

Programación

Vigésimo Tercera semana

Abril 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](#)

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

Esta vez vamos a trabajar con tres ficheros, *index.html*, *main.js* y *main.php*. Estos serán el esqueleto de nuestros ejercicios.

Si el enunciado no lo especifica, se podrá resolver con cualquier método, pero, por ejemplo, si se exige el uso de una conexión de JavaScript a PHP no se podrá utilizar un formulario en HTML directamente.

Ejercicio 1

Crear un formulario en HTML con los siguientes campos y restricciones en JavaScript:

- Nombre, solo puede contener caracteres alfabéticos y como mucho pueden ser 25 caracteres.
- Apellidos, solo puede contener caracteres alfabéticos y como mucho pueden ser 50 caracteres.
- Edad, sólo puede contener caracteres numéricos y como mucho ser un valor entre 10 y 100.
- Nota, sólo puede contener caracteres numéricos y como mucho ser un valor entre 0 y 10.

Ejercicio 2

Crear en PHP una clase Alumno con los atributos del ejercicio anterior.

Ejercicio 3

Utilizando HTML, añade el botón de de enviar, mediante POST, la información de los campos al documento PHP y que se muestre en PHP los valores recibidos. El botón de envíar solo debe de estar disponible si los valores son correctos.

Ejercicio 4

Vamos a reformar el botón de envío. En vez de ser un botón tipo "submit", será un botón cualquiera. Crea un evento en JavaScript enlazado a ese botón que muestre por pantalla "se pulsó el botón de envío con la información:" seguido de la información de los campos.

Ejercicio 5

Añade en el PHP la creación de un objeto del tipo Alumno con los valores que se introducen en el formulario HTML. Añade este objeto a un array.

Ejercicio 6

El HTML envía información a PHP, pero recarga la página y muestra el documento PHP y eso en una aplicación de usuario no tiene mucho sentido.

Con la estructura más o menos terminada, haz que el método de JavaScript del evento del botón, envíe la información al documento PHP. PHP debe retornar el valor OK si se pudo crear el objeto y el valor ERROR si no.

Ejercicio 7

Añade un botón de “Recuperar alumno”. En JavaScript, haz el evento correspondiente, que deberá pedir al documento PHP que devuelva el objeto Alumno que corresponda con el nombre. Muestra el resultado por pantalla.

Tenemos una especie de lista de alumnos preparada.

Ejercicio extra

Esta forma de guardar información es bastante estándar, pues si tenemos en cuenta que el documento PHP se mantiene en servidor y lo que el cliente ve es el HTML y JavaScript, realmente no puede acceder a la información que nosotros no programemos. Para reforzar la seguridad y añadir nuevas funcionalidades, os propongo los siguientes ejercicios:

- Corroborar en PHP que el formato de los valores enviados es correcto, además de en JavaScript. Esto es por si el usuario crea un documento JavaScript e inyecta el código en la página, podría modificar el nuestro, invalidando el trabajo.
- Crea un apartado en HTML que permita visualizar el alumno que se busca.

- Añade, de alguna manera, un valor a la clase Alumno del tipo número de expediente o algún valor identitario, ya que el nombre no debe de ser lo identificativo de un alumno.
- Cuando se busca por el nombre, mostrar por pantalla o HTML los múltiples valores que podría encontrar. Este apartado es un poco más complejo que los anteriores. Como pista, se puede crear una clase "alumnos" que como atributo tenga un array de "alumno" y sea el valor que devuelve PHP, aunque hay otras formas de hacerlo.