

## TEXT CLASS REVIEW

### TEMAS A TRATAR EN EL CUE:

- Definición de clases.
- Herencia.
- Módulos.
- Generadores.

### CLASES

Las clases de ES6 formalizan el patrón común de JavaScript de simulación de jerarquías de herencia, similares a clases mediante funciones y prototipos. En sí, son plantillas para crear objetos, los cuales nos permiten reflejar las propiedades de la programación orientada a objetos y de la herencia. Éstas poseen características a las que llamaremos “atributos”, y comportamientos a los que llamaremos “métodos”.

### OBJETOS

Unidad dentro de un programa informático. Posee un estado y un comportamiento, es decir, que tiene una serie de datos almacenados, y una cantidad de tareas a ejecutar con esos datos. Éstos se pueden crear instanciando clases.

### INSTANCIAS

Se refieren a una “extensión” de una clase u otro objeto. En otras palabras, instanciar, es crear el objeto concreto a partir de un molde. Por ejemplo: si tenemos la clase “auto”, con las propiedades “marca” y “modelo”, una instancia de dicha clase sería “Suzuki alto”.

### HERENCIA

Es uno de los conceptos centrales de los lenguajes de programación orientada a objetos (OOP). Es un mecanismo en el que se puede derivar una clase de otra clase, en una jerarquía que comparte un conjunto de atributos y métodos.

## POLIMORFISMO

Capacidad para procesar objetos de diferentes maneras, es decir, permite a un método ser distinto según la clase que lo esté usando.

## ENCAPSULAMIENTO

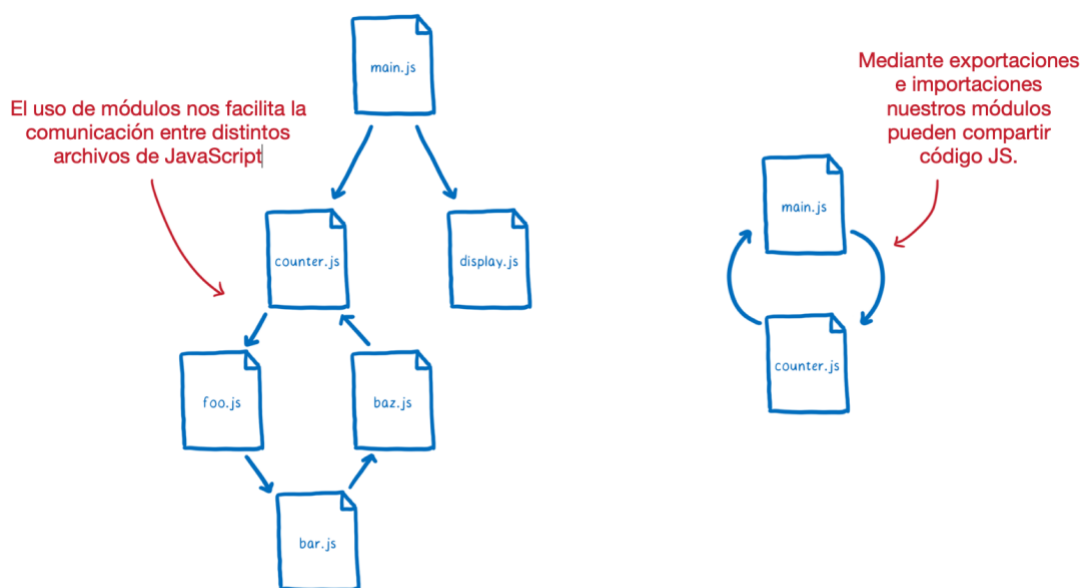
Es la capacidad que tienen los objetos para determinar qué partes puede mostrar a otros objetos. Esto se realiza con los modificadores de acceso (`private`, `protected`, `public`).

## ABSTRACCIÓN

A través de ésta podemos determinar qué atributos y métodos son relevantes para nuestra clase. Entonces, al momento de crear una clase, debemos abstraernos y añadir sólo lo necesario de acuerdo al contexto.

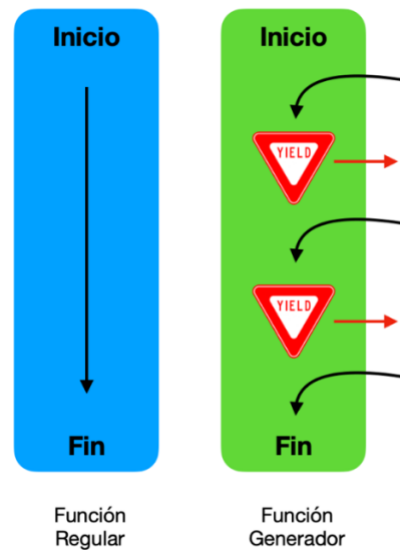
## MÓDULOS

Hasta el momento, hemos trabajado con una cantidad modesta de código JavaScript, pero ¿cómo podríamos organizarlo si empieza a aumentar en tamaño? ES6 nos da la respuesta con módulos. Éstos organizan un conjunto relacionado de código JavaScript, y puede contener variables y funciones. Un módulo no es más que un fragmento de código JavaScript escrito en un archivo. De forma predeterminada, las variables y funciones de un módulo no están disponibles para su uso.



Las variables y funciones dentro de un módulo deben exportarse para que se pueda acceder a ellas desde otros archivos, para esto, se exportan en ES6, pues solo funcionan en modo estricto; por lo que aquellas que sean declaradas en un módulo no serán accesibles globalmente.

## GENERADORES



ES6 introdujo una nueva forma de trabajar con funciones e iteradores en forma de **generadores** (o funciones de generador). Un generador es una función que puede detenerse a mitad de camino, y luego continuar desde donde se detuvo. Como podemos apreciar en el diagrama, una función de generador contiene señales de “ceda al paso”, o **yields** en inglés. Dentro del sintaxis, utilizamos la palabra clave **yield** para detener la ejecución de la función y reanudarla. En resumen, un generador parece ser una función, pero se comporta como un iterador.