In [5]:

```
lista = [2, "4, "c", [7, "a", 7]] #Index devuelve el primer elemento
#pos_7 = lista.index(7) #No está en el la lista y da error
pos_7 = lista[3].index(7)
print(pos_7)
```

```
0
```

In [9]:

```
l = ["primer", 2, "tercer", 4]

if "tercer" in l:
    l.pop(0) #Elimina por posición
    l.remove(4) #Elimina por valor

print(l)
```

```
[2, 'tercer']
```

In [12]:

```
l2 = [2, 3, 3, 3, 3, 3, 4]
for i in range(len(l2)): #Hará tantas iteraciones como elementos tenga l2
    if 3 in l2:
        l2.remove(3)
print(l2)
```

```
[2, 4]
```

In [13]:

```
l2 = [2, 3, 3, 3, 3, 3, 4]
for i in range(len(l2)): #Hará tantas iteraciones como elementos tenga l2
    if 3 in l2:
        l2.remove(3)
    print(l2)
```

```
[2, 3, 3, 3, 3, 4]
[2, 3, 3, 3, 4]
[2, 3, 3, 4]
[2, 3, 4]
[2, 4]
[2, 4]
[2, 4]
```

In [14]:

```
#FUNCIONES

def nombre_funcion():#Cuando defines una funcion python no entra, por eso no lo imprime d
os veces.
    print("Hola")
    x=2
    print(x)

nombre_funcion()
```

```
Hola
2
```

In [16]:

```
nombre_funcion()
```

```
Hola
2
```

In [20]:

```
def nombre_funcion():#Cuando defines una funcion python no entra, por eso no lo imprime dos veces.
    print("Hola")
    x=2
    print(x)

r = nombre_funcion()
print(r) #No retorna nada porque no hay return
```

```
Hola
2
None
```

In [22]:

```
def nombre_funcion():#Cuando defines una funcion python no entra, por eso no lo imprime dos veces.
    print("Hola")
    x=2
    print(x)
    return 7 #Para que retorne tienes que poner return, pero retorna el valor que tiene return a la deracha

r = nombre_funcion() #Lo que guarda la función es el return unicamente es decir 7, pero te imprime hola 2 también porque has llamado a la función también
print(r)
```

```
Hola
2
7
```

In [24]:

```
def suma_2_argumentos(a, b): #No puede albergar valores, solo variables. Es decir, no puede albegar 2, 4.
    return a + b

lo_que_retorna = suma_2_argumentos(a=2, b=4) #Es igual que poner = suma_2_argumentos(2, 4)
lo_que_retorna
```

Out[24]:

```
6
```

In [31]:

```
def add_element(lista1, to_add):
    lista1.append(to_add)

l = [2, 4, 6]
to_add = 9
add_element(lista1=l, to_add=9)
print(lista1) #No está definida
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-31-5560b9fcfc26> in <module>
      5 to_add = 9
      6 add_element(lista1=l, to_add=9)
----> 7 print(lista1) #No está definida

NameError: name 'lista1' is not defined
```

In [30]:

```
def add_element(lista1, to_add):
    lista1.append(to_add)

l = [2, 4, 6]
```

```
to_add = 9
add_element(lista1=l, to_add=9)
print(l)
```

```
[2, 4, 6, 9]
```

In [32]:

```
def k():
    x = 2
    y = 3
    m = 8

k()
print(m) #Al estar dentro de la funcion no se puede mostrar
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-32-c0ea41bfe09e> in <module>
      5
      6 k()
----> 7 print(m) #Al estar dentro de la funcion no se puede mostrar

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

In [33]:

```
def k():
    x = 2
    y = 3
    m = 8

k()
print(k(m))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-33-c7ef2fc7b065> in <module>
      5
      6 k()
----> 7 print(k(m))

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

In [35]:

```
def k():
    x = 2
    y = 3
    m = 8
    print(m) #Este se muestra porque está dentro de la función


k() #llamamos a la función
print(m)
```

```
8

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-35-853d1afa16fa> in <module>
      7
      8 k() #llamamos a la función
----> 9 print(m)

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

In [37]:

```
def k():
    x = 2
    y = 3
```

```
    m = 8
    return 7

print(k())
```

```
7
```

In [48]:

```
def g(l):
    #Devuelve True si la lista list2 contiene un número mayor a 5
    for x in l:
        if x > 5:
            return True #Return también hace función de break, por tanto si hay dos numer
os mayores que 5, parará en el primero

list2=[2, 4, 5, 9]

z = g(l = list2)
print(z)
```

```
True
```

In [54]:

```
def g(l):
    acum = 0

    for pos, val in enumerate(l):
        if pos == 2: #Para que pare en la tercera iteración. Recordemos que la posición e
s la 2, tercera iteración
            break


list2=[2, 4, 5, 9, 10]
z = g(l = list2)
print(z)
```

```
None
```

In [56]:

```
def p1(l):
    acum = 0

    for pos, val in enumerate(l):
        if pos == len(l) -1: #Se cumple cuando sea la últimma iteracion. Nunca puede ser
menos 1 directamente, ya que pos siempre va increchendo, es decir, pos puede ser 0, 1, 2
etc nunca va a llegar a ser -1, ya que no es lo mismo que acceder al último elemento.
            return 2
        else:
            print("True")


list2=[2, 4, 5, 9, 10] #Si pones list2 después de la definición de z va a dar como error
ya que la lista no está definida con anterioridad
z = p1(l = list2)
print(z)
```

```
True
True
True
True
2
```

In [58]:

```
def df(l):
    acum = 0
    return acum
    for i in l:
        if i > 5:
```

```
            acum = acum + 1


list2=[2, 4, 5, 9, 10]
z = df(l = list2) #Devuelve cero ya que en return para
print(z)
```

```
0
```

In [60]:

```
def df(l):
    acum = 0
    print(acum)
    for i in l:
        if i > 5:
            acum = acum + 1


list2=[2, 4, 5, 9, 10]
z = df(l = list2)
print(z)
```

```
0
None
```

In [83]:

```
#Funcion que retorne cuantos números hay mayores a 5
nombres = ["Ana", "Dorado", ["m", 2], 7]

def dj(lip):
    acumu = 0
    for x in nombres:
        if type(x) == int and x > 5:  #Hay que cambiar el orden, es decir, no se puede p
oner primero x > 5 ya que lo hace primero y da error. Hay que poner primero que busque in
t y luego mayor que 5.
            print(x)
            acumu = acumu + 1
    return acumu

z = dj(lip = nombres)
print(z)
```

```
7
1
```

In [79]:

```
liso = ["Ana", "Dorado", ["m", 2], 7]

def dj(p):
    acumu = 0
    for x in p:
        if isinstance(x, int) and x > 5:  #Lo mismo que antes
            print(x) #Aqui muestra el 7
            acumu = acumu + 1
    return acumu

dj(p = liso) #Aquí muestra el 1
```

```
7
```

Out[79]:

```
1
```

In [ ]:

```

```