

Programación

Decimotercera semana - Clases

Enero 2022

Cruz García, Iago



[Anotaciones previas](#)

[Ejercicios](#)

[Ejercicio 0](#)

[Ejercicio 1](#)

[Ejercicio 2](#)

[Ejercicio 3](#)

[Ejercicio 4](#)

[Ejercicio 5](#)

[Ejercicio 6](#)

[Ejercicio 7](#)

[Ejercicio 8](#)

[Ejercicio 9](#)

[Ejercicio 10](#)

[Ejercicio 11](#)

[Ejercicio 12](#)

[Ejercicio 13](#)

[Ejercicio 14](#)

[Ejercicio extra](#)

Anotaciones previas

Estos ejercicios son para familiarizarse con el lenguaje, la sintaxis y cómo resolverlos. Los primeros son sencillos y se va incrementando la dificultad. A continuación se presentan una serie de instrucciones que son necesarias para la resolución de los ejercicios:

- [alert\(parámetro\)](#): esta instrucción permite mostrar por pantalla un cartel con texto para mostrar la solución de algunos ejercicios.
- [console.log\(parámetro\)](#): esta instrucción permite mostrar en consola (F12 en el navegador) la solución de algunos ejercicios o trazar el código para comprobar que todo se ejecuta correctamente.
- [prompt\(texto, ejemplo\)](#): Muestra en pantalla un recuadro de **texto** y un cuadro para introducir texto con un **ejemplo**.
- Para poder ejecutar código JavaScript en Visual Studio Code debéis crear un fichero JavaScript (miScript.js) y un HTML básico (index.html por ejemplo) y dentro de la etiqueta <head> escribir los siguiente:
 - <script src="miScript.js"></script> comillas incluidas
- Ahora que sabemos encapsular creando funciones o métodos, se pueden hacer los ejercicios en el mismo fichero, simplemente comentando las llamadas a métodos que no necesiteis.

```
ejercicio_1()  
//ejercicio_2()  
//ejercicio_3()
```

Ejercicios

IMPORTANTE: A partir de ahora algunos ejercicios deben hacerse en múltiples ficheros .js, por lo que en vez de entregar todos en un mismo main.js, será necesario dividirlos en directorios. Se aconseja la estructura de “PrácticaX_ejercicio1” y dentro el index.html y los ficheros .js necesarios.

Para estos ficheros, lo mejor es agrupar aquellas funciones o métodos que realicen tareas similares (entradas.js o salidas.js por ejemplo). En caso de duda, no importa que un método quede aislado en un fichero.

El fichero que realice las llamadas a los métodos, que aune toda la funcionalidad, debe nombrarse como main.js y no debe tener más que un método que se llame igual y una llamada a este mismo.

Ejercicio 0

Vamos a hacer una serie de ejercicios para afianzar los conceptos de acceso y control de JQuery. Vamos a utilizar el ejercicio del juego que realizamos el último tema, el de clases.

Añadimos en el 'index.html' los siguientes elementos. No hace falta que coincidan en el nombre, pero estos son los que se usarán para los ejercicios:

```
<head>
  <script src="https://code.jquery.com/jquery-3.5.0.js"></script>
</head>
<body>
  <div class="heroe">
    <h1>Heroe</h1>
    <h3>Nivel:
    <lvl id="nivel">-1</lvl>
    </h3>
    <h3>Experiencia:
    <exp id="experiencia">-1</exp>
    </h3>
    <h3>Salud:
    <hp id="salud">-1</hp>
    </h3>
  </div>
  <div class="enemigo">
    <h1>Enemigo</h1>
    <h3>Nivel:
    <lvl id="nivel">-1</lvl>
    </h3>
    <h3>Salud:
    <hp id="salud">-1</hp>
    </h3>
  </div>
```

```
</body>
<footer>
  <script src="main.js"></script>
</footer>
```

En los ejercicios, cuando se pide modificar los valores, debemos utilizar JQuery.

Ejercicio 1

Modifica el valor de nivel del héroe en el HTML para que muestre el nivel actual del héroe. Debe cambiar cuando empiece la partida y cuando este suba de nivel.

Ejercicio 2

Modifica el valor de la experiencia del héroe en el HTML para que muestre la experiencia que tiene en todo momento.

Ejercicio 3

Modifica el valor de salud para que muestre la salud actual y la salud total tal que "Salud: 10/10". Debe cambiarse cada vez que se reduzca o aumente.

Ejercicio 4

Modifica los valores del enemigo para que funcionen como los del héroe. La vida debe modificarse para mostrar el valor actual y el total y el nivel debe cambiar dependiendo del enemigo.

Ejercicio 5

Modifica los atributos del héroe tal que:

- Tamaño de fuente: 20 pts
- Color de fuente: azul
- Centrar en el medio de la página

Pista: `$.style(...)` para modificar la hoja de estilos en JQuery

Ejercicio 6

Modifica los atributos del enemigo tal que:

- Tamaño de fuente: 16 pts
- Color de fuente: rojo
- Añadir un borde negro

Ejercicio 7

Modifica el layout de la página para que héroe y enemigo estén uno a la izquierda y el otro a la derecha.

Ejercicio extra

JQuery permite el uso de animaciones también. Os animo a buscar diferentes ejemplos online y utilizarlos. Algunas podrían ser que el texto tiemble cuando cambia o fade out del personaje que pierda.