

# Programación web en el entorno cliente

# **UF1841. Elaboración de documentos web mediante lenguajes de marca**



Elaborar documentos utilizando lenguajes de marcas y estándares de desarrollo software:

- Determinar las diferentes partes de un documento creado con lenguaje de marcas utilizado para su implementación.
- Reconocer las diferentes técnicas de desarrollo de software existentes en el mercado para mejorar la integración en el sistema y elaboración de documentos según el diseño especificado.
- Utilizar marcas adecuadas para generar la documentación interna en el desarrollo según las especificaciones del diseño.
- Usar marcas para proporcionar diferentes estilos a los documentos desarrollados según el diseño especificado.
- Construir documentos utilizando lenguajes de marcas para permitir al usuario el uso de dispositivos móviles y medios específicos de accesibilidad.

1. Diseño web
2. Lenguajes de marcado para presentación de páginas web
3. Organización de elementos
4. Hojas de Estilos en Cascada
5. Biografía y webgrafía

# **Elaboración de documentos web mediante lenguajes de marca**

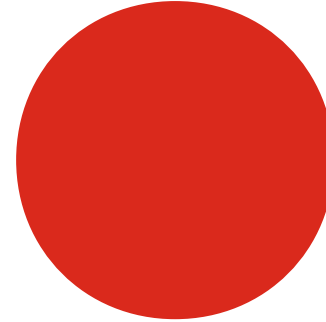
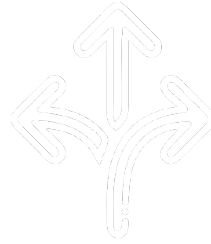
## **1. Diseño web**

- Principios de diseño web
- El proceso de diseño web

# Principios estéticos del diseño web



Contraste



Alineación



Repetición



Proximidad

# Principios estéticos del diseño web: **Contraste**

Destacar elementos de la página para proporcionar aspecto visual atractivo.



# Principios estéticos del diseño web: **Alineación**

- Ningún elemento de la página debe estar puesto arbitrariamente.
- Organiza el contenido bajo un formato específico que facilite la lectura de la página.
- Mantener las proporciones entre los márgenes y el espacio de trabajo utilizado.





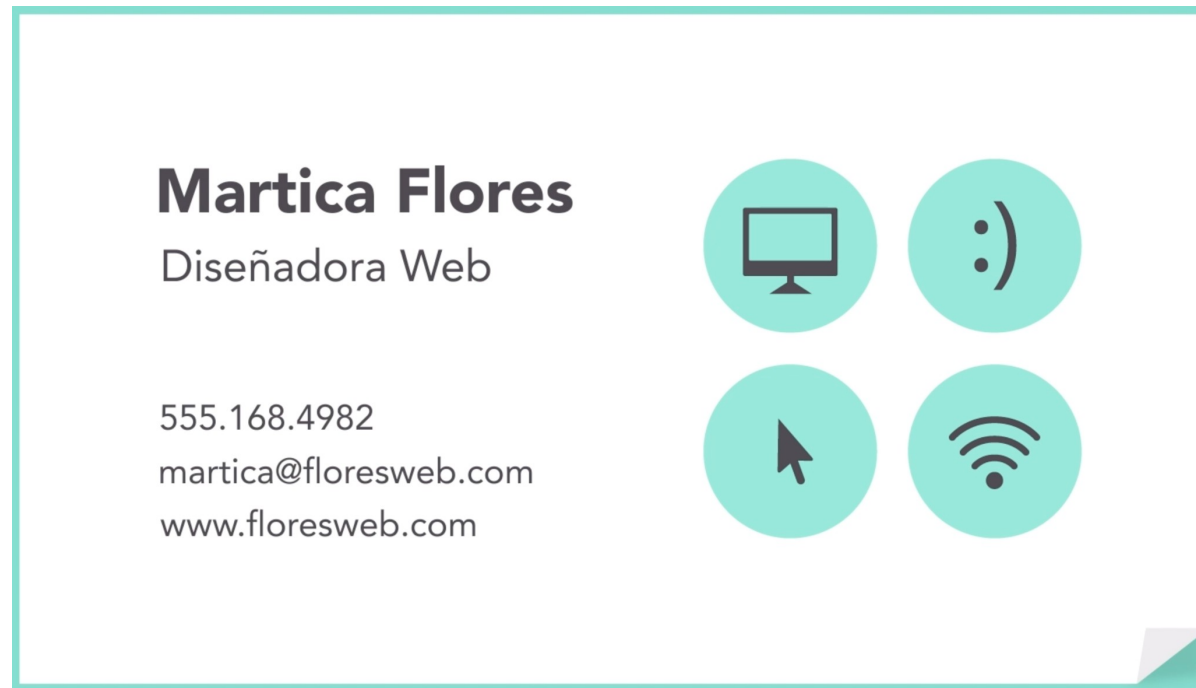
# Principios estéticos del diseño web: **Repetición**

- Repetir elementos de diseño para proporcionar uniformidad: colores, formas, texturas, fuentes, tamaños, conceptos gráficos, etc.
- Proporciona aspecto único, profesional y consistente permitiendo a los usuarios familiarizarse con la web.



# Principios estéticos del diseño web: **Proximidad**

- Organización de los elementos de una página.
- Agrupar los elementos relacionados entre sí.
- Permite organizar la información, reduce el desorden y da al usuario un sentido de estructura.



# ¿Qué es **user-centered design**?

## Investigación



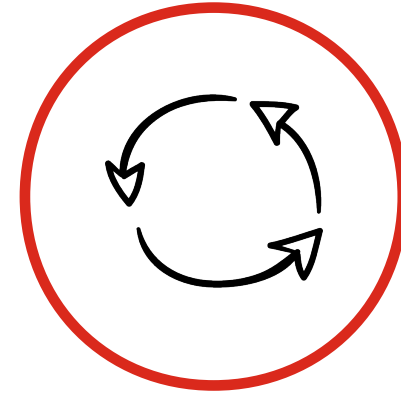
Comienza con un análisis de los usuarios, sus objetivos y contexto

## Empatía



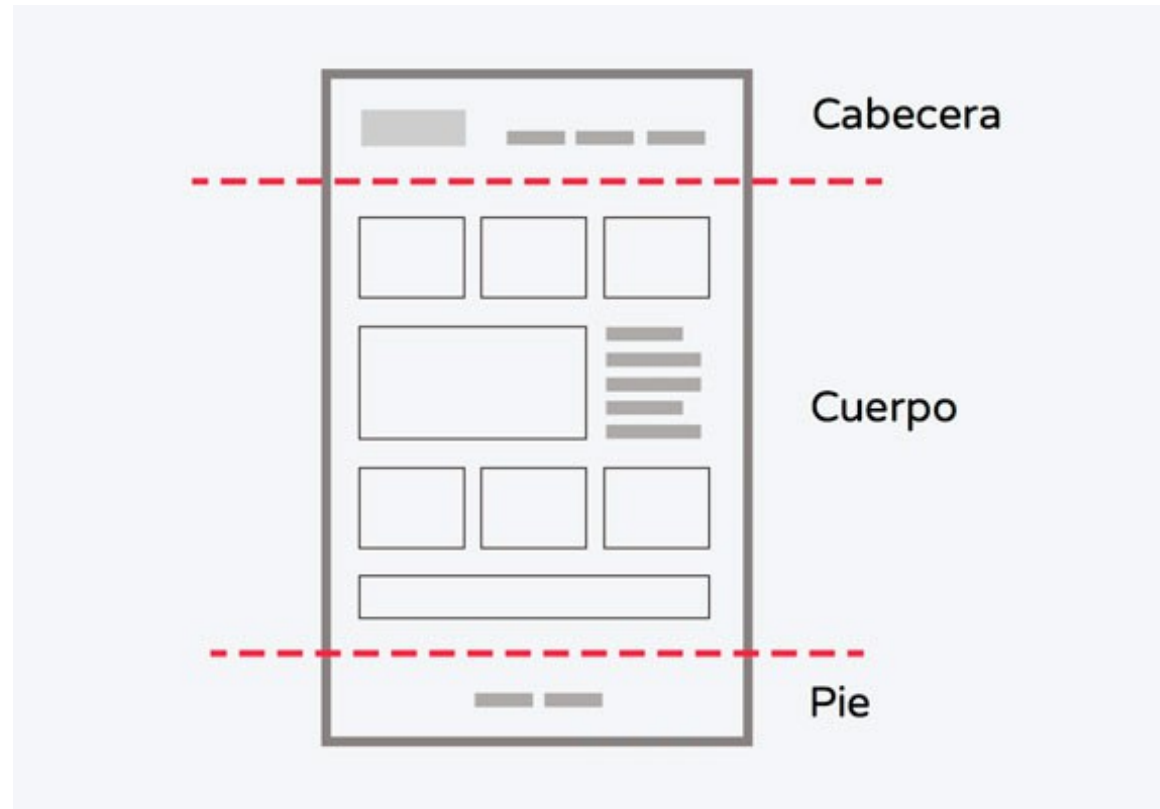
Ponerse en el lugar de los usuarios y entender sus necesidades

## Iteración

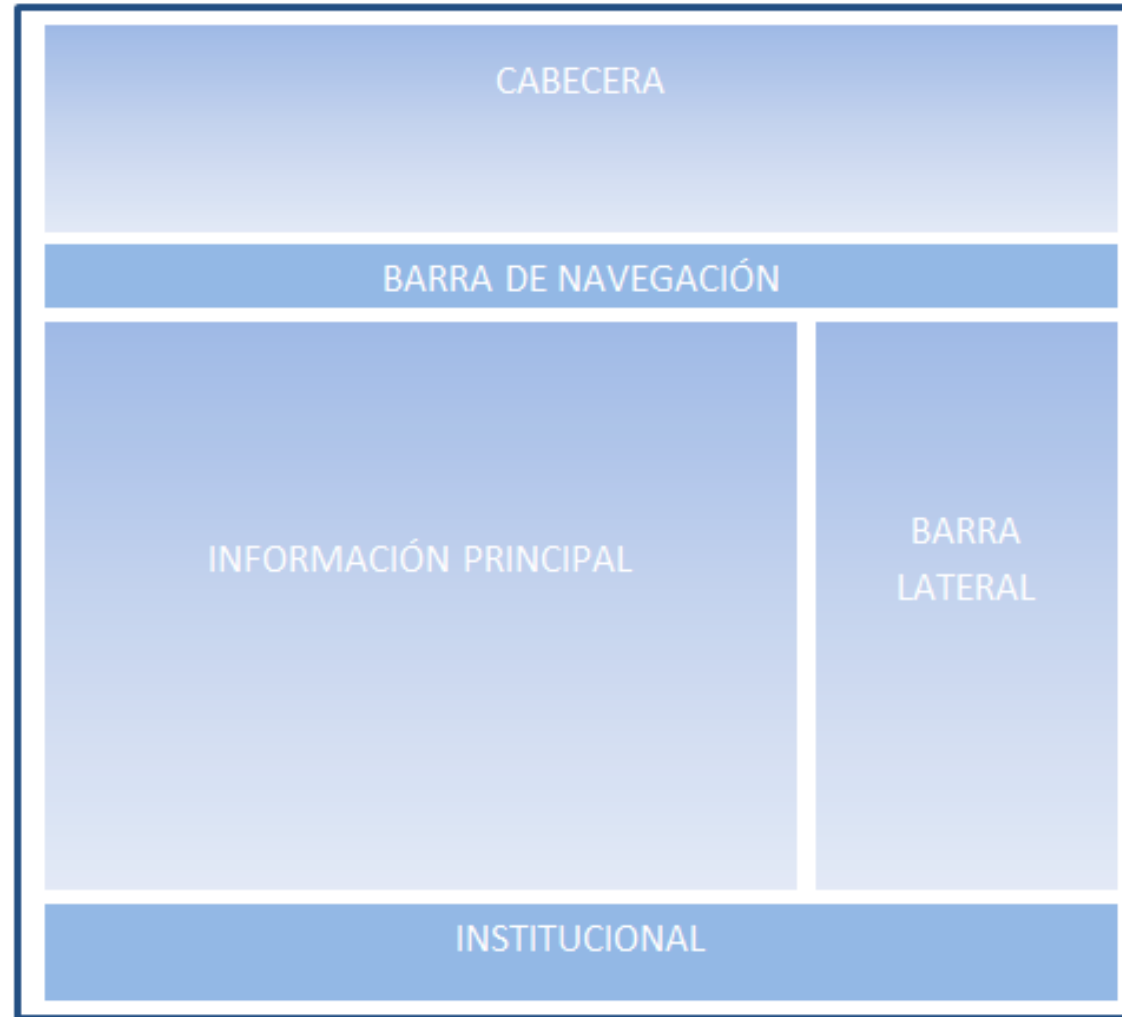


Proceso de iteración que permita la evaluación y la mejora de forma constante

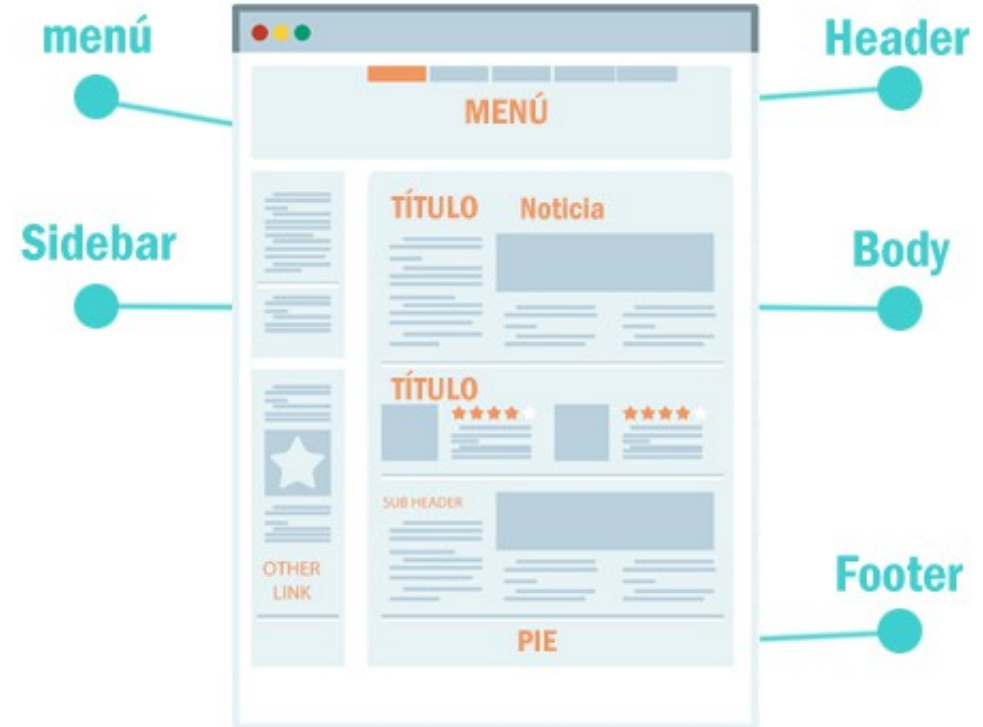
# Proceso de diseño web: **Estructura** de una página web



# Proceso de diseño web: **Organización y elementos de la página**



# Proceso de diseño web: **Organización y elementos de la página (II)**



# Proceso de diseño web: **Herramientas online**

<b>HTML5 Test</b>	Verifica soporte para elementos HTML5 en un navegador concreto <a href="http://html5test.com">html5test.com</a>
<b>W3C Validator</b>	Markup Validation Service <a href="https://validator.w3.org">https://validator.w3.org</a>
<b>Modernizr</b>	Detección de funciones HTML5, CSS3 y JS <a href="https://modernizr.com">https://modernizr.com</a>
<b>Pollyfill</b>	Código JavaScript en la web que proporciona una funcionalidad moderna en navegadores antiguos que no lo admiten de forma nativa.
<b>Codepen</b> (SaaS)	Editor HTML, CSS y Javascript online <a href="https://codepen.io/">https://codepen.io/</a>
<b>Media Query</b>	Módulo CSS3 para adaptar la representación del contenido al dispositivo
<b>CSS3 Test</b>	Verifica soporte para elementos CSS3 en un navegador concreto <a href="https://css3test.com/">https://css3test.com/</a>

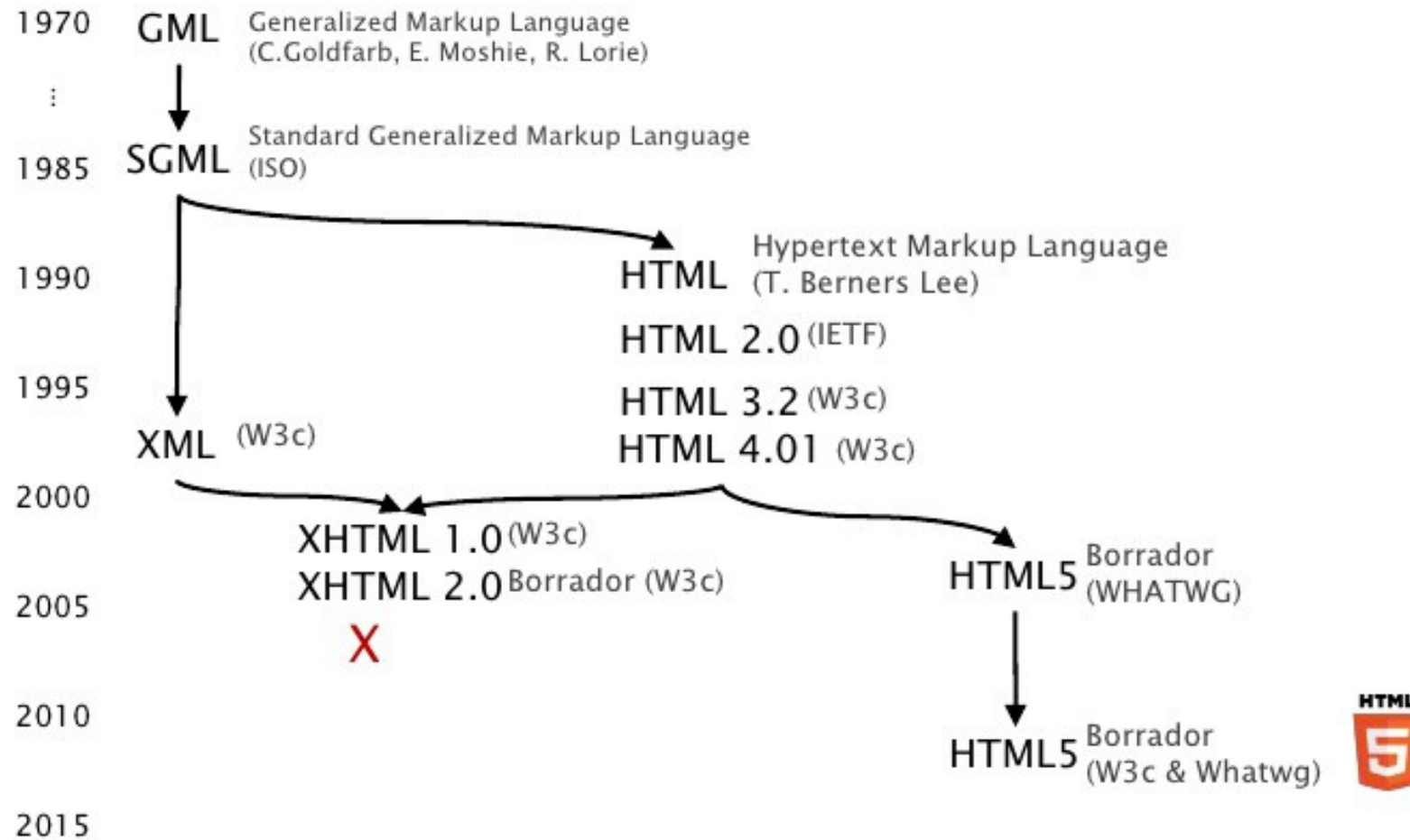
# Elaboración de documentos web mediante lenguajes de marca

## 3. Lenguajes de marcado para presentación de páginas web

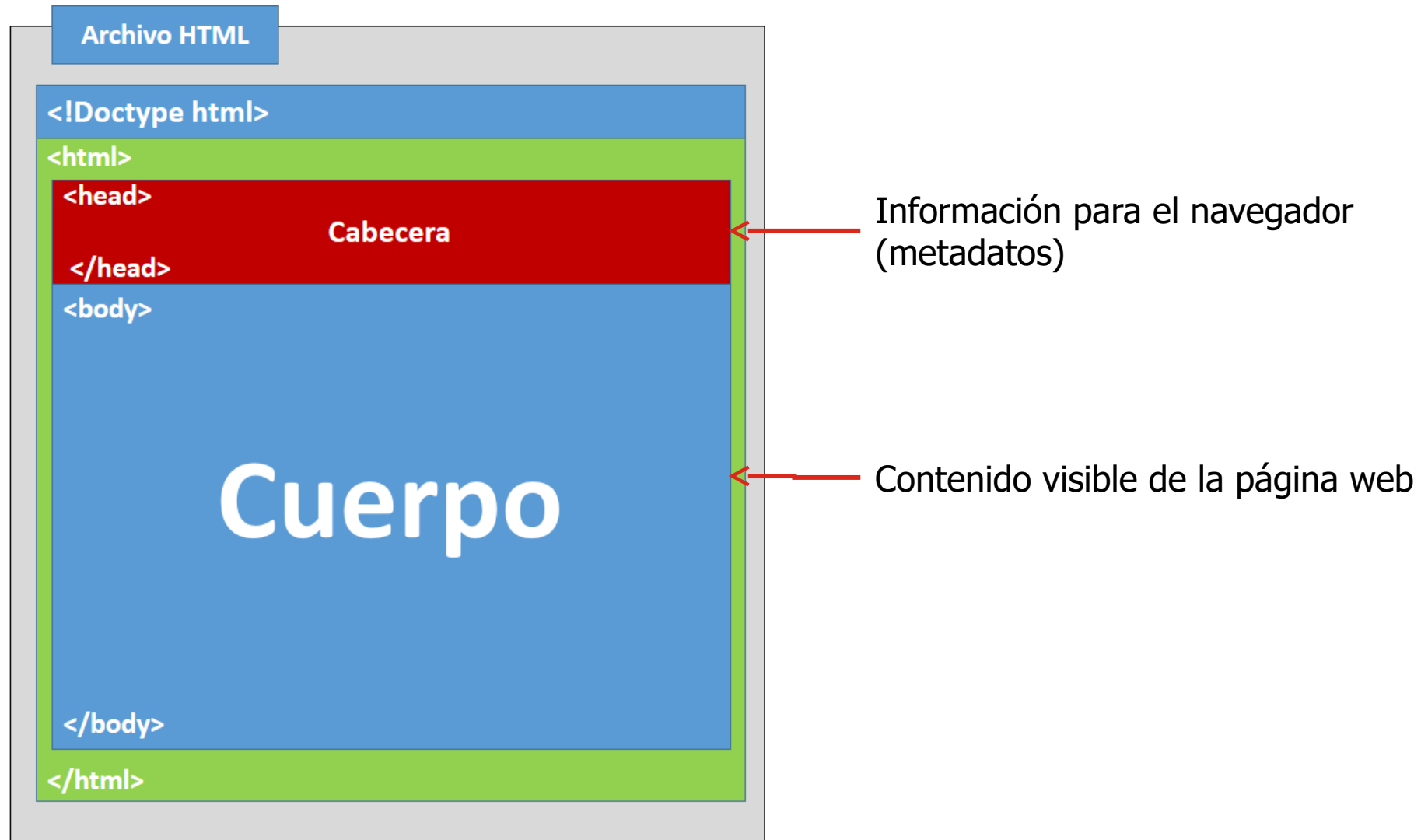
- Evolución de HTML
- Estructura de un documento HTML
- Color
- Texto
- Hipervínculos
- Imágenes
- Listas
- Tablas
- Formularios
- Frames
- Elementos específicos para tecnologías móviles
- Elementos en desuso (deprecated)
- HTML 5



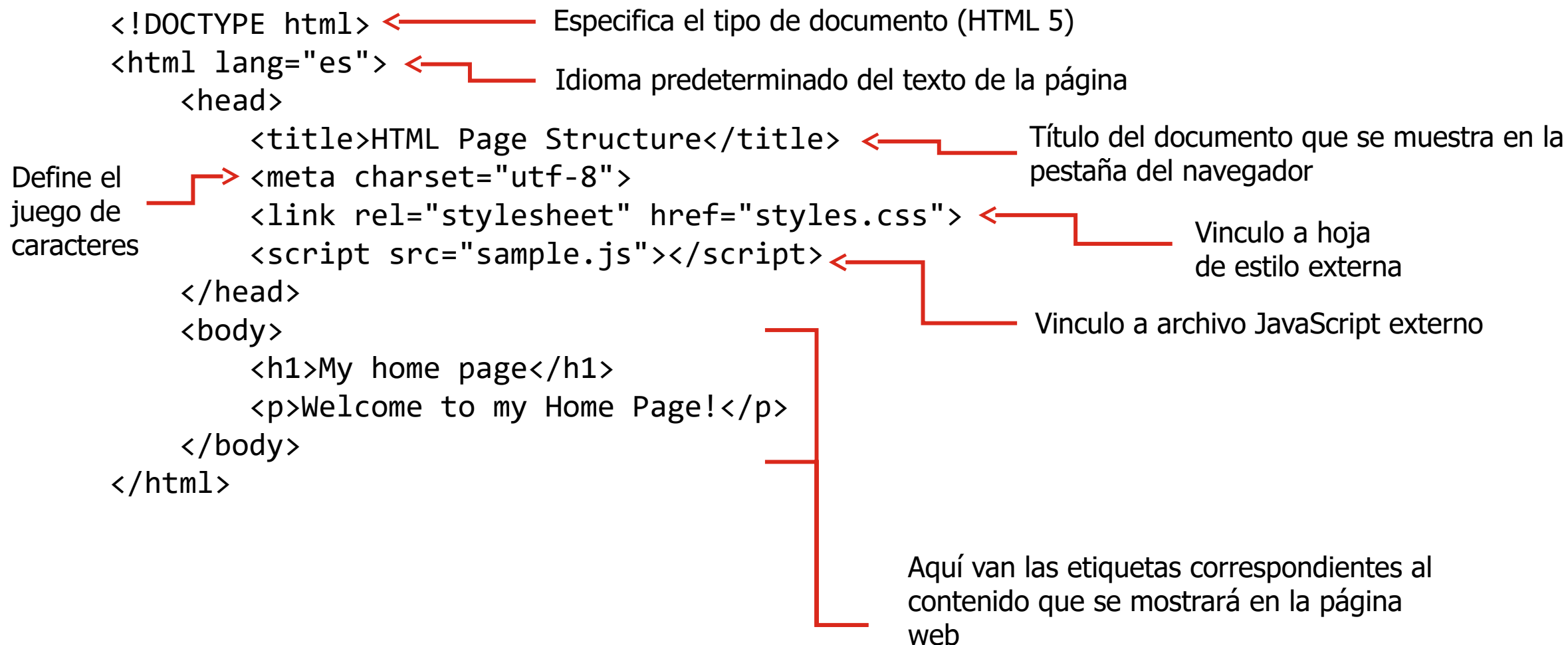
# Evolución de HTML



# Estructura de un documento HTML

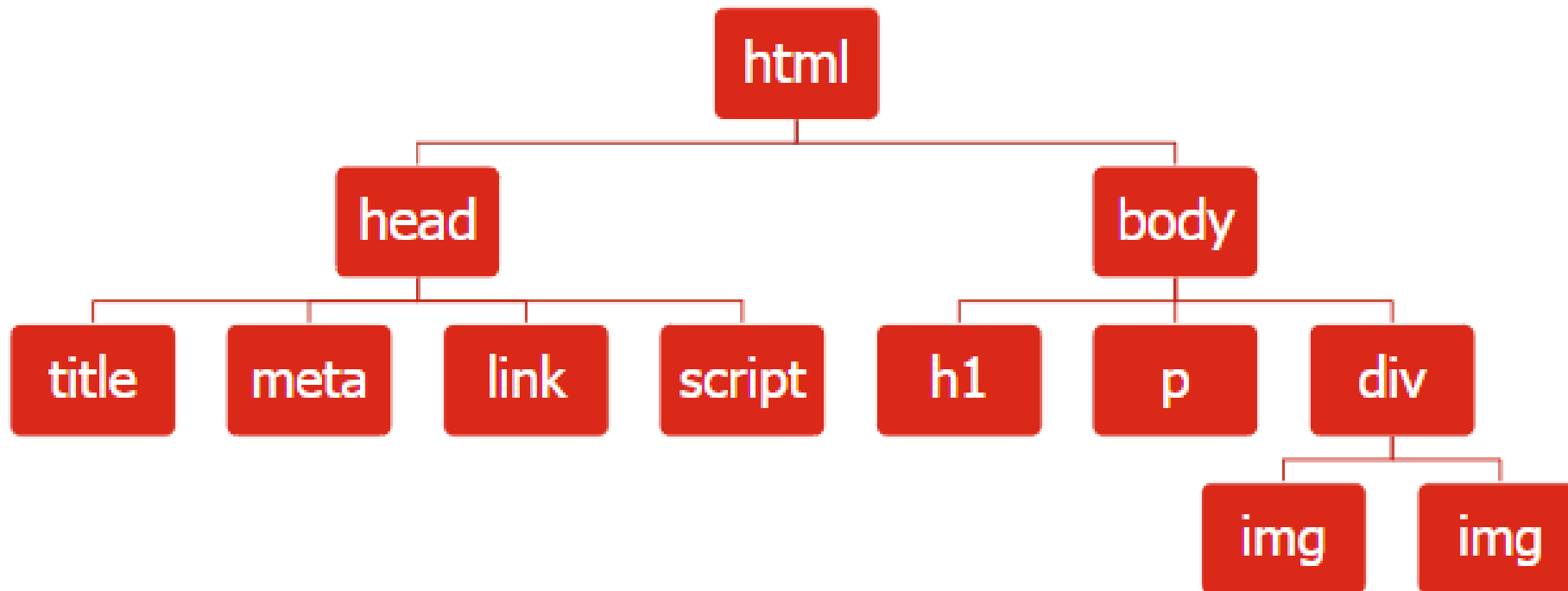


# Estructura de un documento HTML (II)



# Representación Jerárquica de un documento HTML

- Otra forma de representar la estructura de una página web es mediante un árbol en donde cada nodo representa un elemento o etiqueta HTML.



# Sintaxis de un elemento HTML

- HTML define su contenido a través de **etiquetas** o **tags**, que son los **elementos** del documento. La sintaxis de una etiqueta depende de si tiene contenido o no.
- En el caso de las que tienen contenido, la sintaxis es:

```
<nombre-elemento [atributo="valor"]>Contenido</nombre-elemento>
```

Mismo nombre al abrir y cerrar etiqueta

- En el caso de las que no tienen contenido, la sintaxis es:

```
1) <nombre-elemento [atributo="valor"]></nombre-elemento>  
2) <nombre-elemento [atributo="valor"]>
```

- Los atributos son utilizados para especificar propiedades del contenido.

# Sintaxis HTML: Comentarios

- La forma de insertar comentarios en HTML es:

```
<!-- bloque de comentarios -->
```

- Permiten documentar y depurar el código fuente HTML.
- No se muestran en la página web.
- El bloque comentado puede tener una o más líneas y cualquier contenido.

```
<!--  
  Inicio sección de datos  
-->  
<h1> ... </h1>
```

## Elementos de **<head>**

# Elementos HTML básicos: **head**

La etiqueta `<head>` (cabecera) define información del documento que no se muestra directamente al usuario. La sección head puede contener la siguiente información:

- El título que se muestra en la barra o pestaña del navegador
- Metadatos
- Vínculos a hojas de estilo
- Vínculos a scripts

```
<!DOCTYPE html>

<html>
  <head>
  </head>
  <body>

  </body>
</html>
```



## Elementos sección <head>: **title**

- La etiqueta <**title**> define el **título del documento** y se muestra en la barra de título del navegador o en la pestaña de la página.

```
<head>  
  <title>Programas formación</title>  
</head>
```

- Esta etiqueta es **obligatoria** en documentos HTML. Debe ser solo texto y proporciona un título para la página cuando se agrega a favoritos.
- Muestra el título en la página de resultados del motor de búsqueda. El contenido del título de una página es muy importante para la optimización de motores de búsqueda (SEO). Los algoritmos de los motores de búsqueda utilizan el título de la página para determinar el orden de aparición en las páginas de resultados.

## Elementos sección <head>: **meta**

- La etiqueta <meta> (metadatos) permite especificar información general del documento HTML.
- En general se utiliza para especificar el conjunto de caracteres:

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

- Para especificar el autor y la descripción del documento:

```
<meta name="author" content="Netmind">  
<meta name="description" content="Detalle de programas">
```

- Configuración de la ventana gráfica para que el documento se ajuste a todos los dispositivos (diseño responsive):

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

## Elementos sección <head>: **meta**

- Información sobre actualización de la página, que permite que recargue otra página (o la misma) después de cierto tiempo:

```
<meta http-equiv="refresh" content="30">
```

← Actualiza el documento  
cada 30 segundos.

- Definir palabras clave para los motores de búsqueda:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

# Elementos sección <head>: **Enlace**

- La etiqueta **<link>** define la relación entre dos documentos vinculados. Se utiliza para enlazar páginas de estilo externas al documento HTML. Es un elemento vacío, solo contiene atributos.

```
<!-- HTML4 -->
```

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

```
<!-- HTML5 -->
```

```
<link rel="stylesheet" href="styles.css">
```

- Atributos:**
  - rel:** Define la relación entre el documento enlazado con el actual. Valores: `stylesheet` para hojas de estilo CSS, y otros como `alternate`, `bookmark`, `chapter`, `help` y `glossary`.
  - type:** Especifica el tipo MIME (media type) del documento vinculado. Se utiliza `text/css` para hojas de estilo.
  - href:** La URL del documento vinculado.

# Elementos sección <head>: **Script**

- La etiqueta **<script>** permite vincular el documento con un script del lado del cliente (archivo Javascript).

```
<!-- HTML4 -->
```

```
<script type="text/javascript" src="javascript.js"></script>
```

```
<!-- HTML5 -->
```

```
<script src="javascript.js"></script>
```

- **Atributos:**
- **type:** Especifica el tipo de contenido del bloque de código, siendo `text/javascript` su valor por defecto.
- **src:** La URL del script a ser cargado.

## Elementos de **<body>**

# Elementos HTML básicos: **Cajas**

- Se utiliza el tag `<div></div>` para confeccionar una caja básica que, por defecto, ocupará el 100% del ancho de la caja contenedora y siempre tiene un salto de línea (elemento de bloque).

```
<div>Caja 1</div>  
<div>Caja 2</div>  
<div>Caja 3</div>
```

Caja 1  
Caja 2  
Caja 3

# Elementos HTML básicos: **Encabezado**

- Se utilizan los tags `<h1>` a `<h6>` para insertar títulos o subtítulos (headings) dependiendo de la importancia .

```
<h1>Encabezado 1</h1>  
<h2>Encabezado 2</h2>  
<h3>Encabezado 3</h3>  
<h4>Encabezado 4</h4>  
<h5>Encabezado 5</h5>  
<h6>Encabezado 6</h6>
```

**Encabezado 1**

**Encabezado 2**

**Encabezado 3**

**Encabezado 4**

**Encabezado 5**

**Encabezado 6**

- Son de tipo block, pues después de cada uno se agrega un salto de línea.



# Elementos HTML básicos: **Párrafo**

- Para especificar un párrafo, se utiliza la etiqueta **<p>**:

```
<h2>HTML, CSS y JavaScript</h2>  
  
<p>El contenido incluye  
HTML, CSS y JavaScript</p>  
<p>Primero se describe el  
lenguaje HTML</p>
```

**HTML, CSS y JavaScript**

El contenido incluye HTML, CSS y JavaScript

Primero se describe el lenguaje HTML

El espacio entre párrafos  
es doble.

No se aplica salto de línea  
del código fuente.

- Es de tipo block, después de cada párrafo se añade un salto de línea.

# Elementos HTML básicos: **Párrafo**

- Atributo align para especificar la alineación de un párrafo:

```
<h3>Detalles</h3>
```

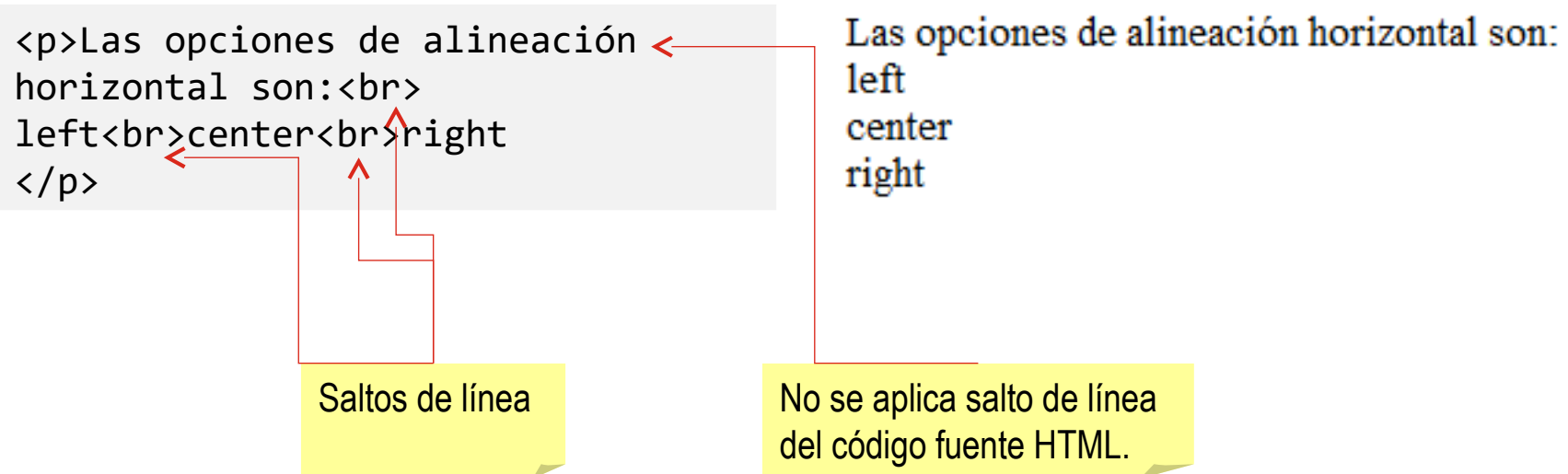
```
<p align="justify">El atributo  
align del párrafo define su  
alineación. Las opciones  
principales son left, center,  
right y justify (horizontal),  
y existen otras como top, middle  
y bottom para la alineación  
vertical.</p>
```

## **Detalles**

El atributo align del párrafo define su alineación. Las opciones principales son left, center, right y justify (horizontal), y existen otras como top, middle y bottom para la alineación vertical.

# Elementos HTML básicos: Salto de línea

El salto de línea del código fuente no tiene efecto sobre el HTML. Para insertar un salto de línea (break) se utiliza `<br>`:



# Elementos HTML básicos: **Formato**

- A un texto se le puede cambiar el aspecto, utilizando:

- **negrita:**

Conocer HTML es  
<b>muy importante</b>.

Conocer HTML es **muy importante**.

- *cursiva:*

El atributo <i>align</i>  
especifica alineación.

El atributo *align* especifica alineación.

- *énfasis:*

Este es el <em>contenido  
básico</em>.

Este es el *contenido básico*.

# Elementos HTML básicos: **Imagen**

- Añadir una imagen mediante la etiqueta **<img>**:

```

```

fuelle: URL o ruta (relativa  
o absoluta) de la imagen.

# Elementos HTML básicos: **Imagen**

- Se puede especificar el **alto** y **ancho** de una imagen, la que se ajusta:

```

```

```

```

```

```

Si se omite uno de los dos atributos, se mantiene la relación de aspecto.

# Elementos HTML básicos: **Imagen**

- Imagen, atributos
- Una imagen puede tener una representación alternativa en caso que no logre cargarse, con el atributo alt:

```

```

- Si el archivo de la imagen no se encuentra, se ve lo siguiente en función del navegador:



# Elementos HTML básicos: **Imagen**

- En general, los elementos HTML tienen un atributo llamado **title**, que despliega un tooltip (texto flotante) cuando se pasa el puntero del mouse sobre éste.

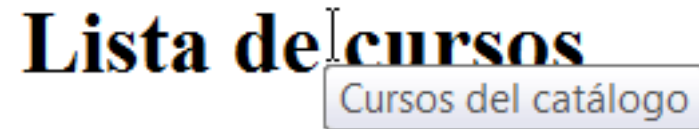
- Ejemplos:

```

```



```
<h1 title="Cursos del catálogo">  
Lista de cursos</h1>
```





# Formatos de imagen

Los navegadores soportan varios tipos de imagen:

- **GIF** (Graphics Interchange Format): Para imágenes de pocos colores, que no deben perder nitidez, como los logos, o textos en formato imagen. Soporta transparencia.
- **JPG** (de JPEG, Joint Photographic Experts Group): es un formato comprimido, que a cambio tiene una pérdida de calidad ajustable en función de cuánto se comprime. Es adecuado para fotografías grandes, donde la pérdida es poco perceptible. No soporta transparencia. Formato preferido para la web ya que ofrece un buen equilibrio entre calidad y tamaño (peso) de la imagen.
- **PNG** (Portable Network Graphics): Tiene tamaño intermedio entre GIF y PNG. Soporta transparencia. Permite utilizar más colores que GIF dependiendo del número de bits (PNG-8, 24 o 32). Se utiliza en los mismos casos que GIF, para imágenes que requieren nitidez como logos o textos en imágenes.

# Principales estilos: **Lista**

- Las listas HTML permiten agrupar elementos relacionados
- Dos tipos de listas: sin orden **<ul>** y ordenadas **<ol>**
- Cada elemento de la lista se define con la etiqueta **<li>**

```
<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
  <li>Orange Juice</li>  
</ul>
```

- Coffee
- Tea
- Milk
- Orange Juice

```
<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
  <li>Orange Juice</li>  
</ol>
```

1. Coffee
2. Tea
3. Milk
4. Orange Juice

# Elementos HTML básicos: **Lista**

- Las listas con orden (opción por defecto) muestran un índice en cada ítem.
- Ejemplo de lista ordenada con números:

```
<h3>Cursos línea Java:</h3>  
<ol>  
  <li>Java Básico</li>  
  <li>Java Intermedio</li>  
  <li>Spring Core</li>  
  <li>Hibernate-JPA</li>  
  <li>Web Services</li>  
</ol>
```

ordered list

list item

## **Cursos línea Java:**

1. Java Básico
2. Java Intermedio
3. Spring Core
4. Hibernate-JPA
5. Web Services

# Elementos HTML básicos: Lista ordenada

- Ordenada con letras minúsculas:

```
<ol type="a">  
  <li>Java Básico</li>  
  <li>Java Intermedio</li>  
  <li>Spring Core</li>  
  <li>Hibernate-JPA</li>  
  <li>Web Services</li>  
</ol>
```

a. Java Básico  
b. Java Intermedio  
c. Spring Core  
d. Hibernate-JPA  
e. Web Services

- Ordenada con números romanos en mayúsculas:

```
<ol type="I">  
  <li>Java Básico</li>  
  <li>Java Intermedio</li>  
  <li>Spring Core</li>  
  <li>Hibernate-JPA</li>  
  <li>Web Services</li>  
</ol>
```

I. Java Básico  
II. Java Intermedio  
III. Spring Core  
IV. Hibernate-JPA  
V. Web Services

Nota: el atributo type está en desuso.

# Elementos HTML básicos: **Hipervínculo**

- Permite vincular o la navegación entre documentos HTML:

```
<a href="catalogo.html">Ir a catálogo</a>
```

[Ir a catálogo](#)



- El principal atributo de <a> es **href** el cual indica el destino del enlace o hipervínculo.
- Un link, una vez visitado, por default cambia su aspecto gráfico: [Ir a catálogo](#)
- Para crear un link que abra el programa de email por defecto del usuario asignar el valor **mailto:** como valor del atributo href.

# Elementos HTML básicos: **Hipervínculo**

- Imagen como hipervínculo:

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

Elemento <img> anidado dentro de <a>



# Elementos HTML básicos: **Hipervínculo**

- En el atributo **href** se puede incluir:
  - Una URL de una página de destino:

```
<a href="http://www.netmind.es">Netmind Home</a>
```

- Una ruta relativa o absoluta:

```
<a href="../pages/catalogo.html">Ir a catálogo</a>
```

```
<a href="C:/html/pages/catalogo.html">Ir a catálogo</a>
```

# Elementos HTML básicos: **Hipervínculo**

- En el atributo **target** permite definir como destino otra ventana o pestaña del navegador:

```
<a href="http://www.netmind.es" target="_blank">Netmind Home</a>
```

- Posibles valores del atributo `target`:
  - `_self`: Valor por defecto. Abre el documento en la misma ventana o pestaña
  - `_blank`: Abre el documento en una nueva ventana o pestaña
  - `_parent`: Abre el documento en el marco principal
  - `_top`: Abre el documento en el cuerpo de la ventana



# Elementos HTML básicos: **Anchor**

- La etiqueta `<a id="valor">` es un anchor (ancla), que permite definir posiciones de un documento para saltar a otra zona del mismo documento o a otro documento externo:

Sintaxis de un anchor: #  
seguido del id del anchor de  
destino

Anchor  
con id

```
<h1>Cursos Java:</h1>
<a href="#net">Ir a cursos .NET</a>
<ul>
  <li>Java Básico</li>
  <li>Java Intermedio</li>
  ...
</ul>
<a id="net"><h3>Cursos .NET:</h3></a>
<ul>
  <li>C# Básico</li>
  <li>C# Avanzado</li>
  ...
</ul>
<h1>Otros cursos:</h1>
...
```

# Elementos HTML básicos: **div**, **span**

- Una página puede ser dividida en secciones, las cuales pueden tener un tratamiento particular. Para ello, se utilizan las etiquetas **<div>** y **<span>**.

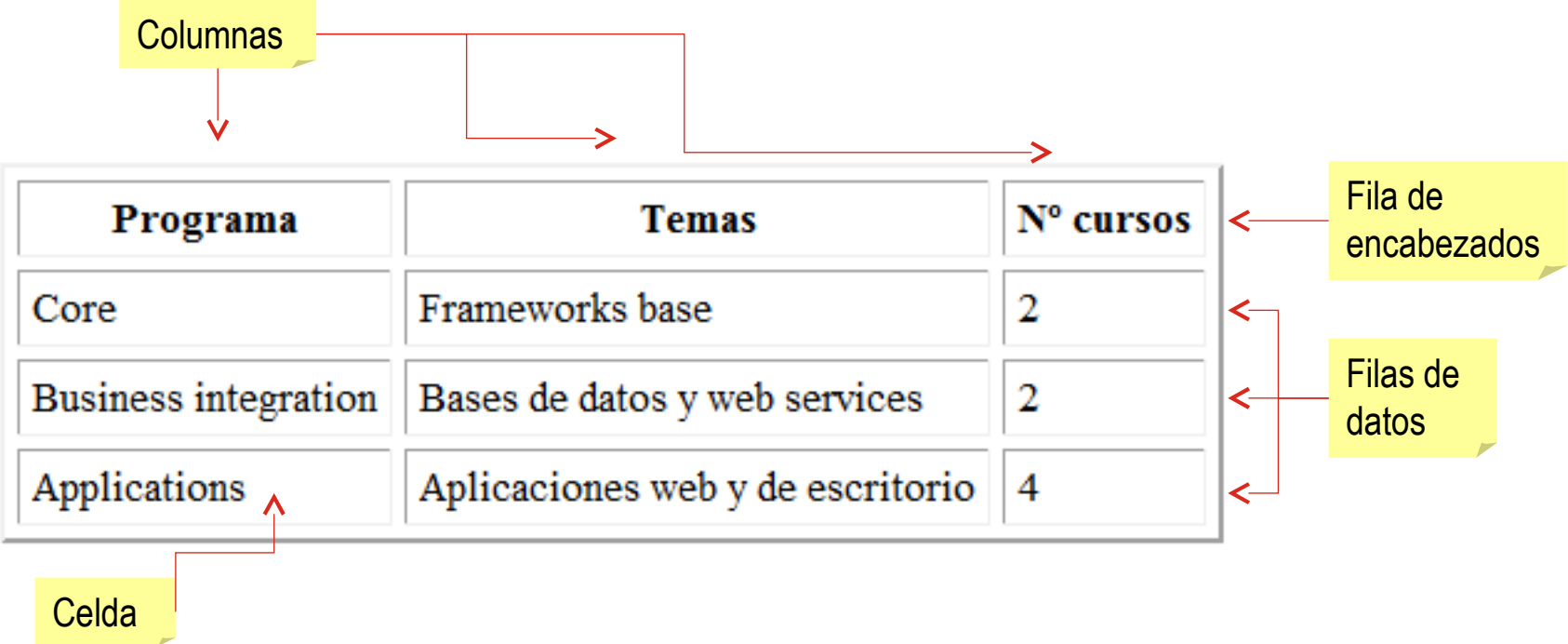
```
<div>  
  <h3>Título</h3>  
  <p>Contenido</p>  
</div>
```

```
<p>Curso <span>(1)</span>: HTML</p>
```

- La diferencia entre **<div>** y **<span>** es que el primero es de tipo block, es decir, agrega un salto de línea, mientras que el segundo es inline, y no lo añade.
- Utilidad del manejo de secciones:
  - Definir características comunes, principalmente a través de reglas de estilo, como se ve más adelante.

# Elementos HTML básicos: **Tabla**

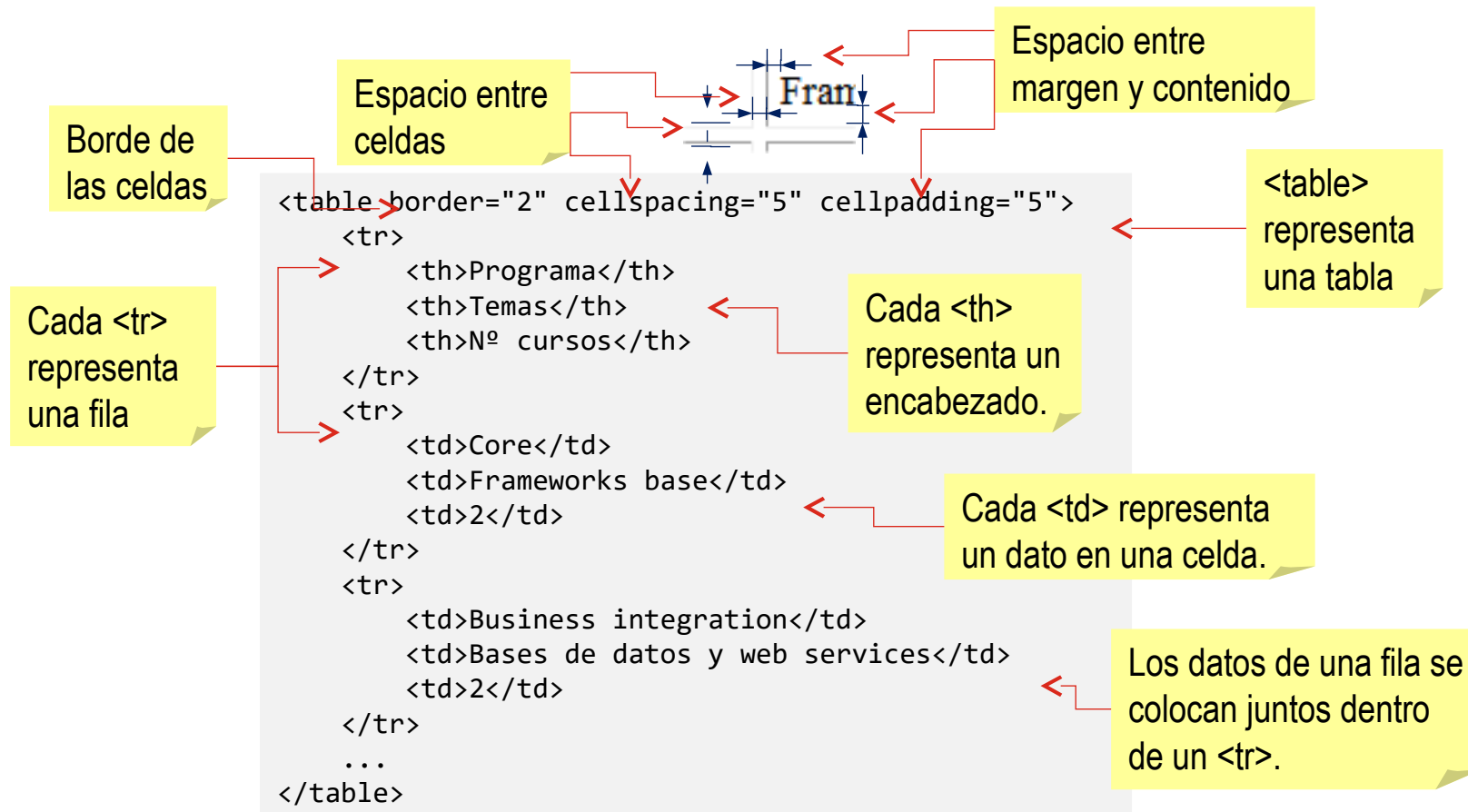
Estructura y elementos de una tabla:



# Elementos HTML básicos: **Tabla**

La etiqueta **<table>** define una tabla junto **<tr>**, **<th>** y **<td>**

Código HTML de la tabla ejemplo anterior:



# Elementos HTML básicos: **Tabla**

## Combinación de celdas con rowspan y colspan:

```
<table border="2" cellspacing="5" cellpadding="5">
  <caption>Cursos de formación</caption>
  <tr>
    <th rowspan="2">Programa</th>
    <th colspan="2">Curso</th>
  </tr>
  <tr>
    <th>Nombre</th>
    <th>Duración</th>
  </tr>
  <tr>
    <td rowspan="2">Business integration</td>
    <td>Persistencia</td>
    <td>24 horas</td>
  </tr>
  <tr>
    <td>Web services</td>
    <td>24 horas</td>
  </tr>
</table>
```

Programa	Curso	
	Nombre	Duración
Business integration	Persistencia	24 horas
	Web services	24 horas

# Elementos HTML básicos: **Formulario**

Un formulario `<form>` agrupa un conjunto de **campos** (controles de formulario) que permiten al usuario insertar datos e interactuar con sus elementos.

The diagram illustrates the mapping between HTML code and its rendered form. On the left, the HTML code is shown in a light gray box. On the right, the rendered form is shown with various controls. Red arrows connect the code to the corresponding UI elements. Yellow callout boxes provide additional context for specific elements.

```
<form action="..." >
  <input type="hidden" name="issueId" id="summ"
    name="summary" value="this is a summary">
</p>
<p>Secret: <input type="password" id="passwd"
  name="secret" size="15" value="mypassword">
</p>
<p>Type: <select name="typeId" id="typeId">
  <option value="1" selected>
    Technology</option>
  <option value="2">Production</option>
</select>
</p>
<p><input type="checkbox" name="hasAttachments">
Has attachments</p>
<p><input type="file" name="attachment">
</p>
...
```

Summary: this is a summary

Secret: .....

Type: Technology ▼

☐ Has attachments

Browse...

Campo oculto

Campo para upload

# Elementos HTML básicos: Formulario

```
...  
<p>State:  
<input type="radio" name="state"  
  value="P">Pending  
<input type="radio" name="state"  
  value="S" checked>Solved  
<input type="radio" name="state"  
  value="C">Closed  
</p>  
<p>Description:<br>  
<textarea rows="3" cols="40" id="desc">  
This is a description  
</textarea></p>  
<p><input type="reset" value="Clean">  
</p>  
<p><input type="submit" value="Send">  
</p>  
</form>
```

El grupo de los radio button lo define el name común a todos.

State: ☐ Pending ☒ Solved ☐ Closed

Description:

This is a description

Clean

Send

Envía el formulario al destino definido en "action" utilizando el "method".

# Elementos HTML básicos: **Formulario**

HTML provee ciertas **validaciones** o comportamientos de los formularios de forma nativa:


- **Habilitación:** se puede deshabilitar un campo de cualquier tipo con el atributo **disabled**. Tiene normalmente texto gris.

```
<input type="text" value="45"  
      name="identifier" disabled>
```



- **Sólo lectura:** se puede dejar un campo de texto como sólo lectura con el atributo **readonly**. Tiene el mismo estilo, pero no permite escribir.

```
<input type="text" value="resumen"  
      name="summary" readonly>
```



- **Largo máximo:** El atributo **maxlength** especifica el número máximo de caracteres permitidos en un campo de entrada, el cual es recomendable añadir por seguridad.



# Elementos HTML básicos: **Formulario**

- Tamaño: el ancho visible, en caracteres, de un campo de entrada se especifica con el atributo **size**.

```
<input type="text" name="description" size="45">
```

- Requerido: el atributo **required** obliga a introducir valores en un campo antes de enviar los datos del formulario.

```
<input type="text" name="id" value="Identifier" required>
```

# Elementos HTML básicos: **Formulario**

- Los controles de un formulario se pueden manipular como un objeto JavaScript, permitiendo aplicar HTML dinámico.

```
<form class="frm" >
  <div>
    <label for="idSumm">Summary:</label>
    <input type="text" size="30" id="idSumm" name="summary" value="this is a summary">
  </div>
  <div>
    <label for="secret">Secret:</label>
    <input type="password" id="secret" size="15">
  </div>
  <div>
    <label for="type">Type:</label>
    <select id="idType" name="type">
      <option value="1" selected="selected">
        Technology</option>
      <option value="2">Production</option>
    </select>
  </div>
</form>
```

# Elementos HTML básicos: iFrame

- La etiqueta `<iframe>` especifica un marco en línea y permite incrustar otro documento HTML dentro del actual.

```
<iframe src="https://www.netmind.es" title="Web de Netmind">
```

- Los atributos `height` y `width` definen el alto y ancho respectivamente del marco en línea.

```
<iframe src="https://www.netmind.es" title="Web de Netmind" height="200" width="300">
```

# Buenas prácticas

# Buenas prácticas

Espacios, tabulaciones y saltos de línea

- HTML sólo considera un espacio, aunque hayan más, e ignora las tabulaciones y saltos de línea:

```
<p>Los múltiples    espacios, tabulaciones y saltos  
de línea se interpretan como un solo espacio</p>
```

Los múltiples espacios, tabulaciones y saltos de línea se interpretan como un solo espacio

- Para insertar más de un espacio, se debe utilizar "&nbsp;".  
Ejemplo:

```
<p>Múltiples&nbsp;&nbsp;&nbsp;espacios entre palabras</p>
```

Tres espacios

# Buenas prácticas

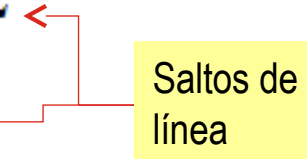
- Block vs inline

En HTML existen etiquetas del tipo **block** y del tipo **inline**.

- Las de tipo block agregan un salto de línea después de su utilización:

```
<h2>Encabezado 2</h2>  
<h3>Encabezado 3</h3>
```

**Encabezado 2**  
**Encabezado 3**

A yellow callout box with the text "Saltos de línea" has two red arrows pointing to the right. The first arrow points to the end of the line "Encabezado 2", and the second arrow points to the end of the line "Encabezado 3".

Saltos de línea

- Las de tipo inline no lo agregan:

```
  

```

Two gray square buttons are shown side-by-side. The first button contains a plus sign (+) and the second contains a right-pointing triangle (▶). A red arrow points from a yellow callout box below to the space between the two buttons.

No hay salto de línea

# Buenas prácticas

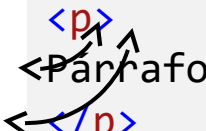
- Anidamiento de etiquetas
- En HTML, las etiquetas se pueden anidar, permitiendo contener elementos en otros:

```
<html>
  <body>
    <div class="frame">
      <div class="brd">
        <div class="sec">Sección</div>
      </div>
    </div>
    <p>Párrafo</p>
  </body>
</html>
```

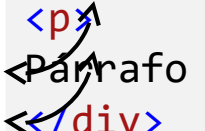
# Buenas prácticas

- Cierre de etiquetas
- Al estar anidadas, las etiquetas deben cerrarse de manera adecuada, respetando el orden de la jerarquía:

```
<div class="sec">  
  <p>  
  <Párrafo  
  </p>  
</div>
```



```
<div class="sec">  
  <p>  
  <Párrafo  
  </div>  
</p>
```





# Caracteres especiales

- En HTML, los caracteres especiales (acentos, eñes y otros) se pueden escribir directamente:

```
<p>la antigüedad en la compañía</p>
```

- No obstante, si el juego de caracteres de la página no corresponde con el del archivo, se pueden producir conflictos que hacen que no se vean correctamente. Por ejemplo, si el archivo HTML utiliza **utf-8** y la página declara **iso-8859-1**, se vería algo así:

la antigÃ¼edad en la compaÃ±a

# Caracteres especiales

- Para evitar los efectos por diferencias de juego de caracteres entre el archivo físico y el declarado en el contenido HTML, existe una nomenclatura para los caracteres especiales:

```
<p>la antig&uuml;edad en compa&ntilde;&iacute;a</p>
```

- Nomenclatura para las vocales acentuadas:

`&[letra][tipo modificador];`

a – e

A – E

En algunos idiomas,  
otras letras no vocales.

Distintos tipos usados en algunos idiomas:

- acute (agudo): á, Á
- grave (grave): à, À
- circ (cincunflejo): â, Â
- tilde (tilde): ã, Ã
- uml: ä, Ä
- ring (anillo): å, Å

# Caracteres especiales

- Además de los acentos, hay otros caracteres especiales, algunos de los cuales se describen en la siguiente tabla:

Caracter	Nombre entidad	Descripción
&	&amp;	símbolo & (ampersand)
(espacio)	&nbsp;	espacio en blanco (non breaking space)
"	&quot;	comillas
'	&apos;	apóstrofe (las llamadas comillas simples)
<	&lt;	menor que (less than)
>	&gt;	mayor que (greater than)
¿	&iquest;	abre signo de interrogación
!	&iexcl;	abre signo de exclamación/admiración
©	&copy;	copyright

# Caracteres especiales

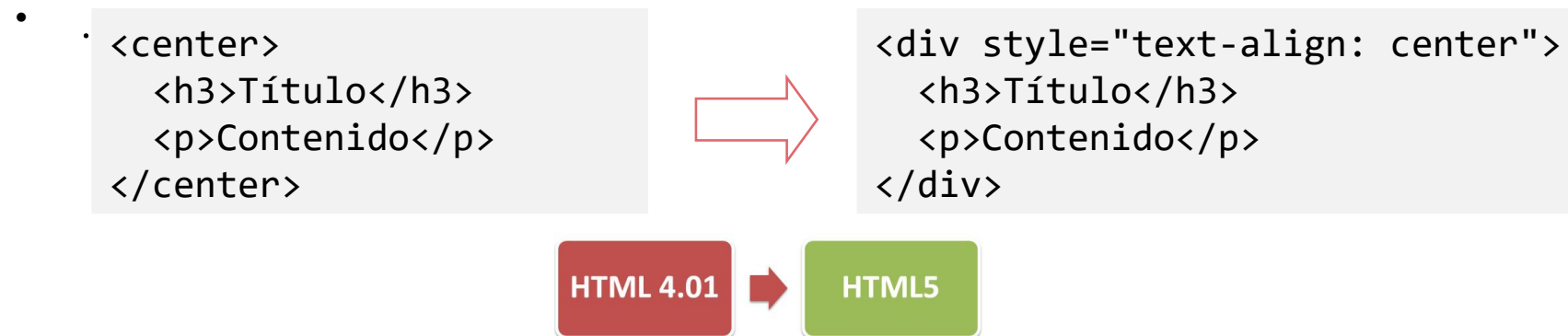
- También se utiliza la nomenclatura "&#[código decimal];" para los caracteres especiales. Ejemplos:

Carácter	Nombre entidad	Descripción
(espacio)	&#160;	Espacio en blanco
&	&#38;	símbolo &
"	&#34;	Comillas
'	&#39;	apóstrofe (las llamadas comillas simples)
<	&#60;	menor que
>	&#62;	mayor que
¿	&#191;	abre signo de interrogación
¡	&#161;	abre signo de exclamación/admiración

# Elementos en desuso (deprecated)

- Algunos elementos y etiquetas HTML en desuso:

- Fuente `<font>`
- Texto parpadeante `<blink>`
- Marquesina `<marquee>`
- Marco o Frame `<frame>`
- Alineaciones `<center>`



# HTML 5

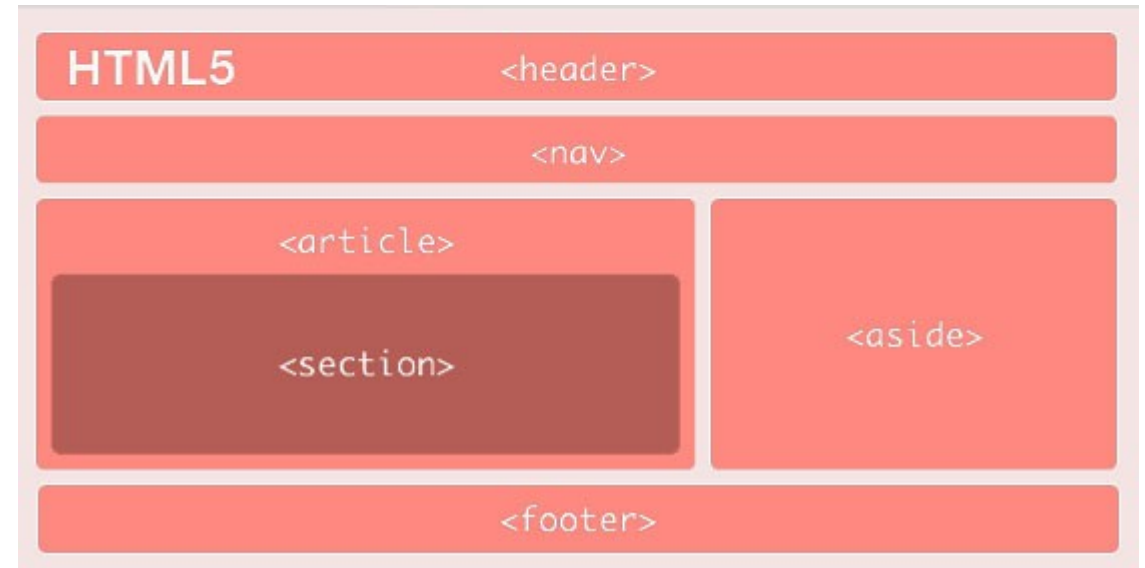
# HTML 5

Novedades más importantes:

- Estructura semántica
- Etiquetas aportan estructura. Atributos aportan funcionalidad.
- Formularios mejorados: significado semántico
- Nuevos valores que puede adoptar la etiqueta `<input>`
- Accesibilidad
- Capacidad de crear atributos personalizados.
- Nuevos elementos multimedia (audio y video)



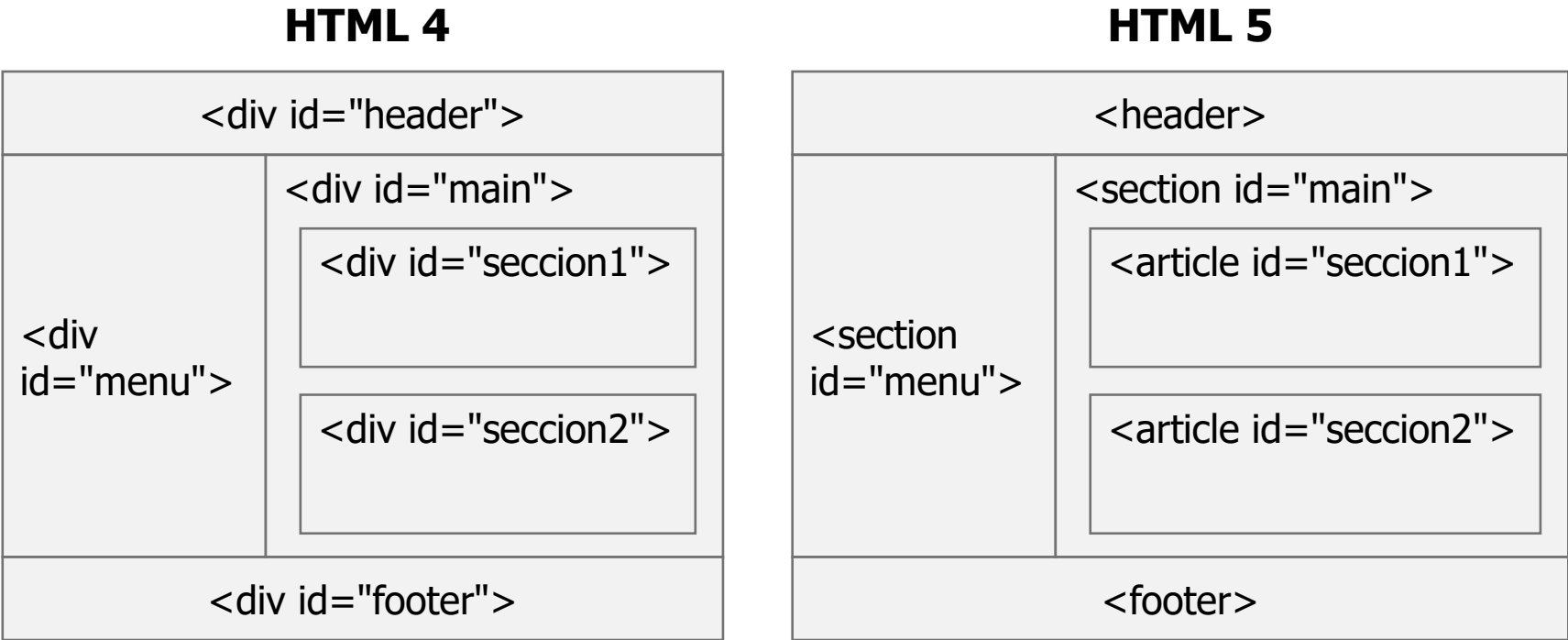
# Estructura de un documento HTML 5





# Estructura de un documento HTML 5

- Ejemplo estructura HTML4 vs HTML5



# HTML5

## Nuevos valores para <input>

```
<p>tel: <input type="tel" /></p>
<p>search: <input type="search" /></p>
<p>url: <input type="url" /></p>
<p>email: <input type="email" /></p>
<p>datetime: <input type="datetime" /></p>
<p>date: <input type="date" /></p>
<p>month: <input type="month" /></p>
<p>week: <input type="week" /></p>
<p>time: <input type="time" /></p>
<p>datetime-local: <input type="datetime-local" /></p>
<p>number: <input type="number" /></p>
<p>range: <input type="range" /></p>
<p>color: <input type="color" /></p>
```

tel:

search:

url:

email:

datetime: 2011-08-30 23:55 UTC

date:

month: 2011-08

week: 2011-W32

time: 23:27

datetime-local: 2011-08-30 23:53

number:

range:

color:

2011-08-02 11:57 UTC

Agosto 2011

Lun	Mar	Mie	Jue	Vie	Sab	Dom
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Hoy

# HTML5

- Marcador de posición o placeholder
  - Muestra un texto al usuario en un tono gris claro, indicando tipo de dato se espera que se introduzca en el cuadro del input.
  - Se implementa como un atributo al que se asigna la cadena a mostrar y es común a todos los valores del atributo type donde tiene sentido: tel, search, url, email.
  - Si el usuario pulsa dentro de la caja de texto, el texto gris claro desaparece.

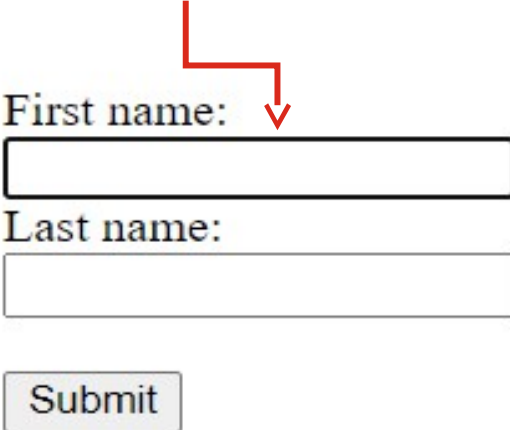
```
<input type="tel" placeholder="format:(xxx)xx-xx-xxx">
```

tel:

# HTML5

- Autoenfoco o autofocus
  - Indica qué control del formulario debe tener el foco cuando se carga la página.
  - Ejemplo:

```
<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" autofocus><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname">  
  <input type="submit">  
</form>
```



# HTML5

- Elemento `<datalist>`
  - El elemento `<datalist>` no tiene definido ningún tipo de presentación, pero cuando se utiliza conjuntamente con otro control, puede presentar un cuadro combinado o combobox.
  - Se complementa con elementos `<option>` para indicar cada uno de los posibles valores de la selección.
  - Ejemplo: Combinar un elemento `<datalist>` con una entrada `<input>` de tipo email.

```
<input list="contactos" name="contactmail">  
<datalist id="contactos">  
    <option value="davidcorrea@gmail.com" label="David">  
    <option value="luisagomez@gmail.com" label="Luisa">  
</datalist>
```

# HTML5

- Entrada de valores numéricos mediante el tipo **range**

```
<label>Years hired: </label>
<input type="range" max="10" min="0" value="5" onchange="funcionJS()">
<span id="showYears">7</span>

<script type="text/javascript">
    function funcionJS() {
        alert();
    }
</script>
```



# HTML5

- Acotación de valores con atributos min y max
  - Podemos acotar aún más las posibilidades de entrada, facilitando la labor de validación y minimizando errores. Tal es el caso de **max** y **min** (valor máximo y mínimo), para el caso de los campos numéricos.

```
<label for="fyear">Years hired:</label>  
<input type="number" max="10" min="0" value="5">
```

Years hired:

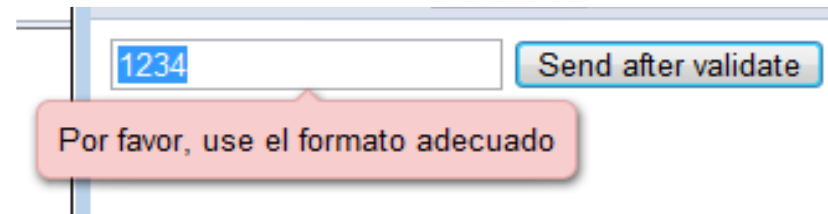
# HTML5

## Validación de campos de formulario

- A través del atributo **pattern** podemos utilizar una expresión regular como **patrón** para restringir la entrada de datos, obteniendo así un primer nivel de validación garantizando, en el ejemplo, que se introducen solamente números y un máximo de tres dígitos:

```
<input type="tel" pattern="\d{3}">
```

- Se recomienda acompañarlo del atributo title para avisar al usuario del patrón válido.
- Si el valor introducido no cumple con la expresión regular, el navegador puede mostrar una etiqueta flotante:







# ACTIVIDAD

---

**Formulario**

## OBJETIVO

# Diseñar un formulario para recopilar datos mediante los principales elementos de entrada.

- Crear un formulario que incluya los elementos y propiedades de los siguientes puntos. **INSTRUCCIONES**
- Añadir los campos para introducir los datos de un usuario: ID, Nombre de usuario, email, fecha de nacimiento y un campo que permita elegir el tipo de usuario de una lista de valores (Superusuario, Estandar, Lector).
- El campo ID deberá introducirse obligatoriamente y debe ser el que reciba el foco cuando se cargue la página.
- El ancho visible del campo de entrada para el nombre debe ser 50 y para el email de 30 caracteres.




30 m

5. La lista de tipos de usuario deberá realizarse con el elemento datalist.
6. Insertar un campo código postal que defina el patrón "`^[0-9]{5}$`". Mostrar un mensaje al usuario cuando el formato del CP introducido no cumpla con el patrón definido.
7. Incluir un área de texto que permita introducir un comentario de 500 caracteres.
8. Añadir un botón "Enviar" que permita el envío de los datos del formulario.
9. Añadir otro botón "Limpiar" para resetear o limpiar todos los campos.

ID:

Nombre de usuario:

Email:

Fecha de nacimiento:  
 

CP:

Tipo de usuario:  

Superusuario

Estandar

Lector



30 m

# Elaboración de documentos web mediante lenguajes de marca

## 4. Organización de elementos

- Tipos de display
- Posicionamientos no estáticos.
- Elementos flotantes

# HTML5

## Tipos de display

La propiedad 'display' define el tipo de caja que generará el elemento y permite modificar el tipo por defecto de la misma.

- **Block** → caja de bloque con salto de línea antes y después del elemento. El elemento ocupa todo el espacio definido por un ancho (width) y un alto (height). En caso de omitirse el alto la caja irá creciendo según se añada contenido.

`<div class='caja4'>Esta caja es display:block con salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height</div>`

`<div class='caja4'>Esta caja es display:block con salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height</div>`

Esta caja es display:block con salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height

---

Esta caja es display:block con salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height

# HTML5

## Tipos de display

- **Inline** → caja en línea sin saltos de línea ni antes ni después del elemento. El elemento inline ocupa solo el espacio que necesita su contenido

`<div class='caja1'>Esta caja es display:inline sin salto de línea y su ancho se ajusta a este contenido</div>`

`<div class='caja1'>Esta caja es display:inline sin salto de línea y su ancho se ajusta a este contenido</div>`

Esta caja es display:inline sin salto de línea y su ancho se ajusta a este contenido Esta caja es display:inline sin salto de línea y su ancho se ajusta a este contenido

# HTML5

## Tipos de display

- **Inline-block** → caja de bloque pero sin saltos de línea ni antes ni después del elemento. Al igual que el display **block** permite definir un ancho y alto

`<div class='caja2'>Esta caja es display:inline-block sin salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height</div>`

`<div class='caja2'>Esta caja es display:inline-block sin salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height</div>`

Esta caja es display:inline-block sin salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height

Esta caja es display:inline-block sin salto de línea y su ancho lo determina el contenido o, si lo tiene especificado, el atributo width y height

El atributo **vertical-align:top** permite que las cajas queden alineadas por su borde superior

Entre cajas contiguas siempre hay un espacio en blanco que puede evitarse si en la capa contenedora de ambas se incluye el parámetro **font-size:0px**

# HTML5

## Tipos de display

- **None** → se oculta la caja de forma que no sea visible y sin ocupar tampoco ningún espacio en el documento web

`<div class='oculta'>Esta caja es display:none y no se visualiza ni ocupa espacio</div>`

`<div class='caja3'>Esta caja ocupa el espacio de la anterior</div>`

Existe otra propiedad CSS que también oculta cajas o etiquetas y es :

`visibility:hidden;`

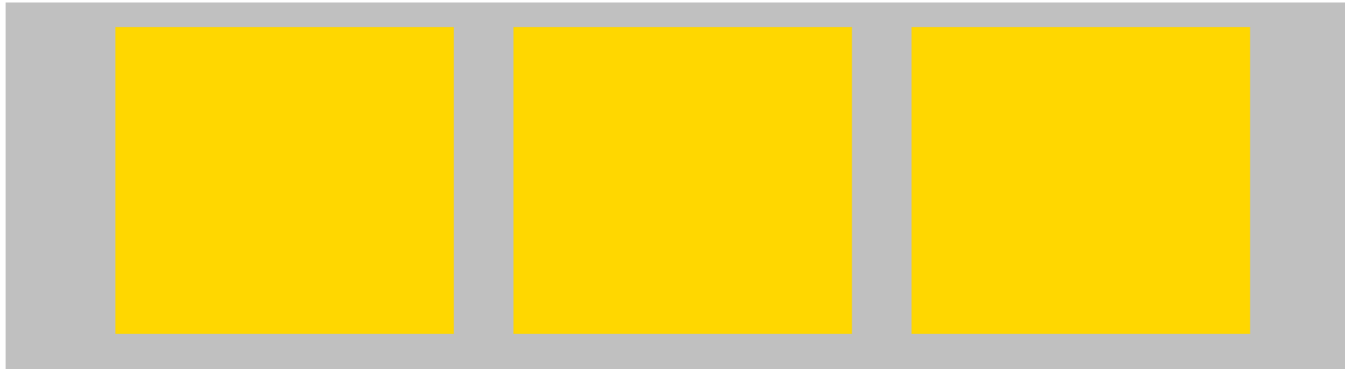
La diferencia es que esta última oculta la caja pero no libera el espacio que ocupa la caja en su estado visible.



# HTML5

## Tipos de display

Cuando varias cajas se disponen unas al lado de otras mediante **inline-block**, es recomendable que éstas se encuentren dentro de una caja contenedora de forma que se pueda centrar a la vez todas ellas en la página



Para centrar cajas con `display:inline-block` SI podemos utilizar la propiedad `text-align:center`

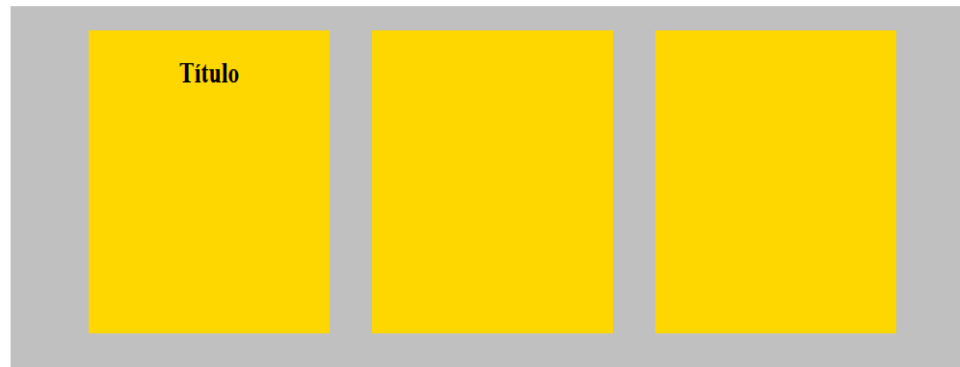
Para centrar cajas con `display:block` tenemos que utilizar `margin:auto`

# HTML5

## Tipos de display

Si intentamos poner, por ejemplo, un título con `h2` dentro de una caja (o cualquier otra etiqueta que lleve implícito un salto de línea) veremos como nos empuja la caja hacia abajo.

Para evitarlo tendremos que incluir un `vertical-align:top` en las cajas amarillas (las que tienen `display:inline-block`)

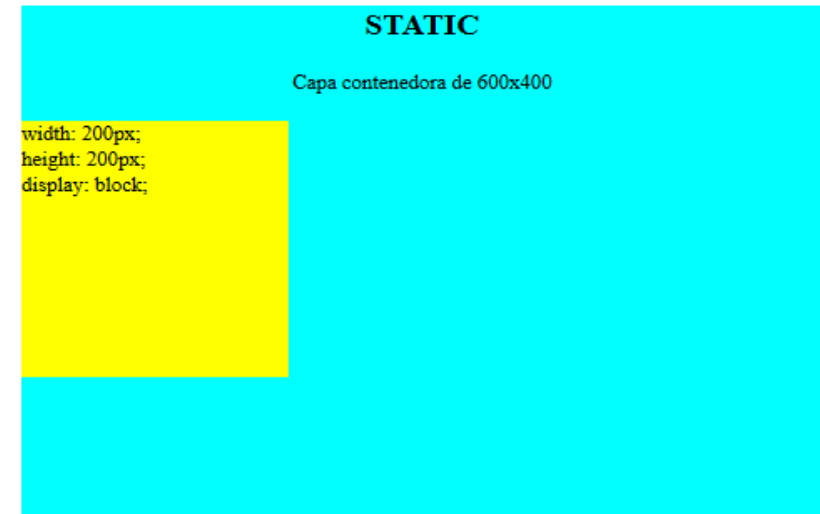


# HTML5

## Posicionamiento de elementos

Esta propiedad define el modo en que se colocan los elementos de un documento HTML en la ventana del navegador. Los posibles valores son ([http://www.w3schools.com/css/css\\_positioning.asp](http://www.w3schools.com/css/css_positioning.asp)):

**Static** → Valor predeterminado. El elemento se coloca según el flujo natural de los elementos y según la propiedad `'display'`

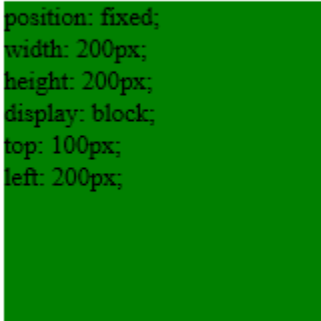


# HTML5

## Posicionamiento de elementos

**Fixed** → El elemento queda anclado en una posición fija en pantalla que marcarán las propiedades **top**, **bottom**, **left** y **right** y no se mueve ni siquiera realizando scroll de pantalla.

```
position: fixed;  
width: 200px;  
height: 200px;  
display: block;  
top: 100px;  
left: 200px;
```



**FIXED**

Capa contenedora de 600x2400



# HTML5

## Posicionamiento de elementos

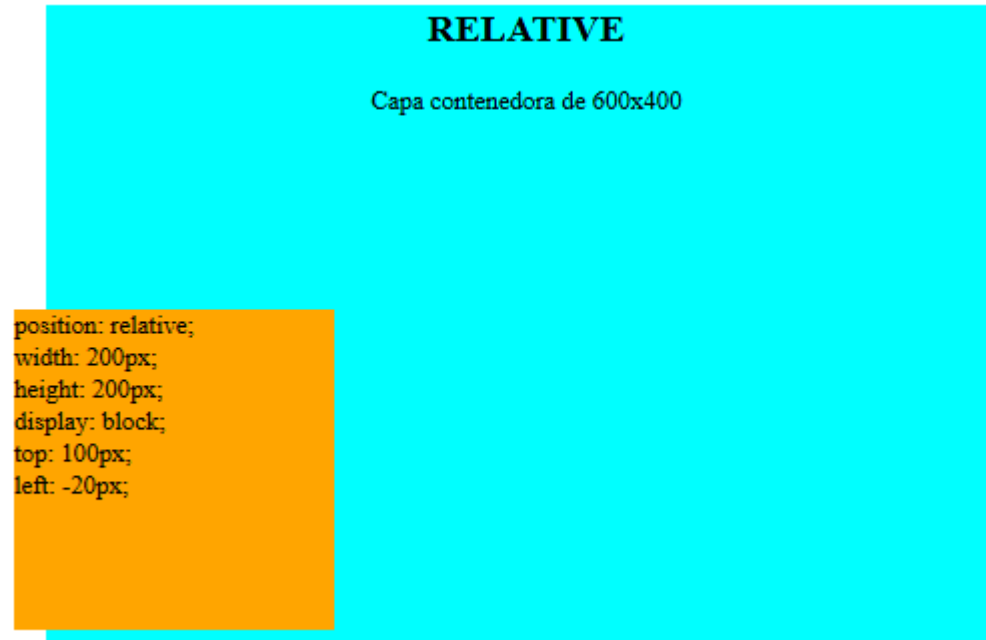
**sticky** → El elemento queda anclado en una posición fija en pantalla en el momento en que llegue a la posición marcada por la propiedad **top**.



# HTML5

## Posicionamiento de elementos

**Relative** → El elemento se colocará en una posición relativa a la que le correspondería en posicionamiento normal (**static**) y en base a distancias indicadas por **top**, **bottom**, **left** y **right** que pueden tener valores positivos y negativos.

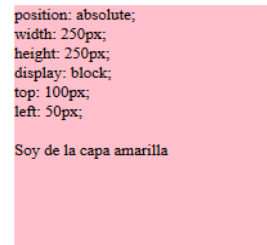


# HTML5

## Posicionamiento de elementos

**Absolute** → El elemento se colocará en una posición relativa al primer elemento padre que tenga un posicionamiento distinto de 'static' y en base a las distancias marcadas por **top**, **bottom**, **left** y **right**

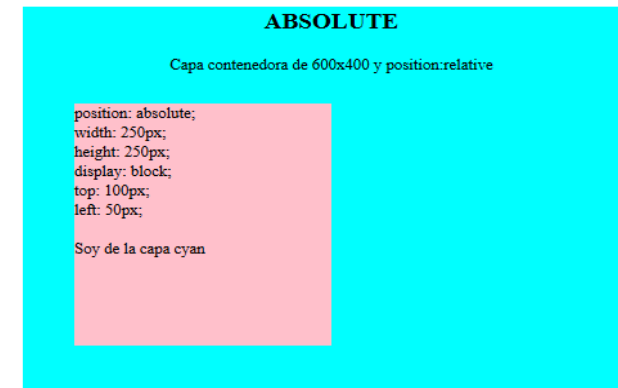
```
position: absolute;  
width: 250px;  
height: 250px;  
display: block;  
top: 100px;  
left: 50px;  
  
Soy de la capa amarilla
```

A pink rectangular box representing an element with absolute positioning. The text inside specifies its dimensions (250px width and height), display type (block), and its position relative to the top and left edges (100px top, 50px left). A label at the bottom identifies it as belonging to the yellow layer.

**ABSOLUTE**

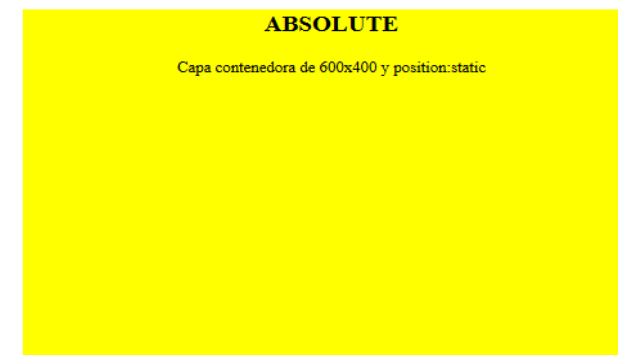
Capa contenedora de 600x400 y position:relative

```
position: absolute;  
width: 250px;  
height: 250px;  
display: block;  
top: 100px;  
left: 50px;  
  
Soy de la capa cyan
```

A diagram illustrating absolute positioning within a relative container. A cyan rectangular box represents the container, labeled 'ABSOLUTE' and 'Capa contenedora de 600x400 y position:relative'. Inside it, a pink rectangular box represents the element, with the same CSS properties as in the first diagram. The text inside the pink box identifies it as belonging to the cyan layer.

**ABSOLUTE**

Capa contenedora de 600x400 y position:static

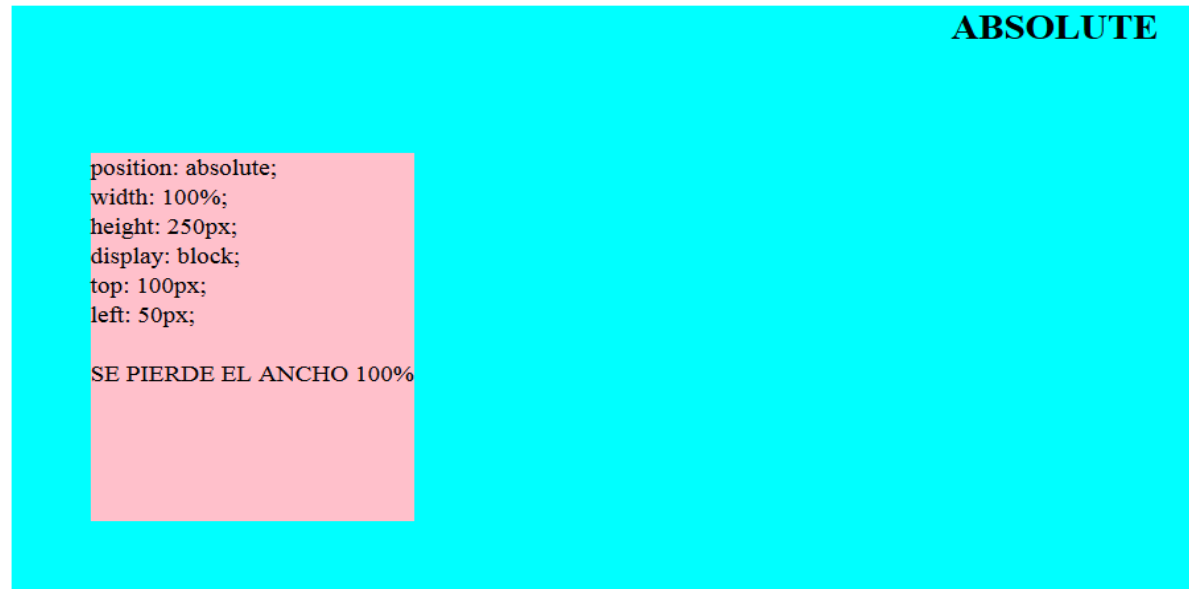
A diagram illustrating static positioning. A yellow rectangular box represents the container, labeled 'ABSOLUTE' and 'Capa contenedora de 600x400 y position:static'. It is empty, representing an element with the default static positioning.

# HTML5

## Posicionamiento de elementos

Cuando utilizamos posicionamiento no natural (**absolute** o **fixed**) con una caja que ocupa el 100% del ancho de la pantalla, ésta se ajustará al ancho del contenido de forma automática.

Para evitarlo debemos indicar expresamente **width:100%** con estos elementos

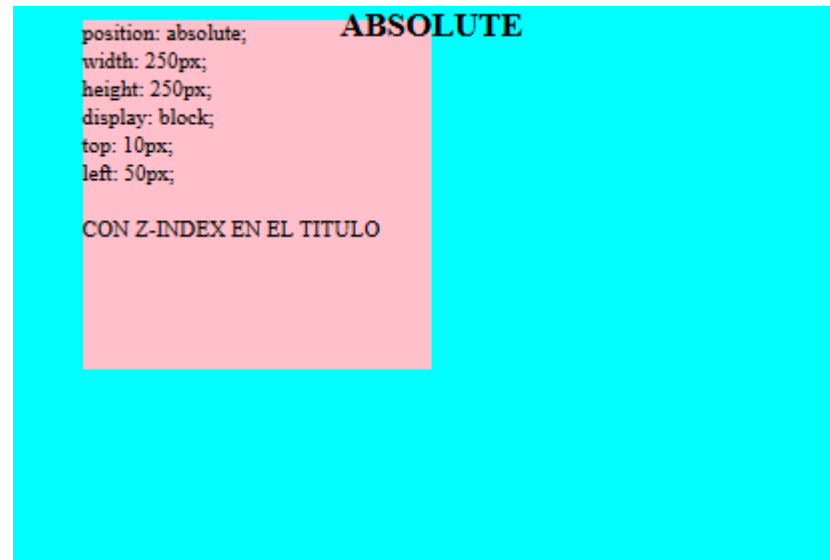
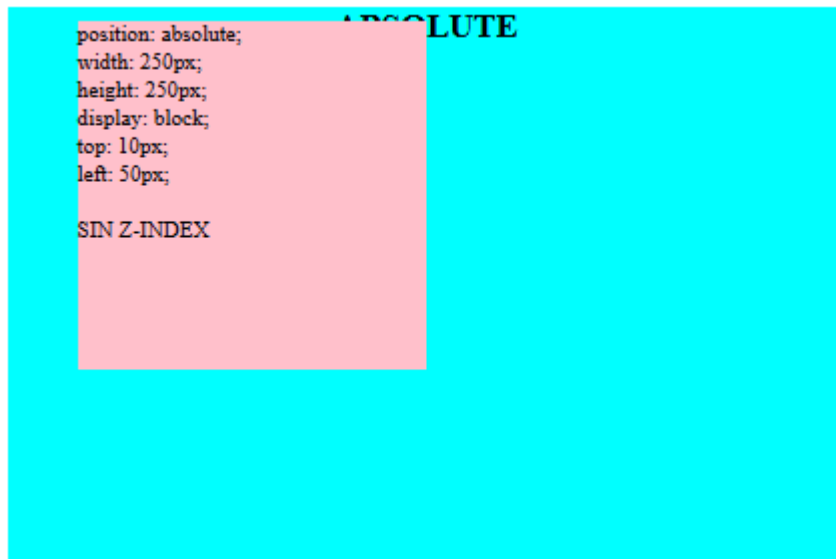




# HTML5

## Posicionamiento de elementos

Cuando utilizamos posicionamiento no natural (**static**) corremos el riesgo de superponer el elemento a otro ya existente. En este caso se utiliza la propiedad CSS **z-index** para indicar el orden del elemento en la pila (si se sitúa por encima o por debajo de ellos)



NOTA: para que el título aparezca por encima éste debe tener posicionamiento distinto de **static**

# HTML5

## Elementos flotantes

O como conseguir el siguiente e interesante efecto utilizando dos estilos CSS:



Lorem ipsum dolor sit  
amet, consectetur  
adipiscing elit, sed do  
eiusmod tempor incididunt  
ut labore et dolore magna  
aliqua. Ut enim ad minim  
veniam, quis nostrud  
exercitation ullamco laboris

nisi ut aliquip ex ea commodo consequat. Duis aute irure  
dolor in reprehenderit in voluptate velit esse cillum dolore eu  
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non  
proident, sunt in culpa qui officia deserunt mollit anim id est  
laborum.

`float:left`

La imagen flota a la izquierda sobre el texto

`margin:10px`

Definimos un margen de 10px alrededor

```
<p><img src='img/imagen.jpg' style='float:left; margin:10px'>Lorem ipsum...</p>
```

# HTML5

## Elementos flotantes

Si, a partir de un punto del texto, necesitamos que éste deje de envolver a la imagen deberemos utilizar **clear**:



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

**clear:left**

El texto deja de envolver la imagen con left

**clear:right**

El texto deja de envolver la imagen con right

**clear:both**

El texto deja de envolver la imagen

```
<p><img src='img/imagen.jpg' style='float:left>Lorem ipsum...</p><p style='clear:both'>Ut  
enum at...</p>
```

# Elaboración de documentos web mediante lenguajes de marca

## 5. Hojas de Estilo en Cascada (CSS)

- Elementos y estructura de una hoja de estilo: Creación de hojas de estilo. Aplicación de estilos. Herencia de estilos y aplicación en cascada. Formateado de páginas mediante estilos. Estructura de páginas mediante estilos.
- Diseño de estilos para diferentes dispositivos.
- Buenas prácticas en el uso de hojas de estilo.

# Hojas de Estilo en Cascada - CSS

- Lenguaje usado para **definir la presentación de un documento** estructurado escrito en HTML o XML.
- Una indicación de estilo recibe el nombre de **regla de estilo**.
- Mediante un **selector** se especifica a qué elemento HTML se aplican los estilos.
- El **World Wide Web Consortium (W3C)** es el encargado de definir la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

# Aplicación de Estilos

## Página HTML

Catálogo de programas:

- Lenguaje: Programación
- Business Integration: Integración con base de datos y web services
- Web Application: Aplicaciones web interactivas integradas con lógica de negocio

## Estilos CSS

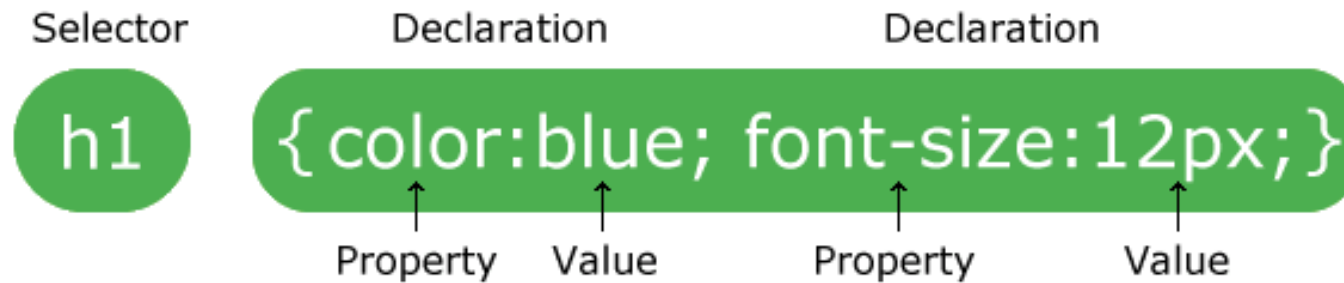
- color de fondo
- fuente de título: tamaño, tipo
- márgenes
- fuente de ítems: negrita y normal

## Página HTML con estilos

Catálogo de programas:

- **Lenguaje:** Programación
- **Business Integration:** Integración con base de datos y web services
- **Web Application:** Aplicaciones web interactivas integradas con lógica de negocio

# Regla de Estilo CSS



- Los **selectores CSS** se utilizan para especificar el/los elementos HTML a los que se desea aplicar estilo.

# CSS interno: Declaración de estilos con <style>

Utiliza la etiqueta <style>, a nivel de página.

```
<head>
  <style>
    h1 {
      color: blue;
      font-family: Verdana;
    }
  </style>
</head>
<body>
  <h1>Título con estilo</h1>
  <p style="color:red">En style se
    pueden definir estilos gráficos<br>
    para los elementos HTML</p>
</body>
```

La nomenclatura utiliza {llaves} y separación por ';' entre estilos, y ':' para la asignación.

Utiliza el atributo style localmente.

\*El uso de <style> en el <head> no es recomendado.



# CSS interno: Ejemplo

```
<head>
  <style>
    h1 {
      color: blue;
      font-family: Verdana;
    }
  </style>
</head>
<body>
  <h1>Título con estilo</h1>
  <p style="color:red">En style se
    pueden definir estilos gráficos<br>
    para los elementos HTML</p>
</body>
```

Este estilo se aplica a todos los elementos h1

Se define para los h1 el color azul y la familia (tipo de letra) Verdana para la fuente.

<p> tiene definido localmente el color de la fuente

**Título con estilo**


En style se pueden definir estilos gráficos para los elementos HTML

# CSS externo: Declaración de estilos con <link>

- La declaración de estilos con la etiqueta <style> en el head, desventajas:
  - No es reutilizable en otras páginas.
  - Combina en un mismo archivo los estilos gráficos y el contenido HTML.
- Buenas prácticas: estilos especificados en un archivo separado llamado **Hoja de Estilo o Cascade Style Sheet (CSS)**.

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

- Ejemplo de uso en la sección <head>:



Referencia a archivo CSS con la hoja de estilos.  
Puede incluir ruta relativa, absoluta o URL.

# CSS externo: Ejemplo

```
<head>
  <link type="text/css"
        rel="stylesheet"
        href="styles.css">
</head>
<body>
  <h1>Título con estilo</h1>
  <p style="color:red">En style se
    pueden definir estilos gráficos<br>
    para los elementos HTML</p>
</body>
```

styles.css

```
h1 {
  color: blue;
  font-family: Verdana;
}
```

## Título con estilo

En style se pueden definir estilos gráficos  
para los elementos HTML

Se pueden utilizar estilos locales (**CSS en línea**) para casos concretos.

# Selectores CSS: **universal**

- El **selector universal** (\*) representa **todos** los elementos HTML:

```
* {  
  color: blue;  
  text-decoration: none;  
}
```

# Selectores CSS: **elemento**

```
p {  
  color: blue;  
}  
  
a {  
  text-decoration: none;  
}  
  
div {  
  padding: 20px;  
}
```

← Todos los párrafos en color azul

← Hyperlink sin texto subrayado

← Espacio entre el borde y el contenido del elemento div

# Selectores CSS: id

Estilo asociado a id.  
Se marca con #.

```
#hjava {  
    color: red;  
}  
  
#hnet {  
    color: blue;  
}
```

id del elemento

```
<h3 id="hjava">Catálogo de cursos Java</h3>  
<h3 id="hnet">Catálogo de cursos .NET</h3>
```

**Catálogo de cursos Java**

**Catálogo de cursos .NET**

# Selectores CSS: **clase**

Nombre de clase. Se expresa con un punto

```
.hred {  
    color: red;  
}  
  
.hblue {  
    color: blue;  
}
```

Nombre de clase.

```
<h3 class="hred">Catálogo de cursos Java</h3>  
<h3 class="hblue">Catálogo de cursos .NET</h3>
```

**Catálogo de cursos Java**

**Catálogo de cursos .NET**

# Selectores CSS: **body**

- Cuando se definen estilos para `<body>`, se aplican a todo el contenido HTML.

```
body {  
    background-color: #F8F8F8;  
    font-family: Arial;  
}
```



# Selectores CSS: **Agrupación**

- Seleccionar elementos HTML con las mismas definiciones de estilo.

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

# Comentarios en CSS

- Sintaxis:

```
/* bloque comentado */
```

- El bloque comentado puede tener una o más líneas.

```
/*  
    Cuando un link es visitado recupera el color original  
*/  
a:visited { text-decoration: none; }
```

- No existen restricciones en el contenido de los comentarios en CSS.

# Principales Estilos

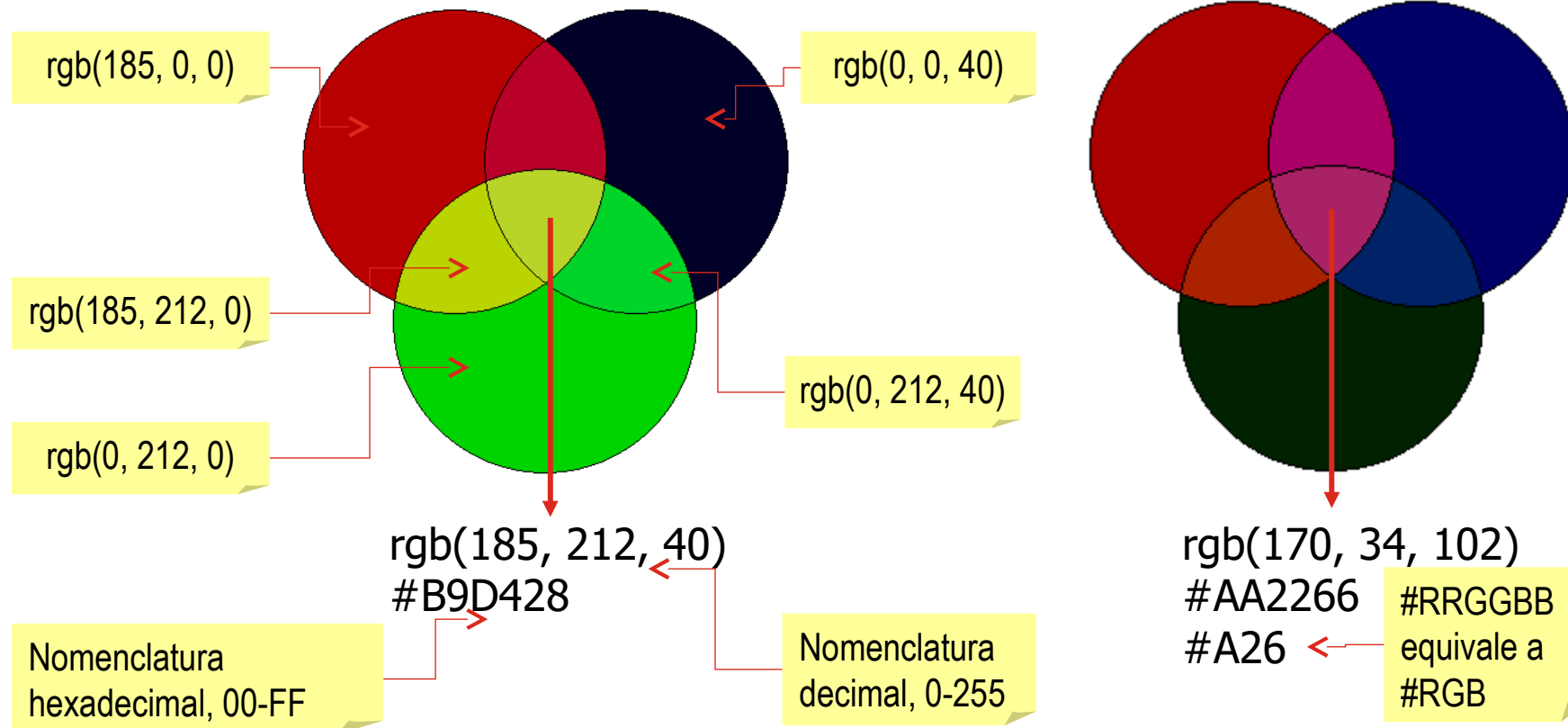
# Declaración de estilo: **Color**

- Permite definir colores utilizando **atributos** de estilo:
  - color: Color de fuente (texto)
  - background-color: Color de fondo
  - border-color: Color de bordes
- Especificación de un color mediante su nombre predefinido:



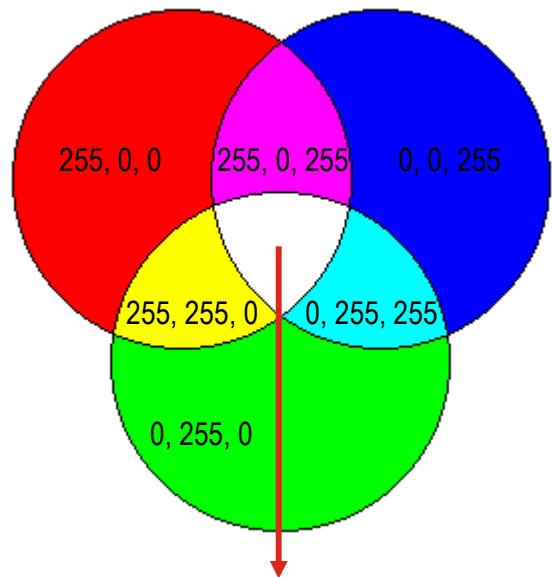
# Declaración de estilo: **Color**

- Modelo RGB (Red, Green, Blue):

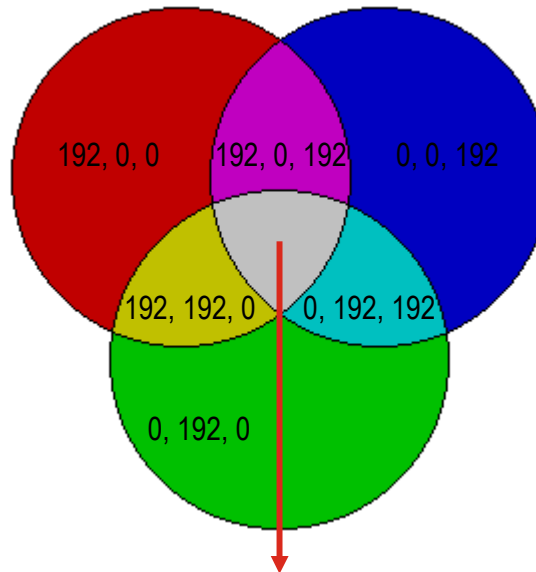


# Declaración de estilo: **Color**

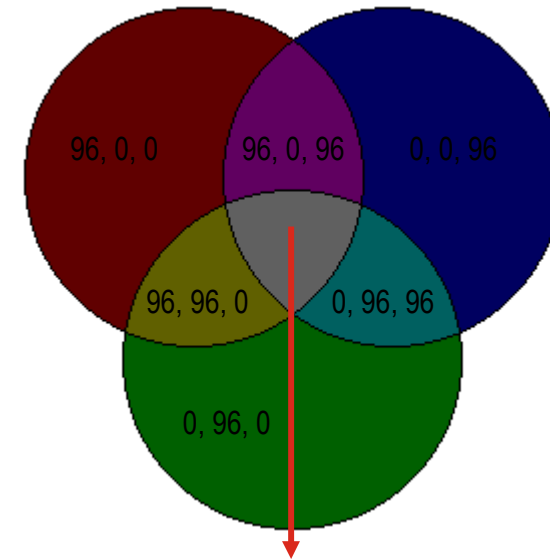
- Forma simplificada de RGB cuando el nivel es el mismo en los tres colores:



rgb(255, 255, 255)  
#FFFFFF  
#FFF



rgb(192, 192, 192)  
#C0C0C0



rgb(96, 96, 96)  
#606060

# Declaración de estilo: **Fuente**

- Principales atributos/propiedades de **tipografía** y sus posibles valores:

font-family	font-size	font-weight	font-style	font-variant
Arial	0.9em 9pt	400 normal	normal	normal
Verdana	1.0em 11pt	<b>700</b> <b>bold</b>	<i>italic</i>	SMALL-CAPS
Sans-serif	1.4em 15pt	lighter	<i>oblique</i>	

# Declaración de estilo: **Texto**

- Estilos para **decorar, alinear o transformar** el texto:

text-align	text-decoration	text-indent	text-shadow	text-transform
left	<u>underline</u>	0px	0em 0.13em gray	none
center	blink	5px	0.13em 0em gray	lowercase
right	<del>line-through</del>	10px	0.12em 0.12em gray	UPPERCASE
A small text with align justify	<u>overline</u>	20px	0.12em 0.12em red	Capitalize First Letter



# Declaración de estilo: **Link**

- Para el hyperlink **<a>**, se pueden definir estilos en función de su estado:

```
a { cursor: pointer; text-decoration: none; } <
```

Estilos para el hyperlink.  
En este caso, se suprime el subrayado.

```
a:link { } <
```

Estilos cuando aún no ha sido visitado.

```
a:visited { text-decoration: none; } <
```

Estilos cuando ya ha sido visitado. En este caso, se elimina el subrayado y recupera el color original.

```
a:hover { text-decoration: underline; } <
```

Estilos cuando se pasa el puntero sobre el hyperlink.

```
a:focus, a:active {text-decoration: underline;} <
```

Estilos cuando tiene el foco (focus), o cuando se presiona (active).

# Principales estilos: **Listas**

- El atributo **style** permite especificar el estilo de los elementos <li>.

```
<ul style="list-style: square">  
  <li>C# Básico</li>  
  <li>C# Avanzado</li>  
</ul>
```

- C# Básico
- C# Avanzado

```
<ul style="list-style: circle">  
  <li>C# Básico</li>  
  <li>C# Avanzado</li>  
</ul>
```

- C# Básico
- C# Avanzado

```
<ul style="list-style: url(play.png)">  
  <li>C# Básico</li>  
  <li>C# Avanzado</li>  
</ul>
```

- ▶ C# Básico
- ▶ C# Avanzado

# Principales estilos: **Listas**

- El atributo **style** permite especificar el estilo de los elementos <li>.

```
<ol style="list-style: upper-roman">  
  <li>Java Básico</li>  
  <li>Java Intermedio</li>  
  <li>Spring Core</li>  
  <li>Hibernate-JPA</li>  
  <li>Web Services</li>  
</ol>
```

I. Java Básico  
II. Java Intermedio  
III. Spring Core  
IV. Hibernate-JPA  
V. Web Services

```
<ol style="list-style: lower-alpha">  
  <li>Java Básico</li>  
  <li>Java Intermedio</li>  
  <li>Spring Core</li>  
  <li>Hibernate-JPA</li>  
  <li>Web Services</li>  
</ol>
```

a. Java Básico  
b. Java Intermedio  
c. Spring Core  
d. Hibernate-JPA  
e. Web Services

```
<ol style="list-style: decimal">  
  <li>Java Básico</li>  
  <li>Java Intermedio</li>  
  <li>Spring Core</li>  
  <li>Hibernate-JPA</li>  
  <li>Web Services</li>  
</ol>
```

1. Java Básico  
2. Java Intermedio  
3. Spring Core  
4. Hibernate-JPA  
5. Web Services

# Principales estilos: **Contenedor o Sección (div)**

- Si se requiere aplicar estilos a una sección en particular de una página, una opción bastante utilizada es a través de un `<div>` o `<span>`.
- Por ejemplo, una sección cuya fuente es de tipo Arial, y con color de fondo "wheat" (uno de los que no son principales):

```
.sample {  
  font-family: Arial;  
  background-color: Wheat;  
}
```

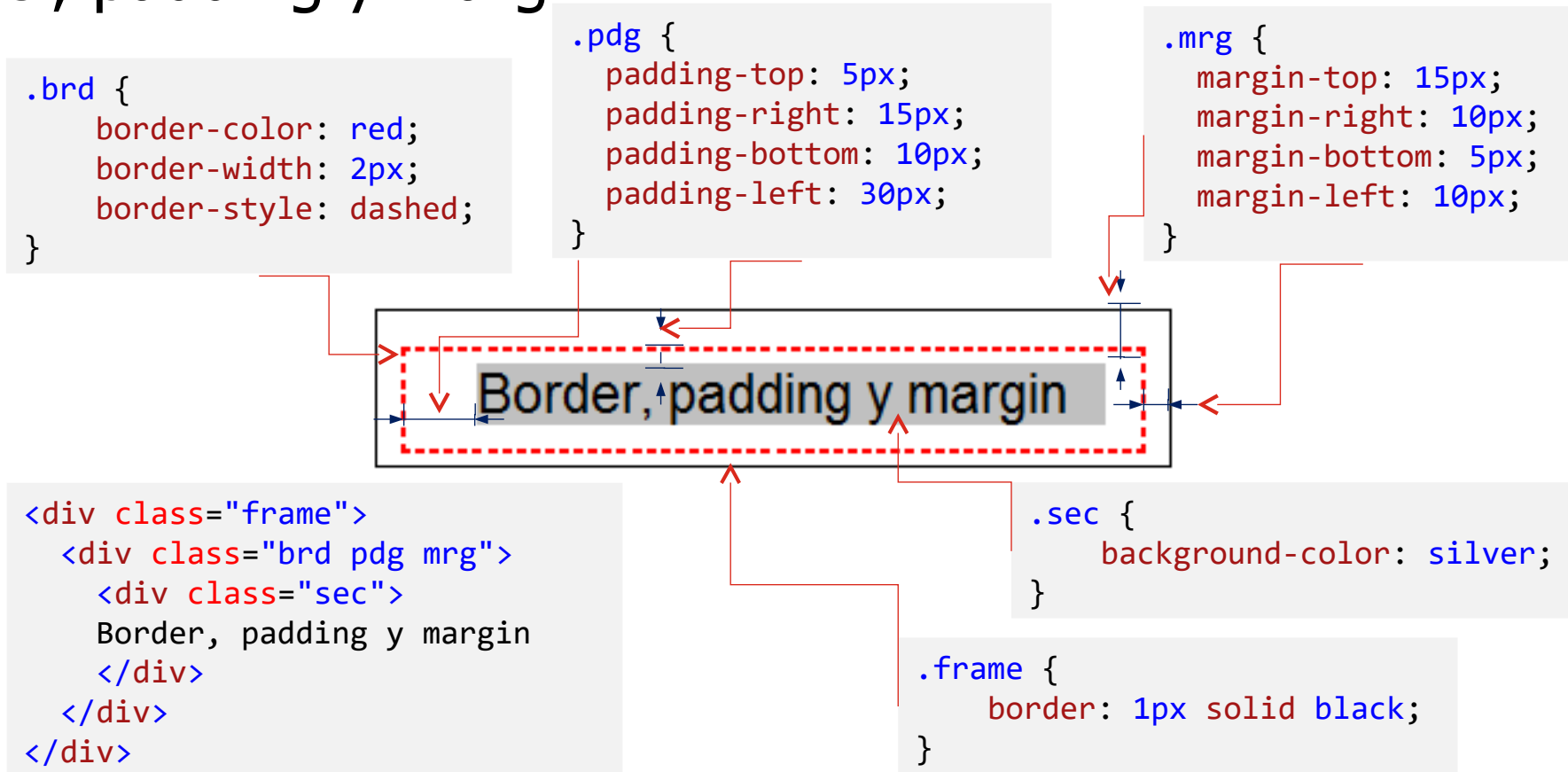
```
<div class="sample">  
  <h3>Título</h3>  
  <p>Contenido</p>  
</div>
```

**Título**

Contenido

# Principales estilos: **margin, border, padding**

- A un elemento HTML se le pueden definir los atributos gráficos **border, padding y margin**:



# Principales estilos: **margin, border, padding**

- Aplicando distintas combinaciones:

```
<div class="frame">
  <div class="brd">
    <div class="sec">Border</div>
  </div></div><br>
<div class="frame">
  <div class="brd pdg">
    <div class="sec">Border y padding</div>
  </div></div><br>
<div class="frame">
  <div class="brd pdg mrg">
    <div class="sec">Border, padding y margin</div>
  </div></div><br>
<div class="frame">
  <div class="pdg mrg">
    <div class="sec">Padding y margin</div>
  </div></div><br>
<div class="frame">
  <div class="mrg">
    <div class="sec">Margin</div>
  </div>
</div>
```

Border

Border y padding

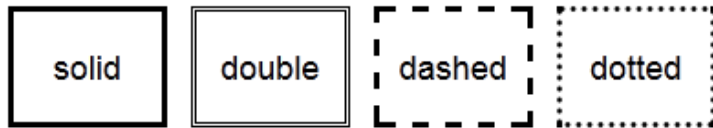
Border, padding y margin

Padding y margin

Margin

# Principales estilos: **border-style**, **border-color**

- Opciones para border:
- style:

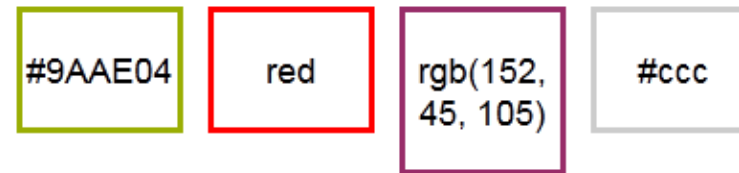


```
border-style: solid;
```

```
border-style: double;
```

```
border-style: dashed;
```

```
border-style: dotted;
```



```
border-color: #9AAE04;
```

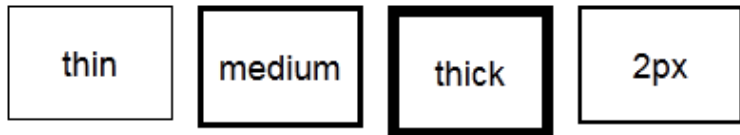
```
border-color: red;
```

```
border-color: rgb(152, 45, 105);
```

```
border-color: #ccc;
```

# Principales estilos: **border-width, border-radius**

- Opciones para border:
- width:

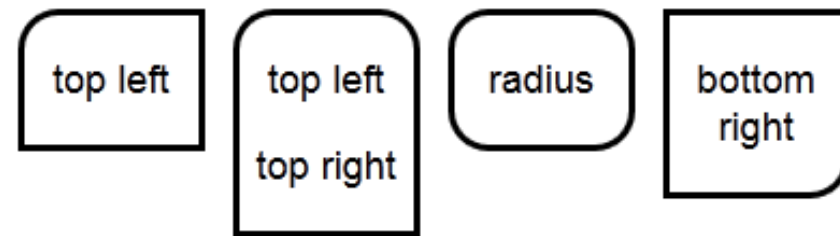


```
border-width: thin;
```

```
border-width: medium;
```

```
border-width: thick;
```

```
border-width: 2px;
```



```
border-top-left-radius: 15px;
```

```
border-top-left-radius: 15px;  
border-top-right-radius: 15px;
```

```
border-radius: 15px;
```

```
border-bottom-right-radius: 15px;
```



# Ejemplo de aplicación de estilos a una Tabla

```
table {  
  border: medium solid #6B002A;  
  caption-side: bottom;  
}  
td, th {  
  font-family: Arial;  
  border: thin dotted blue;  
  padding-left: 15px;  
}  
caption {  
  padding-top: 20px;  
  font-style: italic;  
}
```

Programa	Curso	
	Nombre	Duración
Business integration	Persistencia	24 horas
	Web services	24 horas

*Cursos de formación*

# Simplificación de estilos

- Algunas definiciones de estilos se pueden escribir de forma simplificada. Por ejemplo, el padding:

```
padding-top: 5px;  
padding-right: 15px;  
padding-bottom: 10px;  
padding-left: 30px;
```



```
padding: 5px 15px 10px 30px;
```

top right bottom left

orden "horario":

- El caso del margin es similar:

```
margin-top: 5px;  
margin-right: 15px;  
margin-bottom: 10px;  
margin-left: 30px;
```



```
margin: 5px 15px 10px 30px;
```

# Simplificación de estilos

- Si los cuatro valores son iguales, se puede colocar uno:


```
padding-top: 10px;  
padding-right: 10px;  
padding-bottom: 10px;  
padding-left: 10px;
```



```
padding: 10px;
```

- Similar es el caso de top-bottom y de left-right:

```
margin-top: 5px;  
margin-right: 15px;  
margin-bottom: 5px;  
margin-left: 15px;
```



```
margin: 5px 15px;
```



# Simplificación de estilos

- El caso del border también tiene un shorthand:

```
border-color: red;  
border-width: 2px;  
border-style: dashed;
```



```
border: red 2px dashed;
```

El orden no importa.

- En el caso de las fuentes, también:

```
font-size: 10pt;  
font-family: Arial, Helvetica, sans-serif;  
line-height: 1.2em;
```



```
font: 10pt/1.2em Arial, Helvetica, sans-serif;
```

**font:** font-style font-variant font-weight font-size/line-height font-family

# Principales estilos: **cursor**

- A una sección del documento HTML se le puede cambiar la forma del cursor, utilizando estilos. Esto mejora la interacción con el usuario, indicando por ejemplo que debe esperar, o donde debe presionar:

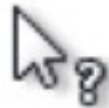
`cursor: pointer;`



`cursor: e-resize;`



`cursor: help;`



`cursor: n-resize;`



`cursor: crosshair;`



`cursor: move;`



`cursor: wait;`



`cursor: progress;`




# Combinación y herencia de estilos CSS

# Combinación de estilos

- Si se quiere que un elemento HTML tenga los estilos gráficos definidos en varias clases, se pueden combinar declarándolas juntas:

```
<element class="class1 class2 class3">...</element>
```



Las clases de estilo se separan por espacio.

# Combinación de estilos

```
.fnt1 {  
    color: blue;  
}  
.fnt2 {  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 14pt;  
}
```

```
<p class="fnt1">Fuente 1</p>
```

Fuente 1

```
<p class="fnt2">Fuente 2</p>
```

Fuente 2

```
<p class="fnt1 fnt2">Fuente 1 2</p>
```

Fuente 1 2

Combina atributos de las dos clases de estilo.



# Herencia de estilos

- En una estructura de etiquetas HTML, a través de CSS se pueden definir sus estilos, de modo que sólo se asigna la clase del elemento raíz. Ejemplo:

```
.frm {  
    border: 1px solid blue;  
    padding: 20px;  
}  
  
.frm fieldset {  
    background-color: silver;  
}  
  
.frm fieldset p {  
    font-family: Arial;  
    font-size: 1.2em;  
}
```

Sólo se especifica clase "raíz".

<fieldset> y <p> heredan los estilos

Estilo del <fieldset> dentro de un elemento con clase "frm".

```
<div class="frm">  
    <fieldset>  
  
        <legend>Datos</legend>  
        <p>Detalles</p>  
    </fieldset>  
</div>  
  
<br>  
<fieldset>  
    Otro fieldset  
</fieldset>  
<p>Otro párrafo</p>
```

No heredan estilos por estar fuera del div con class "frm"



Otro fieldset

Otro párrafo

# Herencia de estilos

- También se pueden especificar estilos de subelementos a través de su **id**, utilizando la nomenclatura con #:

```
.frm {  
    border: 1px solid blue;  
    padding: 20px;  
}  
.frm fieldset {  
    background-color: silver;  
}  
.frm fieldset #instr {  
    font-family: Verdana;  
    font-size: 1.2em;  
    color: navy;  
}  
.frm fieldset #details {  
    font-family: Arial;  
    font-size: 1.0em;  
    color: red;  
}
```

```
<div class="frm">  
  <fieldset>  
    <legend>Datos</legend>  
    <p id="instr">Instrucciones</p>  
    <p id="details">Detalles</p>  
  </fieldset>  
</div>
```



# Combinación y Herencia: Ejemplo

- Para las listas no ordenadas `<ul>`, se pueden definir estilos que permiten tener un aspecto gráfico similar a una barra de herramientas (toolbar). Aplicando herencia y los estilos vistos, resulta:

```
ul { list-style: none; }  
.tbar {  
  width: 28%;  
}  
.tbar ul li {  
  float: left;  
  padding: 2%;  
  border: 2px solid #fff;  
}  
.tbar ul li:hover {  
  border: 2px solid #ccc;  
}  
.tbar ul li a {  
  padding: 2px; width: 100%;  
}  
.tbar ul li a img {  
  border: 0;  
}
```

Elimina  
bullets

Distribución  
horizontal de los `<li>`

Mismo  
ancho

Paso de puntero  
sobre elemento `<li>`

Quita el borde (puesto por IE)

```
<div class="tbar">  
  <ul>  
    <li><a href="#" title="View"></a></li>  
    <li><a href="#" title="Export xls"></a></li>  
    <li><a href="#" title="New"></a></li>  
  </ul>  
</div>
```

Hyperlink a imagen.  
Simula un botón.



# Diseño Responsive

- Una página web responsive utiliza un despliegue visual dinámico, es decir, el contenido que se muestra **se ajusta a dispositivos** con diferentes capacidades, en especial, la manera en que se adapta a los diferentes tipos de pantalla.
- Puntos importantes:
  - Etiqueta viewport
  - Patrones de diseño responsive
  - Media queries
  - Breakpoints
  - Considerar si las imágenes son realmente necesarias





# ACTIVIDAD

---

**Menú de navegación CSS**

## OBJETIVO

**Diseñar un menú de navegación mediante elementos de una lista e hipervínculos definiendo el aspecto utilizando reglas CSS.**

## INSTRUCCIONES

- Crear una lista sin orden con cuatro elementos: Inicio, Sobre nosotros, Servicios y Contacto. Esta lista será nuestro menú de navegación.

```
<ul>
  <li><a href='#>Inicio</a></li>
  <li><a href='#>Sobre nosotros</a></li>
  <li><a href='#>Servicios</a></li>
  <li><a href='#>Contacto</a></li>
</ul>
```



25 m

2. Crear un estilo CSS para todos los elementos de la lista de manera que el menú se muestre de forma horizontal.

```
li {  
    display: inline;  
}
```

3. Definimos ahora algunos de los estilos del menú: color de fuente y tipo, tamaño del menú, color de fondo e indicamos que el texto de los hipervínculos no se muestren subrayados.

```
li a {  
    text-decoration: none;  
    font-family: arial;  
    color: #fff;  
    background-color: #2175bc;  
    padding: 10px;  
    float: left;  
}
```

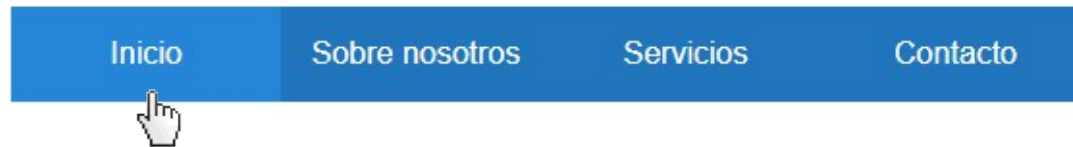
```
/* La propiedad float va posicionando cada elemento del menú a  
la izquierda de su contenedor. */
```



25 m

4. Añadimos un estilo para que, al pasar con el puntero del ratón por encima de cada elemento del menú, cambie su color de fondo.

```
• li a:hover {  
•     background-color: #2586d7;  
• }
```



25 m



# Elaboración de documentos web mediante lenguajes de marca

## 5. Multimedia

- Audio
- Video

# Multimedia: Tipos

- Multimedia hace referencia a cualquier objeto o sistema que utiliza múltiples medios de expresión digitales para presentar la información.
- Los medios pueden ser variados, desde texto e imágenes, gráficos, animación, sonido, video, alto nivel de interactividad.
- La realidad virtual es una extensión de multimedia que utiliza los elementos básicos de ésta como imágenes, sonido y animación.



# Multimedia: Audio

- Formatos y Compatibilidad con navegadores

Formato	Media Type
MP3	audio/mpeg
WAV	audio/wav
OGG	audio/ogg

Browser	MP3	WAV	OGG
Edge/IE	SI	NO	NO
Chrome	SI	SI	SI
Firefox	SI	SI	SI
Safari	SI	SI	NO
Opera	SI	SI	SI

# Multimedia: Audio

- Para incluir y reproducir un archivo de audio se usa el elemento HTML **<audio>**

```
<audio controls>  
  <source src="opera.ogg" type="audio/ogg">  
  <source src="opera.mp3" type="audio/mpeg">  
  Your browser does not support the audio tag.  
</audio>
```

- El atributo controls muestra controles de audio, como reproducción, pausa y volumen.
- El elemento **<source>** permite especificar archivos de audio alternativos entre los que el navegador puede elegir. El navegador utilizará el primer formato reconocido.
- El texto entre **<audio>** **</audio>** sólo se mostrará en los navegadores que no lo soportan. Navegadores compatibles: Chrome 4.0, Internet Explorer 9.0, Firefox 3.5, Opera 10.5 y Safari 4.0.

# Multimedia: Video

- Formatos y Compatibilidad con navegadores

Formato	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

Browser	MP4	WebM	Ogg
Edge	SI	SI	SI
Chrome	SI	SI	SI
Firefox	SI	SI	SI
Safari	SI	SI	NO
Opera	SI	SI	SI

# Multimedia: Video

- Para incluir y reproducir un video se usa el elemento HTML **<video>**

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  Your browser does not support the video tag.  
</video>
```

- El atributo controls muestra controles de video, como reproducción, pausa y volumen. Podemos usar autoplay para que el video se inicie automáticamente.
- Es recomendable especificar los atributos width y height (ancho y alto).

# Multimedia: Video

- El elemento `<source>` permite especificar archivos de video alternativos entre los que el navegador puede elegir. El navegador utilizará el primer formato reconocido.
- El texto entre `<video>` `</video>` sólo se mostrará en los navegadores que no lo soportan. Navegadores compatibles: Chrome 4.0, Internet Explorer 9.0, Firefox 3.5, Opera 10.5 y Safari 4.0.

# Multimedia

Formato	Extensión	Descripción
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML.
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers.
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. Does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.
WebM	.webm	WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and recommended by YouTube.
Formatos compatibles con el estándar HTML: MP4, WebM y Ogg		





- HTML5
- <http://www.w3.org/TR/html5/>
- CSS3
- <https://jigsaw.w3.org/css-validator/>
- World Wide Web Consortium (W3C)
- <https://www.w3.org/>

# Bibliografía y webgrafía



- Webs de referencia
- <http://www.w3schools.com/>
- <http://stackoverflow.com/>
- <http://www.codecademy.com/>

Los derechos de propiedad intelectual sobre el presente documento son titularidad de David Alcolea Martinez Administrador, propietario y responsable de [www.alcyon-it.com](http://www.alcyon-it.com) El ejercicio exclusivo de los derechos de reproducción, distribución, comunicación pública y transformación pertenecen a la citada persona.

Queda totalmente prohibida la reproducción total o parcial de las presentes diapositivas fuera del ámbito privado (impresora doméstica, uso individual, sin ánimo de lucro).

La ley que ampara los derechos de autor establece que: “La introducción de una obra en una base de datos o en una página web accesible a través de Internet constituye un acto de comunicación pública y precisa la autorización expresa del autor”. El contenido de esta obra está protegido por la Ley, que establece penas de prisión y/o multa, además de las correspondientes indemnizaciones por daños y perjuicios, para quienes reprodujesen, plagiaran, distribuyeren o comunicaren públicamente, en todo o en parte, o su transformación, interpretación o ejecución fijada en cualquier tipo de soporte o comunicada a través de cualquier medio.

El usuario que acceda a este documento no puede copiar, modificar, distribuir, transmitir, reproducir, publicar, ceder, vender los elementos anteriormente mencionados o un extracto de los mismos o crear nuevos productos o servicios derivados de la información que contiene.

Cualquier reproducción, transmisión, adaptación, traducción, modificación, comunicación al público, o cualquier otra explotación de todo o parte del contenido de este documento, efectuada de cualquier forma o por cualquier medio, electrónico, mecánico u otro, están estrictamente salvo autorización previa por escrito de David Alcolea. El autor de la presente obra podría autorizar a que se reproduzcan sus contenidos en cualquier sitio web u otro soporte (libro, revista, e-book, etc.) siempre y cuando se produzcan dos condiciones:

1 Se solicite previamente por escrito mediante email o mediante correo ordinario.

de aceptación, no se modifiquen los textos y se cite la fuente con absoluta claridad

**SOC**

Servei d'Ocupació  
de Catalunya

**David Alcolea**  
**david-alcolea@alcyon-it.com**  
**www.alcyon-it.com**



**Generalitat  
de Catalunya**



**Unió Europea**  
**Fons social europeu**  
L'FSE inverteix en el teu futur

Queda totalmente prohibida la reproducció total o parcial de las presentes diapositivas fuera del ámbito privado (impresora doméstica, uso individual, sin ánimo de lucro) así como su distribución sin la autorización explícita y por escrito de su autor.