

1. Introducció

Los **lenguajes de marcas** son lenguajes que permiten estructurar el contenido de un documento a través del uso de **marcas** o **etiquetas** que contienen información sobre el texto englobado dentro de ellas

Características

- Codificados en texto plano
- Sencillos de interpretar
- Independientes de la plataforma
- Compactos (en un mismo documento está el texto y las marcas)
- Flexibles (ampliable según las necesidades)

```
<libro>
  <titulo>El Aleph</titulo>
    <autor>Jorge Luis Borges</autor>
    <genero>Cuentos</genero>
    <fecha>
      <escritura>1945</escritura>
      <publicacion>1949</publicacion>
    </fecha>
</libro>
```

Las **marcas** o **etiquetas** usan < > para abrir y por </ > para cerrar
Pueden tener **atributos** que aportan información al contenido

```
<etiqueta atributo="valor">Contenido</etiqueta>
```

Algunas marcas que no tienen marca de fin

```
<etiqueta> o <etiqueta/>
```

HTML

(*HyperText Markup Language*)

Lenguaje de marcas para la creación de páginas web, estandarizado por el **W3C**

Creado por sir Timothy "Tim" Berners-Lee y su grupo en 1989-1991

También creó...

- **HTTP** (*HyperText Transfer Protocol*) protocolo para la transmisión de documentos HTML
- **URL** (*Uniform Resource Locator*) para enlazar los documentos
- **W3C** (*World Wide Web Consortium*) entidad que recomienda sobre el funcionamiento de Internet



Un poco de historia:

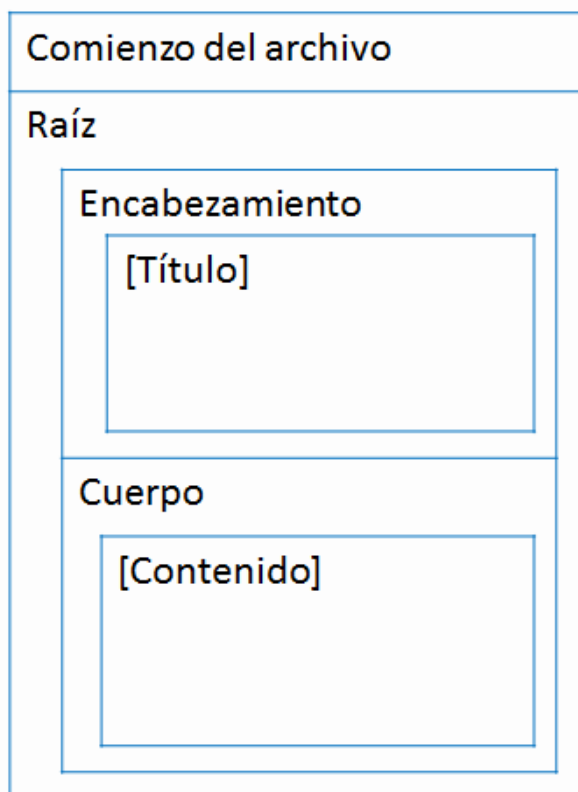
1989 - 1991	HTML 1.0
1995	HTML 2.0 Primer estándar oficial
1995	HTML 3.0
1997	HTML 3.2 Primera recomendación del W3C
1998	HTML 4.0
1999	HTML 4.01 Recomendación del W3C
2001	XHTML 1.0 y 1.1 Lenguaje basado en XML, pero poco compatible
2002	XHTML 2.0
2004	Nace WHATWG (<i>Web Hypertext Application Technology Working Group</i>) al que se sumarían Apple, Google... como alternativa a W3C
2006-2009	W3C abandona XHTML 2.0 y desarrolla HTML5 con las aportaciones de WHATWG
2014	HTML 5.0 Recomendación del W3C

En HTML 4.01

```
<!doctype html public "-//w3c//dtd html
4.01//en"
"http://www.w3.org/tr/html4/strict.dtd">
<html>
  <head>
    <title>Documento</title>
  </head>
  <body>
    Lorem ipsum dolor sit amet
  </body>
</html>
```

En HTML 5

```
<!doctype html>
<html>
  <head>
    <title>Documento</title>
  </head>
  <body>
    Lorem ipsum dolor sit amet
  </body>
</html>
```



```
<!doctype html>
<html>
  <head>
    <title>
    </title>
  </head>
  <body>
    .
    .
    .
  </body>
</html>
```

Elementos de <head>

- **Título** → Aparece en la pestaña del navegador

```
<title>Título del documento</title>
```

- **Metadatos** → Información sobre el documento

- **Charset** → Codificación que se usa en la página

```
<meta charset="utf-8">
```

- **Pares nombre/valor** → Datos específicos

Descripción

```
<meta name="description" content="Descripción">
```

Autor

```
<meta name="author" content="Nombre">
```

Palabras clave

```
<meta name="keywords" content="palabra1,palabra2">
```

...

- **Http-equiv**

Configura el refresco de la página cada x segundos

```
<meta http-equiv="refresh" content="30">
```

Configura una redirección

```
<meta http-equiv="refresh" content="3;url=http://bla.com/">
```

...

- **Estilos** → Definen la apariencia de los elementos

```
<style type="text/css">
  body { font-family: Calibri,Arial,Helvetica,sans-serif; }
  h1 { color: red; }
  p { color: blue; }
</style>
```

También pueden estar definidos en archivos vinculados


```
<link rel="stylesheet" type="text/css" href="estilos.css">
```

- **Scripts** → Código ejecutable por el navegador que hace alguna acción

```
<script type="text/javascript">  
  alert("Hola Mundo!");  
</script>
```

También pueden estar definidos en archivos vinculados

```
<script type="text/javascript" src="hola_mundo.js"></script>
```

- **Enlaces con recursos externos**

```
<link rel="stylesheet" type="text/css" href="estilos.css">
```

```
<link rel="stylesheet" type="text/css" href="impresora.css"  
media="print">
```

```
<link rel="alternate stylesheet" type="text/css" ref="estilo2.css">
```

```
<link rel="icon" href="/favicon.ico" sizes="8x8 16x16 32x32 48x48">
```

Elemento de <body>

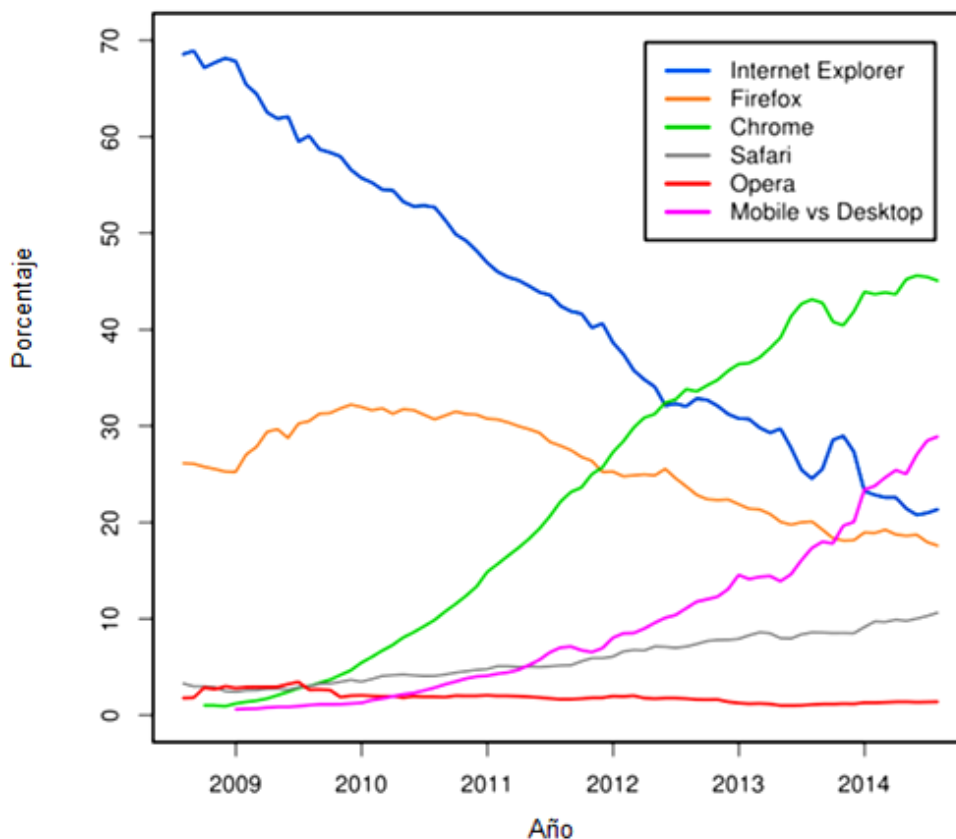
Contenido de la página

...

Navegadores web

Un navegador o *browser* es una aplicación que permite acceder a internet, interpreta documentos HTML y **presenta textos y gráficos** en pantalla

También existen navegadores **sólo de texto** (Lynx, Links, Bobcat, w3m...), útiles para usuarios con alguna minusvalía



Optimización para distintos navegadores

Los navegadores usan un motor de renderización:

- **Blink:** Chrome (v28+), Opera (v15+)...
- **Gecko:** Firefox...
- **Trident:** Internet Explorer...
- **WebKit:** Safari...

Según del sistema operativo, el navegador, la versión del navegador y su motor de renderización la visualización de la página puede variar, así que debemos:

- Comprobar la visualización en diferentes navegadores
- En muchas ocasiones, incluir código adicional para asegurar la visualización correcta



Además, la proliferación de dispositivos ha multiplicado los formatos, debemos prever cómo se verá nuestro diseño en los más frecuentes

Haz una página básica y observa cómo se ve en un navegador

```
<!doctype html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="author" value="Yo">
    <title>Título de la página</title>
  </head>
  <body>
    <h1>Titular de mi página</h1>
    <p>Contenido de mi página</p>
  </body>
</html>
```

1. Introducción HTML5

HTML



-Como bien indica su propio nombre es la **5** versión del estándar de WWW (World Wide Web), el **W3C** ha estado años trabajando en este nuevo estándar después del **XHTML**.

-Esta nueva versión sí es compatible con todas las anteriores, además de incorporar nuevas mejoras, por ejemplo la etiqueta ID de un **div** en mayúsculas o minúsculas y que los navegadores lo reconozcan sin producir un error. Así mismo también incluye nuevas funciones como la reproducción de vídeo y sonido, y algunas etiquetas nuevas. Para ello hicieron un estudio sobre el nombrado de etiquetas de las webs más famosas, y casi todas tenían una barra lateral, llamada **aside**, un encabezado llamado **header**, y un pie llamado **footer**, así como las barras de navegación **nav** y un largo etc.

2. Estructura

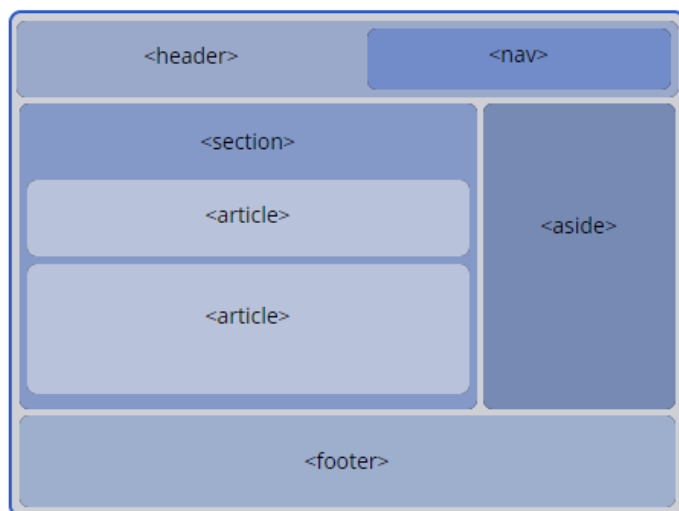
-Con la versión anterior de **HTML** (HTML4) sólo existía un tipo de caja (<div>), por lo que una caja podía contener cualquier tipo de contenido.

-Una de las novedades de **HTML5**, aparte de disponer de esta misma caja genérica (<div>), tenemos diferentes tipos de cajas destinadas cada una a un tipo de contenido concreto. Así, las cajas de tipo <header> están destinadas a contener los encabezados de la página, las de tipo <footer> para el pie de página, los enlaces estarán dentro de las cajas de tipo (<nav>), para el contenido normal tenemos cajas de tipo <section> y <article>, para el contenido relacionado con el tema central pero no directamente tenemos el tipo <aside> o las imágenes que tienen su propia caja <figure>.

-De esta manera, posteriormente entender y encontrar cualquier elemento o contenido es mucho más rápido, sencillo y fiable.

-Además, al ver (nosotros, cualquier otra persona o "araña/robot" que explore nuestra página) una estructura **html** es posible identificar la organización del contenido con un primer golpe de vista.

Las etiquetas semánticas más importantes son:



<header>: Es el equivalente a la cabecera de la página web. Contiene el título o nombre de la empresa/titular de la página, logo e información relacionada.

<nav>: Contiene los enlaces (barra de navegación) externos o internos de la página.

<section>: Es una gran caja que sirve para mostrar grandes bloques de contenido de la página. Puede contener diferentes sub-apartados de diferentes temas (de tipo **<article>**).

<article>: Es una caja independiente de contenido que puede estar contenida (o no) dentro de un **<section>**. Normalmente utilizada para contenidos no demasiado extensos.

<aside>: Define un bloque de contenido relacionado de manera indirecta con el contenido principal, pero que no es esencial para la comprensión del mismo.

<footer>: Equivale al pie de página de un apartado concreto (**<section>**, **<article>**...) o de la página web en general.

Para seleccionar el nombre de estas etiquetas, Ian Hickson (el editor de HTML5) recurrió a las herramientas de Google para extraer datos de más de mil millones de páginas web reales, repasando los ID y los nombres de clase más utilizados en la web.

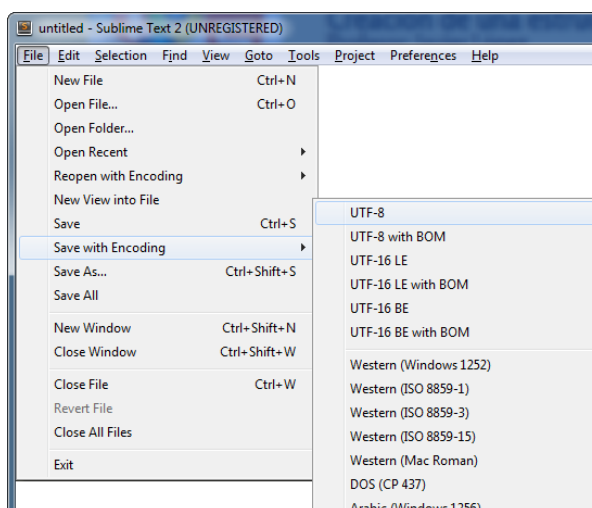
EJEMPLO DE UNA ESTRUCTURA COMPLEJA

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8"/>
    <meta name="description" content="Resumen del contenido de la
    página">
    <title>Título de la página</title>
    <link rel=stylesheet href="css/estilo.css" type="text/css"/>
    <style type="text/css">
    </style>
  </head>
  <body>
    <header>
    </header>
    <section id="contenido">
      Contenido de la web
    </section>
    <footer>
    </footer>
  </body>
</html>
```

GUARDAR CON FORMATO UTF-8

El último punto a tener en cuenta es guardar el documento HTML usando la codificación especificada en la etiqueta <head>:
<meta charset="utf-8"/>

Así, si en la etiqueta anterior le estamos diciendo al navegador que la página va a utilizar una codificación UTF-8 (para que permita reconocer los acentos) la página tiene que tener realmente esta codificación.



2. Marcas para dar formato

Las **marcas de formato** permiten estructurar como en un procesador de texto y ayudan a los buscadores a indexar el contenido

- **Etiquetas de bloque**

No necesitan estar dentro de otras etiquetas, crean su propio bloque

Encabezamientos

```
<h1>Encabezado de nivel 1</h1>
<h2>Encabezado de nivel 2</h2>
<h3>Encabezado de nivel 3</h3>
<h4>Encabezado de nivel 4</h4>
<h5>Encabezado de nivel 5</h5>
<h6>Encabezado de nivel 6</h6>
```

Párrafos

p: <p>Párrafo</p>

Otros

pre: <pre>Texto preformateado, permite espacios</pre>

address: <address>Información del autor</address>

Dentro de una página básica incluye estos fragmentos de código y compara el resultado en un navegador:

```
<p>Párrafo 1</p>
<p>Párrafo 2</p>
<p>Párrafo 3</p>

<p>Primera línea<br/>
Segunda línea<br/>
Tercera línea</p>
```

- **Etiquetas en línea**

Necesitan estar dentro de otras etiquetas, no crean su propio bloque

br:	Salto de línea También funciona sin /
b:	Texto en negrita
strong:	Texto importante
i:	<i>Texto en itálica</i>
em:	Texto enfatizado
dfn:	<dfn>Definición: Acción y efecto de definir.</dfn>
cite:	<cite>Yo no soy mala, es que me han dibujado así</cite>
code:	<code>Texto de código de programa</code>
kbd:	<kbd>Entrada de teclado</kbd>
samp:	<samp>Texto de consola</samp>
abbr:	Pon el cursor sobre <abbr title = "ejemplo">ej.</abbr> para ver qué significa la abreviatura
q:	<q>Texto entrecomillado</q>
bdo:	<bdo dir="ltr">Texto al derecho</bdo> <bdo dir="rtl">Texto al revés</bdo>
del:	Texto eliminado, el malo
ins:	<ins>Texto eliminado, el bueno</ins>
sub:	_{Subíndice}
sup:	^{Superíndice}
mark:	<mark>Texto reetiquetado</mark>
span	Aplicar un formato a un texto dentro de un párrafo

Las etiquetas pueden combinarse, pero respetando el orden de anidamiento

Forma correcta: Texto en negrita y <i>texto en negrita e itálica</i>

Forma incorrecta: Texto en negrita y <i>texto en negrita e itálica</i>

- **Etiquetas de contenedores de texto**

Pueden contener dentro otras etiquetas de bloque

blockquote: <blockquote>Texto con sangría</blockquote>

div: <div>Contenedor</div>

Dentro de una página básica practica todas las etiquetas de bloque, en línea y de contenedores de texto y compara el resultado en un navegador

- **Líneas horizontales**

Permiten separar bloques

hr: <hr/> También funciona sin /

Dentro de una página básica incluye este fragmento de código y observa el resultado en un navegador

```
<p>Texto del párrafo 1</p>
<hr/>
<p>Texto del párrafo 2</p>
<hr/>
<p>Texto del párrafo 3</p>
```

- **Fuentes, tamaño, alineación y colores**

Aunque lo estándar es trabajar con estilos, veremos la antigua etiqueta font

Permite elegir fuente, tamaño y color

```
<p><font face="Arial">Texto en Arial</font></p>
<p><font face="Comic Sans MS,Arial,Verdana">Texto en
Comic Sans</font></p>
```

```
<p><font size="5">Texto de tamaño 5</font></p>
```

```
<p><font color="red">Texto rojo</font></p>
<p><font color="green">Texto verde</font></p>
<p><font color="blue">Texto azul</font></p>
```

```
<p align="right">Texto a la derecha</p>
<p align="center">Texto centrado</p>
<p align="left">Texto a la izquierda</p>
```

Algunos colores se llaman por su nombre, los demás por su código hexadecimal o su

código decimal RGB, consulta la lista completa: <http://www.coloreshtml.com/>

	#FFFFFF	white		#800000	maroon
	#000000	black		#800080	purple
	#000080	navy		#808000	olive
	#0000FF	blue		#808080	gray
	#008000	green		#C0C0C0	silver
	#008080	teal		#FF0000	red
	#00FF00	lime		#FF00FF	fuchsia
	#00FFFF	aqua		#FFFF00	yellow

También podemos aplicar colores a otros elementos como el fondo de la página

Dentro de una página básica sustituye `<body>` por este fragmento de código y observa el resultado en un navegador:

```
<body style="background-color: silver">
```

• Listas

Dentro de una página básica incluye estos fragmentos de código y observa el resultado en un navegador:

Lista no ordenada

```
<ul>  
  <li>Elemento</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
</ul>
```

Lista ordenada

```
<ol>  
  <li>Elemento 1</li>  
  <li>Elemento 2</li>  
  <li>Elemento 3</li>  
</ol>
```

Lista de definición

```
<dl>  
  <dt>Término 1</dt>  
  <dd>Definición 1</dd>  
  <dt>Término 2</dt>  
  <dd>Definición 2</dd>  
  <dt>Término 3</dt>  
</dl>
```

- **Comentarios**

Son notas para documentar el código, pero el navegador no muestra

Dentro de una página básica incluye este fragmento de código y observa el resultado en un navegador:

```
<!-- documentar el código -->
```

- **Entidades**

Son secuencias que permiten usar caracteres conflictivos

Su estructura es:

& + código especial + ;

Caracteres reservados

corchetes angulares < y > = < y >

ampersand = &

Vocales acentuadas y caracteres que no existen en inglés

à = à è = è À = À È = È

á = á é = é Á = Á É = É

ä = ä ë = ë Ä = Ä Ë = Ë

ç = ç Ç = Ç ñ = ñ Ñ = Ñ

¿ = ¿ ¡ = ¡ € = € espacio =

...

3. Enlaces

Los enlaces a otros documentos y recursos se marcan con la etiqueta a y la dirección del enlace se marca con el atributo href

- **Enlaces externos**

Siempre son absolutos

- `<a href="http://www.google.es/">Enlace a Google</a>`
- `<a href="ftp://ftp.servidor.com">Servidor FTP</a>`

- **Enlaces internos**

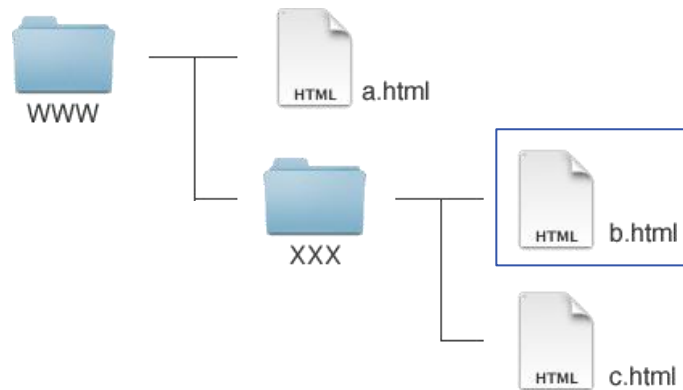
Pueden ser relativos o absolutos

Crea varias páginas según esta estructura y usa enlaces relativos entre ellas

Por ejemplo, desde b.html

```
<a href="c.html">Enlace a C</a>
```

```
<a href="../a.html">Enlace a A</a>
```



Prueba lo mismo con enlaces absolutos

```
<a href="http://www.dominio.com/a.html">Enlace a A</a>
```

```
<a href="http://www.dominio.com/xxx/c.html">Enlace a C</a>
```

- **Enlaces de ancla**

Debemos usar una marca concreta dentro del código

Crea una página con estos elementos

```
<a href="#parrafo1">Ir al párrafo 1</a>
```

```
<a href="#parrafo2">Ir al párrafo 2</a>
```

...

```
<h3 id="parrafo1">Párrafo 1</h3>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing ...</p>
```

...

```
<h3 id="parrafo2">Párrafo 2</h3>
```

```
<p>Sed ut perspiciatis unde omnis iste natus error ...</p>
```

Podemos combinar los enlaces internos o externos con los de ancla

Incluye este enlace

```
<a
```

```
href="http://es.wikipedia.org/wiki/Hiperenlace#Comportamiento_de_lo
```

s_enlaces_en_los_navegadores_web">Enlace

- **Enlaces a documentos**
Relativos

Apuntes

O absolutos

Apuntes

- **Enlaces de correo**

Correo de Glòria

Podemos especificar en qué ventana se abre el enlace, con el atributo target:

- `_blank`: se abre en una ventana nueva
- `_self`: se abre en la propia ventana
- `_parent`: se abre en la ventana padre de la actual
- `_top`: se abre en la ventana que está en primer nivel
- `nombre`: se abre en una ventana concreta

Incluye este enlace

```
<a href="http://www.google.es/" target="_blank">Ir a Google</a>
```

4. Tablas

Aunque se desaconseja el uso de tablas para estructurar la página, sí se usan para exponer datos ordenadamente

Ejemplo básico

```
<table border="1px">

  <tr>
    <td>Fila 1 columna 1</td>
    <td>Fila 1 columna 2</td>
    <td>Fila 1 columna 3</td>
  </tr>

  <tr>
    <td>Fila 2 columna 1</td>
    <td>Fila 2 columna 2</td>
    <td>Fila 2 columna 3</td>
  </tr>

  <tr>
    <td>Fila 3 columna 1</td>
    <td>Fila 3 columna 2</td>
    <td>Fila 3 columna 3</td>
  </tr>

</table>
```

Ejemplo más avanzado

```
<table border="1px">
  <caption>Título la tabla</caption>
  <thead>
    <tr>
      <th>Encabezamiento 1</th>
      <th>Encabezamiento 2</th>
      <th>Encabezamiento 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Fila 1 columna 1</td>
      <td>Fila 1 columna 2</td>
      <td>Fila 1 columna 3</td>
    </tr>
    <tr>
      <td>Fila 2 columna 1</td>
      <td>Fila 2 columna 2</td>
      <td>Fila 2 columna 3</td>
    </tr>
  </tbody>

  <tfoot>
    <tr>
      <td>Resultado 1</td>
      <td>Resultado 2</td>
      <td>Resultado 3</td>
    </tr>
  </tfoot>
</table>
```


Podemos formar grupos de columnas

```
<table border="1px">
  <colgroup span="2" style="background-color: red;"></colgroup>
  <colgroup></colgroup>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
</table>
```

Podemos especificar cada columna

```
<table border="1px">
  <colgroup>
    <col>
      <col style="background-color: red;">
    <col>
  </colgroup>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
</table>
```

Podemos agrupar columnas

```
<table border="1px">
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
  </tr>
  <tr>
    <td colspan="2">Celda 3</td>
  </tr>
</table>
```

Y lo mismo con filas

```
<table border="1px">
  <tr>
    <td rowspan="2">Celda 1</td>
    <td>Celda 2</td>
  </tr>
  <tr>
    <td>Celda 3</td>
  </tr>
</table>
```

Construye una tabla como ésta

Celda 1		Celda 2
		Celda 3
Celda 4	Celda 5	Celda 6

Incluye en los td height y width de 50 px para que las celdas sean cuadradas

5. Capas

Son contenedores que permiten estructurar el contenido y facilitar la aplicación de estilos CSS, pueden usarse para partes concretas

```
<div lang="es" title="Texto original" style="border: solid 1px blue;">
  <p>Lorem Ipsum es simplemente el texto de relleno </p>
</div>
<div lang="en" title="Texto traducido" style="border: solid 1px red;">
  <p>Lorem Ipsum is simply dummy text </p>
</div>
```

Podemos usar capas flotantes a un lado o centradas

```
<div style="float: left; background-color: red; width: 200px; height: 100px;
  margin: 5px;">Capa 1</div>
<div style="float: left; background-color: yellow; width: 200px; height: 100px;
  margin: 5px;">Capa 2</div>
<div style="float: left; background-color: green; width: 200px; height: 100px;
  margin: 5px;">Capa 3</div>
```

Compara los resultados usando float: right y margin: auto

Podemos solapar capas

```
< <div style="position: absolute; background-color: silver; z-index: 1;
  width: 300px; height: 300px;"></div>
<div style="position: absolute; background-color: gray; z-index: 2;
  width: 300px; height: 300px; top: 20%; left: 20%;"></div>
<div style="position: fixed; background-color: black; z-index: 3;
  width: 100%; height: 20px; bottom: 0px;"></div>
```

Compara los resultados usando diferentes valores de z-index, de top y left, sustituyendo por right...

También para la estructura total de una página

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Estructura en capas</title>
</head>
<body>
  <div id="container" style="width: 100%">
    <div id="header" style="background-color: silver; height: 200px;">
      <h1 style="margin-bottom: 0">Cabecera</h1>
    </div>
    <div id="menu" style="background-color: gray; height: 400px; width: 20%; float:
    left">
      <b>Menú</b><br>
      Opción 1<br>
      Opción 2<br>
      Opción 3
    </div>
    <div id="content" style="background-color: #EEEEEE; height: 400px; width: 80%;
    float: left">
      Contenido
    </div>
    <div id="footer" style="background-color: black; height: 100px; clear: both; text-
    align: center; color: white">
      Créditos y Copyright ©
    </div>
  </div>
</body>
</html>
```

Observa que para poner fin al efecto de capas flotantes usamos clear

6. Elementos multimedia

Bajo el término multimedia se engloban todos aquellos elementos no HTML que el navegador puede reproducir

- **Audio**

Con la etiqueta *audio*, y estos atributos:

- *autoplay*: inicia la reproducción tan pronto como sea posible
- *controls*: para que los controles aparezcan
- *loop*: para que se repita la reproducción
- *muted*: para que la salida de sonido esté silenciada
- *preload*: para que precargue y cómo
- *src*: URL del archivo

```
<audio controls src="http://media.w3.org/2010/07/bunny/04-Death_Becomes_Fur.oga">
```

```
<p>Su navegador no soporta este elemento de HTML5</p>  
</audio>
```

- **Vídeo**

Con la etiqueta *video*, que tiene los mismos atributos que *audio*, y además:

- *width* y *height*: especifica la anchura y altura del vídeo
- *poster*: especifica una imagen mientras espera la reproducción

```
<video controls src="http://media.w3.org/2010/05/bunny/movie.ogv">  
<p>Su navegador no soporta este elemento de HTML5</p>  
</video>
```

Páginas como Goear o YouTube ofrecen la posibilidad de incorporar audios y vídeos respectivamente, a través de un enlace:

- http://www.youtube.com/watch?v=8N_tupPBtWQ
- http://youtu.be/8N_tupPBtWQ

O como marco incrustado:

```
<iframe width="560" height="315"  
src="http://www.youtube.com/embed/8N_tupPBtWQ" frameborder="0"  
allowfullscreen></iframe>
```

Algunos formatos de audio y vídeo no son soportados por Explorer y Opera

- **Marquesinas**

Aunque en HTML5 ya no se usan, los navegadores aún las muestran. Su etiqueta es *marquee* y tiene los siguientes atributos:

- *width*: ancho de la marquesina en píxeles
- *direction*: desplazamiento a izquierda (*left*) o derecha (*right*)
- *behavior*: comportamiento del desplazamiento, *scroll* (defecto) o *alternate*
- *scrollDelay*: tiempo en milisegundos entre cambio de posición
- *scrollAmount*: cuántos píxeles se desplaza
- *loop*: para que se repita la reproducción
- *bgcolor*: color de fondo
- *hspace* y *vspace*: espacio en píxeles entre la marquesina y el resto de elementos

`<marquee width="500px">Texto que se desplazará</marquee>`

7. Imágenes

Tipos de formato (subrayados los soportados para web)

- **De mapa de bits** (*bitmap*)
 - Sin compresión o con compresión sin pérdida (BMP, PSD, TIF, RAW...)
 - Con pérdidas al comprimir, soportados para web
 - **GIF** (Graphic Interchange Format)
Compresión indexando el número de colores que forman la imagen, permite transparencias y animaciones, adecuado para imágenes con pocos colores, sin degradados
 - **JPEG** (Joint of Photographic Experts Group)
Compresión graduable, adecuado para fotos de muchos colores
 - **PNG** (Portable Network Graphics)
8 bits, equivalente a GIF y 24 bits, equivalente a JPG
- **De vectores**
Gráficos que se pueden redimensionar sin perder calidad,
 - Adecuados para enviar a imprenta (EPS, AI, PDF...)
 - Soportado para web
 - **SVG** (Scalable Vector Graphics)

Para insertar imágenes usamos la etiqueta *img*, con los siguientes atributos:

- *width* y *height*: anchura y altura de la imagen
- *src*: URL del archivo
- *alt*: texto alternativo que se cargará en lugar de la imagen
- *title*: texto de la etiqueta al colocarse sobre la imagen

```

```

Busca una imagen en Google, copia su enlace y enlázala en una página simple
Modifica su tamaño, prueba usando solo un valor de tamaño

Las imágenes pueden alinearse igual que el texto, y se añaden algunas opciones más

```

```

Compara los resultados usando diferentes valores de align: right, left, center, top, middle y bottom

Prueba también a añadir hspace="8px" vspace="8px" para añadir márgenes a la imagen

Haz una búsqueda de imágenes con las palabras *background* y *tileable*, bájate la que más te guste y úsala como fondo

```
<body background="fondo-acolchado.jpg">
```


8. Marcos

Permiten mostrar el contenido de otra página web

Se delimita con la etiqueta *iframe* (*inline frame*), y la dirección de la página web con el atributo *src*, se puede definir el ancho y alto píxeles con los atributos *width* y *height*

```
<iframe src="http://www.fundaciocim.org/" width="640px"
height="480px">
    <p>Su navegador no soporta marcos incrustados</p>
</iframe>
```

Recuerda:

Páginas como Goear o YouTube ofrecen la posibilidad de incorporar audios y vídeos respectivamente, a través de un enlace:

- http://www.youtube.com/watch?v=8N_tupPBtWQ
- http://youtu.be/8N_tupPBtWQ

O como marco incrustado:

```
<iframe width="560" height="315"
src="http://www.youtube.com/embed/8N_tupPBtWQ" frameborder="0"
allowfullscreen></iframe>
```

10. Diferencias entre XHTML y HTML5

XHTML (*eXtensible HyperText Markup Language*): HTML expresado como XML válido, más estricto para asegurar la visualización correcta en distintos dispositivos. Se diseñó para sustituir a HTML4.01.

- *XHTML DOCTYPE* y atributo *xmlns* en *html* obligatorios
- *html*, *head*, *title* y *body* obligatorios
- Los elementos deben estar correctamente anidados
- Todos los elementos deben cerrarse
- Todos los elementos y sus atributos deben escribirse con minúsculas
- Todos los elementos deben tener un elemento raíz
- Los valores de los atributos deben ir entre comillas
- Los atributos no se pueden minimizar

HTML5: Versión de HTML recomendada desde octubre de 2014

Ambos lenguajes van hacia la Web 3.0, **la web semántica** que es un conjunto de recomendaciones del W3C para separar la presentación y la estructura de contenidos, y con ello facilitar la interpretación automática de contenidos.

Sin embargo, hay algunas diferencias:

1. Algunos elementos que eran obligatorios en sintaxis basada en XHTML ya no lo son para que un documento pase la validación HTML5; ejemplos claros son los elementos *html* y *body*.
2. Los elementos vacíos, o aquellos que no albergan ningún contenido, no se tienen que cerrar utilizando una barra inclinada de cierre; esto ocurre con *<meta>* y *
*.
3. Elementos y atributos pueden estar en mayúsculas, minúsculas o una mezcla de ambos.

4. Las comillas son innecesarias alrededor de valores de atributo, a menos que se utilicen múltiples valores delimitados por espacios, o si una URL aparece como un valor y contiene una cadena de consulta con un carácter de igual (=).
5. Algunos atributos eran obligatorios en la sintaxis basada en XHTML, ya no lo son en HTML5; ejemplos de esto son el atributo *type* para el elemento *script* y el atributo *xmlns* para el elemento *html*.
6. Algunos elementos que se habían desaprobado y, por lo tanto, eran no válidos en XHTML, ahora son válidos, un ejemplo es el elemento *embed*.
7. Texto perdido que no aparece dentro de ningún elemento invalidaría un documento XHTML; este no es el caso en HTML5.
8. Algunos elementos que se tenían que cerrar en XHTML se pueden dejar abiertos sin causar errores de validación en HTML5; por ejemplo, *p*, *li* y *dt*.
9. El elemento *form* ya no necesita tener un atributo *action*.
10. Los elementos de formulario, como *input*, se pueden situar como hijos directos del elemento *form*; en XHTML era necesario otro elemento (como *fieldset* o *div*) para englobarlos.

http://www.w3schools.com/html/html_xhtml.asp

<https://nubedeideas.wordpress.com/2012/04/19/10-diferencias-entre-la-validacion-xhtml-html5/>

11. Introducción a CSS

Las hojas de estilo en cascada (*Cascading Style Sheets*) permiten modificar la presentación de un documento HTML sin alterar su estructura

- Permiten alterar el aspecto por defecto de los elementos HTML
- Facilitan la presentación
- Existen tres formas diferentes de aplicar estilos:

- **En línea:** aplica directamente con *style*

```
<p style="font-family:Arial; font-size:14px; color:blue">texto</p>
```

- **Incrustados:** en *head* con *style*, sólo afectan a ese documento

```
<head>
:
  <style type="text/css">
    p {
      font-family: Arial;
      font-size: 14px;
      color: blue
    }
  </style>
:
</head>
```

- **Enlazados:** en una hoja de estilo externa, que es un archivo de texto plano con extensión .css, afecta a todos los documentos que lo enlacen

```
<head>
:
  <link rel="stylesheet" type="text/css" href="archivo.css">
:
</head>
```

Comprueba que las tres formas que dan el mismo resultado

Conceptos básicos

• Color

- Valor hexadecimal: `h1 { color: #FF0000; }`
- Nombre en inglés: `p { color: red; }`
- Valor RGB (*Red Green Blue*): `h1 { color: rgb(255, 0, 0); }`
- Modelo RGBA (*Red Green Blue Alpha*),
Alpha es la opacidad de 0.0 a 1.0: `p { color: rgba(255, 0, 0, .5); }`
- Cambio de opacidad: `img { opacity: 0.4; }`

• Listas

- Tipo de lista: `ul li { list-style-type: square; }`
- Imagen como marcador: `ul li { list-style-image: url('estrellaroja.gif'); }`
- Poner el marcador hacia dentro: `ul { list-style-position: inside; }`
- Poner el marcador hacia fuera: `ul { list-style-position: outside; }`

• Fondo

- Color de fondo: `body { background-color: red; }`
- Imagen de fondo: `body { background-image: url('fondo.png'); }`
- Repetir imagen en horizontal: `background-repeat: repeat-x;`
- Repetir imagen en vertical: `background-repeat: repeat-y;`
- Imagen de fondo fija: `background-attachment: fixed;`
- Imagen de fondo que se desplaza: `background-attachment: scroll;`

• Tablas

- Aplicar bordes: `table { border: 1px solid black; }`
- Definir anchos y altos: `table { width: 100%; }`
`th { height: 50px; }`
- Alinear el texto: `th { text-align: left; }`
`td { height: 60px; vertical-align: bottom; }`
- Borde interior de la celdas: `td { padding: 20px; }`

- Color: `table { border: 1px solid green; background-color: green; color:white; }`

• Texto

- Alineación a la izquierda: `p { text-align: left; }`
- Alineación al centro: `p { text-align: center; }`
- Alineación a la derecha: `p { text-align: right; }`
- Texto justificado: `p { text-align: justify; }`
- Quitar el subrayado de los enlaces: `a { text-decoration: none; }`
- Con una línea encima: `p { text-decoration: overline; }`
- Texto tachado: `p { text-decoration: line-through; }`
- Texto subrayado: `p { text-decoration: underline; }`
- Transformar a mayúsculas: `p.may { text-transform: uppercase; }`
- Transformar a minúsculas: `p { text-transform: lowercase; }`
- Primera en mayúscula: `li { text-transform: capitalize; }`
- Versalitas: `p { font-variant: small-caps; }`
- Indentar la primera línea del texto: `p { text-indent: 25px; }`
- Texto de izquierda a derecha: `p { direction: ltr; }`
- Texto de derecha a izquierda: `p { direction: rtl; }`
- Tipos de letra
 - `h1 {font-family:"Times New Roman",Geneva,Times,serif; }`
 - `p { font-family:Arial,Verdana,"Lucida Sans Unicode",sans-serif; }`
 - Ningún estilo `p { font-style: normal; }`
 - Cursiva `p { font-style: italic; }`
 - Negrita `p { font-weight: bold; }`
 - Tamaño, en valor absoluto (px o pt) `h1 { font-size: 40px; }`
 - o en valor relativo (em) `h1 { font-size: 2.5em; }`
 - $16px=1em \rightarrow 40px/16=2.5em \mid 30px/16=1.875em \mid 14px/16=0.875em$
 - Espacio entre palabras `p { word-spacing: 30px; }`
 - Espacio entre letras `p { letter-spacing: 2px; }`
 - Espacio entre líneas `p { line-height: 24px; }`
 - Sombra (posX posY difuminado color) `p { text-shadow: 3px 3px 4px red; }`
 - Superíndice `p { vertical-align: sup; }`
 - Subíndice `p { vertical-align: sub; }`
 - Alinear arriba `img { vertical-align: top; }`
 - Alinear en medio `img { vertical-align: middle; }`

- Alinear abajo `img { vertical-align: bottom; }`
- Comentarios `/* documentar el código */`

Selectores

- **Selector de elementos** de HTML, por ejemplo: `<p>texto</p>`
`p { text-align: right; }`
- **Selector de clases**, se aplica a todos los elementos que tengan ese *class*, por ejemplo `<div class="centrado">texto</div>`
`.centrado { text-align: center; }`
- **Selector de id**, se aplica a todos los elementos con esa *id*, por ejemplo `<div id="texto_rojo">texto</div>`
`#texto_rojo { color: red; }`

En cualquiera de los casos, la estructura será la misma:

selector { propiedad:valor; propiedad:valor; }

- Podemos usar pseudoclases para aplicar efectos:

selector:pseudoclase { propiedad: valor; propiedad:valor; }

Con las clases y los id también podemos usar pseudoclases:

selector.clase:pseudoclase { propiedad: valor; propiedad:valor; }

selector#id:pseudoclase { propiedad: valor; propiedad:valor; }

Pseudoclases	en un enlace	
:link	a:link	Aplica estilo a los enlaces no visitados
:visited	a:visited	Aplica estilos a los enlaces visitados
:active	a:active	Aplica estilos al enlace activo
:hover	a:hover	Aplica estilos al pasar sobre un enlace

Crea una página con una hoja de estilos vinculada y haz que estos dos fragmentos de código funcionen:

```
a.rojo:hover { color: #FF0000; }
```

```
<a class="rojo" href="#">texto</a>
```

¿Qué falta para que el texto sea negro y sin subrayado?

Agrupación y abreviaturas

Si existen varios elementos con las mismas definiciones de estilo, se pueden agrupar

```
h1 { color: green; }
```

```
h2 { color: green; }
```

```
p { color: green; }
```

Equivale a:

```
h1, h2, p { color: green; }
```

También se pueden agrupar los atributos

```
.nombre_clase {
```

```
font-style: italic;
```

```
font-variant: normal;
```

```
font-weight: bold;
```

```
font-size: 12px;
```

```
font-family: Arial;
```

```
}
```

Equivale a:

```
.nombre_clase { font: italic normal bold  
12px Arial; }
```

Podemos omitir ceros y unidades de medida innecesarias:

```
padding: 0.5em;
```

Equivale a:

```
padding: .5em;
```

Podemos abreviar el código de color en hexadecimal:

```
#FFFFFF
```

Equivale a:

```
#FFF
```

Anidamiento y herencia

Es posible aplicar un estilo a un selector dentro de otro


```
p { color: blue; text-align: center; }      /* para párrafo */
.remarcado { background-color: red; }      /* para clase */
.remarcado p { color: white; }            /* para clase dentro de párrafo */
```

Comprueba que los elementos contenidos en otros heredan sus estilos

14. Recursos sobre tipografía

- **DaFont:** esta es una de nuestras herramientas favoritas. Ideal para, por ejemplo, estudiantes de diseño. En este sitio podemos encontrar una enorme cantidad de tipografías divididas de acuerdo a estilos, que se pueden descargar con licencia Creative Commons para uso gratuito.
- **Google Web Fonts:** del mismo estilo que Da Font, esta herramienta provista por Google nos permite descargar una serie de tipografías aptas para la web, Open Source.
- **TextMod 2.1:** con esta herramienta podemos revisar la compatibilidad de caracteres entre un set que estemos trabajando y el texto que tengamos que utilizar.
- **MyFontbook:** esta herramienta basada en la web nos permite administrar las fuentes que tengamos instaladas en nuestro ordenador, desde visualización hasta clasificación. Lo mejor es que no requiere instalación.
- **Font Finder:** esta extensión para Firefox nos permite analizar las diferentes tipografías que podemos encontrar en la web para luego modificarlas de la forma que creamos más conveniente.
- **The Taxonomy of Type:** este artículo de referencia ayuda a comprender un poco más la historia de la tipografía. Está en inglés, pero puede ser muy útil para comprender los elementos que tiene cada uno de los caracteres.
- **The Journal of Urban Typography:** se trata de una publicación periódica donde podremos encontrar el trabajo de otros, en formato Tumblr. Es una buena cartelera para encontrar inspiración.
- **Font Trainer:** es un juego para entrenar nuestra mirada a las diferencias que tienen las tipografías. Además de aprender cosas fundamentales, nos podemos divertir un buen rato.
- **Typo/graphic posters:** en la misma línea que TJOUT, este portal nos permite descubrir el trabajo de tipógrafos modernos o de inscripciones interesantes que puedan servir de inspiración.
- **WebFont Specimen:** antes de salir con una web, tenemos que ver que nuestra tipografía funcione. Con este sitio podemos probarlo antes de seleccionar la tipografía para ver cómo se verá en la versión final en la web.

<http://hipertextual.com/archivo/2012/10/10-recursos-online-de-tipografia/>

15. Tipografía y CSS

La declaración de una tipografía se termina con la familia genérica como la última de las opciones alternativas

```
p {  
  font-family:"Times New Roman",Georgia,serif;  
}
```

Las familias genéricas son:

- | | | |
|--------------|---|-----------------|
| • serif | → | Times New Roman |
| • sans-serif | → | Arial |
| • cursive | → | Comic Sans |
| • fantasy | → | Impact |
| • monospace | → | Courier New |

La compatibilidad de la tipografías depende del navegador y del sistema operativo

WOFF (Web Open Font Format)

Es un formato para usarse en páginas web. Fue desarrollado por la Fundación Mozilla, Opera Software y Microsoft, y está en el proceso de normalización por el W3C y convertirse en un estándar.

EOT (Embedded OpenType)

Estas fuentes son una forma compacta de OpenType para su uso como fuentes incrustadas en las páginas web y a su vez protegen los derechos de autor. Podemos vincular una tipografía sin que el usuario la tenga instalada.

SVG (Scalable Vector Graphics)

Es una especificación para describir gráficos vectoriales bidimensionales, estático o animado y fue creado como alternativa de formato abierto a Flash. se emplea como formato tipográfico en la web para dispositivos iOS (iPad, iPhone y iPod) .

TrueType

Es un formato estándar de tipos de letra escalables desarrollado inicialmente por Apple a finales de la década de los ochenta. Es compatible con Firefox 3.5+ , Safari 3.1+, Chrome 4+ y Opera 10+.

Podemos vincular una tipografía, que tengamos bajada, aunque el usuario no la tenga instalada:

```
<style type="text/css">
  @font-face {
    font-family: "Ubuntu";
    src: url("Ubuntu-Title.ttf") format("truetype");
  }
  p {
    font-family: Ubuntu, sans-serif;
  }
</style>
```

También podemos importar una tipografía online, hay dos métodos:

- En el estilo, importarla y luego llamarla

```
<style type="text/css">
  @import url(https://fonts.googleapis.com/css?family=Caesar+Dressing);
  p {
    font-family: "Caesar Dressing", sans-serif;
  }
</style>
```

- En el *head*, fuera de la declaración de estilo, enlazarla y luego llamarla en el estilo

```
<link rel="stylesheet" type="text/css"
href="https://fonts.googleapis.com/css?family=Mystery+Quest" />
<style type="text/css">
  p {
    font-family: "Mystery Quest", sans-serif;
  }
</style>
```

http://www.mclibre.org/consultar/htmlcss/css/css_fuentes_web.html

16. Cajas en CSS

Podemos maquetar asumiendo que cualquier elemento es una caja con:

- *content*: contenido *width* y *height* (en px o %) `p { width:90px; height:50px; }`
 - O establecer máximos o mínimos `p { min-width:100px; }`
- *margin*: margen exterior (en px o %) `p { margin: 20px; }`
- *padding*: margen interior (en px o %) `p { padding: 10px; }`
- *border*: línea que bordea el contenido
 - Línea sólida: `p { border-style: solid; }`
 - Línea discontinua: `p { border-style: dashed; }`
 - Línea de puntos: `p { border-style: dotted; }`
 - Línea doble: `p { border-style: double; }`
 - Surco: `p { border-style: groove; }`
 - Cresta: `p { border-style: ridge; }`
 - Relieve hundido: `p { border-style: inset; }`
 - Relieve saliente: `p { border-style: outset; }`
 - Color del borde: `p { border-color: blue; }`
 - Ancho del borde: `p { border-width: 5px; }`
 - Bordes redondeados `p { border-radius: 5px; }`

Podemos alterar los márgenes, bordes y paddings independientemente

<pre>p { border-top-style: dotted; border-right-style: solid; border-bottom-style: dotted; border-left-style: solid; }</pre>	Equivale a: <pre>p { border-style: dotted solid; }</pre>
--	---

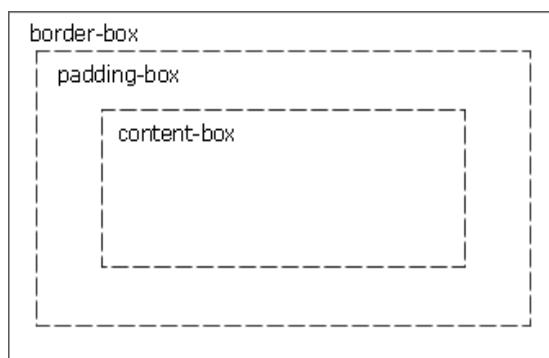
Podemos abreviar agrupando los valores, según esta pauta:

- 4 valores → uno para cada lado en el sentido de las agujas del reloj
- 3 valores → arriba [derecha izquierda] abajo
- 2 valores → [arriba abajo] [derecha izquierda]
- 1 valor → todos iguales

La pauta de agrupación funciona igual para los valores de márgenes y paddings

Formato de cajas

- Sombra `p { box-shadow: 5px 5px 5px red; }`
- Imagen como borde `p { border-image:url(border.png) 30 30 round;}`
- Color de fondo `p { background-color: red; }`
- Imagen de fondo, puede llenar sólo el contenido, hasta el padding o hasta el borde



```
.imagen { background-image: url(fondo.jpg);
            background-repeat: no-repeat;
            background-origin: content-box;
            background-clip: content-box;
            background-size: auto; }
```

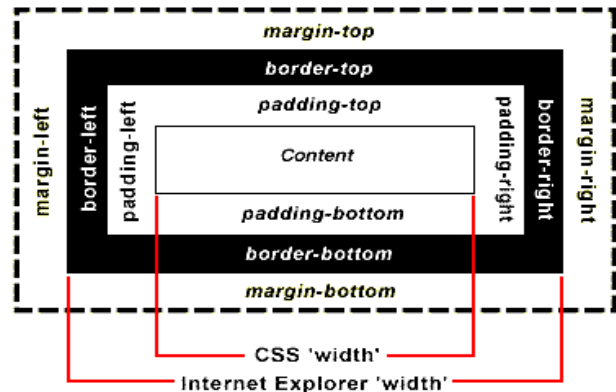
- Degradados lineales
 - Hacia una dirección `#d1 {background: linear-gradient(green, blue); }`
 - Diagonales `#d2 {background: linear-gradient(to left, green, blue); }`
 - Diagonales con grados `#d3 {background: linear-gradient(to top right, green, blue);}`
 - De más de dos colores `#d4 {background: linear-gradient(30deg, green, blue);}`
 - Con transparencia `#d5 {background: linear-gradient(red, yellow, green);}`
 - `#d6 {background: linear-gradient(to left, rgba(255,0,0,0),`
`rgba(255,0,0,1));}`
- Degradados radiales `#d7 {background: radial-gradient(orange 5%, red 20%,`
`black 60% );}`

Si detectamos errores de compatibilidad con algún navegador, debemos añadir su prefijo antes del valor que causa el conflicto

-webkit-	para Chrome o Safari
-o-	para Opera
-moz-	para Firefox

¿Cuánto mide una caja?

```
p {
  margin: 10px;
  padding: 10px;
  width: 250px;
  border: 5px solid gray;
  background-color: red;
}
```



El ancho total del elemento será:

$$250\text{px} + (10\text{px} * 2) + (5\text{px} * 2) + (10\text{px} * 2) = 300\text{px}$$

Posición

- Fija
- Relativa, respecto al flujo normal
- Absoluta, se salta el flujo normal
- Superpuestas, el valor más alto encima

```
.footer {position: fixed; }
.c1 {position: relative; top: 20px; }
.c2 {position: absolute; top: 20px; }
.c3 {position: absolute; z-index:2; }
```

Flotación

- Alinear a la izquierda:
- Alinear a la derecha:
- Romper la flotación

```
img { float: left; }
img { float: right; }
div { clear: both; }
```

Visibilidad

- Elemento invisible, sí ocupa espacio:

```
.escondido {visibility: hidden; }
```

Visualización

- Elemento invisible, no ocupa espacio:
- En línea
- De bloque
- Bloque que se agrupa en línea
- Flexible, los elementos se adaptan al espacio

```
.nulo { display: none; }
.linea { display: inline; }
.bloque {display:block; }
.acumulable {display: inline-block; }
.contenedor { display: flex; }
```

Permite más posibilidades de forma más simple:

Propiedades del contenedor *flex*

Direcció

flex-direction: row | row-reverse | column | column-reverse ;



Ajuste

flex-wrap: nowrap | wrap | wrap-reverse ;



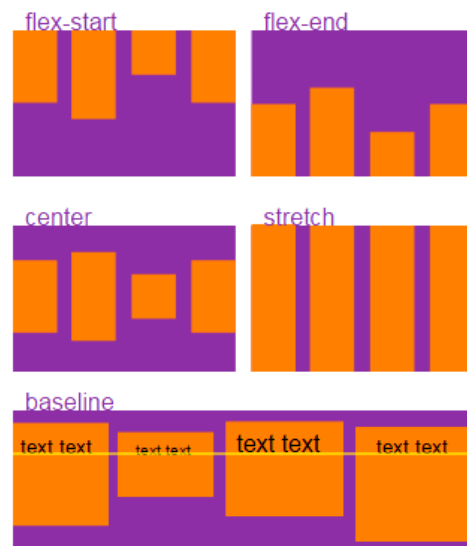
Alinear elementos en horizontal

justify-content: flex-start | flex-end | center | space-between | space-around ;



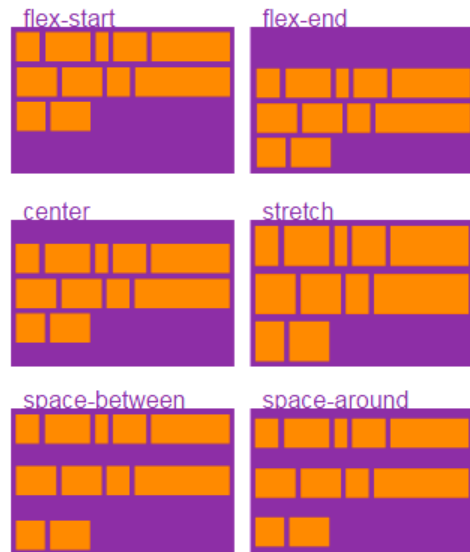
Alinear elementos en vertical

align-items: flex-start | flex-end | center | baseline | stretch ;



Alinear filas de elementos

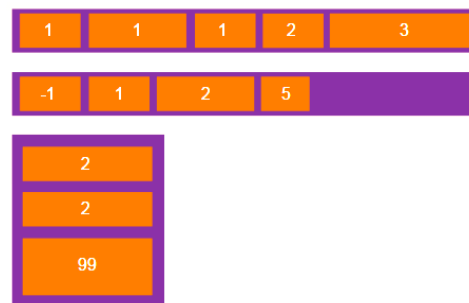
align-content: flex-start | flex-end | center | space-between | space-around | stretch;



Propiedades de los elementos de un contenedor flex

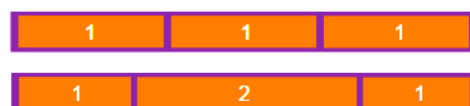
Orden de los elementos

order: <número entero>;



Tamaño

flex: <número entero>;



Alinear elementos individualmente

align-self: auto | flex-start | flex-end | center | baseline | stretch;



<http://css-tricks.com/snippets/css/a-guide-to-flexbox/> (en inglés)

20. Prioridad y selectores CSS

Prioridad

Cuando coinciden diversos estilos para el mismo elemento **por motivos de configuración**, la prioridad será, de menor a mayor:

- | | |
|--|------------------|
| • CSS por defecto del navegador | Navegador |
| • CSS en preferencias de usuario del navegador | Usuario |
| • CSS en hoja externa | Diseñador |
| • CSS en el head de la página | Diseñador |
| • CSS en línea | Diseñador |
| • CSS en el código de la página con !important | Diseñador |
| • CSS en preferencias de usuario del navegador con !important | Usuario |

Cuando coinciden diversos estilos en el código de la página, deberemos comprobar para cada estilo:

¿Es un estilo en línea? Sí = 1 / No = 0	¿Cuántos id? Nº	¿Cuántas clases? Nº	¿Cuántas marcas HTML? Nº
1º dígito	2º dígito	3º dígito	4º dígito

Estos 4 dígitos formarán un número, el estilo con el número más alto tiene prioridad
En caso de igualdad completa, tiene prioridad el último

Selectores CSS

Selector	Ejemplo	Aplica el estilo a...	CSS
<i>.class</i>	.intro	Todos los elementos que tengan el atributo class="intro"	1
<i>#id</i>	#firstname	El elemento que tenga asignado el atributo id="firstname"	1
<i>elemento</i>	p	Todos los elementos <p>	1
<i>elemento,elemento</i>	div,p	Todos los elementos <div> y <p>	1
<i>elemento elemento</i>	div p	Todos los elementos <p> que se encuentran dentro de los elementos <div>	1
:link	a:link	Todos los enlaces no visitados	1
:visited	a:visited	Todos los enlaces visitados	1
:active	a:active	Los enlaces cuando se seleccionan	1
:hover	a:hover	Cuando el puntero del ratón está encima del enlace	1
:first-letter	p:first-letter	La primera letra de cada elemento <p>	1
:first-line	p:first-line	La primera línea de cada elemento <p>	1
*	*	Todos los elementos	2
<i>elemento>elemento</i>	div>p	Todos los elementos <p> cuyo elemento contenedor (padre) sea el elemento <div>	2
<i>elemento+elemento</i>	div+p	Todos los elementos <p> que se encuentran inmediatamente después de elementos <div>	2
[<i>atributo</i>]	[target]	Todos los elementos que tengan el atributo target	2
[<i>atributo=valor</i>]	[target=_blank]	Todos los elementos con el atributo target=" _blank"	2
[<i>atributo~=valor</i>]	[title~=flor]	Todos los elementos que contengan la palabra "flor" en el atributo title	2
[<i>atributo =valor</i>]	[lang =es]	Todos los elementos con el	2

Selector	Ejemplo	Aplica el estilo a...	CSS
		atributo <i>lang</i> cuyo valor comience con "es"	
:focus	input:focus	El elemento input el cual tenga el foco	2
:first-child	p:first-child	Cada elemento <p> que sea el primer hijo de su padre	2
:before	p:before	Inserta contenido antes del contenido de cada elemento <p>	2
:after	p:after	Inserta contenido después de cada elemento <p>	2
:lang(lenguaje)	p:lang(es)	Cada elemento que tenga el atributo <i>lang</i> con el valor igual a "es"	2
elemento1~elemento2	p~ul	Cada elemento que esté precedido por el elemento <p>	3
[atributo^=valor]	a[src^="https"]	Cada elemento <a> donde el valor del atributo <i>src</i> comience con "https"	3
[atributo\$=valor]	a[src\$=".pdf"]	Cada elemento <a> donde el valor del atributo <i>src</i> termine con ".pdf"	3
[attribute*=value]	a[src*="w3schools"]	Cada elemento <a> donde el atributo <i>src</i> contenga la subcadena "w3schools"	3
:first-of-type	p:first-of-type	Cada elemento <p> que es el primer elemento <p> de su padre	3
:last-of-type	p:last-of-type	Cada elemento <p> que es el último elemento <p> de su padre	3
:only-of-type	p:only-of-type	Cada elemento <p> que es el único elemento <p> de su padre	3
:only-child	p:only-child	Cada elemento <p> que es el único hijo de su padre	3
:nth-child(n)	p:nth-child(2)	Cada elemento <p> que es el segundo hijo de su padre	3
:nth-last-child(n)	p:nth-last-child(2)	Cada elemento <p> que es el segundo hijo de su padre, contando desde el final	3
:nth-of-type(n)	p:nth-of-type(2)	Cada elemento <p> que es el segundo elemento <p> de su padre	3

Selector	Ejemplo	Aplica el estilo a...	CSS
:nth-last-of-type(<i>n</i>)	p:nth-last-of-type(2)	Cada elemento <p> que es el segundo elemento <p> de su padre, contando desde el final	3
:last-child	p:last-child	Cada elemento <p> que es el último hijo de su padre	3
:root	:root	La raíz del documento	3
:empty	p:empty	Cada elemento <p> que no es hijo (incluidos nodos de texto)	3
:target	#news:target	El elemento #news activo actualmente (pulsado en una URL que contenga el nombre de ancla)	3
:enabled	input:enabled	Cada elemento <input> activado	3
:disabled	input:disabled	Cada elemento <input> desactivado	3
:checked	input:checked	Cada elemento <input> chequeado	3
:not(<i>selector</i>)	:not(p)	Cada que no es un elemento <p>	3
::selection	::selection	La porción de un elemento que ha sido seleccionado por el usuario	-

21. Menús CSS

Podemos indicar que un elemento funciona como menú de navegación del sitio web dentro de la etiqueta semántica `<nav>` `</nav>`

Con el elemento `<ul>` creamos una lista de enlace y le aplicamos estilos CSS:

```
ul {  
    list-style-type: none;    /* ningún marcador de elemento de lista */  
    margin: 0px;             /* quitar márgenes de elemento de bloque */  
    padding: 0px;           /* quitar rellenos de elemento de bloque */  
}
```

Para crear un **menú vertical**, aplicamos estilos al elemento `<a>`:

```
a {  
  
    display: block;           /* convertir en elemento de bloque */  
    width: 100px;            /* definir una ancho fijo para los enlaces */  
}
```

Para crear un **menú horizontal**, aplicamos estilos al elemento `<li>`:

```
li {  
    display: inline;          /* convertir en elemento en línea */  
}
```

Para que los enlaces de la lista tengan todos el mismo ancho, a los elementos `<li>` y `<a>` hay que aplicarles las siguientes propiedades:

```
li {  
    float: left;              /* flotar el elemento lista */  
}  
a {  
    display: block;           /* convertir en elemento de bloque */  
    width: 100px;            /* asignar un ancho fijo */  
}
```

Un ejemplo: menú vertical

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Menú vertical</title>
  <style type="text/css">
    h1 {
      font-family: Arial;
      color: navy;
      font-size: 40px;
      font-weight: bold;
    }
    nav{
      width: 150px;
    }
    ul{
      list-style-type: none;
      margin: 0px;
      padding: 0px;
    }
    a {
      font-family: Arial;
      color: #000;
      font-size: 16px;
      text-decoration: none;
      display: block;
      padding: 6px;
      border: 1px solid #CCC;
      background-color: #EEE;
    }
    a:hover {
      color: navy;
      background-color: #DDD;
      border-right: 5px solid navy;
    }
    a.featured:hover {
      border-right: 5px solid red;
    }
  </style>
</head>
<body>
  <nav>
    <h1>Discover</h1>
    <ul>
      <li><a href="#" class="welcome">Welcome</a></li>
      :
      <li><a href="#" class="radio">Radio</a></li>
    </ul>
  </nav>
</body>
</html>
```

Otro ejemplo: **menú horizontal**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Menú horizontal</title>
  <style type="text/css">
    h1 {
      font-family: Arial;
      color: navy;
      font-size: 40px;
      font-weight: bold;
    }
    ul{
      list-style-type: none;
      margin: 0px;
      padding: 0px;
    }
    li{
      display: inline;
    }
    a {
      font-family: Arial;
      color: #000;
      text-decoration: none;
      display: block;
      padding: 6px;
      float: left;
      background-color: #eee;
    }
    a:hover {
      color: navy;
      background-color: #ddd;
      border-bottom: 5px solid navy;
    }
    a.news:hover {
      border-bottom: 5px solid red;
    }
  </style>
</head>
<body>
  <nav>
    <h1>BBC</h1>
    <ul>
      <li><a href="#" class="news">News</a></li>
      :
      <li><a href="#" class="sport">Sport</a></li>
    </ul>
  </nav>
</body>
</html>
```


21. Menús CSS

Podemos indicar que un elemento funciona como menú de navegación del sitio web dentro de la etiqueta semántica `<nav>` `</nav>`

Con el elemento `<ul>` creamos una lista de enlace y le aplicamos estilos CSS:

```
ul {  
    list-style-type: none;    /* ningún marcador de elemento de lista */  
    margin: 0px;             /* quitar márgenes de elemento de bloque */  
    padding: 0px;           /* quitar rellenos de elemento de bloque */  
}
```

Para crear un **menú vertical**, aplicamos estilos al elemento `<a>`:

```
a {  
  
    display: block;          /* convertir en elemento de bloque */  
    width: 100px;           /* definir una ancho fijo para los enlaces */  
}
```

Para crear un **menú horizontal**, aplicamos estilos al elemento `<li>`:

```
li {  
    display: inline;         /* convertir en elemento en línea */  
}
```

Para que los enlaces de la lista tengan todos el mismo ancho, a los elementos `<li>` y `<a>` hay que aplicarles las siguientes propiedades:

```
li {  
    float: left;            /* flotar el elemento lista */  
}  
a {  
    display: block;         /* convertir en elemento de bloque */  
    width: 100px;          /* asignar un ancho fijo */  
}
```

Un ejemplo: menú vertical

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Menú vertical</title>
  <style type="text/css">
    h1 {
      font-family: Arial;
      color: navy;
      font-size: 40px;
      font-weight: bold;
    }
    nav{
      width: 150px;
    }
    ul{
      list-style-type: none;
      margin: 0px;
      padding: 0px;
    }
    a {
      font-family: Arial;
      color: #000;
      font-size: 16px;
      text-decoration: none;
      display: block;
      padding: 6px;
      border: 1px solid #CCC;
      background-color: #EEE;
    }
    a:hover {
      color: navy;
      background-color: #DDD;
      border-right: 5px solid navy;
    }
    a.featured:hover {
      border-right: 5px solid red;
    }
  </style>
</head>
<body>
  <nav>
    <h1>Discover</h1>
    <ul>
      <li><a href="#" class="welcome">Welcome</a></li>
      :
      <li><a href="#" class="radio">Radio</a></li>
    </ul>
  </nav>
</body>
</html>
```

Otro ejemplo: **menú horizontal**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Menú horizontal</title>
  <style type="text/css">
    h1 {
      font-family: Arial;
      color: navy;
      font-size: 40px;
      font-weight: bold;
    }
    ul{
      list-style-type: none;
      margin: 0px;
      padding: 0px;
    }
    li{
      display: inline;
    }
    a {
      font-family: Arial;
      color: #000;
      text-decoration: none;
      display: block;
      padding: 6px;
      float: left;
      background-color: #eee;
    }
    a:hover {
      color: navy;
      background-color: #ddd;
      border-bottom: 5px solid navy;
    }
    a.news:hover {
      border-bottom: 5px solid red;
    }
  </style>
</head>
<body>
  <nav>
    <h1>BBC</h1>
    <ul>
      <li><a href="#" class="news">News</a></li>
      :
      <li><a href="#" class="sport">Sport</a></li>
    </ul>
  </nav>
</body>
</html>
```

17. Accesibilidad

¿Qué es la Accesibilidad Web?

Hablar de *Accesibilidad Web* es hablar de un acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.

Con esta idea de accesibilidad nace la Iniciativa de Accesibilidad Web, conocida como WAI (Web Accessibility Initiative). Se trata de una actividad desarrollada por el W3C, cuyo objetivo es facilitar el acceso de las personas con discapacidad, desarrollando pautas de accesibilidad, mejorando las herramientas para la evaluación y reparación de accesibilidad Web, llevando a cabo una labor educativa y de concienciación en relación a la importancia del diseño accesible de páginas Web, y abriendo nuevos campos en accesibilidad a través de la investigación en este área.

¿Para qué sirve?

La idea principal radica en hacer la Web más accesible para todos los usuarios independientemente de las circunstancias y los dispositivos involucrados a la hora de acceder a la información. Partiendo de esta idea, una página accesible lo sería tanto para una persona con discapacidad, como para cualquier otra persona que se encuentre bajo circunstancias externas que dificulten su acceso a la información (en caso de ruidos externos, en situaciones donde nuestra atención visual y auditiva no están disponibles, pantallas con visibilidad reducida, etc.).

¿Cómo funciona?

Para hacer el contenido Web accesible, se han desarrollado las denominadas Pautas de Accesibilidad al Contenido en la Web (WCAG), cuya función principal es guiar el diseño de páginas Web hacia un diseño accesible, reduciendo de esta forma barreras a la información. WCAG consiste en 14 pautas que proporcionan soluciones de diseño y que utilizan como ejemplo situaciones comunes en las que el diseño de una página puede producir problemas de acceso a la información. Las Pautas contienen además una serie de puntos de verificación que ayudan a detectar posibles errores.

Cada punto de verificación está asignado a uno de los tres niveles de prioridad establecidos por las pautas.

- **Prioridad 1:** son aquellos puntos que un desarrollador Web tiene que cumplir ya que, de otra manera, ciertos grupos de usuarios **no podrían acceder** a la información del sitio Web.
- **Prioridad 2:** son aquellos puntos que un desarrollador Web debería cumplir ya que, si no fuese así, sería **muy difícil acceder** a la información para ciertos grupos de usuarios.
- **Prioridad 3:** son aquellos puntos que un desarrollador Web debería cumplir ya que, de otra forma, algunos usuarios experimentarían **ciertas dificultades para acceder** a la información.

En función a estos puntos de verificación se establecen los niveles de conformidad:

- **Nivel de Conformidad "A":** todos los puntos de verificación de prioridad 1 se satisfacen.
 - Elementos no textuales tienen texto equivalente.
 - Que la información transmitida con colores esté disponible sin colores.
 - Identificar claramente los cambios en el idioma del texto.
 - Que el documento se pueda leer sin hojas de estilo.
 - Se ha utilizado un lenguaje simple y claro.
 - Que las imágenes y mapas de imágenes tengan enlaces equivalentes en texto.
 - En las tablas que se identifiquen los encabezamientos de filas y columnas.
 - Que los encabezamientos de niveles tengan marcadores para asociar celdas con encabezados.
 - Que los marcos tengan su correspondiente título.
 - Que si hay scripts, cuando no se ejecuten siga siendo accesible.
 - Si hay vídeo o audio, que haya una descripción alternativa.
 - Si el sitio no es accesible, proporcionar un enlace al sitio que sí lo sea.
- **Nivel de Conformidad "Doble A":** todos los puntos de verificación de prioridad 1 y 2 se satisfacen.

Además de los de Nivel de Conformidad A:

- Suficiente contraste entre el color de fondo y de primer plano.
- Existan marcadores apropiados.
- Usar hojas de estilo para la presentación y la maquetación.
- Usar unidades relativas en lugar de absolutas.
- Usar encabezados para la estructura lógica del documento.
- Marcar correctamente las listas y elementos.
- Las citas que existan marcadas correctamente.
- Sin redirecciones automáticas a otros sitios web.
- No se cambia de ventana sin previo aviso.
- Haya un mapa del sitio o tabla de contenido.
- La navegación sea coherente

- Hayan metadatos para añadir información semántica.
 - Que las tablas no se usen para maquetar.
 - Que se describan el fin de los marcos y como se relacionan entre ellos.
 - En formularios, que haya relación entre el campo y su etiqueta.
 - En formularios, todo elemento asociado explícitamente con su etiqueta.
 - Con scripts, que los manejadores de eventos sean independientes del dispositivo de entrada.
 - Usar manejadores de eventos lógicos en vez de dispositivo.
- **Nivel de Conformidad "Triple A":** todos los puntos de verificación de prioridad 1,2 y 3 se satisfacen.

Además de los de **Nivel de Conformidad "Doble A":**

- La explicación de la abreviatura en su primera aparición.
- Se indica el idioma del documento.
- Se establece un orden lógico de navegación.
- Se establecen atajos de teclado en los vínculos más importantes.
- Se proporcionan barras de navegación.
- Se utilizan mecanismos de búsqueda.
- La información destacada al principio de los encabezamientos, etc.
- Que las imágenes tengan un enlace que las explique textualmente.
- En tablas, proporcionar resúmenes de las listas.
- En tablas, proporcionar abreviaturas para las etiquetas de encabezado.
- En formularios, que se incluyan texto por defecto para cada uno de los elementos

Las pautas describen cómo hacer páginas Web accesibles sin sacrificar el diseño, ofreciendo esa flexibilidad que es necesaria para que la información sea accesible bajo diferentes situaciones y proporcionando métodos que permiten su transformación en páginas útiles e inteligibles.

Igualmente, se han desarrollado Pautas de Accesibilidad para Herramientas de Autor, cuyo objetivo es ayudar a los desarrolladores de software a la hora de crear herramientas de autor para producir contenido Web accesible. También se han desarrollado Pautas de Accesibilidad para XML, donde se explica cómo asegurar la accesibilidad de aplicaciones basadas en XML. Y por último, Pautas de Accesibilidad para Agentes de Usuario 1.0, donde se explica cómo hacer accesible los navegadores, reproductores multimedia y otras tecnologías asistivas.

Por otro lado, se han desarrollado otro tipo de documentos como las Técnicas para Pautas de Accesibilidad al Contenido en la Web, que ofrecen una serie de ejemplos de etiquetado y explicaciones muy detalladas de cómo implementar las Pautas de Accesibilidad al contenido en la Web. Entre ellas se pueden destacar Técnicas esenciales para Pautas de Accesibilidad al Contenido en la Web 1.0, las Técnicas HTML para Pautas de Accesibilidad al Contenido a la Web 1.0 y las Técnicas CSS

para Pautas de Accesibilidad al Contenido en la Web 1.0.

Ejemplos

Un ejemplo de *prioridad 1* sería la identificación clara de cualquier cambio de idioma que se pueda producir en el texto de un documento. Es decir, si se utilizan diferentes idiomas es necesario que cualquier cambio esté claramente señalado con el atributo `lang`:

Un ejemplo de código correcto sería el siguiente:

```
<p>  
  <q>Buenos días Philip</q>  
  <q lang="en">Good morning</q>. Respondió Philip en inglés.  
  <q>¿Qué tal estás?</q>  
  <q lang="fr">Très bien.</q> Volvió a responder, pero esta vez en francés.  
</p>
```

Otro ejemplo de *prioridad 1* sería la utilización del atributo `alt` para incorporar texto equivalente al contenido de una imagen cuando se quieren utilizar gráficos; esto permitiría que dispositivos o personas que no pueden visualizar los gráficos, obtengan una representación alternativa textual. El código correspondiente sería:

```

```



<http://w3c.es/Divulgacion/GuiasBreves/Accesibilidad>

Las 75 directrices de accesibilidad de Jakob Nielsen

A partir de un estudio con personas con baja visión, invidentes o con discapacidad motora

Marca con * las directrices que te parece básico cumplir

Marca con ? las directrices que te parecen más conflictivo cumplir

1. Siga las reglas básicas de un buen diseño

Destaca cuatro normas del buen diseño:

- Diseño centrado en el usuario
- Escribir concisamente evitando el lenguaje comercial superfluo
- Ofertar menos opciones, incluyendo sólo las más importantes
- No incluir gráficos y sonidos sólo por contar con ellos

Gráficos y multimedia

2. Reducir al mínimo el uso de imágenes

Argumenta que reducir el número de imágenes permite aumentar la velocidad de carga de las páginas y disminuye el ruido superfluo, puesto que un exceso de imágenes dificulta comprender la finalidad del sitio.

3. Dar a todos los gráficos (incluso a los banners publicitarios) nombres que sean comprensibles y que transmitan de verdad lo que el gráfico es y hace

Utilice el atributo ALT para describir brevemente las imágenes, y el atributo LONGDESC para describirlas minuciosamente.

El contenido del ALT debe ser conciso y sencillo: su objetivo es transmitir lo que es la imagen, no describirla.

4. Nunca difuminar las imágenes para indicar no disponibilidad

Se refiere por ejemplo a una imagen que funciona como enlace, de tal manera que cuando el enlace no está disponible se indica difuminándola o poniéndola borrosa.

Es mejor idea eliminar por completo los gráficos cuando no están disponibles.

5. Cuando gráficos contengan información útil, también proporcione la información en texto

Cuando los gráficos comprenden información útil, los usuarios deberían ser capaces de acceder a dicha información en formato HTML.

6. Dé a los usuarios maneras alternativas de obtener la información contenida en cualquiera de los gráficos que se encuentren

Muchas veces los usuarios encuentran el gráfico y suponen que es la única fuente de información, pues les pasa desapercibido que hay una versión textual de la misma información. Para que sea más fácil de encontrar debe ir junto al gráfico o vinculado al mismo y no, por ejemplo, en una sección completamente diferente del sitio.

7. No utilice una imagen en miniatura de la página de su sitio para utilizarla como gráfico (o botón) en otra página

Se refiere a utilizar imágenes que son una captura de la pantalla, en pequeño, a modo de enlace para ir a otra página.

Cuando ves la página en su totalidad los tamaños relativos hacen que este uso sea perfectamente claro. Sin embargo, para los usuarios que utilizan magnificadores de pantalla, a 6x de aumento, las imágenes en miniatura de una página adquieren un aspecto real.

8. Al hacer uso de gráficos, elija siempre imágenes claras y nítidas

Las fotos de textos también plantean problemas, ya que las letras tienden a hacerse borrosas a medida que se amplían.

9. Facilite a los usuarios la posibilidad de saltarse cualquier elemento multimedia, aplicación Java o Flash

En especial cuando producen sonidos, pues los usuarios que usan un lector de pantalla necesitan silencio para escucharlo.

10. No cree automáticamente una versión sólo-texto de su sitio

No es aconsejable hacer una versión solo-texto, lo recomendable es hacer accesible la versión gráfica, pues crear y mantener dos versiones del sitio es costoso y lleva mucho tiempo.

Sin embargo, hay empresas que utilizan el grafismo y los elementos multimedia para transmitir cierta impresión, sentimiento o emoción de su sitio. En estos casos sí que puede estar justificado una versión sólo-texto, en cuyo texto se intente transmitir el mensaje que se envía a través del multimedia.

Pero hay que tener en cuenta que esa versión sólo-texto debe ser accesible, y que la página principal desde la que se accede a esta versión también debe ser accesible.

Pops-ups, rollovers, nuevas ventanas y menús en cascada

11. Evite el uso de ventanas emergentes

Incluye ejemplos de lo mucho que desorientan y desconciertan las ventanas emergentes a las personas que utilizan lectores de pantalla o magnificadores así como a las personas con discapacidad motora.

12. En el caso de hacer uso de cuadros de diálogo en pop-ups, asegúrese de que la acción por defecto es la más "perdonable"

Ciertos usuarios pulsaban "Enter" sin más en estas ventanas, sin pararse a leer o a comprobar si la acción por defecto era Aceptar o Cancelar, de manera que se encontraban inadvertidamente instalándose, por ejemplo, Flash Player, algo que no querían ni esperaban hacer.

13. Evite abrir nuevas ventanas del navegador

Al igual que las ventanas emergentes, las nuevas ventanas generan desconcierto en el usuario. Comprobaron que a menudo las ventanas quedaban ocultas unas por otras y cuando el usuario trataba de hacer clic en el botón "Atrás" del navegador éste parecía no funcionar, y por lo general terminan cerrando el navegador para recuperarlo por completo.

14. Si abre nuevas ventanas del navegador, siempre proporcione una forma sencilla de volver a la página principal del sitio principal

A veces las personas que utilizan lectores de pantalla no se dan cuenta de que hay varias ventanas del navegador abiertas a la vez. Si un sitio abre automáticamente una ventana y el botón "Atrás" falla, buscan un vínculo para acceder a la página

principal.

15. No confíe en los rollover de texto (tooltip) para transmitir información

Se refiere a cualquier texto que aparece cuando te pones encima de un elemento. Asegúrese de suministrar la misma información de otra forma más fácilmente accesible, pues muchos usuarios no llevan a cabo la acción o no pueden acceder a la misma. Pone ejemplos de los problemas que les surgen a los usuarios de magnificadores de pantalla.

16. Evite utilizar menús en cascada (menús que se despliegan)

Son muy difíciles de usar para usuarios con magnificadores de pantalla y con problemas motores, pues es necesario poder arrastrar y sostener el ratón mientras se hace clic con precisión. Estos menús son difíciles de seguir, además, con un magnificador de pantalla, hay zonas importantes del mismo que caen fuera de la pantalla y no son visibles.

Enlaces y botones

17. Limite el número de enlaces en una página

Recomienda un máximo de 20 enlaces por página. Los usuarios con lectores de pantalla no pueden echar un vistazo a la página, deben escucharla. Algunos usuarios no piensan en hacer clic hasta que no han escuchado toda la página, otros hacen clic en el primer vínculo que creen que es el adecuado. Aquellos que no utilizaban el ratón por problemas motores acababan agotados de moverse con las teclas de desplazamiento cuando la página era muy grande y había muchos enlaces.

18. Evite pequeños botones y enlaces con texto minúsculo

Es difícil y agotador acertar a pulsarlos para las personas con problemas motores.

19. Deje espacio entre los enlaces y botones

Para las personas con problemas motores o problemas de baja visión, el espacio entre los botones y entre los enlaces es sumamente importante, pero también para los usuarios videntes, cuando los enlaces están muy juntos es muy fácil errar y pulsar otro por equivocación.

20. Evite el uso de imágenes como el único método de enlazar a otra página

También se debe evitar decirles a los usuarios instrucciones del tipo "haga clic en la

imagen de arriba".

21. Asegúrese de que los comandos importantes aparecen con sus propios vínculos exclusivos

Es necesario comprobar que con un lector de pantalla hay pausas suficientes que permitan identificar los enlaces importantes.

22. Subraye todos los enlaces

Es la manera más clara de identificar los enlaces cuando la pantalla se magnifica.

23. Crear vínculos dentro del texto cuando tenga sentido, utilice los botones adicionales sólo cuando sea necesario

Es algo muy típico que en las noticias de un sitio, en vez de que el enlace a la noticia sea el texto, se ponga un botón "Ir". Se ha de evitar esta práctica.

Organización de las páginas

Una buena organización de la página no significa lo mismo para una persona que ve que para una persona que utiliza un lector de pantalla. Para esta última una buena organización será por ejemplo que lo primero que se lea sea el título de la página y los primeros enlaces que se dicten sean los de búsqueda, saltar navegación o versión sólo-texto.

El estudio propone que estos enlaces se hagan mediante imágenes invisibles al comienzo de la página; serían invisible para los videntes y disponibles para los lectores de pantalla y los navegadores que proporcionen listados de enlaces.

24. Confirmar al inicio de la carga de la página principal el nombre de la compañía

Este dato debe ser el primero que lea el lector de pantalla, así sus usuarios, que no pueden oír la página, no dudarán acerca de a dónde han entrado. Para ello es necesario ponerle un "title" significativo a la página y un "alt" al logotipo de la parte superior de la página.

25. Confirmar al inicio de la carga de una página en qué página se está

Los usuarios con un lector de pantalla necesitan saber que al hacer clic en un vínculo este les ha traído a donde pensaban, por ello es necesario, por ejemplo, poner un "title" significativo a todas las páginas.

26. No asociar la palabra "página principal" o "home" con el logotipo de su empresa si planea volver a utilizar el mismo gráfico en todas las páginas

Cuando se usa el logo como enlace a la página principal, el "alt" de esta imagen no debería ser "página principal de XXXX" sino "Enlace a la página principal de XXXX", para que los usuarios de lectores de pantalla no piensen que es el título de la página sino un enlace.

27. Reducir la necesidad de desplazarse con scroll

Escrolar es muy lento y tedioso para los usuarios que usan magnificadores de pantalla.

28. Cuando los usuarios deban hacer una elección, ponga todas las posibilidades en la misma zona

Hay que buscar un equilibrio entre poner los enlaces y botones muy juntos o muy distantes.

29. Cuando los usuarios deben hacer una elección, adviértales de la elección mediante, por ejemplo, un texto del tipo "Selección de pedido", e infórmeles de la cantidad de opciones que tienen

30. Haga un diseño de páginas consistente

Utilice un método de navegación coherente, que no se deba reaprender en cada página. Esto será especialmente útil para los usuarios con magnificadores de pantalla.

31. Considere la posibilidad de utilizar un enlace "Saltar Enlaces" para que los usuarios puedan saltar los enlaces o elementos de navegación

Esto les evita a los lectores de pantalla tener que escuchar una y otra vez todos los enlaces de las barras de navegación superiores y laterales.

32. Elija una dirección Web de su sitio simple e informativa, y mantenga esa URL en el campo de la dirección de la página después de la carga

Puesto que es lo primero que muchos usuarios con lectores de pantalla escuchan, es importante elegir una dirección que transmita el nombre de la organización o del sitio.

Páginas

33. Evite las "páginas portada" previas a la página de inicio de un sitio, haga que la primera página que la gente vea sea la página que mejor describe la compañía y el sitio

Las páginas y los clic extras no sólo no ayudan a la eficiencia del sitio, sino que muchas veces conducen a la frustración de los usuarios que usan lectores de pantalla.

34. Incluya tan sólo los pasos y las páginas necesarias

Campos y formularios

35. Limite la cantidad de información que el formulario requiere; recoja sólo el mínimo necesario

Los formularios demasiado largos suponen problemas para los usuarios con discapacidad motora, puesto que les cuesta rellenar los campos. Además, los usuarios que usan magnificadores de pantalla pierden fácilmente el contexto al intentar escrolar la página.

36. Ponga las etiquetas de texto de los campos muy cerca de los campos que les corresponden

Cuando se magnifica la pantalla lo más probable es que los usuarios no vean la etiqueta que acompaña al campo, y asumirán que le corresponde el texto más cercano.

37. No indique los errores del formulario simplemente destacando un texto en rojo o amarillo

Sin duda esta práctica es una ayuda, pero los usuarios que usan lectores pantalla o dispositivos Braille no tienen manera de saber que el texto es rojo. Estos textos también son difíciles de leer con la pantalla magnificada, más aun cuando a menudo se invierten los colores para ver las cosas mejor.

Propone que cuando haya errores se muestren sólo los campos con errores.

38. No confíe sólo en el asterisco para indicar que un campo es necesario

Algunos usuarios que usaban lectores de pantalla se preguntaban qué significaba la continua repetición de la palabra "asterisco" y si sería un error del lector.

El estudio propone reforzar este indicador con otro, como por ejemplo poner el texto en negrita. También propone poner los campos obligatorios todos juntos al

principio del formulario y evitar solicitar información que no sea necesaria.

39. Asegúrese de que el orden de tabulación es lógico

Muchos usuarios usan el tabulador para desplazarse por los campos del formulario. Además, los usuarios de lectores de pantalla son también dependientes del orden lógico de los campos, puesto que es así como el lector de pantalla les lleva a través del formulario.

40. Haga coincidir el orden de tabulación con la disposición visual de los elementos cuando sea posible

41. Apile los campos en una columna vertical

Los campos en una única columna son mucho más fáciles de rellenar para los usuarios con baja visión que utilizan magnificadores de pantalla.

42. Ofrezca campos de entrada estándar para los números de teléfono

Cuando se ofrecen varios campos para rellenar el teléfono la gente se confunde. Es mejor usar un único campo (o dos, uno para el prefijo y otro para el número) pero no más, especialmente si no se etiquetan.

43. En aquellas páginas que tenga un formulario con un único campo de entrada o selección, poner el botón lo más cerca posible del campo

Muchos usuarios de magnificadores de pantalla no se dan cuenta de que deben desplazarse y encontrar el botón, esperan, por ejemplo, que ocurra algo al seleccionar en la combo una opción.

44. En los formularios, ponga el botón "submit" lo más cerca posible del último campo

A menudo los usuarios de baja visión no encuentran el botón "enviar", más aún si se separa del formulario por ejemplo mediante una línea. También es importante que la tecla "Intro" envíe el formulario.

45. Ponga cualquier instrucción relativa a un campo en particular antes del campo y no después

Los usuarios de lectores de pantalla necesitan escuchar las instrucciones acerca de los campos antes de llegar a la caja de entrada.

46. Considere cuidadosamente cuánto tiempo se pone de tope para hacer una acción como por ejemplo rellenar un formulario

Textos

47. Seleccione colores de texto con buen contraste

48. No use un tamaño de fuente muy pequeño para el texto de la página

El usuario debe poder controlar el tamaño del texto, para lo cual deben tener un tamaño relativo (% , em). Se aconseja que por defecto no tenga un tamaño menor al equivalente de 11 puntos, puesto que así las letras seguirán siendo legibles cuando se magnifiquen.

49. No use texto pequeño o sutil (por ejemplo un gris rebajado) para los encabezados y categorías

50. Cree siempre un buen contraste entre el color del texto y el color de fondo de la página

Cuando no es así muchos usuarios optan por invertir los colores o verla en monocromo.

51. No se base en una imagen de fondo para crear el contraste con el texto

Muchos usuarios con baja visión desactivan la visualización de imágenes en el navegador. Asegúrese de que cuando desactiva las imágenes el color de texto sigue teniendo suficiente contraste con el color de fondo de la página.

52. Pruebe los colores y fuentes de su sitio con un magnificador de pantalla

Hasta a x6. El azul por defecto de los enlaces es poco legible a muchos aumentos, el negro sin embargo sí es legible. El texto de las imágenes tampoco debería estar suavizado para que se pueda leer magnificado.

53. Asegúrese de que es posible magnificar su sitio

Utilice tamaños de fuente relativos (% , em), no absolutos (px , pt)

54. Escriba de forma concisa y elimine el texto superfluo

Tanto a los usuarios de lectores de pantalla como a los de magnificadores de pantalla les molesta escuchar información inútil.

55. Si el nombre de la compañía o el texto en general presenta abreviaturas y acrónimos, dígame al lector de pantalla como pronunciarlo mediante las etiquetas <abbr> y <acronym>

56. Replantéese la forma en la que usa los paréntesis y los asteriscos.

La utilización de estos signos de puntuación distraen a menudo a la gente.

Búsquedas

57. Ofrezca un motor de búsqueda que perdone los errores de ortografía

Google, por ejemplo, ofrece recomendaciones de búsqueda con las palabras escritas correctamente.

58. No base únicamente la capacidad de búsqueda de un sitio en la interfaz de navegación.

Ofrezca siempre una herramienta de búsqueda.

59. Coloque la caja de búsqueda donde los usuarios la esperan encontrar, no en una zona inesperada

Se espera en la parte superior derecha o en la esquina inferior izquierda.

60. Describa claramente los resultados de la búsqueda

Indique de inmediato cuántos resultados ha habido, ponga un título para cada uno y no sólo una url.

61. Avise a los usuarios cuando no han introducido nada en la caja de búsqueda

Si un usuario escribe algo sin tener la caja de búsqueda seleccionada, creará haber escrito algo y que los resultados de la búsqueda se corresponden con lo que ha escrito, sin embargo no tendrán nada que ver con lo que intentaba buscar porque se buscó con la caja vacía.

62. No presente el ranking de relevancia de los resultados de la búsqueda en una tabla

El número que precede al enlace y que indica el ranking, si va en una columna confunde a los usuarios de lectores de pantalla y a los de magnificadores de pantalla, pues no lo contextualizan. Si los resultados se dan en orden, el número es

redundante y no es necesario.

Comercio electrónico

63. Describa minuciosamente las imágenes de los productos que el sitio vende como si no hubiera imágenes

En la compra de ropa, objetos, etc. la imagen es muy importante, por eso las personas que usan lectores de pantalla necesitan una descripción detallada para decidirse a comprar.

64. Ayude a los usuarios a seguir comprando después de haber añadido algo al carrito, dándoles una forma de llegar de nuevo a donde estaban

65. Coloque los botones de "añadir a la cesta de compra" o "realizar pedido" cerca de los elementos a comprar

Si se colocan en la parte superior o inferior de la pantalla pueden pasar desapercibidos.

También debería ser evidente para el usuario cuando se ha añadido algo a la cesta.

66. Tenga en cuenta a los clientes internacionales seleccionando cuidadosamente los términos que utiliza

Por ejemplo es más fácil de comprender "Buy" que "Checkout".

Aconseja usar el atributo "lang" cuando se utilicen frases o palabras en otro idioma, de este modo se advierte a los usuarios de lectores de pantalla que la palabra que van a oír está en otro idioma.

Tablas y frames

67. Evite el uso de tablas para maquetar un sitio

Las tablas se esperan para organizar la información, cuando se utilizan para fijar el tamaño de la página crean confusión a los usuarios que utilizan lectores de pantalla y dispositivos Braille.

68. Evite el uso de tablas muy grandes sin motivo; si debe usarlas, considere proporcionar la información también en formato texto

Con tanta información en una tabla es difícil que los usuarios de lectores de pantalla, de sistemas Braille o de magnificadores de pantalla den sentido a todo y lo

recuerden.

69. En especial en las tablas, no use gráficos para indicar un dato como por ejemplo el estado, es mejor usar texto

Es difícil recordar la leyenda o recordar que la presencia o ausencia de una imagen significa algo en particular.

70. Asegúrese de que las listas alfabéticas visuales siguen estando en orden alfabético cuando son leídas por un lector de pantalla

Si la lista está dispuesta en una tabla puede ser que el orden no coincida.

71. Incluya un resumen ("summary") en todas las tablas

El usuario con un lector de pantalla no puede echar un vistazo a la tabla para hacerse una idea de lo que contiene, el resumen de la tabla cumple esta función.

72. Antes de usar columnas en el diseño considere cómo se le mostrarán a un usuario con un magnificador de pantalla

Las páginas organizadas en columnas se prestan a confusión cuando no se tiene el contexto de toda la pantalla, como les ocurre a los usuarios con magnificadores de pantalla.

73. Describa todos los frames

Los frames son una de las causas más comunes de problemas de accesibilidad, especialmente con lectores de pantalla y dispositivos Braille. Pero si los usas, al menos indica cuántos frames hay en una página y proporciona un resumen de cada uno en el "longdesc", de este modo los usuarios son conscientes de los contenidos generales antes de decidirse a escuchar un frame entero. Acerca de los marcos comenta que si se usa el aumento de fuente del navegador, este sólo se aplica al marco seleccionado, y es muy tedioso ir seleccionando cada marco para subir el tamaño de fuente.

Confianza, estrategia e imagen de la compañía

Cuando los usuarios se encuentran con sitios de difícil acceso es poco probable que vuelvan y están menos satisfechos con esa empresa u organización. Por el contrario, **las empresas y organizaciones con sitios accesibles generan más confianza.**

74. Proporcione en su sitio un servicio de atención al cliente cuyas personas tengan unos básicos conocimientos de los problemas de accesibilidad

75. No se refiera a la gente en silla de ruedas como “silla de ruedas” o lectores de pantalla como “lectores de pantalla”

<http://olgacarreras.blogspot.com.es/2007/02/las-75-directrices-de-accesibilidad-de.html>

Busca en internet validadores de accesibilidad online, compáralos entre ellos

Los 14 consejos básicos según discapnet.es

1. Proporcione alternativas equivalentes para el contenido visual y auditivo
2. No se base sólo en el color
3. Utilice marcadores y hojas de estilo y hágalo apropiadamente
4. Identifique el idioma usado
5. Cree tablas que se transformen correctamente
6. Asegúrese de que las páginas que incorporen nuevas tecnologías se transformen correctamente
7. Asegure al usuario el control sobre los cambios de los contenidos temporales
8. Asegure la accesibilidad directa de las interfaces incrustadas
9. Diseñe para la independencia del dispositivo
10. Utilice soluciones provisionales
11. Utilice las tecnologías y pautas W3C
12. Proporcione información de contexto y orientación
13. Proporcione mecanismos claros de navegación
14. Asegúrese de que los documentos sean claros y simples

Los 10 consejos básicos según w3.org

1. Imágenes y animaciones: Use el atributo alt para describir la función de cada elemento visual
2. Mapas de imagen: Use el elemento map y texto para las zonas activas
3. Multimedia: Proporcione subtítulos y transcripción del sonido, y descripción del vídeo
4. Enlaces de hipertexto: Use texto que tenga sentido leído fuera de contexto. Por ejemplo, evite "pincha aquí"

5. Organización de las páginas: Use encabezados, listas y estructura consistente. Use CSS para la maquetación donde sea posible
6. Figuras y diagramas: Descríbalos brevemente en la página o use el atributo longdesc
7. *Scripts, applets y plug-ins*: Ofrezca contenido alternativo si las funciones nuevas no son accesibles
8. Marcos: Use el elemento noframes y títulos con sentido
9. Tablas: Facilite la lectura línea a línea, resume
10. Revise su trabajo: Verifique. Use las herramientas, puntos de comprobación y pautas de <http://www.w3.org/TR/WCAG>

18. Usabilidad

10 reglas heurísticas de usabilidad de Jakob Nielsen

Jakob Nielsen “el gurú de la usabilidad” estudió 249 problemas de usabilidad y a partir de ellos diseñó lo que denominó las “**reglas generales**” para identificar los posibles problemas de usabilidad.

1. Visibilidad del estado del sistema

Es de vital importancia que tu usuario esté permanentemente informado sobre lo que está pasando cuando interactúa en tu web, ya que “por detrás” sucede un montón de cosas que no ve, y que le pueden producir incertidumbre en muchas situaciones. Esto parece obvio, pero lo cierto es que todavía se cometen errores graves de esta naturaleza, ¡incluso en webs de importantes empresas!



Debes proporcionar al usuario un feedback constante. Ésa es la función que cumplen, por ejemplo:

- Las barras de proceso que nos indican cómo avanza la subida de un archivo.
- Los mensajes que nos confirman que “el formulario se ha enviado correctamente”.
- Una animación que nos indica que algo está siendo procesado sin incidencias.

Por favor, no dejes nunca a tu usuario pensando “¿qué estará pasando ahora?”.

2. Relación entre el sistema y el mundo real

El sistema tiene que “hablar” al usuario en su mismo lenguaje. Y esto no se refiere sólo al texto, sino también a elementos como:

- Las imágenes.
- El orden en que se hacen las cosas.
- La forma en que se presenta la información.
- Consigue que la relación entre el hombre y la máquina sea natural. Internet no es otra realidad, es parte de la vida de las personas, y eso significa que el usuario rechaza, consciente o inconscientemente, interactuar bajo unos códigos de comunicación y de conducta distintos a los suyos.

“Ser original”, “innovar” o “hacer cosas diferentes” no significa que debas imponer a tus usuarios un enfoque que rompa sus esquemas “porque es más creativo”... salvo que seas un artista y en eso consista tu trabajo, pero recuerda que estamos hablando de interfaces de usuario. Cuanto más claro, mejor. 😊

3. Control y libertad del usuario

No fuerces al usuario a seguir un camino determinado, ya sea por un mal diseño o incluso de forma deliberada, porque a ti te interese. Evita a toda costa los “callejones sin salida”.

El usuario tiene que poder navegar libremente, encontrar con facilidad “salidas” y “rutas alternativas”, y tener todas las facilidades que necesite para “hacer” y “deshacer”.

Si el usuario siente que tiene el control, su experiencia de uso será correcta; si siente lo contrario, no tardará en marcharse.

4. Consistencia y estándares

Tu sitio web debe seguir un estándar consistente en todas sus páginas. Es decir, no puedes hacer por ejemplo que el menú funcione de manera diferente en distintas partes de tu web, ni cambiarlo de ubicación porque te parece que “queda mejor”. No puedes usar distintos diseños para la misma cosa en distintos apartados (forma, color), ni una terminología variable. Si en tu tienda online el carrito se llama “cesta” y tiene un icono verde en la home, no puede llamarse “carro” y tener un icono rojo en la ficha de producto; tampoco puede estar unas veces en una esquina y otras en otra. Sé consistente en las decisiones que tomes.

Es muy frecuente encontrar errores de este tipo en Internet, especialmente en webs veteranas que han ido sufriendo modificaciones con el paso del tiempo.

Es un error relativamente frecuente incurrir en inconsistencias graves por supeditar la funcionalidad a meras cuestiones estéticas. Los que más saben (por ejemplo, Amazon o Google), nunca lo hacen. Saca tus conclusiones.

5. Prevención de errores

No esperes a que el usuario cometa un error que sabes que va a cometer, para mostrarle después un mensaje de aviso. La mayor parte de los errores son previsibles, y debes resolverlos de antemano.

Esto lo puedes hacer:

- Incluyendo información contextual preventiva en el punto problemático.
- Suprimiendo condiciones que son proclives a inducir a error.
- Pidiendo confirmación al usuario.
- Haciendo comprobaciones en tiempo real.
- Dos ejemplos típicos de formulario:
 - Un campo que cambia de color para recordarte que lo has dejado en blanco.
 - Una comprobación en tiempo real que te dice que la segunda contraseña que has puesto no coincide con la primera, antes de dar a “enviar”.

6. Reconocimiento antes que recuerdo

El usuario debe tener siempre toda la información a mano, y no verse obligado a usar su memoria para seguir el hilo de la interacción. Pónselo fácil para que no tenga que estar memorizando cómo volver a la página anterior o cómo encontrar aquel producto que ya ha visto y le interesaba:

- Si el usuario tiene que “recordar” cómo se hace algo en tu web en lugar de “saberlo” intuitivamente, tienes un problema.
- Si el usuario tiene que recordar decisiones que ha tomado previamente porque no le muestras esa información cuando la necesita (por ejemplo en un proceso de compra), su experiencia de uso se empobrece considerablemente, y tendrás muchos más abandonos en la web.

7. Flexibilidad y eficiencia de uso

Algunos usuarios ya conocen tu web y realizan siempre las mismas acciones. ¿Tienes “aceleradores” que les permitan realizar de forma más rápida y directa esas acciones frecuentes?

Un buen interfaz de usuario tiene la flexibilidad necesaria para comportarse “a la medida” de usuarios novatos y usuarios expertos.

Ejemplos:

- Un “atajo” en la home a la página que más visitan tus usuarios, saltando pasos intermedios.
- Mostrar los últimos artículos por los que se ha interesado el usuario en su última visita, o en la visita en curso, ya que probablemente querrá volver a consultarlos.

Debes recordar que estos “aceleradores” no deben confundir a los novatos. Algunas webs ocultan automáticamente estas opciones cuando el visitante es nuevo, lo que sin duda es un síntoma de que sus propietarios “juegan en otra liga” en materia de usabilidad. ¡Bien por ellos! 😊

8. Estética y diseño minimalista

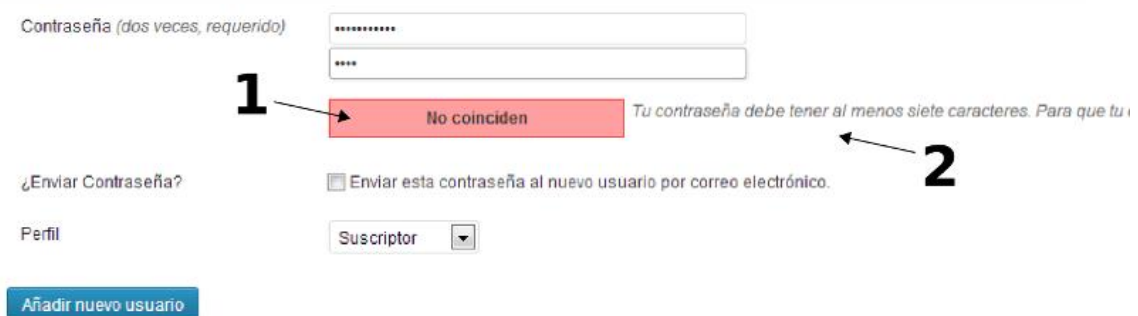
El diálogo que el interfaz mantiene con el usuario no debe contener información irrelevante o de rara utilidad. Dice Nielsen que:

“Cada unidad extra de información en un diálogo compite con las unidades relevantes de información y reduce la visibilidad relativa de éstas.”

Dicho de otra manera, cada palabra de más está oscureciendo las palabras que son realmente importantes.

Mi recomendación es que no limites este principio al texto, sino que lo apliques también al diseño visual. El diseño debe apoyar la función, no “oscurecerla” ni “decorarla” (en mi humilde opinión, insisto, fundamentada por la experiencia).

9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores



The image shows a web form for creating a new user. It has two password input fields. Below the second field, a red error message box says "No coinciden" (They do not match). To the right of this box, a grey message says "Tu contraseña debe tener al menos siete caracteres. Para que tu..." (Your password must have at least seven characters. For your...). A red arrow labeled "1" points to the "No coinciden" message. A black arrow labeled "2" points to the grey message. Below the password fields, there is a checkbox labeled "¿Enviar Contraseña?" and another checkbox labeled "Enviar esta contraseña al nuevo usuario por correo electrónico." Below that is a "Perfil" section with a dropdown menu currently showing "Suscriptor". At the bottom is a blue button labeled "Añadir nuevo usuario".

Los mensajes de error tienen que estar escritos en un lenguaje que el usuario pueda entender -desde luego no con tecnicismos– y deben siempre sugerir una solución o un camino de salida.

Por ejemplo, un ERROR 404 (página no encontrada), que es el más frecuente en cualquier web, debería ser sustituido por una pantalla amistosa donde se dijera algo como: “Vaya, la página que buscas no está aquí. Puede que esté en otro sitio, o que simplemente no exista. Por favor, utiliza este buscador [buscador] o haz clic en el enlace para volver a la página principal [enlace]”.

10. Ayuda y documentación

Por supuesto, es preferible que el sistema pueda usarse de manera intuitiva sin tener que acudir a “unas instrucciones”. No obstante, el usuario necesitará en ocasiones ayuda y documentación.

Es preciso que esta información sea fácil de encontrar, y sobre todo que esté orientada a las tareas concretas que realiza el usuario, antes que a cuestiones teóricas o demasiado genéricas.

Un buen ejemplo es la lista de preguntas frecuentes (normalmente abreviado en inglés como FAQ), que puedes encontrar en la “ayuda” de plataformas sociales como Twitter o Youtube.

La idea es evitar que el usuario tenga que leer contenidos de ayuda que no le interesan antes de llegar al texto que realmente va a resolver su problema.

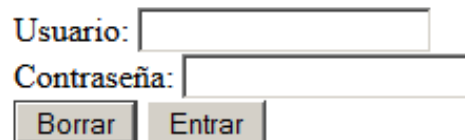
Estas reglas son las claves para poder facilitar el uso de cualquier persona, no solo dentro de los sitios web, sino en cualquier sistema de información.

22. Formularios

Los formularios son el mecanismo que se emplea en HTML para enviar datos a los servidores, y a su vez esperar una respuesta después de enviarlos.

Para crear formularios se emplea el elemento *form*, que es un contenedor para elementos de entrada de datos como campos de texto, menús desplegables de selección, botones de radio (*radiobuttons*), botones de selección (*checkboxes*), etc.

```
<form>
  Usuario: <input type="text" name="usuario"><br>
  Contraseña: <input type="password" name="contrasena">
  <br>
  <input type="reset" value="Borrar">
  <input type="submit" value="Entrar">
</form>
```



Atributos de *form*

(*) Nuevo de HTML5

accept-charset	conjunto caracteres	Codificación de caracteres con que se enviarán los datos
action	URL	dirección a donde enviar los datos
autocomplete (*)	on off	Si se activa o no el autocompletado del formulario
enctype	text/plain multipart/form-data application/x-www-form-urlencoded	Método de codificación de los datos a enviar por el método POST
method	get post	Método de envío de los datos GET → en la URL POST → directamente al servidor
name	<i>nombre</i>	Nombre del formulario
novalidate (*)	novalidate	El formulario no debe ser validado cuando se envíe
target	_blank _self _parent _top	A dónde redirigir la respuesta del servidor a los datos enviados

Elementos de entrada

input	Campo de entrada
textarea	Campo de entrada de texto multilínea
label	Etiqueta del elemento de entrada
fieldset	Grupo de elementos de entrada relacionados
legend	Título del grupo de elementos de entrada relacionados
select	Menú desplegable
optgroup	Grupo de opciones relacionados del menú desplegable
option	Opción de un menú desplegable
button	Botón pulsable
datalist (*)	Lista predefinida de opciones para controles de entrada
keygen (*)	Generador de pares de claves para formularios
output (*)	Salida del resultado de un cálculo

Elemento *input*

El elemento *input* es el elemento básico de entrada de datos.

El elemento *input* no tiene etiqueta de cierre, puede tener atributos y en función de éstos y sus valores tendrá funciones diferentes.

Se aconseja usar el elemento *label* (etiqueta) al elemento *input* cuando se le asocie el título de elemento de entrada.

accept	audio/* video/* image/* tipo MIME	Tipo de archivo aceptado cuando se trata de un campo que acepta archivos (tipo <i>file</i>).
alt	<i>texto alternativo</i>	Texto para los botones de tipo imagen.
autocomplete (*)	on off	Activa/desactiva el autocompletado del campo

autofocus (*)	autofocus	El campo obtiene el foco cuando se carga la página
checked	checked	Campo preseleccionado para los tipos de entrada <i>radiobutton</i> y <i>checkbox</i>
disabled	disabled	El campo está deshabilitado (no editable/seleccionable)
form (*)	form_id	Formulario o formularios a los que pertenece
formaction (*)	URL	Dirección donde procesarán los datos cuando se envíen (botones tipo <i>submit</i> y <i>image</i>)
formenctype (*)	text/plain multipart/form-data application/x-www-form-urlencoded	Tipo de codificación de los datos al enviarlos (botones tipo <i>submit</i> y <i>image</i>)
formmethod (*)	get post	Método por el que se envía los datos (botones <i>submit</i> e <i>image</i>)
formnovalidate (*)	formnovalidate	Los elementos no serán validados cuando se envíen
formtarget (*)	_blank _self _parent _top <i>nombre_marco</i>	Destino de la respuesta del servidor al envío de datos (botones <i>submit</i> y <i>image</i>)
height (*)	<i>pixels</i>	Altura del elemento <i>input</i> de tipo <i>image</i> .
list (*)	<i>datalist_id</i>	Asociación a un elemento <i>datalist</i> que contiene opciones predefinidas para un elemento de entrada.
max (*)	<i>número</i> <i>fecha</i>	Valor máximo de entrada.
maxlength	<i>número</i>	Número máximo de caracteres admitidos.
min (*)	<i>número</i> <i>fecha</i>	Valor mínimo de entrada.

multiple (*)	multiple	Se pueden introducir más de un valor en un campo.
name	<i>nombre</i>	Nombre del elemento de entrada
pattern (*)	<i>expr. regular</i>	Expresión regular para validar el dato introducido
placeholder (*)	<i>texto</i>	Texto breve que describe el valor esperado.
readonly	readonly	El elemento de entrada es de sólo lectura
required (*)	required	Campo obligatorio
size	<i>número</i>	Ancho en caracteres del elemento de entrada
src	URL	El camino completo a la imagen del botón de tipo <i>image</i>
step (*)	<i>número</i>	Número de intervalos para el elemento de entrada
type	button, checkbox, color (*), date (*), datetime (*), datetime-local (*), email (*), file, hidden, image, month (*), number (*), password, radio, range (*), reset, search (*), submit, tel (*), text, time (*), url (*), week (*)	Tipo de elemento de entrada
value	<i>texto</i>	Valor por defecto del campo de entrada
width (*)	<i>pixels</i>	Ancho de la imagen del botón de tipo <i>image</i>

Elemento *textarea*

autofocus (*)	autofocus	Adquiere el foco automáticamente tras la carga de la página
cols	<i>número</i>	Número de columnas del elemento
disabled	disabled	Deshabilita el elemento para su edición
form (*)	<i>id_form</i>	Identificador de formulario al que pertenece el elemento
maxlength (*)	<i>número</i>	Máximo número de caracteres permitidos
name	<i>nombre</i>	Nombre del elemento
placeholder (*)	<i>texto</i>	Texto breve que indica el tipo de dato que se espera
readonly	readonly	El elemento es de sólo lectura
required (*)	required	El elemento es obligatorio
rows	<i>número</i>	Número de filas del elemento
wrap (*)	hard soft	El tipo de ajuste del texto cuando es enviado

Elemento *label*

El elemento *label* define una etiqueta para un elemento *input*.

Este elemento no aporta nada especial en el renderizado de la página. Sin embargo añade una mejora de usabilidad para usuarios de ratón.

La asociación de la etiqueta con el elemento se hace a través del atributo *for*, que ha de tener el mismo valor que el atributo *id* del elemento al que se asocia.

<i>for</i>	<i>id_elemento</i>	Identificador del elemento de entrada al que está asociado
<i>form</i> (*)	<i>id_form</i>	Identificador de formulario a los que pertenece el elemento

Elemento *fieldset*

El elemento *fieldset* se usa para agrupar elementos de entrada relacionados.

Este elemento dibuja un recuadro alrededor de los elementos relacionados.

<i>disabled</i> (*)	<i>disabled</i>	Deshabilita el conjunto de elementos agrupados
<i>form</i> (*)	<i>id_form</i>	Identificador de formulario al que pertenece el elemento
<i>name</i> (*)	<i>nombre</i>	Nombre al que se le puede asignar al conjunto

El elemento *legend* añade un título al elemento *fieldset*.

Datos personales

Nombre	Nombre completo (obligatorio)	*
Apellidos	Apellido completo (obligatorio)	*
Dirección	Dirección completa (obligatorio)	*
Población	--- Seleccione la población ---	*
Código Postal	Código postal (obligatorio)	*
Provincia	--- Escoja la provincia ---	*
Email	Correo electrónico de contacto (obligatorio)	*
Teléfono/Móvil	Teléfono o móvil de contacto (obligatorio)	*

Elemento *select*

Con el elemento *select* se crean listas desplegables o de persiana.

Con el elemento *option* dentro del elemento *select* se añaden los elementos disponibles dentro de la lista.

```
<select name="provincia">
  <option value="">-Escoja la provincia-</option>
  <option value="08">Barcelona</option>
  <option value="17">Girona</option>
  <option value="25">Lleida</option>
  <option value="43">Tarragona</option>
</select>
```

-Escoja la provincia-

Barcelona

Girona

Lleida

Tarragona

autofocus (*)	autofocus	Adquiere el foco automáticamente tras la carga de la página
disabled	disabled	Deshabilita el elemento para su selección
form (*)	<i>id_form</i>	Identificador de formulario al que pertenece el elemento

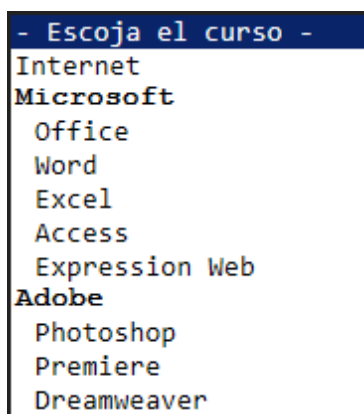
multiple	multiple	Que se puede hacer una selección múltiple
name	<i>nombre</i>	Nombre del elemento
Required (*)	required	Indica que el campo es de selección obligatoria
size	<i>número</i>	Número de opciones visibles en la lista desplegable

Elemento *optgroup*

Con el elemento *optgroup* se pueden agrupar elementos de lista que estén relacionados entre sí.

En caso de listas desplegables extensas permite organizar sus elementos.

```
<option value="">- Escoja el curso -</option>
<option value="WWW">Internet</option>
<optgroup label="Microsoft">
  <option value="MSO">Office</option>
  :
  <option value="EWB">Expression Web</option>
</optgroup>
```



disabled	disabled	Deshabilita el conjunto de opciones de la lista
label	texto	Texto que identifica el grupo de opciones de la lista

Elemento *option*

El elemento *option* define una opción dentro de los elementos de lista.
Puede estar sin atributos, pero normalmente necesita el atributo *value* para indicar el valor a enviar al servidor.

disabled	disabled	Deshabilita el elemento
label	<i>texto</i>	Una breve etiqueta para el elemento
selected	selected	El elemento está seleccionado por defecto
value	<i>valor</i>	Valor al que se asocia a la elemento y que se envía al servidor

Elemento *button*

Con el elemento *button* también se pueden añadir botones a los formularios.
La diferencia con el resto de elementos de formulario es que es clicable.
La principal diferencia entre los botones creados con el elemento *input* y este elemento es que se puede añadir contenido como texto o imágenes.
Conviene especificar siempre el tipo de botón, ya que diferentes navegadores pueden disponer de diferentes tipos de botones por defecto.
Si se usa el elemento *button* en un formulario, diferentes navegadores pueden tener diferentes tipos de botones *submit*, es mejor usar el elemento *input*.

autofocus (*)	autofocus	Adquiere el foco automáticamente tras la carga de la página
disabled	disabled	Deshabilita el elemento para su edición
form (*)	<i>id_form</i>	Identificador de formulario al que pertenece el elemento

formaction (*)	URL	Dirección donde procesarán los datos cuando se envíen (botones tipo <i>submit</i>)
formenctype (*)	text/plain multipart/form-data application/x-www-form-urlencoded	Tipo de codificación de los datos al enviarlos (botones tipo <i>submit</i>)
formmethod (*)	get post	Método por el que se envía los datos (botones <i>submit</i>)
formnovalidate (*)	formnovalidate	Los elementos no serán validados cuando se envíen
formtarget (*)	_blank _self _parent _top <i>nombre_marco</i>	Destino de la respuesta del servidor al envío de datos (botones <i>submit</i>)
name	<i>nombre</i>	Nombre del elemento
type	button reset submit	Tipo de elemento
value	<i>valor</i>	Valor por defecto del elemento

HTML 5

Tag	Info	V	Attributes*
<!-- -->	comment	4 / 5	none
<!DOCTYPE>	document type	4 / 5	none
<a>	hyperlink	4 / 5	href hreflang media ping rel target type
<abbr>	abbreviation	4 / 5	global attributes**
<acronym>	acronym	4	-
<address>	address element	4 / 5	global attributes**
<applet>	applet	4	-
<area>	area inside an image map	4 / 5	alt coords href hreflang media ping rel shape target type
<article>	article	5	global attributes**
<aside>	outside the main flow of the narrative	5	global attributes**
<audio>	sound content	5	autobuffer autoplay controls loop src
	bold text	4 / 5	global attributes**
<base>	base URL for all the page links	4 / 5	href target
<basefont>	Base font for the document	4	-
<bb>	invoked user agent com-	5	type
<bdo>	direction of text display	4 / 5	dir
<big>	big text	4	-
<blockquote>	long quotation	4 / 5	cite
<body>	body element	4 / 5	global attributes**
 	inserts a single line break	4 / 5	global attributes**
<button>	push button	4 / 5	autofocus disabled form formaction formenctype formmethod formnovalidate formtarget name type value
<canvas>	Graphic area	5	height width
<caption>	table caption	4 / 5	global attributes**
<center>	centered text	4	-
<cite>	citation	4 / 5	global attributes**
<code>	computer code text	4 / 5	global attributes**
<col>	attributes for table columns	4 / 5	span
<colgroup>	groups of table columns	4 / 5	span
<command>	command button	5	checked default disabled hidden icon label radiogroup type
<datagrid>	data in a tree, list or tabular	5	disabled
<datalist>	dropdown list	5	global attributes**
<dd>	definition description	4 / 5	global attributes**
	deleted text	4 / 5	cite datetime
<details>	details of an element	5	open
<dialog>	dialog (conversation)	5	global attributes**
<dir>	directory list	4	-
<div>	section in a document	4 / 5	global attributes**
<dfn>	definition term	4 / 5	title
<dl>	definition list	4 / 5	global attributes**
<dt>	definition term	4 / 5	global attributes**
	emphasized text	4 / 5	global attributes**

Tag	Info	V	Attributes*
<embed>	external interactive content or plugin	5	height src type width
<fieldset>	fieldset	4 / 5	disabled form name
<figure>	group of media content, and their caption	5	global attributes**
	text font, size, and color	4	-
<footer>	footer for a section or page	5	global attributes**
<form>	form	4 / 5	action data replace accept accept-charset enctype method target
<frame>	sub window	4	-
<frameset>	set of frames	4	-
<h1> to <h6>	header 1 to header 6	4 / 5	global attributes**
<head>	information about the document	4 / 5	none
<header>	header for a section or page	5	global attributes**
<hgroup>	heading section	5	global attributes**
<hr>	horizontal rule	4 / 5	global attributes**
<html>	html document	4 / 5	manifest
<i>	italic text	4 / 5	global attributes**
<iframe>	inline sub window (frame)	4 / 5	src name sandbox seamless width height
	image	4 / 5	alt src height ismap usemap width
<input>	input field	4 / 5	accept alt autocomplete autofocus checked disabled form formaction formenctype formmethod formnovalidate formtarget height list max maxlength min multiple name pattern placeholder readonly required size src step type value width
<ins>	inserted text	4 / 5	cite datetime
<isindex>	single-line input field	4	-
<kbd>	keyboard text	4 / 5	global attributes**
<label>	label for a form control	4 / 5	for
<legend>	fieldset title	4 / 5	global attributes**
	list item	4 / 5	value
<link>	resource reference	4 / 5	href rel media hreflang type sizes
<mark>	marked text	5	global attributes**
<map>	image map	4 / 5	id
<menu>	menu list	4 / 5	label type
<meta>	meta information	4 / 5	charset content http-equiv name
<meter>	measurement within a predefined range	5	high low max min optimum value
<nav>	navigation links	5	global attributes**
<noframes>	noiframe section	4	-
<noscript>	noscript section	4 / 5	none
<object>	embedded object	4 / 5	data height type usemap width object

* **Attributes:** Lists attributes specific to that tag. Deprecated (html4 only) attributes are not listed

** **Global Attributes:** class | contenteditable | contextmenu | dir | draggable | id | irrelevant | lang | ref | registrationmark | tabindex | template | title

Tag	Info	V	Attributes*
	ordered list	4 / 5	start reversed
<optgroup>	option group	4 / 5	disabled label
<option>	option in a drop-down list	4 / 5	disabled label selected value
<output>	some types of output	5	form
<p>	paragraph	4 / 5	global attributes**
<param>	parameter for an object	4 / 5	name value
<pre>	preformatted text	4 / 5	global attributes**
<progress>	progress of a task of any kind	5	max value
<q>	short quotation	4 / 5	cite
<ruby>	ruby annotations	5	global attributes**
<rp>	provide parentheses around a ruby text	5	global attributes**
<rt>	ruby text component	5	global attributes**
<s>	strikethrough text	4	-
<samp>	sample computer code	4 / 5	global attributes**
<script>	script	4 / 5	async type defer src charset
<section>	section	5	cite
<select>	selectable list	4 / 5	autofocus data disabled form multiple name
<small>	small text	4 / 5	global attributes**
<source>	media resources	5	media src type
	inline section	4 / 5	global attributes**
<strike>	strikethrough text	4	-
	strong text	4 / 5	global attributes**
<style>	style definition	4 / 5	media type scoped
<sub>	subscripted text	4 / 5	global attributes**
<sup>	superscripted text	4 / 5	global attributes**
<table>	table	4 / 5	global attributes**
<tbody>	table body	4 / 5	global attributes**
<td>	table cell	4 / 5	colspan rowspan headers
<textarea>	text area	4 / 5	autofocus cols disabled form name readonly required rows maxlength placeholder wrap
<tfoot>	table footer	4 / 5	global attributes**
<th>	table header	4 / 5	colspan rowspan scope
<thead>	table header	4 / 5	global attributes**
<time>	date/time	5	datetime
<title>	document title	4 / 5	none
<tr>	table row	4 / 5	global attributes**
<tt>	teletype text	4	-
<u>	underlined text	4	-
	unordered list	4 / 5	global attributes**
<var>	variable	4 / 5	global attributes**
<video>	video	5	src poster autobuffer autoplay loop controls width height
<xmp>	preformatted text	4	-

V = Which version of HTML is this tag valid for

HTML 5 - extended

Tag	Info	Attributes
<!-- -->	comment: comments are displayed in code only. Tag contents are not rendered in the browser	none
<!DOCTYPE>	document type: defines which specification the document follows	none
<a>	anchor: used to provide a link to another web resource	href: destination resource of the hyperlink hreflang: gives the language of the linked resource media: describes for which media the target document was designed ping: gives the URLs of the resources that are interested in being notified if the user follows the hyperlink rel: relationship between the document containing the hyperlink and the destination resource [<i>alternate</i> <i>archives</i> <i>author</i> <i>bookmark</i> <i>contact</i> <i>external</i> <i>feed</i> <i>first</i> <i>help</i> <i>icon</i> <i>index</i> <i>last</i> <i>license</i> <i>next</i> <i>nofollow</i> <i>noreferrer</i> <i>pingback</i> <i>prefetch</i> <i>prev</i> <i>search</i> <i>stylesheet</i> <i>sidebar</i> <i>tag</i> <i>up</i>] target: gives the name of the browsing context that will be used [<i>_blank</i> <i>_parent</i> <i>_self</i> <i>_top</i>] type: gives the MIME type of the linked resource
<abbr>	abbreviation: an abbreviation or acronym, optionally with its expansion	<i>global attributes</i> **
<address>	address element: represents the contact information for its nearest article or body element ancestor	<i>global attributes</i> **
<area>	area: either a hyperlink with some text and a corresponding area on an image map, or a dead area on an image map	alt: alternate text for the area coords: coordinates for the clickable area href: destination resource of the hyperlink hreflang: gives the language of the linked resource media: describes for which media the target document was designed ping: gives the URLs of the resources that are interested in being notified if the user follows the hyperlink rel: relationship between the document containing the hyperlink and the destination resource [<i>alternate</i> <i>archives</i> <i>author</i> <i>bookmark</i> <i>contact</i> <i>external</i> <i>feed</i> <i>first</i> <i>help</i> <i>icon</i> <i>index</i> <i>last</i> <i>license</i> <i>next</i> <i>nofollow</i> <i>noreferrer</i> <i>pingback</i> <i>prefetch</i> <i>prev</i> <i>search</i> <i>stylesheet</i> <i>sidebar</i> <i>tag</i> <i>up</i>] shape: defines the shape of the area [<i>default</i> <i>rect</i> <i>rectangle</i> <i>circ</i> <i>circle</i> <i>poly</i> <i>polygon</i>] target: gives the name of the browsing context that will be used [<i>_blank</i> <i>_parent</i> <i>_self</i> <i>_top</i>] type: gives the MIME type of the linked resource
<article>	article element: a section of a page that consists of a composition that forms an independent part of a document, page, or site	<i>global attributes</i> **
<aside>	aside element: a section of a page that consists of content that is tangentially related to the content around the aside element, and which could be considered separate from that content	<i>global attributes</i> **
<audio>	sound content: represents a sound or audio stream	autobuffer: determines if the audio will be buffered [<i>autobuffer</i>] autoplay: determine if the audio will automatically play [<i>autoplay</i>] controls: indicates that the author has not provided a scripted controller and would like the user agent to provide its own set of controls [<i>controls</i>] loop: sets whether the audio will start once the end is reached [<i>loop</i>] src: URL of the audio to play
	bold text: creates text that will be made bold	<i>global attributes</i> **
<base>	base element: base URL for all the page links	href: URL to use as the base URL for links in the page target: sets the base target for links in the page [<i>_blank</i> <i>_parent</i> <i>_self</i> <i>_top</i>]
<bb>	browser button: a user agent command that the user can invoke	type: indicates the kind of command [<i>makeapp</i>]
<bdo>	bdo element: represents explicit text directionality formatting control for its children	dir: direction override [<i>ltr</i> <i>rtl</i>]
<blockquote>	block quote element: a section that is quoted from another source	cite: URL of the origin of the quote
<body>	body element: main content of the document	<i>global attributes</i> **
 	break: inserts a single line break	<i>global attributes</i> **

Tag	Info	Attributes
<button>	button: a button page element	autofocus: indicate that a control is to be focused as soon as the page is loaded [<i>autofocus</i>] disabled: prevents the button from being pressed [<i>disabled</i>] form: used to explicitly associate the button element with its form owner formaction: URL that specifies a form processing agent formenctype: specifies the content type used to submit the form to the server [<i>application/x-www-form-urlencoded</i> <i>multipart/form-data</i> <i>text/plain</i>] formmethod: which HTTP method will be used to submit the forms data [<i>get</i> <i>post</i> <i>put</i> <i>delete</i>] formnovalidate: indicate whether the form is to be validated during submission [<i>formnovalidate</i>] formtarget: gives the target when the form is submitted [<i>_blank</i> <i>_parent</i> <i>_self</i> <i>_top</i>] name: elements name type: controls the behavior of the button when it is activated [<i>submit</i> <i>reset</i> <i>button</i>] value: gives the element's value for the purposes of form submission
<canvas>	canvas element: a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, or other visual images on the fly	height: height of the canvas in pixels - default is 150 width: width of the canvas in pixels - default is 300
<caption>	table caption: the title of the table that is its parent, if it has a parent and that is a table element.	<i>global attributes</i> **
<cite>	citation: represents the title of a work	<i>global attributes</i> **
<code>	computer code text: represents a fragment of computer code. This could be an XML element name, a filename, a computer program, or any other string that a computer would recognize.	<i>global attributes</i> **
<col>	column: defines the attribute values for one or more columns in a table. Used inside of a table or colgroup	span: number of columns the tag should span
<colgroup>	column group: a group of one or more columns in the table that is its parent, if it has a parent and that is a table element	span: number of columns the tag should span
<command>	command button: a command that the user can invoke (like radio button or checkbox)	type: Specifies the type of command [<i>checkbox</i> <i>command</i> <i>radio</i>] label: gives the name of the command, as shown to the user icon: a URL to a picture that represents the command disabled: prevents the command from being executed [<i>disabled</i>] checked: Determines if the command is checked by default [<i>checked</i>] radiogroup: gives the name of the group of commands that will be toggled when the command itself is toggled title: gives a hint describing the command, which might be shown to the user to help them
<datagrid>	datagrid element: an interactive representation of tree, list,	disabled: defines whether the list is selectable [<i>disabled</i>]
<datalist>	dropdown list: a set of option elements that represent predefined options for other controls	<i>global attributes</i> **
<dd>	definition description: description, definition, or value, part of a term-description group in a description list (dl element), and the discourse, or quote, part in a conversation (dialog element)	<i>global attributes</i> **
	deleted text: represents a removal from the document	cite: a URL used to specify the address of a document that explains the change datetime: used to specify the time and date of the change
<details>	details element: represents additional information or controls which the user can obtain on demand	open: indicates whether the details are to be shown to the user [<i>open</i>]
<dialog>	dialog element: represents a conversation, meeting minutes, a chat transcript, a dialog in a screenplay, an instant message log, or some other construct in which different players take turns	<i>global attributes</i> **
<div>	document block: creates a block level element with no special meaning	<i>global attributes</i> **

attributes - values in [] are the accepted values

HTML 5 - extended

Tag	Info	Attributes
<dfn>	definition term: the defining instance of a term	title: the exact value of the term being defined
<dl>	definition list: an association list consisting of zero or more name-value groups (a description list). Each group must consist of one or more names (dt elements) followed by one or more values (dd elements)	<i>global attributes**</i>
<dt>	definition term: the term, or name, part of a term-description group in a description list (dl element), and the talker, or speaker, part of a talker-discourse pair in a conversation (dialog element)	<i>global attributes**</i>
	emphasized text: represents stress emphasis of its contents.	<i>global attributes**</i>
<embed>	embed element: an external (typically non-HTML) application or interactive content	src: URL of the resource being embedded type: gives the MIME type of the plugin to instantiate height: height of the embedded content in pixels width: width of the embedded content in pixels
<fieldset>	fieldset element: a set of form controls grouped under a common name	disabled: controls whether all the form control descendants are disabled [<i>disabled</i>] form: used to explicitly associate the fieldset element with its form owner name: gives the name of the form control
<figure>	figure element: some flow content, optionally with a caption, that is self-contained and is typically referenced as a single unit from the main flow of the document	<i>global attributes**</i>
<footer>	footer element: represents a footer for the section it applies to	<i>global attributes**</i>
<form>	form element: represents a collection of form-associated elements, some of which can represent editable values that can be submitted to a server for processing	accept-charset: gives the character encodings that are to be used for the submission action: URL that specifies a form processing agent autocomplete: determines if form elements will have their autocomplete turned on or off by default [<i>on</i> <i>off</i>] enctype: specifies the content type used to submit the form to the server [<i>application/x-www-form-urlencoded</i> <i>multipart/form-data</i> <i>text/plain</i>] method: which HTTP method will be used to submit the forms data [<i>get</i> <i>post</i> <i>put</i> <i>delete</i>] name: elements name novalidate: indicate whether the form is to be validated during submission [<i>novalidate</i>] target: gives the target when the form is submitted [<i>_blank</i> <i>_parent</i> <i>_self</i> <i>_top</i>]
<h1> to <h6>	headers (1-6): represent headings for their sections. elements have a rank given by the number in their name	<i>global attributes**</i>
<head>	head element: contains information about the document	none
<header>	header element: represents a group of introductory or navigational aids	<i>global attributes**</i>
<hgroup>	heading group: used to group a set of h1-h6 elements when the heading has multiple levels, such as subheadings, alternative titles, or taglines	<i>global attributes**</i>
<hr>	horizontal rule: creates a horizontal rule (line)	<i>global attributes**</i>
<html>	html document: root of an HTML document.	manifest: a URL to the address of the document's application cache manifest
<i>	italic text: indicates the text is to be rendered with emphasis	<i>global attributes**</i>
<iframe>	inline frame: represents a nested browsing window	src: URL of a page that the nested browsing context is to contain name: elements name sandbox: enables a set of extra restrictions on any content hosted by the iframe [<i>allow-same-origin</i> <i>allow-forms</i> <i>allow-scripts</i>] seamless: indicates whether the iframe element's browsing context is to be rendered in a manner that makes it appear to be part of the containing document [<i>seamless</i>] height: height of the frame in pixels width: width of the frame in pixels
	image: represents an image	alt: text to display if the image can not src: a URL to the image file usemap: name of the map to use for the image ismap: provides access to a server-side image map height: height of the image in pixels width: width of the image in pixels

Tag	Info	Attributes
<input>	input field: a typed data field, usually with a form control to allow the user to edit the data	Attributes are dependant upon input type accept: specified to provide user agents with a hint of what file types the server will be able to accept alt: provides the textual label for the alternative button for users and user agents who cannot use the image autocomplete: determines if the data is considered sensitive and if autocomplete will be used [<i>on</i> <i>off</i> <i>default</i>] autofocus: determines if the input will get focus when a page loads [<i>autofocus</i>] checked: determines if the input will be checked by default [<i>checked</i>] disabled: prevents the input from being pressed [<i>disabled</i>] form: used to explicitly associate the button element with its form owner formaction: URL that specifies a form processing agent formenctype: specifies the content type used to submit the form to the server [<i>application/x-www-form-urlencoded</i> <i>multipart/form-data</i> <i>text/plain</i>] formmethod: which HTTP method will be used to submit the forms data [<i>get</i> <i>post</i> <i>put</i> <i>delete</i>] formnovalidate: indicate whether the form is to be validated during submission [<i>formnovalidate</i>] formtarget: gives the target when the form is submitted [<i>_blank</i> <i>_parent</i> <i>_self</i> <i>_top</i>] height: height of the input in pixels list: used to identify an element that lists predefined options suggested to the user max and max: indicate the allowed range of values for the element maxlength: controls the maxlength of the input to a control multiple: indicates whether the user is to be allowed to specify more than one value [<i>multiple</i>] name: elements name pattern: specifies a regular expression against which the control's value is to be checked placeholder: a short hint intended to aid the user with data entry readonly: determines if the control is readonly [<i>readonly</i>] required: determines if the input is required before the form submits [<i>required</i>] size: gives the number of characters that, in a visual rendering, the user agent is to allow the user to see while editing src: URL to an image (image button) step: indicates the granularity that is expected (and required) of the value type: controls the data type (and associated control) of the element [<i>hidden</i> <i>text</i> <i>search</i> <i>tel</i> <i>url</i> <i>email</i> <i>password</i> <i>datetime</i> <i>date</i> <i>month</i> <i>week</i> <i>time</i> <i>datetime-local</i> <i>number</i> <i>range</i> <i>color</i> <i>checkbox</i> <i>radio</i> <i>file</i> <i>submit</i> <i>image</i> <i>reset</i> <i>button</i>] value: sets the element's value width: width of the input in pixels
<ins>	inserted text: an addition to the document	cite: a URL used to specify the address of a document that explains the change datetime: used to specify the time and date of the change
<kbd>	keyboard text: user input (typically keyboard input, although it may also be used to represent other input, such as voice commands)	<i>global attributes**</i>
<label>	label: caption in a user interface	for: specified to indicate a form control with which the caption is to be associated
<legend>	fieldset title: sets the title of a fieldset element	<i>global attributes**</i>
	list item: represents a list item of an Ordered (OL) or Unordered list (UL)	value: used in an Ordered List (OL) to set the display value
<link>	resource link: allows authors to link their document to other resources	href: destination resource of the hyperlink rel: relationship between the document containing the hyperlink and the destination resource [<i>alternate</i> <i>archives</i> <i>author</i> <i>bookmark</i> <i>contact</i> <i>external</i> <i>feed</i> <i>first</i> <i>help</i> <i>icon</i> <i>index</i> <i>last</i> <i>license</i> <i>next</i> <i>nofollow</i> <i>noreferrer</i> <i>pingback</i> <i>prefetch</i> <i>prev</i> <i>search</i> <i>stylesheet</i> <i>sidebar</i> <i>tag</i> <i>up</i>] media: describes for which media the target document was designed hreflang: gives the language of the linked resource type: gives the MIME type of the linked resource sizes: gives the sizes of icons for visual media.
<mark>	marked text: a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.	<i>global attributes**</i>
<map>	image map: in conjunction with any area element descendants, defines an image map	name: gives the map a name so that it can be referenced
<menu>	menu list: a list of commands	label: sets a visible label for the menu type: indicates the kind of menu being declared [<i>context</i> <i>toolbar</i> <i>list</i>]

attributes - values in [] are the accepted values

HTML 5 - extended

Tag	Info	Attributes
<meta>	meta information: sets meta information for the page (like title, description)	charset: specifies the character encoding used by the document content: sets the value of the document metadata http-equiv: sets a pragma directive [<i>content-language</i> <i>content-type</i> <i>default-style</i> <i>refresh</i>] name: set the name of the meta information
<meter>	meter element: scalar measurement within a known range, or a fractional value	high: specifies the range that is considered to be the "high" part low: specifies the range that is considered to be the "low" part min: specifies the lower boundary max: specifies the upper boundary optimum: specifies the range that is considered to be the "optimum" part value: current location within the range
<nav>	navigation element: section of a page that links to other pages or to parts within the page; a section with navigation links	<i>global attributes**</i>
<noscript>	noscript section: represents nothing if scripting is enabled, and represents its children if scripting is disabled	<i>global attributes**</i>
<object>	embedded object: an external resource, which, depending on the type of the resource, will either be treated as an image, as a nested browsing context, or as an external resource to be processed by a plugin	data: specifies the address of the resource name: valid browsing context name usemap: name of the map to use for the image form: form to associate the object with type: gives the MIME type of the plugin to instantiate height: height of the embedded content in pixels width: width of the embedded content in pixels
	ordered list: list of items, where the items have been intentionally ordered	start: the ordinal value of the first list item reversed: indicates that the list is a descending list [<i>reversed</i>]
<optgroup>	option group: a group of option elements with a common label	disabled: disables all options in the group [<i>disabled</i>] label: gives the name of the group, as shown to the user
<option>	option element: an option in a select element or as part of a list of suggestions in a datalist element	disabled: prevent any clicks on an option item [<i>disabled</i>] label: provides a label for element selected: determines if the option is selected by default [<i>selected</i>] value: provides a value for element
<output>	output element: the result of a calculation	form: used to explicitly associate the output element with its form owner for: allows an explicit relationship to be made between the result of a calculation and the elements that represent the values that went into the calculation or that influenced the calculation
<p>	paragraph: creates a paragraph	<i>global attributes**</i>
<param>	parameter element: defines parameters for plugins invoked by object elements. It does not represent anything on its own	name: gives the name of the parameter. value: gives the value of the parameter.
<pre>	preformatted text: represents a block of preformatted text	<i>global attributes**</i>
<progress>	progress element: represents the completion progress of a task.	max: specifies how much work the task requires in total value: specifies how much of the task has been completed
<q>	short quotation: phrasing content quoted from another source	cite: a URL of a page where the quote was taken from
<ruby>	ruby annotations: allows one or more spans of phrasing content to be marked with ruby annotations	<i>global attributes**</i>
<rp>	ruby text parentheses: can be used to provide parentheses around a ruby text component of a ruby annotation	<i>global attributes**</i>
<rt>	ruby text component: marks the ruby text component of a ruby annotation	<i>global attributes**</i>
<samp>	sample: sample output from a program or computing system.	<i>global attributes**</i>
<script>	script element: allows authors to include dynamic script and data blocks in their documents	async: the script will be executed asynchronously, as soon as it is available [<i>async</i>] type: gives the MIME type of the script or format of the data defer: the script is executed when the page has finished parsing [<i>defer</i>] src: gives the address of the external script resource to use charset: specifies the character encoding of the external script resource
<section>	section element: represents a generic document or application section	cite: a URL of a page where the section was taken from

Tag	Info	Attributes
<select>	selectable list: a control for selecting amongst a set of options	autofocus: determines if the controls gets focus when the page loads [<i>autofocus</i>] disabled: prevent the selection of an item [<i>disabled</i>] form: form to associate the select with multiple: allows the selection of multiple items [<i>multiple</i>] size: gives the number of options to show to the user
<small>	small text: small print or other side comments	<i>global attributes**</i>
<source>	source element: allows authors to specify multiple media resources for media elements.	media: gives the intended media type of the media resource src: URL of the media resource type: gives the MIME type of the source
	span: used for an inline element	<i>global attributes**</i>
	strong: represents strong importance for its contents	<i>global attributes**</i>
<style>	style definition: allows authors to embed style information in their documents	media: says which media the styles apply to type: gives the MIME type (default: text/css) scoped: indicates that the styles are intended just for the subtree rooted at the style element's parent element [<i>scoped</i>]
<sub>	subscript: subscript text	<i>global attributes**</i>
<sup>	superscript: superscript text	<i>global attributes**</i>
<table>	table element: represents data with more than one dimension, in the form of a table	<i>global attributes**</i>
<tbody>	table body: represents a block of rows that consist of a body of data for a table	<i>global attributes**</i>
<td>	table cell: represents a data cell in a table	colspan: sets how many columns a cell will span rowspan: sets how many rows a cell will span headers: space separated list of ids corresponding to the th ids and give header information for the cell
<textarea>	text area: a multiline plain text edit control for the element's raw value	autofocus: determines if the textarea gets focus when the page loads [<i>autofocus</i>] cols: specifies the expected maximum number of characters per line disabled: prevents entry of text [<i>disabled</i>] form: form to associate the textarea with readonly: control whether the text can be edited by the user or not [<i>readonly</i>] required: will be required to enter a value before submitting the form [<i>required</i>] rows: specifies the number of lines to show maxlength: controls the maximum amount of characters which can be entered placeholder: a hint intended to aid the user with data entry wrap: defines how text is wrapped [<i>soft</i> <i>hard</i>]
<tfoot>	table footer: the block of rows that consist of the column summaries (footers) for a table	<i>global attributes**</i>
<th>	table header: represents a header cell in a table	colspan: determines how many columns a cell will span rowspan: determines how many rows a cell will span headers: space separated list of ids corresponding to the th ids and give header information for the cell scope: determines where the cell provides its header information [<i>col</i> <i>colgroup</i> <i>row</i> <i>rowgroup</i>]
<thead>	table header: the block of rows that consist of the column labels (headers) for a table	<i>global attributes**</i>
<time>	date/time: a precise date and/or a time in the Gregorian calendar	datetime: date/time using the Gregorian calendar
<title>	title element: sets the title of the document	none
<tr>	table row: a row of cells in a table	<i>global attributes**</i>
	unordered list: a list of items, where the order of the items is not important	<i>global attributes**</i>
<var>	variable: this could be an actual variable in a mathematical expression or programming context	<i>global attributes**</i>
<video>	video element: a video or movie	poster: URL of an image file that the user agent can show while no video data is available autoplay: determines if the audio will be buffered [<i>autoplay</i>] autoplay: determine if the audio will automatically play [<i>autoplay</i>] controls: indicates that the author has not provided a scripted controller and would like the user agent to provide its own set of controls [<i>controls</i>] loop: sets whether the audio will start once the end is reached [<i>loop</i>] src: URL of the audio to play width: width of the video in pixels height: height of the video in pixels

attributes - values in [] are the accepted values

<http://www.veign.com>

Part Number: QRG0009

©2009 Veign, All Rights Reserved

CSS. Conceptos básicos

Propiedades de fuente	
font-family	Define las tipografías y su orden de preferencia. {font-family: Tahoma, "Times New Roman", arial;}
font-style	Da formato a la tipografía (<i>cursiva, oblicua y normal</i>). Valores: normal; italic; oblique; {font-style: italic;}
font-variant	Da formato de versalitas al texto. Valores: small-caps; normal; {font-variant: normal;}
font-weight	Grosor de la letra. Valores: bold; normal; lighter; bolder; y valores numéricos entre 100-900. {font-weight: bold;}
font-size	Tamaño de la letra en (pt, in, cm, px, em, ex y porcentaje). {font-size: 10px;}
font	Permite definir simultáneamente más de una propiedad de la fuente. El orden de los valores es importante: (font-style font-variant font-weight font-size/line-height font-family); {font: italic bold 10px/12pt Tahoma, "Times New Roman", arial;}

Propiedades del texto	
word-spacing	Define la cantidad de espacios en blanco entre palabras. P {word-spacing: 0.5em;}
letter-spacing	Define la cantidad de espacio en blanco entre letras. {letter-spacing: 0.5em;}
text-decoration	Define decoraciones de la fuente como los subrayados. Valores: none; blink; line-through; overline; underline; {text-decoration: underline;}
vertical-align	Alineación vertical respecto a su elemento padre. Valores: baseline; bottom; middle; sub; super; text-bottom; text-top; top; {vertical-align: baseline;}
text-transform	Define mayúsculas/minúsculas en las letras de un elemento. Valores: capitalize; uppercase; lowercase; none; {text-transform: capitalize;}
text-align	Alineación horizontal del texto respecto un elemento. {text-align: justify;}
text-indent	Tabulación de la primera línea en un elemento. {text-indent: 10px;}
line-height	Interlineado del texto. {line-height: 150%;}

Propiedades de clasificación	
display	Usado para sobrescribir el formato por defecto de los elementos HTML. Define el tipo de caja. Valores: none; block; compact(-); inline; inline-block(+); inline-table; list-item; marker(-); run-in; table; table-caption; table-cell; table-column; table-column-group; table-footer-group; table-header-group; table-row; table-row-group; {display: none;}
white-space	Define como son tratados los espacios en blanco y saltos de línea. Valores: normal; nowrap; pre; pre-line(+); pre-wrap; {white-space: nowrap;}
list-style-type	Define el estilo de punto o numeración a usar en una lista. Valores: none; circle; disc; square; decimal; decimal-leading-zero; lower-alpha; upper-alpha; lower-greek; lower-latin; upper-latin; lower-roman; upper-roman; armenian; georgian; hebrew(-); cjk-ideographic(-); hiragana(-); katakana(-); hiragana-iroha(-); katakana-iroha(-); {list-style-type: square;}
list-style-image	Imagen a ser usada como punto en una lista. {list-style-image: url(encabezado.gif);}
list-style-position	Posición del punto o número en una lista con respecto al contenido del elemento de la lista. Valores: inside; outside; {list-style-position: inside;}
list-style	Propiedades compuesta que engloba el resto de propiedades. {list-style: square url(encabezado.gif) inside;}

Propiedades de posicionamiento	
clip	Área de recorte de un elemento para ser dibujada como transparente. {clip: rect (5pt auto 10px auto) }
left	Especifica la posición izquierda de un elemento. {left: 30px;}
top	Especifica la posición superior de un elemento. {top: -10px;}
overflow	Especifica como el contenido que desborda de la caja debe ser manejado. Valores: visible; hidden; scroll; auto; {overflow: scroll;}
position	Especifica como puede ser posionado el elemento. Valores: absolute; fixed; relative; static; {position: relative;}
visibility	Especifica si el elemento es visible. Valores: visible; hidden; collapse; {visibility: hidden;}
z-index	Especifica si el elemento es visto por encima de elementos en capas superiores. Valores: auto; o un número entero. {z-index: 2;}

Color y propiedades de fondo	
color	Color de un elemento en cuestión. {color: #ff3300;}
background-color	Color de fondo de un elemento. {background-color: #BBBBBB; }
background-image	Asignación de imagen de fondo. {background-image: url(img/fondo.gif);}
background-repeat	Tipo de repetición para una imagen de fondo. Valores: repeat; (mosaic), no-repeat; repeat-x; (horizontal), repeat-y; (vertical). {background-repeat: no-repeat;}
background-attachment	La imagen de fondo sigue el movimiento del scroll o no. Valores: scroll; y fixed; {background-attachment: fixed;}
background-position	Punto de comienzo de un fondo, ya sea imagen o color. Valores: Position (x y) o (x%; y%); top; center; bottom; left; right; {background-position: top center;}
background	Permite definir simultáneamente varias propiedades. Los valores pueden ser escritos en cualquier orden. {background: #BBBBBB url(img/fondo.gif) no-repeat fixed center;}

Propiedades de cuadros	
margin-top margin-right margin-bottom margin-left	Controla los márgenes de un elemento (<i>superior, derecho, inferior y izquierdo</i>). {margin-top: 30px;} {margin-right: 30px;} {margin-bottom: 30px;} {margin-left: 30px;}
margin	Tamaño total de los márgenes de un elemento. (Los valores empiezan por arriba y van en dirección horaria) {margin: 30px;} {margin: 10px 15px 10px 15px;}
padding-top padding-right padding-bottom padding-left	Controla los márgenes interiores de un elemento. (<i>superior, derecho, inferior y izquierdo</i>) {padding-top: 10px;} {padding-right: 10px;} {padding-bottom: 10px;} {padding-left: 10px;}
padding	Tamaño total de los márgenes interiores de un elemento. {padding: 10px;} {padding: 10px 15px 10px 15px;}
border-top-width border-right-width border-bottom-width border-left-width	Controla el ancho del borde de un elemento. (<i>superior, derecho, inferior y izquierdo</i>). {border-top-width: 1px;} {border-right-width: 1px;} {border-bottom-width: 1px;} {border-left-width: 1px;}
border-width	Ancho total de todos los bordes de un elemento. {border-width: 1px;}
border-color	Color del borde de un elemento. Se pueden definir independientemente los 4 lados, 2 a 2 (<i>superior y inferior / derecho y izquierdo</i>) o individualmente. {border-color: #ff3300 #ffffff;}
border-style	Estilo del borde de un elemento. Valores: dashed; dotted; double; groove; inset; outset; ridge; solid; none; {border-style: solid;}
border-top border-right border-bottom border-left	Propiedades que definen la anchura, color y estilo del borde de un elemento. (<i>superior, derecho, inferior y izquierdo</i>). {border-top: 1px solid #ff3300;} {border-right: 1px solid #ff3300;} {border-bottom: 1px solid #ff3300;} {border-left: 1px solid #ff3300;}
border	Propiedades que definen la anchura, color y estilo del total de bordes de un elemento. {border: 1px solid #ff3300;}
width	Establece la anchura de un elemento. {width: 80%;}
height	Establece la altura de un elemento. {height: 100%;}
float	Establece el modo flotante del elemento. Valores: left; right; none; {float: left;}
clear	Especifica si el elemento no puede tener elementos flotantes alrededor. Valores: both; left; right; none; {clear: both;}

Tabla de equivalencias							
Pts	Px	Em	%	Pts	Px	Em	%
6pt	8px	0.5em	105%	13pt	17px	1.05em	105%
7pt	9px	0.55em	112.5%	13.5pt	18px	1.125em	112.5%
7.5pt	10px	0.625em	120%	14pt	19px	1.2em	120%
8pt	11px	0.7em	125%	14.5pt	20px	1.25em	125%
9pt	12px	0.75em	130%	15pt	21px	1.3em	130%
10pt	13px	0.8em	140%	16pt	22px	1.4em	140%
10.5pt	14px	0.875em	145%	17pt	23px	1.45em	145%
11pt	15px	0.95em	150%	18pt	24px	1.5em	150%
12pt	16px	1em	160%	20pt	26px	1.6em	160%

Cascading Style Sheets (CSS 3)

BACKGROUND		BORDER		BOX MODEL	
background	<i>background-image</i> <i>background-position</i> <i>background-size</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-origin</i> <i>background-clip</i> <i>background-color</i>	border-top	<i>border-top-width</i> <i>border-style</i> <i>border-color</i>	float	left right none
background-attachment	scroll fixed	border-top-color	<i>border-color</i>	height	auto <i>length</i> %
background-break	bounding-box each-box continuous	border-top-style	<i>border-style</i>	max-height	none <i>length</i> %
background-clip	<i>length</i> % border-box padding-box content-box no-clip	border-top-width	thin medium thick <i>length</i>	max-width	none <i>length</i> %
background-color	<i>color</i> transparent	border-width	thin medium thick <i>length</i>	min-height	none inherit <i>length</i> %
background-image	<i>url</i> none	border-radius	<i>border-top-right-radius</i> <i>border-bottom-right-radius</i> <i>border-bottom-left-radius</i> <i>border-top-left-radius</i>	min-width	none inherit <i>length</i> %
background-origin	border-box padding-box content-box	border-top-right-radius	<i>length</i>	width	auto % <i>length</i>
background-position	top left top center top right center left center center center right bottom left bottom center bottom right <i>x</i> -% <i>y</i> -% <i>x-pos</i> <i>y-pos</i>	border-bottom-right-radius	<i>length</i>	margin	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>
background-repeat	repeat repeat-x repeat-y no-repeat	border-bottom-left-radius	<i>length</i>	margin-bottom	auto <i>length</i> %
background-size	<i>length</i> % auto cover contain	border-top-left-radius	<i>length</i>	margin-left	auto <i>length</i> %
BORDER		FONT		margin-right	auto <i>length</i> %
border	<i>border-width</i> <i>border-style</i> <i>border-color</i>	font	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu message-box small-caption status-bar	margin-top	auto <i>length</i> %
border-break	<i>border-width</i> <i>border-style</i> <i>color</i> close	font-family	<i>family-name</i> <i>generic-family</i> inherit	padding	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>
border-bottom	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>	font-size	xx-small x-small small medium large x-large xx-large smaller larger inherit <i>length</i> %	padding-bottom	<i>length</i> %
border-bottom-color	<i>border-color</i>	font-size-adjust	none inherit <i>number</i>	padding-left	<i>length</i> %
border-bottom-style	<i>border-style</i>	font-stretch	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded inherit	padding-right	<i>length</i> %
border-bottom-width	thin medium thick <i>length</i>	font-style	normal italic oblique inherit	padding-top	<i>length</i> %
border-collapse	collapse separate	font-variant	normal small-caps inherit	marquee-direction	forward reverse
border-color	<i>color</i>	font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	marquee-loop	infinite <i>number</i>
border-image	<i>image</i> [<i>number</i> / % <i>border-width</i> stretch repeat round] none	BOX MODEL		marquee-play-count	infinite <i>integer</i>
border-left	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>	clear	left right both none	marquee-speed	slow normal fast
border-left-color	<i>border-color</i>	display	none inline block inline-block list-item run-in compact table inline-table 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	marquee-style	scroll slide alternate
border-left-style	<i>border-style</i>			overflow	visible hidden scroll auto no-display no-content <i>overflow-x</i> <i>overflow-y</i>
border-left-width	thin medium thick <i>length</i>			overflow-style	auto marquee-line marquee-block
border-right	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>			overflow-x	visible hidden scroll auto no-display no-content
border-right-color	<i>border-color</i>			overflow-y	visible hidden scroll auto no-display no-content
border-right-style	<i>border-style</i>			rotation	<i>angle</i>
border-right-width	thin medium thick <i>length</i>			rotation-point	<i>position</i> (<i>paired value</i> off-set)
				visibility	visible hidden collapse

Cascading Style Sheets (CSS 3)

TEXT	
direction	ltr rtl inherit
hanging-punctuation	none [start end end-edge]
letter-spacing	normal <i>length</i> %
punctuation-trim	none [start end adjacent]
text-align	start end left right center justify
text-align-last	start end left right center justify
text-decoration	none underline overline line-through blink
text-emphasis	none [[accent dot circle disc] [before after]?]
text-indent	<i>length</i> %
text-justify	auto inter-word inter-ideograph inter-cluster distribute kashida tibetan
text-outline	none <i>color</i> <i>length</i>
text-shadow	none <i>color</i> <i>length</i>
text-transform	none capitalize uppercase lowercase
text-wrap	normal unrestricted none suppress
unicode-bidi	normal embed bidi-override
white-space	normal pre nowrap pre-wrap pre-line
white-space-collapse	preserve collapse preserve-breaks discard
word-break	normal keep-all loose break-strict break-all
word-spacing	normal <i>length</i> %
word-wrap	normal break-word
COLUMN	
column-count	auto <i>number</i>
column-fill	auto balance
column-gap	normal <i>length</i>
column-rule	<i>column-rule-width</i> <i>column-rule-style</i> <i>column-rule-color</i>
column-rule-color	<i>color</i>
column-rule-style	<i>border-style</i>
column-rule-width	thin medium thick <i>length</i>
columns	<i>column-width</i> <i>column-count</i>
column-span	1 all
column-width	auto <i>length</i>
COLOR	
color	inherit <i>color</i>
opacity	inherit <i>number</i>

TEMPLATE LAYOUT	
box-align	start end center base-
box-direction	normal reverse
box-flex	<i>number</i>
box-flex-group	<i>integer</i>
box-lines	single multiple
box-orient	horizontal vertical inline-axis block-axis
box-pack	start end center justify
box-sizing	content-box padding-box border-box margin-box
tab-side	top bottom left right
TABLE	
border-collapse	collapse separate
border-spacing	<i>length</i> <i>length</i>
caption-side	top bottom left right
empty-cells	show hide
table-layout	auto fixed
SPEECH	
cue	<i>cue-before</i> <i>cue-after</i>
cue-before	<i>uri</i> [silent x-soft soft medium loud x-loud] none inherit <i>number</i> %
cue-after	<i>uri</i> [silent x-soft soft medium loud x-loud] none inherit <i>number</i> %
mark	<i>mark-before</i> <i>mark-after</i>
mark-before	<i>string</i>
mark-after	<i>string</i>
pause	<i>pause-before</i> <i>pause-after</i>
pause-before	none x-weak weak medium strong x-strong inherit <i>time</i>
pause-after	none x-weak weak medium strong x-strong inherit <i>time</i>
phonemes	<i>string</i>
rest	<i>rest-before</i> <i>rest-after</i>
rest-before	none x-weak weak medium strong x-strong inherit <i>time</i>
rest-after	none x-weak weak medium strong x-strong inherit <i>time</i>
speak	none normal spell-out digits literal-punctuation no-punctuation inherit
voice-balance	left center right leftwards rightwards inherit <i>number</i>
voice-duration	<i>time</i>

SPEECH	
voice-family	inherit [<specific-voice, age, generic-voice, number>]
voice-rate	x-slow slow medium fast x-fast inherit %
voice-pitch	x-low low medium high x-high inherit <i>number</i> %
voice-pitch-range	x-low low medium high x-high inherit <i>number</i>
voice-stress	strong moderate none reduced inherit
voice-volume	silent x-soft soft medium loud x-loud inherit <i>number</i> %
LIST & MARKERS	
list-style	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>
list-style-image	none <i>url</i>
list-style-position	Inside outside
list-style-type	none asterisks box check circle diamond disc hyphen square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha footnotes
marker-offset	auto <i>length</i>
ANIMATIONS	
animation	<i>animation-name</i> <i>animation-duration</i> <i>animation-timing-function</i> <i>animation-delay</i> <i>animation-iteration-count</i> <i>animation-direction</i>
animation-delay	<i>time</i>
animation-direction	normal alternate
animation-duration	<i>time</i>
animation-iteration-count	inherit <i>number</i>
animation-name	none IDENT
animation-play-state	running paused
animation-timing-function	ease linear ease-in ease-out ease-in-out cubic-Bezier (<i>number</i> , <i>number</i> , <i>number</i> , <i>number</i>)
TRANSITIONS	
transition	<i>transition-property</i> <i>transition-duration</i> <i>transition-timing-function</i> <i>transition-delay</i>
transition-delay	<i>time</i>
transition-duration	<i>time</i>
transition-property	none all
transition-timing-function	ease linear ease-in ease-out ease-in-out cubic-Bezier (<i>number</i> , <i>number</i> , <i>number</i> , <i>number</i>)

Cascading Style Sheets (CSS 3)

GRID POSITIONING	
grid-columns	none inherit [length percentage relative length]
grid-rows	none inherit [length percentage relative length]
OUTLINE	
outline	outline-color outline-style outline-width
outline-color	color invert
outline-offset	inherit length
outline-style	None dotted dashed solid double groove ridge inset outset
outline-width	thin medium thick length
3D / 2D TRANSFORM	
backface-visibility	visible hidden
perspective	none number
perspective-origin	[[[percentage> <length> left center right] [<percentage> <length> top center bottom]?] <length>] [[[left center right] [top center bottom]]]
transform	none matrix matrix3d translate3d translateX translateY translateZ scale scale3d scaleX scaleY scaleZ rotate rotate3d rotateX rotateY rotateZ skewX skewY skew perspective
transform-origin	[[[<percentage> <length> left center right] [<percentage> <length> top center bottom]?] <length>] [[[left center right] [top center bottom]]]
transform-style	flat preserve-3d
GENERATED CONTENT	
bookmark-label	content attr string
bookmark-level	none integer
bookmark-target	self uri attr
border-length	auto length
content	normal none inhibit uri
counter-increment	none identifier number
counter-reset	none identifier number
crop	auto shape
display	normal none list-item
float-offset	length length

GENERATED CONTENT	
hyphenate-after	auto integer
hyphenate-before	auto integer
hyphenate-character	auto string
hyphenate-lines	no-limit integer
hyphenate-resource	none uri
hyphens	none manual auto
image-resolution	normal auto dpi
marks	[crop cross] none
move-to	normal here identifier
page-policy	start first last
quotes	none string string string string
string-set	identifier content-list
text-replace	none [<string> <string>]+
LINE BOX	
alignment-adjust	auto baseline before-edge text-before-edge middle central after-edge text-after-edge ideographic alphabetic hanging mathematical length %
alignment-baseline	baseline use-script before-edge text-before-edge after-edge text-after-edge central middle ideographic alphabetic hanging mathematical
baseline-shift	baseline sub super length %
dominant-baseline	auto use-script no-change reset-size alphabetic hanging ideographic mathematical central middle text-after-edge text-before-edge
drop-initial-after-align	alignment-baseline
drop-initial-after-adjust	central middle after-edge text-after-edge ideographic alphabetic mathematical length %
drop-initial-before-align	caps-height alignment-baseline
drop-initial-before-adjust	before-edge text-before-edge central middle hanging mathematical length %
drop-initial-value	initial integer
drop-initial-size	auto integer % line
inline-box-align	initial last integer

LINE BOX	
line-height	normal number length %
line-stacking	line-stacking-strategy line-stacking-ruby line-stacking-shift
line-stacking-strategy	inline-line-height block-line-height max-height grid-height
line-stacking-ruby	exclude-ruby include-ruby
line-stacking-shift	consider-shifts disregard-shifts
text-height	auto font-size text-size max-size
vertical-align	Baseline sub super top text-top middle bottom text-bottom length %
HYPERLINK	
target	target-name target-new target-position
target-name	current root parent new modal string
target-new	window tab none
target-position	above behind front back
POSITIONING	
bottom	auto % length
clip	shape auto
left	auto % length
position	static relative absolute fixed
right	auto % length
top	auto % length
z-index	auto number
RUBY	
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

Cascading Style Sheets (CSS 3)

PAGED MEDIA	
fit	fill hidden meet slice
fit-position	[top center bottom] [left center right] <i>length</i> %
image-orientation	auto <i>angle</i>
orphans	<i>integer</i>
page	auto <i>identifier</i>
page-break-after	auto always avoid left right
page-break-before	auto always avoid left right
page-break-inside	auto avoid
size	auto landscape portrait <i>length</i>
windows	<i>integer</i>
UI	
appearance	normal inherit [icon window desktop workspace document tooltip dialog button push-button hyperlink radio-button checkbox menu-item tab menu menubar pull-down-menu pop-up-menu list-menu radio-group checkbox-group outline-tree range field combo-box signature password]
cursor	auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help <i>url</i>
icon	auto inherit <i>url</i>
nav-index	auto inherit <i>number</i>
nav-up	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
nav-right	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
nav-down	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
nav-left	auto inherit <id> [<i>current</i> <i>root</i> <target-name>]
resize	none both horizontal vertical inherit

- Values in italics are place holders for an actual value (like 1px, 1em, 1%), values in normal text are values that can be used as the actual value
- CSS Properties in **Dark Red** are shorthand properties and each value must be defined. The exception is where the property can define from one to four of the sides of a box element property (Top-Right-Bottom-Left) - i.e. border-width

Cascading Style Sheets (CSS 3)

PSEUDO-CLASS		UNITS		SELECTOR TYPES		
:active	an activated element	ABSOLUTE MEASUREMENT		Name	Info	Example
:focus	an element while the element has focus	%	percentage	Universal	Any element	* { font: 10px Arial; }
:visited	a visited link	cm	centimeter	Type	Any element of that type	h1 { text-decoration: underline; }
:hover	an element when you mouse over it	in	inch	Grouping	Multiple elements of different types	h1, h2, h3 { font-family: Verdana; }
:link	an unvisited link	mm	millimeter	Class	Multiple elements of different types when you don't want to affect all instances of that type	.sampleclass { text-decoration: underline; }
:disabled	an element while the element is disabled	pc	pica (1p = 12 points)	Id	A single element type when you don't want to affect all instances of that type	#sampleid { text-decoration: underline; }
:enabled	an element while the element is enabled	pt	point (1pt = 1/72 inch)	Descendant	An element that is below (in the document tree) another element—no matter how many levels below	#gallery h1 { text-decoration: underline; }
:checked	an element (form element control) that is checked	RELATIVE MEASUREMENT		Child	An element that is directly below (in the document tree) another element	#title > p { font-weight: bold; }
:selection	an element that is currently selected of highlighted by the user	ch	width of the "0" glyph found in the font for the font size used to render	Adjacent Sibling	All elements that share the same parent and elements are in the same immediate sequence	h1 + p { font-style: italic; }
:lang	Allows the author to specify a language to use in a specified element	em	1em = current font size of current element	General Sibling	All elements that share the same parent and elements are in the same sequence (not necessarily immediate)	h1 ~ p { font-style: italic; }
:nth-child(n)	an element that is the n-th sibling	ex	x-height of the element's font	Attribute	An element with that matches the attribute listed	E[selected] - att whatever the value E[att="val"] - att with a specific value E[rel~="next"] - att with a value is a whitespace separated list *[lang]="en" - att value either being exactly "val" or beginning with "val" immediately followed by "-" E[att^="val"] - att value that begins with the prefix "val" E[att\$="val"] - att value that end with the suffix "val" E[att*="val"] - att value contains at least one instance of the substring "val"
:nth-last-child(n)	an element that is the n-th sibling counting from the last sibling	gd	the grid defined by 'layout-grid'			
:first-child	an element that is the first sibling	px	pixel of the viewing device			
:last-child	an element that is the last sibling	rem	the font size of the root element			
:only-child	an element that is the only child	vh	the viewport's height			
:nth-of-type(n)	an element that is the n-th sibling of its type.	vw	the viewport's width			
:nth-last-of-type(n)	an element that is the n-th sibling of its type counting from the last sibling	vm	viewport's height or width, whichever is smaller of the two			
:last-of-type	an element that is the first sibling of its type	ANGLES				
:first-of-type	an element that is the last sibling of its type	deg	degrees			
:only-of-type	an element that is the only child of that type	grad	grads			
:empty	an element that has no children	rad	radians			
:root	root element within the document	turn	turns			
:not(x)	an element not represented by the argument 'x'	TIME				
:target	a target element as specified by a target in a URI	ms	milli-seconds			
PSEUDO-ELEMENT		s	seconds			
::first-letter	Adds special style to the first letter of a text	FREQUENCY				
::first-line	Adds special style to the first line of a text	Hz	hertz			
::before	Inserts some content before an element	kHz	kilo-hertz			
::after	Inserts some content after an element	COLORS				
		color name	red, blue, green, dark green			
		rgb(x,y,z)	red = rgb(255,0,0)			
		rgb(x%,y%,z%)	red = rgb(100%,0,0)			
		rgba(x,y,z, alpha)	red = rgba(255,0,0)			
		#rrggbb	red = #ff0000 (or shorthand = #f00)			
		hsl(hue, saturation, lightness)	red = hsl(0, 100%, 50%)			
		hsla(hue, saturation, lightness, alpha)	red = hsl(0, 100%, 50%)			
		flavor	An accent color (typically chosen by the user) to customize the user interface of the user agent itself			
		currentColor	computed value of the 'currentColor' keyword is the computed value of the 'color' property			