Métodos

En el módulo 2 de este curso estuvimos hablando de las funciones. **Un método es** básicamente **lo mismo que una función pero asociado a un objeto**. Cuando en el siguiente apartado de este módulo hablemos de la palabra reservada static espero aclarar mejor esta diferenciación.

La sintaxis es básicamente la misma que en las funciones.

Vamos a ver un ejemplo:

```
1  /**
2   * @author Pablo Ruiz Soria
3   */
4  public class Coche {
5      String marca;
6      String modelo;
7      Motor motor;
8
9      public Coche(){
10
11     }
12
13     public Coche(String marca, String modelo, Motor motor){
14         this.marca = marca;
15         this.modelo = modelo;
16         this.motor = motor;
17     }
18
19     public void escribirInformacion(){
20         System.out.println("Marca: " + marca + ", modelo: " + this.modelo +
21                 ", motor: " + motor);
22     }
23 }
```

En el ejemplo anterior aparece un método entre las líneas 19 y 22. El método es de acceso público (public), no devuelve nada (void), se llama escribirInformacion y no tiene parámetros. El método tiene una única sentencia que ocupa las líneas 20 y 21 y que muestra por pantalla un texto. Fíjate que puedes acceder a los atributos de la clase con o sin this.

En Java, **dentro de una clase puedes tener 2 métodos que se llamen igual siempre y cuando el número de parámetros que utilicen o el tipo varíe**. A continuación tenemos un ejemplo:

Copyright 2026

```java
/**
 * @author Pablo Ruiz Soria
 */
public class Coche {
    String marca;
    String modelo;
    Motor motor;

    public void escribirInformacion(){
        System.out.println("Marca: " + marca + ", modelo: " + this.modelo +
                ", motor: " + motor);
    }

    public void escribirInformacion(boolean muestraMotor){
        if(muestraMotor){
            escribirInformacion();//puedes anteponer .this
        }else{
            System.out.println("Marca: " + marca + ", "
                    + "modelo: " + this.modelo);
        }
    }
}
```

En el ejemplo anterior vemos que dentro de la misma clase tenemos 2 métodos que se llaman igual (escribirInformacion. Podemos tenerlos porque en la línea 9 definimos uno de ellos sin parámetros y en la línea 14 definimos otro con un solo parámetro de tipo boolean. Podríamos tener otro mas de un solo parámetro siempre y cuando este no fuera de tipo boolean.

En Java existe la [**sobreescritura de métodos**](https://es.wikipedia.org/wiki/Herencia_(inform%C3%A1tica). Esto significa que cuando una clase hereda de otra se puede sobreescribir los métodos de esta. Lo veremos con mas detalle en el apartado de herencia.

Si eres observador/a te habrás dado cuenta de que cuando creo un método no uso la palabra reservada static como hacía con las funciones, en el siguiente capítulo voy a tratar de explicarte el porqué.

Revision #1
Created 2022-02-01 11:11:17 CET by Equipo CATEDU
Updated 2022-02-01 11:11:17 CET by Equipo CATEDU

Copyright 2026