

HTML5

CSS3

HTML5 y CSS3

Repaso JQuery

Selector pattern	Meaning	Examples
<code>\$("Element")</code>	Selects all elements with the given tag name	<code>\$ ("p")</code> selects all the p tags. <code>\$ ("body")</code> selects the body tag.
<code>\$("#id")</code>	Selects the element with the given ID of the attribute	Provides the following code: <pre><div id="box1"></div> <div id="box2"></div></pre> <code>\$ (" #box1 ")</code> selects the highlighted element.
<code>\$(".className")</code>	Selects all elements with the given class attribute	Provides the following code: <pre><div class="apple"></div> <div class="apple"></div> <div class="orange"></div> <div class="banana"></div></pre> <code>\$ (" . apple ")</code> selects the highlighted elements with class set to apple.
<code>\$("selector1, selector2, selectorN")</code>	Selects all elements that match the given selector	Provides the following code: <pre><div class="apple"></div> <div class="apple"></div> <div class="orange"></div> <div class="banana"></div></pre> <code>\$ (" . apple , . orange ")</code> selects the highlighted elements that class is set to, either apple or orange.

HTML5 y CSS3

Repaso JQuery

Event Method	Description
<code>\$(document).ready(function)</code>	Binds a function to the ready event of a document (when the document is finished loading)
<code>\$(selector).click(function)</code>	Triggers, or binds a function to the click event of selected elements
<code>\$(selector).dblclick(function)</code>	Triggers, or binds a function to the double click event of selected elements
<code>\$(selector).focus(function)</code>	Triggers, or binds a function to the focus event of selected elements
<code>\$(selector).mouseover(function)</code>	Triggers, or binds a function to the mouseover event of selected elements

HTML5 y CSS3

HTML5

Para usar HTML5 se utiliza el doctype.

```
<!DOCTYPE HTML>
```

Es compatible con versiones antiguas de navegadores.

HTML5 y CSS3

HTML5

```
<!DOCTYPE html>
<head>
<meta charset="utf-8">
<title>Hello world!</title>
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
<link rel="stylesheet" href="style.css">
<script src="code.js"></script>
</head>
<body>
<hgroup>
<h1>Hola HTML5!</h1>
<h2>Esta es mi primera web</h2>
</hgroup>
</body>
</html>
```

HTML5 y CSS3

HTML5

En HTML4

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

En HTML5

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

En HTML4

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

En HTML5

```
<link rel="stylesheet" href="style.css"/>
```

En HTML4

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

En HTML5

```
<script src="code.js"> </script>
```

HTML5 y CSS3

HTML5

En HTML4 sólo existía un elemento contenedor sin significado semántico **<div>**.

En HTML5 existen **Div**, **Section** y **Article** que además de cumplir con una comportamiento similar a div aportan semántica.

`<div>`

Es el contenedor genérico, es un elemento a nivel de bloque sin sentido adicional semántico.

`<section>`

Es un "documento genérico o sección de aplicación"
Normalmente, tiene un header y suele tener un footer .

`<article>`

Es una parte independiente del documento. También suele llevar un título y un footer.

HTML5 y CSS3

HTML5

`<aside>`

Se usa para definir la sidebar.

`<footer>`

La parte más baja, donde tenemos toda la información

`<header>`

La cabecera de la web.

`<nav>`

Se usa para definir el menú o la navegación de la página.

HTML5 y CSS3

HTML4

```
<div id="header">
```

```
<div id="nav">
```

```
<div id="main">
```

```
<div id="article">
```

```
<div id="article">
```

```
<div id="aside">
```

```
<div id="footer">
```

HTML5 y CSS3

HTML5



- article 
- aside 
- header 
- nav 
- section 

HTML5 y CSS3

HTML5



HTML5 y CSS3

HTML5

Nuevas Características

- Audio & Video
- Canvas
- Almacenamiento local
- WebSockets
- Semántica
- Compatibilidad hacia atrás
- Eficacia en el renderizado

HTML5 y CSS3

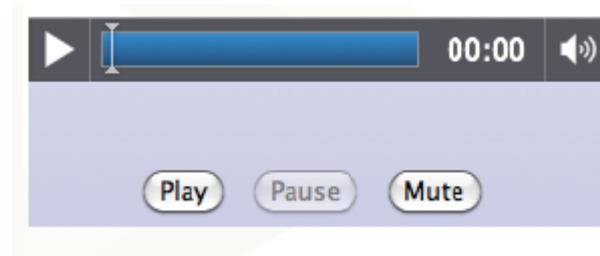
HTML5

Audio

La etiqueta `<audio>` define sonido o cualquier tipo de streaming de audio.

Atributos:

- autoplay
- controls
- loop
- preload
- src



HTML5 y CSS3

HTML5

La etiqueta **<command>** define un botón de comando o un radiobutton o un checkbox.

Atributos:

- checked
- disabled
- icon
- label
- radiogroup
- type

```
<command onclick="alert('Hello World')">Haz click aquí</command>
```

HTML5 y CSS3

HTML5

La etiqueta **<mark>** se usa para destacar una palabra o una parte del texto.

La etiqueta **<hgroup>** se usa para agrupar titulares (h1,h2)

```
<hgroup>  
<h1>Bienvenidos a mi web</h1>  
<h2>La web de HTML5</h2>  
</hgroup>
```

La etiqueta **<progress>** se usa para visualizar el progreso de una tarea que se esté realizando

```
<progress>  
<span id="objprogress">25</span>%  
</progress>
```

HTML5 y CSS3

HTML5

La etiqueta **<source>** se usa para definir la fuente de un archivo de audio o video.

```
<audio controls="controls">  
<source src="cancion.ogg" type="audio/ogg" />  
<source src="cancion.mp3" type="audio/mpeg" />  
</audio>
```

La etiqueta **<summary>** contiene el encabezado para el elemento **details**, que se usa para dar detalles sobre un documento o parte de un documento.

```
<details>  
<summary>Copyright por mi persona.</summary>  
<p>Lorem ipsum dolor sit amet....</p>  
</details>
```

HTML5 y CSS3

Formularios Web

HTML5 y CSS3

HTML5

Placeholder

Un placeholder es un texto que se muestra para ayudar al usuario y darle información.

En cuanto se hace click en el input, desaparece:



```
<form>  
  <input name="name" placeholder="Buscar aquí">  
  <input type="submit" value="Search">  
</form>
```

HTML5 y CSS3

HTML5

Autofocus

```
<input name="q" autofocus>
```

El autofocus no funciona en todos los navegadores, así que si quiere usar y tener máxima compatibilidad, se puede usar JavaScript.

```
<input id="i" autofocus>
<script>
function auto(){
    if (!("autofocus" in document.createElement("input"))) {
        document.getElementById("i").focus();
    }
}
window.onload = auto;
</script>
```

HTML5 y CSS3

HTML5

Tipos de Texto

Información que se puede utilizar para desplegar y/o validar el tipo de texto ingresado.

```
<input type="email">
```

```
<input type="url">
```

```
<input type="tel">
```



```
<input name="xx" type="range" min="0" max="10" step="6" value="4">
```

HTML5 y CSS3

HTML5

Tipos de input para elegir fechas

type=...

- "date"
- "month"
- "week"
- "time"
- "datetime"
- "datetime-local"
- "datetime" picker



HTML5 y CSS3

Canvas

HTML5 y CSS3

HTML5

Un lienzo permite dibujar en el documento HTML y actualizar dinámicamente estos dibujos, por medio de JavaScript.

También puede disparar acciones a partir de los eventos generados por el usuario.

Permite un funcionamiento similar al plugin de Flash, en lo que respecta a renderización de contenidos dinámicos.

```
<canvas id="micanvas" width="200" height="100">  
  Este texto se muestra para los navegadores no compatibles con canvas.  
<br>  
  Por favor, utiliza Firefox, Chrome, Safari u Opera.  
</canvas>
```

HTML5 y CSS3

HTML5

Para utilizar el canvas se debe referenciar primero el elemento canvas y adquirir su contexto.

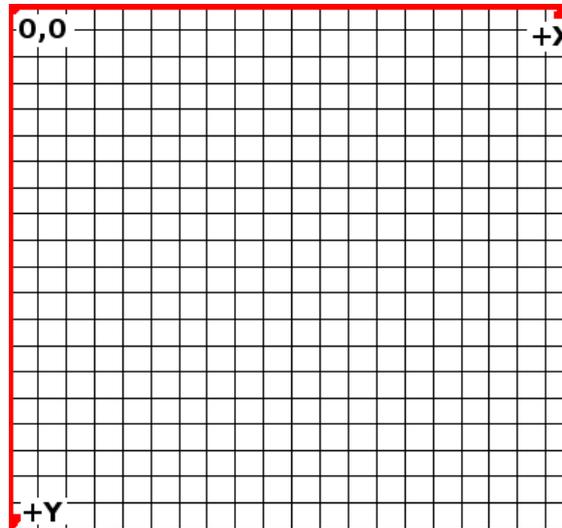
```
var canvas = document.getElementById('entorno_canvas');  
var context = canvas.getContext('2d');
```

Una vez adquirido, se puede empezar a dibujar en la superficie del canvas usando la API.

HTML5 y CSS3

HTML5

```
var canvas = document.getElementById('micanvas');  
//Accedo al contexto de '2d' de este canvas, necesario para  
dibujar  
var contexto = canvas.getContext('2d');  
//Dibujo en el contexto del canvas  
contexto.fillRect(50, 0, 10, 150);
```



HTML5 y CSS3

HTML5

beginPath

Le dice al contexto del canvas que se va a empezar a dibujarse un camino, no tiene ningún parámetro.

Una vez invocada la función se puede empezar a dibujar el camino añadiendo segmentos para completarlo con las diferentes funciones.

moveTo

Sirve para definir el punto donde se comienza a dibujar el segmento.

No dibuja nada, recibe como parámetro los puntos x e y donde ha de moverse el puntero para dibujo.

HTML5 y CSS3

HTML5

lineTo

Dibuja una línea recta desde la posición actual hasta el punto (x,y) que se indique como parámetro. La posición actual del camino se establece indicado previamente con un `moveTo()`, o donde se haya terminado el trazo anteriormente dibujado.

fill

Este método del contexto del canvas sirve para rellenar de color el área circunscrita por un camino.

Para rellenar de color un path, el path tiene que estar cerrado, por lo que, si no lo está, automáticamente se cerrará con una línea recta hasta el primer punto del camino, es decir, donde se comenzó a dibujar.

HTML5 y CSS3

HTML5

closePath

Sirve para cerrar un path, volviendo a su punto inicial de dibujo, no recibe ningún parámetro.

fillText

Permite dibujar texto en el canvas:

```
contexto.font = "bold 12px sans-serif";  
contexto.fillText("Hola", 5, 5);
```

stroke

Es similar al método fill pero traza solo la silueta y no la rellena.

HTML5 y CSS3

HTML5

Ejemplo

```
ctx.beginPath();  
ctx.moveTo(50,5);  
ctx.lineTo(75,65);  
ctx.lineTo(50,125);  
ctx.lineTo(25,65);  
ctx.closePath();  
ctx.fill();
```



HTML5 y CSS3

HTML5

Imágenes

Para dibujar una imagen, se usa la función `drawImage()`, con distintos parámetros según lo que se quiera hacer:

- `drawImage(img, x, y)`: dibuja la imagen `img` en `(x, y)`
- `drawImage(img, x, y, sw, sh)`: dibuja la imagen `img` en `(x, y)` después de establecer las dimensiones `sw` x `sh`.

HTML5 y CSS3

HTML5

Imágenes

Se pueden cargar de dos maneras:

- Usando una imagen cargada previamente con una etiqueta en el documento:

```

<script>
window.onload = function() {
    var canvas = document.getElementById("canvas");
    var context = canvas.getContext("2d");
    var img = document.getElementById("imagen");
    context.drawImage(img, 0, 0);
};
</script>
```

- Cargando la imagen completamente desde javascript:

```
var canvas = document.getElementById("e");
var context = canvas.getContext("2d");
var img = new Image();
img.src = "img/imagen.png"; //Necesita especificarse donde está la imagen
context.drawImage(img, 0, 0);
```

HTML5 y CSS3

HTML5

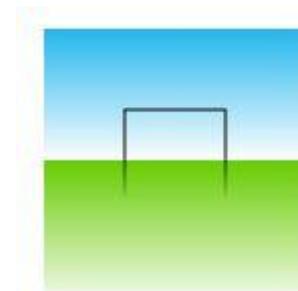
Gradientes

Son transiciones suaves entre dos colores:

- **createLinearGradient(x0, y0, x1, y1)**: pinta a lo largo de una dirección indicada por la línea que va desde el punto (x0, y0) hasta (x1, y1).
- **createRadialGradient(x0, y0, r0, x1, y1, r1)**: pinta a lo largo de un cono, con círculo inicial está en (x0, y0) con radio r0, y con círculo final en (x1, y1) con radio r1.

Se debe crear un objeto Gradient:

```
var my_gradient = contexto.createLinearGradient(0, 0, 300, 0);
```



HTML5 y CSS3

HTML5

Gradientes

Se deben añadir colores de parada en las posiciones del gradiente. Las posiciones van de 0 a 1.

```
my_gradient.addColorStop(0, "black");  
my_gradient.addColorStop(1, "white");
```

Para dibujar algo con gradiente, se asigna el objeto `gradient` a la propiedad `fillStyle`, y se dibuja algo con las técnicas vistas

```
Contexto.fillStyle = my_gradient;  
contexto.fillRect(0, 0, 50, 50);
```

HTML5 y CSS3

Audio y Video

HTML5 y CSS3

Audio

Handling audio events

There are many other events in the audio element. The following table lists a few commonly used audio events:

Event	Discussion
ended	Sent when the audio element finishes a playback
play	Sent when the audio element plays or resumes
pause	Sent when the audio element pauses
progress	Sent periodically when the audio element is downloading
timeupdate	Sent when the <code>currentTime</code> property changes

Here we just listed a few commonly used events; you can reference the complete audio event list in the Mozilla Developer Center at the following URL:

https://developer.mozilla.org/En/Using_audio_and_video_in_Firefox#Media_events

HTML5 y CSS3

Audio

```
<audio id="buttonactive">  
<source src="media/button_active.mp3" />  
<source src="media/button_active.ogg" />  
</audio>  
<script>
```

```
    document.getElementById("buttonactive").play();
```

```
</script>
```

The play function plays the audio from the elapsed time, which is stored in the currentTime property. The default value of currentTime is zero. The following code plays the audio from 3.5 seconds:

```
<script>
```

```
    document.getElementById("buttonactive").currentTime = 3.5;
```

```
    document.getElementById("buttonactive").play();
```

```
</script>
```

```
<script>
```

```
    document.getElementById("buttonactive").pause();
```

```
</script>
```

```
<script>
```

```
    document.getElementById("buttonactive").volume = 0.3;
```

```
</script>
```

HTML5 y CSS3

HTML5

Incluye la etiqueta `<video>`, `<audio/>`

- Sin restricciones para cualquier codec de audio/video.
- No hay ninguna combinación de un solo codec/contenedor que funcione en todos los navegadores.

Por problemas de soporte de navegadores, dentro de una etiqueta video/audio se pueden agregar múltiples src para distintos formatos y codecs

```
<video width="320" height="240" controls>  
  <source src="pr6.mp4" type='video/mp4; codecs="avc1.42E01E,mp4a.40.2"'>  
  <source src="pr6.webm" type='video/webm; codecs="vp8, vorbis"'>  
  <source src="pr6.ogv" type='video/ogg; codecs="theora, vorbis"'>  
</video>
```

HTML5 y CSS3

Web Storage

HTML5 y CSS3

HTML5

Sesión:

Período de tiempo continuo durante el cuál se comunica un navegador con una aplicación web. Durante este tiempo es posible compartir información entre el navegador y la aplicación.

Variable de sesión:

Estructura de datos común a un navegador y a una aplicación web donde se guarda información sobre la sesión. Esta estructura es nueva para cada ventana del navegador.

Cookie:

Es el mecanismo clásico que se ha utilizado hasta ahora para guardar, acceder y modificar información existente en una sesión, dentro de un archivo en el cliente.

HTML5 y CSS3

HTML5

Web Storage

HTML5 introduce dos mecanismos para almacenar información estructurada en el lado del cliente.

- `sessionStorage`
 - permite guardar información en el lado del cliente.
 - es un comportamiento similar a las variables de sesión.
- `localStorage`
 - permite guardar información sobre un sitio web.
 - esta información puede ser compartida entre ventanas y/o pestañas distintas. La información permanece aunque se termine la sesión.

HTML5 y CSS3

HTML5

Web Storage

A pesar de tener un comportamiento similar a las cookies, tiene varias ventajas adicionales:

- las cookies están limitadas a 4 KB de espacio, mientras que con `SessionStorage` se pueden guardar varios MB (depende de cada navegador).
- las cookies se envían en cada petición al servidor, lo cual aumenta la sobrecarga, mientras que la información guardada con `SessionStorage` no se envía automáticamente (aunque se puede configurar si es necesario).

HTML5 y CSS3

HTML5

Web Storage

Los sitios pueden agregar datos al objeto sessionStorage, y se podrá acceder desde cualquier pestaña del mismo sitio abierto en esa ventana.

```
sessionStorage.setItem('key', 'value');  
//Crea una nueva variable en el objeto sessionStorage  
sessionStorage.getItem('key');  
//Accede a la variable 'key'  
var a = sessionStorage.key; //Otra forma de acceso  
sessionStorage.removeItem('key'); //Eliminar variable
```

HTML5 y CSS3

HTML5

Web Storage

```
<body>
<script>
  function guardarEstado(){
    sessionStorage.setItem('state', document.getElementById('cb').checked);
  }
  function mostrarEstado(){
    if(sessionStorage.getItem('state') != null){
      alert(sessionStorage.getItem('state'));
    }else{
      alert('Haz click sobre el checkbox para guardar la información
sobre su estado');
    }
  }
</script>
<section>
<input type="checkbox" id="cb" onchange="guardarEstado()"/> Marca el checkbox
<br/>
<input type="button" onclick="mostrarEstado()" value="Info de SessionStorage"/>
</section>
</body>
```

HTML5 y CSS3

HTML5

Web Storage

El mecanismo LocalStorage de almacenamiento está diseñado para que el almacenamiento sea accesible desde ventanas distintas (que tengan abierto el mismo sitio web), y se prolongue más allá de la sesión actual.

- Las aplicaciones web pueden almacenar megabytes de datos.
- Las cookies no son aconsejables en este caso, ya que se transmiten con cada solicitud.
- El funcionamiento es el mismo que con sessionStorage, pero usando el objeto localStorage en su lugar.

HTML5 y CSS3

HTML5

Web Storge

```
<body>
<script>
  function guardarEstado(){
    localStorage.setItem('state', document.getElementById('cb').checked);
  }
  function mostrarEstado(){
    if(localStorage.getItem('state') != null){
      alert(localStorage.getItem('state'));
    }else{
      alert('Haz click sobre el checkbox para guardar la información
sobre su estado');
    }
  }
</script>
<section>
<input type="checkbox" id="cb" onchange="guardarEstado()"/> Marca el checkbox
<br>
<input type="button" onclick="mostrarEstado()" value="Info de LocalStorage"/>
</section>
</body>
```

HTML5 y CSS3

Drag & Drop

HTML5 y CSS3

HTML5

Drag & Drop

Es una característica, que permite poder arrastrar elementos de un lado a otro en la interfaz.

En HTML5 aparecen

- Nuevos eventos
dragstart, drag, dragenter, dragover,
dragleave, drop, dragend.
- Atributo `draggable="true"` para declarar que un elemento se puede arrastrar.
- Posibilidad de establecer la imagen "ghost" mostrada mientras se desplaza.
- Efectos asociados a copiar, mover...

HTML5 y CSS3

HTML5

Drag & Drop(Eventos)

- **dragstart:** Comienza el arrastrado. El "target" del evento será el elemento que está siendo arrastrado.
- **drag:** El elemento se ha desplazado. El "target" del evento será el elemento desplazado.
- **dragenter:** Se activa al entrar un elemento que se está arrastrando, dentro de un contenedor. El "target" del evento será el elemento contenedor.
- **dragleave:** El elemento arrastrado ha salido del contenedor. El "target" del evento será el elemento contenedor.

HTML5 y CSS3

HTML5

Drag & Drop (Eventos)

- **dragover**: El elemento ha sido movido dentro del contenedor. El "target" será el contenedor. Como el comportamiento por defecto es cancelar "drops", la función debe devolver false o llamar a preventDefault para indicar que se puede soltar dentro de ese contenedor.
- **drop**: El elemento arrastrado ha sido soltado en un contenedor. El "target" del elemento será el contenedor.
- **dragend**: Se ha dejado de arrastrar el elemento, se haya dejado en un contenedor o no. El "target" del evento es el elemento arrastrado.

HTML5 y CSS3

HTML5

Drag & Drop

Para utilizar Drag & Drop:

- Definir un objeto como "arrastrable", estableciendo su propiedad `draggable="true"` (por defecto "true" en imágenes).
- Definir el comportamiento adecuado cuando se detecta un evento relacionado con Drag & Drop:

HTML5 y CSS3

HTML5

Drag & Drop

```
<body>
<div class="area" align="center">
<span>Elige un objeto para arrastrar</span><br/>


</div>
<div class="area" align="center">
<span>Suelta el objeto aquí</span>
<div id="lista"></div>
</div>
</body>
```

HTML5 y CSS3

HTML5

Drag & Drop

```
<script>
  document.getElementById('lista').ondragover = anyadirObjecto;
  document.getElementById('rectangulo').ondragstart = empezar;
  document.getElementById('triangulo').ondragstart = empezar;
  document.getElementById('lista').ondrop = soltar;
</script>
```

HTML5 y CSS3

HTML5

Drag & Drop

```
<script>
  function empezar(e){
    e.dataTransfer.setData('Text', this.id);
    e.dataTransfer.effectAllowed = 'move';
  }
  function anyadirObjeto(e) {
    e.dataTransfer.dropEffect = 'move';
    return false;
  }
  function soltar (e) {
    imagen = new Image();
    imagen.src = e.dataTransfer.getData('Text') + '.jpg';
    document.getElementById('lista').appendChild(imagen);
  }
</script>
```

HTML5 y CSS3

HTML5

Drag & Drop

El elemento a mover, tiene su propiedad `draggable` a 'true'.

El contenedor puede recibir información de los elementos arrastrados desde otros navegadores u otras aplicaciones.

Si utilizamos `setData` y `getData` del objeto `dataTransfer` expuesto en el objeto del evento (`Event Object`).

HTML5 y CSS3

HTML5

Drag & Drop

Tipos para almacenar la Información:

- Texto: Utilizamos *text/plain*.

```
event.dataTransfer.setData("text/plain", "Este es el texto a arrastrar");
```

- Link: Utilizamos *text/plain* o *text/uri-list*.

```
event.dataTransfer.setData("text/plain", "http://www.online.imaginaformacion.com");
```

```
event.dataTransfer.setData("text/uri-list", "http://www.online.imaginaformacion.com");
```

- HTML/XML: Utilizamos *text/plain*, *text/xml* o *text/html*.

```
event.dataTransfer.setData("text/html", "Hola <strong>alumnos</strong>");
```

```
event.dataTransfer.setData("text/plain", "Hola alumnos");
```

HTML5 y CSS3

HTML5

Drag & Drop

Arrastrando archivos

- Un archivo local es arrastrado con el tipo *application/x-moz-file*.
- Las páginas web sin ciertos privilegios, no son capaces de recuperar o modificar datos de este tipo.
- Debido a que un archivo no es una cadena, debe utilizar el método `mozSetDataAt` (en firefox) para asignar los datos.
- Del mismo modo, cuando se recuperan los datos, debe utilizar el método `mozGetDataAt` (en firefox).

```
event.dataTransfer.mozSetData("application/x-moz-file", file, 0);
```

HTML5 y CSS3

HTML5

Drag & Drop

Imágenes:

La mayoría de los navegadores no soportan actualmente arrastrar imágenes.

Por lo general lo que se arrastra es la url de la propia imagen.

Se Utiliza text/plain o text/uri-list.

```
event.dataTransfer.setData("text/plain", imagenURL);  
event.dataTransfer.setData("text/uri-list", imagenURL);
```

En chrome podemos utilizar:

```
event.dataTransfer.setData("image/png", stream, 0);
```

Para saber más acerca de la url de Datos,
https://developer.mozilla.org/en/data_URIs

HTML5 y CSS3

Sprite CSS

HTML5 y CSS3

HTML5

Sprite CSS

Permite ir recopilando las imágenes de la web una a una y después colocarlas todas en un png.

Después desde CSS se escriben las coordenadas de todas y cada una de las imágenes.

Sprite Me, es un programa que se encarga de esta engorrosa tarea por nosotros.

Utiliza menos ancho de banda, ya que con una sola HTTP-Request puede cargar todas las imágenes desde el servidor.

Cada `` o `background-image` es una `HTTP_Request`.

HTML5 y CSS3

HTML5

Sprite CSS

Antes

```
#nav li a {background:none no-repeat left center}  
#nav li a.item1 {background-image:url('../img/img1.jpg')}  
#nav li a:hover.item1 {background-image:url('../img/img1.jpg')}  
#nav li a.item2 {background-image:url('../img/img2.jpg')}  
#nav li a:hover.item2 {background-image:url('../img/img5.jpg')}  
...
```

The diagram illustrates the 'Antes' (Before) state of a website. On the left, five individual image files are shown: img1.jpg, img2.jpg, img3.jpg, img4.jpg, and img5.jpg. In the center, a vertical navigation menu lists five items: Item 1, Item 2, Item 3, Item 4, and Item 5, each with a corresponding icon. On the right, a grey box titled 'ANTES' provides a summary: 'Numero de HTTP-Request' is 10, and 'Tamaño total de las imagenes' is 20,5KB.

Item	Icon
Item 1	img1.jpg
Item 2	img3.jpg
Item 3	img4.jpg
Item 4	img2.jpg
Item 5	img5.jpg

ANTES

Numero de HTTP-Request
10

Tamaño total de las imagenes
20,5KB

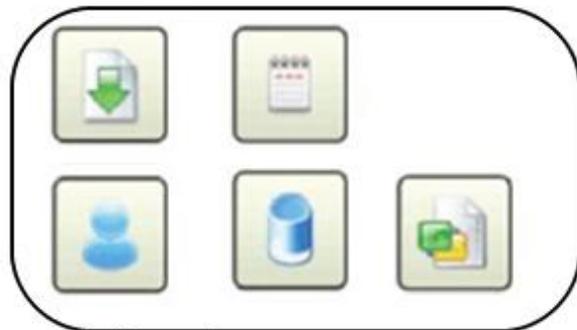
HTML5 y CSS3

HTML5

Sprite CSS

Ahora

```
#nav li a {background-image:url('../img/SpriteImage.jpg')}  
#nav li a.item1 {background-position:0px 0px}  
#nav li a:hover.item1 {background-position:0px -72px} #nav li  
a.item2 {background-position:0px  
-143px;}  
#nav li a:hover.item2 {background-position:0px -215px;}
```



SpriteImage.jpg



DESPUES

Numero de HTTP-Request

1

Tamaño total de las imagenes

13 KB

HTML5 y CSS3

CSS3

HTML5 y CSS3

CSS3

CSS3 es totalmente compatible con HTML5 y sus nuevas etiquetas.

Los nuevos selectores dan más flexibilidad a la hora de seleccionar unos u otros elementos.

Selectores

E:disabled /*Elemento que esta desactivado*/

E:checked /*Elemento es que esta marcado (radiobutton, checkbox)*/

E[foo^="bar"]/*El que el valor del atributo “foo” de E comienza con la cadena “bar”.*/

E[foo\$="bar"]/*El valor del atributo “foo” de E acaba exactamente con la cadena “bar”.*/

E[foo*="bar"]/*el atributo “foo” de E contiene la cadena “bar”.*/

Muchos selectores más...

HTML5 y CSS3

CSS3

Border radius

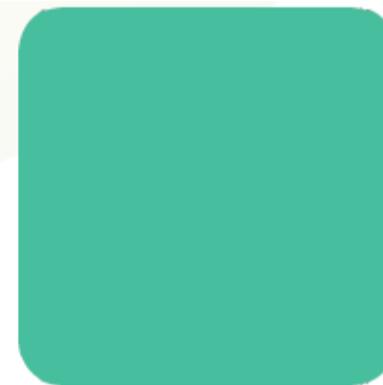
Permite crear cajas con bordes redondeados fácilmente.

Para dotar mayor compatibilidad se usan los prefijos `-webkit-` y `-moz-` para los navegadores basados en Webkit (Safari y Chrome) y los de Mozilla (Firefox).

```
-webkit-border-radius: 4px;
```

```
-moz-border-radius: 4px;
```

```
border-radius: 4px;
```



HTML5 y CSS3

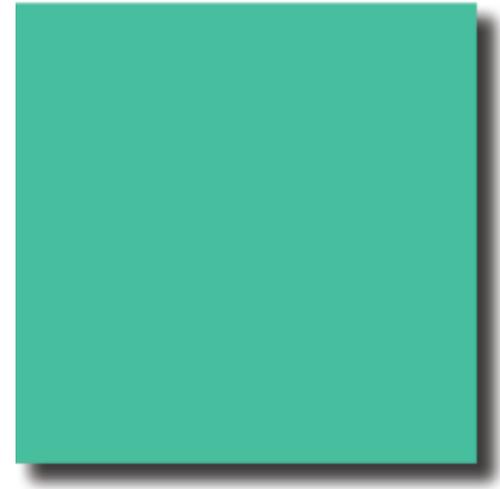
CSS3

Box Shadow

Ofrece la posibilidad de añadir una sombra a nuestra caja.
Admite cuatro parámetros:

- desplazamiento x
- desplazamiento y
- desenfoque
- color de la sombra

```
-webkit-box-shadow: 1px 1px 3px #292929;  
-moz-box-shadow: 1px 1px 3px #292929;  
box-shadow: 1px 1px 3px #292929;
```



HTML5 y CSS3

CSS3

Text shadow

Ofrece la posibilidad de añadir una sombra al texto.

Admite cuatro parámetros:

- desplazamiento x
- desplazamiento y
- desenfoque
- color de la sombra

```
-webkit-box-shadow: 1px 1px 3px #292929;  
-moz-box-shadow: 1px 1px 3px #292929;  
box-shadow: 1px 1px 3px #292929;
```



hello

HTML5 y CSS3

CSS3

Multiple background

Permite agregar varios fondos de forma simultánea.
Tiene las mismas propiedades que background.
(varias direcciones de imágenes).

```
/* modern browsers */  
background: url(image/path.jpg) 0 0 no-repeat,  
url(image2/path.jpg) 100% 0 no-repeat;
```

HTML5 y CSS3

CSS3

Background size

Permite variar el tamaño del background.
Acepta dos parámetros anchura de x e y.

```
body {  
    background: url(path/to/image.jpg) no-repeat;  
    -moz-background-size: 100% 100%;  
    -o-background-size: 100% 100%;  
    -webkit-background-size: 100% 100%;  
    background-size: 100% 100%;  
}
```

@font-FACE

Permite agregar cualquier tipografía local o desde un servidor de Fonts.

```
@font-face {  
    font-family: Gentium;  
    src: url(http://site/fonts/Gentium.ttf);  
}
```

HTML5 y CSS3

CSS3

Gradientes

W3C Spec

```
linear-gradient([<bg-position> || <angle>,<stop>,<color-stop>[, <color-stop>]*);
```

Firefox 3.6

```
-moz-linear-gradient( [<point> || <angle>,<stop>,<stop>[, <stop>]* )
```

Safari 4 Chrome

```
-webkit-gradient(<type>,<point> [, <radius>]?,<point> [, <radius>]? [, <stop>]*)
```

Internet Explorer 5+

```
filter:progid:DXImageTransform.Microsoft.Gradient(sProperties)
```

HTML5 y CSS3

CSS3

Gradientes

Ejemplos:

Stops de multiples colores



```
background-image: -webkit-gradient(linear, left top, left bottom,  
color-stop(0, red), color-stop(0.28, orange),  
color-stop(.42, yellow), color-stop(.56, green),  
color-stop(.7, blue), color-stop(.84, indigo),  
color-stop(1, violet));
```

```
background-image: -moz-linear-gradient(top, red, orange, yellow,  
green, blue, indigo, violet);
```

HTML5 y CSS3

CSS3

Pseudo Clases

Se utilizan para agregar estilo a elementos HTML, cuando una determinada condición se cumple. La condición depende de la pseudo-clase que se añade al selector.

Las pseudo-clases se indican con: un selector (identificador de clase, selector de atributos, etc) y a continuación, seguido de dos puntos (:) se escribe la pseudo-clase.

:hover, se activa cuando el usuario pasa el ratón o cualquier otro elemento apuntador por encima de un elemento.

:active, se activa cuando el usuario activa un elemento, por ejemplo cuando pulsa con el ratón sobre un elemento. El estilo se aplica durante un espacio de tiempo prácticamente imperceptible, ya que sólo dura desde que el usuario pulsa el botón del ratón hasta que lo suelta.

:focus, se activa cuando el elemento tiene el foco del navegador, es decir, cuando el elemento está seleccionado. Normalmente se aplica a los elementos `<input>` de los formularios cuando están activados y por tanto, se puede escribir directamente en esos campos.

HTML5 y CSS3

CSS3

Pseudo Clases

[:first-child](#), selecciona el primer elemento hijo de un elemento.

[:link](#), se aplica a todos los enlaces que todavía no han sido visitados por el usuario.

[:visited](#), se aplica a todos los enlaces que han sido visitados al menos una vez por el usuario.

Más...

HTML5 y CSS3

Ejercicios

1. Utilizar la aplicación spriteme.org y/o <http://spritegen.website-performance.org/>
2. Probar la generación de texto en <http://www.blindtextgenerator.com/es>
3. Leer acerca del uso de variables en css3 en <http://www.w3.org/TR/css-variables/> y en <http://www.inserthtml.com/2012/02/css-variables/>

Probar las herramientas:

<http://www.spritebox.net/> , <http://wearekiss.com/spritepad>

HTML5 y CSS3

Fin