereas in Call by reference, actual and formal arguments will be created in the same memory location

Call by Value

Immutable

where we can not make changes

In [4]:

```python
#for example

def new(x):
    print('old id of x in function is ',id(x))
    x=7
    print('new id of x is ',id(x))
j=11
print('id of j in outside the function is',id(j))
new(j)
```

```
id of j in outside the function is 1724544336
old id of x in function is  1724544336
new id of x is  1724544272
```

Call by reference

where values are mutable

here we can chnage the values suporting same address location

In [5]:

```python
l=[1,2,3,4,5]

def new(l):
    print('old id of list in function is',id(l))
    l[1]=25
    print('new id of list in function is',id(l))

print('old id of list out function is',id(l))
new(l)
```

```
old id of list out function is 98056040
old id of list in function is 98056040
new id of list in function is 98056040
```