Introducción a JavaScript.

Por si mismo, HTML no tiene ninguna habilidad:

- No puede realizar operaciones matemáticas.
- No puede ver si alguien ha rellenado correctamente un formulario.
- No puede tomar decisiones acerca de cómo interactúa con el un visitante Web.
- En realidad, HTML5 permite validaciones básicas de formularios, pero debido a que no todos los navegadores las admiten, se necesitara JavaScript.

Para añadir inteligencia a la Web se necesita JavaScript.

- Se pueden crear formularios que permitan a los visitantes saber cuando han olvidado incluir información necesaria.
- Se puede hacer que los elementos aparezcan, desaparezcan o se muevan en torno a la Web.
- Se pueden actualizar los contenidos de la Web con información extraida de un Servidor Web, sin necesidad de cargar una nueva página Web.

JavaScript es un lenguaje del lado del cliente, es decir, trabaja dentro de un Navegador Web.

El Lenguaje de programación Web alternativo se llama del lado del Servidor y son paginas creadas con PHP, .NET, ASP, Coldfussion, Ruby on Rails, estos lenguajes se ejecutan en un Servidor Web.

El problema de estos lenguajes servidores, es que requieren que el Navegador Web envié peticiones al Servidor Web, forzando al visitante a esperar hasta que llegue una nueva página de información.

Los lenguajes del lado del cliente, pueden reaccionar inmediatamente y cambiar lo que ve un visitante en su navegador Web sin necesidad de cargar una nueva página.

Así, el contenido puede aparecer y desaparecer, moverse por la pantalla o automáticamente actualizarse, basándose en como interactúa el visitante con la página.

Pero, JavaScript no es la única tecnología del lado del cliente que hay en el mercado.

También se pueden usar complementos para añadir programación a una página Web:

- Los applets de Java (programas en Java que funcionan en un Navegador Web).
- Flash, tecnología basada en complementos. (Ej. Mapas de Google)

Ajax, combina el lado del cliente con el del servidor.

Ajax es un método para usar JavaScript de modo que el cliente le hable al servidor.

Recupera información del servidor y actualiza la página Web sin necesidad de cargar una nueva página.

En realidad JavaScript también puede ser un lenguaje de programación del lado del servidor.

Así , el servidor Web **node.js** usa JavaScript como lenguaje del lado del servidor para conectarse a una base de datos, acceder al sistema de archivos del servidor y realizar distintas tareas.

JavaScript no es un lenguaje compilado sino Interpretado, (por otro programa que puede convertir las secuencias de comandos en algo que el ordenador entenderá) en este caso por el Navegador Web.

Los Navegadores Web están construidos para entender HTML y CSS.

La parte del Navegador Web que entiende HTML y CSS se llama: Motor de Diseño o Renderizacion.

La parte del Navegador Web que entiende JavaScript se llama: Interprete de JavaScript.

Como el Navegador normalmente espera HTML tenemos que decirle cuando viene la programación JavaScript, para ello usamos el código estándar de HTML, la etiqueta <script> , y así le decimos al navegador que use el intérprete de JavaScript.

En HTML 4.O1 necesitamos poner el tipo de comandos que sigue a <script>, sino lo ponemos, la pagina funciona bien, pero no se valida correctamente.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
        <title>Mi Web</title>
        <script type="text/javascript">

        </script>
</head>
<body>

</body>
```

En HTML 5 no hace falta.

```
<!doctype html>
<html>
<head>
        <meta charset="UTF-8">
        <title>Mi Web</title>
        <script>
                alert('hello world');
        </script>
</head>
<body>
        <script src="barra_navegacion.js"></script>     → No añadimos código entre las etiquetas
        <script>
              alert('Bieeeeen!');      → Añadimos otras etiquetas <script> para añadir codigo
        </script>
</body>
```

En la mayoría de los casos, se pondrán las etiquetas <script> en el encabezado <head> para mantener el código bien organizado.

Sin embargo, es perfectamente válido poner etiquetas <script> en cualquier sitio dentro del HTML de la página.

Incluso la podemos poner debajo de </body>, de esta forma se ejecutara la página antes que el JavaScript.

Cuando queremos crear código JavaScript que reutilizaremos en diferentes páginas del sitio Web, para ello usaremos archivos JavaScript externos. (Asi no tendremos que copiar y pegar el mismo código en todas las páginas).

Ej: Usar la misma barra de navegación en diferentes páginas del mismo sitio.

Para vincular un archivo externo de JavaScript a una pagina Web se necesita especificar la URL para el atributo src.

URL = Localizador Uniforme de Recursos (Uniform Resource Locator).

Hay tres tipos de rutas, para indicar donde el Navegador Web encontrara el recurso:

- **Ruta Absoluta**: **http://www.nombre_del_host.com/scripts/ejemplo.js** → La carpeta y archivo
  Se usa cuando el archivo no esta en el mismo servidor que la pagina web que construimos.
- **Ruta Relativa en la raiz:** **/scripts/ejemplo.js** → No hay http:// , No hay Nombre del Dominio
  Se usa para archivos JavaScript almacenados en el sitio web que se esta usando.
  Pero las rutas relativas a la raíz no funcionan a menos que se estén viendo las paginas web a través de un servidor Web (ya sea un Servidor Web externo en Internet o un servidor Web instalado en el ordenador del usuario con propósitos de tener un elemento para comprobar.
- **Ruta Relativa subiendo un nivel:** ../scripts/ejemplo.js
  Se usa cuando estamos diseñando en el propio ordenador sin necesidad de un servidor Web.

Se pueden Adjuntar varios archivos de JavaScript a una sola pagina Web.

Ejemplo: Un archivo ej1.js, y otro ej2.js

```
<!doctype html>
<html>
<head>
        <meta charset="UTF-8">
        <title>Mi Web</title>
        <script>
                alert('hello world');
        </script>
</head>
<body>
        <script src="ej1.js"></script>    → Dos archivos vinculados
        <script src="ej2.js"></script>
        <script>
              alert('Bieeeeen!');
        </script>
</body>
```

Muchos desarrolladores Web crean un directorio especial para archivos externos de JavaScript en la carpeta raíz del sitio y los nombres más habituales son: **js** , **libs**

Ejemplo:

Es una página web a la que se le ha añadido un efecto de cascada externo para añadir un pequeño recurso visual.

```
<!DOCTYPE html>
<html>
<head>
	<meta charset="UTF-8">
	<title>Mi titulo</title>
	<link href="../_css/site.css" rel="stylesheet">
	<script>
		alert('Hola Mundo');
	</script>
</head>
<body>
<div class="wrapper">
	<header>
		JAVASCRIPT <span class="amp">&amp;amp;</span> AJAX MANUAL
	</header>
	<div class="content">
		<script>
			document.write('<p>Hola mundo!!</p>');  → Permite escribir directamente en la WEB
		</script>
		<div class="main">
			<h1>Hola Mundo</h1>
			<p>Estamos empezando. </p>
		</div>
	</div>
	<footer>
		<p>JavaScript <a href="http://www.iescamas.com/">Juan Antonio</a></p>
	</footer>
</div>
</body>
</html>
```