

HTML5:

**HTML5** (*HyperText Markup Language*, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como *HTML5* y una variante XHTML conocida como sintaxis *XHTML5* que deberá ser servida como XML. Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5. El desarrollo de este lenguaje de mercado es regulado por el Consorcio W3C.

HTML5 establece una serie de nuevas etiquetas necesarias para el desarrollo de nuestra aplicación. Algunas de las bibliotecas que se usarán tales como hightcharts.js entre otras hacen uso de elementos canvas necesarias para la implementación de las gráficas sobre el árbol DOM. Por otro lado HMTL5 ofrece una serie de servicios y aplicaciones de uso web extras tales como la inclusión de audio, video y mejoras contundentes en los formularios, ampliando el abanico de posibilidades en cuanto a posibles modificaciones de la aplicación en esta memoria se presenta.

El uso de nuestra aplicación en un futuro, así como su situación de código abierto, hacen a este lenguaje de formación de texto web perfecto para dar más vida al programa aquí presentado.

CSS/CSS3:

**Hoja de estilo en cascada** o **CSS** (siglas en inglés de *cascading style sheets*) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo «style».

El avance de la tecnología ha llevado a CSS hasta su desarrollo en CSS3, dónde se establecen nuevas posibilidades, como la creación de animaciones, entornos 3d y mucho más. A continuación paso a listar su progreso a lo largo de sus diversas versiones.

**CSS1**

La primera especificación oficial de CSS, recomendada por la W3C fue CSS1, publicada en diciembre 1996,[3](http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada#cite_note-CSS1-3) y abandonada en abril de 2008.[3](http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada#cite_note-CSS1-3)

Algunas de las funcionalidades que ofrece son:

* Propiedades de las fuentes, como tipo, tamaño, énfasis...
* Color de texto, fondos, bordes u otros elementos.
* Atributos del texto, como espaciado entre palabras, letras, líneas, etcétera.
* Alineación de textos, imágenes, tablas u otros.
* Propiedades de caja, como margen, borde, relleno o espaciado.
* Propiedades de identificación y presentación de listas.

**CSS2**

La especificación CSS2 fue desarrollada por la W3C y publicada como recomendación en mayo de 1998, y abandonada en abril de 2008.[4](http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada#cite_note-4)

Como ampliación de CSS1, se ofrecieron, entre otras:

* Las funcionalidades propias de las capas (<div>) como de posicionamiento relativo/absoluto/fijo, niveles (z-index), etcétera.
* El concepto de "*media types*",
* Soporte para las hojas de estilo auditivas
* Texto bidireccional, sombras, etcétera.

**CSS 2.1**

La primera revisión de CSS2, usualmente conocida como "CSS 2.1", corrige algunos errores encontrados en CSS2, elimina funcionalidades poco soportadas o inoperables en los navegadores y añade alguna nueva especificación.

De acuerdo al sistema de estandarización técnica de las especificaciones, CSS2.1 tuvo el estatus de "candidato" (*candidate recommendation*) durante varios años, pero la propuesta fue rechazada en junio de 2005; en junio de 2007 fue propuesta una nueva versión candidata, y ésta actualizada en 2009, pero en diciembre de 2010 fue nuevamente rechazada.

En abril de 2011, CSS 2.1 volvió a ser propuesta como candidata , y después de ser revisada por el *W3C Advisory Committee*, fue finalmente publicada como recomendación oficial el 7 de junio de 2011.

**CSS3**

A diferencia de CSS2, que fue una gran especificación que definía varias funcionalidades, CSS3 está dividida en varios documentos separados, llamados "módulos".

Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad.

Los trabajos en el CSS3, comenzaron a la vez que se publicó la recomendación oficial de CSS2, y los primeros borradores de CSS3 fueron liberados en junio de 1998.

Debido a la modularización del CSS3, diferentes módulos pueden encontrarse en diferentes estados de su desarrollo de forma que a fechas de noviembre de 2011, hay alrededor de cincuenta módulos publicados, tres de ellos se convirtieron en recomendaciones oficiales de la W3C en 2011: *"Selectores"*, *"Espacios de nombres"* y *"Color"*.

Algunos módulos, como *"Fondos y colores"*, *"Consultas de medios"* o *"Diseños multicolumna"* están en fase de "candidatos", y considerados como razonablemente estables, a finales de 2011, y sus implementaciones en los diferentes navegadores son señaladas con los prefijos del motor del mismo.

A continuación podemos ver un ejemplo de lo que ofrece una de las versiones de css, en este caso css3d:

<https://www.youtube.com/watch?v=nPEYdw2Ssa8>

Javascript:

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,3 basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas4 aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.

**Características:**

Imperativo y estructurado

JavaScript es compatible con gran parte de la estructura de programación de C (por ejemplo, sentencias if, bucles for, sentencias switch, etc.). Con una salvedad, en parte: en C, el ámbito de las variables alcanza al bloque en el cual fueron definidas; sin embargo JavaScript no es compatible con esto, puesto que el ámbito de las variables es el de la función en la cual fueron declaradas. Esto cambia con la versión de JavaScript 1.7, ya que añade compatibilidad con block scoping por medio de la palabra clave let. Como en C, JavaScript hace distinción entre expresiones y sentencias. Una diferencia sintáctica con respecto a C es la inserción automática de punto y coma, es decir, en JavaScript los puntos y coma que finalizan una sentencia pueden ser omitidos.

Dinámico

Como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable. Por ejemplo, una variable x en un momento dado puede estar ligada a un número y más adelante, religada a una cadena. JavaScript es compatible con varias formas de comprobar el tipo de un objeto, incluyendo duck typing.18 Una forma de saberlo es por medio de la palabra clave typeof.

Objetual

JavaScript está formado casi en su totalidad por objetos. Los objetos en JavaScript son arrays asociativos, mejorados con la inclusión de prototipos (ver más adelante). Los nombres de las propiedades de los objetos son claves de tipo cadena: obj.x = 10 y obj['x'] = 10 son equivalentes, siendo la notación con punto azúcar sintáctico. Las propiedades y sus valores pueden ser creados, cambiados o eliminados en tiempo de ejecución. La mayoría de propiedades de un objeto (y aquellas que son incluidas por la cadena de la herencia prototípica) pueden ser enumeradas a por medio de la instrucción de bucle for... in. JavaScript tiene un pequeño número de objetos predefinidos como son Function y Date.

Evaluación en tiempo de ejecución

JavaScript incluye la función eval que permite evaluar expresiones como expresadas como cadenas en tiempo de ejecución. Por ello se recomienda que eval sea utilizado con precaución y que se opte por utilizar la función JSON.parse() en la medida de lo posible, pues resulta mucho más segura.

Funcional

Funciones de primera clase

A las funciones se les suele llamar ciudadanos de primera clase; son objetos en sí mismos. Como tal, poseen propiedades y métodos, como .call() y .bind().19 Una función anidada es una función definida dentro de otra. Esta es creada cada vez que la función externa es invocada. Además, cada función creada forma una clausura; es el resultado de evaluar un ámbito conteniendo en una o más variables dependientes de otro ámbito externo, incluyendo constantes, variables locales y argumentos de la función externa llamante. El resultado de la evaluación de dicha clausura forma parte del estado interno de cada objeto función, incluso después de que la función exterior concluya su evaluación.20

Prototípico

Prototipos

JavaScript usa prototipos en vez de clases para el uso de herencia.21 Es posible llegar a emular muchas de las características que proporcionan las clases en lenguajes orientados a objetos tradicionales por medio de prototipos en JavaScript.22

Funciones como constructores de objetos

Las funciones también se comportan como constructores. Prefijar una llamada a la función con la palabra clave new crear una nueva instancia de un prototipo, que heredan propiedades y métodos del constructor (incluidas las propiedades del prototipo de Object).23 ECMAScript 5 ofrece el método Object.create, permitiendo la creación explícita de una instancia sin tener que heredar automáticamente del prototipo de Object (en entornos antiguos puede aparecer el prototipo del objeto creado como null).24 La propiedad prototype del constructor determina el objeto usado para el prototipo interno de los nuevos objetos creados. Se pueden añadir nuevos métodos modificando el prototipo del objeto usado como constructor. Constructores predefinidos en JavaScript, como Array u Object, también tienen prototipos que pueden ser modificados. Aunque esto sea posible se considera una mala práctica modificar el prototipo de Object ya que la mayoría de los objetos en Javascript heredan los métodos y propiedades del objeto prototype, objetos los cuales pueden esperar que estos no hayan sido modificados.25



Bootstrap:

Es una colección de herramientas para creación de sitios web y aplicaciones web. Contiene HTML y CSS diseñando plantillas con tipografía propia, botones, formatos, barras de navegación y otros componentes de interfaz. En Junio de 2014 supuso el primer proyecto en cabeza en GitHub con más de 73000 stars y más de 27000 forks. Su uso se extiende a lugares como NASA o MSNBC.

De este modo constituye una de las herramientas para la creación de entornos web más utilizadas, y con ello amplía la comunidad que se ha interesado por ella. Gracias a este interés y avance comunitario podemos encontrar por la red multitud de plantillas gratuitas que nos pueden facilitar el entorno de nuestra aplicación, así como otras muchas de pago de las que podemos hacer uso del mismo modo.

Meteor?: