

Índice

Junes, 19 de febrero de 2018 11:38

Introducción al HTML y el CSS

1. Lenguajes de marcado
 - a. Inicio de los lenguajes de Marcado.
 - b. HTML. Origen y evolución. HTML5
 - c. XML
2. Entorno de trabajo. Metalenguajes
 - a. Navegadores y servidores Web
 - b. Editores de texto. VSC
 - c. Metalenguajes:
 - i. Emmet.
 - ii. Preprocesadores de CSS (mencionar)
3. Documentos HTML
 - a. Elementos del lenguaje
 - b. Estructura del documento.
 - c. Head
 - d. Concepto de validación
 - e. Carácter descriptivo. Tipos de elementos
4. Formato. Estilos CSS.
 - a. Elementos básicos.
 - b. DOM. Selectores. Herencia
 - c. Cascada de estilos. Precedencia. Especificidad
5. Elementos de diseño avanzados
 - a. Selectores: pseudoclases y pseudoelementos
 - b. Nuevos selectores en CSS3
 - c. Variables en CSS3. Funciones: calc()
 - d. Modelo de caja. Concepto. Modelos disponibles
 - e. Inicialización. Reseteo y normalización..
 - f. Soporte y prefijos



Lenguajes de marcado

lunes, 19 de febrero de 2018 11:37

1. Inicio de los lenguajes de Marcado.
2. HTML. Origen y evolución. HTML5
3. XML



Inicio de los lenguajes de Marcado

lunes, 19 de febrero de 2018 11:15

Standard Generalized Markup Language - SGML (Lenguaje de Marcado Estándar Generalizado).

1974 — IBM define GLM, un sistema para la organización y etiquetado de documentos

1986 — La Organización Internacional de Estándares (ISO) normalizó este lenguaje

El lenguaje SGML sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.

ISO 8879:1986, *Information processing — Text and office systems - Standard Generalized Markup Language (SGML)*.

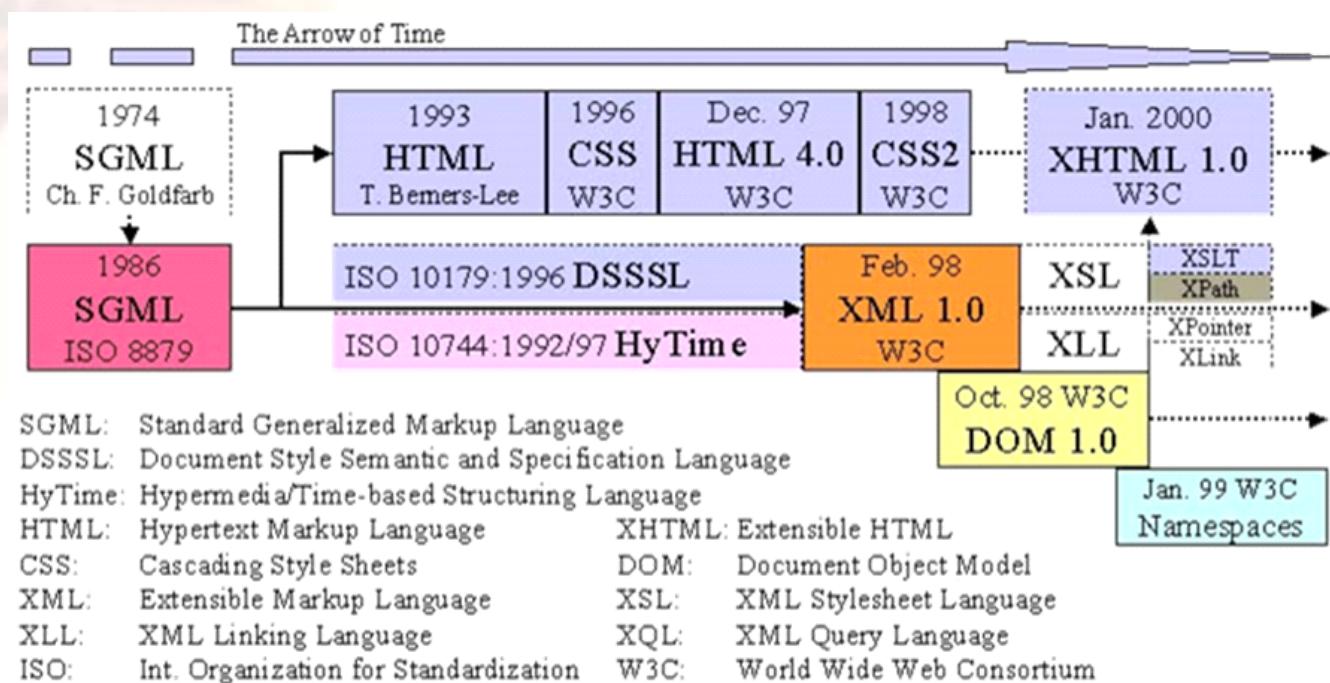
Document: Bungler OED At: "<entry>"

```
<entry>
  <hwsec>
    <hwsp>
      <hwlem>bungler</hwlem>
      <pron>b<I>ʌŋglər</I></pron>. </hwsp>
      <vfl>Also <vd>b</vd> <vf>bongler</vf>,
            </vfl>
      <etym>f. as prec. + <xra><xlem>-ER</xlem></xra>
    <sen>One who bungles; a clumsy unskillful person.
    <quot>
      <qdat>1533 </qdat>
      <auth>MORE </auth>
      <wk>Answe. Poyson. 3k. </wk>Ms. (1557)
      <qtxt>He is even but a very bungler.
            </qtxt>
```

Evolución

lunes, 19 de febrero de 2018 11:49

Evolución de los lenguajes de marcado

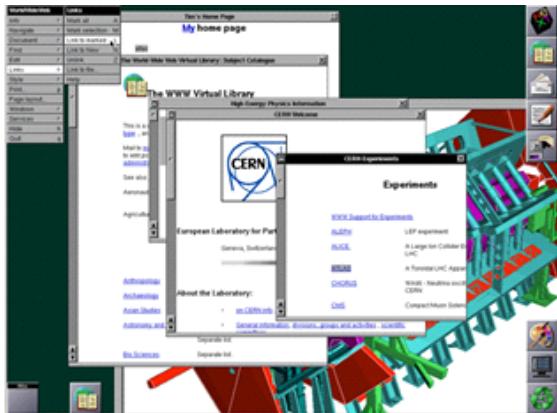




Inicios del HTML: la Web

1990 (octubre): Tim Berners-Lee desarrolla todos los elementos de un sistema de hipertexto.

- Un editor y navegador gráfico
- Un servidor de documentos
- Un protocolo de comunicaciones: HTTP
- Un sistema de identificación: URL
- Un lenguaje: HTML

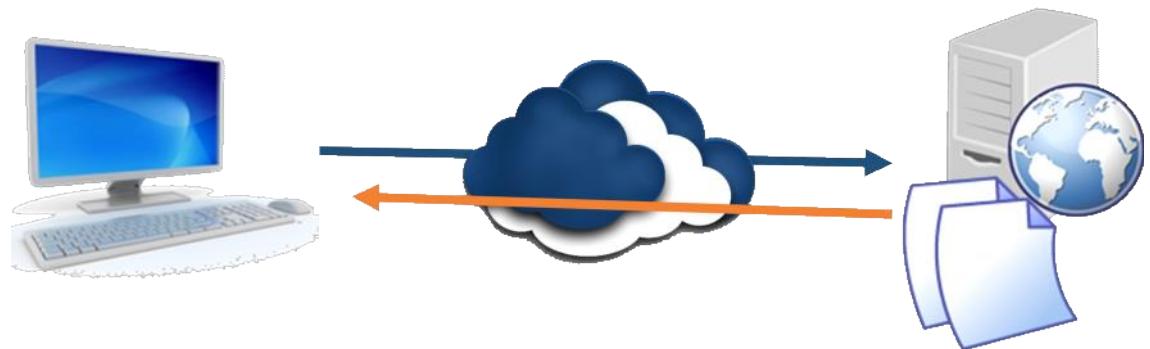


1990 (noviembre): se instala el primer servidor web en un *NeXT*, en el que se publica la primera página web. TBL accede a ella utilizando su primer navegador

Detalles

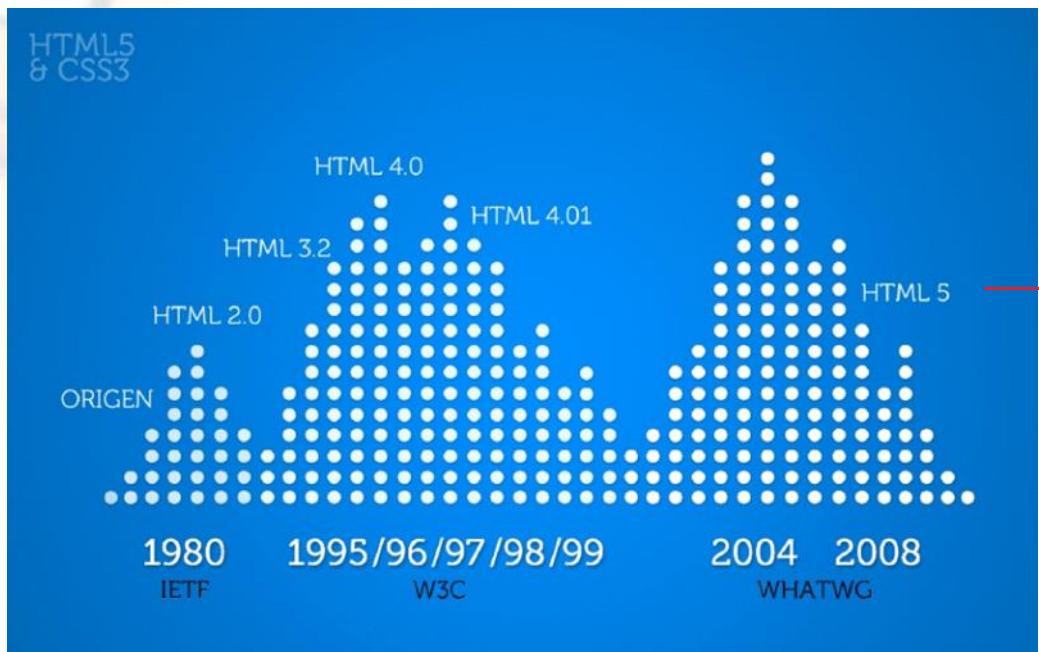
lunes, 19 de febrero de 2018 11:53

- se basa en la arquitectura de software **cliente – servidor**
 - Un servidor que almacena / genera recursos
 - Un cliente, e.g. navegador gráfico, solicita, recibe y gestiona esos recursos
- ambas capas se comunican mediante el **protocolo HTTP**
- cualquier recurso del servidor se localiza inequívocamente gracias al **sistema de identificación URL**
- los recursos más importantes que recibe el cliente están escritos en **HTML**, un lenguaje diseñado especialmente para definir la estructura y la semántica del **contenido**



Evolución

lunes, 19 de febrero de 2018 11:54



La versión definitiva de HTML5 se publicó en octubre de 2014

Actualmente

HTML 5.2

W3C Recommendation, 14 December 2017

<https://www.w3.org/TR/html5/>

HTML5 / CSS3

lunes, 19 de febrero de 2018 11:55

Conceptos semánticos

Nuevas Etiquetas

Nuevas APIs JavaScript



Semantics



CSS3



Multimedia



Graphics & 3D



Device Access



Performance



Offline & Storage



Connectivity

Tecnologías asociadas



Elementos clave en HTML5

Cambio de paradigma → Nueva filosofía basada en el pragmatismo
En cierto modo lo opuesto a XHTML

Predominio de los aspectos semánticos → Redefinición de etiquetas
Toda una serie de etiquetas nuevas
↓
Construidas en base al principio anterior

Soporte en los navegadores

domingo, 14 de mayo de 2017 13:39

Can I use <https://caniuse.com/>

The screenshot shows the CanIuse.com homepage. At the top, there's a navigation bar with 'Home' (highlighted in yellow), 'News', the date 'May 13, 2017 - New feature added: CSS text-orientation', and links for 'Compare browsers' and 'About'. Below the header is a large orange banner with the 'Can I use' logo and a settings icon. A message asks if the user wants to import usage data for Spain, with 'Import' and 'No thanks' buttons. The main content area has sections for 'Index of features', 'Latest features', 'Most searched features', 'Did you know?', 'Browser scores', and 'Third party tools'. The 'Browser scores' section shows a chart with Chrome at 290, Firefox at 287, Safari at 244, and Edge at 221. The 'Most searched features' section lists Flexbox, CSS Grid, CSS transforms, SVG, and CSS calc(). The 'Did you know?' section mentions importing usage data via the Settings panel. The 'Third party tools' section lists various CanIuse-related tools like the Embed, Component, command line tool, and Doluse..?.

Detected your country as "Spain". Would you like to import usage data for that country?

Import No thanks

Index of features

Latest features

- CSS text-orientation
- Encrypted Media Extensions
- CSS display: contents
- Link type "noreferrer"
- DOMMatrix

Most searched features

- Flexbox
- CSS Grid
- CSS transforms
- SVG
- CSS calc()

Did you know?

Usage data for all countries and continents can be imported via the **Settings panel**.

Browser scores

Current version	Dev version
Chrome 58: 290	
Firefox 53: 287	
Safari 10.1: 244	
Edge 15: 221	

About these scores

Third party tools

- The CanIuse Embed — Add support tables to your site
- Caniuse Component — Add support tables to your presentations
- Caniuse command line tool
- Doluse..? — Lint your CSS to check what features work
- I want to use — Select multiple features and see what % of users can use them

See full list

XML

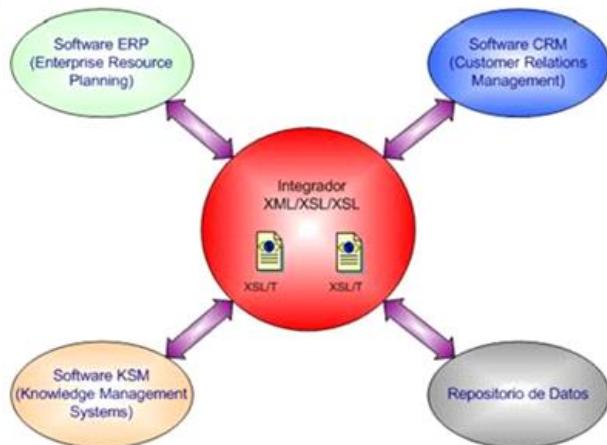
Lunes, 19 de febrero de 2018 12:02

eXtensible Markup Language - XML
(lenguaje de marcas extensible)

W3C (1998)



No se crea pensando en Internet, sino como estándar para el intercambio de información estructurada entre diferentes plataformas



Permite definir la gramática de lenguajes específicos para almacenar datos en forma legible en documentos estructurados de gran tamaño.

- bases de datos,
- editores de texto,
- hojas de cálculo
- casi cualquier cosa.

Entorno de trabajo

Lunes, 19 de febrero de 2018 11:37

Navegador	Chrome. Firefox, ... - herramientas de desarrollo
Editor de código	VSC - <i>plugins</i> adecuados a cada lenguaje / <i>framework</i> - Metalenguaje de escritura ágil de código: Emmet
Git	gestión del repositorio de software
Node.js:	entorno JS en el lado servidor permite crear servidores Web a medida
npm	gestor de paquetes, utilizado por otros muchos elementos
Servidor Web local	IIS / Apache (carpeta del proyecto como sitio) http-server desde cualquier carpeta
npm, Grunt, Gulp, Webpack...	gestión de tareas, como la compilación a CSS Metalenguajes: Preprocesadores de CSS



Navegadores

sábado, 9 de septiembre de 2017 18:20

Herramienta de desarrollador en las últimas versiones de Chrome, en este caso la 60.0.3



Herramienta de desarrollador en las últimas versiones de Firefox, en este caso la 55.0.3



Herramienta de desarrollador en Edge, el navegador incorporado desde Windows 10



Versiones "especiales" para desarrolladores



<https://www.google.es/chrome/browser/canary.html>



<https://www.mozilla.org/es-ES/firefox/developer/>

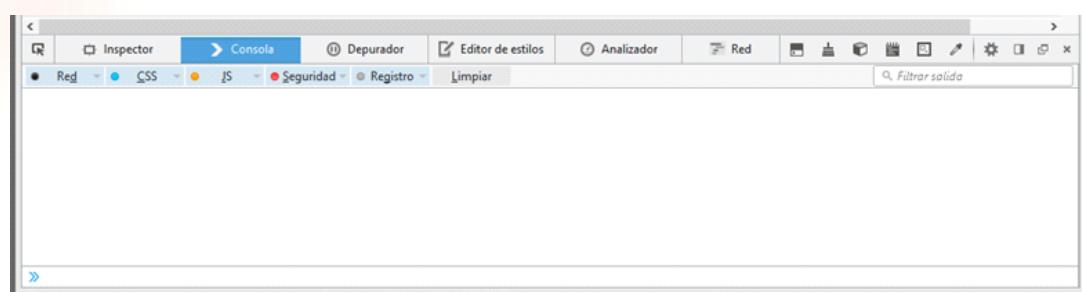
Herramientas

lunes, 19 de febrero de 2018 12:29

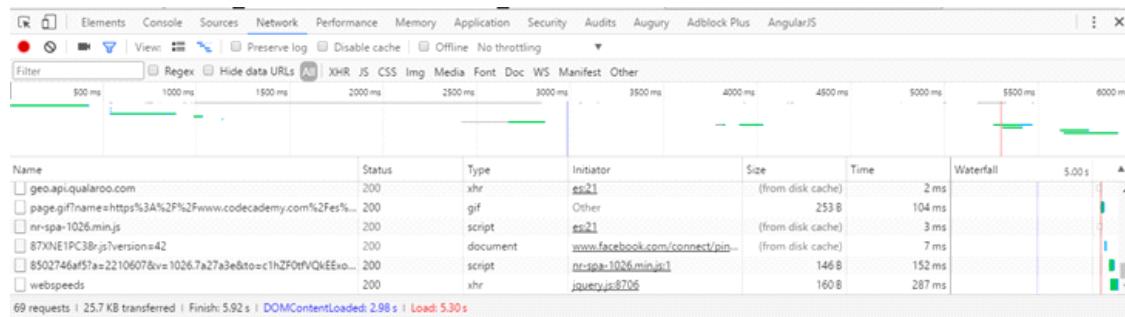
Consola JavaScript

El ambiente en el que se ejecutan los scripts (navegador) proporciona un objeto console, que corresponde a la consola JS que podemos hacer visible en la parte inferior del navegador.

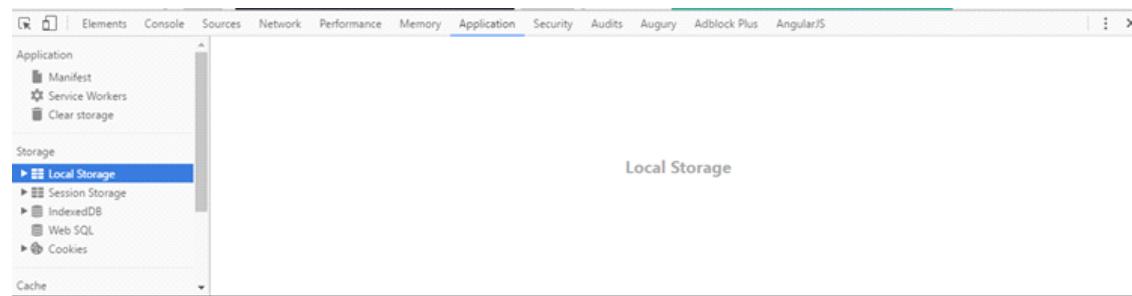
Los métodos console.log y console.dir son otra alternativa para presentar texto en pantalla desde un script. Algunos autores la prefieren a alert(), por considerarla menos intrusiva.



Datos de la red



Datos de la aplicación

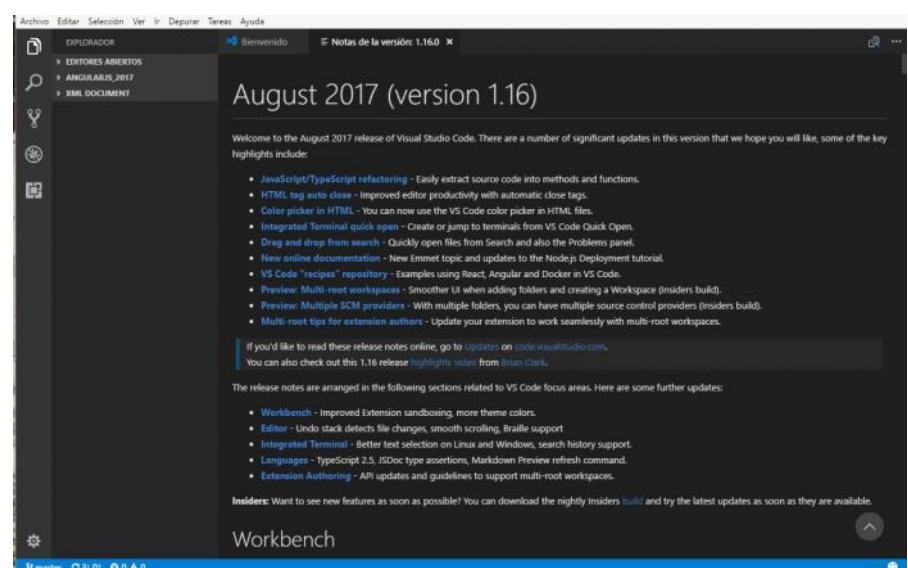
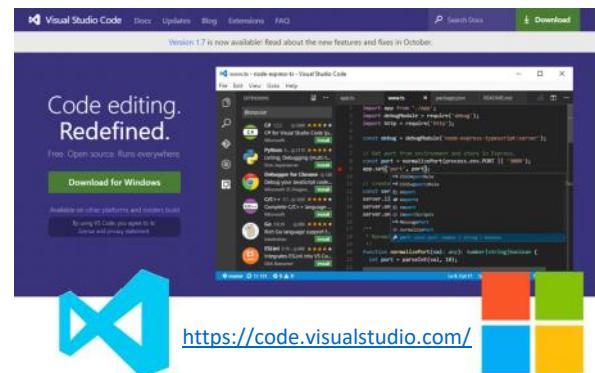


Editores de código

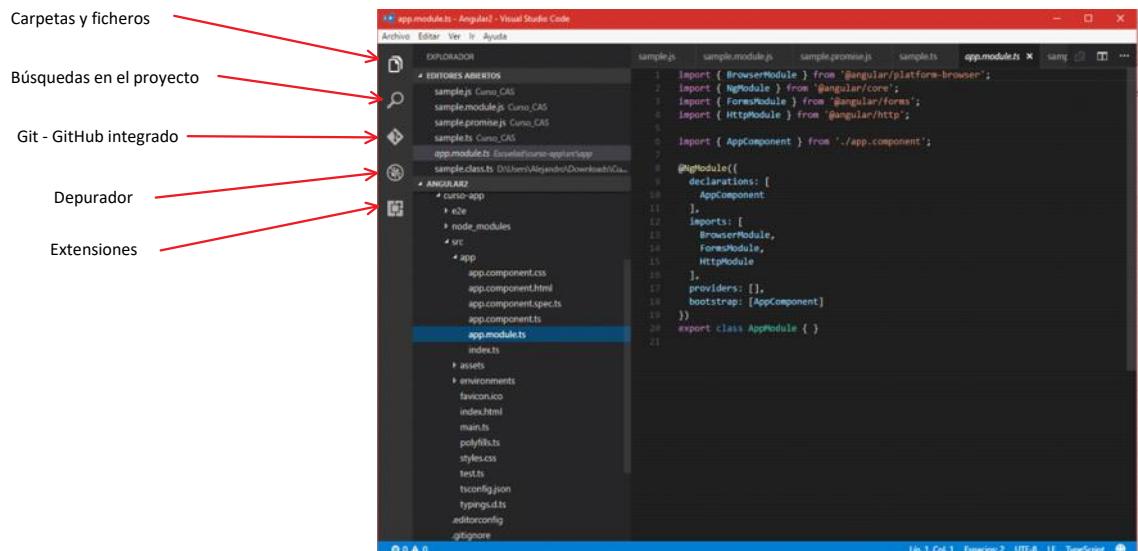
sábado, 9 de septiembre de 2017 18:21



Visual Studio Code



Visual Studio Code

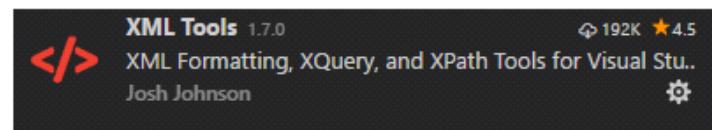
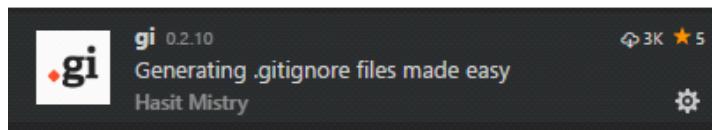
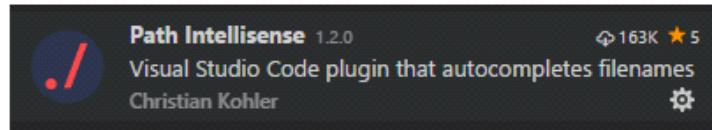


Extensiones

lunes, 19 de febrero de 2018 12:32

Extensiones de VSC

- Debugger para Chrome *****
(Microsoft)
- npm ***
(egamma)
- **Path Intellisense *****
(Christian Kohler)



Emmet

domingo, 14 de mayo de 2017 14:10

<https://emmet.io/>

- Herramienta de transformación a HTML con su sintaxis específica
- Escribe Emmet, aplicas el comando de transformación (expandir) y obtienes HTML
- Tienes que instalarse en el editor de código o IDE

The screenshot shows the official Emmet website. At the top left is the Emmet logo (a stylized 'E' icon) and the word "Emmet". Below it, the tagline "Tools for web-developers" is visible. On the left sidebar, there are links for "DOCUMENTATION", "DOWNLOAD", "CREDITS", and "BLOG". Below these, under "More developer tools:", are links for "Emmet LiveStyle" (Real-time bi-directional edit tool for CSS, LESS and SCSS) and "Emmet Review" (Fast and easy way to test responsive design side-by-side). A "Rollbar" advertisement is also present. The main content area features a large heading "Emmet — the essential toolkit for web-developers" and a subtext explaining it's a plugin for text editors. It includes a "Watch demo" button with a play icon. Below this, three cards highlight features: "HTML from CSS", "Dynamic snippets", and "Ultra-fast coding".

Emmet — the essential toolkit for web-developers

Emmet is a plugin for many popular text editors which greatly improves HTML & CSS workflow:

Watch demo

HTML from CSS You've already known how to use Emmet abbreviations: its syntax is inspired by CSS selectors.	Dynamic snippets Each abbreviation is transformed in runtime: just slightly change its name to get a different result .	Ultra-fast coding With Emmet you can quickly write a bunch of code , wrap code with new tags , quickly traverse and select important code parts and more!
---	---	---

Disponibilidad

domingo, 14 de mayo de 2017 16:32

Download

 Sublime Text cross-platform	 Atom cross-platform	 Coda OS X
 Eclipse/Aptana cross-platform	 TextMate 2 OS X	 Espresso OS X
 Chocolat OS X	 Komodo Edit cross-platform	 Notepad++ Windows
 PSPad Windows	 textarea browser based, cross-platform	 CodeMirror browser based, cross-platform
 Brackets cross-platform	 NetBeans cross-platform	 Adobe Dreamweaver Windows, OS X

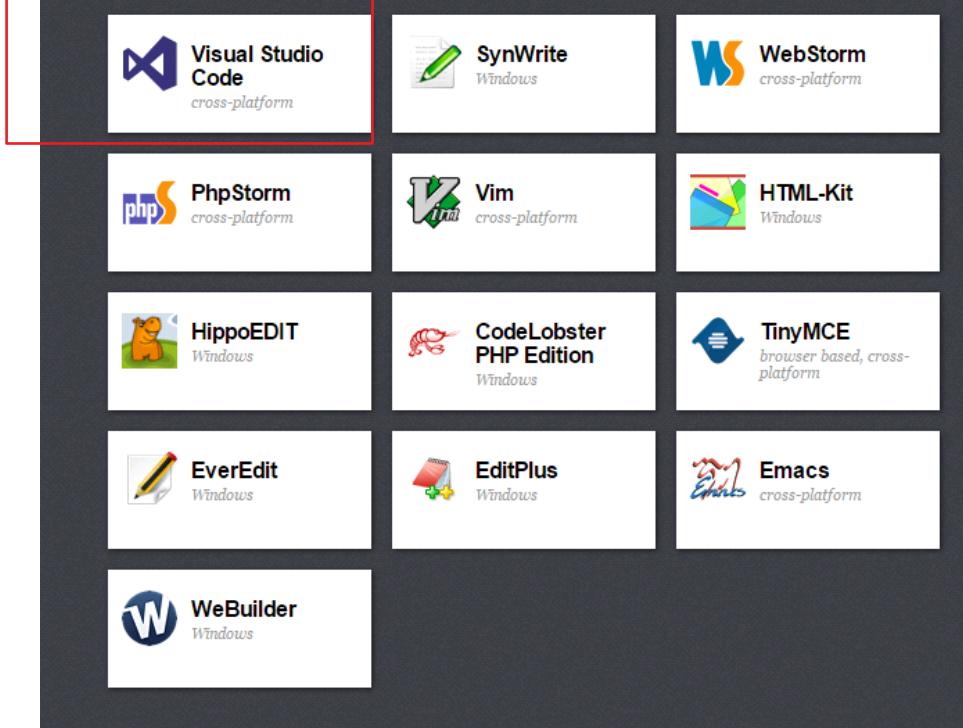
Online services

 jsfiddle the web playground	 JS Bin A JavaScript, HTML and CSS playground	 CodePen A front-end playground
 ICEcoder Online IDE	 Divshot The Interface Builder for Web Apps	 Codio Web IDE for web development, teaching and learning
 Codeanywhere Multi-platform cloud editor	 SourceLair In-browser IDE for web technologies	

Third-party support

The plugins for these editors are developed by third-party developers. May not support all Emmet features and actions.

En el caso de [Visual Studio Code](#), su propia implementación de [Emmet](#) ya está integrada en el editor, sin necesidad de añadir ningún plugin



Sintaxis

domingo, 14 de mayo de 2017 16:43

<https://docs.emmet.io/>

Muy similar a los selectores CSS

Selectores individuales

elemento
elemento.clase.clase
elemento#id
elemento[atributo atributo]

Contenido y repetición

{contenido}
* n : número de veces que se repite
\$: referencia al número de item

Selectores múltiples

Child: >
Sibling: +
agrupamientos ()

Node y npm

sábado, 9 de septiembre de 2017 18:21

JS en el servidor (SSJS): Node.js

<https://nodejs.org/>



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo.

Important security releases, please update now!

Descargar para Windows (x64)

v6.11.3 LTS

Recomendado para la mayoría

v8.4.0 Actual

Últimas características

Otras Descargas | Cambios | Documentación del API

Otras Descargas | Cambios | Documentación del API

Node.js® es un **entorno de ejecución para JavaScript** (una plataforma de software)

- Se emplea para construir aplicaciones de red escalables (especialmente servidores).
- Está construido con el motor de JavaScript V8 de Chrome
- Utiliza un modelo de operaciones que lo hace liviano y eficiente, gracias a
 - o **operaciones E/S sin bloqueo** y
 - o orientado a eventos, con un **bucle de eventos de una sola hebra**

Además, el **ecosistema de paquetes de Node.js, npm**, es el ecosistema más grande de librerías de código abierto en el mundo.

Su funcionalidad es especialmente adecuada en

- operaciones en tiempo real (e.g. chats)
- Bases de datos *NoSQL* / No relacionales

Node.js: ampliación de JS



Al *core* de JS no le acompañan las APIs habituales en cualquier lenguaje de programación

→ Node.js puede entenderse como la ampliación de JS para llegar a ser un lenguaje "completo", independiente de un entorno huésped

v8 (JavaScript)

- Una gramática que define la sintaxis del lenguaje
- Un intérprete/compilador que lo sabe interpretar y ejecutar
- Mecanismos para interactuar con el mundo exterior (llamadas al sistema)
- Librería estándar (consola, ficheros, red, etc...)
- Utilidades (intérprete interactivo, depurador, paquetes)

Node.js

También puede verse como un entorno huésped para JS en el servidor

Node.js: orígenes



Tiene su origen en un proyecto de **Ryan Dahl** y sus colaboradores en la empresa *Joyent*, que fue presentado en una conferencia en la *JSConf* de 2009.

<https://youtu.be/ztspvPYybIY>

Escrito en C/C++ y JS

Objetivo del proyecto

→ Escribir aplicaciones muy eficientes en E/S con el lenguaje dinámico más rápido (v8) para soportar miles de conexiones simultáneas

Planteado sin complicaciones innecesarias

- Concurrencia sin paralelismo
- Lenguaje sencillo y muy extendido: JS
- API muy pequeña y muy consistente
- Apoyándose en Eventos y *Callbacks*

npm

lunes, 19 de febrero de 2018 12:37

Paquetes: npm

Los módulos suelen distribuirse en forma de paquetes (*packages*) o aplicaciones:

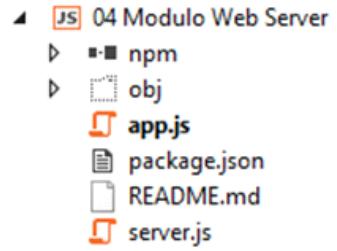
- colección de módulos
- fichero *manifest* (JSON)
- fichero de ayuda (md)

describe el paquete y sus dependencias

pueden publicarse de forma pública o privada

npm es la herramienta de instalación de paquetes incluida en Node.js

Los proyectos Node.js creados en Visual Studio ya responden a la estructura de paquetes, para facilitar la creación de estos



Instalación de Node.js & npm

sábado, 9 de septiembre de 2017 20:35

The screenshot shows the Node.js Setup Wizard. The first window is titled "Welcome to the Node.js Setup Wizard". It displays the Node.js logo and a message: "The Setup Wizard will install Node.js on your computer. Click Next to continue or Cancel to exit the Setup Wizard." Below this are "Back", "Next", and "Cancel" buttons. A second window titled "Installation Folder" is overlaid, showing the path "Program Files\nodejs\". It includes a "Change..." button and a tree view with three items: "Node.js runtime", "npm package manager", and "Online documentation shortcuts". A red box highlights the "node" and "npm" items under the "Online documentation shortcuts" section.

A screenshot of the Windows Start Menu. It shows five icons: "Node.js command prompt", "Node.js documentation", "Node.js website", "Node.js", and "Uninstall Node.js". A red arrow points from the "Node.js command prompt" icon to a terminal window below.

A terminal window titled "Node.js command prompt". It displays the message: "Your environment has been set up for using Node.js 8.1.4 (x64) and npm." Below this, the command "C:\Users\alce6>S" is entered. A red box highlights the message area.

A terminal window titled "Símbolo del sistema". It shows the command "C:\Users\alce6>node -v" followed by "v8.1.4". Below it, the command "C:\Users\alce6>npm -v" is shown followed by "5.0.3".

A terminal window titled "Node.js command prompt - node". It shows a session in the Node.js REPL. The user types "var a = 12;" and "var b = 3;", then "console.log(a+b);", which outputs "36". A red box highlights the output "36". Below the window, a gray box contains the text "Salida ctrl-C o ctrl-D".

Con el comando "node" entramos en la consola de Node, un entorno REPL (*Read-Eval-Print-Loop*) similar a la consola de JS en los navegadores

El mismo comando node <fichero.js>, permite la ejecución de un fichero

Servidor Web

lunes, 19 de febrero de 2018 12:37

1. Elementos del lenguaje
2. Estructura del documento.
 - Inicio del. Etiquetas !DOCTYPE y HTML
 - Jerarquía: Cabecera y cuerpo. Etiquetas HEAD y BODY
3. Head
 - Cabecera. Etiqueta TITLE. Inclusiones e iconos. Metadatos.
 - Conjunto de caracteres. Caracteres especiales
4. Concepto de validación
 - Primera página Web
 - Omisiones y errores: funcionamiento del navegador
5. Carácter descriptivo. Tipos de elementos



Etiquetas en HTML

nombre de
la etiqueta

<p>

paréntesis
en ángulo
izquierdo

paréntesis
en ángulo
derecho

nombre de
la etiqueta

</p>

paréntesis
en ángulo
izquierdo

barra
(slash)

paréntesis
en ángulo
derecho

Etiqueta de apertura

Etiqueta de cierre

Etiquetas dobles (two-sided tag)

```
<i>Everything</i>
```

```
<a href="index.htm">Click Here</a>
```

Etiquetas en HTML

nombre de
la etiqueta

paréntesis
en ángulo
izquierdo

paréntesis
en ángulo
izquierdo

La etiqueta es auto
conclusiva: se cierra
ella misma y contiene
toda la información
necesaria

En XHTML sería:

Etiquetas

```

```

HTML

Etiquetas sencillas (one-sided tag)

```

```

HTML

```

```

XHTML

Atributos

lunes, 19 de febrero de 2018 14:27

Etiquetas en HTML: atributos

nombre de
la etiqueta

nombre del
atributo

valor del
atributo

< p lang=“es” >

< nombre_etiqueta nombre_atributo = “valor del atributo” >

< a href=“index.htm”> Click Here </ a >

Etiqueta: <a>

Atributo: href=“...”

Elemento Click Here

< img src=“tree.gif” />

Etiqueta:

Atributo: scr=“...”

Atributos globales

Lunes, 19 de febrero de 2018 14:24

Los atributos suelen ser específicos de una o varias etiquetas
Existen unos pocos atributos globales: válidos para todas las etiquetas

Atributos globales

class	Nombre de estilo aplicable a diversos elementos
id	Nombre de estilo aplicable a un elemento
style	Estilo CSS en línea
title	Información extra sobre un elemento
lang	Idioma asociado a un elemento
translate	Elemento que se debe traducir en una página “localizado”
spellcheck	Activa o no la comprobación de ortografía y gramática
dir	Dirección del texto
tabindex	Orden tabular de un elemento
acceskey	Tecla rápida que activa/da el foco a un elemento
contenteditable	Carácter editable o no
contextmenu	Menú contextual de un elemento
draggable	Posibilidad de arrastrar de un elemento
dropzone	Acción por defecto al arrastrar un elemento
hidden	Indica el carácter no relevante de un elemento

Atributos id y class

Permiten identificar a cualquier etiqueta utilizada.

```
<etiqueta id="id_1">
```

La etiqueta utilizada queda identificada exclusivamente como id_1
En un documento nunca debe asignarse la misma id a más de una etiqueta

```
<tag_2 class="cls_1 cls_2 ...">
```

La etiqueta utilizada queda identificada como perteneciente a una clase,
En la que se puede incluir cualquier número de todo tipo de etiquetas

Convenciones y Comentarios

Lunes, 19 de febrero de 2018 14:22

Etiquetas en HTML: formato y “convenciones”

Los nombres de etiqueta, los nombres de atributos y los valores pueden ir en mayúscula o minúscula

lo habitual es que , los nombres de atributos y los valores vayan en minúscula

No es necesario encerrar los valores de los atributos entre comillas

lo habitual es que los valores de los atributos se indiquen entre comillas

Comentarios

texto del comentario

<!--Comentario-->

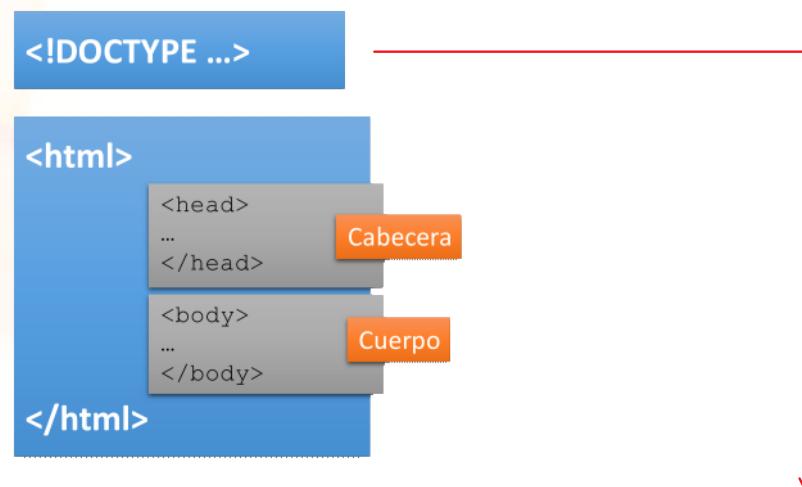
paréntesis en
ángulo izquierdo,
admiración,
dos guiones

dos guiones,
paréntesis en
ángulo
izquierdo

<!--Esto es un comentario en HTML-->

Estructura del documento

lunes, 19 de febrero de 2018 14:36



Inicio del documento HTML

<!DOCTYPE ...>

Especificación del tipo de documento
(opcional, excepto en XHTML)

Le indica al navegador que el documento cumple determinado estándar, que será el que aquel utilizará para procesar el documento

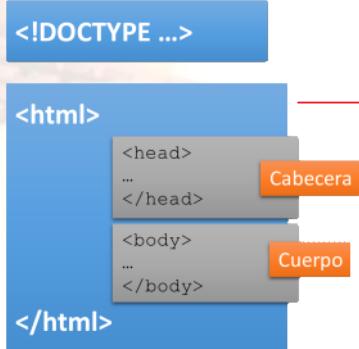
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

<!DOCTYPE html>

HTML5





Inicio del documento HTML

Documento HTML
(todo lo demás)

```

<html>
...
</html>

```

Es el contenedor raíz en la estructura HTML, es decir dentro del que vamos a incluir todo el documento

```

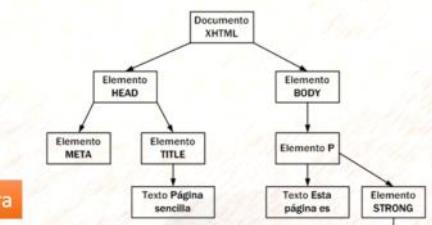
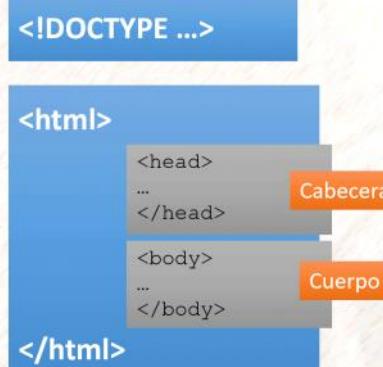
<html lang="es">
...
</html>

```



En la etiqueta html se incorpora el **atributo global "lang"**
lang= idioma (es, en, fr, ge, it, ru...)
Los idiomas pueden incluir dialectos: es~es (español de España)

Partes del documento HTML: cabecera (head) y cuerpo (body)



Refleja el carácter de estructuras jerárquicas de los documentos HTML

lunes, 19 de febrero de 2018 14:44

Head

lunes, 19 de febrero de 2018 14:48

La cabecera (head)

Cabecera

```
<head>  
...  
</head>
```

Titulo de la página

```
<title>Curso HTML5</title>
```

Metadatos

```
<meta name="valor" content="valor">
```

Inclusiones de archivos externos

```
<link rel="valor" href="valor">
```

Estilos y script embebidos en el documento

```
<style>.....</style>
```

Etiqueta <base>

<base>

especifica la url por defecto para todas las urls relativas y el destino por defecto de los hiperenlaces incluidos en al página

Atributos

href	corresponde a la url por defecto
target	indicael destino por defecto de los hiperenlaces

_blank
_parent
_self
_top
framename

```
<base href="http://www.miweb.com/images/" target="_blank">
```

Etiqueta <title>

<title>Título</title>

Corresponde al título del documento web

<title>Hola Mundo!</title>

Normalmente será el texto mostrado en la barra superior de la ventana o pestaña del navegador



<title> es también un **atributo global**, que incorpora información adicional a cualquier etiqueta. En muchos casos se muestra como un recuadro emergente al pasar el ratón sobre un objeto

Inclusiones: css y scripts

<link>

usado para incorporar archivos externos, correspondientes a .

- estilos,
- imágenes o iconos

```
<link rel="stylesheet" href="misestilos.css">
```

Atributos

rel

significa "relación"; establece el uso que se hará del archivo:

- stylesheet,
- shortcut icon,
- apple-touch-icon

href

corresponde a la referencia del archivo

<script>

usado para incorporar archivos externos, correspondientes a códigos de **scripts** (i.e. JavaScript)

```
<script type='text/javascript' src='.../jquery.min.js'>
```



El atributo `rel` permite definir otros usos de la etiqueta `link`, como la inclusión de componentes web:

```
<link rel="import" href="componente.html">
```

Muy recientemente (a finales de 2017) se han introducido novedades que permiten el uso de módulos ES6 por parte de los navegadores

```
<script type="module" src="app.js"></script>
```

Iconos: favicon

```
<link rel="shortcut icon" href="fav_icon.png">
```

.ico → problemas fuera
de Windows
.png



25x25 px

El equivalente para los dispositivos Apple es
<link rel="apple-touch-icon" href="favicon.png">

Link junto con el atributo **alternate** se utiliza para incluir información sindicada, en sus formatos rss y atom

```
<link rel="alternate" href="atom.xml" type="application/atom+xml" title="Atom">  
<link rel="alternate" href="rss.xml" type="application/rss+xml" title="RSS">
```

Metadatos: name/content

<meta ...>

Atributos
name
... content

Cada etiqueta <meta> establece un metadato, utilizando los atributos: name y content, http-equiv y content o charset. Pueden sucederse tantos <meta> como sea necesario

```
<meta name="keywords"  
      content="pests, weeds, plants, flowers, ....">  
<meta name="author" content="Hege Refsnes">  
<meta name="application-name" content="...">  
<meta name="description" content="...">  
<meta name="generator" content="...">
```

Combinando los atributos *name / content* se establece el valor de una serie de metadatos o descriptores con información sobre el documento

- nunca se muestra en el navegador
- la utilizan motores de búsqueda y dispositivos que necesitan hacer una vista previa del documento u obtener un sumario de la información que contiene.

Metadatos: http-equiv

Atributo
http-equiv
... content

```
<meta http-equiv="refresh"  
      content="5; url=http://.../newpage.htm">  
<meta http-equiv="content-type" content=  
      "text/html; charset=UTF-8">  
<meta http-equiv="default-style" content="....">
```

- después de **content-type**, el atributo *content* permite indicar el tipo MIME del contenido (text/html) y el charset o página de códigos de caracteres (no se usa en HTML5)
- después de **default-style**, el atributo *content* permitía indicar la hoja de estilo preferida (si se han incluido varias)
- después de **refresh**, el atributo *content* permite definir un intervalo de tiempo para que se refresque la pantalla,
Además se puede indicar que cargue una URL diferente: permite por tanto redirigir a una URL diferente de la indicada en el navegador

Character Sets (<meta charset>)

Atributo
charset

<meta charset="utf-
8">

HTML5



La innovación de este elemento en HTML5, como en la mayoría de los casos, es solo **simplificación**. La nueva etiqueta <meta> para la definición del tipo de caracteres es más corta y simple.

Definición de la páginas de códigos que utilizará el agente de usuario (navegador) al renderizar un documento HTML

es crítico para la correcta presentación del documento, al determinar la página de códigos que utilizará el navegador debe coincidir con la codificación empleada en el programa que genera el documento HTML

Páginas de códigos: ASCII

Las páginas de códigos son tablas que definen el conjunto de caracteres, es decir que código lingüístico corresponde a cada conjunto de bits, dependiendo del alfabeto utilizado por un documento.

Inicialmente se utilizaba el conjunto de caracteres **ASCII** (*American Standard Code for Information Interchange*):

Cada carácter está representado por un número de 7 bits, lo que permite codificar únicamente 128 caracteres: letras mayúsculas y minúsculas, números y algunos caracteres habituales.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	□	Γ	└	┘	↙	☒	☑	^K	↶	↷	≡	ψ	₩	◀	⊗	○
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/ A
3	0	1	2	3	4	5	6	7	8	9	:	<	=	>	?	B
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ó C
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\	^	_	D
6	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	O E
7	p	q	r	s	t	u	v	w	x	y	z	{	}	~	DEL F	

Estos caracteres son entendidos en cualquier ordenador y por tanto pueden ser usados con libertad en un documento HTML

Páginas de códigos: ISO

Las páginas de **códigos iso-8859** son todos super sets del estándar US-ASCII, donde cada carácter se codifica mediante 1 Byte (8 bits), por lo que solo tienen cabida 256 caracteres diferentes: los 128 del ASCII y 128 del ASCII “extendido”
En este último existen variaciones entre los distintos conjuntos de códigos ISO

Una cadena de un Byte, de valor 11100001, (corresponde decimal 225)

Si se aplica la página de códigos iso-8859-1 (Latín)

Si se aplica la página de códigos iso-8859-6 (Árabe)

Si se aplica la página de códigos iso-8859-7 (Griego)

á
ف
α

Páginas de códigos: ISO

iso-8859-1	Latin 1	Lenguajes de Europa Occidental (Inglés, Francés...)
iso-8859-2	Latin 2	Lenguajes de Europa Central y Oriental (Checo, Polaco...)
iso-8859-3	Latin 3	Lenguajes de Europa del Sur, Esperanto y otros
iso-8859-4	Latin 4	Lenguajes de Europa del Norte (Letón, Lituano, Groenlandés...)
iso-8859-5	Cyrillic	Lenguajes de alfabeto cirílico (Búlgaro, Ruso, Serbio...)
iso-8859-6	Arabic	Lenguajes de alfabeto arábico
iso-8859-7	Greek	Griego moderno, signos matemáticos
iso-8859-8	Hebrew	Lenguajes de alfabeto hebreo
iso-8859-9	Latin 5	Turco
iso-8859-10	Latin 6	Lenguajes Nórdicos (Islandés, Inuit)
iso-8859-15	Latin 9 / Latin 0	Modificación del iso-8859-1 que incluye algunos caracteres nuevos, incluyendo el símbolo del Euro
ISO-2022-JP	Japanese 1	Lenguaje japonés
ISO-2022-JP-2	Japanese 2	Lenguaje Japonés
ISO-2022-KR	Korean	Lenguaje coreano

UNICODE

A partir de HTML 4.0 se soporta el **conjunto de caracteres (charset)** UNICODE.

Definido por el consorcio internacional UNICODE (<http://www.unicode.org/>) encargado de mantener el estándar en colaboración con ISO/IEC

- sistemas de escritura **modernos**: árabe, braille, copto, cirílico, griego, hebreo, latino, devanagari (sánscrito), sinogramas (hanja coreano, hanzi chino y kanji japonés), silabarios japoneses (hiragana y katakana), ...
- escrituras **históricas** extintas, para propósitos académicos: cuneiforme, griego antiguo, lineal B micénico, fenicio y rúnico.
- caracteres **no alfabéticos** como símbolos musicales y matemáticos, fichas de juegos como el dominó, flechas, iconos etc.



La versión 6.0 contiene
109.449 caracteres

Planos UNICODE

Cada conjunto definible por 2 bytes constituye un “plano” (*Plane*). En teoría podría corresponder a 65.536 caracteres (depende del sistema de codificación)

- BMP (*Basic Multilingual Plane*)
Agrupa los símbolos más habituales de la mayoría de lenguas actuales
- SMP (*Supplementary Multilingual Plane*): Lenguas antiguas y mas
- SIP (*Supplementary Ideographic Plane*): Ext. CJK (China, Japón, Corea)
- TIP (*Tertiary Ideographic Plane*): Lenguas antiguas asiáticas
- SSP (*Supplementary Special-purpose Plane*): usos especiales

Compatibilidad. BMP comienza por los caracteres del ASCII (a los que asigna el mismo número que tenían originalmente)

Páginas de códigos: UTF

UNICODE, permite varios sistemas de codificación conocidos como los conjuntos UTF (*Unicode Transformation Format*)

- UTF-8 utiliza un sistema no modal de longitud variable, de 1 a 4 Bytes , donde el valor de los primeros bits determina cuantos Bytes corresponden a un determinado código. Con ello se supera sin limitaciones el numero de posibles caracteres optimizando el uso de memoria para los primeros signos codificados (i.e. lenguas latinas)
- UTF-16 es también un sistema de codificación de caracteres de longitud variable, utiliza 2 Bytes para codificar el plano BMP y 4 Bytes para otros planos
- UTF 32 es de longitud fija de 4 Bytes, resultando más eficiente con los planos que no son BMP.
Teóricamente se podría llegar a codificar $2^{32} = 4.294.967.296$

Niveles de codificación

Al generar un documento

Le indicamos al editor en que código estamos trabajando

En un editor tan simple como el bloc de notas podemos elegir varias formas de codificación

Nombre: Nuevo documento de texto.txt

Tipo: Documentos de texto (*.txt)

Ocultar carpetas

Codificación:

- ANSI
- ANSI
- Unicode
- Unicode big endian
- UTF-8

Al leer un documento en el navegador

El navegador configurado correctamente utiliza el código indicado en los metadatos del documento

Caracteres especiales

Entidades

Secuencia de escape con nombre : &Nombre;
Secuencia de escape sin nombre : \$#123;

& (ampersand)	&	&	© (copyright)	©	©
< (menor que)	<	<	® (marca registrada)	®	®
> (mayor que)	>	>	° (grado)	°	°
(espacio fuerte)	&nbsp	 	± (más / menos)	±	±
¢ (centavo)	¢	¢	“ (comillas)	"e	"
£ (libra)	£	£	† (cruz o dagger)	†	†
¥ (yen)	¥	¥	™ (trademark)	™	™

El espacio fuerte (*nonbreaking space*) suele ser usado en exceso, incluso por algunos editores WYSIWYG y conversores de texto. Su uso debe ser el imprescindible y en ningún caso puede sustituir una correcta disposición (*layout*)

Carácteres especiales del español

En español, el uso de vocales acentuadas, la eñe y los símbolos de admiración e interrogación suponía el empleo de otros tantos caracteres especiales

Carácter	Secuencia escape	Sec. escape numérica	Carácter	Secuencia escape	Sec. escape numérica
á	á	á	Á	Á	Á
é	é	é	É	É	É
í	í	í	Í	Í	Í
ó	ó	ó	Ó	Ó	Ó
ú	ú	ú	Ú	Ú	Ú
ü	ü	ü	Ü	Ü	Ü
ñ	ñ	¤	Ñ	Ñ	¥
¡	-----	¡	¿	-----	¿

Bibliotecas de iconos: Font awesome

lunes, 19 de febrero de 2018 14:51

Instalación

```
npm install font-awesome
```

Utilización

Ejemplo librería que proporciona elementos de interfaz basándose únicamente en la incorporación de CSS



<https://fontawesome.com/>

Se añaden como Link CSS [o como *import*]

```
<link rel="stylesheet" href="../../node_modules/font-awesome/css/font-awesome.min.css">  
[@import "../../node_modules/font-awesome/css/font-awesome.min.css"]
```

Se insertan los iconos como elementos HTML vacíos, e.g. <i></i> a los que se aplica la clase adecuada

```
<i class="fa fa-star" aria-hidden="true"></i>
```

Validación

lunes, 19 de febrero de 2018 15:20

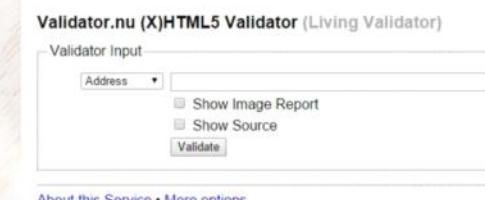
Validación

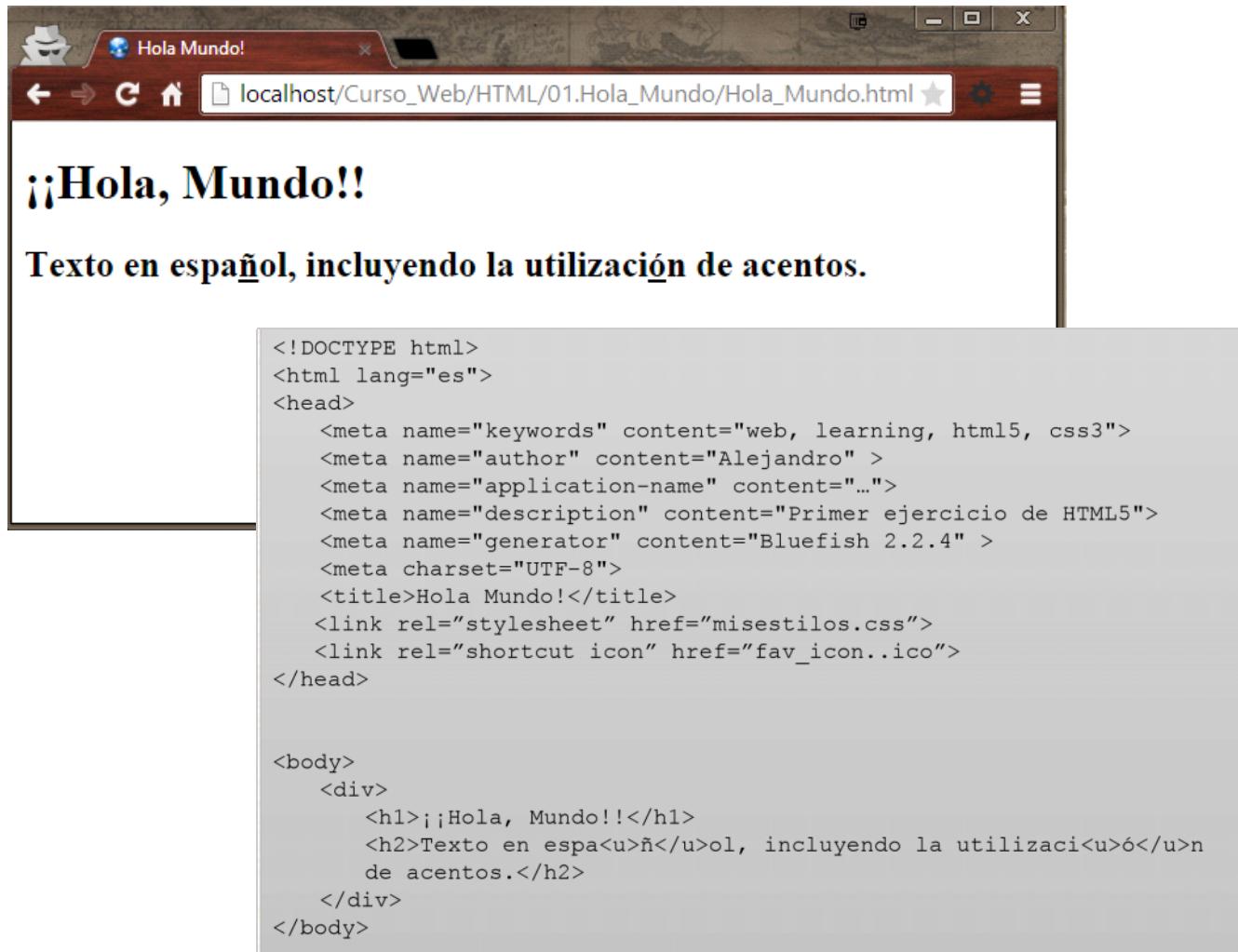
Herramienta para detectar errores o ausencias en el código que puedan provocar problemas de visualización en algún navegador

- El estilo en la sintaxis es independiente de la validación.
- Queda como opción de cada desarrollador.
- Conviene respetar cierta coherencia (Buenas prácticas)

W3C Markup Validation Service (<http://validator.w3.org/>)

(X)HTML5 Validator (<http://html5.validator.nu/>)





¡¡Hola, Mundo!!

Texto en español, incluyendo la utilización de acentos.

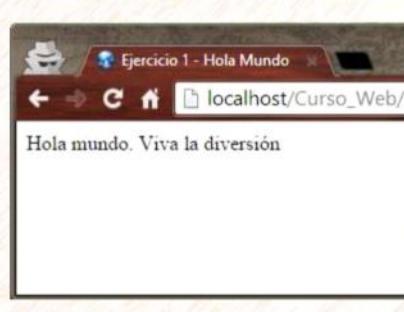
```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta name="keywords" content="web, learning, html5, css3">
    <meta name="author" content="Alejandro" >
    <meta name="application-name" content="...">
    <meta name="description" content="Primer ejercicio de HTML5">
    <meta name="generator" content="Bluefish 2.2.4" >
    <meta charset="UTF-8">
    <title>Hola Mundo!</title>
    <link rel="stylesheet" href="misestilos.css">
    <link rel="shortcut icon" href="fav_icon..ico">
</head>

<body>
    <div>
        <h1>¡¡Hola, Mundo!!</h1>
        <h2>Texto en español, incluyendo la utilización de acentos.</h2>
    </div>
</body>
```

Solución mínima

lunes, 19 de febrero de 2018 15:23

Ejercicio (1): "solución mínima"



```
1 <!DOCTYPE html>
2
3 <html lang="es">
4
5   <head>
6     <meta charset="UTF-8">
7     <title>Ejercicio 1 - Hola Mundo</title>
8     <link rel="shortcut icon" href="fav_icon.png">
9   </head>
10
11   <body>
12     Hola mundo. Viva la diversión
13   </body>
14
15 </html>
```



```
1 ll Hola Mundo!!
2
```

Omisiones y navegadores



Si ignoramos todas las normas del HTML

El navegador intenta llenar los huecos (e.g. crea elementos html, head y body)

Al final presenta tantos elementos como pueda

- El papel del **navegador** es muy activo: siempre intenta presentar el documento
- Las **diferencias** entre navegadores cobran mayor importancia

HTML como lenguaje descriptivo

Objetivo → describir cómo deberá construirse una página, de manera que examinando esa descripción el navegador del usuario final sea capaz de mostrarlo de la mejor manera posible

Con HTML podremos indicarle al navegador que este es un titular, aquí comienza un párrafo, estos son elementos de una lista, etc. posteriormente el navegador decidirá como mostrar esos elementos.

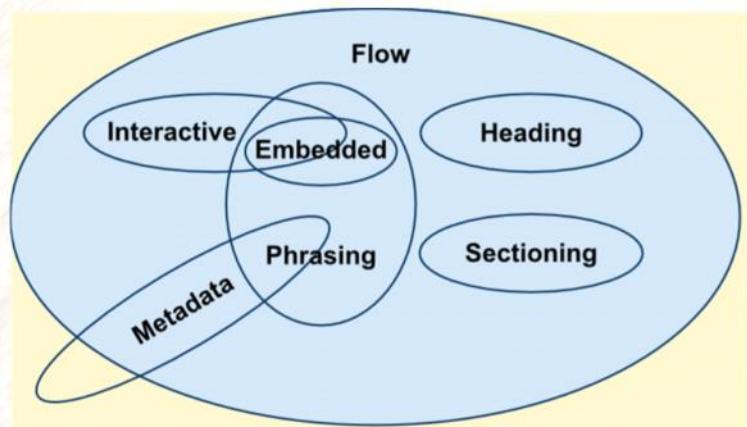
Consecuencia → Lenguaje independiente del S.O, de las características gráficas, o incluso de la ausencia de gráficos

Categorías de contenido

HTML5 redefine / completa
estos conceptos
de bloque y linea

Categorías de contenido
(*content categories*)

- Flow
- Phrasing
- Heading
- Sectioning
- Embedded
- Interactive
- Metadata

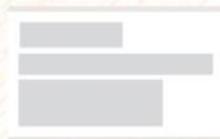


contenido dinámicos o de flujo (flow) : a nivel de bloque"
contenido de enunciados (phrasing) : en línea

<https://www.w3.org/TR/html52/dom.html#kinds-of-content>

Elementos “en bloque” y “en línea”

Elementos “en bloque” (block level elements)



empiezan en una nueva ‘línea’ en la ventana del navegador

`<p>`, `<h1>`, `<hr>` ``, y ``.

Cualquier conjunto de elementos puede convertirse en un bloque incluyéndolo en la etiqueta `<div>...</div>`

Elementos “en línea” (inline elements)



si es posible, ocupan la misma línea que los elementos que les rodean

``, `<i>`, `<a>`, ``

Cualquier conjunto de elementos dentro de un bloque puede delimitarse como elemento en línea incluyéndolo en la etiqueta `...`

Aunque útil en la práctica, esta distinción corresponde sólo al valor por defecto de una propiedad, que puede modificarse en cualquier caso



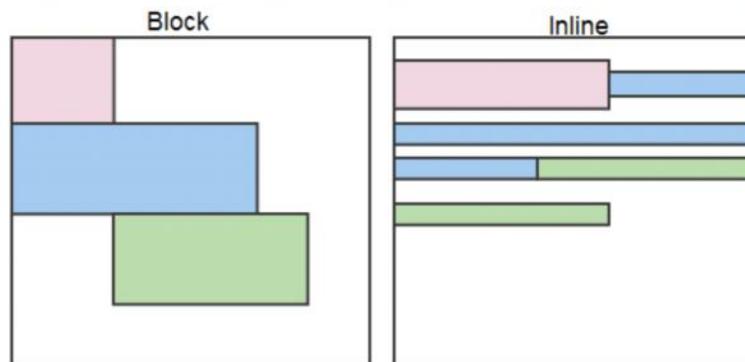
Display

Desde CSS 2.1,
la propiedad
display

Establece el tipo de caja generada por un elemento, independientemente de su modo de funcionamiento por defecto

Valores
básicos

Inline : contenido en línea
block: : contenido en bloque



Elementos div y span

Las etiquetas **<div>** y ****, que define un grupo de elementos en bloque o en línea sin modificarlos en nada, son muchas veces el paso previo para aplicarles un identificado individual de clase:

```
<span id="nombre_estilo">...</span>
```

Al definir clases propias de una página HTML, estos selectores pueden utilizarse con un carácter eminentemente semántico, en la misma línea que las nuevas etiquetas para divisiones incorporadas en HTML5, especialmente combinados con las etiquetas
<div> y ****,

Texto, Hipertexto...Imágenes

Como indica el propio nombre, HTML es un sistema de hipertexto, concepto aparecido en 1965 y llevado a la práctica en 1967. Ya entonces supone

- grandes cantidades de texto en pantalla,
- enlaces dentro de un documento que conducen
- a otras partes del mismo documento o a otro documento.

En HTML, junto al manejo de los **textos**, resulta clave la creación de **enlaces** (o hiperenlaces)

<a>

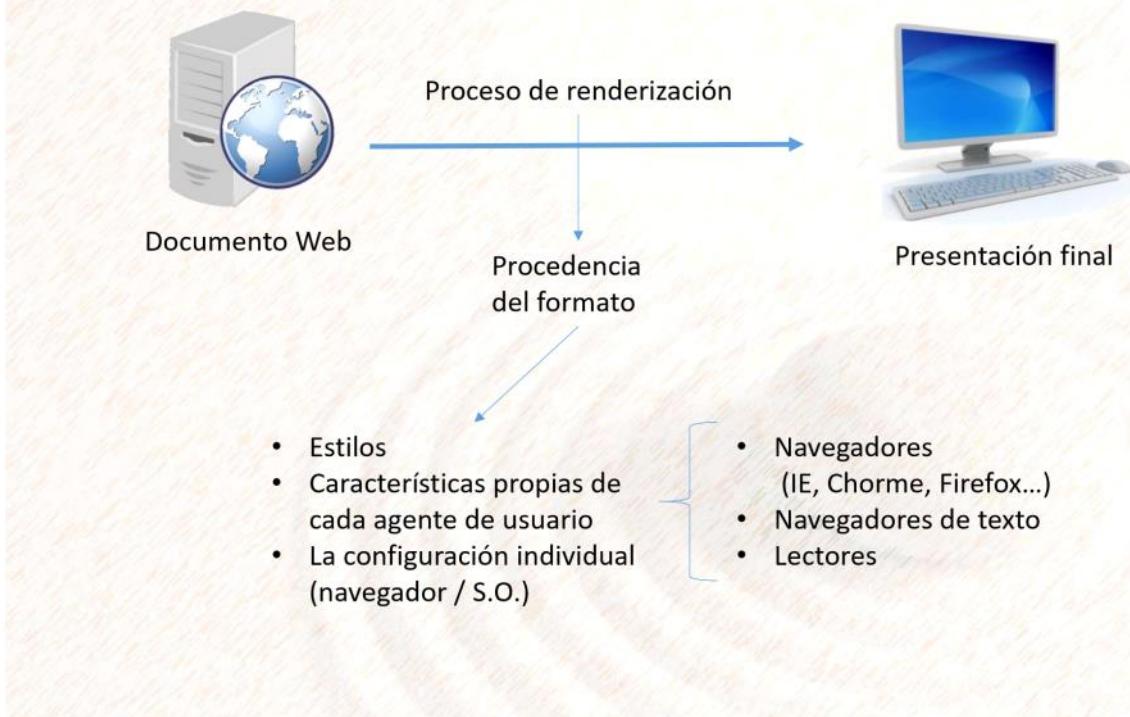
 Texto

Tan sólo 3 años después de su creación se incorpora el tercer elemento clave, las **imágenes**

- a. Elementos básicos.
 - 1. Concepto de estilos CSS
 - 2. Sintaxis: declaraciones de estilos y comentarios
 - 3. Ubicación
 - 4. Directivas en CSS
- b. DOM. Selectores
 - 1. DOM
 - 2. Selectores
 - 3. Herencia
- c. Múltiples CSS
 - 1. Cascada de estilos. Precedencia.
 - 2. Especificidad



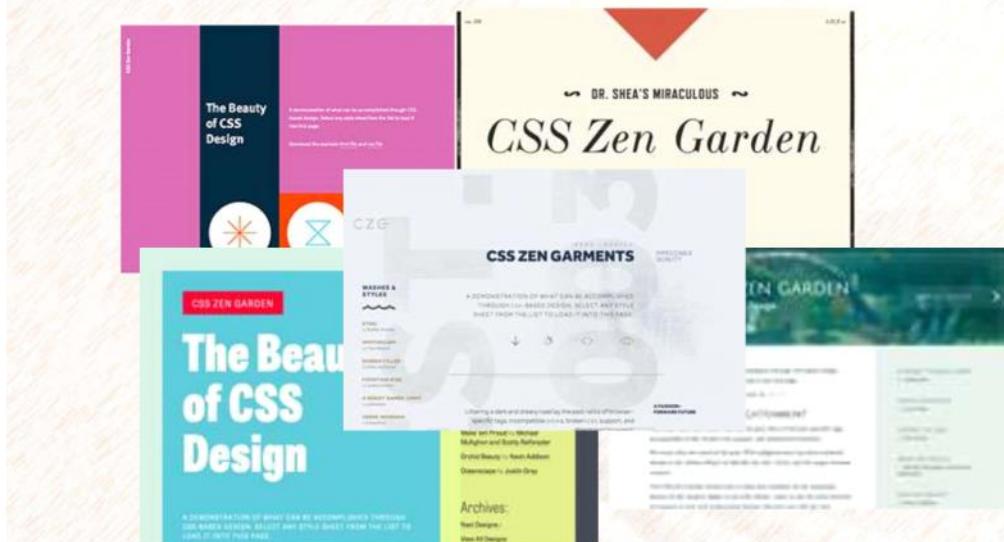
Formato en los documentos HTML



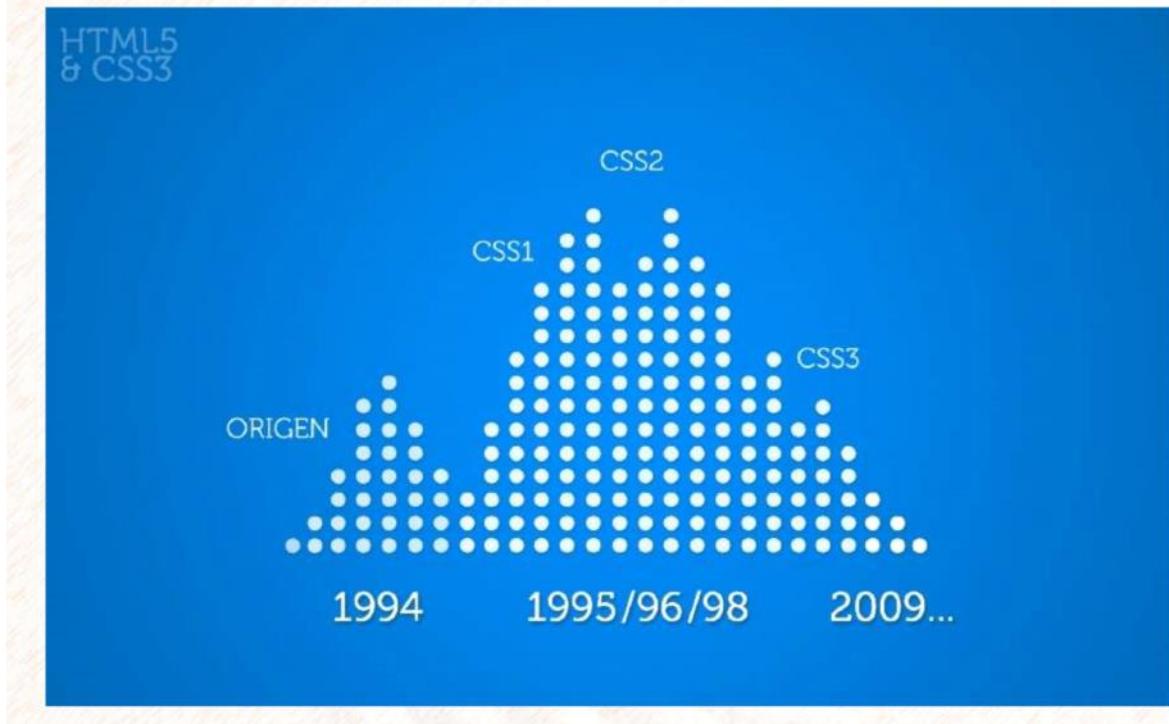
CSS Zen Garden

<http://csszengarden.com/>

Una web para mostrar como cambia radicalmente el aspecto de una Web cambiando únicamente los estilos CSS que se aplican



Estilos



Estilos

Inicialmente

HTML es un lenguaje descriptivo de la estructura del documento, en el que las etiquetas no dan formato al contenido

Más adelante,

HTML 3.2 incorpora etiquetas como `` y atributos de color que contradicen el planteamiento inicial

Como reacción

En HTML 4.0, el W3C incorpora las Hojas de Estilos en Cascada, (*Cascading Style Sheets, CSS*)

Regresamos
así al inicio

Todos los elementos relativos al formato se retiran del cuerpo del documento HTML y se incorporan a los estilos, que en el mejor de los casos pasan a un fichero independiente; CSS



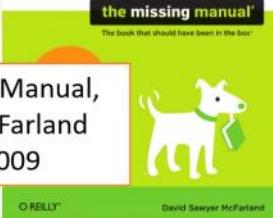
Más información

lunes, 19 de febrero de 2018 11:39

Más información

"The Missing Manual series is simply the most intelligent and visible series of guidesbooks..."
—KEVIN KELLY, CO-FOUNDER OF WIRED

CSS



CSS: The Missing Manual,
David Sawyer McFarland
O'Reilly Media, 2009



Handcrafted CSS: More
Bulletproof Web Design,
Dan Cederholm
New Riders, 2009

Sintaxis de los estilos

Selector

h1

Declaración

{color:blue; font-size:12px}

Declaración

Propiedad Valor

Propiedad Valor

Propiedad color

Valor:

red, purple, fuchsia, blue, navy, teal, aqua
green, lime, olive, yellow, maroon,
silver, gray, white, black,

Propiedad font-size

Valor: ... 12px ... 16px (valor por defecto) ...

Nombre	Color
White	
Black	
Red	
Lime	
Blue	
Yellow	
Silver	
Gray	
Maroon	
Purple	
Fuchsia	
Green	
Olive	
Navy	
Teal	
Aqua	

Comentarios

texto del comentario

/* Comentario */

Barra *

* Barra

/* Esto es un comentario en CSS */

(No confundir con HTML)

<!--Esto es un comentario en HTML-->

Ubicación de los estilos

Se pueden incorporar los estilos en tres posiciones

Estilos en línea

Junto al elemento al que afectan, con el atributo **style**=“declaración”

Estilos embebidos o internos

En la cabecera del documento **<style> lista de estilos </style>**

```
<style>
h1 {color:blue; font-size:12px}
</style>
```

Estilos externos

En un fichero de extensión .css incluye la **lista de estilos**

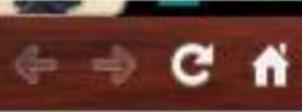
```
h1 {color:blue; font-size:12px}
```

La cabecera incluirá **<link rel="stylesheet" href="mystyle.css">**

Ejemplos de estilos

Estilo como atributo

```
<body>
<h1 style=
  "color:red;
  font-size:14px;">
Título
</h1>
</body>
```



Título

Estilo embebido

```
<head>
<style>
h1 {
  color: red;
  font-size: 14px;
}
</style>
</head>

<body>
<h1> Título </h1>
</body>
```



El resultado es el mismo en los tres casos

Fichero de estilo

```
h1 {
  color: red;
  font-size: 14px;
}
```

Fichero HTML

```
<body>
<h1> Título </h1>
</body>
```

AC3

AC5

Importación de estilos: @import

@import

directiva CSS que permite incorporar reglas desde otras hojas de estilo (como link a nivel HTML)

@import url [list-of-media-queries]

```
@import url("tables.css");
```

se pueden añadir *media queries* que filtraran la importación

```
@import url("shetland.css") screen and  
(max-device-width: 480px);
```

@import debe preceder a todas las demás reglas de estilo, excepto @charset.

Directivas CSS: @ condicionales

@media → permite definir un conjunto de reglas anidadas que sólo se aplicaran en función la *media query* indicada

```
@media screen and (max-device-width: 480px)  
{.column {float: none;} ... }
```

@page → permite modificar algunas propiedades CSS cuando se imprime el documento

@supports → permite comprobar si una condición es válida en un navegador, en cuyo caso la directiva devuelve true y ejecuta la serie de reglas anidadas indicada

@document → restringe el uso de las reglas en función de la url del documento (experimental)

Otras directivas CSS @

@charset → especifica el conjunto de caracteres utilizado en la hoja de estilos:
debe ser siempre el primer elemento que aparece en ella)

```
@charset "UTF-8";
```

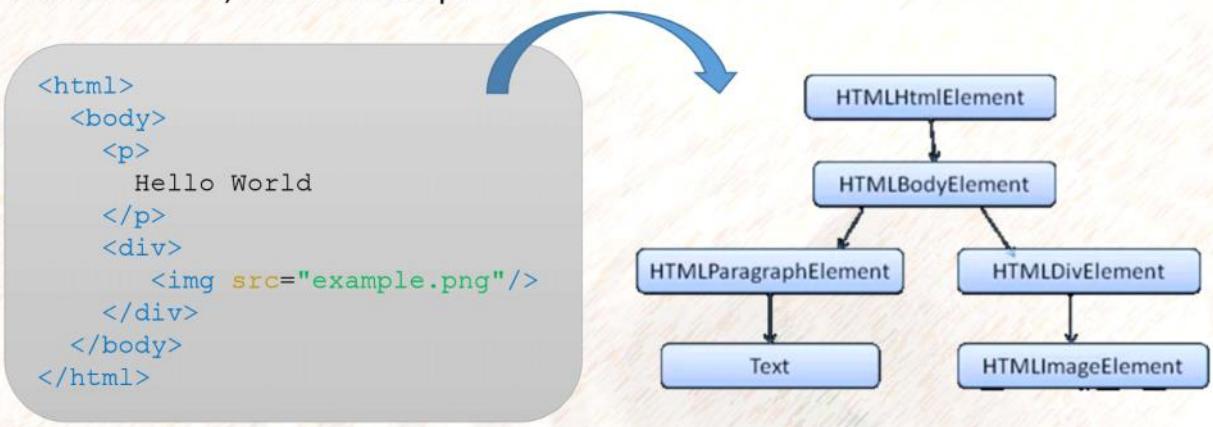
Relativas a elementos concretos (fuentes, animaciones)

@font-face → permite especificar fuentes online para
visualizar en las páginas web.

@keyframes → permite controlar los pasos intermedios en una
secuencia de animación CSS (experimental)

DOM (*Document Object Model*)

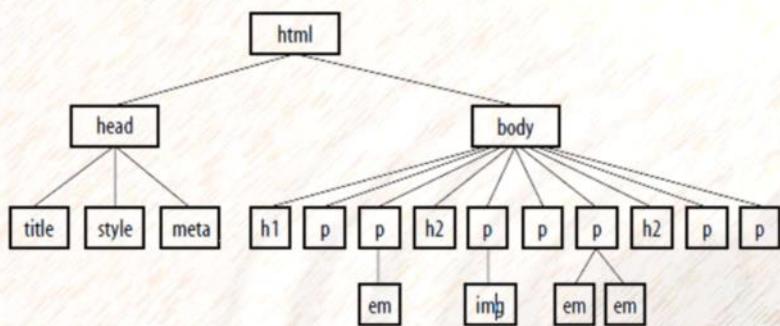
DOM (**modelo de objetos del documento**) es la representación estándar de los objetos del documento HTML y la interfaz (API) de los elementos HTML para el mundo exterior, como JavaScript



El árbol de salida ("árbol de análisis") de un motor de renderización está formado por nodos: elementos y atributos del DOM.

Relaciones entre elementos del DOM

Desde el punto de vista de CSS, la estructura del documento es una versión simplificada del DOM en la que se definen las relaciones entre los elementos como si fueran **relaciones de parentesco**



El DOM, interpretado como un "árbol familiar"

- padres (*parents*)
- hijos (*child*)
- hermanos (*siblings*)
- ascendentes (*ancestors*)
- descendientes (*descendants*)

Múltiples estilos: herencia

Existen propiedades que automáticamente son heredadas en los elementos hijos. Tal es el caso de las propiedades que afectan al texto (**font-family, color...**): si se definen en <body> se aplicaran a todos los elementos contenidos en él.

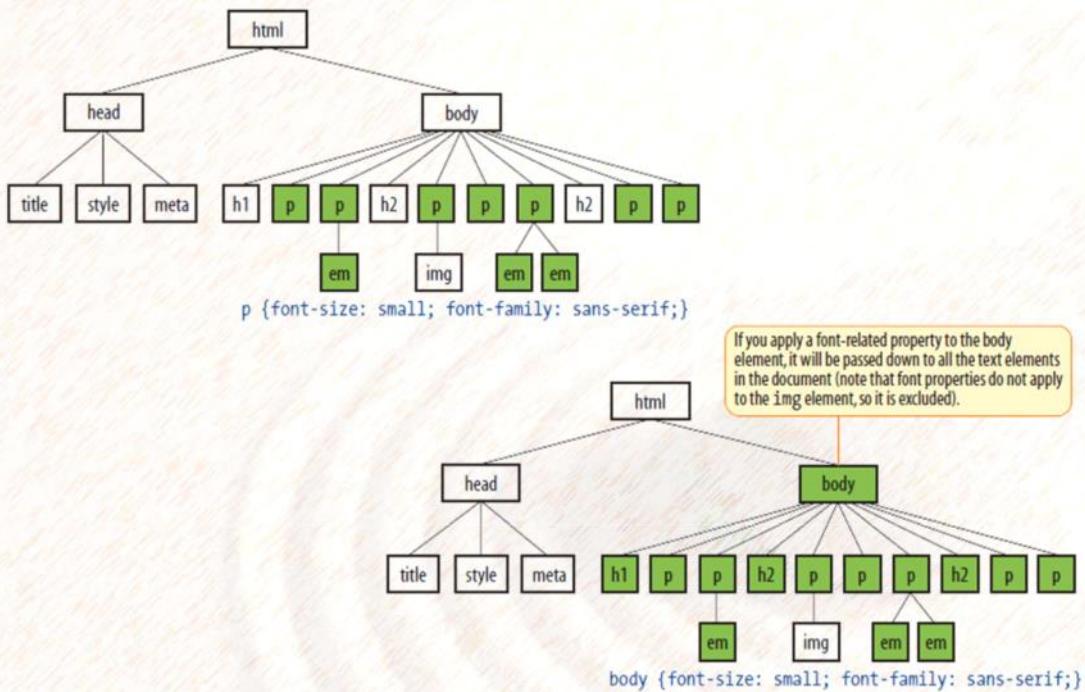
Existen otras propiedades que no son heredadas automáticamente. Tal es el caso de las que afectan a los contenedores del texto, i.e. las "cajas" (**background-color, margin, padding, border...**)

Para la mayoría de estos casos, se puede forzar la herencia de los valores del elemento padre indicando como valor el término *inherit*

```
body {  
    font-family: Arial, Verdana, sans-serif;  
    color: #665544;  
    padding: 10px;}  
  
.page {  
    border: 1px solid #665544;  
    background-color: #efefef;  
    padding: inherit;}
```

La clase page hereda automáticamente **font-family** y **color** y además forzamos que herede **padding**

Ejemplo de herencia



Referencia: Selectores

Selector

Declaración

h1

{color:blue; font-size:12px}

Hace referencia a la parte del documento a la que se aplicara el correspondiente estilo

- Referencia por etiqueta:
 - el selector es el nombre de una etiquete HTLM:
h1, p, span, div...
 - el estilo se aplica siempre que aparezca dicha etiqueta
- Referencia individual (por el atributo id)
- Referencia por clase (por el atributo class)

Referencias individuales

Selector

#estilo1

{color:blue; font-size:12px}

- Se define con el selector `#nombre_estilo`, para especificar un estilo exclusivo para un único elemento
- El estilo se aplica la única vez que aparece el atributo id correspondiente:
`<id="nombre_estilo">`

Las etiquetas `<div>` y ``, que define un grupo de elementos en bloque o en línea sin modificarlos en nada, son muchas veces el paso previo para aplicarles un estilo:

```
<span id="nombre_estilo">...</span>
```

Referencias de clase

.estilo1

{color:blue; font-size:12px}

Se define con el selector `.nombre_estilo`

El estilo se aplica cada vez que aparece el atributo **class**:

```
<class="nombre_estilo">
```

Nuevamente es muy habitual su uso con las etiquetas **<div>** y ****,

```
.termino_info {color:blue; font-size:12px}
```

```
<span class="termino_info">OOP</span>
```

El selector puede ser una clase asociada a una etiqueta concreta

```
p.párrafo_resumen {color:blue; font-size:12px}
```

Ejemplos del uso de id y class

Estilos embebidos

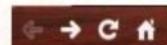
```
<head>
<style>
#titulo{
    color: red;
    font-size: 28px;
}
.autores{
    color: blue;
    font-size: 14px;
}
</style>
</head>
```

Fichero de estilo

```
#titulo{
    color: red;
    font-size: 28px;
}
.autores{
    color: blue;
    font-size: 14px;
}
```

Cuerpo del Fichero HTML

```
<body>
<h1 id="titulo"> Titulo </h1>
<p class="autores">Primer Autor</p>
<p>Biografia, Obras ...</p>
<p class="autores">Segundo Autor</p>
<p>Biografia, Obras ...</p>
</body>
```



Título

Primer Autor
Biografia, Obras ...
Segundo Autor
Biografia, Obras ...

Selectores agrupados

lunes, 19 de febrero de 2018 16:55

Selectores agrupados

Selectores

h1, h2, p

Declaraciones

**{color:blue;
font-size:12px}**

Para aplicar una o varias declaraciones a un conjunto de selectores, es suficiente indicarlos uno tras otro, separados mediante comas

Con frecuencia, un mismo selector se utiliza agrupado o por separado

```
h1, h2, p, em, img {border: 1px solid blue; }  
h1 {font-size: 1.5em}  
h2 {font-size: 1.3em}
```

Múltiples estilos. Cascadas

Lectura

1. Estilos “por defecto” del agente de usuario del navegador
2. Estilos definidos por el usuario a nivel del navegador. E.g. en Ópera, Ver → Estilo → Administrar modos... → Pantalla → Mi hoja de estilo.
3. Estilos aplicados por un autor mediante CSS externa
4. Estilos embebidos en la página web
5. Estilos “en-línea”, aplicados a una etiqueta.

Precedencia

- En caso de selectores idénticos, el último leído tiene precedencia
- Los estilos más específicos tienen precedencia sobre los más generales
- la propiedad **!important**: se impone a las reglas anteriores, haciendo que el estilo al que se aplica tenga precedencia

Ejemplo de múltiples estilos

```
<body>
  <h1>Tipos de elfos</h1>
  <p id="intro">Existen <i>docenas</i> de diferentes variedades de
  <b>elfos</b>.</p>
  <p>Entre otros pueden distinguirse por su ubicación: de los bosques, de
  los ríos, de las montañas...</p>
</body>

  * {
    font-family: Arial, Verdana, sans-serif;}
  h1 {
    font-family: "Courier New", monospace;}
  i {
    color: green;}
  i {
    color: red;}
  b {
    color: pink;}
  p b {
    color: blue !important;}
  p b {
    color: violet;}
  p#intro {
    font-size: 100%;}
  p {
    font-size: 75%;}
```

Tipos de elfos

Existen *docenas* de diferentes variedades de **elfos**.

Entre otros pueden distinguirse por su ubicación: de los bosques, de los ríos, de las montañas...

Especificidad

lunes, 19 de febrero de 2018 19:36

a | b | c | d

- a** es igual a 1 si la declaración está definida como estilos en línea
- b** es igual al número de id's
- c** es igual al número de otros atributos y pseudo-clases
- d** es igual al número de elementos y pseudo-elementos

0 | 0 | 0 | 1
elemento {}
::pseudo-elemento {}

0 | 0 | 1 | 0
.clase {}
[atributo] {}
:pseudo-clase {}

0 | 1 | 0 | 0
#identificador {}

1 | 0 | 0 | 0
style="color: lightcoral"

0 | 0 | 0 | 0
>, +, *, ~, not()
@media

Se mejora el rendimiento utilizando estilos con la menor especificidad posible y cargando los estilos según el mismo criterio

0010 < 0012
0 | 0 | 1 | 0 > 0 | 0 | 0 | 13

<https://specificity.keegan.st/>

Specificity Calculator

`li:first-child h2 .title`

0 Inline styles	0 IDs	2 Classes, attributes and pseudo-classes	2 Elements and pseudo-elements
--------------------	----------	---	-----------------------------------

+ Duplicate

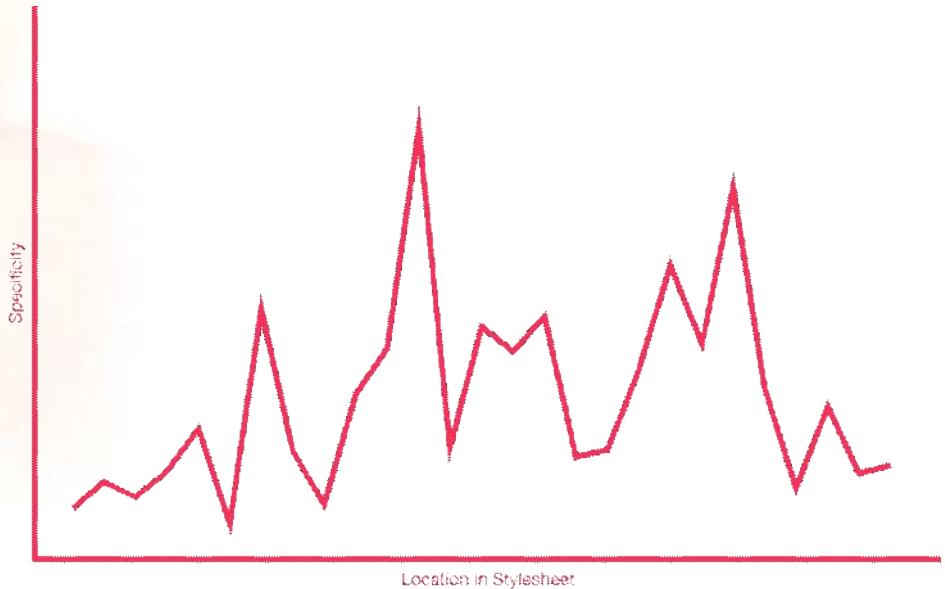
`#nav .selected > a:hover`

0 Inline styles	1 IDs	2 Classes, attributes and pseudo-classes	1 Elements and pseudo-elements
--------------------	----------	---	-----------------------------------

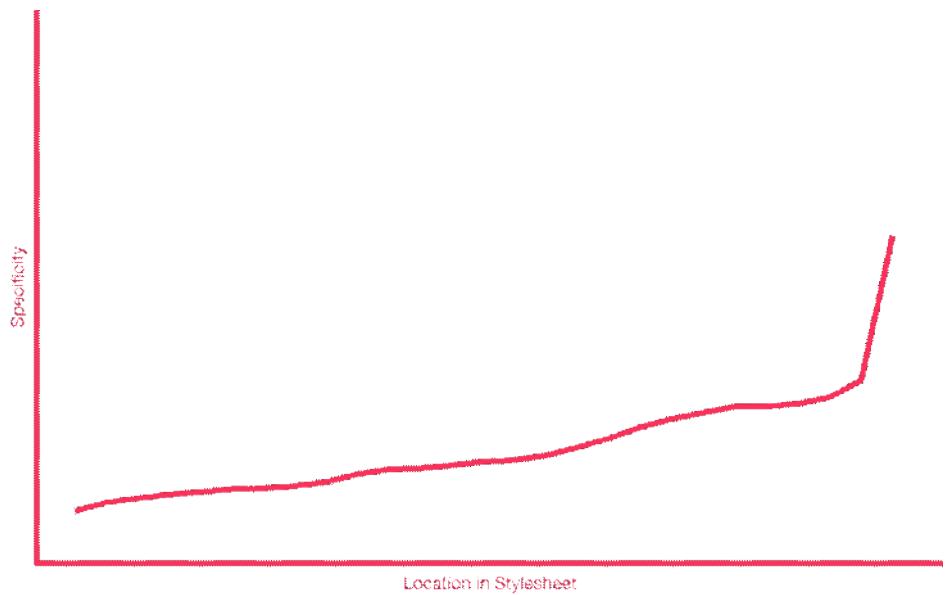
+ Duplicate

Gráficos de especificidad. Arquitecturas

lunes, 19 de febrero de 2018 19:46



Distribución de los niveles de especificidad en distintos archivos



Reordenación de los estilos en diversos archivos
de acuerdo con una arquitectura CSS

Elementos de diseño avanzados

lunes, 19 de febrero de 2018 11:37

1. Selectores, pseudoclases y pseudoelementos
 - a. Selectores contextuales
 - b. Selector de atributo
2. Nuevos selectores en CSS3
 - Pseudoclases relativas a la posición en el DOM
 - Selector universal
 - Pseudoclase :not()
 - Selectores contextuales: Selector ~
3. Variables en CSS3. Funciones: calc()
4. Modelo de caja. Concepto.
 - a. Modelos disponibles: content-box / border-box
5. Inicialización. Reseteo y normalización.
6. Soporte y prefijos

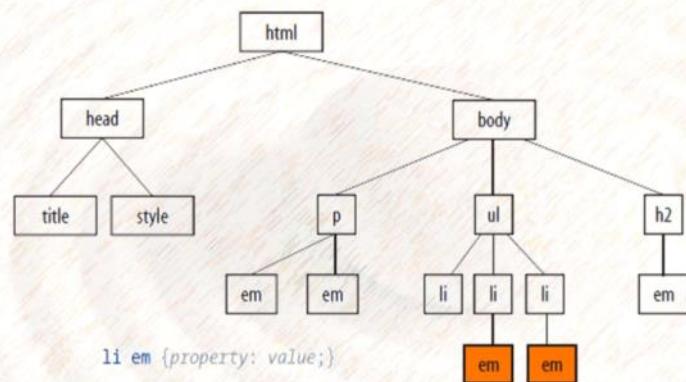


Selector de "descendientes"

Selector “espacio” (Descendant Selector) referencia elementos que descienden de otro, sin que necesariamente tengan que ser hijos

Por ejemplo, `div p {}` referenciará cada elemento `<p>` que está en un elemento `<div>`. Sólo se ve afectado el segundo selector, mientras que el primero le da el contexto

Anteriormente denominados **“selectores contextuales”**, HTML5 se refiere a este tipo como **“combinadores”**



Selectores contextuales / combinadores

- **Selector “espacio” (*Descendant Selector*)**
- **Selector > (*Child Selector*)** Este selector referencia al elemento de la derecha cuando tiene el elemento de la izquierda como parente.
Por ejemplo, `div > p {}` referenciará cada elemento `<p>` que es hijo de un elemento `<div>`.
- **Selector + (*Adjacent Sibling Selector*)** Este selector referencia elementos que son hermanos. La referencia apuntará al elemento de la derecha cuando es inmediatamente precedido por el de la izquierda.
Por ejemplo, `span + p {}` afectará a los elementos `<p>` que son hermanos y están ubicados luego de un elemento ``.
- **Selector ~ (*General Sibling Selector*)** Este selector es similar al anterior, pero en este caso el elemento de la derecha no tiene que estar ubicado inmediatamente después del de la izquierda.



Selector de Atributo



palabraclave[atributo=valor]

podemos utilizar otros atributos además de id y class para asignar estilos a elementos que tiene un atributo particular, incluso con un valor específico.

```
p[name="texto"] {color:blue; font-size:12px}
```

referenciará cada elemento <p> con un atributo llamado name y el valor "texto".

Esta referencia puede ser aún más específica incorporando símbolos

- ^= elementos que comienzan con el valor provisto
- \$= elementos que terminan con ese valor
- *= elementos que contienen el texto provisto

```
p[name^="texto"] {color:blue; font-size:12px}
```

se aplica a elementos <p> con un atributo name cuyo valor comienza por "texto".

Pseudoclases

Son clases definidas dinámicamente, sin un atributo específico que las identifique

- en función de acciones del usuario, como la posición del cursor (:link, :visited, :active, :hover, :focus)
- en función de la posición de los elementos y sus relaciones con otros (i.e su posición en el DOM)

[selector]:pseudoclase

Este aspecto dinámico les da un valor condicional

- Si se dan determinadas circunstancias se aplica un estilo
- Si no se dan no se aplica

Hiperenlaces y Pseudoclases

Las **pseudo-classes** permiten responder a **eventos** ligados a una etiqueta con la aplicación de estilos particulares



El estilo varia en función de elemento: se introduce dinamismo sin necesidad de scripts

:link

Para el identificador **a** se han definido dos de estas **pseudo-clases**, que permiten modificar el comportamiento que presentan los hiperenlaces por defecto: su cambio de color una vez que han sido visitados

:visited

a:link {propiedad: valor}

Enlace no visitado

a:visited {propiedad: valor}

Enlace visitado

Otras Pseudoclases

:focus

Elemento seleccionado
(con el teclado)

:hover

Elemento sobre el que se sitúa el ratón
u otro dispositivo apuntador

:active

Enlace clicado.
Si es un hiperenlace <a>,
a continuación cambiara a visitado

Muy habituales con hiperenlaces

a:hover {propiedad: valor}

:lang(<l>)

El estilo se aplicara solo si el elemento
afectado esta definido con el idioma <l>

Pseudoelementos primera letra y línea

Los **pseudo-elementos** permiten aplicar un estilo a parte de un elemento

::first-letter

Afectará a la primera letra de un elemento

::first-line

Afectará a la primera línea de un elemento

Los orígenes de Internet se remontan a la década de 1960, dentro de ARPA (hoy DARPA), como respuesta a la necesidad de esta organización de buscar mejores maneras de usar los computadores de ese entonces, pero enfrentados al problema de que los principales investigadores y laboratorios deseaban tener sus propios computadores, lo que no sólo era más costoso, sino que provocaba una duplicación de esfuerzos y recursos

```
p.intro:first-line {font-weight: bold;}  
p.intro:first-letter {font-size: 200%;}
```

Pseudelementos *before* y *after*

Son estilos que sirven para definir una parte adicional del elemento al que se aplican, al principio o al final de este.

[selector]::pseudelemento

::before

Permiten añadir contenido antes o después del elemento al que se aplican

::after

Para ello es necesario utilizar la propiedad **content** (incluso si su valor es "")

```
h1::before {  
    content: url(smiley.gif);  
}
```

Content

content

Permite añadir contenido desde la hoja de estilo

content: **normal** | **none** | counter | attr | <**string**> | open-quote | close-quote | no-open-quote | no-close-quote | url | initial | inherit;

counter → añade un contador

attr (<atributo>) → Recoge el valor del atributo indicado del selector

url (url) → Incorpora un recurso (img, video audio)

open-quote (no-open-quote)
close-quote (no-close-quote) → Abrimos cerramos comillas en el selector

Content: las comillas

content open-quote (no-open-quote)
 close-quote (no-close-quote)

Es frecuente su uso ligado a los pseudoelementos ::after y ::before

Existe además la posibilidad de definir el tipo de comillas

quotes

modificar el
aspecto de las
comillas, e.g. en
<q> o al añadirlas
con content

"	double quote	"
'	single quote	'
<	single, left angle quote	‹
>	single, right angle quote	›
«	double, left angle quote	«
»	double, right angle quote	»
‘	left quote (single high-6)	‘
’	right quote (single high-9)	’
“	left quote (double high-6)	“
”	right quote (double high-9)	”
„	double quote (double low-9)	„

Novedades en CSS3

- Pseudoclases relativas a la posición en el DOM
 - pseudoclase :nth-child()
 - Pseudoclase :first-child
 - Pseudoclase :last-child
 - Pseudoclase :only-child
- Selector universal y pseudoclase :not()
- Selectores contextuales
(con operadores)
 - Selector ~
(amplia los ya existentes
Selector > / Selector +)



Pseudoclases jerárquicas (1)



- **:nth-child(n)** – referencia a un hijo de índice (n), o los hijos pares o impares (odd/even)

```
p:nth-child(2) {  
    background: #ff0000;  
}
```

Se aplica a cualquier párrafo que sea segundo entre los párrafos hijos

N puede ser un número, una palabra clave (como odd/even) o una fórmula tipo $a n + b$ donde a representa un ciclo, n es un contador a partir de 0 y b es un valor de desplazamiento

```
p:nth-child(3n+0) {  
    background: #ff0000;  
}
```

Se aplica a cualquier párrafo cuyo índice como hijo sea múltiplo de 3

Pseudoclases jerárquicas (2)



- **:nth-of-type(n)** -- referencia a un hijo específico (n), del mismo tipo que el selector

```
p:nth-child(2) {  
    background: #ff0000;  
}
```

Se aplica a cualquier párrafo que sea segundo hijo, sea el primero del tipo que sea

N puede ser un número, una palabra clave (como odd/even) o una fórmula

- **:first-child** – referencia al primer hijo, similar a :nth-child(1)
- **:last-child** – referencia al último hijo
- **:only-child** – referencia a un elemento que es el único hijo disponible de un mismo elemento padre

Ejemplo

The screenshot shows a web browser window titled "Estilos en CSS3" with the URL "localhost/Curso_Web/HTML/08.Estilos/Sample_CSS3.html". The page content is a heading "Ejemplo de pseudoclases jerarquicas" followed by three paragraphs: "Párrafo 1", "Párrafo 2", and "Párrafo 3". A blue box highlights the CSS rule "p:nth-child(2) {color: maroon;}" which styles "Párrafo 2". A grey box highlights the CSS rule "p:nth-of-type(2) {color: blue;}" which styles "Párrafo 2". Below the browser window is a code editor showing the HTML and CSS source code:

```
13
14 <style type="text/css">
15   p:nth-child(2) {color: maroon;}
16   p:nth-of-type(2) {color: blue;}
17 </style>
18 </head>
19
20 <body>
21   <h1>Ejemplo de pseudoclases jerarquicas</h1>
22   <p title="Párrafo 1">Párrafo 1</p>
23   <p title="Párrafo 2">Párrafo 2</p>
24   <p title="Párrafo 3">Párrafo 3</p>
25 </body>
26 </html>
```

Selector universal y pseudoclase :not()



*

{color:black; font-size:16px}

el selector universal * asigna el mismo estilo a todos los elementos del documento.

:not (p)

{color:black; font-size:16px}

La pseudoclase :not() hace referencia a todos los elementos excepto el declarado entre paréntesis.

Ambos selectores representan el conjunto de todos los elementos en el cuerpo del documento o todos menos las excepciones indicadas, y son útiles cuando necesitamos establecer ciertas reglas básicas.

Variables y funciones en CSS

Lunes, 19 de febrero de 2018 21:07

Variables

Las variables permiten asignar cualquier valor a una propiedad cuyo nombre podemos elegir libremente.

```
:root {  
    --color-principal: #06c;  
}
```

se definen dentro de la etiqueta :root

Esos valores se reutilizan en cualquier otra propiedad gracias a la función var()

```
#foo h1 {  
    color: var(--color-principal);  
}
```

<http://librosweb.es/tutorial/como-usar-las-nuevas-variables-css/>

Pueden utilizarse expresiones complejas como

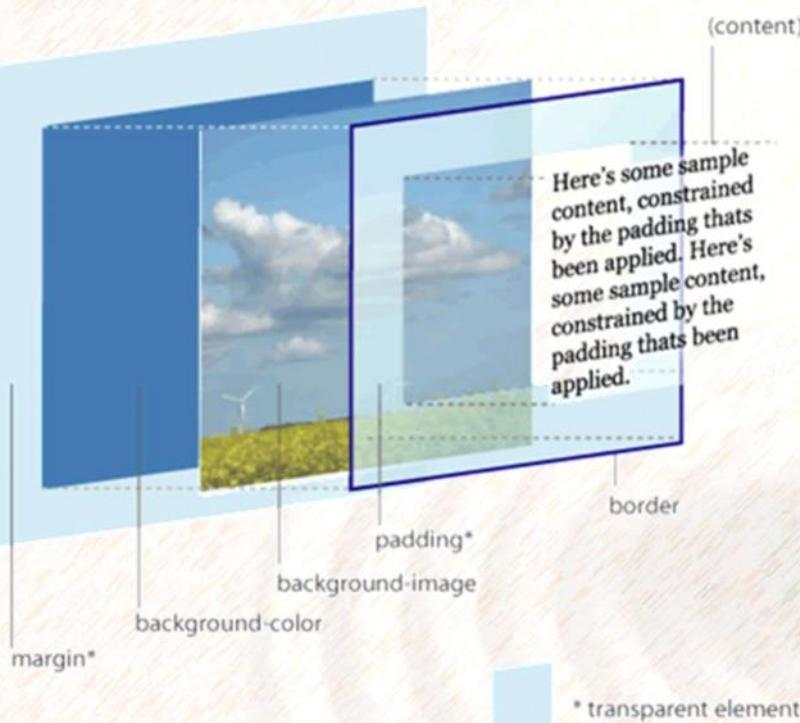
```
--foo: if(x > 5) this.width = 10;
```

Función calc()

La función calc() es muy útil para realizar cálculos al definir los valores de las propiedades CSS. Todos los navegadores modernos la soportan sin problemas y se puede combinar con las variables CSS. Ejemplo:

```
.foo {  
    --separacion: 20;  
    margin-top: calc(var(--separacion) * 1px);  
}
```

Modelo de cajas (Box Model)



- Superposición y transparencias
- Interacción entre elementos
- definición del tamaño

Box-sizing: content-box

domingo, 14 de mayo de 2017 10:10

Por defecto cualquier elemento en HTML adapta su tamaño a su contenido.

En los elementos “en bloque” el ancho es el del contenedor y el alto se ajusta al número de líneas escritas

En los div, por defecto la altura es 0 si no contienen nada

También podemos definir el tamaño de forma explícita

box-sizing

content-box

Define como se contabilizan la altura y la anchura de los elementos de tipo bloque, de forma predeterminada. Era la única forma disponible antes de CSS3

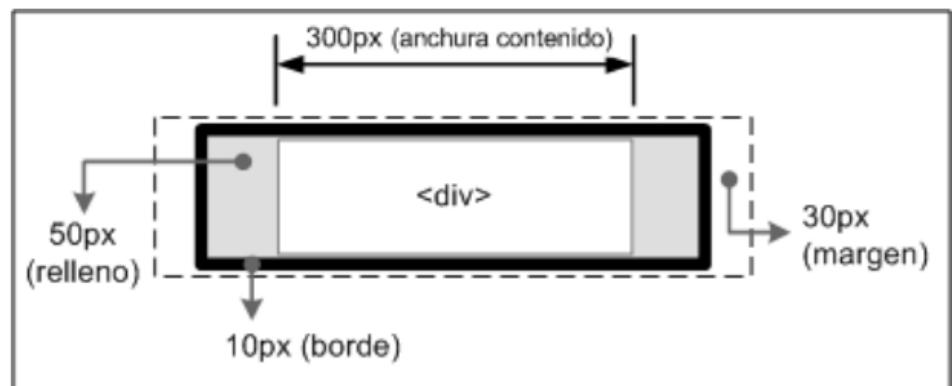
- Auto (valor por defecto)
- valor absoluto (pixeles, cm,...)
- valor relativo (%)

height

son solo las medidas en el contenido,

El tamaño que ocupa un elemento corresponde a la suma de margen + borde + relleno (padding) + tamaño indicado

En la figura vemos un div que realmente ocupa 480px



Box-sizing: border-box

lunes, 19 de febrero de 2018 18:46

box-sizing

border-box

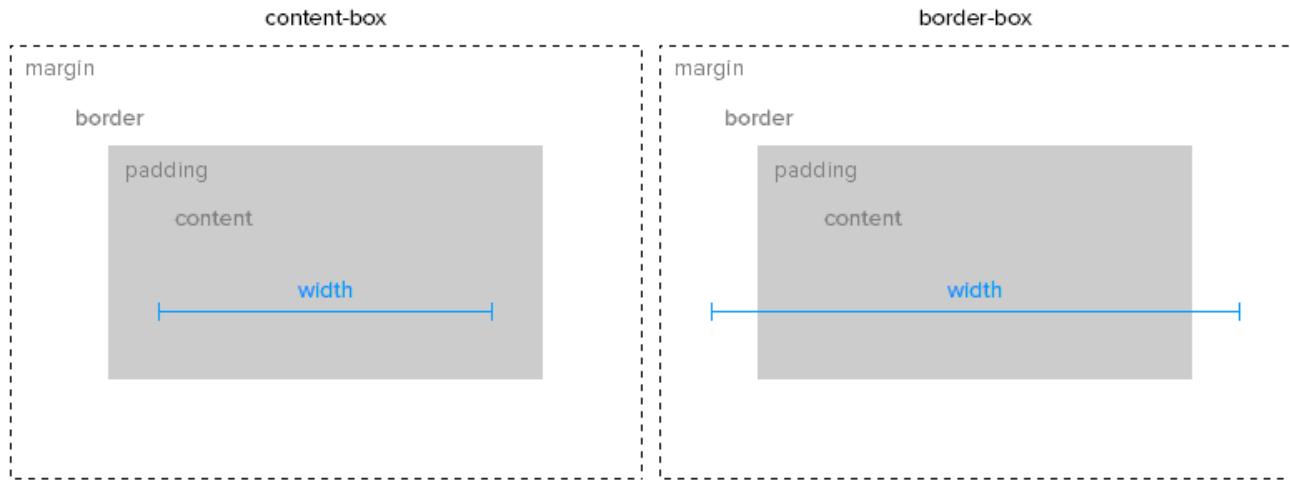
height

width



incluyen el borde y el relleno (padding), que por tanto se descuenten del tamaño indicado, en lugar de sumarse,

El tamaño final sería margen + tamaño indicado.



El nuevo modelo supone un gran cambio:

De leer el ancho y alto en función del contenido (modelo antiguo)
pasamos a hacerlo en función de borde + padding + content

Cajas: límites de tamaño

min-height,
max-height

min-width,
max-width

Valores mínimos y máximos del ancho y alto de un elemento bloque

longitud (px, in, cm ...)
porcentaje (%)

Definimos el comportamiento del contenido cuando las dimensiones de la caja no están en auto, y por tanto no se adaptan automáticamente a él

overflow

hidden
scroll

clip

→ rect(*top, right, bottom, left*)

RESULT

Fender Stratocaster

The Fender Stratocaster or "Strat" is one of the most popular electric guitars of all time, and its design has been copied by many guitar makers.

Gibson Les Paul

The Gibson Les Paul is a solid body electric guitar that was first sold in 1952. The Les Paul was designed by

para un elemento de posición definida de forma absoluta, fija (clips) el espacio que ocupa, sin que se pueda salir de el

Reseteo y normalización

Junes, 19 de febrero de 2018 13:45

Suele incluirse un CSS base, que consiste en hojas de normalización o de *reset*,

Una de sus funciones principales es aplicar de forma genérica, el Box-sizing: border-box
Con frecuencia en asociado al selector genérico (*) y añadiendo las pseudo clases ::after y ::before.

```
*,
*::before, *::after,
*:before, *:after {
  box-sizing : border-box;
}
```

Otros elementos reseteados con frecuencia

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  font: inherit;
  vertical-align: baseline;
}
```

Prácticamente todas las etiquetas quedan "a cero"

```
body {
  line-height: 1;
}
ol, ul {
  list-style: none;
}
table {
  border-collapse: collapse;
  border-spacing: 0;
}
```

Tamaño de línea 1

Listas sin esilos

Tablas sin bordes

```
blockquote, q {
  quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
  content: '';
  content: none;
}
```

Citas sin comillas

article, aside, details, figcaption, figure,

Solo para navegadores antiguos,

```
article, aside, details, figcaption, figure,  
footer, header, hgroup, menu, nav, section {  
    display: block;  
}
```

Solo para navegadores antiguos,
se establece el carácter de
bloque para los componentes de
este tipo definidos en HTML5

Ejemplo más completo: <http://html5doctor.com/html-5-resetstylesheet/>

Una alternativa más potente es el uso
de herramientas de normalización

<https://necolas.github.io/normalize.css/>



Normalize.css

**A modern, HTML5-ready alternative
to CSS resets**

Soporte en los navegadores

Existen estilos que están aún en proceso de incorporación al estándar pero ya son soportados por los navegadores.

En ese caso es necesario indicarlo en los estilos, con los prefijos de los navegadores afectados

-webkit-	Safari, Chrome y Opera 15 o >
-moz-	Firefox
-o-	Opera < 15
-ms-	Internet Explorer

Para asegurarnos un valor flex = 1 en todos los navegadores que lo soporten

Chorme 29.0 (21.0 -webkit-)
IE 11.0 (10.0 -ms-)
Firefox 28.0 (18.0 -moz-)
Safari 6.1 -webkit-
Opera 12.10

```
#main div {  
    -webkit-flex: 1;  
    -moz-flex: 1;  
    -ms-flex: 1;  
    flex: 1;  
}
```