|  |  |
| --- | --- |
| guia - instructivo |  |
| **CREACIÓN DE PAGINAS WEB** |  |
| EN ESTA EDICIÓN  PÁG. 2  Introducción y explicación del marco teórico para la creación de páginas web.  PÁG. 7  Marco teórico –práctico de las estructuras de caja propias de las páginas web  PÁG. 104  Practica dirigida de creación de páginas web |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | ¿que es una pagina web?Se conoce como página Web, a la página electrónica o página digital a un documento digital de carácter multimediático (es decir, capaz de incluir audio, video, texto y sus combinaciones), adaptado a los estándares de la World Wide Web (WWW) y a la que se puede acceder a través de un navegador Web y una conexión activa a Internet. Se trata del formato básico de contenidos en la red. En Internet existen más de mil millones de páginas Web de diversa índole y diverso contenido, provenientes del mundo entero y en los principales idiomas hablados. Esto **representa el principal archivo de información de la humanidad** que existe actualmente, **almacenado a lo largo de miles de servidores a lo largo del planeta**, a los que es posible acceder velozmente **gracias a un sistema de protocolos de comunicación (HTTP).** |  |
|  | ¿como se crean? Las **páginas web** desde sus inicios son **producto de la combinación de HTML, CSS y Javascript**. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto que (como veremos más adelante) es extremadamente significativo.  Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones.  Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio.  Para trabajar y crear sitios webs y aplicaciones con HTML5, necesitamos saber primero cómo esa estructura es construida.  **Nota**: En el presente tutorial no tocaremos el lenguaje de programación JAVASCRIP. |  |
|  |  |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Visual Studio Code** es un [editor de código fuente](https://youtu.be/6aAUznSlHAY?t=1118) que permite trabajar con diversos lenguajes de programación, admite gestionar tus propios atajos de teclado y refactorizar (es decir cambiar el comportamiento, más no) el código. Es gratuito, de código abierto y nos proporciona una utilidad para descargar y gestionar extensiones con las que podemos personalizar y potenciar esta herramienta.  Las **extensiones de Visual Studio Code** nos otorgan infinidad de opciones, como colorear tabulaciones, etiquetas o recomendaciones de autocompletado. También hay extensiones que nos ayudan con el lenguaje de programación que vayamos a usar, como por ejemplo para Python, C / C++, JavaScript, etc.. [Wikipedia](https://es.wikipedia.org/wiki/Visual_Studio_Code).  [**Idiomas**](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=visual+studio+code+idiomas&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQ6BMoAHoECD0QAg)**:**Multilingüe  [**Programado en**](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=visual+studio+code+programado+en&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQ6BMoAHoECEAQAg)**:**[TypeScript](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=TypeScript&stick=H4sIAAAAAAAAAONgVuLSz9U3yDM1yKgoW8TKFVJZkBqcXJRZUAIA8z90HhsAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoAXoECEAQAw), [JavaScript](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=JavaScript&stick=H4sIAAAAAAAAAONgVuLQz9U3MCqwNF_EyuWVWJYYnFyUWVACABxPL_8ZAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoAnoECEAQBA), [CSS](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=CSS&stick=H4sIAAAAAAAAAONgVuLUz9U3MDQtycpYxMrsHBwMAG1TLNUTAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoA3oECEAQBQ)  [**Desarrollador**](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=visual+studio+code+desarrollador&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQ6BMoAHoECEIQAg)**:**[Microsoft](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=Microsoft&stick=H4sIAAAAAAAAAONgVuLQz9U3MCkuM1nEyumbmVyUX5yfVgIAEdSQixgAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoAXoECEIQAw)  [**Licencia**](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=visual+studio+code+licencia&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQ6BMoAHoECEYQAg)**:**Código fuente: Licencia MIT; Binarios: Freeware  [**Tipo de programa**](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=visual+studio+code+tipo+de+programa&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQ6BMoAHoECEgQAg)**:**editor de código fuente  [**Lenguajes de programación**](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=visual+studio+code+lenguajes+de+programaci%C3%B3n&stick=H4sIAAAAAAAAAOPgE-LWz9U3MDQ2qSgvytbSySi30k_Oz8lJTS7JzM_TL85PKylPLEq1KijKTy9KzM3NzEtXyEnMSy9NTE9dxKpblllcmpijUFxSmpKZr5Ccn5KqkJMKks1KLVYAcqDaEpMzD2_OAwBrUzrXbQAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQ6BMoAHoECEMQAg)**:**[JavaScript](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=JavaScript&stick=H4sIAAAAAAAAAOPgE-LWz9U3MDQ2qSgvylbiAHGMCizNtXQyyq30k_NzclKTSzLz8_SL89NKyhOLUq0KivLTixJzczPz0hVyEvPSSxPTUxexcnklliUGJxdlFpTsYGUEAHq0UGdXAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoAXoECEMQAw), [HTML](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=HTML&stick=H4sIAAAAAAAAAOPgE-LWz9U3MDQ2qSgvylbiAHGM040MtHQyyq30k_NzclKTSzLz8_SL89NKyhOLUq0KivLTixJzczPz0hVyEvPSSxPTUxexsniE-PrsYGUEAFMbCx1RAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoAnoECEMQBA), [Java](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=Java&stick=H4sIAAAAAAAAAOPgE-LWz9U3MDQ2qSgvylbiAnHMi5Oy05K0dDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrQqK8tOLEnNzM_PSFXIS89JLE9NTF7GyeCWWJe5gZQQAWACIwlMAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoA3oECEMQBQ), [Hoja de estilos en cascada](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=CSS&stick=H4sIAAAAAAAAAOPgE-LWz9U3MDQ2qSgvylbiBHNMS7IytHQyyq30k_NzclKTSzLz8_SL89NKyhOLUq0KivLTixJzczPz0hVyEvPSSxPTUxexMjsHB-9gZQQAJSW3aVEAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoBHoECEMQBg), [TypeScript](https://www.google.com/search?sxsrf=AOaemvLU5M2QFRRnRn5OsE-qEFtz7_NN9g:1631380819955&q=TypeScript&stick=H4sIAAAAAAAAAOPgE-LWz9U3MDQ2qSgvylbiAnHyTA0yKsq0dDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrQqK8tOLEnNzM_PSFXIS89JLE9NTF7FyhVQWpAYnF2UWlOxgZQQAZi1zUlkAAAA&sa=X&ved=2ahUKEwjc1ZCSt_fyAhVnQjABHQKSCWsQmxMoBXoECEMQBw) |  |  |  |  |
| El **IDE** **QUE** USAREMOS PARA NUESTRO DESARROLLO SERÁ: VISUAL STUDIO CODE | |

|  |  |  |
| --- | --- | --- |
|  |  | ¿que son las extensiones de visual studio code? Extensiones Visual Studio Code, que personalmente uso:  ***Auto Close Tag***  Con este plugin, VSC tiene soporte de etiqueta cerrada para archivos HTML, Handlebars y Razor. Esta extensión está habilitada para otros lenguajes como XML, PHP, Vue, JavaScript, TypeScript, JSX, TSX, etc.  ***Beautify***  Beautify es una de las extensiones más conocidas por todos los desarrolladores aquí. Porque de hecho hace su papel, que es: idente su código: HTML, CSS, Sass, JavaScript y JSON para dejar los códigos dentro del patrón de formato y de manera legible. Este tip es nuevo para muchos desarrolladores.  ***Color Info***  Extensión de código de Visual Studio que proporciona colores CSS de información rápida.  ***HTML Tag Wrapper***  Esta APP puede envolver la selección o selecciones que abarcarían una o varias líneas (funciona tanto con selecciones simples como con selecciones múltiples a la vez). De forma predeterminada, al presionar la barra espaciadora se anulará la selección de las etiquetas de cierre, por lo que puede agregar atributos a las etiquetas de apertura (puede desactivar esta función, ver más abajo). Si no se selecciona nada cuando ejecuta htmltagwrap, agregará una etiqueta de apertura y cierre en la posición del cursor. NOTA: Esta extensión funciona mejor en archivos que utilizan tabulaciones o espacios para la sangría. Puede que no funcione tan bien con pestañas mixtas  ***Icon Fonts***  Esta extensión facilita la inserción de fragmentos de una variedad de fuentes de iconos (ver detalles) en nuestro desarrollo web. Este paquete también está disponible para Atom y Sublime Text.  ***Live Server***  Es una aplicación de nodo simple para el directorio de trabajo y sus subdirectorios. También monitorea los cambios de archivos y, cuando esto sucede, envía un mensaje al navegador a través de una conexión de socket web, indicándole que vuelva a cargar. Para que el cliente admita esto, el servidor inyecta un pequeño fragmento de código JavaScript en cada archivo html solicitado. Este script establece una conexión de socket web y escucha las solicitudes de recarga. Al buscar las hojas de estilo a las que se hace referencia en el DOM y activar el navegador para recuperarlas y analizarlas nuevamente, el archivo CSS se puede actualizar sin la necesidad de volver a cargar toda la página.  ***Spanish Language Pack***  Visual Studio Code se envía de forma predeterminada con el inglés como idioma de visualización y otros idiomas dependen de las extensiones del paquete de idioma disponibles en Marketplace. VS Code detecta el idioma de la interfaz de usuario del sistema operativo y le pedirá que instale el paquete de idioma apropiado, si está disponible en Marketplace. |
|  |  |
|  |

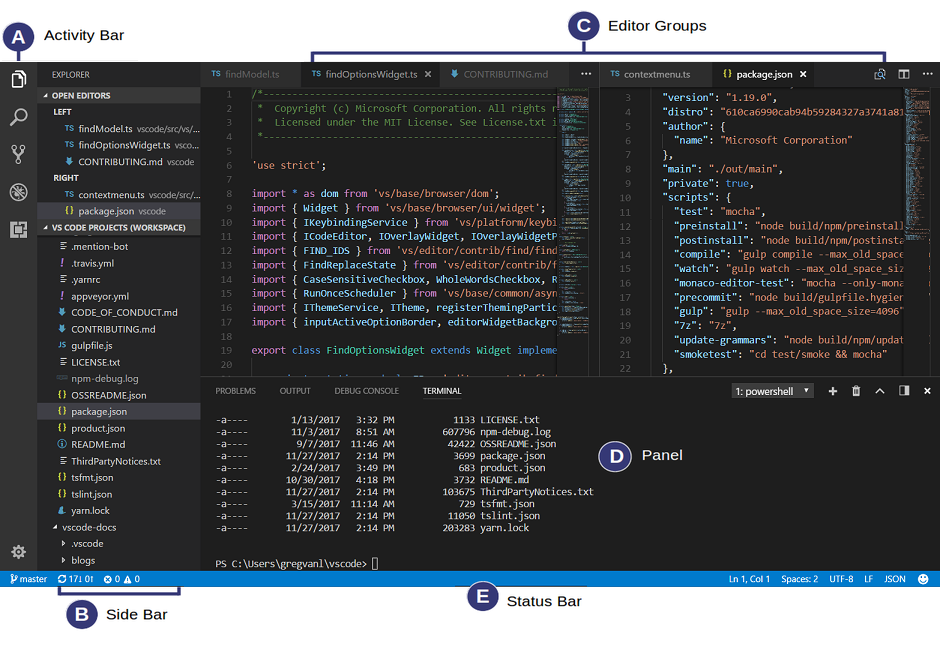
**A partir de este punto iniciamos los alcances**

**teórico-práctico para elaborar páginas web**

|  |
| --- |
| contenidos Iniciaremos nuestro estudio revisando los siguientes contenidos:  Sec 1. INTERFAZ DE USUARIO VISUAL STUDIO CODE  Sec 2. COMO ESCRIBIR CÓDIGO HTML Y CSS MAS RÁPIDO CON EMMET (ZEN CODING)  Sec 3. INTRODUCCIÓN A LA CREACIÓN DE PAGINAS WEB – HTML & CSS  Sec 4. INTRODUCCIÓN AL DISEÑO CON HOJAS DE ESTILO EN CASCADA (CSS)  Sec 5. LISTADO DE PROPIEDADES DE CSS  Sec 6. GUÍA DE REFERENCIA RÁPIDA DE CSS  Sec 7 CASCADA, HERENCIA Y ESPECIFICIDAD  Sec 8 UNIDADES DE MEDIDA Y COLORES  Sec 9 INTRODUCCIÓN AL MODELO DE CAJA DE CSS  Sec 10. RECURSOS PARA EL DISEÑADOR WEB  Sec 11. SECCION DE PRACTICAS  (Los files fuentes se encuentran en el DVD adjunto) |
|
|

**Sec. 1 INTERFAZ DE USUARIO VISUAL STUDIO CODE**

En esencia, Visual Studio Code es un editor de código. Como muchos otros editores de código, VS Code adopta una interfaz de usuario común y un diseño de un explorador a la izquierda, que muestra todos los archivos y carpetas a los que tiene acceso, y un editor a la derecha, que muestra el contenido de los archivos que ha abierto. .



Diseño básico

VS Code viene con un diseño simple e intuitivo que maximiza el espacio provisto para el editor mientras deja un amplio espacio para navegar y acceder al contexto completo de su carpeta o proyecto. La interfaz de usuario se divide en cinco áreas:

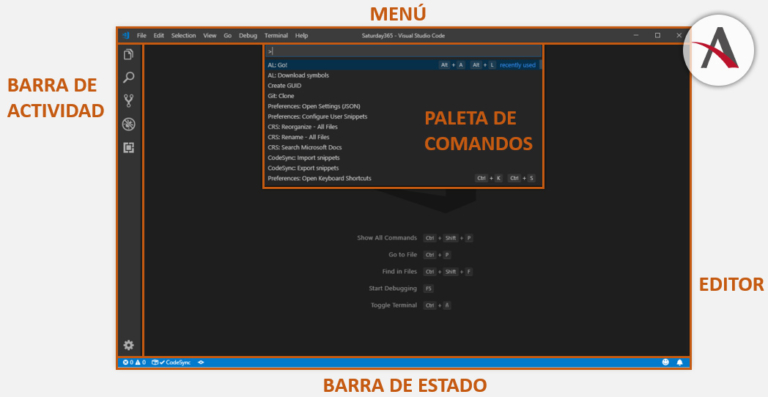
* **Editor** : el área principal para editar sus archivos. Puede abrir tantos editores como desee, uno al lado del otro, vertical y horizontalmente.
* **Barra lateral** : contiene diferentes vistas como el Explorador para ayudarlo mientras trabaja en su proyecto.
* **Barra de estado** : información sobre el proyecto abierto y los archivos que edita.
* **Barra de actividad** : ubicada en el extremo izquierdo, esto le permite cambiar entre vistas y le brinda indicadores adicionales específicos del contexto, como la cantidad de cambios salientes cuando Git está habilitado.
* **Paneles** : puede mostrar diferentes paneles debajo de la región del editor para obtener información de salida o depuración, errores y advertencias, o una terminal integrada. El panel también se puede mover hacia la derecha para obtener más espacio vertical.

Cada vez que inicia VS Code, se abre en el mismo estado en el que estaba cuando lo cerró por última vez. Se conservan la carpeta, el diseño y los archivos abiertos.

Los archivos abiertos en cada editor se muestran con encabezados con pestañas (pestañas) en la parte superior de la región del editor. Para obtener más información sobre los encabezados con pestañas, consulte la sección [Pestañas a](https://code.visualstudio.com/docs/getstarted/userinterface#_tabs) continuación.

**Consejo:** puede mover la barra lateral hacia el lado derecho ( **Ver** > **Mover barra lateral hacia la derecha** ) o alternar su visibilidad ( Ctrl + B ).

Resumiendo este es el entorno de IDE Visual Studio Code:

[](https://blog.aitana.es/wp-content/uploads/2020/02/visual-studio-code.jpg)

La barra de actividades es uno de los elementos que más se utilizan cuando estamos programando en VSCode. Podemos cambiar de ventana mucho más rápido aplicando estos atajos de teclado:

* **Ctrl + B**: Ocultar/Mostrar barra de actividades
* **Ctrl + Shift +**
  + **E** – **E**xplorador
  + **F** – Buscar (pensar en **F**ind)
  + **G** – Control de código (pensar en **G**it)
  + **D** – **D**epurador
  + **X** – E**x**tensiones

Por otro lado, la paleta de comandos es la principal vía de interacción con VSCode, por lo que saber acceder rápidamente a ella puede ser de mucha utilidad.

* **Ctrl + Shift + P**
  + Acceso a toda la funcionalidad a través del teclado.
* **Ctrl + P**
  + Acceso a los ficheros.

**Configurabilidad**

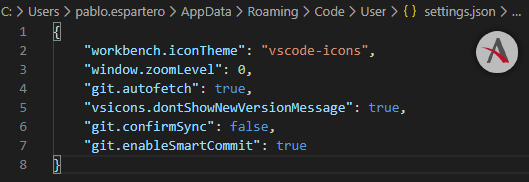
Las opciones de VSCode se almacenan en ficheros JSON (JavaScript Object Notation), lo que nos permite editar toda la configurabilidad de este.

Para editar la misma podemos acceder de dos formas:

* Ctrl+Shift+P -> Preferences
* File -> Preferences

**¿Qué podemos configurar?**

* Prácticamente todo el comportamiento del editor, como el tabulado de líneas, el auto formateo, el zoom por defecto, etc. Estos ajustes se pueden guardar a tres distintos niveles aplicándolos solo en esos casos:
  + Usuario
  + Espacio de trabajo (Workspace)
  + Lenguaje (AL, C#, JavaScript, etc.)

[](https://blog.aitana.es/wp-content/uploads/2020/02/visual-studio-code-02.jpg)

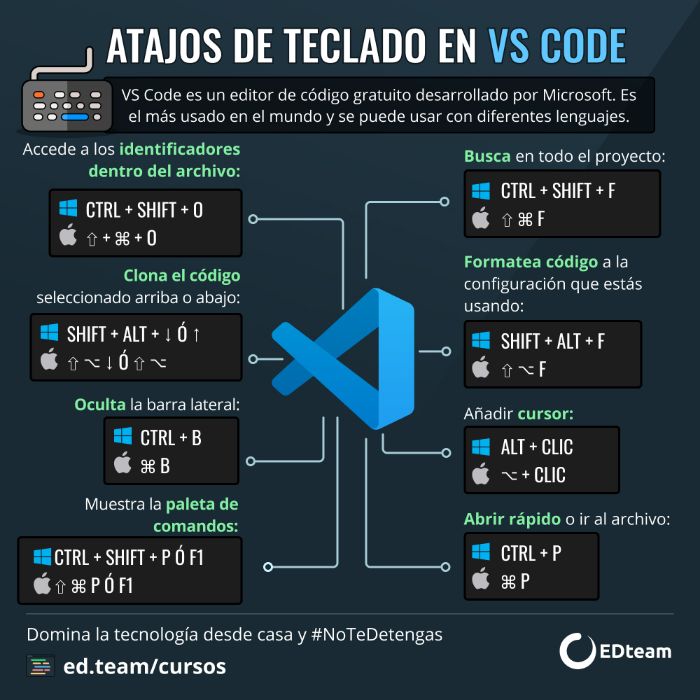
* Los atajos de teclado. VSCode dispone de muchos atajos de teclados, pero si no nos gusta alguno, podemos modificarlo a nuestro gusto.
* El aspecto: Tema del editor, iconos, colores…
* Snippets, trozos de código generados con solo un par de teclas. Podemos crear todos los que queramos y configurarlos para funcionar donde queramos.

**Trucos de edición**

Existen muchas combinaciones de teclado que nos permiten realizar tareas triviales de manera mucho más eficiente.

* Con el multi-cursor podemos editar múltiples puntos del fichero simultáneamente. Se puede realizar de varias maneras:
  + Para hacerlo con el ratón: **Alt + Click**
  + Desde el teclado: **Ctrl + Alt + Up/Down**
  + También podemos seleccionar todas las ocurrencias del texto seleccionado: **Ctrl + Shift + L**
* Cuando queremos modificar una selección de texto:
  + Mover línea: **Alt + Up/Down**
  + Copiar línea: **Alt + Shift + Up/Down**
  + Expandir selección**: Shift + Alt + Right/Left**
* Syde by Syde editing, o abrir varios ficheros en la misma pantalla:
* Sobre explorador: **Ctrl + Enter**
* Sobre fichero: **Crtl + \**

Navegar entre editores: **Ctrl + 1,2,3, etc**



**Sec. 2 COMO ESCRIBIR CÓDIGO HTML Y CSS MAS RÁPIDO CON EMMET (ZEN CODING)**

Aprende como utilizar el plugin Emmet y cómo escribir código HTML y CSS mas rápido y sencillo con este grandioso plugin.

Cuando ya se tiene algo de experiencia dentro del diseño web las cosas comienzan a parecer tediosas, el repetir una y otra vez lo mismo para llegar a una simple estructura html puede ser aburrido y desesperante en ocasiones, es por eso que como buenos profesionales y amantes del buen front-end debemos reducir el tiempo que tardamos durante el desarrollo de un proyecto, esto incluye reducir el tiempo que usamos a la hora de escribir código.

Así que por eso hoy te comparto una herramienta que te permitirá hacer cosas geniales como escribir código HTML y CSS mas rápido de lo que lo has hecho antes. No se trata de un secreto ni mucho menos, es de hecho una herramienta súper famosa y utilizada por todo el mundo, pero si estás leyendo este post seguramente no la conozcas.

Se trata de Emmet, anteriormente conocido como Zed Coding, no es nada más que un simple pero fantástico plugin para nuestros editores de código que nos permitirá escribir código html y css mediante abreviaturas y con solo presionar una tecla se convertirán en nuestro código, listo para editar.

Para poder agregar la anterior estructura gracias a emmet lo único que tendríamos que hacer es escribir html:5 presionar la tecla TAB y automáticamente tendrás la estructura básica de un documento html5 tal como te la mostré en el fragmento de código anterior..

Y prácticamente es la base de todo lo que se puede hacer, puedes hacer muchas estructuras desde lo mas básico hasta lo mas avanzado, puedes desde agregar un link hasta agregar un div con varios elementos dentro, todo con un simple atajo de 1 sola linea.

Emmet funciona tanto para HTML como para CSS y lo mejor de todo es que es un plugin disponible para la mayoría de los editores de código por lo cual es 100% recomendado que lo tengas si eres un diseñador web.

**PRIMERO CODIGO – PRACTICAS DE ESCRITURA USAN CODIFICACION EMMENT:**

<!-- ! -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

</html>

**OTROS CODIGOS (REVISAR MARCAS DE COLOR VERDE):**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

    <body>

        <!--

            Child: >

            nav>ul>li

        -->

        <nav>

            <ul>

                <li></li>

            </ul>

        </nav>

        <!--

            Sibling: +

            div+p+bq

        -->

        <div></div>

        <p></p>

        <blockquote></blockquote>

        <!-- Climb-up: ^ -->

        <!-- div+div>p>span+em^bq -->

        <div></div>

        <div>

            <p><span></span><em></em></p>

            <blockquote></blockquote>

        </div>

        <!-- div+div>p>span+em^^bq -->

        <div></div>

        <div>

            <p><span></span><em></em></p>

        </div>

        <blockquote></blockquote>

        <!-- Multiplication: \* -->

        <!-- ul>li\*5 -->

        <ul>

            <li></li>

            <li></li>

            <li></li>

            <li></li>

            <li></li>

        </ul>

        <!-- Naming and numbering: $ -->

        <!-- ul>li.sample$\*5 -->

        <ul>

            <li class="sample1"></li>

            <li class="sample2"></li>

            <li class="sample3"></li>

            <li class="sample4"></li>

            <li class="sample5"></li>

        </ul>

        <!-- h$[title=topic$]{Headline $}\*3 -->

        <h1 title="topic1">Headline 1</h1>

        <h2 title="topic2">Headline 2</h2>

        <h3 title="topic3">Headline 3</h3>

        <!-- ul>li.item$$$\*5 -->

        <ul>

            <li class="item001"></li>

            <li class="item002"></li>

            <li class="item003"></li>

            <li class="item004"></li>

            <li class="item005"></li>

        </ul>

        <!-- ul>li.item$@-\*5 -->

        <ul>

            <li class="item5"></li>

            <li class="item4"></li>

            <li class="item3"></li>

            <li class="item2"></li>

            <li class="item1"></li>

        </ul>

        <!-- ul>li.item$@3\*5 -->

        <ul>

            <li class="item3"></li>

            <li class="item4"></li>

            <li class="item5"></li>

            <li class="item6"></li>

            <li class="item7"></li>

        </ul>

        <!-- ID and CLASS attributes -->

        <!-- #header -->

            <div id="header"></div>

        <!-- .title -->

            <div class="title"></div>

        <!-- form#search.wide -->

            <form id="search" class="wide">

            </form>

        <!-- p.class1.class2.class3 -->

            <p class="class1 class2 class3"></p>

            <!-- Custom attributes -->

            <!-- p[title="Hello world"] -->

            <p title="Hello world"></p>

            <!-- td[rowspan=2 colspan=3 title] -->

            <td rowspan="2" colspan="3" title=""></td>

            <!-- [a=‘value1‘ b="value2"] -->

            <div a="value1" b="value2"></div>

            <!-- Text: {} -->

            <!-- a{Click me} -->

            <a href="">Click me</a>

            <!-- p>{Click }+a{here}+{ to continue} -->

            <p>Click

                <a href="">here</a>

                to continue

            </p>

            <!-- Implicit tag names -->

            <!-- .class -->

            <div class="class"></div>

            <!-- em>.class -->

            <em><span class="class"></span></em>

            <!-- ul>.class -->

                <ul>

                    <li class="class"></li>

                </ul>

            <!-- table>.row>.col -->

                <table>

                    <tr class="row">

                        <td class="col"></td>

                    </tr>

                </table>

    </body>

</html>

**Sec. 3 INTRODUCCIÓN A LA CREACIÓN DE PAGINAS WEB – HTML & CSS**

**¿Qué es una página Web?**

**Fuente:** [**https://concepto.de/pagina-web/**](https://concepto.de/pagina-web/)

Se conoce como página Web, **página electrónica o página digital a un documento digital de carácter multimediático (es decir, capaz de incluir audio, video, texto y sus combinaciones)**, adaptado a los estándares de la World Wide Web (WWW) y a la que se puede acceder a través de un navegador Web y una conexión activa a Internet. Se trata del formato básico de contenidos en la red.

En Internet existen más de mil millones de páginas Web de diversa índole y diverso contenido, provenientes del mundo entero y en los principales idiomas hablados. Esto representa el principal archivo de información de la humanidad que existe actualmente, almacenado a lo largo de miles de servidores a lo largo del planeta, a los que es posible acceder velozmente gracias a un sistema de protocolos de comunicación (HTTP).

En muchos casos, el acceso a una página Web o a sus contenidos puntuales puede estar sometido a prohibiciones, pagos comerciales u otro tipo de métodos de identificación (como el registro on-line).

El contenido de esta inmensa biblioteca virtual no está del todo supervisado, además, y su regulación representa un reto y un debate para las instituciones tradicionales de la humanidad, como la familia, la escuela o incluso las leyes de los países.

Las páginas Web se encuentran programadas en un formato HTML o XHTML, y se caracterizan por su relación entre unas y otras a través de hipervínculos: enlaces hacia contenidos diversos que permiten una lectura compleja, simultánea y diversa, muy distinta a la que podemos hallar en los libros y revistas.

Por último, no es lo mismo hablar de página Web (Webpage) y de sitio Web (Website), ya que estos últimos contienen un número variable de las primeras.

**¿Para qué sirve una página Web?**

Las páginas Web también permiten la interacción con el usuario.

Las páginas Web cumplen básicamente con la tarea de brindar información de cualquier índole y en cualquier estilo o grado de formalidad.

Algunas, al mismo tiempo, permiten distintos grados de interacción entre usuarios o con alguna institución, como son las páginas de foros, servicios de citas o redes sociales, las páginas de compra y venta de bienes, las páginas de consulta o de contacto con empresas, instituciones gubernamentales o con ONGs, e incluso las páginas de soporte técnico especializado.

En principio, las funciones de una página Web son tan amplias como la demanda de los usuarios y la oferta de sus creadores.

**Tipos de página Web**

Existen dos tipos de página Web, conforme al modo en que se genera su contenido:

Páginas Web estáticas. Operan mediante la descarga de un fichero programado en código HTML, en el que están todas las instrucciones para que el navegador reconstruya la página Web, accediendo a las ubicaciones de sus elementos y siguiendo un orden preconcebido, rígido, que no permite la interacción con el usuario. Este tipo de páginas son meramente informativas, documentales, no interactivas.

Páginas Web dinámicas. A diferencia de las anteriores, las páginas Web dinámicas se generan en el momento mismo del acceso del usuario, empleando para ello algún lenguaje interpretado (como el PHP), lo cual le permite recibir solicitudes del usuario, procesarlas en bases de datos y ofrecer una respuesta acorde a sus requerimientos.

**Navegador Web**

Las páginas Web requieren de un navegador Web para ser utilizadas.

Un navegador Web es un software de aplicación que sirve para abrir páginas Web tanto en una ruta local (como el disco rígido) o provenientes de la Internet.

Se les conoce como “navegadores” o “exploradores”, a partir de la metáfora de que la Red es un lugar vasto y abarrotado, para el que se necesita de una plataforma.

En ese sentido, los navegadores Web nos permiten “entrar” a Internet y visualizar distintos contenidos a partir del ingreso de direcciones URL o del empleo de servicios online de búsqueda de datos (conocidos como Buscadores Web). - Fuente: <https://concepto.de/pagina-web/>

**Estructura del código HTML5 – Anatomía de una página web.**

Asentar las bases de una forma sencilla y básica mostrando la estructura y las partes principales de un código HTML5 para los que aún no se han familiarizado con él.

**LISTADO ETIQUETAS HTML**

A continuación, voy a indicarte cuales son las etiquetas HTML más usadas y también te indicaré su finalidad. Voy a centrarme en las principales etiquetas y sobre todo las que están soportadas en **HTML 5**. Ahí van:

**ETIQUETAS INICIALES O DE RAÍZ**

* **<!DOCTYPE html>** Indica al navegador que el documento está basado en el estándar HTML5
* **<html> </html>** Representa la raíz de un documento HTML. Todos los demás elementos de la estructura HTML deben ser recogidos dentro de estas etiquetas.

**METADATOS DEL DOCUMENTO**

* **<head> </head>** Representa una colección de metadatos acerca del documento, incluyendo enlaces a, o definiciones de, scripts y hojas de estilo. El resto de etiquetas de metadatos, **irán recogidas dentro de las etiquetas de apertura y cierre del head**. Importante explicar que estos metadatos del documento, es información para el navegador y no contenido que será visible en la página web. A excepción de la etiqueta <title> que veremos a continuación.
* **<title> </title>** Etiqueta usada para definir el título de la página web.
* **<link>** Se usa para enlazar recursos externos al documento HTML. El ejemplo más común son **las hojas de estilos CSS.**
* **<meta>** Etiqueta usada para definir otros metadatos que no se pueden definir con una etiqueta HTML especifica. Por ejemplo, para definir el autor del sitio, o la descripción del mismo.
* **<style> </style>** Etiquetas usadas para introducir código CSS en línea, es decir, en el propio documento HTML.

**ETIQUETAS DE SECCIONES O PARA ESTRUCTURAR EL HTML**

* **<body> </body>** Al contrario que la etiqueta de metadatos <head>, todo lo que quieras mostrar en la página web debe ir recogido dentro de las etiquetas de apertura y cierre de <body>. **Este contenido será el que se muestre en la web.**
* **<nav> </nav>** Usadas para definir el contenido que será la sección de navegación de la web.
* **<main> </main>** Se usa para definir el contenido principal del documento. Solamente puede existir uno por documento.
* **<section> </section>** Define una sección del documento
* **<article> </article>** Define contenido independiente de la web.
* **<div> </div>**Sirve para crear secciones o agrupar contenidos.
* **<aside> </aside>** Dentro de estas etiquetas suele alojarse el contenido adicional de la web. Suele ser contenido relacionado con la web pero de poca importancia
* **<h1>,<h2>,<h3>,<h4>,<h5>,<h6>** Son etiquetas HTML muy importantes, ya que son usadas para jerarquizar el contenido de la web. Las etiquetas se usan para explicar brevemente el contenido que irá a continuación.
* **<header> </header>** Se usan para definir la cabecera la página web. Suele contener el logotipo, menú de navegación, etc.
* **<footer> </footer>** Usadas para definir el pie de página.

**ETIQUETAS PARA LA AGRUPACIÓN DE CONTENIDO**

* **<p> </p>** Etiqueta usada para escribir párrafos de texto.
* **<hr>** Etiqueta utilizada para “romper” entre dos secciones de una web. Usada comúnmente como separador.
* **<pre> </pre>** Usada para pegar texto manteniendo el pre formato propio del texto.
* **<blockquote> </blockquote>** Se usan para indicar que el contenido es texto citado.
* **<ol> </ol>** Etiquetas para crear una lista ordenada
* **<ul> </ul>** Etiquetas para crear una lista des-ordenada
* **<li> </li>** Etiquetas que recogen el contenido de un elemento de una lista, sea ordenada o no.
* **<dl> </dl>** Usada para crear una lista de definiciones.
* **<dt> </dt>** Representa un término definido por la siguiente etiqueta <dd>
* **<dd> </dd>** Se usa para definir los términos listados antes que él.
* **<figure> </figure>** Indica una figura ilustrada como parte del documento HTML5.
* **<figcaption> </figcaption>** Utilizada para definir la leyenda de una figura.
* **<div> </div>** Etiqueta común utilizada para crear un contenedor genérico.

**ETIQUETAS SEMÁNTICAS PARA TEXTO**

* **<p> </p>** Etiqueta utilizada para insertar párrafos en el documento HTML
* **<span> </span>** Etiqueta HTML sin ningún significado específico. Se usa conjuntamente con los atributos “class” o “id” para atribuirle ciertas características
* **<a> </a>** Etiqueta utilizada para crear hiperenlaces en el documento HTML
* **<strong> </strong>** Etiqueta para definir una palabra o conjunto de ellas como importantes. **Tiene una fuerte importancia en el SEO de la página.**
* **<em></em>** es para indicar énfasis (reemplazando la etiqueta <i> que utilizábamos anteriormente, actualmente algunos la usan para declarar íconos).
* **<small> </small>** Utilizada para dejar un comentario aparte, del tipo una nota de derechos de autoría, u otros textos que no son esenciales para la comprensión del documento.
* **<cite> </cite>** Para indicar el título de una obra
* **<sub> </sub>** y **<sup> </sup>** Etiquetas utilizadas para representar un subíndice o superíndice.
* **<mark> </mark>** Usada para resaltar texto
* **<br>** Etiqueta utilizada para crear un salto de línea

**ETIQUETAS PARA INCRUSTAR CONTENIDO**

* **<img>** Etiqueta para “pintar” una imagen en la página web.
* **<iframe> </iframe>** Es una etiqueta que sirve para anidar otro documento HTML dentro del documento principal.
* **<embed>** Usada para integrar una aplicación o contenido interactivo externo que no suele ser HTML.
* **<object> </object>** Utilizada llamar a un recurso externo de la web. Este recurso será tratado como una imagen, o un recurso externo para ser procesado por un plugin.
* **<video> </video>** Se usa para reproducir video en la página web junto a sus archivos de audio y capciones asociadas.
* **<audio> </audio>** Usada para cargar un archivo de audio o stream de audio.
* **<source>** Permite a autores especificar recursos multimedia alternativos para las etiquetas de <video> o <audio>
* **<svg> </svg>** Se usa para llamar a una imagen vectorizada.

**ETIQUETAS PARA LA CREACIÓN DE TABLAS**

* **<table> </table>** Etiquetas de apertura y cierre de una tabla. El resto de etiquetas de la tabla han de ir siempre recogidas entre estas dos etiquetas.
* **<caption> </caption>** Usada para indicar el título de la tabla.
* **<colgroup> </colgroup>** Etiqueta utilizada para agrupar dos o más columnas de una tabla.
* **<tbody> </tbody>** Usada para describir los datos concretos de una tabla.
* **<thead> </thead>** Indica el bloque de filas que describen las etiquetas de las columnas de la tabla.
* **<tfoot> </tfoot>** Indica los bloques de filas que describen los resúmenes, o datos totales de una columna de una tabla.
* **<tr> </tr>** Se usa para indicar una fila de celdas de una tabla.
* **<td> </td>** Usada para definir una celda de una tabla.
* **<th> </th>** Etiqueta que se usa para definir el encabezado de una celda

**ETIQUETAS PARA LA CREACIÓN DE FORMULARIOS**

* **<form> </form>** Etiqueta de apertura y cierre de un formulario de una página web. El resto de etiquetas de formulario deben ir siempre recogidas entre estas etiquetas de apertura y cierre de formulario.
* **<fieldset> </fieldset>** Etiqueta que representa un conjunto o agrupación de elementos de un formulario. “Pinta” un recuadro alrededor de las etiquetas que estén contenidas dentro del <fieldset>
* **<legend> </legend>** Etiqueta ligada a <fieldset>. Indica el título del <fieldset>
* **<label> </label>** Se usa para definir el nombre o título de un control del formulario.
* **<input>** Pinta un campo de introducción de datos para el usuario. Es de las principales etiquetas de un formulario.
* **<button> </button>** Etiqueta utilizada para representar un botón en el formulario.
* **<select> </select>** Input que permite una selección entre un conjunto de opciones.
* **<option> </option>** Etiqueta ligada a <select>. Permite añadir diferentes opciones al <select>
* **<textarea> </textarea>** Añade un campo al usuario para que pueda introducir texto en unas líneas máximas de texto que el desarrollador puede definir.

**TIPOS DE ATRIBUTOS HTML**

Existen diferentes tipos de atributos para las etiquetas. Algunos son de ámbito global, mientras que otros son más específicos y solo son válidos para algunas etiquetas HTML. En general podemos ordenarlos de la siguiente manera:

* **Atributos requeridos:** son atributos necesarios para la etiqueta HTML. Sin ellos, la etiqueta por si sola no tendrá el comportamiento esperado.
* **Atributos opcionales**: en este caso, estos atributos son utilizados para modificar la funcionalidad por defecto de la etiqueta HTML.
* **Atributos globales**: son atributos de ámbito general que pueden ser utilizados en casi cualquier etiqueta HTML.

El detalle de estos atributos deberá ser revisados de los libros adjuntos:

* El gran libro de HTML5 CSS3 y JavaScript 3era Edición.
* El gran libro de HTML5 CSS3 y JavaScript 4ta

**CONCLUSIÓN**

Además de las etiquetas HTML que aquí te he mostrado, existen muchas más. Pero sinceramente, creo que**con este listado podrás hacer el 95% de las páginas web**. Y si tienes dudas con alguna etiqueta, o bien has encontrado alguna de la que no te he hablado aquí y quieres saber más, consulta los siguientes enlaces.

W3BAI.COM: **http://www.w3bai.com/es/css/default.html**

W3SCHOOLS: **https://www.w3schools.com/default.asp**

Y el que recomiendo ampliamente y ser muy completo.

**MODULO DISEÑO DE INTERFACES WEB**:

**https://www.eniun.com/diseno-interfaces-web-ciclo-formativo/**

**EJEMPLO DE CÓDIGO HTML DE UNA PÁGINA SENCILLA**

<!DOCTYPE html5>

<head>

    <title>Another Square</title>

    <meta http-equiv="Content-Language" content="English" />

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

    <link rel="stylesheet" type="text/css" href="css/style.css" media="screen" />

</head>

<body>

    <div id="wrap">

        <div id="header">

            <h1><a href="#">Blue</a></h1>

            <h2>Subheader, website description H2</h2>

        </div>

        <div id="menu">

            <ul>

                <li><a href="#">Home</a></li>

                <li><a href="#">About</a></li>

                <li><a href="#">Photos</a></li>

                <li><a href="#">Links</a></li>

                <li><a href="#">Forum</a></li>

                <li><a href="#">Free Stuff</a></li>

                <li><a href="#">Downloads</a></li>

                <li><a href="#">Products</a></li>

                <li><a href="#">SiteMap</a></li>

            </ul>

        </div>

        <div id="content">

            <div class="right">

                <h2><a href="#">Título con Enlace - Ejemplo Cabecera 2</a></h2>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Consequuntur mollitia sequi commodi beatae maiores rerum, consequatur repellendus. Impedit, quos deserunt. <a href="http://www.free-css-templates.com/">Free CSS templates</a> Lorem ipsum dolor sit ametconsectetur adipisicing elit. Iusto quia doloremque beatae eligendi dolores laboriosam recusandae, fugiat tempore! Nulla neque accusamus deserunt, animi dolorum quisquam. Commodi quos temporibus rem architecto. Excepturi soluta laudantium earum est inventore temporibus ducimus doloremque minus, quisquam voluptas distinctio animi laborum aspernatur dicta beatae, exercitationem incidunt!                <br /><br />

                <img src="images/pic.jpg" alt="Example pic" style="border: 3px solid #ccc;" />

                <br /><br />

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Officiis quae voluptates nihil dolores, ut nona eos eligendi architecto modi explicabo libero quas ea at nemo dignissimos nam possimus sed tempora inus, ullam tempore laudantium doloribus pariatur? Minima necessitatibu possimus, odio, deleniti oluptates consequuntur veritatis voluptatibus natus, neque distinctio blanditiis? Ipsum itaque hic ipsam inventore, eligendi reprehenderit doloribus voluptas deleniti?.

                <h2><a href="#">Título con Enlace - Ejemplo Cabecera 2</a></h2>

Donec nulla. Aenean eu augue ac nisl tincidunt rutrum. Proin erat justo, pharetra eget, posuere at, malesuada et, nulla. Donec pretium nibh sed est faucibus suscipit. Nunc nisi. Nullam vehicula. In ipsum lorem, bibendum sed, consectetuer et, gravida id, erat. Ut imperdiet, leo vel condimentum faucibus, risus justo feugiat purus, vitae congue nulla diam non urna.            </div>

            <div class="left">

                <h2>Categories :</h2>

                <ul>

                    <li><a href="#">World Politics</a></li>

                    <li><a href="#">Europe Sport</a></li>

                    <li><a href="#">Networking</a></li>

                    <li><a href="#">Nature - Africa</a></li>

                    <li><a href="#">SuperCool</a></li>

                    <li><a href="#">Last Category</a></li>

                </ul>

                <h2>Archives</h2>

                <ul>

                    <li><a href="#">January 2007</a></li>

                    <li><a href="#">February 2007</a></li>

                    <li><a href="#">March 2007</a></li>

                    <li><a href="#">April 2007</a></li>

                    <li><a href="#">May 2007</a></li>

                    <li><a href="#">June 2007</a></li>

                    <li><a href="#">July 2007</a></li>

                    <li><a href="#">August 2007</a></li>

                    <li><a href="#">September 2007</a></li>

                    <li><a href="#">October 2007</a></li>

                    <li><a href="#">November 2007</a></li>

                    <li><a href="#">December 2007</a></li>

                </ul>

            </div>

            <div style="clear: both;"> </div>

        </div>

        <div id="footer">

            Designed by <a href="http://www.free-css-templates.com/">Free CSS Templates</a>, Thanks to <a

                href="http://www.legalhelpers.com/bankruptcy-information.html">Bankruptcy Information</a>

        </div>

    </div>

</body>

</html>

**Sec. 4 INTRODUCCIÓN AL DISEÑO CON HOJAS DE ESTILO EN CASCADA (CSS)**

1. ¿Qué es CSS?

es un lenguaje que nos permite controlar el aspecto de las páginas web escritas en HTML o en cualquier lenguaje de marcado basado en XML.

El lenguaje CSS está creado por el World Wide Web Consortium (W3C), la comunidad internacional que desarrolla estándares que aseguran el crecimiento futuro de la web y vela por conseguir webs disponibles para todo el mundo.

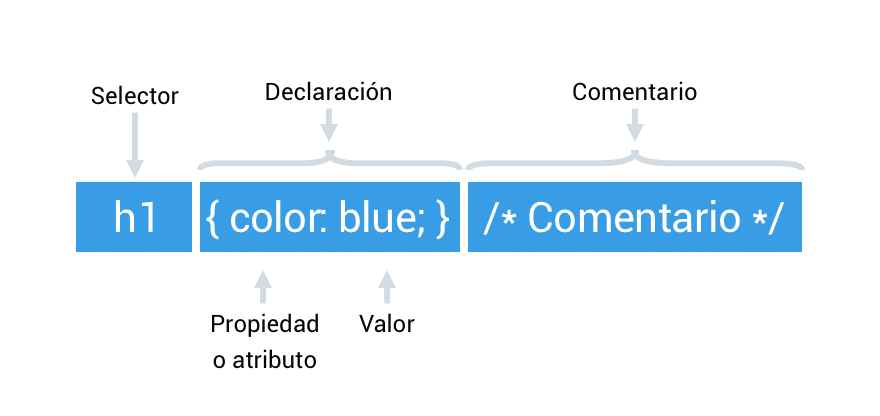
El lenguaje CSS se ha ido creando a lo largo del tiempo en varios niveles. Cada nivel de CSS se ha construido sobre el anterior, generalmente añadiendo funcionalidades nuevas. En la página oficial de W3C ([w3.org/Style/CSS/](https://www.w3.org/Style/CSS/)) se pueden consultar todas las publicaciones relacionadas con las novedades del estándar CSS en sus diferentes versiones.



El estándar CSS2 define más de 100 propiedades, cada una de ellas con su lista de valores permitidos. Por su parte, el estándar CSS3 ya incluye más de 200 propiedades o atributos. A lo largo de todo el capítulo, se detallan las propiedades CSS más utilizadas en relación a los colores, el texto, los fondos, las listas, las tablas y el modelo de cajas. En el siguiente enlace puedes ver un [resumen de las propiedades](https://www.eniun.com/resumen-tabla-propiedades-css-valores/) que se van a estudiar

1.1. Sintaxis CSS

En CSS se utiliza la siguiente sintaxis para asignar valores a las propiedades de cada selector:



* **Selector:** indica sobre qué elemento se aplican los estilos CSS.
* **Propiedad o atributo**: indica qué característica se va a cambiar.
* **Valor:** indica el valor de la propiedad que se desea modificar.
* **Comentario:** los comentarios se escriben entre el carácter de apertura “/\*” y el carácter de cierre “\*/”.

2. Estilos CSS en un documento HTML

Hay tres formas de aplicar estilos CSS en un documento HTML: en línea, incrustado en la cabecera y mediante hojas de estilo externas.

2.1. CSS en línea

Los estilos en línea son declaraciones CSS que se integran en las etiquetas HTML mediante el atributo style. Este método tan solo afecta al elemento en el que se integra el código. El CSS en línea es complicado de entender y mantener ya que mezcla los estilos CSS con el código HTML.

Ejemplo:

<p style="color:green">Párrafo de color verde.</p>

2.2. CSS incrustado en la cabecera

Otra manera muy simple de añadir estilo con CSS es utilizando la etiqueta <style> en la cabecera <head> del fichero HTML del sitio. La desventaja de este método es que a la hora de realizar cualquier cambio, se debe realizar en múltiples páginas diferentes y el código estará repetido. Su uso puede llegar a ser necesario en el caso de utilizar un gestor de contenido que no permita modificar el archivo CSS directamente.

Ejemplo:

<html>

<head>

<title>CSS incrustado en la cabecera</title>

<style> p { color: green; } </style>

</head>

<body>

<p>Párrafo de color verde.</p>

</body>

</html>

2.3. CSS en hojas de estilo externas

Mediante hojas de estilo externas se consigue separar el archivo de estilos del fichero HTML. El archivo de estilos cuenta con la extensión .css y se referencia desde HTML mediante el elemento <link>. Este es el método más eficiente y más sencillo de mantener ya que el código CSS se encuentra separado del fichero HTML.

Ejemplo:

**inicio.html**

<html>

<head>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<p>Párrafo de color verde.</p>

</body>

</html>

**styles.css**

p {color:green;}

**3. Selectores**

Como hemos visto anteriormente, los selectores nos ayudan a indicar el elemento sobre el que se van a aplicar los estilos. Existen muchos tipos de selectores y algunos de los más destacados son los que se detallan a continuación.

**3.1. Selector universal**

**Sintaxis:**\* { atributo:valor; }

**Ejemplo:**\* { color: grey; }/\* El estilo se aplicará a todos los elementos de la página\*/

**3.2. Selector etiqueta**

**Sintaxis:** etiqueta { atributo:valor }

**Ejemplo:** p {color: green;}  /\* El estilo se aplicará a todos los elementos <p>.\*/

**3.3. Selector clase**

**Sintaxis:**.clase { atributo:valor }

**Ejemplo:** .blend{color: red;} /\* El estilo se aplicará a cualquier elemento que tenga la clase .blend \*/

**3.4. Selector identificador**

El **selector identificador**utiliza el atributo id para seleccionar un elemento. Solo puede haber un elemento con un id dado en un documento.

**Sintaxis:** #id { atributo:valor }

**Ejemplo:** #cent {color: blue;} /\* El estilo se aplicará al elemento que tenga el id #cent \*/

**3.5. Selector descendiente**

Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre.

**Sintaxis:**selector1 selector2 selectorN {atributo: valor;} /\* El estilo se aplica sobre el selector N \*/

**Ejemplo:** div p { color: black;} /\* El estilo se aplica a todos los párrafos que se encuentren dentro de una etiqueta div \*/

**3.6. Combinación de selectores**

La combinación de selectores nos permite dar un estilo a todos los selectores indicados.

**Sintaxis:** selector1, selector2, selector3{atributo: valor;} /\* El estilo se aplica sobre los selectores indicados \*/

**Ejemplo:** div, p { color: orange;} /\* El estilo se aplica a todos los divs y párrafos \*/

**3.7. Selector de hijos**

Se usa para seleccionar un elemento que es hijo de otro elemento.

**Sintaxis:** selector1 > selector2 {atributo: valor;}  /\* El estilo se aplica sobre el selector 2 \*/

**Ejemplo:** div > p { color: white;} /\* El estilo se aplica a todos los párrafos que sean hijos de un div \*/

**3.8. Selector adyacente**

Se usa para seleccionar elementos que son hermanos, es decir, su elemento padre es el mismo y están seguidos en el código HTML.

**Sintaxis:** selector1 + selector2{ atributo: valor; } /\* El estilo se aplica al selector 2 \*/

**Ejemplo:** div + p { color: black;} /\* El estilo se aplica a todos los párrafos que sean hermanos de un div \*/

3.9. Resumen de selectores básicos

|  |  |
| --- | --- |
| **Selector** | **Descripción** |
| **\*** | Selecciona todos los elementos del DOM |
| **etiqueta** | Selecciona todas las etiquetas indicadas |
| **.class** | Selección de los elementos con la clase .class |
| **#id** | Selección del elemento con id #id |
| **sel1 sel2** | Selección de los selectores sel2 que se encuentren dentro de los selectores sel1 |
| **.class1.class2** | Selección de los elementos con las dos clases: class1 y class2 |
| **sel1.class1** | Selección de todos los selectores sel1 con clase class1 |
| **sel1, sel2** | Selección de todos los selectores separados por comas |
| **sel1 > sel2** | Selección de los selectores sel2 cuando son hijos de sel1 |
| **sel1 + sel2** | Selección del selector sel2 cuando es hermano de sel1 (su elemento padre es el mismo) |

4. Unidades de medida

El tamaño de los elementos de una web se puede expresar en unidades **absolutas**y **relativas**.

4.1. Unidades absolutas

Las **unidades absolutas** mantienen su aspecto y se visualizan siempre igual independientemente de las características del dispositivo.

|  |  |
| --- | --- |
| in | Pulgadas (1 pulgada = 2.54 cm) |
| cm | Centímetros |
| mm | Milímetros |
| pt | Puntos (1 pt = 1/72 pulgadas) |
| pc | Picas (1 pica = 12 puntos) |

4.2. Unidades relativas

Las**unidades relativas** se ajustan a cada tipo de dispositivo ya que dependen de la resolución de cada pantalla.

|  |  |
| --- | --- |
| px | Píxeles (relativo al dispositivo) |
| em | Relativo al tamaño de la fuente del elemento ( 2 em significa 2 veces el tamaño de la fuente actual) |
| % | Porcentaje (relativo al elemento padre) |
| vh y vw | Medidas relativas de acuerdo al viewport1 vh = 1% de la altura del viewport100 vh = altura del viewport |

4.3. Usos de unidades

Normalmente es recomendable usar unidades relativas en la medida de lo posible, ya que mejora la accesibilidad de la página web y permite que los documentos se adapten fácilmente a cualquier medio. Por tanto, para la creación de una página web, el uso de medidas absolutas queda descartado.

Aunque no hay un criterio definido, el organismo W3C, recomienda el uso de la unidad em para indicar el tamaño del texto. El tamaño de los ems se establece en base al tamaño que tenga definido el navegador.

Usualmente el tamaño de una fuente por defecto en los navegadores es de 16px. Por tanto, tendríamos que 16px = 1em y podríamos definir la siguiente conversión entre unidades.

|  |  |  |
| --- | --- | --- |
| **px** | **em** | **%** |
| 12 | 0,750 | 75 |
| 14 | 0,875 | 87,5 |
| 16 | 1,000 | 100 |
| 18 | 1,125 | 112,5 |
| 20 | 1,250 | 125 |

 Hay que tener en cuenta que el tamaño base definido en los navegadores puede ser modificado por los usuarios. Este tema lo trataremos en detalle en la unidad sobre accesibilidad web.

5. Colores y fondo

Los valores de los colores se pueden definir mediante su nombre, en código hexadecimal o mediante sus valores en los siguientes formatos:

* RGB (Red, Green, Blue)
* HSL (Hue, Saturation, Lightness)
* RGBA (Red, Green, Blue, Alpha)
* HSLA (Hue, Saturation, Lightness, Alpha)

5.1. Tabla de colores básicos

HTML soporta alrededor de [140 nombres de colores](https://www.eniun.com/lista-codigos-de-colores-html-css/) diferentes. Entre ellos puedes encontrar los siguientes colores básicos:

|  |  |  |  |
| --- | --- | --- | --- |
| **Color** | **Nombre** | **HEX** | **RGB** |
|  | black | #000000 | 0,0,0 |
|  | white | #ffffff | 255,255,255 |
|  | red | #ff0000 | 255,0,0 |
|  | blue | #0000ff | 0,0,255 |
|  | yellow | #ffff00 | 255,255,0 |
|  | gray | #808080 | 128,128,128 |
|  | green | #008000 | 0,128,0 |
| Table 5.1. Colores básicos | | | |

5.2. Propiedades más utilizadas

Algunas de las propiedades relacionadas con el color y el fondo más utilizadas son las siguientes:

| **Propiedad** | **Descripción** | **Valores** |
| --- | --- | --- |
| **color** | Color del texto | RGB | HSL | HEX | nombre del color | RGBA | HSLA |
| **background-color** | Color de fondo | RGB | HSL | HEX | nombre del color | RGBA | HSLA |
| **background-image** | Imagen de fondo | url(…) | none |
| **background-repeat** | Repetición de la imagen de fondo | repeat | repeat-x | repeat-y | no-repeat |
| **background-attachment** | Desplazamiento de la imagen de fondo | scroll | fixed |
| **background-position** | Posición de la imagen de fondo | percentage | length | left | center | right |
| **background-size** | Tamaño de la imagen de fondo | valor |
| **Opacity** | Transparencia de un elemento | [ 0 – 1 ] (0 → totalmente transparente) |
| Tabla 5.2: Propiedades colores y fondo | | |

6. Propiedades de texto

Las propiedades de texto son las que nos permiten controlar el texto como bloque, es decir, afectan al interlineado, a la separación entre palabras, al tabulado, etc. En la tabla 6.1 se muestran**las propiedades de texto** más utilizadas.

| **Propiedad** | **Descripción** | **Valores** |
| --- | --- | --- |
| **text-indent** | Desplazamiento de la primera línea del texto | longitud | porcentaje |
| **text-align** | Alineamiento del texto | left | right | center | justify |
| **text-decoration** | Efectos de subrayado, tachado | none | underline | overline | line-through | \* |
| **letter-spacing** | Espacio entre caracteres | normal | longitud |
| **word-spacing** | Espacio entre palabras | normal | longitud |
| **text-transform** | Transformación a mayúsculas / minúsculas | capitalize | uppercase | lowercase | none |
| **line-height** | Tamaño del espacio entre líneas | longitud | porcentaje |
| **vertical-align** | Alineación vertical | top | middle | bottom baseline | sub | super | valor |
| Tabla 6.1: Propiedades de texto | | |

7. Propiedades de las fuentes

Las propiedades CSS de las fuentes son las que permiten controlar el tamaño, el tipo, el grosor o el estilo de las letras, entre otras cosas. En la siguiente tabla se muestran**las propiedades de las fuentes** más destacadas.

| **Propiedad** | **Descripción** | **Valores** |
| --- | --- | --- |
| **font-family** | Familias de fuentes | nombre-familia|nombre-familia-genérica| \* |
| **font-style** | Estilo de la fuente | normal | italic | oblique |
| **font-variant** | Convierte a mayúsculas manteniendo un tamaño inferior | normal | small-caps |
| **font-weight** | Anchura de los caracteres. Normal = 400, Negrita = 700 | normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 |
| **font-size** | Tamaño de la fuente | xx-small | x-small | small | medium | large | x-large | xx-large | larger | smaller | longitud | porcentaje |
| Tabla 7.1: Propiedades de las fuentes | | |

8. Propiedades de las listas

Las propiedades CSS de las listas son las que nos permiten controlar los estilos de los marcadores y la posición de los elementos dentro de las listas. En la tabla 8.1 se muestran**las propiedades de las listas** más utilizadas.

| **Propiedad** | **Descripción** | **Valores** |
| --- | --- | --- |
| **list-style-type** | Estilo aplicable a los marcadores visuales de las listas | disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian | georgian | lower-alpha | upper-alpha | none |
| **list-style-image** | Imagen aplicable a los elementos de las listas | url(“…”) | none |
| **list-style-position** | Posición del marcador dentro de la lista | inside | outside |
| **list-style** | Permite establecer varios estilos de la lista en una sola propiedad | list-style-type | list-style-position | list-style-image |
| Tabla 8.1: Propiedades de las listas | | |

9. Propiedades de las tablas

Las propiedades CSS de las tablas son las que nos permiten controlar los estilos de los títulos de la tabla, el tamaño de las celdas, las filas y las columnas o espaciado entre los bordes. En la tabla 9.1 se muestran**las propiedades de las tablas** más utilizadas.

| **Propiedad** | **Descripción** | **Valores** |
| --- | --- | --- |
| **caption-side** | Posición del título respecto la tabla | top | bottom |
| **table-layout** | Formato de las celdas, filas y columnas | auto | fixed |
| **border-collapse** | Selección del modelo de los bordes | collapse | separate |
| **border-spacing** | Espaciado entre los bordes de celdas adyacentes | longitud |
| **empty-cells** | Visibilidad de los bordes de celdas sin contenido | show | hide |
| Tabla 9.1: Propiedades de las tablas | | |

10. Pseudo-clases y pseudo-elementos en CSS

Gracias a las pseudo-clases y los pseudo-elementos de CSS podemos realizar una selección más específica de los elementos a los que queremos aplicar un cierto estilo sin necesidad de crear una clase concreta.

10.1. Pseudo-clases para selección de hijos o hermanos

Las pseudo-clases más importantes para la selección de hijos o hermanos son las que se muestran en la siguiente tabla.

| **Pseudo-clase** | **Descripción** |
| --- | --- |
| :first-child | Primer hijo |
| :last-child | Último hijo |
| :first-of-type | Primer hermano de su tipo |
| :last-of-type | Último hermano de su tipo |
| :only-child | Hijos únicos |
| :only-of-type | Únicos hermanos de su tipo |
| :empty | Elementos que no tienen hijos |
| :nth-child(n) | Enésimo elemento hijo |
| :nth-last-child(n) | Enésimo elemento hijo contando desde el último |
| :nth-of-type(n) | Enésimo hermano de su tipo |
| :nth-last-of-type(n) | Enésimo hermano de su tipo comenzando desde el último |

#### **Ejemplo pseudo-clase nth-child:**

Veamos cómo funciona la pseudo-clase nth-child mediante un ejemplo en el que aprovechamos la jerarquía de los elementos para referenciar una etiqueta. En el siguiente código tenemos cuatro elementos <p> que son hermanos entre sí e hijos del elemento <div>.

<div>

    <p>Primer párrafo</p>

    <p>Segundo párrafo</p>

    <p>Tercer párrafo</p>

    <p>Cuarto párrafo</p>

</div>

Utilizando la pseudo-clase **nth-child()**podemos seleccionar un hijo específico. El número que indiquemos entre paréntesis será la posición del hijo.

En el siguiente código, seleccionaremos todos los párrafos que sean el primer, segundo y cuarto hijo de un contenedor (div, span, etc.).

p:nth-child(1){ color: red; }

p:nth-child(2){ color: green; }

p:nth-child(4){ color: blue; }

#### **Ejemplo odd (impar) y even (par):**

Los valores odd y even son muy útiles para seleccionar elementos pares e impares. El valor *even* representa elementos en posición par y el valor oddrepresenta los elementos en posición impar. En el siguiente ejemplo en el que se ha creado una tabla se muestran en verde las filas impares y en rojo las filas pares.

tr:nth-child(odd){ color: green;}

tr:nth-child(even){ color: red;}

#### **Ejemplo pseudo-clase first-child y last-child:**

Vamos a utilizar ahora las propiedades first-child y last-child para seleccionar los párrafos que sean el primer y el último hijo de un contenedor.

p:first-child{ color: red; }

p:last-child{ color: green; }

10.2. Pseudo-clases para los estados de un elemento

Podemos utilizar diferentes pseudo-clases para definir las propiedades de ciertos elementos con diferentes estados. Uno de los usos más comunes es en los enlaces <a> aunque se pueden aplicar en otros elementos.

|  |  |
| --- | --- |
| **Pseudo-clase** | **Descripción** |
| :link | No visitado por el usuario |
| :visited | Visitado por el usuario |
| :hover | Modifica el estilo cuando un elemento apuntador pasa por encima |
| :active | Se activa cuando el usuario pulsa el elemento |
| :focus | Se activa cuando tiene el foco sobre el elemento |

Aplica las siguientes propiedades sobre un enlace y observa las diferencias entre los distintos valores.

**a:hover** {  text-decoration: overline;}

**a:active** { background-color: yellow;}

**a:link** { color: green;}

10.3. Pseudo-elementos

A diferencia de las pseudo-clases, los pseudo-elementos no describen un estado sino que permiten añadir estilos a una parte específica del documento. Por ejemplo, el pseudo-elemento :first-line selecciona la primera línea del elemento indicado por el selector.

|  |  |
| --- | --- |
| **Pseudo-elemento** | **Descripción** |
| ::first-line | Primera línea de texto de un elemento |
| ::first-letter | Primera letra de la primera línea de texto de un elemento |
| ::before | Añade contenido al principio del documento |
| ::after | Añade contenido al final del documento |
| ::selection | Coge la porción del texto que se está seleccionando por el usuario |

Aplica las siguientes propiedades sobre diferentes elementos y observa las diferencias entre los distintos resultados.

p::**first-line** { color: green; }

h2::**first-letter** { color: blue; }

p::**after** { content: " - Aloja more don";  color: red; }

p::**before** { content: "\*. "; }

p::**selection** { color: brown; }

**Dos temas claves pero extensos los revisaremos desde la web, estos son**:

**Alinear contenidos con CSS3**: https://www.wextensible.com/temas/css3-alinear/

**Alinear elementos de bloque en CSS3**: https://www.wextensible.com/temas/css3-alinear/block.html#alignmargin

**Sec. 5 LISTADO DE PROPIEDADES DE CSS**

| **Properties** | |
| --- | --- |
| align-content | comportamiento de la propiedad flex-wrap |
| align-items | alineación para artículos dentro del contenedor |
| align-self | alineación para el elemento seleccionado |
| all | cambia todas las propiedades |
| animation | une una animación a un elemento |
| animation-delay | retrasa el inicio de la animación |
| animation-direction | animación inversa o en ciclos alternos |
| animation-duration | duración de la animación en segundos o ms |
| animation-fill-mode | estilo cuando la animación no se está reproduciendo |
| animation-iteration-count | número de repeticiones de una animación |
| animation-name | nombre de la animación @keyframes |
| animation-play-state | la animación se está ejecutando o en pausa |
| animation-timing-function | curva de velocidad de una animación |
| backface-visibility | es el elemento visible cuando no está frente a la pantalla |
| background | todas las propiedades de fondo en una declaración |
| background-attachment | ¿La imagen de fondo está fija o se desplaza? |
| background-blend-mode | modo de fusión de cada capa de fondo |
| background-clip | área de pintura del fondo |
| background-color | color de fondo |
| background-image | imagen de fondo |
| background-origin | donde se coloca la imagen de fondo |
| background-position | posición inicial de la imagen de fondo |
| background-repeat | la forma en que se repite la imagen de fondo |
| background-size | tamaño de la imagen de fondo |
| border | establece todas las propiedades del borde en una línea |
| border-bottom | propiedades del borde inferior en una línea |
| border-bottom-color | color del borde inferior |
| border-bottom-left-radius | borde inferior izquierdo radio |
| border-bottom-right-radius | borde inferior derecho radio |
| border-bottom-style | estilo de borde inferior |
| border-bottom-width | ancho inferior del borde |
| border-collapse | colapso de la frontera |
| border-color | color del borde |
| border-image | establece una imagen como borde |
| border-image-outset | el área de la imagen del borde se extiende más allá del cuadro del borde |
| border-image-repeat | imagen de borde repetida, redondeada o estirada |
| border-image-slice | cómo cortar la imagen del borde |
| border-image-source | camino a la imagen del borde |
| border-image-width | ancho de la imagen del borde |
| border-left | propiedades del borde izquierdo en una línea |
| border-left-color | color del borde izquierdo |
| border-left-style | estilo de borde izquierdo |
| border-left-width | borde izquierdo ancho |
| border-radius | radio de borde de las cuatro esquinas redondeadas |
| border-right | propiedades del borde derecho en una línea |
| border-right-color | color del borde derecho |
| border-right-style | estilo de borde derecho |
| border-right-width | borde derecho ancho |
| border-spacing | espaciado de borde |
| border-style | estilo de borde |
| border-top | propiedades del borde superior en una línea |
| border-top-color | color superior del borde |
| border-top-left-radius | borde superior izquierdo radio |
| border-top-right-radius | borde superior derecho radio |
| border-top-style | estilo de borde superior |
| border-top-width | ancho superior del borde |
| border-width | ancho del borde |
| bottom | desplazamiento inferior para elementos relativos y absolutos |
| box-shadow | sombra al elemento |
| box-sizing | propiedades de tamaño de caja |
| caption-side | colocación de un título de tabla |
| clear | negar la flotación de un elemento |
| clip | recortar un elemento absolutamente posicionado |
| color | color de texto |
| column-count | dividir el contenido en columnas |
| column-fill | relleno equilibrado o no |
| column-gap | espacio entre las columnas |
| column-rule | separador entre columnas, como borde |
| column-rule-color | color de la regla de columna |
| column-rule-style | estilo de regla de columna |
| column-rule-width | ancho de la regla de columna |
| column-span | lapso de columna |
| column-width | ancho de columna |
| columns | establecer el ancho de la columna y el recuento de columnas |
| content | insertar contenido: antes y después de los elementos |
| counter-increment | contar secciones |
| counter-reset | reiniciar contador |
| cursor | tipo de cursor cuando el elemento está suspendido |
| direction | dirección de escritura, el árabe está usando rtl |
| display | tipo de pantalla de caja |
| empty-cells | ocultar bordes y fondo en celdas de tabla vacías |
| filter | efectos de imagen: escala de grises, desenfoque, invertir, etc. |
| flex | longitud del artículo, en relación con otros dentro del contenedor |
| flex-basis | longitud inicial de un artículo flexible |
| flex-direction | dirección de los elementos flexibles |
| flex-flow | abreviatura de flex-direction y flex-wrap |
| flex-grow | cuánto crecerá el artículo en relación con otros artículos |
| flex-shrink | cómo encoger el artículo en relación con otros artículos |
| flex-wrap | envolver artículos flexibles |
| float | elementos flotantes a la izquierda o a la derecha |
| font | todas las propiedades de la fuente en una línea |
| @font-face | declarar fuentes no seguras para la web |
| font-family | fuente del elemento |
| font-size | tamaño de fuente |
| font-size-adjust | controlar el tamaño de la fuente si la primera opción declarada no está disponible |
| font-stretch | ampliar o estrechar el texto |
| font-style | estilo de fuente: normal, cursiva, oblicua |
| font-variant | establecer versalitas |
| font-weight | use caracteres en negrita o delgados |
| hanging-punctuation | ¿Se puede colocar un signo de puntuación fuera del recuadro de línea? |
| height | altura del elemento |
| justify-content | justifica los elementos del contenedor flexible horizontalmente si es necesario |
| @keyframes | especifica el código de animación |
| left | desplazamiento a la izquierda para elementos relativos y absolutos |
| letter-spacing | espacio entre personajes |
| line-height | altura de línea de texto o elementos de bloque en línea |
| list-style | todas las propiedades de la lista en una línea |
| list-style-image | Reemplazar el marcador del elemento de la lista con una imagen. |
| list-style-position | enumerar marcadores de elementos dentro o fuera del flujo de contenido |
| list-style-type | establecer el tipo de marcador de elemento de lista |
| margin | establecer los márgenes superior, derecho, inferior e izquierdo en una línea |
| margin-bottom | margen inferior |
| margin-left | margen izquierdo |
| margin-right | Margen derecho |
| margin-top | margen superior |
| max-height | altura máxima del elemento |
| max-width | ancho máximo del elemento |
| @media | ver consultas de medios |
| min-height | altura mínima |
| min-width | ancho mínimo |
| nav-down | dónde navegar cuando se presiona el botón de flecha hacia abajo |
| nav-index | establece el orden de navegación secuencial |
| nav-left | dónde navegar cuando se presiona el botón de flecha izquierda |
| nav-right | dónde navegar cuando se presiona el botón de flecha hacia la derecha |
| nav-up | dónde navegar cuando se presiona el botón de flecha hacia arriba |
| opacity | nivel de transparencia de un elemento |
| order | reordenar elementos en un contenedor |
| outline | ahogar un borde exterior alrededor de los elementos |
| outline-color | color del contorno |
| outline-offset | espacio entre el elemento y el contorno |
| outline-style | estilo de contorno |
| outline-width | ancho del contorno |
| overflow | ocultar, mostrar o desplazarse si el contenido desborda su contenedor |
| overflow-x | desbordamiento horizontal |
| overflow-y | desbordamiento vertical |
| padding | relleno entre el borde del elemento y el contenido |
| padding-bottom | fondo acolchado |
| padding-left | relleno a la izquierda |
| padding-right | acolchado a la derecha |
| padding-top | acolchado superior |
| page-break-after | agrega un salto de página después de un elemento |
| page-break-before | agrega un salto de página antes de un elemento |
| page-break-inside | permitir salto de página dentro de un elemento |
| perspective | cuántos píxeles se coloca el elemento 3D de la vista |
| perspective-origin | donde está el elemento 3D basado en el eje xey |
| position | tipo de posicionamiento: absoluto, fijo, relativo, estático |
| quotes | poner comillas para envolver un elemento |
| resize | declarar elementos redimensionables |
| right | desplazamiento a la derecha para elementos relativos y absolutos |
| tab-size | longitud del espacio de caracteres de tabulación |
| table-layout | algoritmo de diseño de tabla |
| text-align | alineación horizontal del texto |
| text-align-last | alineación horizontal de la última línea de texto |
| text-decoration | sobrepasar, subrayar o repasar el texto |
| text-indent | sangría de la primera línea del texto |
| text-overflow | la forma en que se marca el contenido desbordado (puntos suspensivos) |
| text-shadow | sombra de texto |
| text-transform | capitalización del texto |
| top | desplazamiento superior para elementos relativos y absolutos |
| transform | Transformación 2D 3D. Ver widget. |
| transform-origin | cambia la posición de los elementos transformados |
| transform-style | renderizar elementos anidados en 3D |
| transition | propiedades de transición en una línea |
| transition-delay | retraso antes del inicio del efecto de transición |
| transition-duration | duración del efecto de transición |
| transition-property | qué propiedad CSS es la transición que afecta |
| transition-timing-function | curva de velocidad de la transición |
| unicode-bidi | ¿Debería anularse el texto para admitir más idiomas? |
| user-select | deshabilitar la selección de contenido del usuario |
| vertical-align | alineamiento vertical |
| visibility | visibilidad: los elementos ocultos dejan un hueco |
| white-space | ¿Cómo se manejan los espacios en blanco? |
| width | ancho de un elemento |
| word-break | reglas de ruptura de texto cuando el texto llega al final del contenedor |
| word-spacing | tamaño del espacio en blanco entre palabras |
| word-wrap | romper palabras largas y pasar a la siguiente línea |
| z-index | orden de pila del elemento |

**Sec. 6 GUÍA DE REFERENCIA RÁPIDA DE CSS 3**

*Notas:  
  
· Las propiedades escritas en texto normal equivalen a valores a usar tal y como se indican.  
· Las propiedades escritas en texto cursiva equivalen a valores numéricos acompañados con la unidad de medida correspondiente, ejemplo: 2px, 2%, 2em.  
· No todas las propiedades que se definen en esta guía de referencia son compatibles con todos los navegadores. Algunas de ellas están en fase experimental y requieren de nomenclatura especial si se desea usarse en navegadores como Firefox, Safari o Chrome. Para más aclaración acerca de la*[***compatibilidad del CSS 3 con los navegadores consulta este artículo.***](http://theproc.es/2010/1/29/12237/manual-sobre-lo-nuevo-en-css-3) *Ejemplo: para aplicar bordes redondeados en Firefox debemos escribir: -moz-border-radius: 5px, para Chrome: -webkit-border-radius: 5px.*

background

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*15/10/2009*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**background-image**](http://theproc.es/page/guia_referencia_css3?layout=false) | *url(..) none* | background-image: url(../images/fondo.jpg); background-image: none; |
| [**background-position**](http://theproc.es/page/guia_referencia_css3?layout=false) | *top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right x-% y-% (ex. 50% 50%) x-pos y-pos (ex. 10px 20px)* | background-position: top left; background-position: 50px 50px; background-position: 50% 50%; |
| [**background-size**](http://theproc.es/page/guia_referencia_css3?layout=false) | *longitud en píxels longitud en % auto, cover, contain* | background-size: 200px 50px; background-size: 100% 50%; |
| [**background-repeat**](http://theproc.es/page/guia_referencia_css3?layout=false) | *repeat, repeat-x, repeat-y, no-repeat* | background-repeat: none; background-repeat: repeat-x; background-repeat: repeat-y; |
| [**background-attachment**](http://theproc.es/page/guia_referencia_css3?layout=false) | *scroll, fixed* | background-attachment: scroll;  background-attachment: fixed; |
| [**background-origin**](http://theproc.es/page/guia_referencia_css3?layout=false) | *border-box, padding-box, content-box* | background-origin: border-box; background-origin: padding-box; background-origin: content-box; |
| [**background-clip**](http://theproc.es/page/guia_referencia_css3?layout=false) | *longitud en píxels longitud en % border-box, padding-box, content-box, no-clip* | background-clip: border-box; background-clip: 10px 10px; background-clip: 20% 40%; |
| [**background-color**](http://theproc.es/page/guia_referencia_css3?layout=false) | *código de color transparent* | background-color: #333333; background-color: transparent; background-color: rgb (255,0,0) |

border

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*15/10/2009*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**border-width**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **border-top-width** * **border-right-width** * **border-bottom-width** * **border-left-width** | *thin, medium, thick longitud en píxels (ex. 1px)* | border-width: 5px; border-top-width: 5px; border-left-width: 5px; border-bottom-width: 5px; border-right-width: 5px; |
| [**border-style**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **border-top-style** * **border-right-style** * **border-bottom-style** * **border-left-style** | *none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset* | border-style: 1px solid #000000; border-bottom-style: 1px solid #000000; |
| [**border-color**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **border-top-color** * **border-right-color** * **border-bottom-color** * **border-left-color** | código de color | border-color:#000000; border-top-color:#ffffff; border-bottom-color:#999999; border-left-color:#666666; border-right-color:#cc0000; |
| [**border-radius**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **border-top-left-radius** * **border-top-right-radius** * **border-bottom-left-radius** * **border-bottom-right-radius** | *longitud en píxels (ex. 5px)* | border-radius: 5px; border-top-left-radius:10px; |
| [**box-shadow**](http://theproc.es/page/guia_referencia_css3?layout=false) | *inset ( distancia\_horizontal distancia\_vertical desenfoque código\_color )* | box-shadow: 3px 3px 6px #888888; |

font

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*18/06/2009*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**font-family**](http://theproc.es/page/guia_referencia_css3?layout=false) | *nombre de la familia de la fuente nombre genérico de la familia inherit* | font-family: 'Arial', 'Times-New-Roman', sans-serif; |
| [**font-size**](http://theproc.es/page/guia_referencia_css3?layout=false) | *xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, inherit longitud en pixels longitud en %* | font-size: 1em; font-size: 12px; font-size: 105%; |
| [**font-size-adjust**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, inherit  número* | font-size-adjust: 0.5; font-size-adjust: none; |
| [**font-stretch**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded, inherit* | font-stretch: expanded;  font-stretch: condensed; |
| [**font-style**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal, italic, oblique, inherit* | font-style: normal; font-style: italic; |
| [**font-variant**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal, small-caps, inherit* | font-variant: normal; font-variant: small-caps; |
| [**font-weight**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit* | font-weight: bold; |

box model

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*09/08/2007*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**clear**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, both, left, right* | clear:both; clear:left; clear:right; |
| [**display**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, inline, block, inline-block, list-item, run-in, compact, table, inline-table, table-row-group, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption, ruby, ruby-base, ruby-text, ruby-base-group, ruby-text-group* | Ejemplos de las propiedades compatibles con todos los navegadores:  [**display: block;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**display: inline;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**display: list-item;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**float**](http://theproc.es/page/guia_referencia_css3?layout=false) | *left, right, none* | [**float:none;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**float:left;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**float:right;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**height**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **max-height** * **min-height** | *auto longitud en píxels longitud en %* | [**height: 100px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**height: 25%;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**width**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **max-width** * **min-width** | *auto longitud en píxels longitud en %* | [**width: 100px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**width: 25%;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**margin**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **margin-top** * **margin-bottom** * **margin-left** * **margin-right** | *auto longitud en píxels longitud en %* | [**margin-top: 20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**margin-left: 20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**margin-right: 20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**margin-bottom: 100px;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**padding**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **padding-top** * **padding-bottom** * **padding-left** * **padding-right** | *auto longitud en píxels longitud en %* | [**padding-top: 20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**padding-left: 20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**padding-right: 20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**padding-bottom: 100px;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**marquee-direction**](http://theproc.es/page/guia_referencia_css3?layout=false) | *foward, reversed* | [**marquee-direction:foward;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**marquee-direction:reversed;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**marquee-play-count**](http://theproc.es/page/guia_referencia_css3?layout=false) | *infinite íntegro* | [**marquee-play-count:infinite;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**marquee-play-count:2;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**marquee-speed**](http://theproc.es/page/guia_referencia_css3?layout=false) | *slow, normal, fast* | [**marquee-speed:slow;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**marquee-speed:fast;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**marquee-style**](http://theproc.es/page/guia_referencia_css3?layout=false) | *scroll, slide, alternate* | [**marquee-style:scroll;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**marquee-style:slide;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**overflow**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **overflow-x** * **overflow-y** | *visible, hidden, scroll, auto, no-display, no-content* | [**overflow:hidden;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**overflow:scroll;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**rotation**](http://theproc.es/page/guia_referencia_css3?layout=false) | ángulo de rotación | [**rotation:45deg;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**visibility**](http://theproc.es/page/guia_referencia_css3?layout=false) | *visible, hidden, collapse* | visibility:visible; visibility:hidden; |

text

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*06/03/2007*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**direction**](http://theproc.es/page/guia_referencia_css3?layout=false) | *ltr, rtl, inherit* | direction:ltr; direction:rtl; |
| [**hanging-punctuation**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, (inicio final esquina-final)* | Todavía en fase de desarrollo. |
| [**letter-spacing**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal longitud en pixels  longitud en em* | [**letter-spacing:2px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**letter-spacing:2em;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**punctuation-trim**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, (inicio fin adyacente)* | Todavía en fase de desarrollo. |
| [**text-align**](http://theproc.es/page/guia_referencia_css3?layout=false) | *start, end, left, right, center, justify* | [**text-align:left;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-align:right;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-align:center;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**text-decoration**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, underline, overline, line-through, blink* | [**text-decoration:underline;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-decoration:overline;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-decoration:line-through;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-decoration:blink;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**text-emphasis**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, (accent, dot, circle, disc)* | Sólo para escrituras asiáticas. |
| [**text-indent**](http://theproc.es/page/guia_referencia_css3?layout=false) | longitud en píxels longitud en em | [**text-indent:20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-indent:50%;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**text-justify**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, inter-word, inter-ideograph, inter-cluster, distribute, kashida, tibetan* | [**text-justify:inter-word;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-justify:distribute;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**text-outline**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none  color  longitud en píxels* | Todavía en desarrollo |
| [**text-shadow**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none  longitud en píxels (distancia\_vertical distancia\_horizontal desenfoque color)* | [**text-shadow: 1px 1px 2px #666;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**text-transform**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, capitalize, uppercase, lowercase* | [**text-transform: uppercase;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-transform: capitalize;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-transform: lowercase;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**text-wrap**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal, none, unrestricted, suppress* | [**text-wrap:normal;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-wrap:none;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-wrap:unrestricted;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**text-wrap:suppress;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**unicode-bidi**](http://theproc.es/page/guia_referencia_css3?layout=false) | *inherit, normal, embed, bidioverride* | [**unicode-bidi:inherit;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**unicode-bidi:normal;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**unicode-bidi:embed;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**unicode-bidi:bidi-override;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**white-space**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal, pre, nowrap, prewrap, pre-line* | [**white-space:normal;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**white-space:pre;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**white-space:nowrap;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**white-space:pre-wrap;**](http://theproc.es/page/guia_referencia_css3?layout=false) [**white-space:pre-line;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**white-space-collapse**](http://theproc.es/page/guia_referencia_css3?layout=false) | *preserve, collapse, preserve-breaks, discard* | Todavía en desarrollo. |
| [**word-break**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal, keep-all, loose, break-strict, break-all* | word-break:normal; word-break:keep-all; word-break:loose; word-break:break-strict; word-break:all; |
| [**word-spacing**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none  longitud en píxels  longitud en %* | [**word-spacing:5px;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
|  |  |  |

column

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*30/06/2009*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**column-count**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto número entero* | [**-moz-column-count:2;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**column-gap**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal longitud en píxels* | [**-moz-column-gap:10px;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**column-rule**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **column-rule-width** * **column-rule-color** * **column-rule-style** | *Valores para cada sub-propiedad:*   * *longitud en em o píxels* * *código del color* * *mismos parámetros que border-style* | [**-moz-column-count:2; -moz-column-gap: 20px; -moz-column-rule-width: 40px; -moz-column-rule: 4px solid #cc0000; -moz-column-rule-color: black;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**column-width**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto longitud en píxels o em* | [**-moz-column-width:10px;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
|  |  |  |

color

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*21/07/2008*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**color**](http://theproc.es/page/guia_referencia_css3?layout=false) | *inherit código de color* | [**color:#cc0000;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| [**opacity**](http://theproc.es/page/guia_referencia_css3?layout=false) | *inherit grado de transparencia en número (valor del 0 al 1)* | [**opacity:0.5; filter:alpha(opacity=50); /\* para IE \*/**](http://theproc.es/page/guia_referencia_css3?layout=false) |

template layout

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*09/08/2007*

Esta especificación se encuentra en fase de desarrollo, más concretamente en modo "borrador" según el WWW Consortium.   
Se sabe que dependerá de otros módulos para su elaboración como el *box module, text module, positioning module,* etc...  
El objeto de esta especificación será el de establecer una plantilla invisible que nos sevirá para maquetar el contenido de una página web. Su uso será ideal en caso de páginas web muy complejas y/o formularios complicados.

table

*Según el*[***W3 Consortium***](http://www.w3.org/)*el modelo de tablas será el mismo que el estipulado para CSS2 salvo algunos desarrollos más que no tienen establecido fecha de revisión por el momento.*

Las tablas son una forma más que permite mostrar contenido de forma ordenada y estructurada a través de filas y columnas.   
El modelo de tablas para CSS3 parece que será igual que el actual en CSS2 pero abarcando más detalles. A día de hoy esta espeficiación se encuentra en fase "borrador" y con prioridad baja, no se conoce fecha de próxima revisión en el WWW Consortium.  
A modo de recordatorio se exponen las propiedades actuales de *table*

| Propiedades actuales para *table* en CSS2 | Parámetros |
| --- | --- |
| [**table-layout**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, fixed, inherit* |
| [**border-collapse**](http://theproc.es/page/guia_referencia_css3?layout=false) | *collapse, separate, inherit* |
| [**border-spacing**](http://theproc.es/page/guia_referencia_css3?layout=false) | *inherit longitud longitud* |
| [**caption-side**](http://theproc.es/page/guia_referencia_css3?layout=false) | *top, bottom, inherit* |
| [**empty-cells**](http://theproc.es/page/guia_referencia_css3?layout=false) | *show, hide, inherit* |

speech

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*16/12/2004*

Esta especificación contiene propiedades que permiten que un documento sea interpretado por un sintetizador y reproducido en forma de audio. Ya existía un módulo en CSS2. Las propiedades implementadas en CSS3 son las mismas que en CSS2 pero con diferentes valores.  
Muy usado para usuarios con discapacidad visual.

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**cue**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **cue-before** * **cue-after** | (cue-before cue-after)\*  \* Cada uno de estos parámetros puede adquirir estos valores y deberá ser expresado con esta estructura:  url (ruta\_archivo\_audio ( número | % | silent | x-soft | soft | medium | loud | x-loud | none | inherit )) donde:  **número**, equivale a un valor entre 0 y 100 que indica el volumen con el que se reproducirá el sonido **%**, equivale a un valor entre 0 y 100 que igualmente indica el volumen con el que se reproducirá el sonido *silent | x-soft | soft | medium | loud | x-loud*, son valores establecidos de secuencias monótonas y no decrecientes en volumen de sonidos donde*silent* es 0 y *x-loud* es 100. | cue-before: url(bell.aiff); cue-after: url(dong.wav); cue-before: url(pop.au) 80; cue-after: url(pop.au) 50%; cue: url(pop.au); |
| [**mark**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **mark-before** * **mark-after** | (mark-before mark-after) | mark-before: "start";  mark-after: "end"; mark: "start" "end"; |
| [**pause**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **pause-before** * **pause-after** * **phonemes** | *inherit, (pause-before pause-after)* | pause: 20ms; /\* pause-before: 20ms; pause-after: 20ms \*/ pause: 30ms 40ms; /\* pause-before: 30ms; pause-after: 40ms \*/ pause-after: 10ms; /\* pause-before: unspecified; pause-after: 10ms \*/ |
| [**rest**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **rest-before** * **rest-after** | *none, x-weak, weak, medium, strong, x-strong, inherit tiempo (en segundos o milisegundos)* | rest: 2s 1s; /\*[rest-before rest-after]\*/ |
| [**speak**](http://theproc.es/page/guia_referencia_css3?layout=false) | *none, normal, spell-out, digits, literal-punctuation, no-punctuation, inherit* | speak:normal; speak:spell-out; |
| [**voice-balance**](http://theproc.es/page/guia_referencia_css3?layout=false) | *left, center, right, leftwards, rightwards, inherit número (entre -100 (left) y 100 (right))* | voice-balance:left; voice-balance:90; |
| [**voice-duration**](http://theproc.es/page/guia_referencia_css3?layout=false) | tiempo (en segundos o milisegundos) | voice-duration: 3s; |
| [**voice-family**](http://theproc.es/page/guia_referencia_css3?layout=false) | *(specific-voice, age, generic-voice, number)\*  \* donde:  specific-voice son valores específicos de una voz (ej. "comedian") age tiene como posibles valores child, young y old generic-voice son valores de familias de voces, tiene como valores posiblemale, female y neutral  number especifica la variante preferida dentro de un conjunto de voces característicos. Su valor debe ser un número entero positivoinherit* | voice-family: announcer, old male; voice-family: romeo, young male; voice-family: juliet, female; voice-family: male 3; |
| [**voice-rate**](http://theproc.es/page/guia_referencia_css3?layout=false) | *x-slow, slow, medium, x-fast, fast, inherit % (un 50% significa la mitad del rango de una voz)* | voice-rate:fast; voice-rate:60%; |
| [**voice-pitch**](http://theproc.es/page/guia_referencia_css3?layout=false) | *x-low, low, medium, high, x-high, inherit número entero no negativo (de Hercios) %* | voice-pitch:medium; voice-pitch: 210hz; |
| [**voice-pitch-range**](http://theproc.es/page/guia_referencia_css3?layout=false) | *x-low, low, medium, high, x-high, inherit número entero no negativo (de Hercios) %* | voice-pitch-range:medium; voice-pitch-range: 210hz; |
| [**voice-stress**](http://theproc.es/page/guia_referencia_css3?layout=false) | *strong, moderate, none, reduced, inherit* | voice-stress:strong; |
| [**voice-volume**](http://theproc.es/page/guia_referencia_css3?layout=false) | *silent, x-soft, soft, medium, loud, x-loud, inherit porcentaje %* | voice-volume:medium; voice-volume: 50%; |

list & markers

*Prioridad: Baja | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*07/11/2002*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**list-style**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **list-style-image** * **list-style-position** * **list-style-type** | * *none, url* * *inside, outside* * *none, normal, box, check, circle, square, diamond, disc, hyphen* | [**list-style: none inside circle;**](http://theproc.es/page/guia_referencia_css3?layout=false) |
| **::marker** | Como pseudo-elemento de bloque puede adoptar todas las propiedades normales de un elemento de bloque como *display*.**Propiedad en fase de desarrollo.** | Ejemplo de *marker* aplicado a un elemento *p*  CSS: <STYLE type="text/css"> P { margin-left: 12 em; } P.Note::marker {  content: url("note.gif") "Note " counter(note-counter) ":"; text-align: left; width: 10em; } P.Note { display: list-item; counter-increment: note-counter; } </STYLE> |

animation

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*20/03/2009*

**Este comportamiento se encuentra en fase de desarrollo.** Las animaciones son transiciones que cambian los valores de CSS con respecto al tiempo. Estos valores son especificados con el uso de *keyframes* (puntos concretos de la animación en la que se producen cambios en los valores CSS).

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**animation**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **animation-delay** * **animation-direction** * **animation-duration** * **animation-iteration-count** * **animation-name** * **animation-play-state** * **animation-timing-function** | *Valores para cada propiedad:*   * *tiempo* * *normal, alternate* * *tiempo* * *inherit, infinitetiempo* * *none, IDENT* * *running, paused* * *ease, linear, ease-in, ease-out, cubic-Bezier (número número número número)* | Ejemplo de código de una animación:  div { animation-name: 'diagonal-slide'; animation-duration: 5s; animation-iteration-count: 10; }  @keyframes 'diagonal-slide' {  from { left: 0; top: 0; }  to { left: 100px; top: 100px; }  } |

transition

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*01/12/2009*

**Este comportamiento se encuentra en fase de desarrollo.** Las transiciones son efectos de presentación en los que un elemento concreto cambia sus propiedades de CSS de un valor a otro. [**Sólo propiedades animables pueden ser utilizadas para crear estos efectos de transición.**](http://www.w3.org/TR/css3-transitions/#transitions-)

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**transition**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **transition-delay** * **transition-direction** * **transition-duration** * **transition-property** | *Valores para cada propiedad:*   * *tiempo* * *tiempo* * *none, all* * *ease, linear, ease-in, ease-out, cubic-Bezier (número número número número)* | Ejemplo de código de una transición: |

grid positioning

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*05/09/2007*

**Este módulo se encuentra en fase de desarrollo.** Este módulo añade la capacidad de adaptar en tamaño y posición los elementos de un documento web en base a un sistema de rejilla creado con CSS que sirve como guía.

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**grid**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **grid-columns** * **grid-rows** | *none, inherit  (longitud porcentaje longitud\_relativa)* | Ejemplo de *grid* para un documento web: body {  grid-columns: \* \* (0.5in \* \*)[2]; grid-rows: 20% \*; columns:3;  column-gap:0.5in;  } |

[**Subir**](http://theproc.es/page/guia_referencia_css3?layout=false#selectores)

outline

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*11/05/2004*

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**outline**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **outline-color** * **outline-offset** * **outline-style** * **outline-width** | *Los posibles valores para cada parámetro son:*   * *invert, código de color* * *inherit, longitud* * *none, dotted, dashed, solid, double, groove, ridge, inset, ouset* * *thin, medium, thick o longitud* | [**outline-offset:20px;**](http://theproc.es/page/guia_referencia_css3?layout=false) |

3D/2D transform

**Este módulo se encuentra en fase de desarrollo.** La propiedad *transform* permite modificar el estado de un elemento ya sea por traslación, rotación, escalado o perspectiva en cualquier de los ejes de coordenadas espaciales (x,y,z).

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*01/12/2009*

| Propiedades | Parámetros |
| --- | --- |
| [**transform**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **transform** * **transform-origin** * **transform-style** * **perspective** * **perspective-origin** * **backface-visibility** | *Los posibles valores para cada parámetro son:*   * ***transform*** *none, matrix, matrix3d, translate3d, translateX, translateY, translateZ, scale, scale3d, scaleX, scaleY, scaleZ, rotate, rotate3d, rotateX, rotateY, rotateZ, skew, skewX, skewY, perspective código de color* * ***transform-origin*** *[ [ porcentaje | longitud | left | center | right ] [ porcentaje | longitud | top | center | bottom ]? ] | [ [ left | center | right ] || [ top | center | bottom ] ]* * ***transform-style*** *flat, preserve-3d* * ***perspective*** *none, número* * ***perspective-origin*** *[ [ porcentaje | longitud | left | center | right ] [ porcentaje | longitud | top | center | bottom ]? ] | [ [ left | center | right ] || [ top | center | bottom ] ]* * ***backface-visibility*** *visible, hidden* |

line box

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*15/05/2002*

**La mayoría de las propiedades de este modelo están en borrador y no son soportadas por los navegadores.**

Esta especificación controla los aspectos visuales de los elementos de bloque. Es una propiedad nueva de CSS3. Por norma general el contenido de un elemento de bloque suele empezar en la esquina superior izquierda de este mismo bloque, así pues, las propiedades de *line-box* pretenden establecer nuevos aspectos.

Permite crear inicios de línea como los que se muestran en elas siguientes imágenes:

Para entender los selectores que se describen a continuación es necesario conocer los distintos tipos de línea base o *baseline* que se pueden establecer:

| Propiedades | Parámetros |
| --- | --- |
| [**alignment-adjust**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, baseline, before-edge, text-before-edge, middle, central, after-edge, text-after-edge, ideo-graphic, alphabetic, hang-ing, mathematical longitud  %* |
| [**alignment-baseline**](http://theproc.es/page/guia_referencia_css3?layout=false) | *baseline, use-script, before-edge, text-before-edge, after-edge, text-after-edge, central, middle, ideographic, alphabetic, hanging, mathematical* |
| [**baseline-shift**](http://theproc.es/page/guia_referencia_css3?layout=false) | *baseline, sub, super longitud  %* |
| [**dominant-baseline**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, use-script, no-change, reset-size, alphabetic, hanging, ideographic, mathematical, central, middle, text-after-edge, text-before-edge* |
| [**drop-initial-after-align**](http://theproc.es/page/guia_referencia_css3?layout=false) | Mismos valores que *alignment-baseline* |
| [**drop-initial-after-adjust**](http://theproc.es/page/guia_referencia_css3?layout=false) | *central, middle, after-edge, text-after-edge, ideographic, alphabetic, mathematical longitud %* |
| [**drop-initial-value**](http://theproc.es/page/guia_referencia_css3?layout=false) | *initial número* |
| [**drop-initial-size**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto número % línea* |
| [**inline-box-align**](http://theproc.es/page/guia_referencia_css3?layout=false) | *initial, last número* |
| [**line-height**](http://theproc.es/page/guia_referencia_css3?layout=false) | *normal número longitud %* |
| [**line-stacking**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **line-stacking-strategy** * **line-stacking-ruby** * **line-stacking-shift** | Valores para cada caso:   * *inline-line-height, block-line-height, max-height, grid-height* * *exclude-ruby, include-ruby* * *consider-shifts, disregard-shifts* |
| [**text-height**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, font-size, text-size, max-size* |
| [**vertical-align**](http://theproc.es/page/guia_referencia_css3?layout=false) | *baseline, sub, super, top, text-top, middle, bottom, text-bottom longitud %* |

hyperlink

*Prioridad: Baja | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*24/02/2004*

Este módulo proporcionará diferentes propiedades visuales relacionadas con el comportamiento de los enlaces.

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**target**](http://theproc.es/page/guia_referencia_css3?layout=false)   * **target-name** * **target-new** * **target-position** | Valores en cada caso:   * *current, root, parent, new, modal, cadena* * *window, tab, none* * *above, behind, front, back* | target-new: tab ! important; |

positioning

*Prioridad: Baja | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*no hay fecha publicada*

Este módulo propiedades que indican como deben fluir en el documento web los elementos. No existe en estos momentos desarrollo en CSS3 de sus propiedades, las que existen son las ya indicadas en CSS2.

| Propiedades | Parámetros | Ejemplo |
| --- | --- | --- |
| [**bottom**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto % longitud* | { position: absolute; bottom: 0px; } /\* Localizaría al elemento siempre a una distancia de 0px con respecto al límite inferior de la página.\*/ |
| [**clip**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto shape* | img { position:absolute; clip:rect (0px,60px,200px,0px); } /\* Crearía una máscara rectangular con las dimensiones indicadas entre paréntesis y con una posición absoluta \*/ |
| [**left**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto % longitud* | { position: absolute; left: 0px; } /\* Localizaría al elemento siempre a una distancia de 0px con respecto al límite izquierdo de la página.\*/ |
| [**position**](http://theproc.es/page/guia_referencia_css3?layout=false) | *static, relative, absolute, fixed* | position: relative;  position: absolute; |
| [**right**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto % longitud* | { position: absolute; right: 0px; } /\* Localizaría al elemento siempre a una distancia de 0px con respecto al límite derecho de la página.\*/ |
| [**top**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto % longitud* | { position: absolute; top: 0px; } /\* Localizaría al elemento siempre a una distancia de 0px con respecto al límite superior de la página.\*/ |
| [**z-index**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto número* | { position:relative; z-index:4 } |

ruby

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*14/05/2003*

Este módulo describe las propiedades CSS necesarias para manipular la posición del "ruby", que son pequeñas anotaciones en lo alto de algunas palabras o cerca de ellas que se usan habitualmente para dar una explicación adicional de aquello que se menciona en el texto. Es especialmente utilizado en idiomas como el chino o el japonés.

<

| Propiedades | Parámetros |
| --- | --- |
| **ruby-align** | *auto, start, left, center, end, right, distribute-letter, distribute-space, line-edge* |
| **ruby-overhang** | *auto, start, end, none* |
| **ruby-position** | *before, after, right, inline* |
| **ruby-span** | *attr[x], none* |

paged media

*Prioridad: Media | Última actualización realizada por el*[***W3 Consortium:***](http://www.w3.org/)*10/10/2006*

Este módulo permite ajustar los estilos de un documento web para ser visualizados con proyectores u otros dispositivos generales. Ya existían propiedades al respecto en CSS2 que se han desarrollado mas para CSS3, permite establecer cabeceras, pies de página, saltos de página, contadores de página, etc.

A modo de referencia se citan sus propiedades a continuación.

| Propiedades | Parámetros |
| --- | --- |
| [**fit**](http://theproc.es/page/guia_referencia_css3?layout=false) | *fill, hidden, meet, slice* |
| [**fit-position**](http://theproc.es/page/guia_referencia_css3?layout=false) | *[ top | center | bottom ], [ left | center | right ] longitud %* |
| [**image-orientation**](http://theproc.es/page/guia_referencia_css3?layout=false) | *[ top | center | bottom ], [ left | center | right ] longitud %* |
| [**orphans**](http://theproc.es/page/guia_referencia_css3?layout=false) | número |
| [**page**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto identificador* |
| [**page-break-after**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, always, avoid, left, right* |
| [**page-break-before**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, always, avoid, left, right* |
| [**page-break-inside**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, avoid* |
| [**size**](http://theproc.es/page/guia_referencia_css3?layout=false) | *auto, landscape, portrait  número* |
| [**windows**](http://theproc.es/page/guia_referencia_css3?layout=false) | número |

**Sec. 7 CASCADA, HERENCIA Y ESPECIFICIDAD**

**Fuente: https://developer.mozilla.org/es/docs/Learn/CSS/Building\_blocks/Cascade\_and\_inheritance#resumen**

El objetivo de este artículo es desarrollar la comprensión de algunos de los conceptos fundamentales de CSS (cascada, especificidad y herencia) que controlan cómo se aplica el CSS al HTML y cómo se resuelven los conflictos.

A medida que avances en este apartado verás que puede resultar menos relevante y un poco más académico que otros artículos, pero la comprensión de estas cuestiones te ahorrará problemas más adelante. Te animamos a que trabajes meticulosamente este apartado y verifiques que entiendes los conceptos antes de continuar.

## [Reglas conflictivas](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#reglas_conflictivas)

CSS significa **hojas de estilo en cascada** (cascading style sheets), y es muy importante entender la palabra cascada. La forma en que se comporta la cascada es la clave para comprender el CSS.

En algún momento trabajarás en un proyecto y encontrarás que el CSS que pensabas que debería aplicarse a un elemento no funciona. Por lo general, el problema suele ser que has creado dos normas que podrían aplicarse al mismo elemento. La **cascada**, y el concepto estrechamente relacionado de **especificidad** son mecanismos que controlan qué regla se aplica cuando aparecen tales conflictos. Es posible que la regla que se aplica finalmente a tu elemento no sea la que esperas, por lo que debes comprender cómo funcionan estos mecanismos.

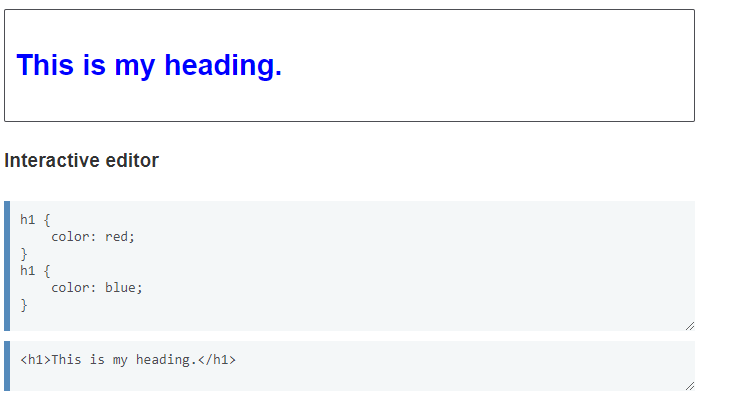
También es significativo el concepto de **herencia**, que significa que algunas propiedades CSS heredan por defecto los valores establecidos en el elemento padre, pero otras no. Esto también puede causar una respuesta diferente a la que esperas.

Vamos a empezar por echar un vistazo rápido a los principales elementos que nos interesan, y a continuación veremos cómo interactúan entre sí y con tu CSS. Pueden resultar un poco difíciles de entender, pero a medida que practiques escribiendo CSS te resultará más fácil de entender la manera cómo funcionan.

### [**Cascada**](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#cascada)

En un primer nivel de simplicidad, la **cascada** en las hojas de estilo significa que el orden de las reglas importa en CSS: cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el CSS.

En el ejemplo siguiente tenemos dos reglas que pueden aplicarse al h1. El h1 acaba siendo de color azul porque estas normas tienen un selector idéntico y, por lo tanto, tienen la misma especificidad. Por esta razón, se aplica la última que aparece.

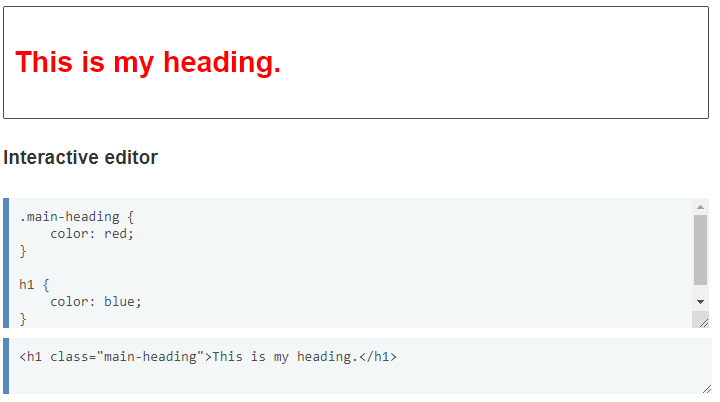


## Especificidad

La especificidad es el modo que tiene el navegador de decidir qué regla se aplica si diversas reglas tienen selectores diferentes, pero podrían aplicarse a un mismo elemento. Básicamente, la especificidad mide cuán específica es la selección de un selector:

* Un selector de elemento es menos específico (selecciona todos los elementos de aquel tipo que aparecen en la página) por lo que presenta una puntuación más baja en especificidad.
* Un selector de clase es más específico (selecciona solo los elementos de una página que tienen un valor de atributo “class” dado), y por tanto recibe una puntuación mayor.

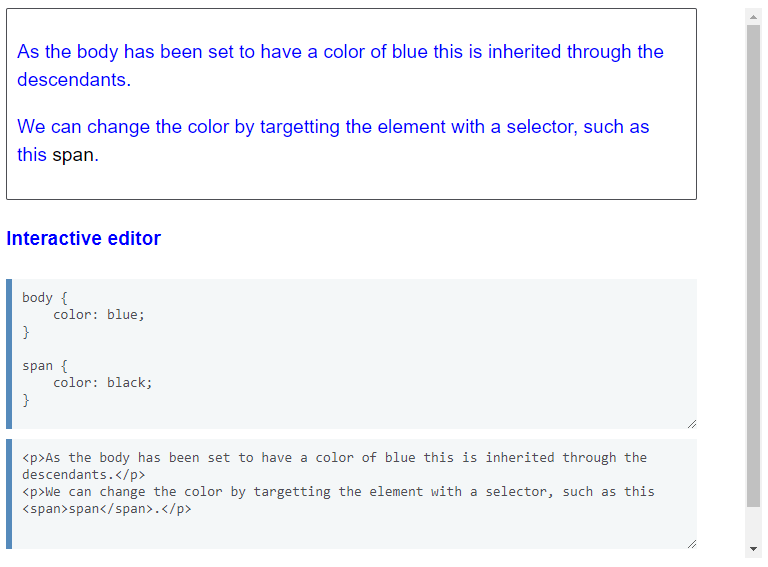
Veamos un ejemplo. Aquí abajo encontrarás dos reglas que pueden aplicarse al elemento h1. Este elemento h1 termina siendo de color rojo: el selector de clase confiere a esta regla una mayor especificidad, así que se aplicará a pesar de la regla para el selector de elemento que aparece más abajo en el orden del código.



### [**Herencia**](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#herencia)

La herencia también debe entenderse en este contexto: algunos valores de las propiedades CSS que se han establecido para los elementos padre los heredan los elementos hijo, pero otros no.

Por ejemplo, si para un elemento se establece el color (color) y el tipo de letra (font-family), cada elemento que se encuentre dentro de él también se mostrará de ese color y con ese tipo de letra, a menos que les se haya aplicado un color y un tipo de letra diferentes directamente



Algunas propiedades no se heredan. Por ejemplo, si para un elemento se establece un ancho [width](https://developer.mozilla.org/es/docs/Web/CSS/width) del 50%, sus descendientes no tendrán un 50% de ancho con respecto al de sus padres. Si este fuera el caso, ¡sería muy frustrante usar CSS!

## [Comprender cómo trabajan juntos estos conceptos](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#comprender_c%C3%B3mo_trabajan_juntos_estos_conceptos)

Estos tres conceptos controlan qué CSS se aplica a qué elemento. En las secciones siguientes veremos cómo funcionan en conjunto. A veces puede parecer un poco complicado, pero lo irás recordando a medida que ganes experiencia con el CSS, y siempre puedes consultar los detalles si se te olvidan. ¡Incluso los desarrolladores experimentados lo hacen!

## [Comprender la herencia](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#comprender_la_herencia)

Vamos a empezar con la herencia. En el ejemplo siguiente tenemos un elemento [<ul>](https://developer.mozilla.org/es/docs/Web/HTML/Element/ul) con dos niveles de listas no ordenadas anidadas en él. Hemos establecido para el <ul> exterior un borde, un relleno y un color de fuente.

El color se ha aplicado a los hijos directos y también a los hijos indirectos: los elementos hijo <li> inmediatos y los que están dentro de la primera lista. A continuación, hemos añadido a la segunda lista anidada una clase especial y le hemos aplicado un color diferente, que los elementos hijo de esta heredarán.





La anchura (como se mencionó anteriormente), los márgenes, el relleno y los bordes no se heredan. Si los elementos hijo de nuestra lista heredaran los bordes, todas las listas y los elementos de lista ganarían un borde cada vez ¡y no es probable que vez quieras un efecto así!

Las propiedades que se heredan por defecto y las que no son cuestión, en gran medida, de sentido común.

### [**Control de la herencia**](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#control_de_la_herencia)

CSS proporciona cuatro valores de propiedad universales especiales para el control de la herencia. Todas las propiedades CSS aceptan estos valores.

[**inherit**](https://developer.mozilla.org/es/docs/Web/CSS/inherit)

Establece que el valor de la propiedad que se aplica a un elemento determinado sea exactamente igual al del elemento padre. En la práctica, esto "activa la herencia".

[**initial**](https://developer.mozilla.org/es/docs/Web/CSS/initial)

Establece que el valor de la propiedad que se aplica a un elemento seleccionado tenga el mismo valor que esté establecido para esa propiedad en la hoja de estilo por defecto del navegador.

[**unset (en-US)**](https://developer.mozilla.org/en-US/docs/Web/CSS/unset)

Restablece la propiedad a su valor natural, lo que significa que, si la propiedad se hereda de forma natural, actúa como inherit, y en caso contrario como initial.

A continuación, veremos una lista de enlaces y exploraremos cómo funcionan los valores universales. El ejemplo en vivo de abajo te permite jugar con el CSS y ver lo que sucede cuando se hacen cambios. Jugar con el código es la mejor forma de enfrentarse al HTML y el CSS.

Por ejemplo:

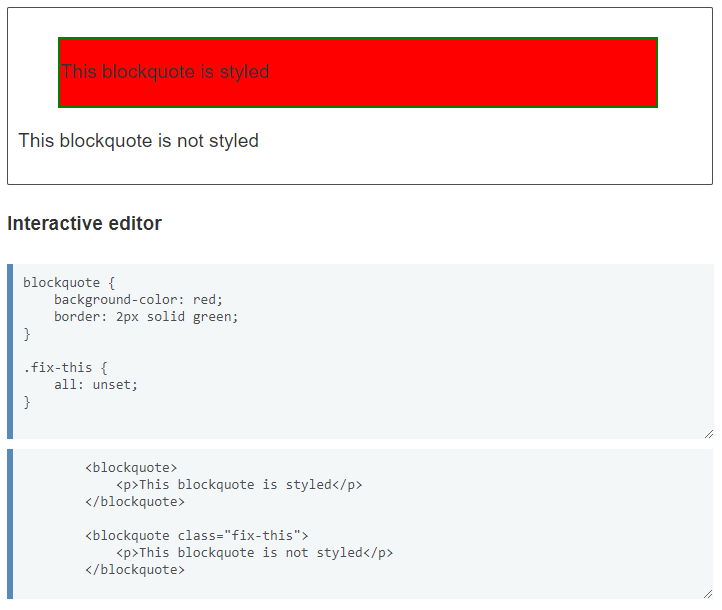
1. Se ha aplicado la clase my-class-1 al segundo elemento de lista. Esto establece por herencia el color del elemento <a> que está anidado en él. ¿Cómo cambia el color del enlace si quitamos esta regla?
2. ¿Entiendes por qué el tercer y el cuarto enlace se ven de este color? En caso contrario, comprueba la descripción de los valores anteriores.
3. ¿Cuál de los enlaces va a cambiar de color si se define un nuevo color para el elemento <a>, por ejemplo, a { color: red; }?



### [**Restablecer todos los valores de propiedad**](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#restablecer_todos_los_valores_de_propiedad)

La propiedad CSS abreviada all se puede utilizar para aplicar uno de estos valores de herencia a (casi) todas las propiedades a la vez. Su valor puede ser cualquiera de los valores de herencia (inherit, initial, unset, o revert). Es una forma práctica de deshacer los cambios realizados respecto al estilo para que puedas volver a un punto de partida conocido antes de empezar a introducir cambios.

En el ejemplo siguiente hay dos bloques de cita. El primero ya tiene un estilo aplicado al propio elemento de cita, mientras que el segundo tiene una clase aplicada al bloque de cita que establece el valor all en unset.



## [Comprender la cascada](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#comprender_la_cascada)

Ahora entendemos por qué un párrafo que está anidado en la estructura del HTML es del mismo color que el CSS aplicado al cuerpo (body) del HTML y, a partir de los artículos de introducción sabemos cómo cambiar el CSS aplicado a algo en cualquier parte del documento, ya sea mediante la asignación de CSS a un elemento o la creación de una clase. Ahora vamos a echar un vistazo a la forma en que el concepto de cascada define qué reglas CSS se aplican cuando más de un elemento de estilo puede aplicar estilo a un elemento.

Hay que considerar tres factores, que se enumeran a continuación en orden de importancia creciente. Los posteriores invalidan los anteriores:

1. **Orden en el código**
2. **Especificidad**
3. **Importancia**

Vamos a explicarlos para ver cómo los navegadores determinan exactamente que CSS deben aplicar.

### [**Orden en el código**](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#orden_en_el_c%C3%B3digo)

Ya hemos visto cómo el orden en el código es importante en el concepto de cascada. Si tienes más de una regla con exactamente el mismo peso, la que ocupa el último lugar en el CSS gana. Puedes entenderlo como que las reglas que están más cerca del elemento considerado sobreescriben las anteriores hasta que la última gana y da formato al elemento.

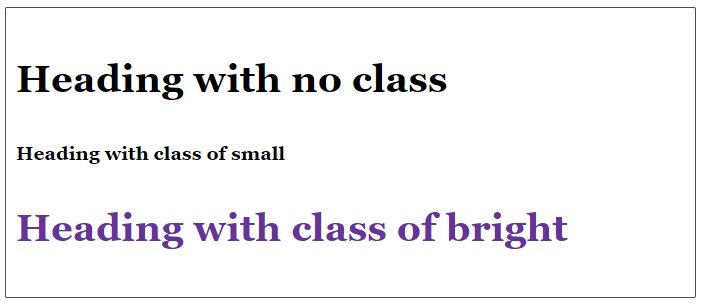
### [**Especificidad**](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#especificidad_2)

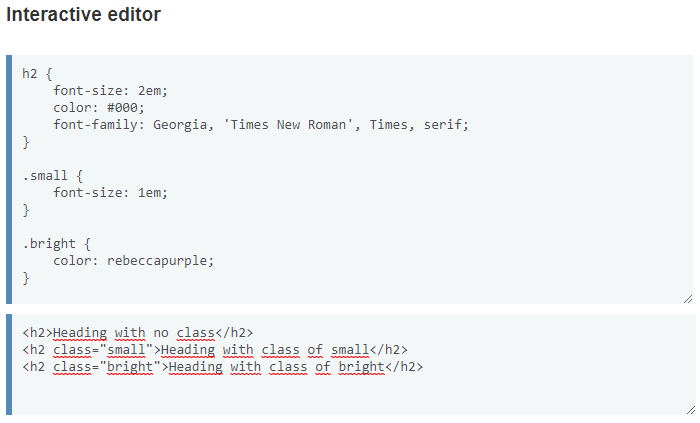
Una vez entendido el hecho de que el orden de los elementos en el código es importante, te encontrarás en alguna situación en la que sabes cuál es la última norma en la hoja de estilo, pero se aplica una regla anterior. Esto se debe a que la regla anterior tiene **una especificidad mayor**, es decir, es más específica y, por lo tanto, el navegador la escoge como la que debe dar forma al elemento.

Como hemos visto anteriormente en este mismo artículo, un selector de clase tiene más peso que un selector de elemento, por lo que las propiedades que se definen en la clase tienen prioridad sobre las que se aplican directamente en el elemento.

Un elemento que hay que tener en cuenta es que aunque pensamos en términos de selectores y reglas que se aplican a lo que estos seleccionan, no es toda la regla lo que se sobrescribe, sino solo las propiedades que entran en conflicto.

Este comportamiento ayuda a evitar repeticiones en el CSS. Una práctica común es definir estilos genéricos para los elementos básicos y luego, crear clases para los elementos que son diferentes. Por ejemplo, en la hoja de estilo que mostramos a continuación hemos definido estilos genéricos para los encabezados de nivel 2; posteriormente hemos creado algunas clases que solo cambian algunas de las propiedades y los valores. Los valores definidos inicialmente se aplican a todos los encabezados, y entonces los valores más específicos se aplican a los encabezados con las clases.





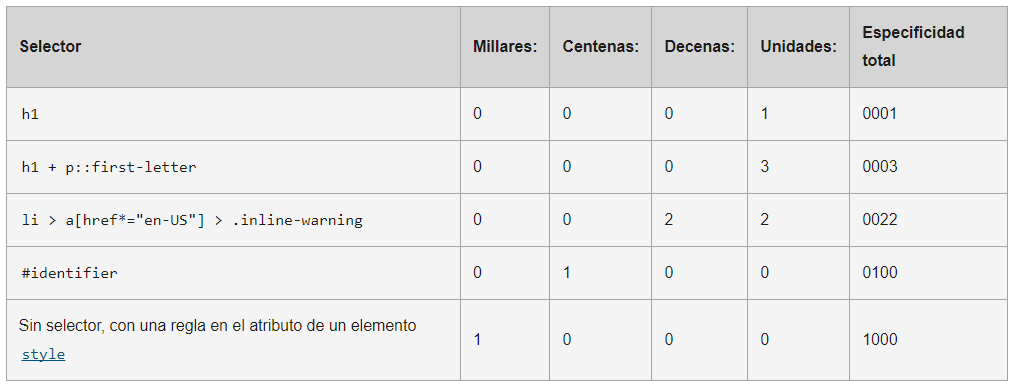
Ahora vamos a echar un vistazo a cómo el navegador calcula la especificidad. Ya sabemos que un selector de elemento tiene una especificidad baja y se puede sobrescribir con un elemento de clase. Esencialmente se otorga un valor de puntos a los diferentes tipos de selectores y la suma de estos establece la importancia de ese selector en particular, que a continuación puede evaluarse ante otras posibles coincidencias.

La cantidad de especificidad de un selector se mide usando cuatro valores diferentes (o componentes), que pueden describirse como millares, centenas, decenas y unidades (cuatro dígitos individuales dispuestos en cuatro columnas):

1. **Millares**: Se suma un punto en esta columna si la declaración está en un atributo de [style](https://developer.mozilla.org/es/docs/Web/HTML/Global_attributes#attr-style) o, como suelen denominarse, estilos en línea. Tales declaraciones no tienen selectores, por lo que su especificidad siempre es 1000.
2. **Centenas**: Se suma un punto en esta columna por cada selector con ID particular que esté contenido en el selector general.
3. **Decenas**: Se suma un punto en esta columna por cada selector de clase, de atributo o pseudoclase que estén contenidos en el selector general.
4. **Unidades**: Se suma un punto en esta columna por cada selector o pseudoelemento que esté contenido en el selector general.

La tabla siguiente muestra algunos ejemplos concretos para ayudarte a entenderlo mejor. Analízalos y trata de entender por qué tienen la especificidad que les hemos dado. Aún no hemos explicado los selectores de forma detallada, pero puedes encontrar detalles de cada selector en los selectores de referencia) de MDN.

(https://developer.mozilla.org/es/docs/Web/CSS/CSS\_Selectors).



**Sec. 8 UNIDADES DE MEDIDA Y COLORES**

Muchas de las propiedades de CSS que se ven en los próximos capítulos permiten indicar medidas y colores en sus valores. Además, CSS es tan flexible que permite indicar las medidas y colores de muchas formas diferentes.

Por este motivo, se presentan a continuación todas las alternativas disponibles en CSS para indicar las medidas y los colores. En los siguientes capítulos, cuando una propiedad pueda tomar como valor una medida o un color, no se volverán a explicar todas estas alternativas.

**Unidades de medida**

Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto. Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida).

CSS divide las unidades de medida en dos grupos: absolutas y relativas. Las medidas relativas definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado. Las unidades absolutas establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.

Si el valor es 0, la unidad de medida es opcional. Si el valor es distinto a 0 y no se indica ninguna unidad, la medida se ignora completamente, lo que suele ser uno de los errores más habituales de los diseñadores que empiezan con CSS. Algunas propiedades permiten indicar medidas negativas, aunque habitualmente sus valores son positivos. Si el valor decimal de una medida es inferior a 1, se puede omitir el 0 de la izquierda (0.5em es equivalente a .5em).

**3.1.1. Unidades absolutas**

Una medida indicada mediante unidades absolutas está completamente definida, ya que su valor no depende de otro valor de referencia. A continuación se muestra la lista completa de unidades absolutas definidas por CSS y su significado:

* in, pulgadas ("inches", en inglés). Una pulgada equivale a 2.54 centímetros.
* cm, centímetros.
* mm, milímetros.
* pt, puntos. Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros.
* pc, picas. Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros.

A continuación se muestran ejemplos de utilización de unidades absolutas:

*/\* El cuerpo de la página debe mostrar un margen de media pulgada \*/*

body { margin: 0.5in; }

*/\* Los elementos <h1> deben mostrar un interlineado de 2 centímetros \*/*

h1 { line-height: 2cm; }

*/\* Las palabras de todos los párrafos deben estar separadas 4 milímetros entre si \*/*

p { word-spacing: 4mm; }

*/\* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos \*/*

a { font-size: 12pt }

*/\* Los elementos <span> deben tener un tamaño de letra de 1 pica \*/*

span { font-size: 1pc }

La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios. Su principal desventaja es que son muy poco flexibles y no se adaptan fácilmente a los diferentes medios.

De todas las unidades absolutas, la única que suele utilizarse es el punto (pt). Se trata de la unidad de medida preferida para establecer el tamaño del texto en los documentos que se van a imprimir, es decir, para el medio print de CSS, tal y como se verá más adelante.

**3.1.2. Unidades relativas**

La unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios.

A continuación se muestran las tres unidades de medida relativas definidas por CSS y la referencia que toma cada una para determinar su valor real:

* em, (no confundir con la etiqueta <em> de HTML) relativa respecto del tamaño de letra del elemento.
* ex, relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento.
* px, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

Las unidades em y ex no han sido creadas por CSS, sino que llevan décadas utilizándose en el campo de la tipografía. Aunque no es una definición exacta, la unidad 1em equivale a la anchura de la letra M ("eme mayúscula") del tipo y tamaño de letra del elemento.

La unidad em hace referencia al tamaño en puntos de la letra que se está utilizando. Si se utiliza una tipografía de 12 puntos, 1em equivale a 12 puntos. El valor de 1ex se puede aproximar por 0.5 em.

Si se considera el siguiente ejemplo:

p { margin: 1em; }

La regla CSS anterior indica que los párrafos deben mostrar un margen de anchura igual a 1em. Como se trata de una unidad de medida relativa, es necesario realizar un cálculo matemático para determinar la anchura real de ese margen.

La unidad de medida em siempre hace referencia al tamaño de letra del elemento. Por otra parte, todos los navegadores muestran por defecto el texto de los párrafos con un tamaño de letra de 16 píxel. Por tanto, en este caso el margen de 1em equivale a un margen de anchura 16px.

A continuación se modifica el ejemplo anterior para cambiar el tamaño de letra de los párrafos:

p { font-size: 32px; margin: 1em; }

El valor del margen sigue siendo el mismo en unidades relativas (1em) pero su valor real ha variado porque el tamaño de letra de los párrafos ha variado. En este caso, el margen tendrá una anchura de 32px, ya que 1em siempre equivale al tamaño de letra del elemento.

Si se quiere reducir la anchura del margen a 16px pero manteniendo el tamaño de letra de los párrafos en 32px, se debe utilizar la siguiente regla CSS:

p { font-size: 32px; margin: 0.5em; }

El valor 0.5em se interpreta como "la mitad del tamaño de letra del elemento", ya que se debe multiplicar por 0.5 su tamaño de letra (32px x 0.5 = 16px). De la misma forma, si se quiere mostrar un margen de 8px de anchura, se debería utilizar el valor 0.25em, ya que 32px x 0.25 = 8px.

La gran ventaja de las unidades relativas es que siempre mantienen las proporciones del diseño de la página. Establecer el margen de un elemento con el valor 1em equivale a indicar que "el margen del elemento debe ser del mismo tamaño que su letra y debe cambiar proporcionalmente".

En efecto, si el tamaño de letra de un elemento aumenta hasta un valor enorme, su margen de 1em también será enorme. Si su tamaño de letra se reduce hasta un valor diminuto, el margen de 1em también será diminuto. El uso de unidades relativas permite mantener las proporciones del diseño cuando se modifica el tamaño de letra de la página.

El funcionamiento de la unidad ex es idéntico a em, salvo que en este caso, la referencia es la altura de la letra x minúscula, por lo que su valor es aproximadamente la mitad que el de la unidad em.

Por último, las medidas indicadas en píxel también se consideran relativas, ya que el aspecto de los elementos dependerá de la resolución del dispositivo en el que se visualiza la página HTML. Si un elemento tiene una anchura de 400px, ocupará la mitad de una pantalla con una resolución de 800x600, pero ocupará menos de la tercera parte en una pantalla con resolución de 1440x900.

Las unidades de medida se pueden mezclar en los diferentes elementos de una misma página, como en el siguiente ejemplo:

body { font-size: 10px; }

h1 { font-size: 2.5em; }

En primer lugar, se establece un tamaño de letra base de 10 píxel para toda la página. A continuación, se asigna un tamaño de 2.5em al elemento <h1>, por lo que su tamaño de letra real será de 2.5 x 10px = 25px.

Como se vio en los capítulos anteriores, el valor de la mayoría de propiedades CSS se hereda de padres a hijos. Así por ejemplo, si se establece el tamaño de letra al elemento <body>, todos los elementos de la página tendrán el mismo tamaño de letra, salvo que indiquen otro valor.

Sin embargo, el valor de las medidas relativas no se hereda directamente, sino que se hereda su valor real una vez calculado. El siguiente ejemplo muestra este comportamiento:

body {

font-size: 12px;

text-indent: 3em;

}

h1 { font-size: 15px }

La propiedad text-indent, como se verá en los próximos capítulos, se utiliza para tabular la primera línea de un texto. El elemento <body> define un valor para esta propiedad, pero el elemento <h1> no lo hace, por lo que heredará el valor de su elemento padre. Sin embargo, el valor heredado no es 3em, sino 36px.

Si se heredara el valor 3em, al multiplicarlo por el valor de font-size del elemento <h1> (que vale 15px) el resultado sería 3em x 15px = 45px. No obstante, como se ha comentado, los valores que se heredan no son los relativos, sino los valores ya calculados.

Por lo tanto, en primer lugar se calcula el valor real de 3em para el elemento <body>: 3em x 12px = 36px. Una vez calculado el valor real, este es el valor que se hereda para el resto de elementos.

**3.1.3. Porcentajes**

El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a em, ex y px. Un porcentaje está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida. Cada una de las propiedades de CSS que permiten indicar como valor un porcentaje, define el valor al que hace referencia ese porcentaje.

Los porcentajes se pueden utilizar por ejemplo para establecer el valor del tamaño de letra de los elementos:

body { font-size: 1em; }

h1 { font-size: 200%; }

h2 { font-size: 150%; }

Los tamaños establecidos para los elementos <h1> y <h2> mediante las reglas anteriores, son equivalentes a 2em y 1.5em respectivamente, por lo que es más habitual definirlos mediante em.

Los porcentajes también se utilizan para establecer la anchura de los elementos:

div#contenido { width: 600px; }

div.principal { width: 80%; }

<div id="contenido">

<div class="principal">

...

</div>

</div>

En el ejemplo anterior, la referencia del valor 80% es la anchura de su elemento padre. Por tanto, el elemento <div> cuyo atributo class vale principal tiene una anchura de 80% x 600px = 480px.

**3.1.4. Recomendaciones**

En general, se recomienda el uso de unidades relativas siempre que sea posible, ya que mejora la accesibilidad de la página y permite que los documentos se adapten fácilmente a cualquier medio y dispositivo.

El documento [Recomendaciones sobre técnicas CSS para la mejora de la accesibilidad de los contenidos HTML](https://www.w3.org/TR/WCAG10-CSS-TECHS/), elaborado por el organismo W3C, recomienda el uso de la unidad em para indicar el tamaño del texto y para todas las medidas que sean posibles.

Normalmente se utilizan píxel y porcentajes para definir el layout del documento (básicamente, la anchura de las columnas y de los elementos de las páginas) y em y porcentajes para el tamaño de letra de los textos.

**Colores**

Los colores en CSS se pueden indicar de cinco formas diferentes: palabras clave, colores del sistema, RGB hexadecimal, RGB numérico y RGB porcentual. Aunque el método más habitual es el del RGB hexadecimal, a continuación se muestran todas las alternativas que ofrece CSS.

**3.2.1. Palabras clave**

CSS define 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:

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



**Figura 3.1** Colores definidos mediante las palabras clave de CSS

La imagen anterior ha sido extraída de la [sección sobre colores de la especificación oficial de CSS](https://www.w3.org/TR/CSS21/syndata.html#value-def-color).

Aunque es una forma muy sencilla de referirse a los colores básicos, este método prácticamente no se utiliza en las hojas de estilos de los sitios web reales, ya que se trata de una gama de colores muy limitada.

Además de la lista básica, los navegadores modernos soportan muchos otros nombres de colores. La lista completa se puede ver en [en.wikipedia.org/wiki/Websafe](https://en.wikipedia.org/wiki/Websafe).

**3.2.2. RGB decimal**

En el campo del diseño gráfico, se han definido varios modelos para hacer referencia a los colores. Los dos modelos más conocidos son RGB y CMYK. Simplificando su explicación, el modelo RGB consiste en definir un color indicando la cantidad de color rojo, verde y azul que se debe mezclar para obtener ese color. Técnicamente, el modelo RGB es un modelo de tipo "aditivo", ya que los colores se obtienen sumando sus componentes.

Por lo tanto, en el modelo RGB un color se define indicando sus tres componentes R (rojo), G (verde) y B (azul). Cada una de las componentes puede tomar un valor entre cero y un valor máximo. De esta forma, el color rojo puro en RGB se crea mediante el máximo valor de la componente R y un valor de 0 para las componentes G y B.

Si todas las componentes valen 0, el color creado es el negro y si todas las componentes toman su valor máximo, el color obtenido es el blanco. En CSS, las componentes de los colores definidos mediante RGB decimal pueden tomar valores entre 0 y 255. El siguiente ejemplo establece el color del texto de un párrafo:

p { color: rgb(71, 98, 176); }

La sintaxis que se utiliza para indicar los colores es rgb() y entre paréntesis se indican las tres componentes RGB, en ese mismo orden y separadas por comas. El color del ejemplo anterior se obtendría mezclando las componentes R=71, G=98, B=176, que se corresponde con un color azul claro.

Si se indica un valor menor que 0 para una componente, automáticamente se transforma su valor en 0. Igualmente, si se indica un valor mayor que 255, se transforma automáticamente su valor a 255.

**3.2.3. RGB porcentual**

Las componentes RGB de un color también se pueden indicar mediante un porcentaje. El funcionamiento y la sintaxis de este método es el mismo que el del RGB decimal. La única diferencia es que en este caso el valor de las componentes RGB puede tomar valores entre 0% y 100%. Por tanto, para transformar un valor RGB decimal en un valor RGB porcentual, es preciso realizar una regla de tres considerando que 0 es igual a 0% y 255 es igual a 100%.

El mismo color del ejemplo anterior se puede representar de forma porcentual:

p { color: rgb(27%, 38%, 69%); }

Al igual que sucede con el RGB decimal, si se indica un valor inferior a 0%, se transforma automáticamente en 0% y si se indica un valor superior a 100%, se trunca su valor a 100%.

**3.2.4. RGB hexadecimal**

Aunque es el método más complicado para indicar los colores, se trata del método más utilizado con mucha diferencia. De hecho, prácticamente todos los sitios web reales utilizan exclusivamente este método.

Para entender el modelo RGB hexadecimal, en primer lugar es preciso introducir un concepto matemático llamado sistema numérico hexadecimal. Cuando realizamos operaciones matemáticas, siempre utilizamos 10 símbolos para representar los números (del 0 al 9). Por este motivo, se dice que utilizamos un sistema numérico decimal.

No obstante, el sistema decimal es solamente uno de los muchos sistemas numéricos que se han definido. Entre los sistemas numéricos alternativos más utilizados se encuentra el sistema hexadecimal, que utiliza 16 símbolos para representar sus números.

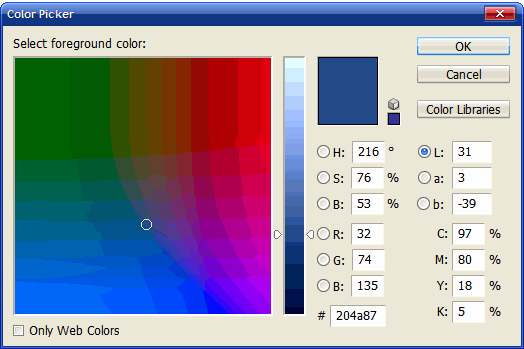
Como sólo conocemos 10 símbolos numéricos, el sistema hexadecimal utiliza también seis letras (de la A a la F) para representar los números. De esta forma, en el sistema hexadecimal, después del 9 no va el 10, sino la A. La letra B equivale al número 11, la C al 12, la D al 13, la E al 14 y la F al número 15.

Definir un color en CSS con el método RGB hexadecimal requiere realizar los siguientes pasos: - Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176 - Transformar el valor decimal de cada componente al sistema numérico hexadecimal. Se trata de una operación exclusivamente matemática, por lo que puedes utilizar una calculadora. En el ejemplo anterior, el valor hexadecimal de cada componente es: R = 47, G = 62, B = B0 - Para obtener el color completo en formato RGB hexadecimal, se concatenan los valores hexadecimales de las componentes RGB en ese orden y se les añade el prefijo #. De esta forma, el color del ejemplo anterior es #4762B0 en formato RGB hexadecimal.

Siguiendo el mismo ejemplo de las secciones anteriores, el color del párrafo se indica de la siguiente forma utilizando el formato RGB hexadecimal:

p { color: #4762B0; }

Recuerda que aunque es el método más complicado para definir un color, se trata del método que utilizan la inmensa mayoría de sitios web, por lo que es imprescindible dominarlo. Afortunadamente, todos los programas de diseño gráfico convierten de forma automática los valores RGB decimales a sus valores RGB hexadecimales, por lo que no tienes que hacer ninguna operación matemática:



**Figura 3.2** Herramienta de color de Photoshop para definir los colores según los modelos RGB, CMYK, Lab, HSB y RGB hexadecimal

El formato RGB hexadecimal es la forma más compacta de indicar un color, ya que incluso es posible comprimir sus valores cuando todas sus componentes son iguales dos a dos:

#AAA = #AAAAAA

#FFF = #FFFFFF

#A0F = #AA00FF

#369 = #336699

En el siguiente ejemplo se establece el color de fondo de la página a blanco, el color del texto a negro y el color de la letra de los titulares se define de color rojo:

body { background-color: #FFF; color: #000; }

h1, h2, h3, h4, h5, h6 { color: #C00; }

Las letras que forman parte del color en formato RGB hexadecimal se pueden escribir en mayúsculas o minúsculas indistintamente. No obstante, se recomienda escribirlas siempre en mayúsculas o siempre en minúsculas para que la hoja de estilos resultante sea más limpia y homogénea.

**3.2.5. Colores del sistema**

Los colores del sistema son similares a los colores indicados mediante su nombre, pero en este caso hacen referencia al color que muestran algunos elementos del sistema operativo del usuario.

Existen varios colores definidos, como por ejemplo ActiveBorder, que hace referencia al color del borde de las ventanas activas. Consulta la [lista completa de colores del sistema](https://www.w3.org/TR/CSS21/ui.html#system-colors).

Aunque es posible definir los colores en CSS utilizando estos nombres, se trata de un método que nunca se utiliza, por lo que se puede considerar prácticamente como una rareza de CSS.

**3.2.6. Colores web safe**

Como cada componente RGB de los colores puede tomar un valor entre 0 y 255, el número total de colores que se pueden representar con este formato es de 256 x 256 x 256 = 16.777.216 colores. Sin embargo, en la década de los 90 los monitores de los usuarios no eran capaces de mostrar más de 256 colores diferentes.

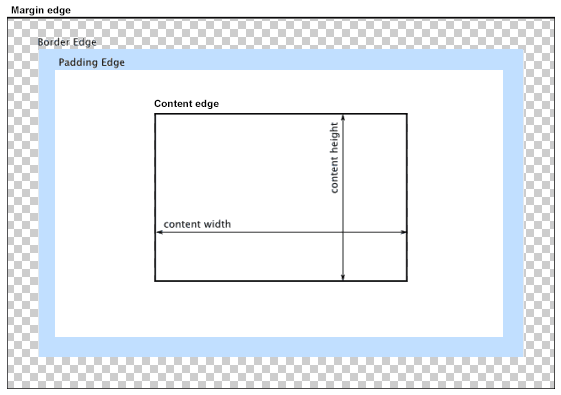
A partir de todos los colores disponibles, se eligieron 216 colores que formaron la paleta de colores "web safe". Esta paleta de colores podía ser utilizada por los diseñadores con la seguridad de que se verían correctamente en cualquier navegador de cualquier sistema operativo de cualquier usuario.

Hoy en día, su importancia ha descendido notablemente, ya que prácticamente todos los usuarios utilizan dispositivos con una profundidad de color de 16 y 32 bits. No obstante, el auge en el uso de los dispositivos móviles hace que siga siendo un tema a considerar, ya que las pantallas de muchos móviles sólo pueden representar un número reducido de colores.

**Sec. 9 INTRODUCCIÓN AL MODELO DE CAJA DE CSS**

Al diseñar un documento, el motor de representación del navegador representa cada elemento como un cuadro rectangular según el estándar **modelo de caja de CSS**. CSS determina el tamaño, la posición y las propiedades (color, fondo, tamaño del borde, etc.) de estos cuadros.

Cada caja se compone de cuatro partes (o áreas), definidas por sus respectivos límites: el límite del contenido, el límite del relleno (padding), el límite del borde y el límite del margen.



El **área de contenido**, delimitada por el límite del contenido, contiene el contenido "real" del elemento, como lo puede ser texto, imagen o un reproductor de video. Sus dimensiones son el ancho del contenido (o el ancho de la caja de contenido) y la altura del contenido (o la altura de la caja de contenido). A menudo tiene un color de fondo o una imagen de fondo.

Si la propiedad [box-sizing](https://developer.mozilla.org/es/docs/Web/CSS/box-sizing) está configurada en content-box (default), el tamaño del área de contenido se puede definir explícitamente con las propiedades de [width](https://developer.mozilla.org/es/docs/Web/CSS/width), [min-width](https://developer.mozilla.org/es/docs/Web/CSS/min-width), [max-width (en-US)](https://developer.mozilla.org/en-US/docs/Web/CSS/max-width), [height](https://developer.mozilla.org/es/docs/Web/CSS/height), [min-height](https://developer.mozilla.org/es/docs/Web/CSS/min-height) y [max-height](https://developer.mozilla.org/es/docs/Web/CSS/max-height).

El **área de relleno** (padding), delimitada por el límite del relleno, extiende el área de contenido para incluir el relleno del elemento. Sus dimensiones son el ancho de la caja de relleno y la altura de la caja de relleno. Cuando el área de contenido tiene un fondo, se extiende dentro del relleno.

El espesor del relleno está determinado por las propiedades [padding-top](https://developer.mozilla.org/es/docs/Web/CSS/padding-top), [padding-right (en-US)](https://developer.mozilla.org/en-US/docs/Web/CSS/padding-right), [padding-bottom](https://developer.mozilla.org/es/docs/Web/CSS/padding-bottom), [padding-left (en-US)](https://developer.mozilla.org/en-US/docs/Web/CSS/padding-left), y la propiedad abreviada [padding](https://developer.mozilla.org/es/docs/Web/CSS/padding).

El **área del borde**, delimitada por el límite del borde, extiende el área de relleno para incluir los bordes del elemento. Sus dimensiones son el ancho de la caja del borde y la altura de la caja del borde.

El espesor de los bordes está determinado por las propiedades [border-width](https://developer.mozilla.org/es/docs/Web/CSS/border-width) y la propiedad abreviada [border](https://developer.mozilla.org/es/docs/Web/CSS/border). Si la propiedad [box-sizing](https://developer.mozilla.org/es/docs/Web/CSS/box-sizing) se establece en border-box, el tamaño del área del borde se puede definir explícitamente con las propiedades [width](https://developer.mozilla.org/es/docs/Web/CSS/width), min- width, [max-width (en-US)](https://developer.mozilla.org/en-US/docs/Web/CSS/max-width), [height](https://developer.mozilla.org/es/docs/Web/CSS/height), [min-height](https://developer.mozilla.org/es/docs/Web/CSS/min-height), y max-height.

El **área del margen**, delimitada por el límite del margen, extiende el área del borde para incluir un área vacía utilizada para separar el elemento de sus vecinos. Sus dimensiones son el ancho de la caja del margen y la altura de la caja del margen.

El tamaño del área del margen está determinado por las propiedades [margin-top (en-US)](https://developer.mozilla.org/en-US/docs/Web/CSS/margin-top), [margin-right](https://developer.mozilla.org/es/docs/Web/CSS/margin-right), [margin-bottom](https://developer.mozilla.org/es/docs/Web/CSS/margin-bottom), { {cssxref("margin-left")}}, y la propiedad abreviada [margin](https://developer.mozilla.org/es/docs/Web/CSS/margin). Cuando se produce el [colapso del margen](https://developer.mozilla.org/en-US/CSS/margin_collapsing), el área del margen no está claramente definida ya que los márgenes se comparten entre las cajas.

Finalmente, ten en cuenta que para elementos en línea no reemplazados, la cantidad de espacio ocupado (la contribución a la altura de la línea) está determinada por la propiedad [line-height](https://developer.mozilla.org/es/docs/Web/CSS/line-height), aunque los bordes y el relleno todavía se muestran alrededor del contenido.

**RESUMIR CÓDIGO CSS**

Siempre es conveniente escribir código resumido para lograr un archivo CSS más corto y más liviano.

#### Resumir los 4 lados:

Algunas propiedades como el margin, padding, border, hacen referencia a los 4 lados de un elemento (top, right, bottom, left, en ese orden.). Por cada uno de los lados, existe una propiedad específica, por ejemplo: margin-top, margin-right, margin-bottom, margin-left. Pero podemos resumir el código si respetamos el orden de los valores.

**Ejemplo 1**:

Si tenemos que definir los 4 padding iguales podemos escribir las 4 propiedades o resumir:

{

padding-top: 20px;

padding-right: 20px;

padding-bottom: 20px;

padding-left: 20px

}

**RESUMIDO**

{

padding: 20px

}

**Ejemplo 2**:

También podemos escribir los 4 padding distintos

{

padding-top: 40px;

padding-right: 25px;

padding-bottom: 20px;

padding-left: 35px

}

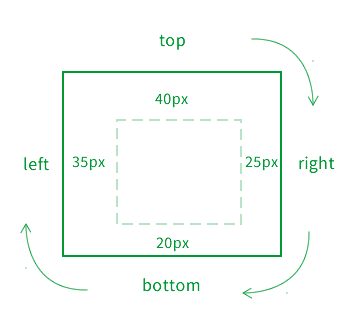
**RESUMIDO**

{

padding: 40px 25px 20px 35px

}

Es necesario respetar el orden: top, right, bottom, left:



**Ejemplo 3**:

Si tenemos que definir 2 padding de 20px y 2 de 40px, podemos escribir:

{

padding-top: 20px;

padding-right: 40px;

padding-bottom: 20px;

padding-left: 40px

}

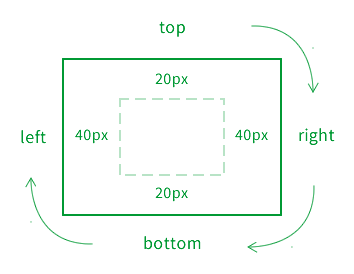
**RESUMIDO**

{

padding: 20px 40px

}

El navegador reconoce que el primer valor corresponde a top y bottom, y el segundo a left y right.



#### Resumir color:

Si un color en sistema hexadecimal se compone de 3 pares de dígitos iguales, los podemos resumir en sólo 3 dígitos, por ejemplo: **{background: #330099}**, es lo mismo que **{background: #309}**

#### Resumir background:

Si queremos usar una imagen de fondo que no se repita y que esté centrada y marginada arriba, podemos escribirla con 3 propiedades separadas o todos los valores en una sola propiedad:

{

background-image: url(foto.jpg);

background-repeat: no-repeat;

background-position: top center;

}

**RESUMIDO**

{

background: url(foto.jpg) no-repeat top center;

}

#### Resumir border:

Para agregar un borde tenemos que definir el grosor, el estilo y el color. Podemos escribir las 3 propiedades o resumirlas:

{

border-width: 1px;

border-style: solid;

border-color: #666

}

**RESUMIDO**

{border: 1px solid #666}

**Sec. 10 RECURSOS PARA EL DISEÑADOR WEB**

**[Herramientas para el DISEÑO WEB HTML y CSS]**

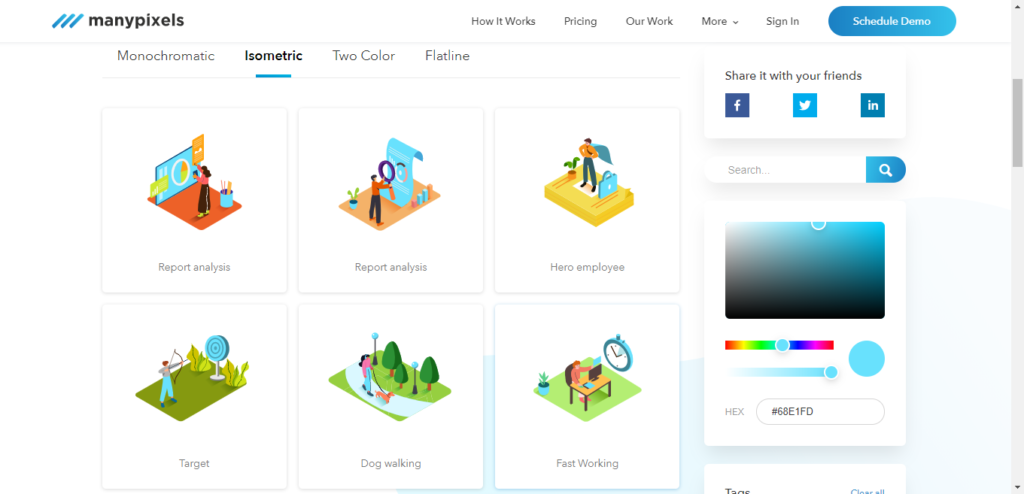
**Fuente:** Blog de AlexCGDesin (recomiendo ampliamente su canal de youtube)

Bienvenido a este artículo, donde te resumiré cuales son mis herramientas o recursos para el diseño web, estos son los que yo más utilizo y con los que les doy un toque genial y diferente a mis diseños, ¿Quieres conocerlo?, Quédate aquí…

 Pero antes de empezar, ¿Ya viste el vídeo donde enseño como usar estas herramientas? Aquí te lo dejo

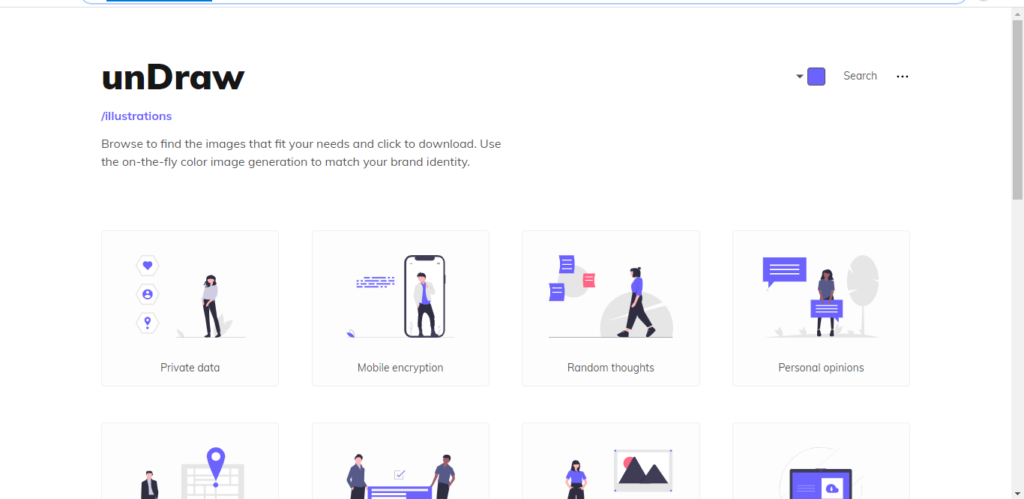
¿Deseas verlo en Youtube?

**Herramienta N° 1 Many Pixels**



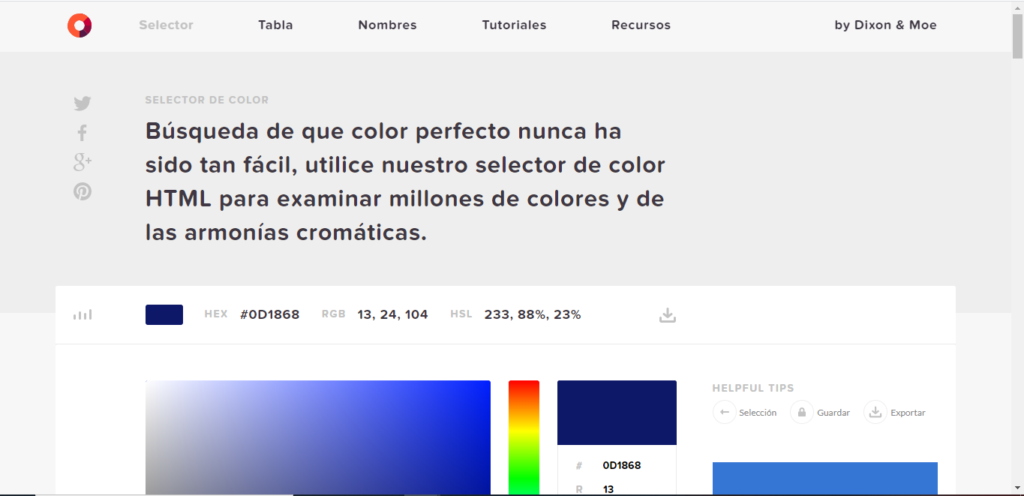
Many pixels es un recurso para crear diseños web increíbles debido a que Many pixels, nos ofrece bastantes ilustraciones svg para usar en nuestros proyectos de manera gratuita, además que tiene distintos tipos de ilustraciones, Monochromatic, Isometric, Two Color y Flatline, realmente una excelente alternativa.

**Herramienta N° 2 Undraw**



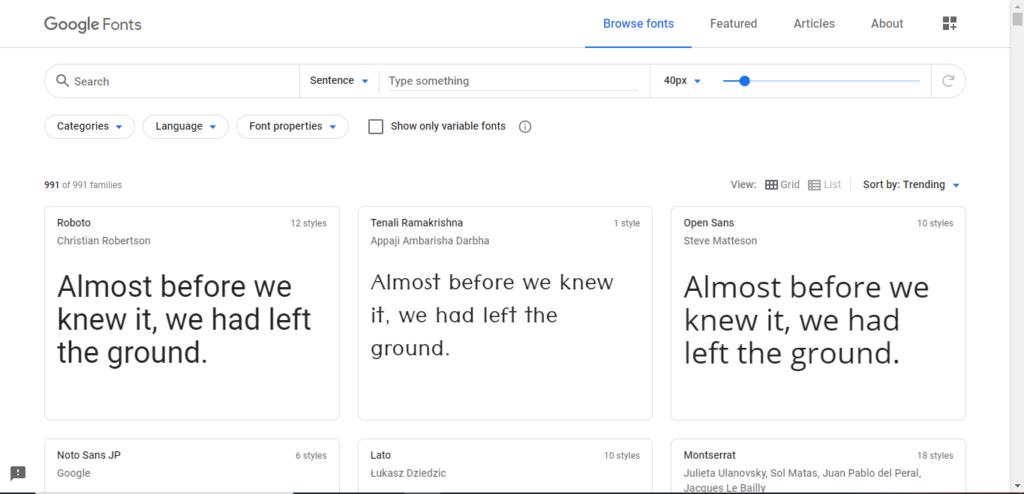
Undraw.co es uno de los recursos más famosos, debido a todas las ilustraciones que este tiene, pero sobre todo se actualiza cada semana y sus ilustraciones son totalmente gratis sin derechos de autor, algo bastante genial.

**Herramienta N° 3 Color codes**



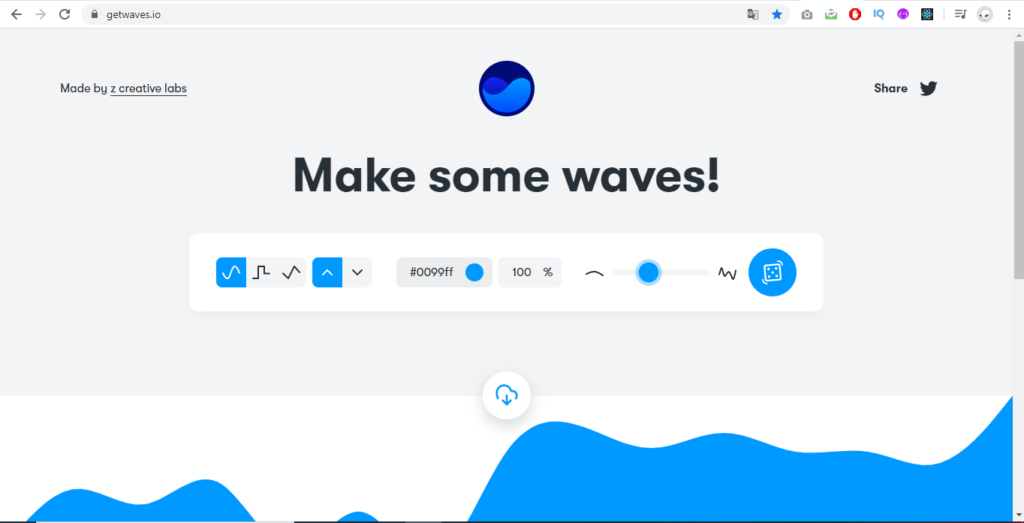
HTML  Color Codes, es un asombroso recurso para el diseño web, ¿Por qué? Tenemos bastantes generadores de colores, pero en este generador tu seleccionas el color de tu preferencia, después puedes elegir entre diversas técnicas de diseño para crear tu paleta de colores perfecta y en armonía.

**Herramienta N° 4 Google fonts**



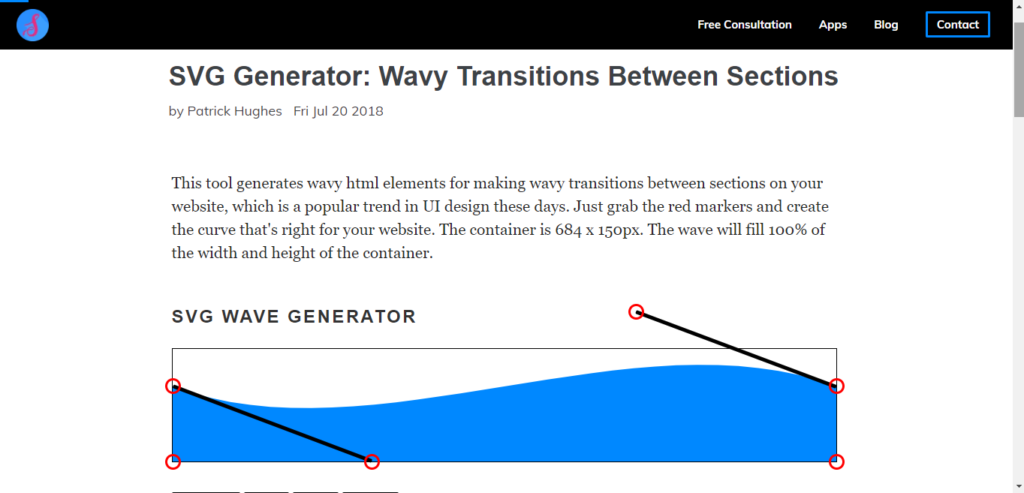
Google fonts, es la librería de fuentes más famosa, sin duda es la mejor herramienta de diseño web en cuestión de fuentes, porque nos proporciona cientos de fuentes que podemos usar libremente en nuestro proyecto web, de seguro hay una fuente adecuada para tu proyecto.

**Herramienta N° 5 Get Waves**



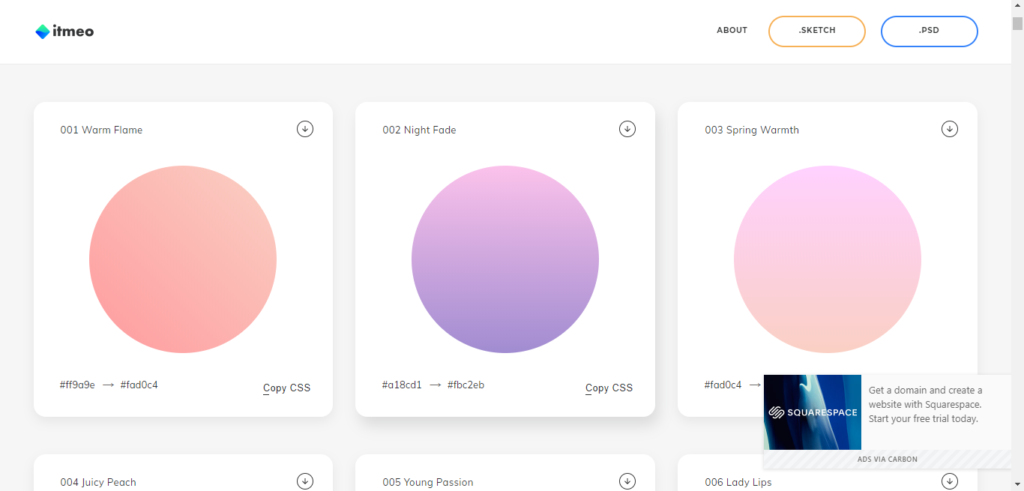
Getwaves son unas herramientas muy geniales que nos permitirá generar waves svg y después integrarlas a nuestro proyecto web para tener lo que se denomina un efecto wave, una técnica muy moderna y que nos ayudará bastante.

**Herramienta N° 6 SVG Generator**



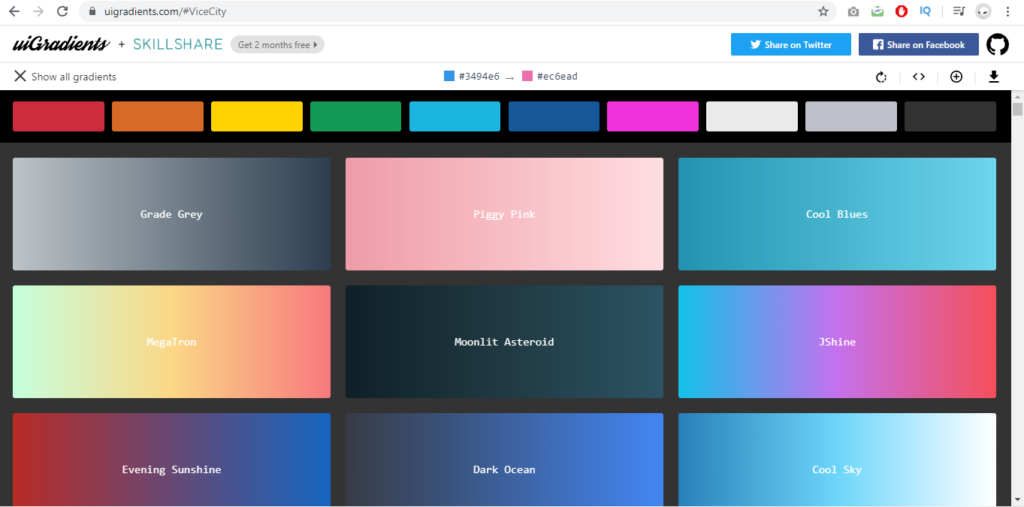
SVG Generator Wavy, de igual manera que Get waves, nos permite generar waves en formato svg, es una herramienta igual de genial que get waves.

**Herramienta N° 7 Web gradients**



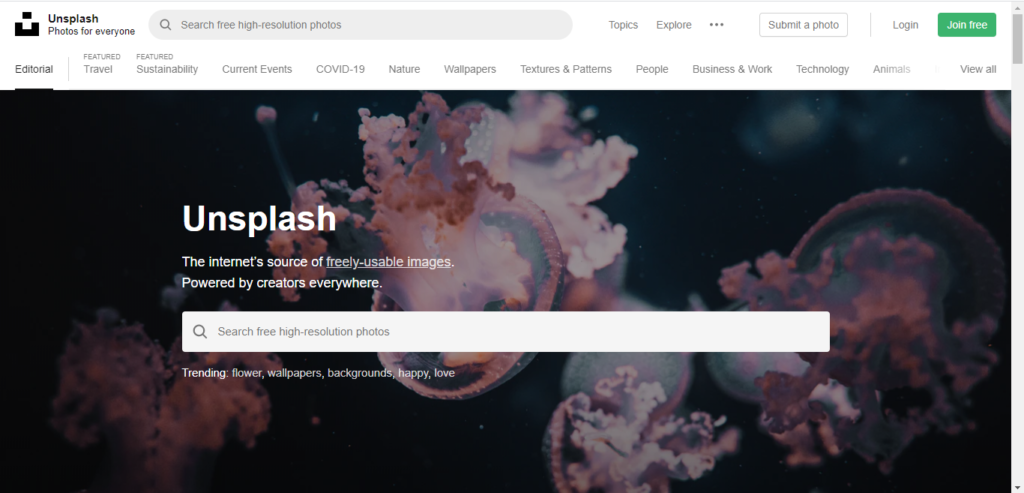
Web gradients es una genial herramienta para el diseño web, el cual es una biblioteca de gradientes ya establecidos, pero que son verdaderamente geniales, tu puedes seleccionar el que más te guste y te dará el código css o podrás también descargarlo como png.

**Herramienta N° 8 Ui Gradients**



Ui Gradients, al igual que web gradients, es una enorme biblioteca en la cual nos da gradientes, con un filtro para encontrar los gradientes o degradados respectivos al color que hemos seleccionado, al igual que en webgradients, te dará el código css del gradiente pero este te lo dará con prefijos.

**Herramienta N° 9 Unsplash**



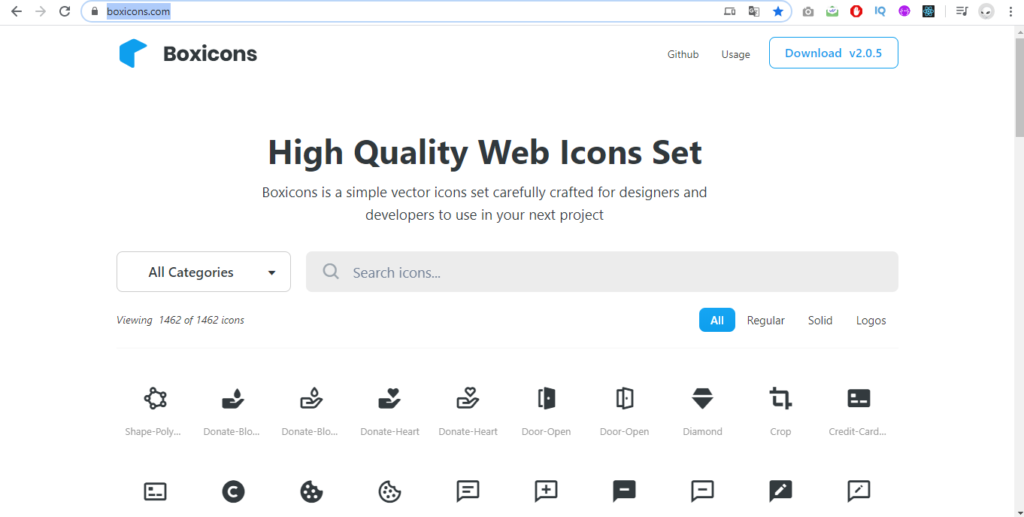
Unsplash es una biblioteca de imagenes gratuita, que nos permitirá descargar imagenes de distintos temas, todas estas imagenes libres de derechos de autor, por lo que podrás usarlas en tus proyectos web sin ningún problema.

**Herramienta N° 10 Pexels**



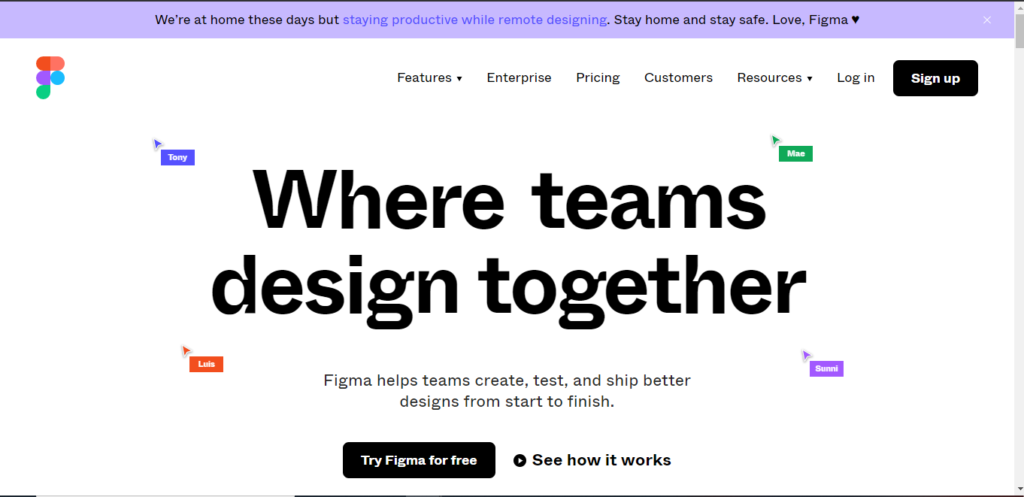
Pexels, al igual que unsplash es un banco de imágenes en el cual igual podrás descargar imágenes libres de derechos de autor y utilizarlas en tus proyectos, hay imágenes de distintos ramos y temáticas.

**Herramienta N° 11 Box Icons**



Box icons es un banco de iconos el cual nos proporciona cientos de iconos, estos iconos los podemos descargar como svg o png, también podemos usarlos como web components y como fuente, verdaderamente genial para incluir iconos en tus proyectos.

**Herramienta N° 12 Figma**



Figma es una gran herramienta para el diseño web, además de estar creada para diseñar interfaces, también nos va a permitir modificar elementos svg como ilustraciones e iconos, de esta manera podemos personalizarlos al máximo, además figma es totalmente gratuito para individuos y se ejecuta en el navegador por lo que no tendrás que preocuparte por la compatibilidad.