**TEMA 4: Programación con funciones, arrays y objetos**

**1. Funciones**

**1.1. Funciones predefinidas por el lenguaje**

JavaScript cuenta con funciones predefinidas que permiten realizar una serie de acciones.

**1.2. encodeURI/encodeURIComponent**

Codifican una URI en formato ISO, para que pueda ser entendida por cualquier navegador.

**1.3. eval**

Evalúa un código JavaScript representado por una cadena.

**1.4. isNaN()**

Evalúa si un argumento es un número.

**1.5. Crear y vincular hojas de estilos**

Pese a no ser un lenguaje tipado, en ocasiones es necesario convertir variables en un tipo concreto:

* String(): Convierte un objeto en cadena
* parseInt(): Convierte un objeto en número
* parseFloat(): Convierte un objeto en número de punto flotante.

**1.6 Funciones de usuario**

Como en cualquier lenguaje, JavaScript permite crear funciones personalizadas.

Una función está compuesta de:

* La palabra reservada **function**
* Nombre
* Parámetros
* Instrucciones
* La palabra reservada **return** (opcional)

**2. Arrays**

Los arrays permiten almacenar, recorrer y editar un conjunto de datos.

**2.1 Manejando arrays**

Antes de poder usar un array es necesario declararlo.

Se puede visualizar el contenido de un array recorriéndolo con un for.

**2.2 Propiedades y funciones de arrays**

Propiedades:

* length: Devuelve el tamaño del array.

Métodos:

* push: Permite añadir nuevos elementos al array
* sort: Ordena los elementos del array

**3. Objetos**

En JavaScript se pueden definir los objetos de más de una forma.

**3.1 Definiendo objetos**

Una forma habitual es crear una función y añadirle propiedades y métodos mediante this. El objeto se instancia usando new.

Otra forma es creando una variable con sus propiedades y métodos definidos entre llaves.

**3.2 Acceso y propiedades**

Se puede acceder a las propiedades de un objeto mediante la notación de punto o la de corchete.

**3.3 Funciones**

Se pueden crear métodos específicos del objeto a través de funciones.

**3.4 Prototipos y Herencia**

En JavaScript la herencia de propiedades y métodos se realiza a través de prototipos.

Object.create() crea un prototipo del objeto existente. El nuevo objeto es una instancia y mantiene un vínculo con su prototipo. Si el prototipo cambia, las instancias también.