

Recursividad

Una función recursiva debe tener dos componentes clave: un **caso base** para detener la recursión y un **caso recursivo** donde la función se llama a sí misma con un argumento modificado. Este enfoque es similar a las iteraciones

```
def sum(k):  
    if k == 0 :  
        return 1  
    else:  
        return k + sum(k-1)
```

La función se llama a sí misma n veces hasta que $n = 0$. Las llamadas recursivas `sum(n-1)` se almacenan en memoria esperando a ejecutarse, cuando la última termine. Una vez termina la última, devuelve un valor que las llamadas previas pueden usar para construir un resultado.

Construyendo el resultado (subiendo)

Ahora cada función termina y devuelve su resultado:

```
sum(0) devuelve 0  
sum(1) devuelve 1 + 0  
sum(2) devuelve 2 + (1 + 0)  
sum(3) devuelve 3 + ( 2 + ( 1 + 0 ))
```

