


# Introducción a las Tecnologías Web

Desarrollo de Aplicaciones en Internet  
Universidad Autónoma de Zacatecas



# Tecnologías Web

- Conjunto de mecanismos ó técnicas que nos permiten desarrollar aplicaciones que facilitan la compartición de información y colaboración en línea a través de Internet
  - HTTP
  - ~~HTML~~ 
    - XHTML
    - CSS



## Introducción a las Tecnologías Web

# HTTP



# HTTP

- HyperText Transfer Protocol
- Lenguaje de la Web
  - Protocolo usado para comunicar navegadores web (clientes) con servidores web
  - Puerto TCP 80 por default
  - Descrito en [RFC 1945](#)



# Modelo Cliente/Servidor

- El modelo cliente/servidor puede describir una relación entre dos programas de computadora— el "**cliente**" y el "**servidor**".
- Cliente
  - Solicita algún tipo de servicio (tal como un archivo o un acceso a una base de datos) al servidor.
- Servidor
  - Realiza la solicitud y transmite los resultados al cliente a través de la red



# Cliente Web

- Se encuentra conectado al Internet cuando se necesita
- Usualmente ejecuta algun software navegador de web (cliente) tal como Internet Explorer o Firefox
- Usa HTTP (Hypertext Transfer Protocol)
- Solicita paginas web del servidor
- Recibe paginas web y archivos del servidor



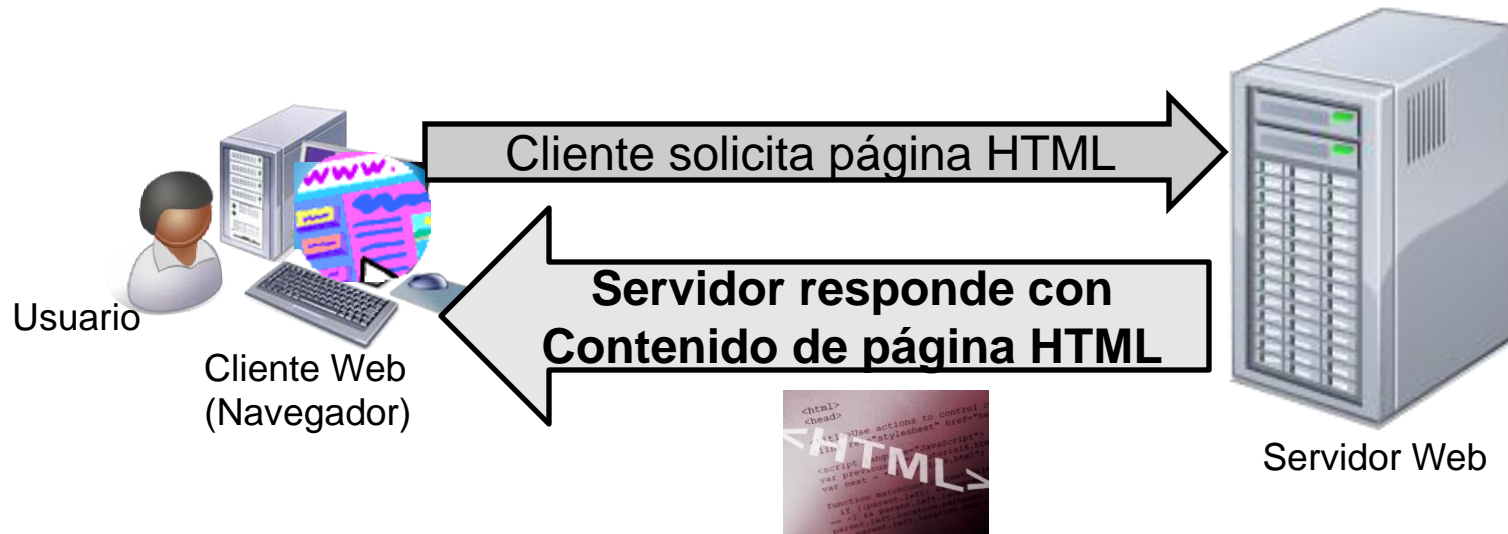


# Servidor Web

- Continuamente conectado al Internet
- Ejecuta algun software de servidor web (tal como Apache ó Internet Information Server)
- Usa HTTP (Hypertext Transfer Protocol)
- Recibe solicitudes de páginas web
- Responde a las solicitudes y transmite codigos de status, páginas web y archivos asociados



# Solicitud Simple de HTTP







# URL

- Universal Resource Locator

`http: //www.banxico.org.mx:80 /estadisticas/index.html`

- **Protocolo usado para comunicarse con el recurso**  
(cada protocolo define su propio formato de URL)
- **Dirección de Recurso en Internet**
  - **Dirección del Servidor**
    - Dirección IP (Internet Protocol) del servidor  
(170.70.84.76)
    - Nombre de Dominio del Servidor (www.banxico.org.mx)
  - **Número de Puerto en el Servidor**  
Opcional : Cada protocolo tiene un número de puerto por default
- **Detalles del Recurso (específico al protocolo)**
  - (directorio y nombre de archivo de una página HTML)



# Mensajes en HTTP

- Si bien HTTP envía varios mensajes que no son mostrados normalmente al usuario, un entendimiento básico de los mismos puede ser muy útil
- Cada mensaje enviado, ya sea una solicitud o una respuesta, tiene tres partes:
  1. Línea de solicitud o respuesta
  2. Sección de encabezado
  3. Cuerpo del mensaje

# Trabajo realizado por el cliente (1)



- El cliente envía un mensaje al servidor en un puerto particular (80 por default)
- La primer parte del mensaje es la **línea de solicitud**, que contiene:
  - Un **método** (comando HTTP) tal como **GET** o **POST**
  - La dirección del documento y
  - Un número de versión de HTTP
- Ejemplo:
  - **GET /index.html HTTP/1.1**



# Otros métodos

- Existen otros métodos aparte de **GET** y **POST** que son:
  - **HEAD**: Igual a **GET**, pero solicita que *solamente* un encabezado sea regresado
  - **PUT**: Solicita almacenar el cuerpo de la entidad en el URL
  - **DELETE**: Solicita la eliminación de los datos en el URL
  - **LINK**: Solicita que la información del encabezado sea asociada a un documento en el servidor
  - **UNLINK**: Solicita deshacer una solicitud **LINK**
  - **OPTIONS**: Solicita información sobre las opciones de comunicación en el servidor
  - **TRACE**: Solicita que el cuerpo de la entidad sea regresado tal como se recibe (usado para depurar)

# Trabajo realizado por el cliente (2)



- La segunda parte de una solicitud es la **información de encabezado** opcional, tal como:
  - Que software es el cliente
  - Que formatos puede aceptar
- Toda la información está en la forma:  
**Nombre: Valor**
- Ejemplo:  
**User-Agent: Mozilla/2.02Gold (WinNT; I)**  
**Accept: image/gif, image/jpeg, \*/\***
- Una *línea en blanco* termina el encabezado

# Encabezados de Solicitud del Cliente



- **Accept:** *tipo/subtipo, tipo/subtipo, ...*
  - Especifica los tipos de medios que el cliente prefiere aceptar
- **Accept-Language:** *en, fr, de*
  - Lenguaje preferido (Ingles, Frances, Aleman)
- **User-Agent:** *string*
  - Descripción del navegador u otro programa cliente que envía la solicitud
- **From:** *rsolis@acm.org*
  - Dirección de e-mail del usuario del programa cliente
- **Cookie:** *nombre=valor*
  - Información sobre una cookie para el URL
  - Múltiples cookies pueden estar separadas por comas

# Trabajo realizado por el cliente (3)



- La tercera parte de una solicitud (después de la línea en blanco) es el **cuerpo de la entidad**, que contiene datos opcionales
  - La parte del cuerpo de la entidad es usada mayormente en solicitudes **POST**
  - La parte del cuerpo de la entidad está siempre vacía para una solicitud **GET**

# Trabajo realizado por el servidor (1)



- La respuesta del servidor esta también en tres partes
- La primer parte es la línea de status, que especifica:
  - La versión de HTTP
  - Un código de status
  - Una breve descripción de lo que significa el código de status
- Ejemplo: **HTTP/1.1 404 Not Found**
- Los códigos de status se dividen en grupos:
  - 100-199** De Información
  - 200-299** La solicitud fue exitosa
  - 300-399** La solicitud fue redireccionada
  - 400-499** La solicitud falló
  - 500-599** Ocurrió un error en el servidor





# Códigos de status comunes

- **200 OK**
  - Todo funcionó bien, aquí están los datos
- **301 Moved Permanently**
  - URL fue movido, pero aquí esta la nueva dirección
- **302 Moved temporarily**
  - URL temporalmente fuera de servicio, manten la anterior dirección pero usa esta dirección por ahora
- **400 Bad Request**
  - Hay un error de sintaxis en la solicitud
- **403 Forbidden**
  - No puedes hacer esto, y no te diremos por qué
- **404 Not Found**
  - No existe tal documento

# Trabajo realizado por el servidor (2)



- La segunda parte de la respuesta es la información del encabezado, terminada por una línea en blanco
- Ejemplo:  
HTTP/1.0 200 OK  
Date: Thu, 07 Jun 2012 17:24:44 GMT  
Server: Apache  
Content-Language: en  
Vary: Accept-Encoding, Cookie  
Last-Modified: Wed, 06 Jun 2012 13:39:43 GMT  
Content-Encoding: gzip  
Content-Length: 24522  
Content-Type: text/html; charset=UTF-8  
Connection: keep-alive
- Visor de encabezados: <http://www.delorie.com/web/headers.html>

También puede usar el add-on **LiveHTTPHeaders** de Firefox ó **HTTPHeaders** de Chrome

# Encabezados de respuesta del servidor



- **Server: NCSA/1.3**
  - Nombre y versión del servidor
- **Content-Type: *tipo/subtipo***
  - Debería ser de uno de los tipos y subtipos especificados por el encabezado **Accept** del cliente
- **Set-Cookie: *nombre=valor; opciones***
  - Solicita al cliente almacenar una cookie con el nombre y valor dados

# Trabajo realizado por el servidor (3)



- La tercer parte de la respuesta del servidor es el cuerpo de la entidad
- Esto es a menudo una página HTML
  - Pero puede ser también un jpeg, un gif, texto plano, etc. Cualquier cosa que el navegador (u otro cliente) esté preparado para aceptar

# HTTP es un protocolo sin estado (1)

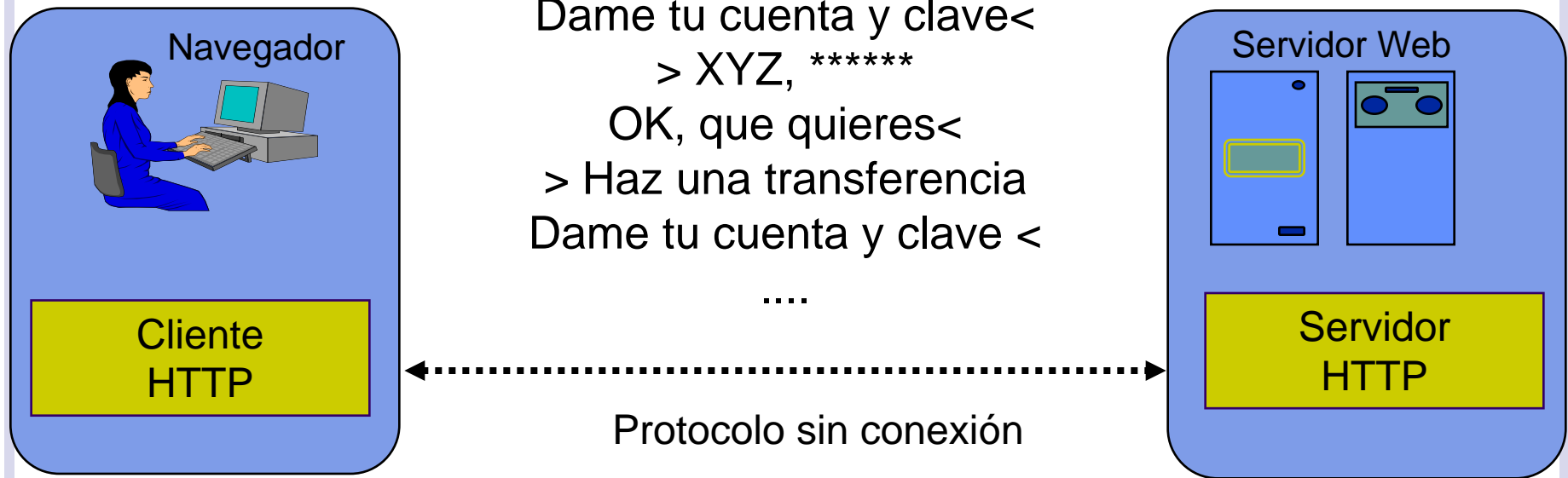


- Debido a la naturaleza de HTTP (Conexion, Solicitud, Respuesta, Desconexión) se dice que HTTP es un protocolo sin estado
  - Al moverse de una página web a otra no hay nada en el protocolo que permita a un programa web mantener el “estado” del programa.
  - El “estado” puede mantenerse haciendo trucos si es que es necesario

# HTTP es un protocolo sin estado (2)



- **Beneficio:** Rendimiento del Servidor
- **Costo:** Servidor no puede mantener información sobre el usuario



# Manteniendo el “estado” del programa (1)

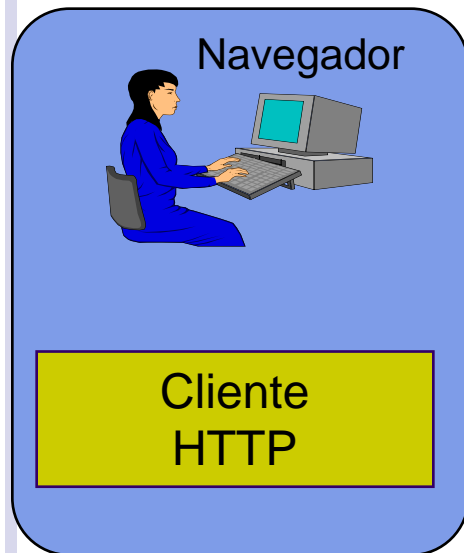


- Variables ocultas (<input type=hidden>
- Sesiones
  - Etiquetas de encabezado especiales que son interpretadas por el servidor
    - Usadas por ASP, PHP, JSP
      - Implementadas en el lenguaje al nivel de API

# Manteniendo el “estado” del programa (2)

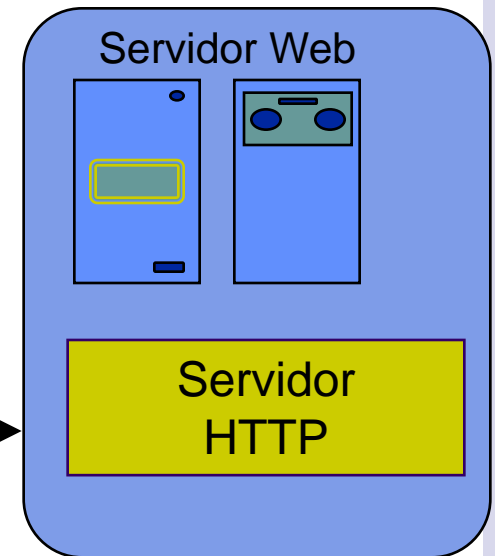


El estado puede mantenerse con mensajes que viajan entre el servidor y el navegador



> Quiero conectarme  
Dame tu cuenta y clave<  
> XYZ, \*\*\*\*\*  
OK, tu llave es 1067<  
> Muestrame mi saldo,  
mi llave es 1067  
Aqui esta tu saldo, \$\$\$\$ <

.....  
Protocolo sin conexión







Introducción a las Tecnologías Web

# HTML



# ¿Qué es HTML?

- HTML (HyperText Markup Language) es el conjunto de símbolos ó códigos de marcado que se colocan en un archivo y cuya finalidad es ser desplegados en un navegador de páginas.
- Fue creado por Tim Berners-Lee en CERN
- Es un estándar abierto desarrollado bajo la supervisión del World Wide Web Consortium (<http://w3c.org>)





# Elementos de HTML

- Cada código de marcado individual se conoce como **elemento** o **etiqueta(tag)**.
- Cada etiqueta tiene un propósito.
- Las etiquetas están encerradas entre los símbolos "<" y ">".
- La mayoría de las etiquetas aparece en pares; una etiqueta de apertura y una de terminación.



# Historia de HTML

- 1991 – Aparece la primer descripción de HTML
- 1995 – Se publica el HTML 2.0 por parte de la IETF (RFC 1866)
- 1996 – Se publica HTML 3.2 (primera version desarrollada y estandarizada por W3C)
- 1997 – Se publica HTML 4.0, la cual separa el contenido de la presentación y ofrece tres variaciones: Strict (se prohíben elementos obsoletos), Transitional (se permiten elementos obsoletos) y Frameset(solo elementos relacionados con frames son permitidos)
- 1999 – Se publica HTML 4.01
- 2008 – Se comienza a trabajar en HTML5



# Formato de una Página HTML

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE> Mi primer pagina web! </TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <H1> Hola Mundo!!! </H1>
```

```
    <HR>
```

```
    <!-- Resto de pagina. Esto es un comentario. -->
```

```
  </BODY>
```

```
</HTML>
```



# Problemas con HTML

- HTML comenzó como una forma de describir la *estructura* de los documentos, con etiquetas para indicar encabezados, párrafos, etc.
- Debido a que las personas querían controlar la *apariciencia* de los documentos, HTML empezó a agregar atributos a las etiquetas para controlar tipos de letra, alineación, etc.
- El resultado fue un lenguaje de marcado que podía hacer ambas cosas (estructura y presentación), pero no era muy bueno en ninguna



# ¿Qué es XHTML?

- **XHTML** es por Extensible Hypertext Markup Language
- El fin de XHTML es reemplazar HTML
- XHTML es casi idéntico a HTML 4.01
- XHTML es una versión mas estricta y limpia de HTML
- **XML** (Extensible Markup Language) es un lenguaje de marcado diseñado para describir *datos*
- XHTML es HTML redefinido como una aplicación de XML
- XHTML es un “puente” entre HTML y XML



# HTML vs XML

- XML se parece mucho a HTML, pero ...

- HTML usa un conjunto fijo de etiquetas
- HTML está diseñado para desplegar datos a los humanos
- Los navegadores son muy tolerantes a errores en HTML
- Todos los navegadores pueden desplegar HTML

- Con XML puede hacer sus propias etiquetas (y definir lo que significan en un documento separado)
- XML está diseñado para describir datos a las computadoras
- Los documentos XML deben estar bien formados (sintácticamente correctos)
- La mayoría de los navegadores modernos pueden desplegar XML

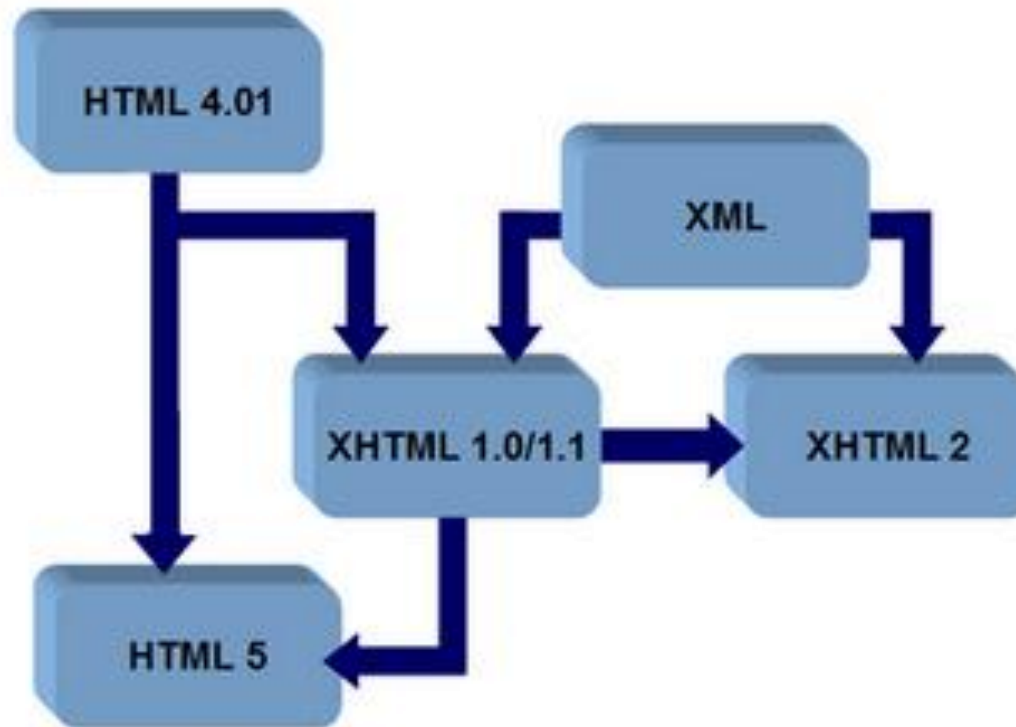




# Historia de XML/XHTML

- 1998 – Surge el estándar XML para escribir lenguajes Web
- 2000 – Se publica XHTML 1.0, el cual ofrece las mismas tres variaciones que HTML 4.0
- 2001 – Se publica XHTML 1.1, el cual esta basado en la versión Strict de XHTML 1.0 con cambios menores
- 2002 – Aparece XHTML 2.0, el cual no es un estándar y es incompatible con XHTML 1.x
- Actualmente se trabaja en XHTML5, el cual está siendo definido junto con HTML5

# Relación entre XHTML, HTML y XML





# De HTML a XHTML(1)

- Los elementos XHTML deben estar apropiadamente anidados  
`<b><i>negrita e italica</b></i>` es *incorrecto*
- Los documentos XHTML deben estar bien formados  
`<html>  
<head> ... </head>  
<body> ... </body>  
</html>`
- Los nombres de etiquetas deben estar en minúscula
- *Todos* los elementos XHTML deben estar cerrados
  - Si una etiqueta HTML no es un contenedor, se cierra como en los siguientes ejemplos:  
`<br />, <hr />, <image src="mifoto.gif" />`



# De HTML a XHTML(2)

- Los nombres de atributos deben ir en minúsculas
  - Ejemplo: `<table width="100%">`
- Los valores de atributos deben ir entre comillas
  - Ejemplo: `<table width="100%">`
- La minimización de atributos está prohibida
  - Ejemplo: `<frame noresize="noresize">`,  
no puede ser abreviada como `<frame noresize>`
- El atributo `id` reemplaza el atributo `name`
  - Incorrecto: ``
  - Correcto: ``
  - Lo mejor: ``



# Atributos en XHTML

- Solo se usan en la etiqueta de apertura
- Tres tipos:
  - Opcionales: Varían con el tipo de elemento
  - Estándar: id, class, title, style, dir, lang, xml:lang
  - De eventos: onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
    - Usados cuando se manejan scripts



# SGML y DTDs

- SGML es por “Standard Generalized Markup Language”
- HTML, XHTML, XML y muchos otros lenguajes de marcado están definidos en SGML
- Un DTD o “Document Type Definition” describe la sintaxis a usar para el documento actual
- Existen tres diferentes DTDs para XHTML y puedes seleccionar el que desees
  - Estos DTDs son *publicos* y están en la web
  - Debes comenzar tu documento XHTML con una referencia a uno de estos DTDs



# Declaración DOCTYPE (1)

- Todo documento XHTML debe comenzar con una de las declaraciones **DOCTYPE** (DTDs):
  - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd`
  - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd`
  - `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd`
- Posteriormente se indica el espacio de nombres `<html xmlns="http://www.w3.org/1999/xhtml">`



# Declaración DOCTYPE (2)

- Los tres DTDs se describen a continuación:
  - **Strict**
    - Se usa para un marcado realmente limpio, sin información de despliegue (información de tipos de letra, color, o tamaño)
    - Se usa con CSS (Cascading Style Sheets) si desea definir como debe lucir el documento
  - **Transitional**
    - Se usa con HTML estándar y/o con CSS
    - Permite el uso de elementos HTML obsoletos
  - **Frameset**
    - Se usa si el documento usa frames de HTML





# Ejemplo de XHTML

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />
    <title>Documento simple</title>
  </head>
  <body>
    <h1>Mi primer pagina</h1>
    <p>Parrafo simple</p>
  </body>
</html>
```



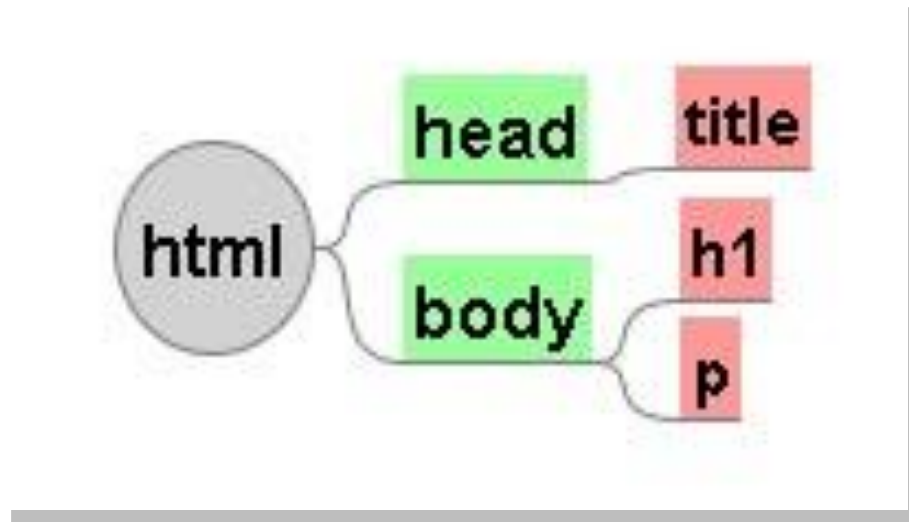
# Herramientas

- La herramienta de validación de W3C HTML <http://validator.w3.org/> es una forma en HTML para verificar (no arreglar) documentos HTML y XHTML

# Estructura Jerárquica en XHTML



- Los archivos XHTML bien formados forman una jerarquía





# Secciones head y body

- **Sección head**

Contiene información que describe el documento de la página web

`<head>`

*...la sección head va aquí*

`</head>`

- **Sección body**

Contiene el texto y los elementos que se despliegan en el documento de la página web

`<body>`

*...la sección body va aquí*

`</body>`



# Elemento de Encabezado

## Heading Level 1

`<h1>Heading Level 1</h1>`

`<h2>Heading Level 2</h2>`

`<h3>Heading Level 3</h3>`

`<h4>Heading Level 4</h4>`

`<h5>Heading Level 5</h5>`

`<h6>Heading Level 6</h6>`

## Heading Level 2

### Heading Level 3

#### Heading Level 4

##### Heading Level 5

###### Heading Level 6



# Etiqueta <p>

- Elemento de párrafo

<p> ...*el párrafo va aquí*... </p>

- Agrupa sentencias y secciones de texto.
- Configura una línea en blanco arriba y abajo del párrafo



# Etiqueta `<br />`

- Elemento de Avance de Línea
  - Etiqueta independiente

*...texto va aquí<br />*

*Esto comienza en una nueva línea....*

- Causa que el siguiente elemento o texto se despliegue en una nueva línea



# Etiqueta `<hr />`

- Configura una línea horizontal





# Etiqueta `<blockquote>`

- Elemento `blockquote`
  - Indenta un bloque de texto para su énfasis

`<blockquote>`

*...aquí va el texto...*

`</blockquote>`



# Listas en XHTML

- Lista de definiciones
- Lista Ordenada
- Lista Sin Orden



# Lista de Definiciones

- Util para desplegar una lista de términos y definiciones o una lista de preguntas y respuestas (FAQ)
  - Etiqueta `<dl>`  
Contiene la lista de definiciones
  - Etiqueta `<dt>`  
Contiene un termino definido  
Configura un avance de linea abajo y arriba del texto
  - Etiqueta `<dd>`  
Contiene una definición de datos o descripción. Indenta el texto

# Ejemplo de Lista de Definiciones



**<dl>**

**<dt>IP</dt>**

**<dd>Internet Protocol</dd>**

**<dt>TCP</dt>**

**<dd>Transmission Control Protocol</dd>**

**</dl>**

**IP**

Internet Protocol

**TCP**

Transmission Control Protocol



# Lista Ordenada

- Proporciona información de manera ordenada
  - Etiqueta `<ol>`  
Contiene la lista ordenada
    - El atributo `type` determina el esquema de numeración de la lista, por default son números
  - Etiqueta `<li>`  
Contiene un ítem de la lista



# Ejemplo de Lista Ordenada

**<ol>**

**<li>Pagar derecho de inscripcion</li>**

**<li>Entregar ficha de deposito</li>**

**<li>Llenar hoja de carga</li>**

**<li>Llevar hoja de carga para su  
autorizacion</li>**

**</ol>**

1. Pagar derecho de inscripcion
2. Entregar ficha de deposito
3. Llenar hoja de carga
4. Llevar hoja de carga para su autorizacion



# Lista Sin Orden

- Despliega la información con viñetas
  - Etiqueta `<ul>`  
Contiene la lista sin orden
    - El atributo `type` determina el tipo de viñeta, por default es un disco (aunque esto depende del navegador utilizado)
  - Etiqueta `<li>`  
Contiene un ítem de la lista



# Ejemplo de Lista Sin Orden

**<ul>**

**<li>TCP</li>**

**<li>IP</li>**

**<li>HTTP</li>**

**<li>FTP</li>**

**</ul>**

- TCP
- IP
- HTTP
- FTP



# Elementos de Indicación de Estilo Lógico en XHTML



- Indican el estilo lógico del texto a desplegar
- Etiquetas comunes:
  - `<strong></strong>`
    - Causa que el texto sea enfatizado o que “resalte” del texto que se encuentra alrededor. Normalmente lo hace en negrita.  
`<strong>Esto es importante</strong>`
  - `<em></em>`
    - Causa que el texto sea enfatizado en relación al otro texto en la página. Normalmente lo hace en italico.  
`<em>Favor de notar que ...</em>`

# Elementos de Indicación de Estilo Físico en XHTML



- Proporcional al navegador información del tipo de letra a usar
  - Util para los navegadores– pero no siempre aplicable a otros dispositivos o agentes de usuario
- Etiquetas comunes
  - `<b></b>`
    - Despliegan el texto en negrita  
`<b>Esto es importante</b>`
  - `<i></i>`
    - Despliegan el text en itálica  
`<i>Favor de notar que...</i>`

# Caracteres Especiales en XHTML



- Despliegan caracteres especiales tales como comillas, el símbolo de copyright, etc.

Caracter	Código
©	&copy;
<	&lt;
>	&gt;
&	&amp;
	&nbsp;



# Etiqueta <a> (1)

- Denominado elemento ancla (anchor)
  - Especifica una referencia hiperenlace (href) a un archivo
  - El texto entre <a> y </a> se despliega en la página web.

**<a href="contacto.html">Contactanos</a>**

- Atributo href
  - Indica el nombre del archivo o URL  
Documento de página web, foto, pdf, etc.



# Etiqueta <a> (2)

- Enlace absoluto

- Enlace a otros sitios web

**<a href="http://yahoo.com">Yahoo</a>**

- Enlace relativo

- Enlace a páginas en tu propio sitio

**<a href="index.html">Home</a>**

# Enlaces internos usando la etiqueta <a>

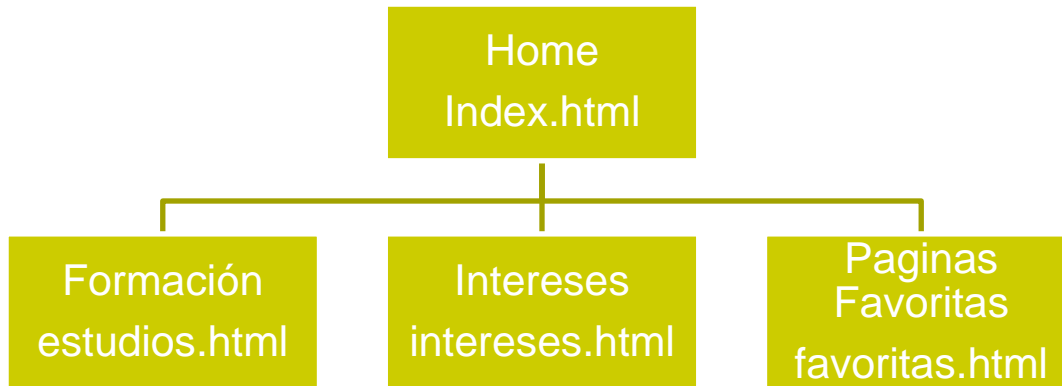


- Enlace a una parte/sección de una página web
- También denominados bookmarks, fragmentos con nombre, anclas con nombre
- Dos componentes:
  1. La etiqueta de ancla que identifica un bookmark o fragmento con nombre de una página web. Esto requiere dos atributos: el atributo id y el atributo name.  
**<a name="top" id="top"></a>**
  2. La etiqueta ancla que enlaza al bookmark o fragmento con nombre de una página web, usando el atributo href.  
**<a href="#top">Volver al Inicio</a>**



# Ejercicio

- Realizar cuatro páginas con la siguiente jerarquía:



- Todas las páginas deben tener un título en la sección head y un título con estilo `<h1>` en la sección body, debajo del cual debe haber un párrafo con el texto que deseen en negrita e itálica.
- Deben ser páginas XHTML transicional válidas

- index.html tendrá un enlace a cada una de las otras tres páginas
- estudios.html tendrá una lista de definiciones donde indique donde estudio la primaria, secundaria, etc
- intereses.html tendrá una lista ordenada con sus intereses personales
- favoritas.html tendrá una lista sin orden con al menos 5 páginas favoritas que tenga (con hiperenlaces a cada una de estas)



# Etiqueta <img>

- Configura imágenes graficas en una página web

```

```

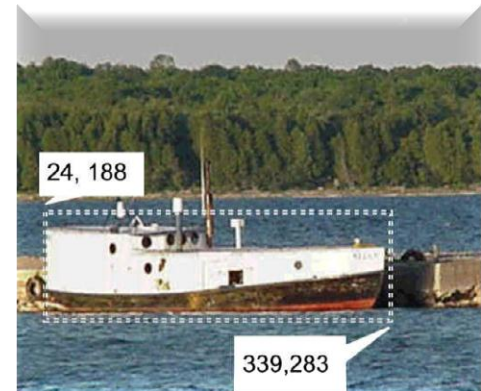
- Atributo src
  - Nombre del archivo que contiene el gráfico
- Atributo alt
  - Configura un contenido de texto alternativo(descripción)
- Atributo height
  - Altura del gráfico en pixeles
- Atributo width
  - Anchura del gráfico en pixeles
- Una imagen puede ser convertida en hiperenlace al ponerlo dentro de una etiqueta <a>



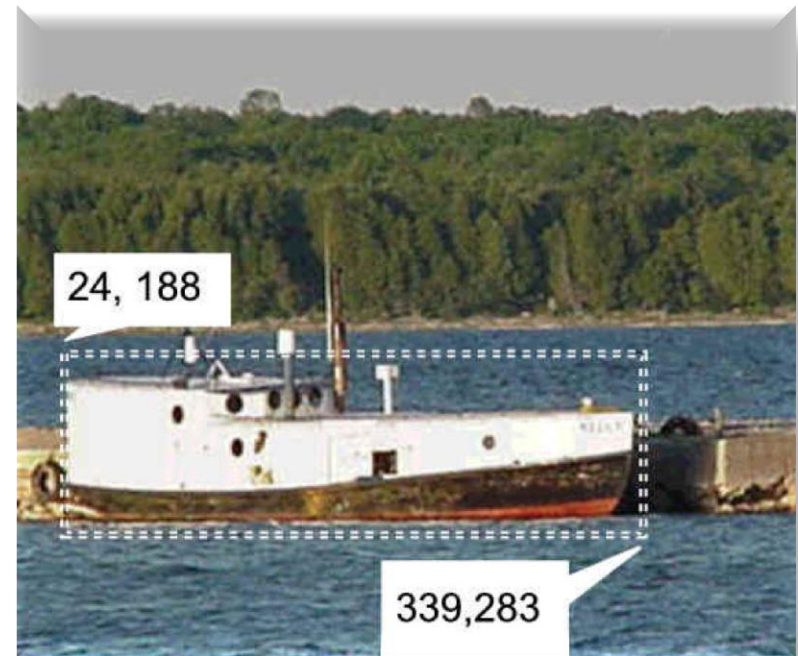


# Mapas (1)

- Elemento <map>
  - Define el mapa
- Elemento <area>
  - Define un area específica en el mapa
  - Puede ser establecida a un rectangulo, circulo o polígono
    - Atributo href
    - Atributo shape
    - Atributo coords



# Mapas (2)



```
<map name="bote" id="bote">
<area href="http://www.bote.com" shape="rect"
  coords="24, 188, 339, 283" alt="bote de pesca" />
</map>

```



# Tablas en XHTML

- Compuestas de renglones y columnas.
- Cada celda individual de una tabla se encuentra en la intersección de un renglón y columna específicos.
- Se usan las etiquetas `<table>`, `<tr>` y `<td>` (y opcionalmente `<th>`)



# Elementos de una tabla

- Etiqueta `<table>`  
Contiene la tabla  
Atributos comunes: border, width, align
- Etiqueta `<tr>`  
Contiene un renglón de la tabla
- Etiqueta `<td>`  
Contiene una celda de la tabla
- Etiqueta `<th>`  
Contiene una celda de encabezado de la tabla



# Ejemplo de Tabla XHTML

```
<table border="1">
  <tr>
    <th>Nombre</th>
    <th>Fecha de Nacimiento</th>
  </tr>
  <tr>
    <td>Juan</td>
    <td>11/08</td>
  </tr>
  <tr>
    <td>Karla</td>
    <td>4/17</td>
  </tr>
  <tr>
    <td>Salvador</td>
    <td>11/28</td>
  </tr>
</table>
```



# Atributos Comunes de Tabla

- align
- bgcolor
- border
- bordercolor (no es de W3C)
- cellpadding
- cellspacing
- summary
- title
- width



# Atributos Comunes de Celdas

- align
- bgcolor
- colspan
- rowspan
- valign
- width



# Atributo colspan

```
<table border="1">
  <tr>
    <th colspan="2">Lista de Cumple</th>
  </tr>
  <tr>
    <td>Juan</td>
    <td>11/08</td>
  </tr>
  <tr>
    <td>Karla</td>
    <td>4/17</td>
  </tr>
  <tr>
    <td>Salvador</td>
    <td>11/28</td>
  </tr>
</table>
```





# Formas en XHTML

## ● Form

- Etiqueta de XHTML que contiene y organiza otros objetos tales como
  - Cajas de texto
  - Botones
  - Cajas combo

y que acepta información de los visitantes al sitio web.

# Dos Componentes en las Formas



## 1. La forma XHTML

-- interface de usuario en la página web

y

## 2. El procesamiento del lado del servidor

**PHP**

**ASP**

**JSP**

**etc.**



# Uso de formas XHTML

- Etiqueta `<form>`
  - Contiene los elementos de la forma en la página web
  - Etiqueta contenedora
- Etiqueta `<input />`
  - Configura una variedad de elementos de forma incluyendo cajas de texto, radio buttons, check boxes, y botones
  - Etiqueta individual
- Etiqueta `<textarea>`
  - Configura una caja de texto con varias líneas
  - Etiqueta contenedora
- Etiqueta `<select>`
  - Configura un select box (lista drop down)
  - Etiqueta contenedora
- Etiqueta `<option>`
  - Configura un opción en el select box
  - Etiqueta contenedora



# Forma XHTML de ejemplo

```
<form>  
  Email: <input type="text" name="EMail" id="EMail" />  
  <br />  
  <input type="submit" />  
</form>
```

E-mail:



# Etiqueta <form> de XHTML

- Atributos de la etiqueta form :
  - action
    - Especifica el programa o script del lado del servidor que procesará los datos de su forma
  - method
    - get – valor por default, datos de la forma pasados en el URL
    - post – más segura, datos de la forma pasados en el cuerpo de la Entidad HTTP
  - name
    - Identifica la forma
  - id
    - Identifica la forma

# Etiqueta <input> (Caja de texto)



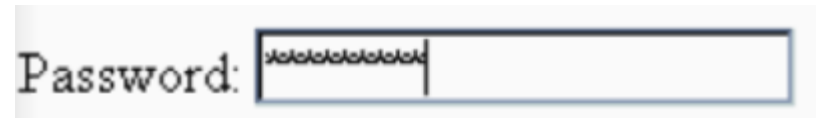
- Acepta información de texto
- Atributos:
  - type="text"
  - name
  - id
  - size
  - maxlength
  - value

E-mail:

# Etiqueta <input> (Password box)



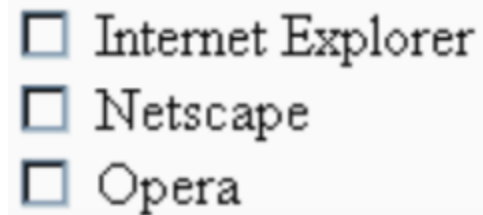
- Acepta información de texto que necesita ser ocultada conforme es introducida
- Atributos:
  - type="password"
  - name
  - id
  - size
  - maxlength
  - value





# Etiqueta <input> (Check box)

- Permite al usuario seleccionar uno o más de un grupo de items predeterminados
- Atributos:
  - type="checkbox"
  - name
  - id
  - checked
  - value

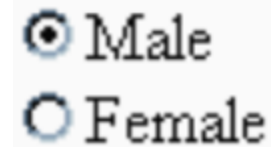




# Etiqueta <input> (Radio button)



- Permite al usuario seleccionar exactamente uno de un grupo de items predeterminados
- A cada radio button en un grupo se le da el mismo nombre y un valor único
- Atributos:
  - type="radio"
  - name
  - id
  - checked
  - value





# Etiqueta <textarea>

- Configura una caja de texto con varias líneas y scrollbars
- Atributos:
  - name
  - id
  - cols
  - rows



# Etiqueta <select>

- Configura una lista de selección (con ayuda de etiquetas <option>)
- También conocido como: Select Box, Drop-Down List, Drop-Down Box, y Option Box.
- Permite al usuario seleccionar uno o más items de una lista de opciones preseleccionadas.
- Atributos:
  - name
  - id
  - size
  - multiple



# Etiqueta <option>

- Configura las opciones en una lista de selección
- Atributos:
  - value
  - selected

# Etiqueta <input> Botón de Enviar (Submit)



- Envía la información de la forma
- Cuando se le da click:
  - Se activa el método **action** de la etiqueta **<form>**
  - Envía los datos de la forma (el par nombre=valor para cada elemento de la forma) al servidor web.
- Atributos:
  - type="submit"
  - name
  - id
  - value



# Etiqueta <input> (Boton Reset)

- Reestablece los campos de la forma a sus valores iniciales
- Atributos:
  - type="reset"
  - name
  - id
  - value

# Etiqueta <input> (Datos ocultos)



- Este elemento de la forma *no* es desplegado en la página web.
- Campos ocultos de la forma
  - Pueden ser accedidos con scripts del lado del cliente y del lado del servidor
  - Algunas veces se usan para contener información necesaria conforme el visitante se mueve de una página a otra.
- Atributos:
  - type="hidden"
  - name
  - id
  - value



# Etiqueta <label>

- Asocia una etiqueta de texto a un elemento de la forma
- Dos diferentes formatos:

```
<label>Email: <input type="text"
    name="EMail"    id ="Email" /></label>
```

ó

```
<label for="EMail">Email: </label>
<input type="text" name="EMail" id="Email"
/>
```





## Introducción a las Tecnologías Web

# CSS



# ¿Para qué sirve CSS?

- Las CSS (Cascade Style Sheets):
  - Estan en un lenguaje flexible y multiplataforma, basado en estándares, desarrollado por W3C
  - Son usadas para controlar como se presentan los elementos en una página web
  - Usan una sintaxis diferente a la de HTML/XHTML
  - Funcionan con los navegadores mas comunes (Internet Explorer, FireFox, Opera)
  - Para una demostración de lo que se puede lograr, entre a <http://www.csszengarden.com>



# Historia de CSS

- 1996 – Se publica la recomendación CSS 1, por parte de W3C, la cual incluye capacidades de manejo de tipos de letra, colores, alineación, márgenes e identificación y clasificación de atributos
- 1998 – Se publica la recomendación CSS 2, que incluye mejoras como el posicionamiento de elementos
- 2004 – Se comienza a trabajar en CSS 2.1, la cual corrige errores de CSS 2
- 2011 – Para tres de los módulos de CSS 3, se publican recomendaciones (a diferencia de las versiones anteriores que eran una sola especificación, en CSS 3, el trabajo se divide en documentos separados denominados módulos)



# Ventajas de CSS

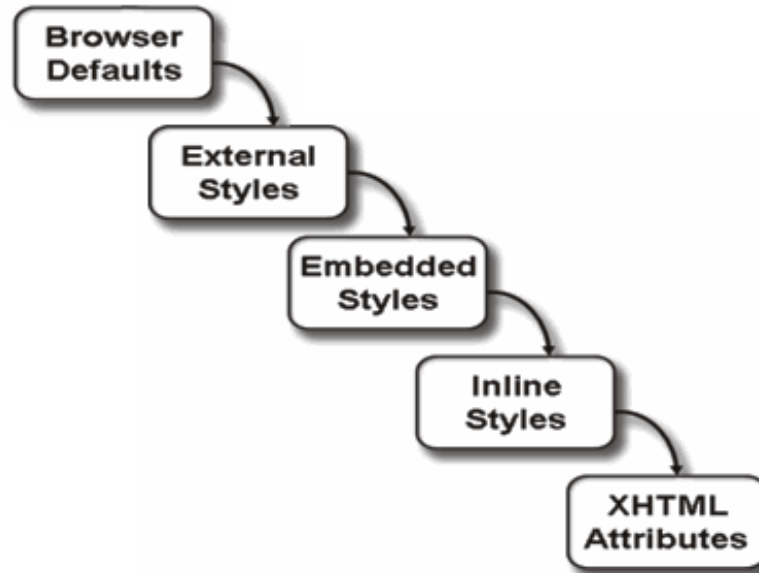
- Mayor tipografía y control de la distribución de la página
- El estilo se separa de la estructura
- Los estilos pueden ser almacenados en un documento separado al cual se puede enlazar desde cualquier pagina web
- Los documentos son potencialmente más pequeños
- Mantenimiento mas fácil del sitio



# Tipos de CSS

- **Estilos en línea (inline)**
  - Se configuran en la sección body de la página web
  - Usan el atributo style de una etiqueta XHTML
  - El estilo se aplica solo al elemento en cuestión
- **Estilos incrustados (embedded)**
  - Se configuran en la sección header de la página web.
  - Usan el elemento <style> de XHTML
  - El estilo se aplica al documento en su totalidad
- **Estilos externos**
  - Se configuran en un archivo de texto separado con extensión .css
  - El elemento XHTML <link /> en la sección header de una página web asocia a tal página con el archivo .css

# ¿Por qué Cascada?



- Los estilos se aplican por un orden de precedencia.
- Estilos globales a nivel de sitio se configuran en CSS externos.
- Los estilos pueden ser sobreescritos para cuando sea necesario para una página o elemento específico



# Sintaxis de CSS

- Las hojas de estilo estan compuestas de reglas que describen el estilo a ser aplicado.
- Cada regla contiene un selector y una declaración, la cual consta de una propiedad y un valor

Selector	Propiedad	Valor
p	{font-family:	times; }

Note la puntuación: La propiedad es seguida del símbolo de dos puntos (:) y el valor se seguido por un punto y coma (;)



# Ejemplo de CSS

- Se desea una configuración para que la pagina web se despliegue con texto azul y fondo amarillo.

```
body { color: blue;  
        background-color: yellow; }
```

- Esto se pudo haber escrito usando valores de color en hexadecimal como se muestra a continuación.

```
body { color: #0000ff;  
        background-color: #ffff00; }
```



# Propiedades Comunes de Formateo en CSS



- background-color
- color
- font-family
- font-size
- font-style
- font-weight
- line-height
- margin
- text-align
- text-decoration
- width

# Uso de Colores en Páginas Web (1)



- El monitor de una computadora despliega el color como intensidades de rojo, verde y azul
- Color RGB
- Los valores de rojo, verde y azul varían de 0 a 255.
- Los números hexadecimales (base 16) representan a estos valores de color.

#FFFFFF	#FFFFCC	#FFFF99	#FFFF66	#FFFF33	#FFFF00
#FFCCFF	#FFCCCC	#FFCC99	#FFCC66	#FFCC33	#FFCC00
#FF99FF	#FF99CC	#FF9999	#FF9966	#FF9933	#FF9900

# Uso de Colores en Páginas Web (2)



- # se usa para indicar un valor en hexadecimal
- El valor para cada parte del color va de 00 a ff
- Tres valores hexadecimales describen un color RGB

**#000000** negro

**#ff0000** rojo

**#0000ff** azul

**#00FF00** verde

#FFFFFF	#FFFFCC	#FFFF99	#FFFF66	#FFFF33	#FFFF00
#FFCCFF	#FFCCCC	#FFCC99	#FFCC66	#FFCC33	#FFCC00
#FF99FF	#FF99CC	#FF9999	#FF9966	#FF9933	#FF9900



# Paleta de Colores Web

- Colección de 216 colores
- Son los que despliegan mas similares en las diferentes plataformas (Mac, PC)
- Valores en hexadecimal: 00, 33, 66, 99, cc, ff

Tabla de Colores en:  
<http://webdevfoundations.net/color>

Web Safe Color Chart

#FFFFFF	#FFFFCC	#FFFF99	#FFFF66	#FFFF33	#FFFF00
#FFCCFF	#FFCCCC	#FFCC99	#FFCC66	#FFCC33	#FFCC00
#FF99FF	#FF99CC	#FF9999	#FF9966	#FF9933	#FF9900
#FF66FF	#FF66CC	#FF6699	#FF6666	#FF6633	#FF6600
#FF33FF	#FF33CC	#FF3399	#FF3366	#FF3333	#FF3300
#FF00FF	#FF00CC	#FF0099	#FF0066	#FF0033	#FF0000
#CCFFFF	#CCFFCC	#CCFF99	#CCFF66	#CCFF33	#CCFF00
#CCCCFF	#CCCCCC	#CCCC99	#CCCC66	#CCCC33	#CCCC00
#CC99FF	#CC99CC	#CC9999	#CC9966	#CC9933	#CC9900
#CC66FF	#CC66CC	#CC6699	#CC6666	#CC6633	#CC6600
#CC33FF	#CC33CC	#CC3399	#CC3366	#CC3333	#CC3300
#CC00FF	#CC00CC	#CC0099	#CC0066	#CC0033	#CC0000
#99FFFF	#99FFCC	#99FF99	#99FF66	#99FF33	#99FF00
#99CCFF	#99CCCC	#99CC99	#99CC66	#99CC33	#99CC00
#9999FF	#9999CC	#999999	#999966	#999933	#999900
#9966FF	#9966CC	#996699	#996666	#996633	#996600
#9933FF	#9933CC	#993399	#993366	#993333	#993300
#9900FF	#9900CC	#990099	#990066	#990033	#990000

# Configuración de Color con CSS inline



- Ejemplo 1: Configurar el texto en color rojo para un elemento `<h1>` :

```
<h1 style="color:#ff0000">Texto en rojo </h1>
```

- Ejemplo 2: Configurar el texto en color rojo para un elemento `<h1>` con fondo gris

```
<h1 style="color:#ff0000;background-color:#cccccc">  
  Texto rojo con fondo gris</h1>
```



# Estilos CSS Incrustados (1)

- Configurados en la sección header de una página web.
- Se aplican a todo el documento
- Las declaraciones de estilo están contenidas en etiquetas<style>
- El atributo de tipo indica el tipo MIME text/css
- Ejemplo: Configurar una pagina web para que tenga texto blanco con fondo negro

```
<style type="text/css">
body { background-color:#000000;
      color:#ffffff;
}
</style>
```



# Estilos CSS Incrustados (2)

```
<style type="text/css">
body { background-color:
    #e6e6fa;
        color: #191970;}
h1 { background-color: #191970;
    color: #e6e6fa;}
h2 { background-color: #aeaed4;
    color: #191970;}
</style>
```

- El selector **body** establece las reglas de estilo globales para toda la página.
- Estas reglas globales son sobreescritas para los elementos `<h1>` y `<h2>` por las reglas de estilo **h1** y **h2**.

# Configuración de Texto con CSS



- Las propiedades CSS para configurar texto:
  - font-weight
    - Configura que tan negrita es la letra
  - font-style
    - Configura el text a un estilo italico
  - font-size
    - Configura el tamaño del texto
  - font-family
    - Configura el tipo de letra del texto





# Propiedad font-size

Text Values	Em Units	Px Units	Pt Units	Percentage
xx-small	.5 em	10 px	6 pt	50%
x-small	.60 em	11 px	8 pt	60%
small	.75 em	13 px	10 pt	75%
medium	1 em	16 px	12 pt	100%
x-large	1.5 em	24 px	18 pt	150%
xx-large	2 em	28 px	24 pt	200%

- Recomendación de Accesibilidad:
  - Use tamaños de letra en em o en porcentajes – estos pueden ser fácilmente agrandados en todos los navegadores por los usuarios



# Propiedad font-family

- No todos tienen las mismas fuentes (tipos de letra) instaladas en su computadora
- Configura una lista de tipos de letras e incluye un nombre genérico de familia

**p {font-family: Arial, Verdana, sans-serif;}**

Font-family Category	Font Typeface
serif	Times New Roman, Georgia, Times
sans-serif	Arial, Verdana, Geneva
monospace	Courier New, Lucida Console
cursive	Brush Script MT, Comic Sans MS
fantasy	Jokerman, Curlz MT

# Ejemplo de Estilos Incrustados



```
<style type="text/css">
body { background-color: #e6E6fa;
      color: #191970;
      font-family: Arial, Verdana, sans-serif; }
h1 { background-color: #191970;
     color: #e6e6fa;
     line-height: 200%;
     font-family: Georgia, "Times New Roman", serif; }
h2 { background-color: #aeaed4;
     color: #191970;
     font-family: Georgia, "Times New Roman", serif; }
p {font-size: .90em; }
ul {font-weight: bold; }
</style>
```



# Selectores CSS

- Las reglas de estilo CSS pueden ser configuradas para un:
  - Selector de elemento HTML
  - Selector de clase (class)
  - Selector de id



# Usando CSS con “class”

- Selector class

- Se usa para aplicar una regla CSS a cierta "clase" de elementos en una página web
- No asocia el estilo a un elemento particular de XHTML

```
<style type="text/css">
.nueva { color: #ff0000;
         font-style: italic;
       }
</style>
```

- Se configura con .nombreclase

- El ejemplo crea una clase llamada “nueva” con texto rojo y en italico.

- Para usar la clase, se hace lo siguiente:

```
<p class="nueva">Este texto esta en rojo e italico</p>
```



# Usando CSS con “id”

- Selector id
  - Usado para aplicar una regla CSS a SOLO UN elemento en la página web.
- Se configura con #nombreid

```
<style type="text/css">
#nuevo {color: #ff0000;
        font-size:2em;
        font-style: italic;
        }
</style>
```

- El ejemplo crea un id denominado “nuevo” con texto grande, rojo e italico.
- Para usar el id, se hace lo siguiente:

```
<p id="nuevo">Este texto esta grande, en rojo e italico</p>
```



# Elemento XHTML <div>

- Elemento a nivel de bloque
- Propósito: configurar una división o area formateada de manera especial en una página web
  - Existe un avance de línea antes y después de la división.
  - Puede contener otros elementos a nivel de bloque o en línea.
- Util para definir un area que contendrá otras etiquetas a nivel de bloque (tales como párrafos) dentro de ella.



# Ejemplo de <div>

- Se desea configurar un area para servir de pie de página
- CSS Incrustado:

```
<style type="text/css">
.footer { font-size: small;
          text-align: center; }
</style>
```

- XHTML:

```
<div class="footer">Copyright &copy; 2012</div>
```





# Elemento XHTML `<span>`

- Elemento en línea (inline-level)
- Propósito:
  - Configurar un area formateada de manera especial que se despliega en línea con otros elementos tales como un párrafo.
- No hay ningun avance de línea antes o después del span.



# Ejemplo de <span>

- CSS Incrustado:

```
<style type="text/css">
.nombrecompania {
    font-weight: bold;
    font-family: Georgia, "Times New Roman", serif;
    font-size: 1.25em;
}
```

- XHTML:

```
<p>Sus necesidades son importantes para nosotros en
<span class="nombrecompania">RSR Software</span>.
Trabajaremos con usted para contruir su sitio web.</p>
```



# Hojas de Estilo Externas (1)

- Las reglas de estilo CSS están contenidas en un archivo de texto separado de los documentos XHTML.
- El archivo de texto de Hoja de Estilos externo:
  - Tiene extensión ".css"
  - Contiene solo reglas de estilo
  - No contiene ninguna etiqueta XHTML

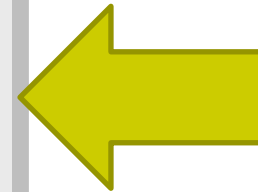


# Hojas de Estilo Externas (2)

- Múltiples páginas web pueden asociarse con el mismo archivo de hojas de estilo externo.

sitio.css

```
body {background-color:#e6e6fa;
      color:#000000;
      font-family:Arial, sans-serif;
      font-size:90%; }
h2 {  color: #003366; }
.nav { font-size: 16px;
       font-weight: bold; }
```



index.html

clientes.html

acerca.html

etc...



# Elemento <link>

- Etiqueta auto-contenida
- Se coloca en la sección header
- Propósito: asociar el archivo de hojas de estilo externo con la página web.
- Ejemplo:

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

# Centrando el Contenido de la Página con CSS

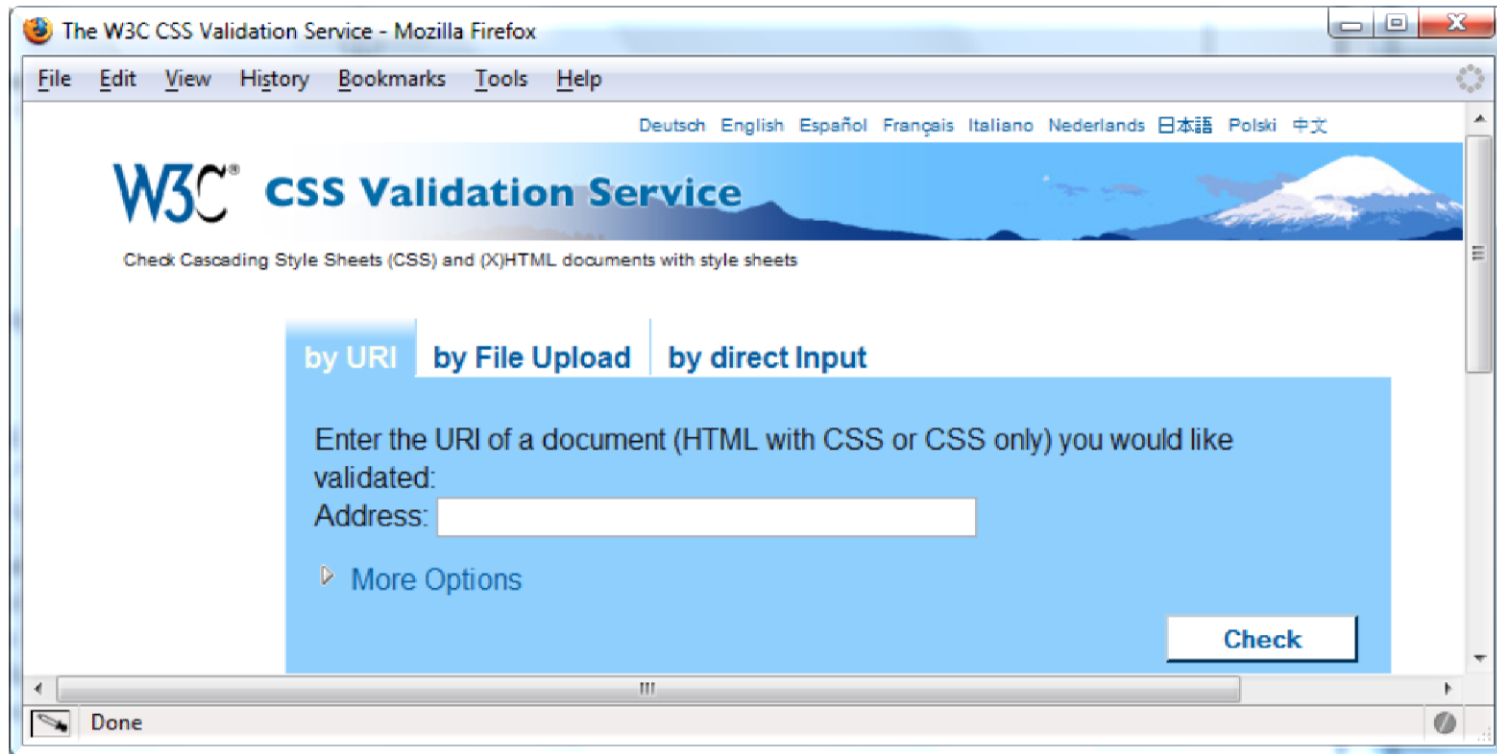


```
#container {    margin-left: auto;
                margin-right: auto;
                width:80%; }
```



# Validación de CSS por W3C

- <http://jigsaw.w3.org/css-validator/>





# Guías para CSS

- Revise el diseño de la página
  - Configure un tipo de letra y propiedades de color globales para el selector body
  - Identifique elementos típicos (tales como <h1>, <h3>, etc.) y declare reglas de estilo para estos si son necesarias.
  - Identifique areas de paginas tales como logo, navegación, pie de página, etc. y configure una clase o id apropiados para cada uno.
- Cree una página prototipo que contenga la mayoría de los elementos que planea usar y probar.
  - Revise su CSS conforme sea necesario.
  - Una vez que su diseño este establecido, mueva los estilos a un archivo .css externo



# Solución de Errores Comunes en CSS



- Verifique que está usando los símbolos : y ; en los lugares adecuados, es muy facil confundirse.
- Verifique que no esté usando el signo = en vez de : entre cada propiedad y su valor.
- Verifique que los símbolos { y } estan colocados apropiadamente
- Verifique la sintaxis de sus selectores, sus propiedades y valores.
- Si parte de su CSS funciona, y otra parte no:
  - Revise su CSS
  - Determine la primer regla que no es aplicada.  
A menudo el error se encuentra en la regla anterior a la que no es aplicada.
- Valide su CSS en <http://jigsaw.w3.org/css-validator>

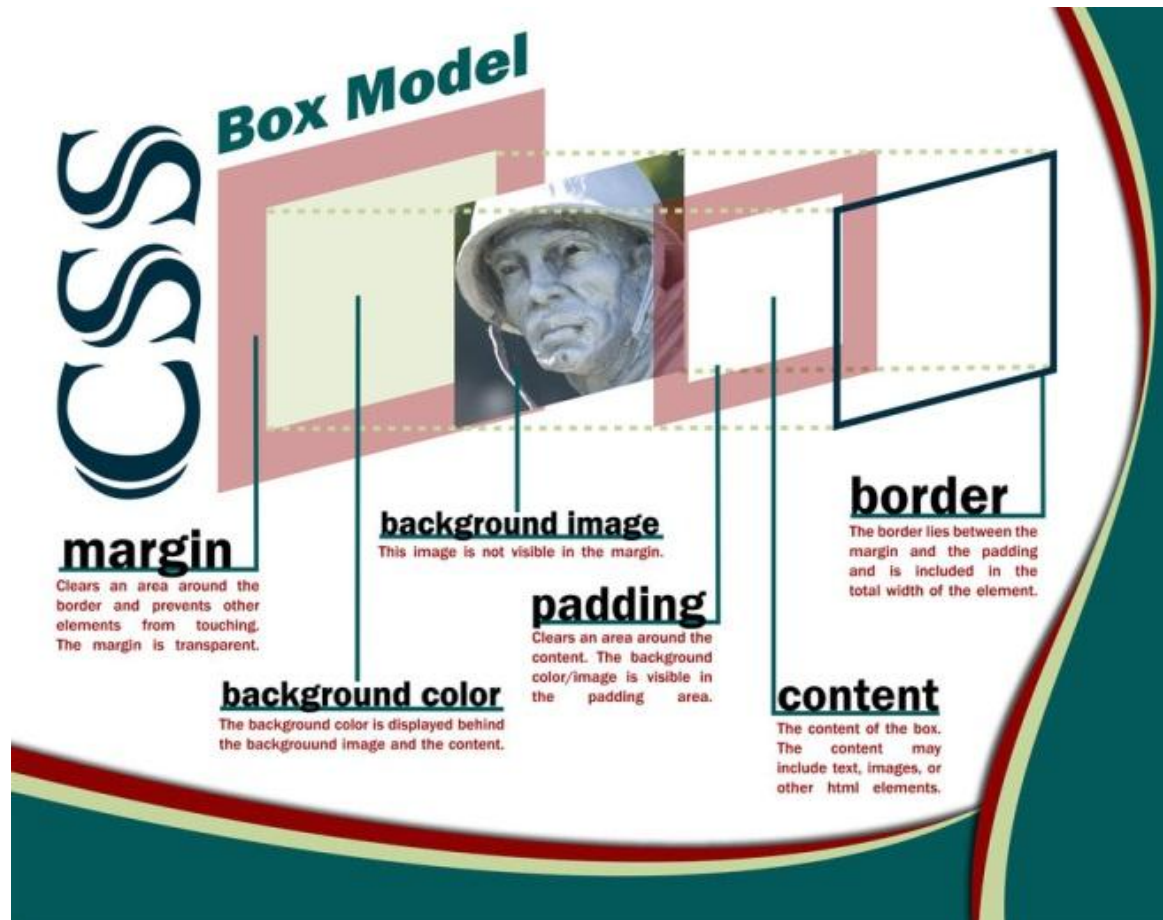


Introducción a las Tecnologías Web

# MAS CSS



# Modelo de Caja (Box Model)





# Propiedad border

- Configura un borde en la parte superior, derecha, inferior e izquierda de un elemento
- Consiste de
  - Anchura del borde (border-width)
  - Estilo del borde (border-style)
  - Color del borde (border-color)
- Ejemplo:

```
h2 { border: 2px solid #ff0000 }
```

# Diferencias en bordes para elementos de bloque o en línea



- Elemento de bloque
  - La anchura por default del contenido del elemento se extiende al margen del navegador (o la anchura especificada)
- Elemento en línea
  - El borde delimita de manera cercana al contenido del elemento

```
h2 { border: 2px solid #ff0000; }  
a   { border: 2px solid #ff0000; }
```

# El despliegue de bordes puede variar con los navegadores



## CSS border-style Values

### Firefox

default

none

inset

outset

double

groove

ridge

solid

dashed

dotted

### Internet Explorer

default

none

inset

outset

double

groove

ridge

solid

dashed

dotted

# Configuración de Lados Específicos de un Borde



- Se puede usar CSS para configurar una línea en uno o más de los lados de un elemento
  - border-bottom
  - border-left
  - border-right
  - border-top
- Ejemplo:

```
h2 { border-bottom: 2px solid #ff0000 }
```



# Propiedad padding

- Configura el espacio vacío entre el contenido del elemento XHTML y el borde
- Se establece a 0px por default
- Ejemplo:

```
h2 { border: 2px solid #ff0000;  
      padding: 5px; }
```



# Configuración de padding para lados específicos de un elemento



- Se puede usar CSS para configurar el padding sobre uno o más de los lados de un elemento
  - padding-bottom
  - padding-left
  - padding-right
  - padding-top

- Ejemplo:

```
h2 { border: 2px solid #ff0000; background-color: #cccccc;  
      padding-left: 5px;  
      padding-bottom: 10px;  
      padding-top: 10px;}
```

# Atajo de 2 valores en la propiedad padding



- Dos valores numéricos o porcentajes
  - El primer valor configura el padding superior e inferior
  - El segundo valor configura el padding izquierdo y derecho
- Ejemplo:

```
h2 { border: 2px solid #ff0000;  
      background-color: #cccccc;  
      padding: 20px 10px;  
    }
```

# Atajo de 4 valores en la propiedad padding



- Cuatro valores numéricos o porcentajes
  - Configuran el padding superior, derecho, inferior e izquierdo respectivamente

- Ejemplo:

```
h2 { border: 2px solid #ff0000;  
      width: 250px;  
      background-color: #cccccc;  
      padding: 30px 10px 5px 20px;  
}
```

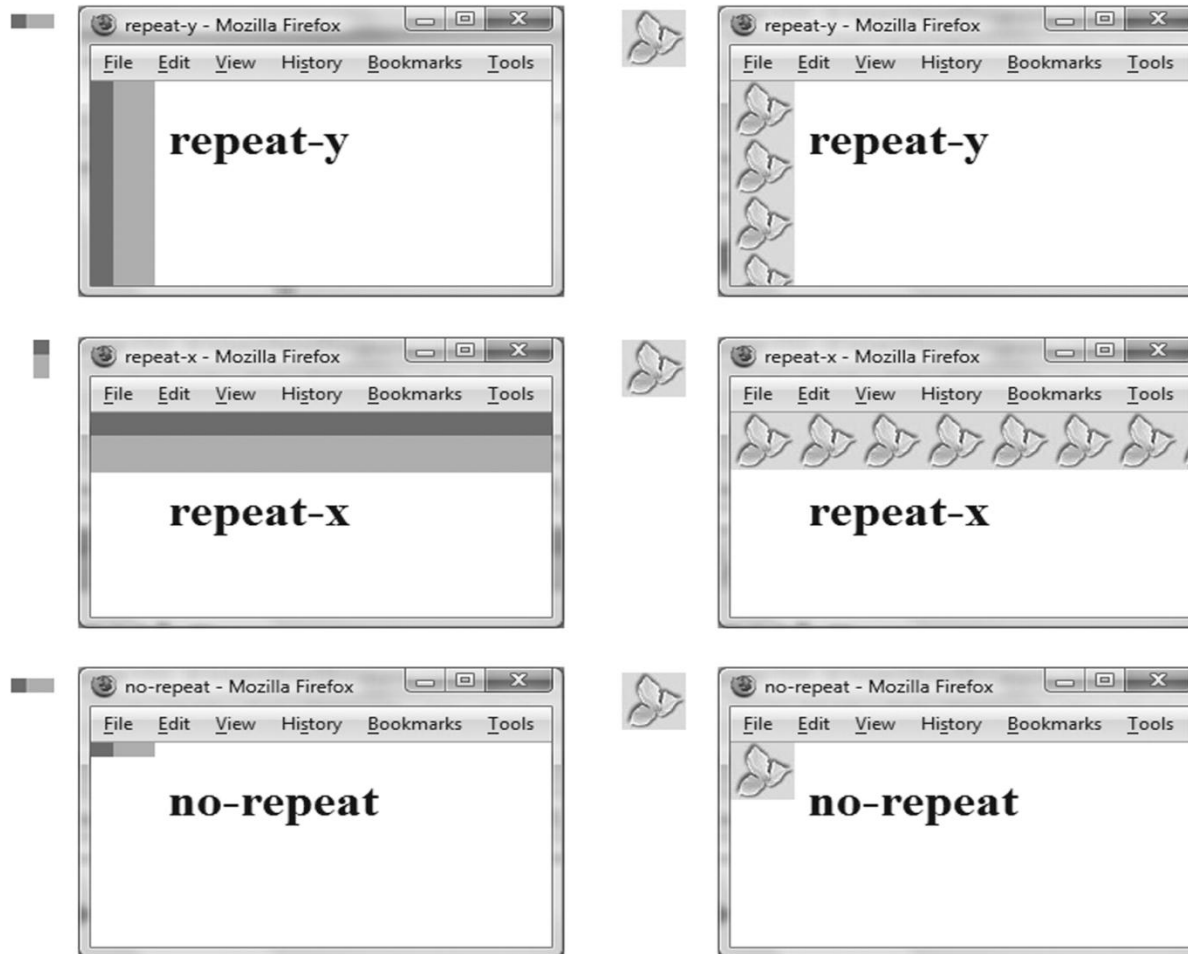


# Propiedad background-image

- Configura una imagen de fondo
- Por default, las imagenes de fondo se repiten (forma de mosaico)
- Ejemplo:

```
body { background-image: url(fondo1.gif); }
```

# Propiedad background-repeat





# Pseudo clases en CSS

- Pseudo clases y en el elemento ancla
  - **link** – estado por default para un enlace (<a>)
  - **visited** –un enlace que ya ha sid visitado
  - **hover** – activado cuando el mouse pasa encima del enlace
  - **active** – activado cuando se hace click en el enlace

```
a:link      {color:#000066;}  
a:visited   {color:#003366;}  
a:hover     {color:#0099CC;}  
a:active    {color#FF0000;}
```

# Estableciendo estilo a botones en CSS



## CSS border-style Values

### Firefox

default	none
<div>inset</div>	<div>outset</div>
<div>double</div>	<div>groove</div>
<div>ridge</div>	<div>solid</div>
<div>dashed</div>	<div>dotted</div>

### Internet Explorer

default	none
<div>inset</div>	<div>outset</div>
<div>double</div>	<div>groove</div>
<div>ridge</div>	<div>solid</div>
<div>dashed</div>	<div>dotted</div>



# Botones con “CSS”

```
<style type="text/css">
.button { border: 2px inset #cccccc;
width: 100px;
padding: 3px 15px;
color: #ffffff;
background-color: #006600;
font-family: Arial, Helvetica, sans-serif;
font-size: 16px;
font-weight: bold;
text-align: center;
text-decoration: none;
}
a.button:link { color : #FFFFFF; }
a.button:visited { color : #cccccc; }
a.button:hover { color : #66cc33;
border: 2px outset #cccccc;
}
</style>
```

## CSS Buttons!

[Home](#)[Products](#)[Services](#)[Contact](#)[About](#)

```
<div align="center">
<h2>CSS Buttons!</h2>
<a href="index.html" class="button">Home</a>
<a href="products.html"
class="button">Products</a>
<a href="seviles.html"
class="button">Services</a>
<a href="contact.html"
class="button">Contact</a>
<a href="about.html" class="button">About</a>
</div>
```



# Uso de CSS para dar estilo a una tabla



Atributo XHTML	Propiedad CSS
align	Para alinear una tabla: <code>table {width: 75%; margin:auto; }</code> Para alinear una celda: <code>text-align</code>
bgcolor	<code>background-color</code>
cellpadding	<code>padding</code>
cellspacing	Para configurar un borde común y eliminar espacio entre celdas: <code>table {border-collapse: collapse; }</code>
height	<code>height</code>
valign	<code>vertical-align</code>
width	<code>width</code>