

Cosas sueltas 01 - Funciones

En Java no se programa colocando todo el código seguido en un único bloque **main**. Cuando tienes un montón de líneas de código llenas de repetitivas, condicionales, etc. se suele **dividir** el código en métodos o funciones.

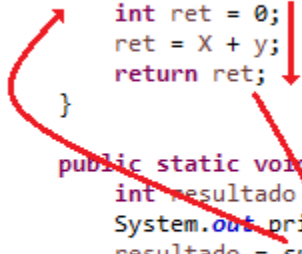
Dado que aún no hemos empezado a trabajar con Clases y Objetos, la explicación que vamos a dar en este documento es limitada. Más adelante profundizaremos en todos estos aspectos.

Un método es en realidad otro bloque de **código separado** al que tú puedes ‘llamar’ o ‘saltar’ escribiendo su nombre. Dado que el bloque siempre está ahí, puedes ‘llamarlo’ tantas veces como quieras desde donde quieras. Por ejemplo:

```
public class EjemploMetodos {  
  
    public static void holaMundo () {  
        System.out.println ("Hola, Mundo!!!");  
    }  
  
    public static void main(String[] args) {  
        holaMundo();  
        holaMundo();  
        holaMundo();  
    }  
}
```

Si ejecutamos este programa, el main ‘llamará’ a **holaMundo()** tres veces seguidas, escribiendo cada vez el texto “Hola, Mundo!!!” por pantalla. Puedes hacer esto tantas veces como creas necesario. No hay límite en cuanto al número de métodos que tiene una clase, aunque tienen que tener todos diferente nombre y/o diferentes variables dentro del (...).

```
public class EjemploMetodos {  
  
    public static int suma (int X, int y) {  
        int ret = 0;  
        ret = X + y;  
        return ret;  
    }  
  
    public static void main(String[] args) {  
        int resultado = 0;  
        System.out.println ("Vamos a sumar...");  
        resultado = suma (10, 20);  
        System.out.println ("Resultado: " + resultado);  
    }  
}
```

A red arrow originates from the line 'resultado = suma (10, 20);' in the main method and points to the 'suma' method definition, illustrating the call to the function.

En el ejemplo de aquí arriba, cuando llamamos a **suma (...)** ocurre lo siguiente:

- 1) El programa 'salta' a la primera línea de la función **suma (...)**, como puedes ver con el debugger.
- 2) Los valores 10 y 20 del (...) se pasan al método. Allí las variables X e Y pasan a valer $X = 10$ e $Y = 20$.
- 3) Se ejecuta el método paso a paso. Se obtiene que $ret = 30$, y al llegar a **return**, se vuelve a la línea desde la que se 'saltó' en el main.
- 4) Se ejecuta la asignación de return, por lo que $return = 30$.

Esto es lo que se llama **pasar parámetros a un método**. Tú puedes pasarle a uno todo lo que necesitas, mientras el tipo de variables que 'le metes' coincidan con los de las variables del método.

También puedes hacer que el método **devuelva un resultado**. Para indicárselo, le indicas el tipo de variable que retorna antes del nombre del método. En este caso, **suma (...)** retorna un entero → **int suma (int x, int y)**

Si vas a devolver un valor, recuerda que tendrás que utilizar **return**. Si no, el return no se usa y el tipo de dato que retorna el método será **void**.

Los métodos o funciones se usan sobre todo para separar el código en bloques lógicos que tengan sentido. En la página siguiente tienes un ejemplo sencillo:

```

public class EjemploMetodos {

    public static int mostraMenu () {
        int ret = 0;
        Scanner teclado = new Scanner(System.in);
        System.out.print ("Elige una opción (1 suma | 2 resta): ");
        ret = teclado.nextInt();
        teclado.close();
        return ret;
    }

    public static int pedirOperando() {
        int ret = 0;
        Scanner teclado = new Scanner(System.in);
        System.out.print ("Operando: ");
        ret = teclado.nextInt();
        teclado.close();
        return ret;
    }

    public static int sumar (int X, int y) {
        int ret = 0;
        ret = X + y;
        return ret;
    }

    public static int restar (int X, int y) {
        int ret = 0;
        ret = X - y;
        return ret;
    }

    public static void main(String[] args) {
        int resultado = 0;
        int operacion = 0;
        int operando1 = 0;
        int operando2 = 0;

        operacion = mostraMenu ();
        operando1 = pedirOperando();
        operando2 = pedirOperando();

        if (operacion == 1) {
            resultado = sumar (operando1, operando2);
        } else {
            resultado = restar (operando1, operando2);
        }

        System.out.println ("Resultado: " + resultado);
    }
}

```