Desarrollo Web en Entorno Cliente.

Unidad 4.

JavaScript
Ejercicios.

- Funciones Anónimas y Autoejecutables.

Ejercicio 1.

Escribir el código de una función a la que se pasa como parámetro un número entero y devuelve como resultado una cadena de texto que indica si el número es par o impar. Mostrar por pantalla el resultado devuelto por la función.

Ejercicio 2.

Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.

Ejercicio 3.

Queremos crear una función llamada holaMundo1 de tal forma que muestre por consola el texto “Hola Mundo 1”, ¿Cómo harías la llamada para ejecutarla?.
Queremos crear una variable llamada holaMundo2 de tal forma que tenga asignada una función anónima y que muestre por consola “Hola Mundo 2”. ¿Cómo harías la llamada para ejecutarla?.

Ejercicio 4.

Entendiendo que una función es un objeto en JavaScript, y que se puede pasar como parámetro.
Crea una función llamada “saluda” que muestre “Hola” en un alert y que existe una función llamada ejecuta que recibirá como argumento la función anterior. ¿Cómo harías para ejecutar ésta última?

Ejercicio 5.

Crear una función llamada “saludador” de tal forma que reciba como argumento un nombre y que devuelva una función anónima de tal forma que el contenido de esta función anónima es Hola y el nombre pasado como argumento. ¿Cómo ejecutarías la función saludador, si ésta es asignada a una variable llamada saluda?

Ejercicio 6.

Crea una función autoejecutable y anónima de tal forma que reciba como argumento un nombre y el valor establecido para este argumento es “mundo”. Dicha función autoejecutable pintará por consola “hola” y el nombre pasado como argumento.

Ejercicio 7.

Un closure combina una función y el entorno (contexto - scope) en que se creó.
Ejecuta el siguiente código e interpreta qué es lo que ocurre.

```
function creaSumador(x) {
  return function(y) {
    return x + y;
  };
}

var suma5 = creaSumador(5);
var suma10 = creaSumador(10);

console.log(suma5(2));  // muestra 7
console.log(suma10(2)); // muestra 12
```

Realiza un ejemplo propio utilizando el concepto de closure.

Ejercicio 8.

El concepto de arguments → Es un objeto que nos indica los argumentos pasados a una función.
- Crea una función llamada “nombres” de tal forma que nos muestre los nombres pasados a la función como argumentos. Pero no utilices el nombre del parámetro sino el concepto arguments, que no es estrictamente un array, sino un objeto, pero sí que tiene la propiedad length de los argumentos pasados a la función. (function.length)
– Crea otra función pero en este caso que sea anónima y autoejecutable de tal forma que nos muestre los argumentos pasados en la autoejecución.

Ejercicio 9.

– El objeto arguments podría convertirse en un array real, comprueba primero que arguments, no acepta el método pop (que es un método para arrays) y posteriormente comprueba que se puede convertir en un array real, utilizando el siguiente código según:

```
// ES2015
const arrayArgumentos = Array.from(arguments);
```

Comprueba que como ya, sí que es un array podemos añadir y eliminar elementos.