

```
In [5]: ## Cuenta regresiva : crea una función que acepte un número como entrada.  
## Devuelve una nueva lista que cuenta hacia atrás en uno, desde el número  
## (como el elemento 0) hasta 0 (como el último elemento).  
## Ejemplo: La cuenta regresiva (5) debería devolver [5,4,3,2,1,0]  
def cuentaRegresiva(a) :  
    lista=[]  
    for x in range (a, -1, -1):  
        lista.append(x)  
    return lista  
  
print(cuentaRegresiva(100))
```

```
[100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86, 85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74,  
73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 4  
6, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19,  
18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

```
In [6]: ## Imprimir y volver : crea una función que recibirá una lista con dos números. Imprima el primer valor y devuel  
## Ejemplo: print_and_return ([1,2]) debería imprimir 1 y devolver 2  
  
def ImprimirVolver(a,b):  
    print(a)  
    return b  
  
print(ImprimirVolver(2,3))
```

```
2  
3
```

```
In [8]: def ImprimirVolver(arr):  
        print(arr[0])  
        return arr[1]  
  
print(ImprimirVolver([2,3]))
```

```
2  
3
```

In [14]: *## First Plus Length : crea una función que acepta una lista y devuelve la suma del primer valor de la lista más de la lista. Ejemplo: first\_plus\_length ([1,2,3,4,5]) debería devolver 6 (primer valor: 1 + longitud: 5)*

```
def firstPlusLength(arr):  
    return arr[0]+len(arr)  
  
arr=[1,2,3,4,5]  
print(firstPlusLength(arr))
```

6

In [29]: *## Valores mayores que el segundo : escribe una función que acepte una lista y crea una nueva lista que contenga de la lista original que sean mayores que su segundo valor. Imprima cuántos valores son y luego devuelva la n la lista tiene menos de 2 elementos, haga que la función devuelva False  
## Ejemplo: values\_greater\_than\_second ([5,2,3,2,1,4]) debería imprimir 3 y devolver [5,3,4]  
## Ejemplo: values\_greater\_than\_second ([3]) debería devolver False*

```
def mayor(arr):  
    if len(arr)<2:  
        return "False"  
    else:  
        arrnew=[]  
        for i in range(len(arr)):  
            if arr[i]>arr[1]:  
                arrnew.append(arr[i])  
        print(len(arrnew))  
        return arrnew  
  
print(mayor([5,2,3,2,1,4]))  
print(mayor([5]))
```

3

[5, 3, 4]

False

```
In [17]: ## Esta longitud, ese valor : escribe una función que acepte dos enteros como parámetros: tamaño y valor. La fun  
## y devolver una lista cuya longitud es igual al tamaño dado y cuyos valores son todos los valores dados.  
## Ejemplo: length_and_value (4,7) debería devolver [7,7,7,7]  
## Ejemplo: length_and_value (6,2) debería devolver [2,2,2,2,2,2]
```

```
def tamañoValor(longitud, valor):  
    lista=[]  
    for i in range(longitud):  
        lista.append(valor)  
    return lista  
  
print(tamañoValor(10, 7))
```

```
[7, 7, 7, 7, 7, 7, 7, 7, 7, 7]
```

```
In [ ]:
```