



# Comunicación Persona-Máquina

Éstandar HTML5

# Diseño Web

- A la hora de diseñar un sitio web, distinguimos:
  - Contenido: Todo lo que representa INFORMACIÓN dentro de un sitio web, así como su estructura y organización.
  - Presentación: Todo lo relativo al aspecto visual del sitio web.

# HTML

- Lenguaje de Marcado de Hipertexto (*HiperText Markup Language*).
  - Conjunto de normas para escribir una página web entendible por un navegador
  - Lenguaje utilizado para representar el **CONTENIDO** de las páginas Web.
- Estándar reconocido y cuyas normas define un organismo denominado W3C (*World Wide Web Consortium*).
  - Empresas, organismos públicos, desarrolladores de navegadores...



# HTML

- Comenzó como **único** lenguaje Web (1991):  
Contenidos + Presentación
  - Utilidad prácticamente académica
  - Poca importancia al aspecto gráfico, apenas sin presentación
- La popularización de Internet creó la necesidad de introducir más elementos de presentación en HTML
  - Mezcla de elementos de presentación y contenidos en un mismo documento
    - Códigos demasiado extensos
    - Páginas poco mantenibles

# HTML

- Nace CSS (*Cascade Style Sheets*) para posibilitar la separación de código (1996)
  - Ahora esa separación se convierte en obligatoria
  - Las etiquetas HTML relativas a presentación se mantienen únicamente por **retrocompatibilidad** → **NUNCA** se deben utilizar
- Versiones actuales HTML
  - HTML 4.01 es la última versión final
  - XHTML 1.0
    - Basado en XML añade semántica al HTML
  - HTML 5.0 es el presente y futuro

# HTML 5

- Es la versión de HTML vigente
  - Vio la luz en 2014 pero sigue estando en continuo desarrollo
  - Da nombre al estándar web actual, que incluye HTML 5, CSS3 y JavaScript
  - El W3C decidió excluir de esta versión tecnologías externas muy demandadas en los 90 (Flash, Silverlight, etc.)
    - Como alternativas se proponen nuevas funcionalidades CSS y JavaScript → En continuo desarrollo

# CSS

- *Cascade Style Sheet*
  - Complementa el HTML
- Define la **PRESENTACIÓN**, es decir, el formato y la apariencia de la página web
- Versiones
  - CSS I
  - CSS II
  - CCS<sub>3</sub> (adaptado a HTML 5.0)

# ¿Qué significa ser un estándar?

- Tanto HTML como CSS son estándares, definidos por el W3C
  - <http://www.w3.org>
  - Marca las directrices de la Web que seguirán
    - los navegadores y ...
    - ...otras herramientas Web.
  - Al tratarse de estándares, la interpretación puede ser diferente según el navegador utilizado
    - Probar las páginas siempre con varios navegadores diferentes



# El estándar HTML5

- Cuando se habla de HTML 5, tanto nos podemos referir:
  - A la versión 5 de HTML
  - Al estándar W3C que incluye:
    - HTML5
    - CSS 3
    - JS

# El estándar HTML 5

- El estándar HTML 5 es mucho más restrictivo respecto a la inclusión de tecnologías externas
  - CSS en su versión 3, y el apoyo de JS pretenden suplir estas carencias
  - En continuo desarrollo
- Características
  - Estructura semántica
  - Soporte multimedia nativo
  - Formularios mejorados
  - Gráficos y animaciones
  - Almacenamiento local y sin conexión

# Herramientas

- Editores Web

- Un documento .html es un simple archivo de texto que es interpretado por un navegador web
  - Podemos utilizar Notepad++
- Si queremos estructurar y organizar los proyectos
  - NetBeans, Aptana Studio, BlueGriffon, Aloha Editor, Maquetta...
  - Editores online: CodePen, Playcode, JSFiddle...

- Tutoriales

- W3Schools: <http://www.w3schools.com/>

# Herramientas

- Navegadores
  - Chrome, Firefox, Opera, etc...
- Para evaluar si tu navegador soporta HTML 5.0
  - <http://html5test.com/>
- Comprobación:
  - [HTML 5 Test](#) | [Find Me By IP](#) | [Can I Use](#)

# HTML: El contenido

El estándar HTML 5

# Características

- Solamente marcaremos contenido
- Las etiquetas responden a una semántica y no a una visualización
  - Solamente hay dos excepciones, que se tratarán más adelante
    - DIV
    - SPAN
- Cualquier etiqueta que pasa la frontera de la presentación deberá especificarse en CSS

# Elementos HTML

- Un elemento HTML consta de:
    - **Etiqueta de apertura + Contenido + etiqueta de cierre**
      - `<p>El contenido</p>`
    - **Solo una etiqueta**
      - `<hr>` : Separa dos textos diferentes conceptualmente (se visualizará una línea horizontal)
- NOTA: En versiones anteriores a HTML5 se requería “autocierre” en estas etiquetas: `<hr/>`*
- Algunas etiquetas tienen **parámetros**, cuyos valores van entre **comillas dobles**
    - `<img src=“imagen.jpg” width=“300px” height=“150px” />`

# Parámetros en etiquetas HTML

- Algunas etiquetas tienen **parámetros**, cuyos valores van entre **comillas dobles**
  - ``
- Los parámetros booleanos no tienen valor
  - Si los ponemos → true
  - Si no los ponemos → false

`<input type="checkbox" checked>`

*NOTA: En versiones anteriores a HTML5 los parámetros booleanos recibían su propio nombre como valor, para definirlos como ciertos:*

*`<input type="checkbox" checked="checked">`*



# Estructura de la página

`<!DOCTYPE html>`

Especifica el tipo de documento y las normas

`<html lang="es">`

Inicio de la página web

`<head>`

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

Codificación

**Cabecera**

Titulo, autores,  
estilos...(Información no visible)

`<title>Página simple</title>`

`</head>`

`<body>`

`<h1>Página simple</h1>`

`<p>Esto es un`

`<a href="demo.html">ejemplo</a> simple.</p>`

`<!-- this is a comment -->`

`</body>`

**Cuerpo**

Contenido de la página  
(Información visible)

`</html>`

Fin de la página web

# La cabecera <head></head>

Su contenido **no se muestra**, pero proporciona información útil

- <title>
- <link >
- <style>
- <script>
- <meta>
- ...

# El cuerpo <body></body>

Es la parte visible de la página web.

- <p>
- <h1, ..., <h6>
- <br>
- <hr>

# Visualización por defecto

- En ausencia de CSS, el navegador aplica estilos muy básicos al HTML
- Sirven de guía al diseñador para ver diferencias entre elementos antes de crear CSS
  - Cuando creamos HTML, **no debemos pensar en la presentación**

## Página simple

Esto es un [ejemplo](#) simple.

# Elementos semánticos

- Los elementos HTML sirven para **marcar** partes de una web: Títulos, párrafos, listas...
- Salvo dos elementos muy concretos (DIV y SPAN), **todos** responden a una semántica
- Debemos marcar elementos **únicamente atendiendo a la semántica de la etiqueta** y nunca a la visualización por defecto
- Ejemplo: Un título de nivel 1 se muestra más grande que de nivel 2, pero **nunca** pondremos por estética un título 2 sin estar precedido por un título 1.

# Reutilización visualización por defecto

- En ocasiones, la visualización por defecto del navegador, nos puede servir para su representación final.
  - **Esto no significa que debemos pensar en esa visualización -> Concordancia semántica**
- **Errores típicos:**
  - `<br><br><br>`
  - `<p>Quiero poner en negrita <strong>una palabra</strong> que no es importante</p>`
  - `<h1> Titulo secundario que quiero ver en grande</h1>`
  - Ect.

# Estructura en el <body></body>

- HTML 5 incluye varias etiquetas que nos permiten definir la estructura del documento
- Estas etiquetas no tienen una visualización por defecto en el navegador
- Nos permiten:
  - Definir bloques
  - Asignar roles a los bloques
    - Cada etiqueta define **semánticamente** al bloque
- Ejemplos: header, footer, section, article, aside, etc.

# Marcado genérico de bloques:

## La etiqueta `<div></div>`

Si se quiere definir un bloque que no tiene concordancia con las etiquetas semánticas vistas, usaremos `<div>`

- Los parámetros *id* y *class* nos permiten identificarlos en el css
  - Estos parámetros **son comunes en todos los elementos** del body, pero imprescindibles en el div
  - `<div id="15" class="CPM">El recurso 15 de CPM</div>`
- Las etiquetas semánticas estructurales aparecieron en HTML5 → Antes se utilizaba `<div>` para todo
  - `<div id="cabecera">` → `<header>`



# Etiquetas de texto

- Nos permiten marcar partes de texto como elementos semánticos concretos.
- Todos ellos, tienen un estilo por defecto aplicado por los navegadores.
  - Recordemos, que este se puede “reciclar”, pero **nunca** utilizar el marcado para conseguir ese efecto si no se corresponde semánticamente
- Ejemplos:
  - Destacar: `<strong>` `</strong>`
  - Enfatizar: `<em>` `</em>`
  - Párrafos: `<p>` y `</p>`
  - Citas: `<cite>` y `</cite>`
  - Etc.

# El problema de la retrocompatibilidad

- Por razones obvias, HTML ha de ser retro compatible, incluso con las primeras versiones del estándar.
- En estas versiones de HTML (cuando no había, o no se había desarrollado al 100% CSS) se incluían etiquetas de formato:
  - `<b>` Negrita (bold) `</b>`
  - `<i>` Cursiva (Italic) `</i>`
  - etc
- Estas etiquetas **no deben utilizarse en ninguna circunstancia** puesto que hacen referencia al diseño y no al tipo de información.

# Marcado genérico de texto:

## Etiqueta `<span>`

- Existen muchos contenidos textuales para los cuales HTML no especifica etiquetas.
- `<span>` permite marcar cualquier elemento que no se puede marcar con las otras etiquetas definidas.
- Esta etiqueta nos permitirá aplicar un estilo CSS a los fragmentos de texto que no coincidan semánticamente con ninguna de las etiquetas vistas
- Al igual que con los elementos `<div>`, los parámetros *id* y *class* cobran especial relevancia

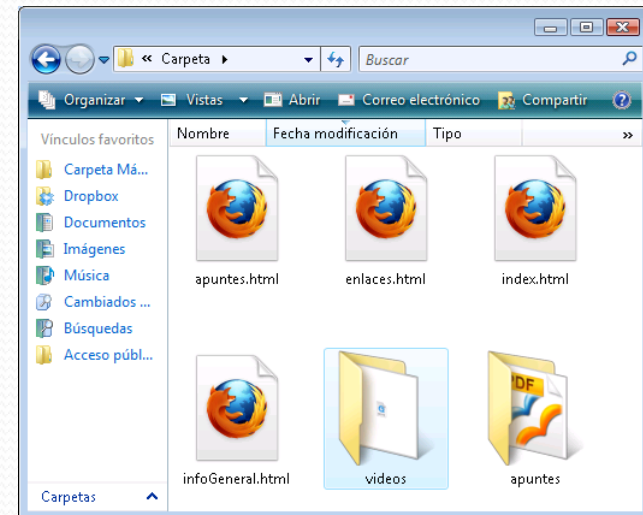
# Enlaces

- En función de la ubicación de la página:
  - Enlaces **externos**: Apuntan a páginas ubicadas en otros servidores.
  - Enlaces **internos**: Apuntan a páginas ubicadas en el mismo servidor en el que nos encontramos.
- En función del formato del enlace:
  - Enlaces **absolutos**: Incluyen toda la información necesaria (protocolo, servidor y ruta). **Normalmente se usan para enlaces externos.**
  - Enlaces **relativos**: Prescinden de parte de la información necesaria. **Normalmente se usan para enlaces internos.**

# Ejemplos enlaces

*La etiqueta html para enlaces, es “a”, con un parámetro “href” donde especificamos la url destino.*

- Referencias absolutas (enlace externo)
  - `<a href=http://www.google.es> Google </a>`
- Referencias relativas (enlaces internos)
  - `<a href= apuntes.html >`  
Apuntes `</a>`
  - `<a href= videos/video.avi >`  
Video `</a>`
  - `<a href= ../index.html>`  
Inicio `</a>`



# Imágenes, videos, audios

- Solamente los que formen parte de la información
- Las imágenes decorativas → CSS
- Imprescindible proporcionar alternativas al contenido multimedia → Accesibilidad
  - ``
  - `<video/audio controls [width="640" height="360"]>`
    - `<source src="video.mp4" type="video/mp4">`
    - `<source src="video.webm" type="video/webm">`
    - `<source src="video.ogg" type="video/ogg">`Tu navegador no soporta el elemento de video.  
`</video>`

*Dar varias alternativas -> Accesibilidad*

# Listas e información tabular

- Igual que el resto de estructuras semánticas, solo se utilizarán para indicar listados o información tabular.
  - Por sus características, se suelen cometer errores en su utilización
    - Listas para organizar secciones
    - Tablas para organizar espacialmente información no relacionada
    - Tablas para maquetar la página (Muy de moda hace años...)