

INTRODUCCIÓN A HTML5 Y CSS3

Víctor Custodio, 2016

¿QUÉ ES HTML5?

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web

HTML5 BREVE HISTORIA

- El limitado objetivo de HTML motivó a varias compañías a desarrollar nuevos lenguajes y programas para agregar características a la web nunca antes implementadas.
- De las opciones propuestas, Java y Flash fueron las más exitosas; ambas fueron masivamente adoptadas y ampliamente consideradas como el futuro de Internet
- Limitaciones presentes en estas dos tecnologías probaron ser una sentencia de muerte:
 - **Falta de integración:** Ambos fueron concebidos desde el principio como complementos (plug-ins), algo que se inserta dentro de una estructura pero que comparte con la misma solo espacio en la pantalla. No existía comunicación e integración alguna entre aplicaciones y documentos

HTML5 BREVE HISTORIA

- Javascript, un lenguaje interpretado incluido en navegadores, claramente era la manera de mejorar la experiencia de los usuarios y proveer funcionalidad para la web.
- Sin embargo, después de algunos años de intentos fallidos para promoverlo y algunos malos usos, el mercado nunca lo adoptó plenamente y pronto su popularidad declinó.
 - En ese momento, Javascript no era capaz de reemplazar la funcionalidad de Flash o Java.
 - Populares funciones como la reproducción de video se estaban convirtiendo en una parte esencial de la web y solo eran efectivamente ofrecidas a través de estas tecnologías.

HTML5 BREVE HISTORIA

- Respecto a Java, la naturaleza compleja del lenguaje, su evolución lenta y la falta de integración disminuyeron su importancia hasta el punto en el que hoy día no es más usado en aplicaciones web de importancia. Sin Java, el mercado volcó su atención a Flash. Pero el hecho de que Flash comparte las mismas características básicas que su competidor en la web lo hace también susceptible de correr el mismo destino.
- Mientras JavaScript seguía evolucionando con nuevas funciones y técnicas y los navegadores también mejoraban los interpretes del lenguaje.
- Impulsados por las mejoras otorgadas por los navegadores, los desarrolladores web comenzaron a aplicar Javascript en sus aplicaciones de un modo nunca visto. Gracias a estas nuevas implementaciones, Javascript, HTML y CSS se convirtieron pronto en la más perfecta combinación para la necesaria evolución de la web.

HTML5 BREVE HISTORIA

- HTML5 es, de hecho, una mejora de esta combinación, el pegamento que une todo. HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. A partir de ahora:
 - HTML provee los elementos estructurales,
 - CSS se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista
 - Javascript tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales

HTML5- ESTRUCTURA BÁSICA

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Mi primer documento Html 5</title>
```

```
  </head>
```

```
  <body>
```

```
    Hola Mundo HTML5!
```

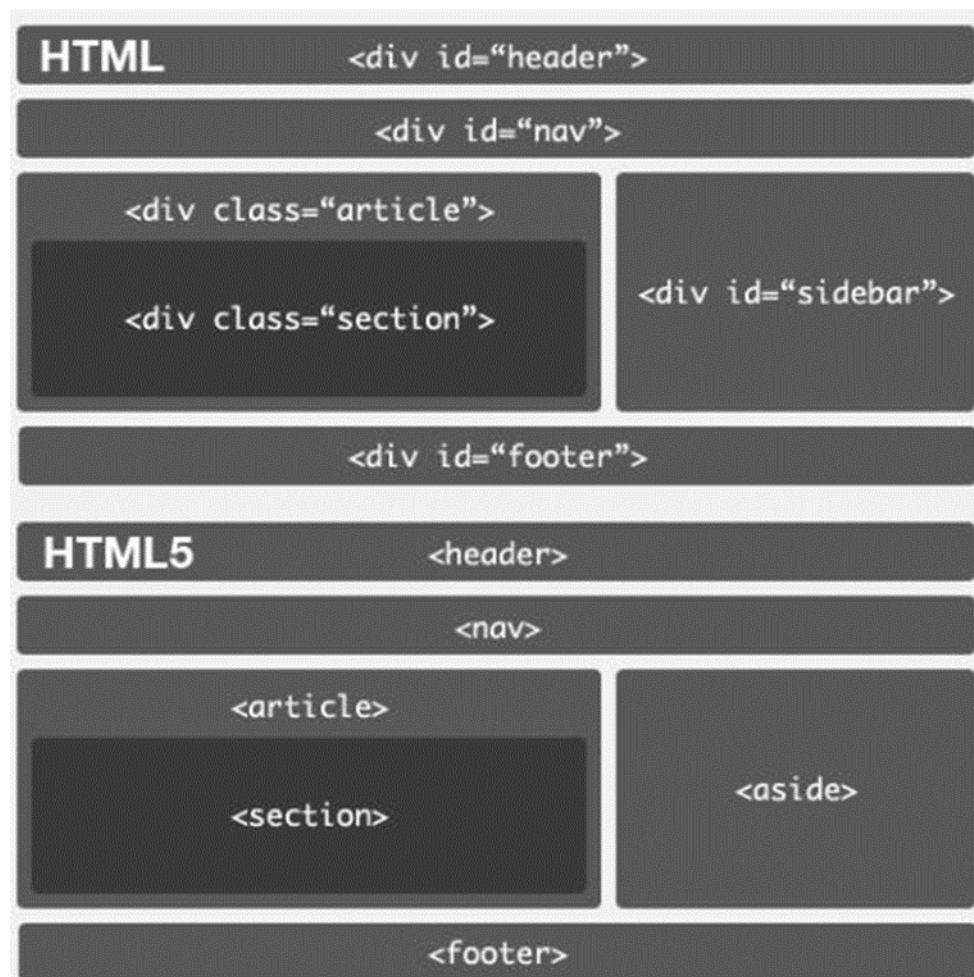
```
  </body>
```

```
</html>
```

HTML5- ESTRUCTURA BASE PARA UN DISEÑO

```
<head>
  <meta charset="utf-8">
  <title>Estructura Basica</title>
  <meta name="description" content="">
  <meta name="keywords" content="" />
  <meta name="author" content="">
  <!-- CSS -->
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div id="container">
    <header id="cabecera">
      <nav id="menú">
        <ul>
          <li><a href="#" title="Inicio">Inicio</a></li> <li><a href="#" title="Acerca De">Acerca De</a></li>
          <li><a href="#" title="Servicios">Servicios</a></li> <li><a href="#" title="Contacto">Contacto</a></li>
        </ul>
      </nav>
    </header>
    <main id="principal">
      <article id="articulo1">
        <section id="subsection"> </section>
      </article>
      <article id="Articulo2"></article>
    </main>
    <aside id="columna">
      <ul>
        <li><a href="#" title="Articulo1">Articulo 1</a></li>
        <li><a href="#" title="Acerca De">Articulo 2</a></li> </ul>
    </aside>
    <footer>Todos los derechos reservados </footer>
  </div>
  <!-- Javascript en el footer para que la pagina cargue mas rápido -->
  <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
  <script type="text/javascript" src="scripts/funcionalidades.js"></script>
</body>
</html>
```


HTML5- ESTRUCTURA BASE PARA UN DISEÑO



HTML5

Además de las etiquetas estructurales recientemente vistas, html 5 trae consigo nuevas etiquetas de estructuras, algunas importantes se muestran en la siguiente tabla:

Etiqueta	Descripción
<code><details></code>	Define los detalles adicionales que el usuario pueda ver u ocultar
<code><figcaption></code>	Define un título para un elemento <code><figure></code>
<code><figure></code>	Define el contenido en sí misma, como ilustraciones, diagramas, fotos, listas de códigos, etc
<code><mark></code>	Define texto marcado o destacado
<code><progress></code>	Define el progreso de una tarea
<code><summary></code>	Define un título visible para un elemento <code><details></code>

<DETAILS> AND <SUMMARY>

<details>

<summary>Copyright 1999-2014.</summary>

<p> - by Refsnes Data. All Rights Reserved.</p>

<p>All content and graphics on this web site are the property of the company Refsnes Data.</p>

</details>

► Copyright 1999-2014.

Note: Actualmente sólo soportado por Safari, Opera y Chrome.

▼ Copyright 1999-2014.

- by Refsnes Data. All Rights Reserved.

All content and graphics on this web site are the property of the company Refsnes Data.

Note: Actualmente sólo soportado por Safari, Opera y Chrome.

<FIGURE> AND <FIGCAPTION>

```
<!DOCTYPE html>
<html>
<body>
<p>The Pulpit Rock is a massive cliff 604 metres in Norway.</p>
<figure>
  
  <figcaption>Fig.1 - A view of the pulpit rock in Norway.</figcaption>
</figure>
</body>
</html>
```

<FIGURE> AND <FIGCAPTION>

The Pulpit Rock is a massive cliff 604 metres in Norway.



Fig.1 - A view of the pulpit rock in Norway.

<MARK>

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>Do not forget to buy  
<mark>milk</mark> today.</p>
```

```
</body>
```

```
</html>
```

Do not forget to buy milk today.

<PROGRESS>

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

Downloading progress:

```
<progress value="22" max="100"></progress>
```

```
<p><strong>Note:</strong> Etiqueta no  
soportada en Internet Explorer 9 y  
anteriores.</p>
```

```
</body>
```

```
</html>
```

Downloading progress: 

Note: The progress tag is not supported in Internet Explorer 9 and earlier versions.

HTML5 AUDIO Y VIDEO

La importancia del audio y sobretodo del video en los últimos años llevó a los creadores de html5 a desarrollar la reproducción de estos medios de forma estándar y nativa para los navegadores (hasta entonces tarea encargada a software externo(java,flash,quicktime....)):

HTML5 <VIDEO>

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  Your browser does not support the video tag.  
</video>
```

Los formatos de video soportados son :

- **MP4** = MPEG 4 files with H264 video codec and AAC audio codec
- **WebM** = WebM files with VP8 video codec and Vorbis audio codec
- **Ogg** = Ogg files with Theora video codec and Vorbis audio codec

HTML5 <AUDIO>

<audio controls>

<source src= "audio.ogg" type="audio/ogg">

<source src="audio.mp3" type="audio/mpeg">

Your browser does not support the audio tag.

</audio>

Los formatos de audio soportados son:

- MP3
- OGG
- WAV

HTML5 AUDIO Y VIDEO

Existen multitud de atributos, métodos y eventos para trabajar con audio y video en HTML5:

- Algunos métodos (accesibles por javascript) como `load ()`, `play()`, `pause()`..
- Algunos atributos como `audioTracks`, `autoplay`, `currentTime`, `duration`...
- Algunos eventos que pueden ser escuchados como: `playing`, `ended`, `waiting`, `volumeChange`..

Puedes ver la lista completa con ejemplos en http://www.w3schools.com/tags/ref_av_dom.asp

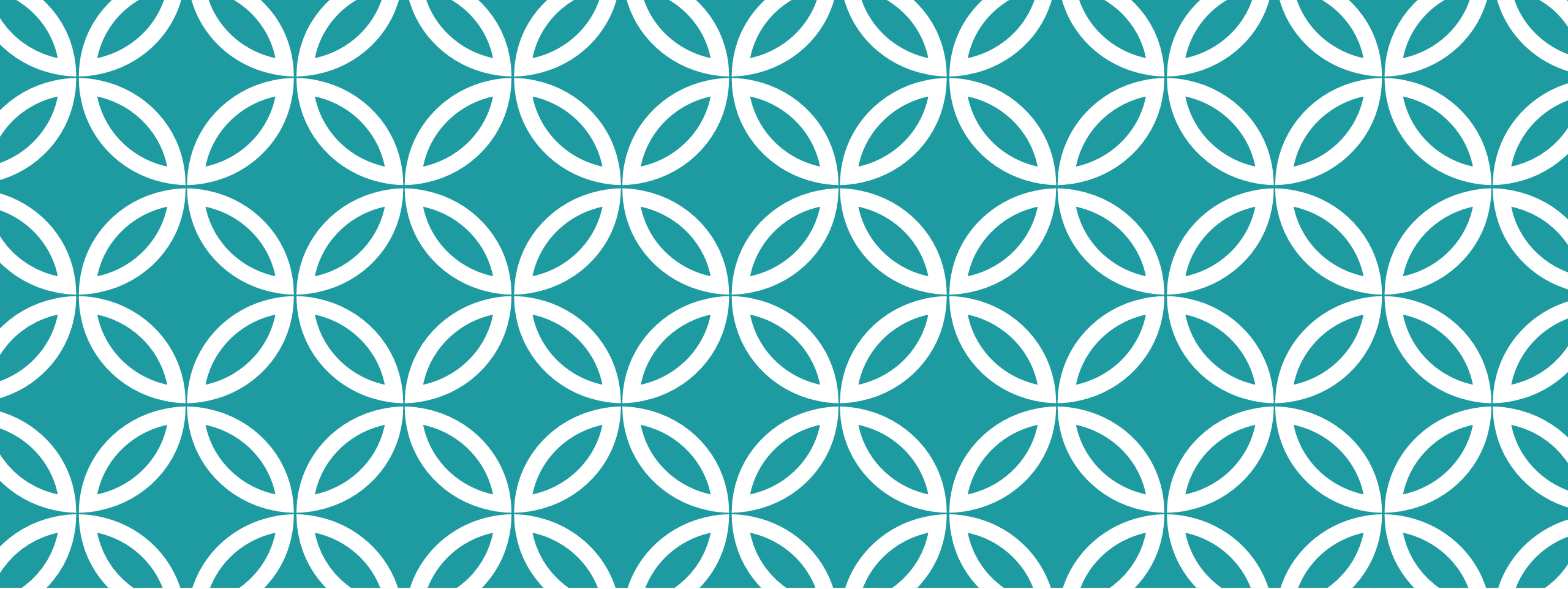
AÑADIR OTROS ELEMENTOS

Podemos visualizar otros elementos como applets, flash, pdf e imágenes con las etiquetas `<object>` y `<embed>`

```
<embed width="400" height="50" src="document.pdf" />
```

```
<object width="400" height="50" data="document.pdf" />
```

Puedes usar `<object>`, `<embed>`, o `<iframe>` para insertar videos de youtube en tu página.



CSS3 Y HTML5

CSS Y HTML

- HTML5 no describe solo los nuevos elementos HTML o el lenguaje mismo. La web demanda diseño y funcionalidad, no solo organización estructural o definición de secciones. En este nuevo paradigma, HTML se presenta junto con CSS y Javascript como un único instrumento integrado
- Oficialmente CSS nada tiene que ver con HTML5. CSS no es parte de la especificación y nunca lo fue. Este lenguaje es, de hecho, un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML.
- Al comienzo, atributos dentro de las etiquetas HTML proveían estilos esenciales para cada elemento, pero a medida que el lenguaje evolucionó, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo más satisfacer las demandas de diseñadores.

CSS Y HTML

- CSS ha crecido y ganado importancia, pero siempre desarrollado en paralelo, enfocado en las necesidades de los diseñadores y apartado del proceso de evolución de HTML.
- La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web y esta es la razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a CSS3, aunque oficialmente se trate de dos tecnologías completamente separadas

CSS: REFERENCIAS

Existen varios métodos para seleccionar cuáles elementos HTML serán afectados por las reglas CSS:

- Referencia por la palabra clave del elemento:

- `p { font-size: 20px }`

- Referencia por el atributo **id**:

- `#texto1 { font-size: 20px }`

`<p id="texto1">Mi texto</p>`

- Referencia por el atributo **class**:

- `.texto1 { font-size: 20px }`

`<p class="texto1">Mi texto</p>`

`<p class="texto1">Mi texto</p>`

CSS: REFERENCIAS

- Referencias combinadas
 - `p.texto1 { font-size: 20px }`
- Referencias con cualquier atributo:
 - `p[name="mitexto"] { font-size: 20px }`
- Referencias con expresiones regulares
 - `p[name^="mi"] { font-size: 20px }` -> todos los p con name que comience por "mi"
 - `p[name$="mi"] { font-size: 20px }` -> todos los p con name que finalice en "mi"
 - `p[name*="mi"] { font-size: 20px }` -> todos los p con name que incluya "mi"

CSS: REFERENCIAS

- **Referencias con pseudoclases** : son clases no definidas, se basan en la estructura de árbol de las etiquetas de html o acciones del usuario. Ejemplo:

```
<body>
<div id="wrapper">
<p class="mitexto1">Mi texto1 </p>
<p class="mitexto2">Mi texto2</p>
<p class="mitexto3">Mi texto3</p>
<p class="mitexto4">Mi texto4</p>
</div>
</body>
```

Contiene cuatro elementos **<p>** que, considerando la estructura HTML, son hermanos entre sí e hijos del mismo elemento **<div>**.

CSS: REFERENCIAS

- **Referencias con pseudoclases:**

- **Ejemplo:**

- `p:nth-child(2){ background: #999999;}`

Afectará a toda caja `<p>` que sea 2º hijo

- **Ejemplo: Efecto cebra**

- `p:nth-child(odd){ background: #999999; }`

- `p:nth-child(even){background: #CCCCCC; }`

Afectará a los hijos pares e impares de forma distinta

Más pseudoclases y pseudoelementos en ->

http://librosweb.es/libro/css_avanzado/capitulo_3/pseudo_clases.html

CSS: SELECTORES

- **Selector de herencia**

- **div > p{ color: #990000; }** -> Afecta a los elementos p cuando tienen a div como su padre(selector por defecto si no se indica símbolo).
- **p.mitexto2 + p{ color: #990000; }** -> Afecta a un elemento p cuando se encuentra a continuación de un p de clase miTexto2.
- **p.mitexto2 ~ p{color: #990000;}** -> todos los p precedidos de un p de clase miTexto2, aunque tengan otros elementos por medio

CSS 3, LA EVOLUCIÓN

En un intento por reducir el uso de código Javascript y para estandarizar funciones populares, CSS3 no solo cubre diseño y estilos web sino también forma y movimiento.

Algunas de las funciones que veremos aun están en fase de experimentación en algunos navegadores, cuando alguna propiedad no funcione en un determinado navegador, es necesario añadir el prefijo del navegador antes de la propiedad para usar su propiedad en fase experimental.

Los prefijos para los navegadores más comunes son los siguientes:

- **-moz-** para Firefox.
- **-webkit-** para Safari y Chrome.
- **-o-** para Opera.
- **-khtml-** para Konqueror.
- **-ms-** para Internet Explorer.
- **-chrome-** específico para Google Chrome

CSS 3, LA EVOLUCIÓN

Probemos los nuevos elementos con una nueva plantilla más simple:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Nuevos Estilos CSS3</title>
<link rel="stylesheet" href="nuevocss3.css">
</head>
<body>
<header id="principal">
<span id="titulo">Estilos CSS Web 2.0</span>
</header>
</body>
</html>
```

CSS 3, ESTILOS BÁSICOS

```
body {  
  text-align: center;  
}  
#principal {  
  display: block;  
  width: 500px;  
  margin: 50px auto;  
  padding: 15px;  
  text-align: center;  
  border: 1px solid #999999;  
  background: #DDDDDD;  
}  
#titulo {  
  font: bold 36px verdana, sans-serif;  
}
```

CSS 3, BORDER RADIUS

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
-moz-border-radius: 20px;  
-webkit-border-radius: 20px;  
border-radius: 20px;  
}
```

Border-radius tiene cuatro parámetros pero al igual que margin o padding se puede especificar 1,2 o 4, si se quiere un valor distinto en cada esquina.

En este ejemplo se puede ver el uso de los **prefijos**, por simplicidad no se volverá a ver en el resto de ejemplos.

CSS 3, BORDER RADIUS

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
border-radius: 20px / 10px;  
}
```

Border-radius También podemos dar forma a las esquinas declarando un segundo grupo de valores separados por una barra. Los valores a la izquierda de la barra representarán el radio horizontal mientras que los valores a la derecha representan el radio vertical. La combinación de estos valores genera una elipsis

CSS 3, BOX SHADOW

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
border-radius: 20px;  
box-shadow: rgb(150,150,150) 5px 5px;  
}
```

Box-shadow: Necesita al menos tres valores:

- El color (esta vez en rgb y decimal)
- El desplazamiento de la sombra (positivo o negativo) horizontal y vertical desde la sombra al elemento.

CSS 3, BOX SHADOW

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
border-radius: 20px;  
box-shadow: rgb(150,150,150) 5px 5px 10px;  
}
```

Box-shadow: Un cuarto valor se puede agregar a la propiedad ya estudiada es la distancia de difuminación. Con este efecto ahora la sombra lucirá más real.

CSS 3, BOX SHADOW

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
border-radius: 20px;  
box-shadow: rgb(150,150,150) 5px 5px 10px inset;  
}
```

Box-shadow: El último valor posible para **box-shadow** no es un número sino más bien una palabra clave: **inset**. Esta palabra clave convierte a la sombra externa en una sombra interna, lo cual provee un efecto de profundidad al elemento afectado

TEXT-SHADOW

box-shadow fue diseñada especialmente para ser aplicada en cajas .Para crear sombras para figuras irregulares como textos, existe una propiedad especial llamada **text-shadow**.

```
#titulo {  
font: bold 36px verdana, sans-serif;  
text-shadow: rgb(0,0,150) 3px 3px 5px;  
}
```

Los valores son similares a los usados en la box-shadow

@FONT-FACE

La propiedad **@font-face** permite a los diseñadores proveer un archivo conteniendo una fuente específica para mostrar sus textos en la página. Ahora podemos incluir cualquier fuente que necesitemos con solo proveer el archivo adecuado:

```
#titulo {  
font: bold 36px MiNuevaFuente, verdana, sans-serif;  
text-shadow: rgb(0,0,150) 3px 3px 5px;  
} @  
font-face { font-family: 'MiNuevaFuente';  
src: url('font.ttf');  
}
```

Debemos tener el fichero Font.ttf en la misma carpeta que el archivo css

GRADIENTE LINEAL

Los gradientes son uno de los efectos más atractivos entre aquellos incorporados en CSS3. Una propiedad **background** con algunos pocos parámetros es suficiente para dotar a una página web con un aspecto profesional

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
border-radius: 20px;  
box-shadow: rgb(150,150,150) 5px 5px 10px;  
background: linear-gradient(to top, #FFFFFF, #006699);
```

- Los gradientes son configurados como fondos, por lo que podemos usar las propiedades **background** o **background-image** para declararlos.
- La sintaxis para los valores declarados en estas propiedades es **linear-gradient (posición inicio, color inicial, color final)**.
 - El primer valor puede ser especificado en pixeles, porcentaje o usando las palabras clave **top**, **bottom**, **left** y **right**

GRADIENTE LINEAL

- El punto de comienzo puede ser reemplazado por un ángulo para declarar una dirección específica del gradiente

```
background: linear-gradient(30deg, #FFFFFF, #006699);
```

- También podemos declarar los puntos de terminación para cada color

```
background: linear-gradient(to top, #FFFFFF 50%, #006699 90%);
```


GRADIENTE RADIAL

- La sintaxis estándar para los gradientes radiales solo difiere en unos pocos aspectos con respecto a la anterior. Debemos usar la función `radial-gradient()`.

Algunos ejemplos

- Múltiples colores : **`background: radial-gradient(blue,red, green)`**
- Con forma : (circle o ellipse) **`background :radial-gradient(circle, red, yellow, green);`**
- Especificando punto de terminación para cada color:
`background: radial-gradient(red 5%, green 15%, blue 60%);`

RGBA

Hasta este momento los colores fueron declarados como sólidos utilizando valores hexadecimales o la función **rgb()** para decimales. CSS3 ha agregado una nueva función llamada **rgba()** que simplifica la asignación de colores y transparencias.

La función **rgba()** tiene cuatro atributos. Los primeros tres son similares a los usados en **rgb()** y simplemente declaran los valores para los colores rojo, verde y azul en números decimales del 0 al 255. El último, en cambio, corresponde a la nueva capacidad de opacidad. Este valor se debe encontrar dentro de un rango que va de 0 a 1, con 0 como totalmente transparente y 1 como totalmente opaco.

```
#titulo {  
font: bold 36px MiNuevaFuente, verdana, sans-serif;  
text-shadow: rgba(0,0,0,0.5) 3px 3px 5px;  
}
```

OUTLINE

Era una propiedad vieja de CSS, era usada para crear un segundo borde, y ahora ese borde puede ser mostrado alejado del borde real del elemento.

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
outline: 2px dashed #000099;  
outline-offset: 15px;  
}
```

La propiedad **outline** tiene similares características y usa los mismos parámetros que **border**. La propiedad **outline-offset** solo necesita un valor en pixeles

TRANSFORM Y TRANSITION

Los elementos HTML, cuando son creados, son como bloques sólidos e inamovibles. Pueden ser movidos usando código Javascript o aprovechando librerías populares como jQuery (www.jquery.com), por ejemplo, pero no existía un procedimiento estándar para este propósito hasta que CSS3 presentó las propiedades **transform** y **transition**.

Ahora ya no tenemos que pensar en cómo hacerlo. En su lugar, solo tenemos que conocer cómo ajustar unos pocos parámetros y nuestro sitio web puede ser tan flexible y dinámico como lo imaginamos.

TRANSFORM

La propiedad **transform** puede operar cuatro transformaciones básicas en un elemento: **scale** (escalar), **rotate** (rotar), **skew** (inclinarse) y **translate** (trasladar o mover).

TRANSFORM- SCALE

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
transform: scale(2);  
}
```

En el ejemplo, duplicamos la escala de la caja.

La función **scale** recibe dos parámetros: el valor **X** para la escala horizontal y el valor **Y** para la escala vertical. Si solo un valor es provisto el mismo valor es aplicado a ambos parámetros.

Prueba también con valores negativos (1,-1) por ejemplo

TRANSFORM- ROTATE

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
transform: rotate(30deg);  
}
```

La función **rotate** rota el elemento en la dirección de las agujas de un reloj. El valor debe ser especificado en grados usando la unidad “**deg**”:

En el ejemplo, rotamos la caja 30 grados

TRANSFORM- SKEW

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
transform: skew(20deg);  
}
```

La función **skew** cambia la simetría del elemento en grados y en ambas dimensiones.

La función **skew** usa dos parámetros, pero a diferencia de otras funciones, cada parámetro de esta función solo afecta una dimensión (los parámetros actúan de forma independiente).

Solo declaramos el primer parámetro, por lo que solo la dimensión horizontal de la caja será modificada.

En el ejemplo, cambiamos la simetría de la caja 20deg.

TRANSFORM- TRANSLATE

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
transform: translate(100px);  
}
```

La función **translate** mueve o desplaza el elemento en la pantalla a una nueva posición.

La esquina superior izquierda del elemento es la posición **0,0**, por lo que valores negativos moverán al objeto hacia la izquierda o hacia arriba de la posición original, y valores positivos lo harán hacia la derecha o hacia abajo.

Dos valores pueden ser declarados en esta función si queremos mover el elemento horizontal y verticalmente

En el ejemplo, movemos la caja 100 px hacia la derecha.

TRANSFORM- TODO A LA VEZ

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
transform: translate(0px,100px) rotate(45deg) scale(0.3);  
}
```

TRANSFORM- DINÁMICAS

Podemos aprovecharnos de la combinación de transformaciones y pseudo clases para convertir nuestra página en una aplicación dinámica:

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
}  
#principal: hover{  
transform: rotate(5deg);  
}
```

TRANSFORM- DINÁMICAS

Otras pseudoclasas son:

:link	a:link	Selecciona todos los links
:visited	a:visited	Seleeciona todos los links visitados
:active	a:active	Selecciona todos los links activos
:hover	a:hover	Selecciona los links sobre los que pasa el raton
:focus	input:focus	Selecciona el input sobre el que esta el foco
::first-letter	p::first-letter	Selecciona la primera letra del element p
::first-line	p::first-line	Selecciona la primera linea del element p
:first-child	p:first-child	Selecciona todos elementos <p> que sea los primeros hijos

TRANSICIONES

hermosos efectos usando transformaciones dinámicas son accesibles y fáciles de implementar. Sin embargo, una animación real requiere de un proceso de más de dos pasos.

La propiedad **transition** fue incluida para suavizar los cambios, creando mágicamente el resto de los pasos que se encuentran implícitos en el movimiento

TRANSICIONES

```
#principal {  
display: block;  
width: 500px;  
margin: 50px auto;  
padding: 15px;  
text-align: center;  
border: 1px solid #999999;  
background: #DDDDDD;  
transition: transform 1s ease-in-out 0.5s;  
}  
# principal:hover{  
transform: rotate(5deg);  
}
```

Transition puede tomar hasta cuatro parámetros separados por un espacio.

- El primer valor es la propiedad que será considerada para hacer la transición. Esto es necesario debido a que varias propiedades pueden cambiar al mismo tiempo
- El segundo parámetro especifica el tiempo que la transición se tomará para ir de la posición inicial a la final.
- El tercer parámetro puede ser cualquiera de las siguientes palabras clave: **ease**, **linear**, **ease-in**, **ease-out** o **ease-in-out**. Indica como se realiza la transición.
- El ultimo es el retardo, indica cuánto tiempo tardará la transición en comenzar

MEDIA QUERYS

Para poder realizar diseños adaptativos , hasta ahora se solían especificar las alturas y anchuras en porcentaje en vez de en pixeles para que el contenido se ajuste al tamaño de la pantalla (aunque no siempre se conseguía el efecto deseado). Pero ahora, con css3 y sus media queries es posible ajustar más las webs a los tamaños.

Basta con añadir algunas líneas como esta:

```
<!-- CSS media query en el elemento link -->
```

```
<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />
```

```
<!-- CSS media query dentro de la hoja de estilos -->
```

```
<style>
```

```
@media (max-width: 600px) {
```

```
  .facet_sidebar {
```

```
    display: none;
```

```
  }
```

```
}
```

```
</style>
```

MEDIA QUERYYS

Realmente se pueden hacer muchas discriminaciones usando media queries (orientación de la pantalla, si es una pantalla de ordenador o una smart tv, ...)

@media (min-width: 700px) and (orientation: landscape)

@media tv and (min-width: 700px) and (orientation: landscape)

y cada vez van saliendo más por lo que recomiendo visitar de tanto en cuando la web de la [w3c](#) que va publicando las nuevas queries que se van desarrollando.

MEDIA QUERYS — PLANTILLA (1/2)

Plantilla para abarcar un alto rango de dispositivos

```
/* Pantallas de gran resolución ----- */
@media all and (max-width : 1824px) {
}
/* PCs de escritorio y portátiles ----- */
@media all and (max-width : 1370px) {
}
/* Tabletas (Vertical y Horizontal) ----- */
@media all
and (min-width : 768px)
and (max-width : 1024px) {
}
/* Tabletas (Horizontal) ----- */
@media all and (max-width : 800px) {
}
/* Pequeñas Tabletas (Vertical y Horizontal) ----- */
@media all
and (min-width : 600px)
and (max-width : 800px) {
}
/* Pequeñas Tabletas (Apaisado o Horizontal) y Tabletas (Vertical) ----- */
@media all and (max-width : 750px) {
}
/* Smartphones (Apaisado o Horizontal) ----- */
@media all and (max-width : 640px) {
}
```

MEDIA QUERYS —PLANTILLA (2/2)

```
/* Pequeñas Tabletas (Vertical) ----- */
@media all and (max-width : 600px) {
}

/* Smartphones (Vertical y Horizontal) ----- */
@media all
and (min-width : 320px)
and (max-width : 480px) {
}

/* Smartphones (Vertical) ----- */
@media all and (max-width : 480px) {
}

/* Pequeños Smartphones (Apaisado o Horizontal) ----- */
@media all and (max-width : 430px) {
}

/* Smartphones (Vertical) ----- */
@media all and (max-width : 375px) {
}

/* Pequeños Smartphones (Vertical) ----- */
@media all and (max-width : 320px) {
}
```