

Funciones y uso de la palabra reservada *return*

Las funciones en computación tienen un comportamiento similar a las funciones matemáticas: reciben **parámetros**, los **procesa** y devuelve un **resultado**.

Los métodos, a diferencia de las funciones, realizan ciertos procesos pero **no regresan ningún resultado**.

En *Python*, y en muchos otros lenguajes, existe la palabra reservada **return** que permite devolver un resultado calculado dentro de una función. Es común que las personas que están aprendiendo a programar interpreten su significado como “salir de la función” lo cual es incorrecto y puede generarles problemas a futuro cuando utilicen lenguajes tipados como Java.

El siguiente es un ejemplo de mal uso de return:

```
def imprimir_serie(inicio_serie, fin_serie):
    if inicio_serie == fin_serie:
        return
    else:
        print(inicio_serie, end=' ')
        imprimir_serie(inicio_serie + 1, fin_serie)
```

Python al ser un lenguaje débilmente tipado no presenta problemas con este código pues en la firma de la función no se ha especificado el tipo de dato a retornar. El siguiente código es la misma lógica utilizando lenguaje *Java*:

```
void imprimirSerie(inicio_serie, fin_serie){
    if (inicio_serie == fin_serie){
        return;
    }
    else {
        System.out.print(inicio_serie + ' ');
        imprimirSerie(inicio_serie + 1, fin_serie);
    }
}
```

El código anterior ni siquiera superaría la fase de compilación pues se está declarando que el método no devolverá ningún tipo de valor (**void**) y luego se contradice lo anterior tratando de retornar algo con la palabra **return**.

Una buena forma de evitar lo anterior es pensar bien la lógica del código: ¿se necesita retornar un valor o simplemente se requiere que se haga algo mientras se cumplan ciertas condiciones?

Una correcta implementación del código anterior sería la siguiente:

```
def imprimir_serie(inicio_serie, fin_serie):
    if inicio_serie != fin_serie: # puede ser inicio_serie < fin_serie
        print(inicio_serie, end=' ')
        imprimir_serie(inicio_serie + 1, fin_serie)
```

Explicación: Se necesita que se imprima el número **inicio_serie** cada vez que este no sea igual a **fin_serie**, si ambos son iguales no haga nada.