



HTML 5



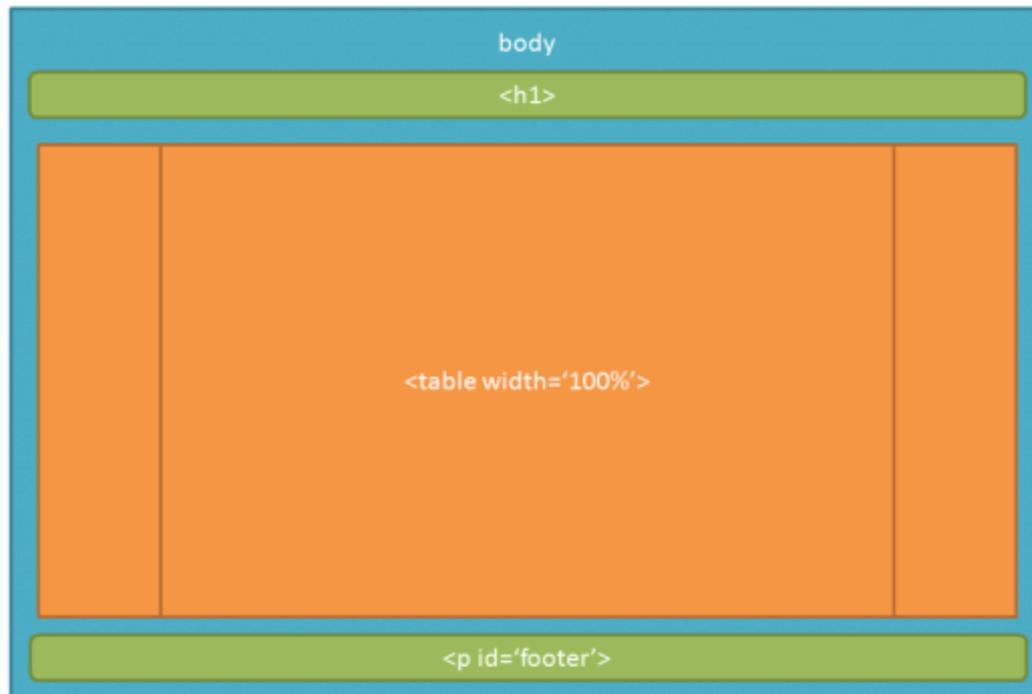
HTML5, W3C

qué falta para el HTML 5

- ▶ **HTML 5** es la nueva versión, tras **más de 10 años**, de la forma en la que se construyen las páginas de Internet, y uno de los responsables de esta nueva versión es uno de los trabajadores de Google.
- ▶ Con este nuevo formato podremos conectarnos a dispositivos externos, geolocalizar a los usuarios, cachear información en el navegador, el AJAX se mejorará, los formularios aumentan la capacidad de la Web 2.0...
- ▶ Todos los navegadores soportan HTML 5, pero es posible usarlo en versiones anteriores con una línea de JavaScript y un par en los CSS, activando todo su potencial.
- ▶ Los buscadores como Google o Bing ya son capaces de interpretar mayormente los elementos más interesantes para SEO del HTML 5.

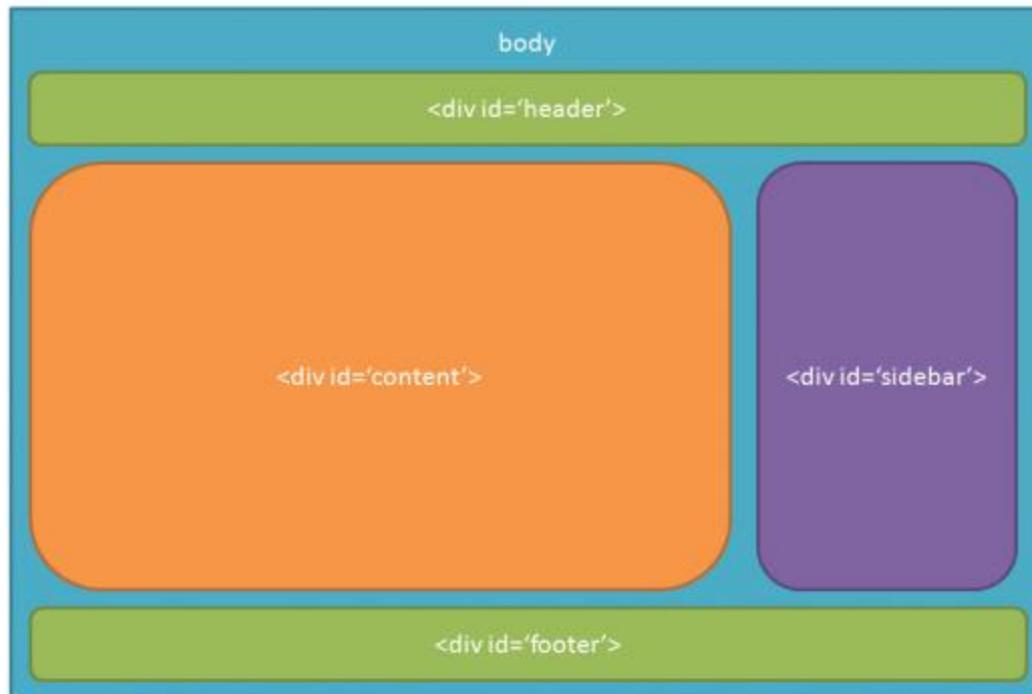
El formato de las páginas

- ▶ Estructura básica con HTML 3



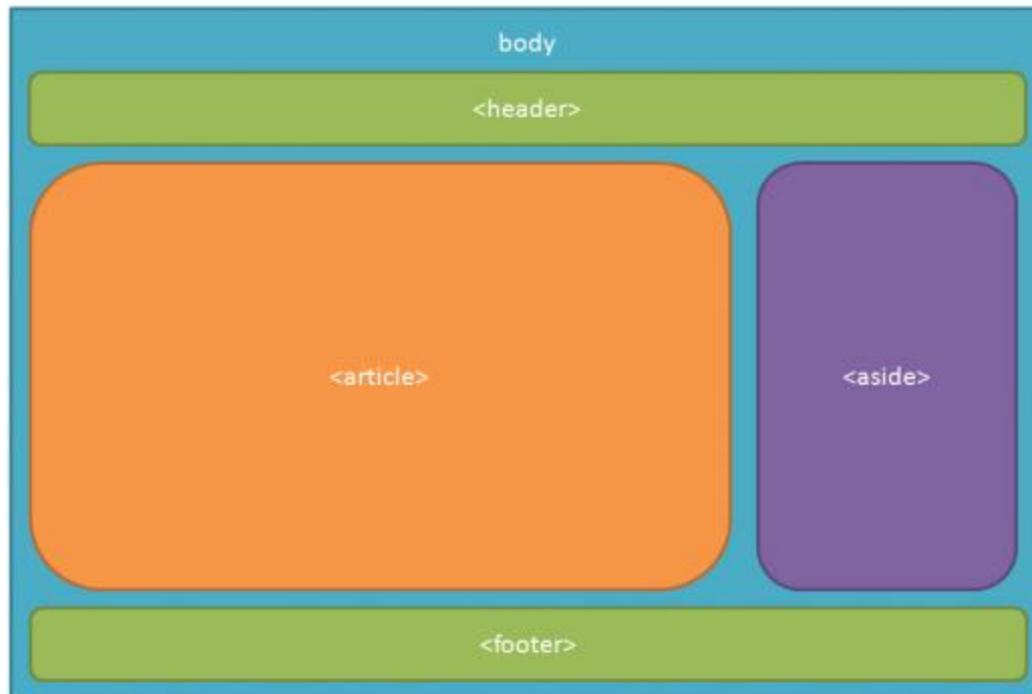
El formato de las páginas

- ▶ Estructura básica con HTML 4



El formato de las páginas

- ▶ Estructura básica con HTML 5



las metaetiquetas

- ▶ HTML 5 establece los **meta-tags** que ha de llevar una página, de forma que el resto puede obviarse:
 - ▶ **application-name**: se usa en el caso de que el contenido no sea una página web, por lo que, en general, no se usará.
 - ▶ **author**: es el nombre de la persona o empresa creadora del contenido (no de la web).
 - ▶ **description**: sólo ha de haber uno por página y es una breve descripción de lo que encontraremos en ella; recomiendo que tenga entre 125 y 175 caracteres, que es lo que aparecerá en los resultados de los buscadores.
 - ▶ **generator**: es el programa que ha creado la página y que, personalmente, recomiendo no incluir.
 - ▶ **keywords**: una serie de conceptos separados por comas que indican qué tendría que encontrar un usuario en la página; personalmente recomiendo entre 4 y 6 conceptos.

las metaetiquetas

- ▶ **robots**: es el que se utiliza para bloquear a los robots de búsqueda.
- ▶ **audience**: una lista de palabras que definirían el tipo de personas que sí o no pueden ver los contenidos.

- ▶ **El http-equiv** que hace referencia a algunas cabeceras:
 - ▶ **content-language**: define el idioma que tendrá la página en general.
 - ▶ **content-type**: es el tipo de contenido MIME que tendrá la página; habitualmente text/html.
 - ▶ **default-style**: establece cuál es la hoja de estilos CSS por defecto.
 - ▶ **refresh**: indica que esa página ha de refrescarse cada N segundos. Si se le indica después una URL se actualizará hacia esa dirección.

las nuevas etiquetas contenedoras

- ▶ `<header>`

- ▶ La inicio de los distintos elementos...

- ▶ `<footer>`

- ▶ El final de los distintos elementos

- ▶ `<article>`

- ▶ Lo más importante de la página (o sea, lo que hay que «posicionar»).

- ▶ `<aside>`

- ▶ Los contenidos relacionados con el contenido principal (article)

las nuevas etiquetas contenedoras

- ▶ `<section>`

- ▶ Las diferentes secciones con «identidad propia» donde organizar contenidos.

- ▶ `<nav>`

- ▶ El bloque que agrupa los enlaces

- ▶ `<figure>`

- ▶ El bloque que agrupa imágenes, elementos multimedia...

Otras cosas que vienen con HTML 5

- ▶ Se va a poder tener acceso a dispositivos externos como una cámara web gracias al tag **device**. Ahora mismo nadie da soporte a este elemento, pero básicamente tendrá 3 opciones gracias al parámetro type: **media** (una cámara web), **fs** (un USB o similar) y **rs232** (un puerto serie del dispositivo).
- ▶ A parte de esto, el **canvas**, y sobretodo algunos elementos externos como los **WebWorkers** (que vienen a ser como los “hilos” en programación que permiten ejecutar procesos en background), el **Web Storage** (o caché temporal en el navegador), los **WebSockets** para mantener bidireccionalidad en las comunicaciones, el **Web SQL Database**, a modo de base de datos temporal en el navegador, la **Geolocation** que empieza a tomar fuerza, los **SVG** o gráficos vectoriales, el **MathML** para incorporar elementos matemáticos o una nueva versión del **XMLHttpRequest** para todo lo llamado “AJAX”.

Anidación de elementos

- ▶ Otra cosa a tener en cuenta es la anidación de elementos. No podemos hacer cosas como esta:

```
<p>Esto es un <em>muy <strong>mal</em>
ejemplo</strong>.</p>
```

- ▶ Pero sí cosas como esta:

```
<p>Esto es un <em>muy <strong>buen</strong>
ejemplo</em>.</p>
```

- ▶ Además de potenciar elementos:

```
<p>Esto es un muy <strong><strong>buen</strong></strong>
ejemplo.</p>
```

la importancia de los enlaces

- ▶ sigue existiendo el «href»
- ▶ con el «target» podemos abrir las páginas en nuevas ventanas:
 - ▶ `_self`
 - ▶ `_parent`
 - ▶ `_top`
 - ▶ `_blank`
- ▶ el «media» el sitio en el que se abrirá:
 - ▶ `screen`
 - ▶ `tty`
 - ▶ `tv`
 - ▶ `projection`
 - ▶ `handheld`
 - ▶ `print`
 - ▶ `braille`
 - ▶ `aural`
 - ▶ `all`

la importancia de los enlaces

- ▶ indicamos el idioma de destino con «**hreflang**»
- ▶ el «**type**» permite indicar el MIME del contenido de destino
 - ▶ rel=**alternate**: elemento alternativo
 - ▶ rel=**archives**: página histórica
 - ▶ rel=**author**: creador del contenido
 - ▶ rel=**bookmark**: URL única del «**article**»
 - ▶ rel=**external**: la URL no forma parte del documento
 - ▶ rel=**help**: ayuda al contenido actual
 - ▶ rel=**license**: la licencia de ese artículo
 - ▶ rel=**nofollow**: los contenidos no tienen nada que ver con el destino
 - ▶ rel=**noreferrer**: no se ha de enviar el «**referrer**» por el navegador
 - ▶ rel=**search**: se enlaza a una página de resultados de búsqueda
 - ▶ rel=**sidebar**: contenidos secundarios relacionados con el principal
 - ▶ rel=**tag**: enlace al «**tag**» del contenido (no usar en nubes de tags)
 - ▶ rel=**index**, rel=**up**, rel=**first**, rel=**last**, rel=**next**, rel=**prev** ...

formularios para la web 2.0

- ▶ Los `<input>` evolucionan para permitir más contenidos:
 - ▶ **hidden**: Es un contenido oculto que suele corresponder a una serie de caracteres alfanuméricos. Este valor no puede ser manipulado por el usuario.
 - ▶ **text**: Campo de texto que no permite saltos de línea. Si se añade el atributo `value` será el contenido original del campo.
 - ▶ **search**: Campo de texto que no permite saltos de línea y que indica que será usado como contenido de una búsqueda. Si se añade el atributo `value` será el contenido original del campo.
 - ▶ **tel**: Campo de texto que indica un número de teléfono. No tiene ningún formato en particular, por lo que se pueden incluir espacios o guiones, por ejemplo.
 - ▶ **url**: Campo de texto en el que se pueden incluir direcciones URL absolutas. Se puede hacer una especie de autosuggest junto a otros elementos (ver ejemplo más abajo).
 - ▶ **email**: Campo de texto que permite incluir correo electrónico. Principalmente si se incluye el atributo `value` permite una única dirección, y si es el `multiple` puede incluir varias.

formularios para la web 2.0

- ▶ **password:** Campo de texto, pero que aparece oculto para el usuario y así proteger la información.
- ▶ **datetime:** Permite una fecha y hora válida en un formato como 0037-12-13T00:00Z, 1979-10-14T12:00:00.001-04:00 o 8592-01-01T02:09+02:09. Si se indica un número, serán los milisegundos transcurridos desde el 01/01/1970.
- ▶ **date:** Permite indicar una fecha. El atributo value indica la fecha en cuestión y el min y max los límites del rango de fecha. El atributo step marca los milisegundos (por defecto 1 día) entre un valor y otro. El formato sería 2010-03-09.
- ▶ **month:** Permite indicar un mes. Tiene los mismos atributos que date. El formato sería 2010-03.
- ▶ **week:** Permite indicar la semana del año. Tiene los mismos atributos que date. El formato sería 2010-W03 (tercera semana).
- ▶ **time:** Permite indicar una hora. Tiene los mismos atributos que date. El formato sería 17:23 y opcionalmente los segundos, 17:23:59.
- ▶ **datetime-local:** Es igual que el datetime pero no permite indicar ningún tipo de desfase horario.

formularios para la web 2.0

- ▶ **number**: Representa un valor numérico. Este puede ser negativo (si empieza con un guión -), seguido de un número y con decimales (si se le indica un punto .) y para acabar, un exponente, empezando con una letra e, seguida de un símbolo positivo + o negativo – y la cantidad exponencial. Un ejemplo completo sería: -12345.678e+90.
- ▶ **range**: El tipo range permite un slide en donde el número (valor) no es lo 100% importante, sino que es el formato en el que se muestra. Un poco más abajo, un ejemplo.
- ▶ **color**: Este elemento representa un color. El formato es del estilo #A0F59E (o sea, el RGB).
- ▶ **checkbox**: Este elemento ya existía, y es el que permite elegir unas cuantas opciones de un grupo de ellas.
- ▶ **radio**: Al igual que el anterior, permite elegir de un grupo, pero sólo una de las opciones.
- ▶ **file**: Éste permite la opción de subir archivos. Con él se pueden seleccionar archivos que luego serán enviados al sistema. Si se le indica el atributo multiple se podrán seleccionar varios ficheros a la vez. Además, si se indica el atributo accept se puede limitar ficheros según su MIME-type, por ejemplo con audio/*, video/* o image/*.

formularios para la web 2.0

- ▶ **submit**: Este es el botón que permite el envío de información al sistema. Puede ir acompañado de otros atributos que hacen que “ese botón” haga cosas que no son las generales del formulario. Los parámetros son: **formaction** (que hace ese botón), **formenctype** (la codificación), **formmethod** (la forma de envío), **formnovalidate** (la validación) y **formtarget** (en qué “ventana” se abre).
- ▶ **image**: Es igual que el submit pero permite ser una imagen en vez de un botón en sí. Para ello ha de ir acompañado del atributo `src` que indica la dirección URL de la imagen.
- ▶ **reset**: Otro botón que si pulsas deja los valores del formulario en su valor inicial.
- ▶ **button**: Un botón que “no hace nada”. Suele utilizarse habitualmente con funciones JavaScript que son los que realizan la acción.

Etiquetas que desaparecen

- ▶ **basefont, big, center, font, s, strike, tt, u**
 - ▶ principalmente por su posibilidad de uso con CSS
- ▶ **frame, frameset, noframes**
 - ▶ es obvio que ya no haya webs con frames, ¿verdad?
- ▶ **acronym, applet, isindex, dir**
 - ▶ principalmente por falta de uso...

Atributos globales

▶ **accesskey**

- ▶ Este atributo **accesskey** básicamente lo que permite es una mejora en la accesibilidad del sitio, pudiendo llegar a esa zona de la página pulsando una letra o número del teclado.
- ▶ Un ejemplo podría ser el de un enlace o el de un cajetín de un formulario:

```
<a accesskey="A" href="http://example.com/">Ejemplo</a>  
<label>buscador: <input type="search" name="q"  
accesskey="b"></label>
```

- ▶ Para el primer caso, el usuario debería usar la combinación de teclas Control + Alt + A para acceder a esta tecla. De todas formas, cada navegador puede tener su propia combinación de teclas.

Atributos globales

▶ **class**

- ▶ El atributo **class** básicamente se usa para lo que siempre se ha utilizado, y es para controlar las hojas de estilo. Dentro de un *class* se pueden poner varios contenidos haciendo referencia a los correspondientes en la hoja de estilos.

```
<a class="enlace" href="http://example.com/">Ejemplo</a>
```

- ▶ En este caso, si tuviéramos una definición del estilo “enlace” en el CSS, se aplicaría en este caso.

Atributos globales

▶ **contenteditable**

- ▶ Este es uno de los nuevos atributos que aparece, y es que **contenteditable** va a permitir que un acceso externo permita o no editar o cambiar el contenido de una etiqueta. Es un elemento heredado de los padres, por lo que si un padre no se puede editar, los hijos tampoco.
- ▶ Al ser un elemento booleano, sólo es necesario ponerlo con un “true” o “false” para que funcione.

```
<div contenteditable="false">Este contenido no debería ser  
editable desde el exterior...</div>
```

Atributos globales

▶ **contextmenu**

- ▶ Este es otro de los nuevos elementos que encontramos. **contextmenu** básicamente viene a dar un nombre a uno de los también nuevos tags, *menu*.
- ▶ Para resumir, viene a ser como si le pusieramos un “id” (un identificador, como os explico más adelante), pero para el caso especial de este tag. e de reconocer que es algo bastante nuevo y aún no le encuentro aplicación práctica (que seguro que la tiene).

Atributos globales

▶ **dir**

- ▶ El atributo **dir** básicamente indica la dirección del texto, o, mejor dicho, de escritura, para aquellos idiomas en los que se escribe de derecha a izquierda y no de izquierda a derecha como lo hacemos nosotros.
- ▶ Tiene tan sólo dos opciones que corresponden a lo que acabo de comentar: *ltr* (left to right) o *rtl* (right to left).

```
<div dir="ltr">Este contenido está escrito de izquierda a derecha...</div>
```

Atributos globales

▶ **draggable**

- ▶ Otra de las nuevas incorporaciones es el `draggable`, que cada vez que lo veo me da a pensar en las ventanitas y bloques que se pueden mover dentro de una web. Supongo que en un futuro ese será el uso principal que se hará, aunque en la actualidad va a permitir incorporar o eliminar elementos de una página simplemente arrastrándolos.
- ▶ Un ejemplo “de lo que hasta ahora podía tratarse como tal” sería una imagen que, al arrastrarla fuera de la ventana y colocarla en un editor de imágenes, se podía abrir. Otro ejemplo el de un enlace que se arrastraba sobre una pestaña del navegador y ejecutaba esa dirección.
- ▶ Los parámetros que permite son *true*, *false* o *auto*, que es el que está por defecto, dejando a cada usuario / navegador decidir los elementos.

Atributos globales

▶ **hidden**

- ▶ Seguro que el elemento **hidden** va a dar mucho que hablar, sobre todo en lo que a SEO se refiere. Hasta ahora cuando querías que un bloque de contenido no se viera tenías que utilizar los CSS para ocultarlo. Ahora eso ya no será necesario, ya que si un elemento incorpora este atributo simplemente no se mostrará por pantalla.
- ▶ De todas formas, este elemento no debe usarse para ocultar bloques o pestañas, sino que se ha de utilizar para elementos que no son todavía o no van a ser relevantes.
- ▶ Por ahora no hay ningún navegador que lo soporte, por lo que habrá que esperar a ver qué ocurre con el contenido que va dentro de estos elementos.

```
<div hidden>Este contenido está oculto por no ser relevante para el usuario...</div>
```

Atributos globales

▶ **id**

- ▶ Sin duda uno de los pilares básicos hoy en día en cualquier sitio, el **id** simplemente identifica un elemento de forma única (por lo que sólo puede haber uno que se llame así por página. A parte de utilizarse en las hojas de estilo, lo más habitual es su uso con *AJAX* y la función *getElementById()*.

```
<div id="explicacion">Este contenido puede variar si se  
accede mediante AJAX o similar...</div>
```

Atributos globales

▶ lang

- ▶ Otro de los clásicos sin duda es el **lang**, que marca el idioma del contenido de ese fragmento de texto. Hay que tener en cuenta que, aunque un documento se le aplique un idioma, por normal general, puede haber fragmentos de texto de contenidos que se encuentran en otros idiomas, por lo que hay que indicarlos.

```
<div lang="es">este bloque se encuentra escrito en español...</div>
```

```
<div lang="en">...and this is in english...</div>
```

Atributos globales

▶ **spellcheck**

- ▶ Otra de las novedades que aún no sé para qué, pero bueno, gracias a **spellcheck** se podrá indicar qué contenidos o no queremos que puedan sufrir una corrección ortográfica o gramatical. Seguramente acabará usándose en los formularios en los que haya que incluir palabras sin sentido, o en textos en un idioma distinto del general de la página...
- ▶ Tiene tres posibles valores, que son *true*, *false* y *default*.

Atributos globales

▶ **style**

- ▶ De los **style** no voy a contar gran cosa, ya que eso es todo un mundo, pero básicamente a cualquier elemento se le puede aplicar directamente, en línea, algunos atributos que hacen referencia a las hojas de estilo.

Atributos globales

▶ **tabindex**

- ▶ Este es quizá uno de los desconocidos a la hora de mejorar la usabilidad de un sitio, y es que gracias a **tabindex** cada vez que pulsemos en la tecla “tab” vaya a un sitio y otro de la página. De todas formas, no todos los elementos permiten tener este atributo, sólo aquellos en los que el usuario puede focalizarse. Lo más habitual es su uso en formularios, para que el usuario vaya saltando de un campo a otro en el orden que tú le digas, y no en el que el código fuente tenga.

Atributos globales

▶ **title**

- ▶ El genial **title** básicamente hace lo que su nombre indica: *titular*. Y es que muchos elementos, cuando se pone el ratón encima permiten mostrar más información de lo que eso significa. Un ejemplo muy habitual sería el las abreviaturas...

```
<p>Internet usa mucho el protocolo <abbr title="Hypertext  
Transport Protocol">HTTP</abbr> y el <abbr title="File  
Transport Protocol">FTP</abbr>.</p>
```

Algunos ejemplos útiles

- ▶ Los CSS pasan de:

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

- ▶ A algo como:

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

Algunos ejemplos útiles

- ▶ Los FAVICON pasan a:

```
<link rel="shortcut icon" href="favicon.ico">
```

- ▶ Los CHARSET pasan a:

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

- ▶ Los SCRIPT pasan a:

```
<script src="game-engine.js"></script>
```

Adelantándonos al usuario

- ▶ El **link prefetching** básicamente lo que permite es descargar las URL indicadas antes de que se vayan a visitar... y el HTML 5 incluye un sistema para avisarlo, ya de forma estándar.

```
<link rel="prefetch" href="http://sitio.web/url-siguiente/">
```

```
<link rel="prefetch" href="http://sitio.web/imagen-siguiente.png">
```

```
<a rel="prefetch" href="http://sitio.web/url-siguiente/">
```

Más y más información

- ▶ algunas novedades antes de empezar
- ▶ atributos globales
- ▶ cómo es la cabecera de una página
- ▶ cómo estructurar, ahora, una página
- ▶ los bloques de contenido textual
- ▶ enlaces que cobran vida
- ▶ dándole color a lo que escribimos
- ▶ imagen grande, ande o no ande
- ▶ las tablas... ¿de multiplicar?
- ▶ formularios, el elemento 2.0 (parte 1)
- ▶ formularios, el elemento 2.0 (parte 2)
- ▶ los “malditos” iframes
- ▶ las pseudo-clases para mayor interactividad
- ▶ el “rich media” está de moda
- ▶ el gran hermano te geolocaliza



Gracias



Javier Casares

javier.casares@kisslab.com - <http://javiercasares.com>