



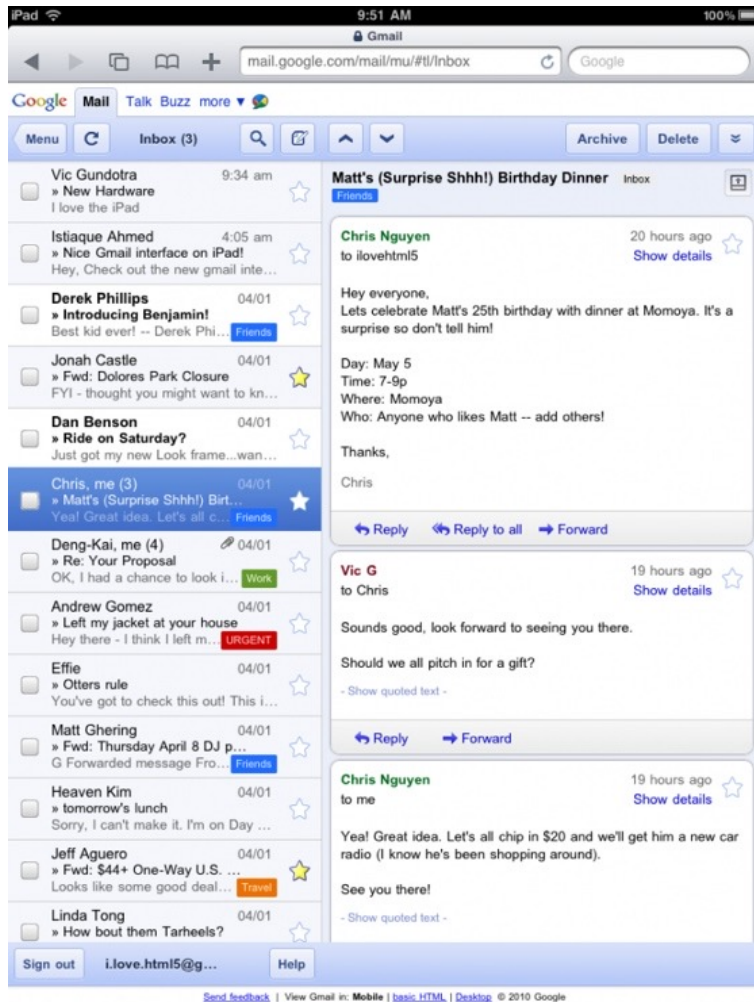
HTML5 y el Futuro de la Web

¿Qué es HTML5?

- Estándar abierto y accesible
- Elementos nuevos
- Web Semántica
- API's de desarrollo
- Retrocompatible
- Nueva filosofía de trabajo
- Impulsado por los grandes y pequeños



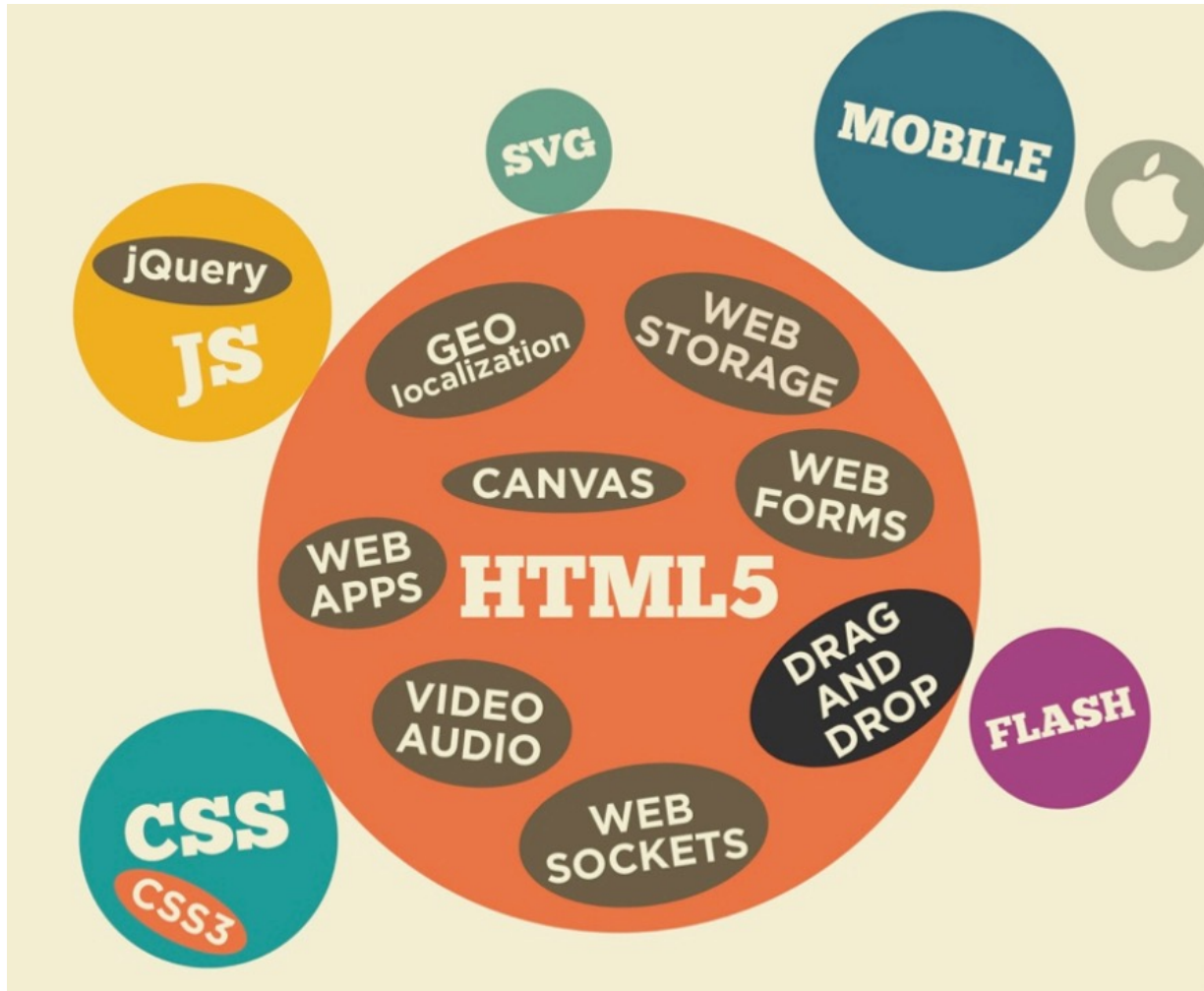
RIA



Y Flash?



¿Qué es HTML5?



Trabaja sobre

- Firefox 3.5+
- Chrome 3.0+
- Safari 3.1+
- Opera 10.5+
- IE 9.0+
- iPhone
- iPod Touch
- iPad
- Android OS

<http://html5test.com/>

Estructura mínima de un documento

```
<!DOCTYPE html>  
<html lang="ES">  
  <head>  
    <title>Este es un ejemplo de HTML5</title>  
    <meta charset="UTF-8">  
    <link rel="stylesheet" href="estilo.css">  
  </head>  
  <body>Este es el cuerpo de la página</body>  
</html>
```

Tipo de documento

- Tipo de Documento

<!DOCTYPE html>

- Lenguaje del Documento

<html lang="ES">

- Codificación del Documento

<meta charset="UTF-8">

- Enlaces a CSS, Favicon y RSS

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

<script src=jquery.js></script>

Estructura semántica

HTML



Estructura semántica

`<header>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`

Estructura semántica

<header> El elemento header debe utilizarse para marcar la cabecera de una página (contiene el logotipo del sitio, una imagen, un cuadro de búsqueda etc).

El elemento header puede estar anidado en otras secciones de la página (es decir que no solo se utiliza para la cabecera de la página).

<nav> El elemento nav representa una parte de una página que enlaza a otras páginas o partes dentro de la página. Es una sección con enlaces de navegación.

No todos los grupos de enlaces en una página deben ser agrupados en un elemento nav. únicamente las secciones que consisten en bloques de navegación más importantes son adecuados para el elemento de navegación.

Estructura semántica

<section> El elemento section representa un documento genérico o de la sección de aplicación. Una sección, en este contexto, es una agrupación temática de los contenidos. Puede ser un capítulo, una sección de un capítulo o básicamente cualquier cosa que incluya su propio encabezado.

Una página de inicio de un sitio Web puede ser dividida en secciones para una introducción, noticias, información de contacto etc.

<footer> El elemento footer se utiliza para indicar el pie de la página o de una sección. Un pie de página contiene información general acerca de su sección el autor, enlaces a documentos relacionados, datos de derechos de autor etc.

Estructura semántica

<aside> El elemento aside representa una nota, un consejo, una explicación. Esta área son representados a menudo como barras laterales en la revistas impresa.

El elemento puede ser utilizado para efectos de atracción, como las comillas tipográficas o barras laterales, para la publicidad, por grupos de elementos de navegación, y por otro contenido que se considera por separado del contenido principal de la página.

<article> El elemento article representa una entrada independiente en un blog, revista, periódico etc.

Cuando se anidan los elementos article, los artículos internos están relacionados con el contenido del artículo exterior. Por ejemplo, una entrada de blog en un sitio que acepta comentarios, el elemento article principal agrupa el artículo propiamente dicho y otro bloque article anidado con los comentarios de los usuario.

Estructura semántica

CONSEJOS PARA VIAJAR EN EL TIEMPO

Consejos Paradojas Contacto

¡No to

El Viajero a través del Tiempo (pues convendrá llamarle así al hablar de él) nos exponía una misteriosa cuestión. Sus ojos grises brillaban lanzando centellas, y su rostro, habitualmente pálido, mostrábase encendido y animado.

¡Quiero más consejos!

NAV

ARTICLE

ARTICLE

VIDEO

FIGURE

FIGURE

La línea temporal

El Condensador de Fluzo

El fuego ardía fulgurante y el suave resplandor de las lámparas incandescentes, en forma de lirios de plata, se prendía en las burbujas que destellaban y subían dentro de nuestras copas

La realidad alternativa

El fuego ardía fulgurante y el suave resplandor de las lámparas incandescentes, en forma de lirios de plata, se prendía en las burbujas que destellaban y subían dentro de nuestras copas

Efectos de las paradojas

El fuego ardía fulgurante y el suave resplandor de las lámparas incandescentes, en forma de lirios de plata, se prendía en las burbujas que destellaban y subían dentro de nuestras copas

¿Listo para viajar en el

Vehículo:

8

Buscar viajes

FOOTER

SECTION

HEADER

NAV

SECTION

ASIDE

Estructura semántica: Otros

Mark: Mi nombre es `<mark>Pedro</mark>`

Figure:

**Address
Time**

```
<figure>  
    
</figure>
```


Resumen



Estructura semántica: Ejercicio

Generar un documento HTML utilizando las estructuras semánticas básicas.

Ejemplos

<http://twitter.github.com/bootstrap/>

<http://coding.smashingmagazine.com/2009/08/04/designing-a-html-5-layout-from-scratch/>

http://www.webmonkey.com/2010/02/building_web_pages_with_html_5/

<http://tutorialzine.com/2010/02/html5-css3-website-template/>

Controles de formulario

HTML



Nuevos inputs

- **Telephone**
- **Url**
- **Email**
- **Password**
- **Datetime**
- **Date**
- **Month**
- **Week**
- **Number**
- **Range**
- **Color**

Nuevos inputs

```
<input type="color" name="favcolor" />
```

```
<input type="date" name="bday" />
```

```
<input type="datetime" name="bdaytime" />
```

```
<input type="datetime-local" name="bdaytime" />
```

```
<input type="email" maxlength="10" size="5"  
name="usremail" />
```

```
<input type="month" name="bdaymonth" />
```

```
<input type="number" name="quantity" min="1" max="5" />
```

```
<input type="range" name="points" min="1" max="10" />
```

```
<input type="search" name="googlesearch" />
```

```
<input type="time" name="usr_time" />
```

```
<input type="week" name="week_year" />
```

Nuevos inputs

Datalist:

```
<input list="browsers" />
```

```
<datalist id="browsers">
```

```
<option value="Internet Explorer">
```

```
<option value="Firefox">
```

```
<option value="Chrome">
```

```
<option value="Opera">
```

```
<option value="Safari">
```

```
</datalist>
```

Nuevos inputs

Datalist:

```
<input list="browsers" />
```

```
<datalist id="browsers">
```

```
<option value="Internet Explorer">
```

```
<option value="Firefox">
```

```
<option value="Chrome">
```

```
<option value="Opera">
```

```
<option value="Safari">
```

```
</datalist>
```

Nuevos inputs

Output:

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0  
  <input type="range" name="a" value="50" />100  
  +<input type="number" name="b" value="50" />  
  =<output name="x" for="a b"></output>  
</form>
```


Form attributes

Autocomplete

```
<form action="demo_form.jsp" autocomplete="on">  
  First name:<input type="text" name="fname" /><br />  
  Last name: <input type="text" name="lname" /><br />  
  E-mail: <input type="email" name="email"  
autocomplete="off" /><br />  
  <input type="submit" />  
</form>
```

Form attributes

Novalidate

```
<form action="demo_form.jsp" novalidate="novalidate">  
  E-mail: <input type="email" name="user_email" />  
  <input type="submit" />  
</form>
```

Form attributes

Novalidate

```
<form action="demo_form.jsp" novalidate="novalidate">  
  E-mail: <input type="email" name="user_email" />  
  <input type="submit" />  
</form>
```

Form attributes

Form

```
<form action="demo_form.jsp" id="form1">
```

```
First name: <input type="text" name="fname" /><br />
```

```
<input type="submit" value="Submit" />
```

```
</form>
```

```
Last name: <input type="text" name="lname" form="form1" />
```

Form attributes

Formaction

```
<form action="demo_form.jsp">
```

```
First name: <input type="text" name="fname" /><br />
```

```
Last name: <input type="text" name="lname" /><br />
```

```
<input type="submit" value="Submit" /><br />
```

```
<input type="submit" formaction="demo_admin.jsp"  
value="Submit as admin" />
```

```
</form>
```

Form attributes

Formenctype

```
<form action="demo_post_enctype.jsp" method="post">
```

```
First name: <input type="text" name="fname" /><br />
```

```
<input type="submit" value="Submit" />
```

```
<input type="submit" formenctype="multipart/form-data"  
value="Submit as Multipart/form-data" />
```

```
</form>
```

Form attributes

Formaction

```
<form action="demo_form.jsp" method="get">  
First name: <input type="text" name="fname" /><br />  
Last name: <input type="text" name="lname" /><br />  
<input type="submit" value="Submit" />  
<input type="submit" formmethod="post"  
formaction="demo_post.jsp" value="Submit using POST" />  
</form>
```

Form attributes

Formnovalidate

```
<form action="demo_form.jsp">  
E-mail: <input type="email" name="userid" /><br />  
<input type="submit" value="Submit" /><br />  
<input type="submit" formnovalidate="formnovalidate"  
value="Submit without validation" />  
</form>
```


Form attributes

Formtarget

```
<form action="demo_form.jsp">
```

```
First name: <input type="text" name="fname" /><br />
```

```
Last name: <input type="text" name="lname" /><br />
```

```
<input type="submit" value="Submit as normal" />
```

```
<input type="submit" formtarget="_blank" value="Submit to a  
new window" />
```

```
</form>
```

Form attributes

Image X Y

```
<form action="demo_form.asp">
```

```
  First name: <input type="text" name="fname" /><br />
```

```
  Last name: <input type="text" name="lname" /><br />
```

```
  <input type="image" src="img_submit.gif" alt="Submit"
width="48" height="48"/>
```

```
</form>
```

Form attributes

Max & Min

```
<input type="date" name="bday" max="1979-12-31">
```

```
<input type="date" name="bday" min="2000-01-02">
```

```
<input type="number" name="quantity" min="1" max="5" />
```

Form attributes

Multi file

```
<input type="file" name="img" multiple="multiple" />
```

Form attributes

Pattern

```
<input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code" />
```

Form attributes

Placeholder

```
<input type="text" name="fname" placeholder="First name" />
```

Form attributes

Required

```
<input type="text" name="username" required="required" />
```

Form attributes

Steps

```
<input type="number" name="points" step="3" />
```


Formularios: Ejercicio

Generar un formulario (todos los campos requeridos) respetando los tipos de datos según corresponda.

Nombre

Apellido

Email

Web site

Nro. de documento

Monto premio

Fecha de nacimiento

Storage

HTML



Storage

SessionStorage

LocalStorage

Web Db / Index Db

Local Cache

Storage

SessionStorage

- Permite almacenar información en el navegador siempre y cuando nos mantengamos en la misma ventana o tab.
- La información no es persistente
- Tamaño max 5mb

Storage

LocalStorage

- Permite almacenar información en el navegador x cada sitio
- La información es persistente
- Tamaño max 5mb

Storage

localStorage.setItem("size", "6");

var size = localStorage.getItem("size");

var size = localStorage["size"];

```
interface Storage {  
    readonly attribute unsigned long length;  
    DOMString key(in unsigned long index);  
    getter any getItem(in DOMString key);  
    setter creator void setItem(in DOMString key, in any value);  
    deleter void removeItem(in DOMString key);  
    void clear();  
};
```

Storage

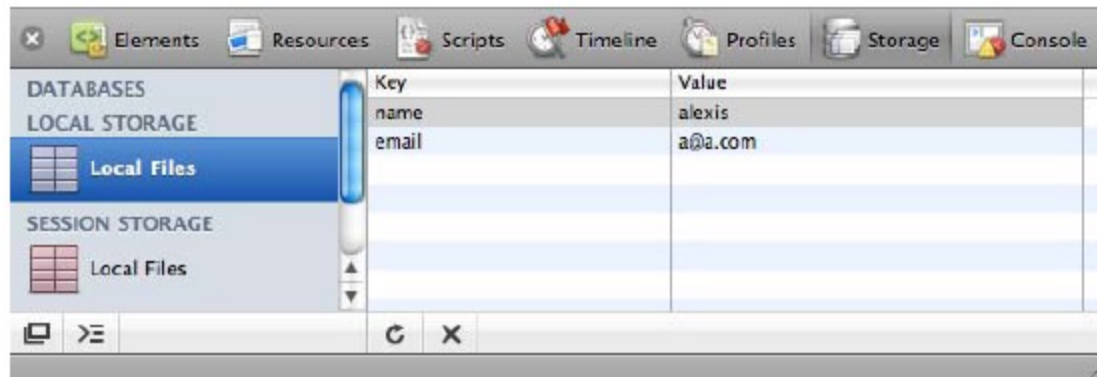
sessionStorage.setItem("size", "6");

var size = sessionStorage.getItem("size");

var size = sessionStorage["size"];

```
interface Storage {  
    readonly attribute unsigned long length;  
    DOMString key(in unsigned long index);  
    getter any getItem(in DOMString key);  
    setter creator void setItem(in DOMString key, in any value);  
    deleter void removeItem(in DOMString key);  
    void clear();  
};
```

Storage



The screenshot shows a web browser's developer console with the 'Storage' tab selected. The left sidebar is divided into 'DATABASES', 'LOCAL STORAGE', and 'SESSION STORAGE'. Under 'LOCAL STORAGE', 'Local Files' is selected. The main area displays a table of stored data.

| Key | Value |
|-------|---------|
| name | alexis |
| email | a@a.com |
| | |
| | |
| | |
| | |

Ejercicio

Modificar el formulario para que guarde la información directamente en el navegador. Opcional utilizar JSON parse / stringify

Geolocation

HTML



Geolocation

Recupera la lng y lat mediante:

- IP
- Wifi
- GPS
- Red celular

Geolocation

The `getCurrentPosition()` Method - Return Data

The `getCurrentPosition()` method returns an object if it is successful. The latitude, longitude and accuracy properties are always returned. The other properties below are returned if available.

| Property | Description |
|--------------------------------------|---|
| <code>coords.latitude</code> | The latitude as a decimal number |
| <code>coords.longitude</code> | The longitude as a decimal number |
| <code>coords.accuracy</code> | The accuracy of position |
| <code>coords.altitude</code> | The altitude in meters above the mean sea level |
| <code>coords.altitudeAccuracy</code> | The altitude accuracy of position |
| <code>coords.heading</code> | The heading as degrees clockwise from North |
| <code>coords.speed</code> | The speed in meters per second |
| <code>timestamp</code> | The date/time of the response |

Ejemplo

```
<script>
var x=document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML="Geolocation is not supported.";
  }
}
function showPosition(position) {
  x.innerHTML="Latitude: " + position.coords.latitude +
  "<br />Longitude: " + position.coords.longitude;
}
</script>
```

Ejemplo

```
var x=document.getElementById("demo");  
  
function getLocation()  
{  
  if (navigator.geolocation)  
  {  
    navigator.geolocation.getCurrentPosition(showPosition,showError);  
  }  
  else{x.innerHTML="Geolocation is not supported by this browser.";}  
}  
  
function showPosition(position)  
{  
  x.innerHTML="Latitude: " + position.coords.latitude +  
  "<br />Longitude: " + position.coords.longitude;  
}
```

```
function showError(error)  
{  
  switch(error.code)  
  {  
    case error.PERMISSION_DENIED:  
      x.innerHTML="User denied the request for Geolocation."  
      break;  
    case error.POSITION_UNAVAILABLE:  
      x.innerHTML="Location information is unavailable."  
      break;  
    case error.TIMEOUT:  
      x.innerHTML="The request to get user location timed out."  
      break;  
    case error.UNKNOWN_ERROR:  
      x.innerHTML="An unknown error occurred."  
      break;  
  }  
}
```

Ejercicio

Generar un mapa con google maps que muestre la posición actual. Integrar la misma con el ejemplo del formulario.

WebSocket

HTML



WebSocket

- + rápido que Ajax
- Protocolo propio ws://
- Trabaja con firewalls y proxys
- Nuevo modelo de programación

WebSocket

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web.

- Útil para
- Aplicaciones en tiempo real
- Chats
- Resultados deportivos
- Notificaciones
- Etc

WebSocket

```
var ws = new WebSocket("ws://localhost:9998/echo");
```

```
ws.onopen = function() {  
    ws.send("Hola Mundo");  
    alert("enviando msg...");  
};
```

```
ws.onmessage = function (evt) {  
    var received_msg = evt.data;  
    alert("Llego algo...");  
};
```

```
ws.onclose = function() {  
    alert("Cerrando el socket...");  
};
```

Ejercicio

Utilizando el ejercicio de formularios agregar un ws que notifique mediante un mensaje al momento de guardar la información.

Offline

HTML



Offline cached applications

Permite navegar una aplicación web 100% offline (sin tener internet).

<html manifest="example.manifest">

Cache manifest

CACHE MANIFEST

#This is a comment

CACHE

index.html

style.css

image.jpg

image-med.jpg

image-small.jpg

notre-dame.jpg

Cache manifest

```
<html manifest="someapp.manifest">
```

```
...
```

```
</html>
```



CACHE MANIFEST

#v1.01

#Explicitly cached files

CACHE:

index.html

Stylesheet.css

Images/logo.png

NETWORK:

Search.cfm

Login.cfm

/dynamicpages

FALLBACK:

/dynamicpage.cfm /static.html

Offline cached applications

```
if (navigator.onLine) {  
    alert('online')  
} else {  
    alert('offline');  
}
```

```
window.addEventListener("offline", function(e) {  
    alert("offline");  
}, false);
```

```
window.addEventListener("online", function(e) {  
    alert("online");  
}, false);
```

Ejercicio

Realizar una pagina web y probar las características de la navegación Offline

History API

HTML



History API

Permite gestionar el botón de siguiente y anterior en los casos donde trabajamos con AJAX.

```
history.pushState(stateObject, title, url);
```

```
history.pushState({foo: 'bar'}, 'Title', '/baz.html')
```

```
history.length
```

```
history.forward()
```

```
history.back()
```

```
history.go(-3)
```

History API

```
window.onpopstate = function(event) {  
  
    alert("location: " + document.location + ", state: " +  
    JSON.stringify(event.state));  
  
};
```

```
history.pushState({page: 1}, "title 1", "?page=1");  
history.pushState({page: 2}, "title 2", "?page=2");
```

Ejercicio

Realizar un conjunto de llamadas ajax con JQuery y probar el History API

Web Workers

HTML



Web Workers

Los Web Workers proveen un medio sencillo para que el contenido web ejecute scripts en threads (hilos) en segundo plano.

```
var worker = new Worker('worker.js');  
worker.postMessage("Información para el worker");  
worker.terminate();
```


Controles de audio y video

HTML



<video> y <audio>

- Permiten reproducir audio y video libremente
- Video formats: MP4, OGV, FLV, WebM, AVI
- Audio formats: MP3, AAC

<video>

- **autoplay** Para que el video se reproduzca automáticamente. Puede tener valor *TRUE* o *FALSE*.
- **controls** Para que se muestre barra de control debajo del video.
- **height** Altura del video.
- **width** Ancho del video.
- **loop** El video se reproduce de manera repetitiva.
- **poster** Imagen si el video no está disponible.
- **preload** Controla si el video es precargado. Puede tener 3 valores: none, metadata ó auto.
- **src** Indica la url del video.

Antes de <video>

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="425"
height="344"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=6,0,40,0"> <param name="allowFullScreen" value="true" /> <param
name="allowscriptaccess" value="always" /> <param name="src"
value="http://www.youtube.com/v/oHg5SJYRHA0&hl=en&fs=1&" /> <param
name="allowfullscreen" value="true" /> <embed type="application/x-shockwave-
flash" width="425" height="344"
src="http://www.youtube.com/v/oHg5SJYRHA0&hl=en&fs=1&"
allowscriptaccess="always" allowfullscreen="true"> </embed> </object>
```

Con <video>

```
<video width="640" height="360"  
  src="http://www.youtube.com/demo/google_main.mp4"  
  controls autobuffer>  
</video>
```

<video>

```
<video id="movie" width="320" height="240" preload controls>
  <source src="pr6.mp4" />
</video>
<script>
  var v = document.getElementById("movie");
  v.onclick = function() {
    if (v.paused) {
      v.play();
    } else {
      v.pause();
    }
  });
</script>
```

<audio>

<audio src="audiohola.ogg" controls="controls"></audio>

- autoplay**
- controls**
- loop**
- preload**
- src**

Canvas

HTML



Canvas

Canvas es un elemento incorporado en HTML5 que permite la generación de gráficos dinámicamente por medio del scripting.

- Generar gráficos vectoriales
- Animaciones 2d y 3d
- Juegos

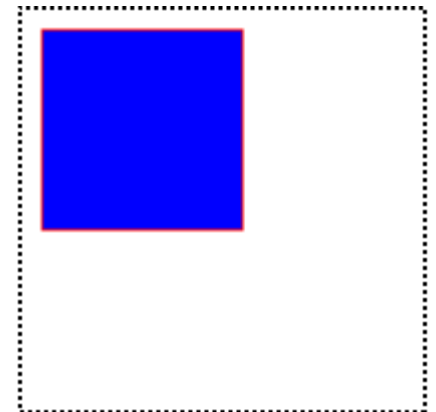
Canvas

`<canvas id="miCanvas"></canvas>`

`<script>`

```
function draw(){  
    var canvas=document.getElementById('myCanvas');  
    var context=canvas.getContext('2d');  
    context.strokeStyle = "red";  
    context.fillStyle = "blue";  
    context.fillRect(10,10,100,100);  
    context.strokeRect(10,10,100,100);  
}
```

`</script>`



Canvas

<canvas id="miCanvas"></canvas>

```
<script type="text/javascript">
```

```
var c=document.getElementById( "myCanvas" );  
var ctx=c.getContext( "2d" );  
var grd=ctx.createLinearGradient(0,0,175,50);  
grd.addColorStop(0,"#FF0000");  
grd.addColorStop(1,"#00FF00");  
ctx.fillStyle=grd;  
ctx.fillRect(0,0,175,50);
```

```
</script>
```



Otros

HTML



Otros

AttrData

Drag and Drop

Notifications

File

XMLHTTP2

HTML5 + CSS3