

Tema 5

Tecnologías web: HTTP

Antonio Sanz – ansanz@unizar.es
José Luis Salazar – jsalazar@unizar.es

Comercio Electrónico



Tecnologías web: HTTP

Índice

- Protocolo HTTP
- Programación estática: HTML/CSS



Tecnologías web: HTTP

HTTP

- HTTP (Hypertext Transfer Protocol) → Protocolo de nivel de aplicación del TCP/IP (como DNS, FTP, SMTP)
- Es el corazón de la WWW → Se encarga de transferir información entre el cliente y el servidor
- Define “per se” la arquitectura cliente/servidor



Tecnologías web: HTTP

HTTP

1. El cliente abre una conexión TCP contra el puerto 80 (por defecto) de la IP que ha dado el servidor DNS
2. El servidor que está en el puerto 80 recibe la conexión y notifica al cliente



Tecnologías web: HTTP

HTTP

3. El cliente manda una petición (request) con la URL al servidor
4. El servidor manda una respuesta (response) con la información pedida, y cierra la conexión
5. El cliente recibe la información y la muestra en el navegador



Tecnologías web: HTTP

HTTP

- ¿Y si hay más de un objeto que descargar?
- HTTP 1.0 → De uno en uno (poco eficiente)
- HTTP 1.1 → Conexión persistente, se mantiene abierta la conexión hasta que se descargan todos (keep-alive connection)
- Truco Firefox: Activar el pipelining →
about:config + network.http.pipelining=true



Tecnologías web: HTTP

HTTP

- Ejemplo de HTTP request

GET /somedir/page.html HTTP/1.0

Host: www.example.com

User-agent: Mozilla/4.0

**Accept: text/html,
image/gif, image/jpeg**

Accept-language: fr

(extra carriage return, line feed)



Tecnologías web: HTTP

HTTP

- HTTP usa ASCII (texto plano) → Se puede leer
- GET: Método HTTP (se usa GET casi siempre)
- Host: Nombre al que nos conectamos
- Resto: Opciones extra
- ¡¡Importante el CRLF final!!



Tecnologías web: HTTP

HTTP

■ Ejemplo de HTTP response

HTTP/1.0 200 OK

Date: Thu, 06 Aug 1998 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 ...

Content-Length: 6821

Content-Type: text/html

data data data data data ...



Tecnologías web: HTTP

HTTP

- HTTP/1.0 200 OK → El 200 es un “código de respuesta”, que indica el estado de la operación
- Datos del servidor (fecha, tipo del contenido)
- Información requerida



Tecnologías web: HTTP

HTTP

■ Métodos HTTP:

- HEAD → Pide solo la cabecera (se usa para recibir solo la metainformación)
- GET → Pide la información (es el más usado)
- POST → Envía información (se usa en formularios web)
- PUT → Sube contenidos al servidor



Tecnologías web: HTTP

HTTP

- DELETE → Borra contenidos (no se usa)
- TRACE → Devuelve la misma respuesta
- OPTIONS → Devuelve los métodos que soporta el servidor
- CONNECT → Establece una conexión que puede convertirse en un túnel (por ejemplo, para SSL)



Tecnologías web: HTTP

HTTP

- Códigos de respuesta → Agrupados por familias:
 - 1xx Informational
 - 2xx Success (200 OK)
 - 3xx Redirection (302 Moved)
 - 4xx Client Error (**404 Not found**, 403 Forbidden)
 - 5xx Server Error (500 Internal Server Error)



Tecnologías web: HTTP

HTTP

■ HTTP Headers → Opciones adicionales

Header 	Description	Example	Types 
Accept	Content-Types that are acceptable	Accept: text/plain	requests
Accept-Charset	Character sets that are acceptable	Accept-Charset: iso-8859-5	requests
Accept-Encoding	Acceptable encodings	Accept-Encoding: compress, gzip	requests
Accept-Language	Acceptable languages for response	Accept-Language: da	requests
Accept-Ranges	What partial content range types this server supports	Accept-Ranges: bytes	responses
Age	The age the object has been in a proxy cache in seconds	Age: 12	responses
Allow	Valid actions for a specified resource. To be used for a <i>405 Method not allowed</i>	Allow: GET, HEAD	responses
Authorization	Authentication credentials for HTTP authentication	Authorization: Basic QWxhZGRpbjpvYGVuIHhlc2FtZQ==	requests
Cache-Control	Controls how proxies may cache this object	Cache-Control: no-cache	responses
Connection	What type of connection the user-agent would prefer	Connection: close	requests
Content-Encoding	The type of encoding used on the data	Content-Encoding: gzip	responses
Content-Language	The language the content is in	Content-Language: da	responses
Content-Length	The length of the content in bytes	Content-Length: 348	responses



Tecnologías web: HTTP

HTTP

Content-MD5	An MD5 sum of the content of the response	Content-MD5: 3167b9c13ad2b6d36946493fc47976c8	responses
Content-Range	Where in a full body message this partial message belongs	Content-Range: bytes 21010-47021/47022	responses
Content-Type	The mime type of this content	Content-Type: text/html; charset=utf-8	responses
Date	The date and time that the message was sent	Date: Tue, 15 Nov 1994 08:12:31 GMT	responses, requests
Host	The domain name of the server (for virtual hosting)	Host: www.w3.org	requests
If-Modified-Since	Allows a <i>304 Not Modified</i> to be returned	If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT	requests
Last-Modified	The last modified date for the requested object	Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT	responses
Location	Used in redirection	Location: http://www.w3.org/pub/WWW/People.html	responses
Server	A name for the server	Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)	responses
User-Agent	The user agent string of the user agent	User-Agent: Mozilla/5.0 (Linux; X11; UTF-8)	requests



Tecnologías web: HTTP

HTTP

- **Accept / Content-type**: Lista los tipos de contenido que acepta el cliente → basado en los tipos MIME (Multipurpose Internet Mail Extensions)
- Tipos MIME : Definidos en categoría/tipo
- Ejemplo: image/gif , application/javascript , text/html
- Vitales porque el navegador usa el tipo MIME para saber con qué programa abrirlo



Tecnologías web: HTTP

HTTP

- **Accept-Charset:** Indica los tipos de caracteres que soporta el navegador
- Importante para los idiomas (caracteres chinos, cirílico ... ¡¡y para los acentos!!)
- Lo normal en Europa : ISO-8859-1 (*Latin-1 Western European*) y UTF-8
- Importante: Servidor y página web tienen que tener el mismo charset (si hay conflicto, el navegador puede no saber cómo representar la página)



Tecnologías web: HTTP

HTTP

- **Host:** Indica el nombre de dominio al cual nos queremos conectar
- 1 web, 1 servidor → Falso
- Es posible poner miles de páginas web en un servidor → Virtual Hosting
- La directiva Host indica la página exacta dentro del servidor



Tecnologías web: HTTP

HTTP

- **User-Agent:** Indica al servidor el tipo de navegador que tenemos (Firefox, Internet Explorer)
- Cada navegador se comporta de manera distinta → Es posible dar versiones “personalizadas” de una página web
- Útil también para contenido móvil (ver una web en el teléfono)



Tecnologías web: HTTP

HTTP

- **Cache-Control:** Permite controlar cuándo el navegador puede guardar en local una copia de la información
- Importante ahorro de ancho de banda → pero puede traer problemas con páginas que cambian muy rápidamente



Tecnologías web: HTTP

HTTP

- **Authorization:** El propio servidor puede de forma integral pedir al cliente que se autentique con usuario y contraseña sin programación adicional
- Tipos: Basic Access Auth/Digest Access Auth
- Basic Auth: La contraseña va en claro en la propia petición, codificada en base64



Tecnologías web: HTTP

HTTP

- Ejemplo: (contraseña: *open sesame*)

```
GET /private/index.html HTTP/1.0
```

```
Host: localhost
```

```
Authorization: Basic QWxhZGRpbjpvYGVuIHNLc2FtZQ==
```

- Muy problemática en seguridad
- Digest access auth: Emplea MD5 para mandar un hash de la contraseña → Mucho más seguro



Tecnologías web: HTTP

HTTP

- HTTP es un protocolo sin estado (stateless) → No guarda el estado de la comunicación
- Problema: En muchas ocasiones, necesitamos conocer y mantener el estado de una comunicación
- Ejemplo: Carrito de la compra, zona de usuario, correo web



Tecnologías web: HTTP

HTTP

- ¿Cómo podemos mantener el estado de una comunicación?
 - En el cliente → Cookies
 - En el servidor → Sesiones



Tecnologías web: HTTP

HTTP

- **Cookies:** Información que se almacena en el ordenador cliente y que ayuda a interactuar con el servidor
- Usos: Mantener el estado de una comunicación, o guardar preferencias (por ejemplo, el estilo de una web, tamaño de letra, colores)



Tecnologías web: HTTP

HTTP

- Ejemplo de una cookie :

HTTP/1.1 200 OK

Content-type: text/html

Set-Cookie: name=value
(content of page)

- Con esto, se crea un fichero que guarda el par “name=value”



Tecnologías web: HTTP

HTTP

- A partir de ese momento, el navegador manda la cookie para la página que se está visitando:

GET /spec.html HTTP/1.1

Host: www.w3.org

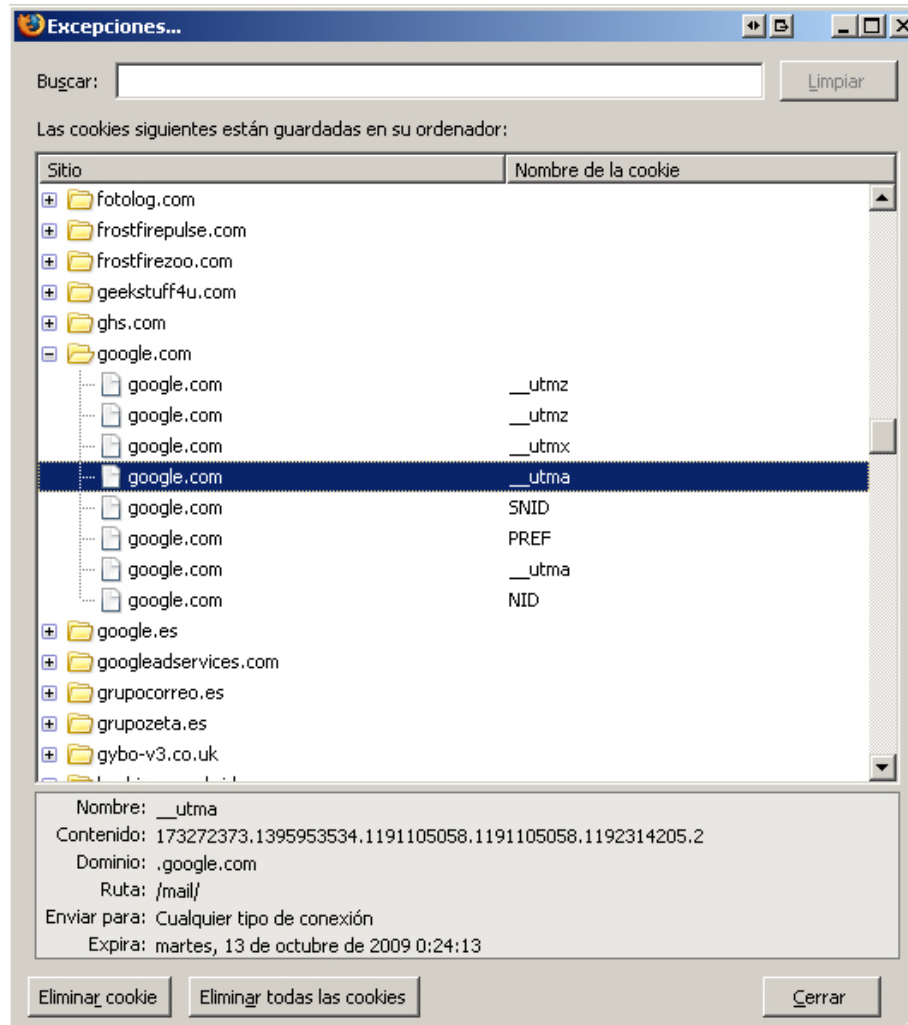
Cookie: name=value

- Ejemplo sencillo: Elegir el idioma de la página



Tecnologías web: HTTP

HTTP



En Firefox :
Herramientas
→ Opciones →
→ Privacidad →
Cookies



Tecnologías web: HTTP

HTTP

- El servidor puede pedir siempre la cookie al cliente → pero sólo de su dominio (a priori)
- Si la cookie no está bien hecha, otras páginas pueden pedirla → Problemas de privacidad
- Todas las cookies tienen una fecha de expiración → Pasado ese tiempo, dejan de ser útiles



Tecnologías web: HTTP

HTTP

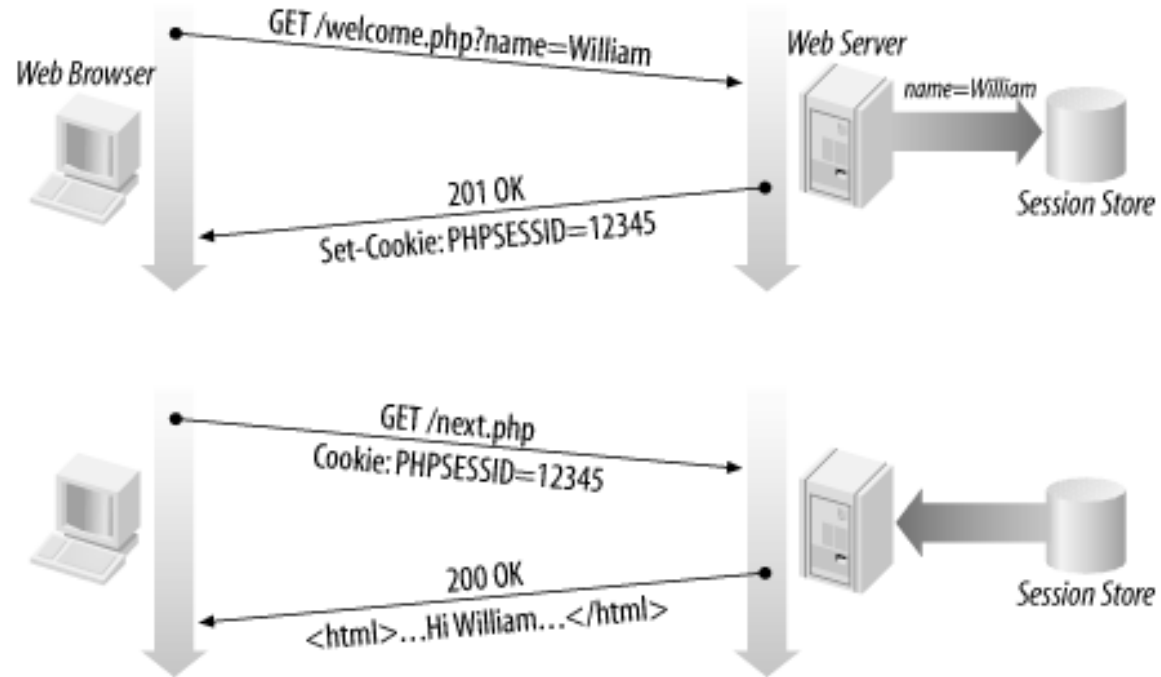
- **Sesiones:** Todo el estado se almacena en el servidor
- Usan una única cookie para guardar el identificador de sesión (SessionID)
- Las sesiones (como las cookies) se realizan mediante programación



Tecnologías web: HTTP

HTTP

■ Ejemplo de una sesión:



Tecnologías web: HTTP

HTTP

- Son más complejas de programar que las cookies, pero mucho más potentes (y más seguras)
- También tienen expiración (normalmente, cuando se cierra el navegador)



Tecnologías web: HTTP

Conclusiones

- WWW → Conjunto de sistemas destinados al manejo de información
- HTTP → Protocolo sobre el cual se monta la WWW
- HTML → Protocolo que se monta sobre HTTP



Tecnologías web: HTTP

HTTP

¿Algún código HTTP de duda?

¿?



Tecnologías web: HTTP

HTML



Tecnologías web: HTTP

Introducción

- Tenemos ya una idea, unos contenidos, una estructura y unos servicios
- ¡¡Toca ponerse manos a la obra!!
- Objetivo: Conocer qué tecnologías se emplean para construir páginas web



Tecnologías web: HTTP

HTML

- HTML = HyperText Markup Language → Lenguaje de marcas o etiquetas (tags)
- Empleado por todos los navegadores del mundo → es el “esperanto” de la WWW
- Evolución: HTML 2.0 (1985) → HTML 4.0 (1997) → HTML 5.0 en proceso de estandarización



Tecnologías web: HTTP

HTML

- HTML → Lenguaje de marcas o etiquetas → Asignamos etiquetas para modificar el texto
- Ejemplo: `<b> Hola, mundo! </b>` pone el texto en negrita
- Separa estructura de contenido
- Siempre es texto ASCII (fácil de leer e interpretar)
- Las etiquetas no son sensitivas a las mayúsculas



Tecnologías web: HTTP

HTML

- Una etiqueta HTML va entre corchetes (<>), y tiene que tener un final marcado con / (</>) → Problemas en la representación si no se cumple
- Es posible anidar etiquetas HTML
- Ejemplo: <h1>Gran Titular</h1>
- Algunas etiquetas tienen atributos:
- Ejemplo: Hola



Tecnologías web: HTTP

HTML

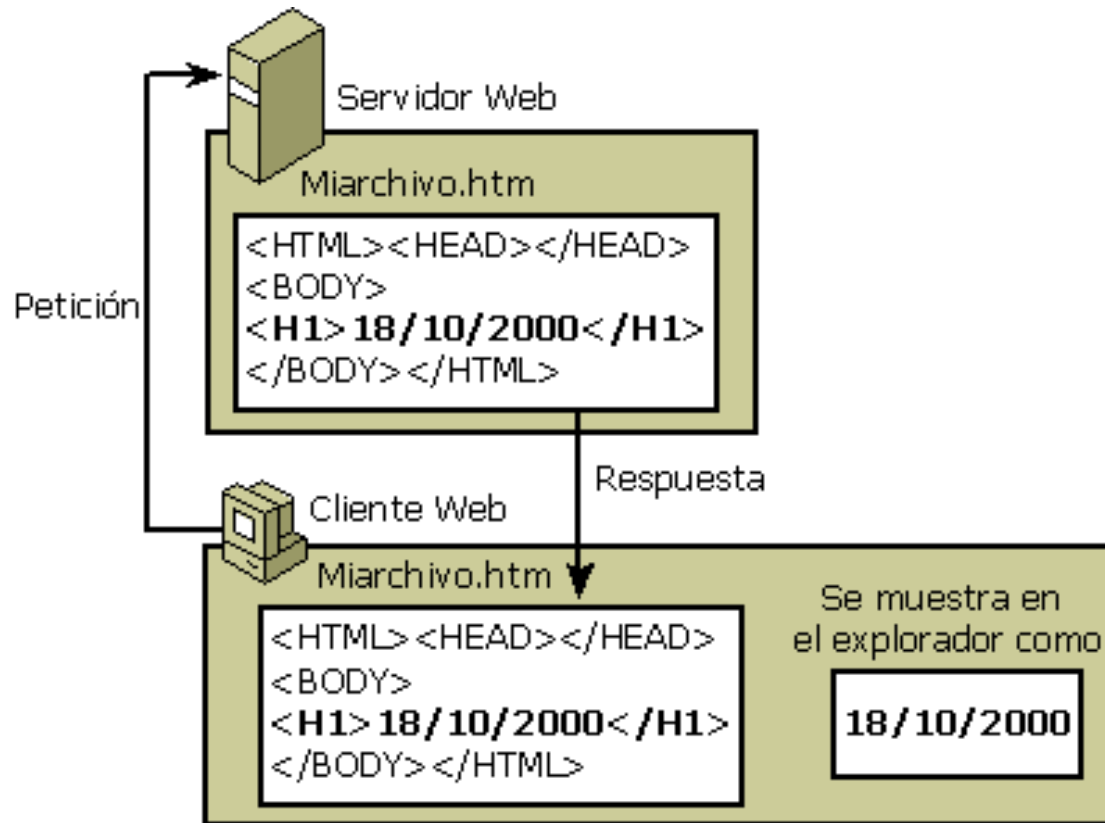
■ Ejemplo de página web funcional:

```
<html>
<head>
  <title> Mi primera web</title>
</head>
<body>
  Hola , mundo !!
</body>
</html>
```



Tecnologías web: HTTP

HTML



Tecnologías web: HTTP

HTML

■ Tags de estructura:

- `<html>` → Código HTML (vital)
- `<head>` → Cabecera del HTML
- `<body>` → Cuerpo del mensaje
- `<frame>` → Marcos (en desuso)
- `<table>` → Tablas de contenido
- `<li>` y `<ol>` → Listas
- `<div>` → Capas de contenido (VIP!!)



Tecnologías web: HTTP

HTML

■ Tratamiento de textos:

- `<title>` → Título de la página
- `<h1-6>` → Titulares
- `<p>` → Párrafos
- `<br>` → Salto de línea (no necesita `</br>`)
- `<b>` → Texto en negrita
- `<i>` → Texto en cursiva
- `<font>` → Definición de la fuente a usar



Tecnologías web: HTTP

HTML

- Comentarios: `<!--      -->`

`<!-- Esto es un comentario    -->`

- Caracteres especiales:

`&` → `&`

`"` → `"`

` ` → (espacio en blanco) → Muy usado

`á` → `á`

`é` → `é`

`í` → `í`

`ó` → `ó`

`ú` → `ú`

`ñ` → `ñ`



Tecnologías web: HTTP

HTML

■ Enlaces web: `<a href>`

- `<a href="http://www.unizar.es/">` Universidad de Zaragoza `</a>`
- `<a href="login.html">` Inicio de Sesión `</a>`
- `<a href="catalogo/inicio.html">` Nuestros productos `</a>`
- `<a href="ftp://ftp.rediris.es">` Rediris FTP Server `</a>`
- `<a href="mailto:info@miproyecto.com">` Contacte con nosotros `</a>`



Tecnologías web: HTTP

HTML

■ Listas de elementos:

Lista sin numerar:

```
<ul>  
  <li>Uno</li>  
  <li>Dos</li>  
  <li>Tres</li>  
</ul>
```

Lista numerada:

```
<ol>  
  <li>Uno</li>  
  <li>Dos</li>  
  <li>Tres</li>  
</ol>
```

- Muy útiles para hacer menús (con un poco de “ayuda”)



Tecnologías web: HTTP

HTML

■ **Imágenes:** ``

- src = Ruta de la imagen
- alt = Texto alternativo
- align = Posicionamiento
- width & height = Anchura y altura

■ Imágenes con enlaces:

```
<a href="ejemplo.html"></a>
```



Tecnologías web: HTTP

HTML

■ Tablas:

```
<TABLE>  
<TR>  
<TD>Fila 1 - Celda 1</TD>  
<TD>Fila 1 - Celda 2</TD>  
</TR>
```

```
<TR>  
<TD>Fila 2 - Celda 1</TD>  
<TD>Fila 2 - Celda 2</TD>  
</TR>
```

```
<TR>  
<TD>Fila 3 - Celda 1</TD>  
<TD>Fila 3 - Celda 2</TD>  
</TR>  
</TABLE>
```

- TR = Fila , TD = Columna
- Indentar el código, y ser consecuentes con la definición de tablas



Tecnologías web: HTTP

HTML

- **Capas:** `<div>`
- Empleadas principalmente para dividir secciones de contenido y crear la estructura (más eficiente que tablas o frames)
- Uso real → Junto con las CSS
- Ej: `<div class="title"> <h1>Comercio Electrónico</h1> </div>` → el estilo “title” está en la hoja de estilos



Tecnologías web: HTTP

HTML

- **Formularios:** `<form>`
- Permiten que el usuario introduzca información
→ Para entregarla a otro programa
- Muy importantes en las aplicaciones web → Tienen sus propias etiquetas que solo se aplican dentro de un formulario
- Ejemplo: `<form action="/opinion2.php" method="post">`
 - Action = Programa a ejecutar
 - Method = método http (GET o POST)



Tecnologías web: HTTP

HTML

■ *Ejemplos visuales:*

http://www.w3schools.com/html/html_examples.asp

■ *Campo de texto:*

Ejemplo: `<input type="text" name="nombre">`

- Crea un campo de texto simple para rellenar



Tecnologías web: HTTP

HTML

- *Campo de texto con contraseña:*

Ej : `<input type="password" name="password">`

- Crea un campo con texto pero que visualiza asteriscos

- *Campo de area de texto:*

Ej: `<textarea rows="10" cols="30"> </textarea>`

- Crea un campo de texto con varias líneas



Tecnologías web: HTTP

HTML

- *Checkbox* → Cuando podemos marcar una opción o varias
- Ej: `<input type="checkbox" name="vehicle" value="Bike" />`
 - `name` = nombre de la variable
 - `value` = valor que se le pasa al script o programa



Tecnologías web: HTTP

HTML

- *Botón radio* → Cuando hay que elegir una entre varias opciones

Ej: `<input type="radio" checked="checked" name="votacion" value="Si">`
`<input type="radio" name="votacion" value="No">`

- `checked`: valor que queremos activar por defecto



Tecnologías web: HTTP

HTML

- *Listas desplegables:*

```
<select name="cars">  
<option value="volvo">Volvo</option>  
<option value="saab">Saab</option>  
<option value="fiat">Fiat</option>  
<option value="audi">Audi</option>  
</select>
```

- Permiten seleccionar entre varios valores



Tecnologías web: HTTP

HTML

- Formularios: Herramientas vitales para un negocio online
- Hay que diseñarlos con cuidado pensando en la usabilidad
- Consejos:
 - Facilitar el trabajo a los usuarios
 - Acotar posibles entradas
 - Elegir bien los campos por defecto
 - Validar los formularios de forma inteligente



Tecnologías web: HTTP

HTML 5

- Desarrollo en paralelo con XHTML.
- Etiquetas (canvas 2D y 3D, audio, video) con codecs para mostrar contenidos multimedia
- Etiquetas para manejar grandes conjuntos de datos: Datagrid, Details, Menu y Command
- Mejoras en los formularios con nuevos tipos de datos (eMail, number, url, datetime ...)
- Visores: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales).
- Drag & Drop.
- Etiquetas para manejar la Web Semántica (Web 3.0): header, footer, article, nav, time (fecha del contenido), link rel="" (tipo de contenido que se enlaza).



Tecnologías web: HTTP

HTML

■ Ventajas:

- Estándar abierto
- Usado en el mundo entero
- Muy sencillo de aprender y manejar

■ Desventajas:

- Solo mostramos datos, no tenemos dinamismo (interacción con el usuario) → Programación dinámica



Tecnologías web: HTTP

CSS – Hojas de estilos

CSS



Tecnologías web: HTTP

CSS – Hojas de estilos

- Problema: Queremos cambiar el fondo de nuestra web de verde botella a verde césped → Cambiar la etiqueta HTML “bgcolor” una a una en las 36 páginas
- ¿No estaría genial cambiarlo solo en un sitio, y que se aplicara a todos a la vez?
- Solución: CSS (Cascading Style Sheets) → Hojas de Estilo



Tecnologías web: HTTP

CSS – Hojas de estilos

- CSS – Hojas de estilos: Forma de separar el contenido y la presentación en una web
- Tenemos las páginas con el texto, y éstas consultan a la CSS sobre cómo formatearse
- De esta forma, es muy sencillo modificar o renovar completo una página web



Tecnologías web: HTTP

CSS – Hojas de estilos

- Ejemplo:

```
body
{
text-align: center;
color: black;
font-family: arial
}
```

- Traducción al castellano: Todo texto que esté dentro de una etiqueta `<body>` estará alineado a la izquierda, será de color negro y usará la fuente Arial



Tecnologías web: HTTP

CSS – Hojas de estilos

■ Reglas básicas :

- Definición básica: selector {propiedad: valor;}
- Todas las líneas de un estilo terminan en “;”
- Cada estilo se define entre { }
- Los estilos nuevos se definen con **.nombre**
- Ejemplo: .miestilo { text-align: left ;}



Tecnologías web: HTTP

CSS – Hojas de estilos

■ Localización de la CSS:

- Dentro del propio documento (dentro del <head>)
- En un fichero externo ← Optimo

Ejemplo: <link rel="stylesheet" type="text/css" href="default.css" media="screen"/>

■ CSS → Cascading: Propiedades apilables en cascada (de menor a mayor prioridad)

- Navegador
- Hoja de estilo externa
- Hoja de estilo interna (dentro del tag <head>)
- Estilo del elemento (dentro del tag HTML)



Tecnologías web: HTTP

CSS – Hojas de estilos

- CSS → Apilamiento de estilos (dos etiquetas se comportan de forma diferente dependiendo del bloque en el que se encuentren)
- Ejemplo:
 - `.content li {color: red }`
 - `.item li {color : black }`
 - `li {color: green }`



Tecnologías web: HTTP

CSS – Hojas de estilos

- Atributo “class” → Para el HTML → Permite crear distintos estilos para un mismo tipo de etiqueta HTML
- Ejemplo (en la CSS):
`h1.izquierda { text-align: right ;}` `h1.centrado {text-align: center }`
- En el HTML:
`<h1 class=“izquierda”> Titular alineado a la izquierda </h1>`
`<h1 class="centrado"> Titular centrado </h1>`



Tecnologías web: HTTP

CSS – Hojas de estilos

- El atributo “class” también nos permite crear nuevos estilos (sobre todo con las etiquetas <div>) → Uso fundamental de las CSS
- Ejemplo (CSS):
`.item {padding: 6px 12px; border: 1px solid #EEE; background: #FFF; margin-bottom: 8px;}`
- Ejemplo (HTML):
`<div class="item"> Esto es una noticia </div>`



Tecnologías web: HTTP

CSS – Hojas de estilos

■ Ejemplo: CSS básica

```
/* Body: Cuerpo del texto, lo que se aplica si no hay ningún otro estilo que lo modifique */
body {
    background-color: #ddd;                /* Color de fondo */
    font-family: Vernada, Arial, Times;    /* Tipo de letra a usar */
    font-size: 11px;                      /* Tamaño de la letra */
    color: #666;                          /* Color de la letra */
    margin: 5px;                          /* Margen con respecto a la capa */
}
/* Header : Zona de contenido que se emplea como cabecera */
.header {
    margin: auto;                        /* Se emplea para que la capa se redimensione de forma automática */
    width: 740px;                       /* Tamaño de la capa → Escogido para 800x600 */
    background-color: #ccc;
    padding: 8px;                        /* Zona de margen interior de la capa, se aplica en todos los bordes */
    border: 2px solid #999;              /* Zona de margen externo de la capa, con un color aplicado */
    font-size: 15pt;
    font-weight: bold;
    text-align: right;
}
/* Justificado del texto a la derecha */
/* 740 + 8x2 + 2x2 = 760 = espacio dejado para el navegador */
}
/* Container : Contenedor que engloba al menú y a la zona de contenido central */
.container {
    margin: auto;
    width: 760px;
    background-color: #FFF;
    margin-top: 2px;
}
/* el margen se puede aplicar solo a un borde */
```



Tecnologías web: HTTP

CSS – Hojas de estilos

■ Ejemplo: CSS básica

```
/* Menu: Zona de contenido a la izquierda que contiene el menú de navegación */
.menu {
    background-color: #ccc;
    width: 150px;
    float: left;
}
/* Contenido : Zona de contenido principal */
.contenido {
    width: 590px;
    background-color: #eee;
    float: left;
    padding: 10px 10px 10px 10px;
    text-align: justify;
    color: #000;
}
/* Añadido especial para que el Firefox coopere */
.cleaner {
    clear: both;
    font-size: 1px;
}
/* Pie de pagina : Para informacion de contacto u otras cosas */
.footer {
    clear: both;
    width: 760px;
    margin: auto;
    background-color: #FFF;
    margin-top: 2px;
    text-align: center;
}
```

/ Un tamaño recomendable para un menu */*
/ Para que se apile a la izquierda */*

/ 590 + 150 + 10 + 10 = 760 Voilà */*
/ Se apila a la izda ... pero después de la OTRA */*

/ Esto impide que ninguna capa se acople a izda o dcha → pone orden */*
/ Firefox descarta las capas sin contenido, por eso ponemos un pixel */*



Tecnologías web: HTTP

CSS – Hojas de estilos

- Consejos de uso de las CSS:
 - Crear primero la estructura de la web
 - Indentar el código
 - Poner primero la parte de la estructura y luego los estilos
 - Juntar todos los estilos anidados
 - No usar extensiones “extrañas” (compatibilidad)



Tecnologías web: HTTP

CSS – Hojas de estilos

■ Ventajas:

- Independizan contenido de formato → Sensacionales para hacer modificaciones
- Ofrecen mucha potencia (Ejemplo: esquinas redondeadas, menús desplegables)
- Es posible definir la estructura de una web con ellas

■ Desventajas:

- Mucha potencia hace que a veces sean muy complejas
- Compatibilidad entre todos los navegadores



Tecnologías web: HTTP

Estándares web

- **W3C** (World Wide Web Consortium) → Conseguir que todas las web usen el mismo idioma
- Estándares: XHTML 1.0, CSS 2.0
- Ventajas de usar los estándar:
 - Más compatibilidad de navegadores
 - Más sencillez a la hora de interoperar (servicios web, por ejemplo)
 - Mejor para el posicionamiento



Tecnologías web: HTTP

Estándares web

- **XHTML 1.0** → HTML + XML
- Objetivo: Aumentar la compatibilidad con programas automáticos
- HTML 4.0 → Lenguaje muy “relajado”
- XHTML → Normas más “estrictas”



Tecnologías web: HTTP

Estándares web

■ Reglas básicas:

- Los elementos tienen que estar siempre cerrados y con `</>`

Ej: `<li> Lista1` ← MAL
`<li> Lista1 </>` ← Bien
``
`<br />`

- Los elementos tienen que estar anidados correctamente:

Ejemplo: `<b><i>This text is bold and italic</b></i>` ← NO
`<b><i>This text is bold and italic</i></b>` ← OK



Tecnologías web: HTTP

Estándares web

- Todos los documentos tienen que tener un elemento raíz (`<html>` al menos)
- Todos los elementos tienen que ir en minúsculas
- Los atributos tienen que ir entre comillas:

Ejemplo: `<font color=red>Hola mundo</font>` ← Mal
`<font color="red">Hola mundo</font>` → OK



Tecnologías web: HTTP

Estándares web

- Todos los documentos tienen que tener una declaración DOCTYPE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
  Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
  transitional.dtd<html xmlns="http://www.w3.org/1999/xhtml"  
  xml:lang="es" lang="es">  
<head> </head>  
<body> </body>  
</html>
```



Tecnologías web: HTTP

Estándares web

- DTD: Document Type Definitions
 - Strict → Cuando tenemos un código muy limpio + CSS
 - Transitional → Cuando queremos una amplia compatibilidad
 - Frameset → Cuando queremos usar marcos



Tecnologías web: HTTP

Consejos de diseño

- Emplear CSS para crear la estructura de la página
- Usar tablas para contenido (nunca para estructura)
- Dividir la página en bloques (hacer cada bloque como un `<div>`)
- Compatibilidad con Internet Explorer + Firefox
→ no usar extensiones “raras”



Tecnologías web: HTTP

Consejos de diseño

- Cumplir los estándares → Validar el código HTML y CSS
- Hacer pruebas en local (XAMPP)
- Tener en cuenta la resolución
- Cuidado con la elección de colores RGB
- Imágenes a 72ppp & dimensionadas



Tecnologías web: HTTP

Conclusiones

- HTML → Corazón de la web (todo al final es HTML)
- CSS → Separan contenido de presentación (obligatorias)
- W3C → Estándares web



Tecnologías web: HTTP

Recursos adicionales

Excelentes plantillas web de HTML + CSS (¡y gratis !)

<http://www.oswd.org>

<http://www.mezzoblue.com/zengarden/alldesigns/>

Tutoriales sobre HTML – CSS – JavaScript – PHP

<http://www.webestilo.com/>

<http://www.w3schools.com/default.asp>

Tutoriales sobre CSS

<http://www.w3schools.com/css/default.asp>

http://www.w3schools.com/css/css_examples.asp

Estándares W3C

Web Developer's Handbook

<http://www.alvit.de/handbook/>

Tutoriales de PHP y formularios web

<http://www.desarrolloweb.com/php/>

<http://www.webestilo.com/php/>

