

---

## Tema 2

# CSS. Hojas De Estilo En Cascada



# El Origen De CSS

- Con XHTML, se tenía poco control sobre el aspecto de los documentos
  - Ofrece muy pocas opciones para dar formato al texto.
  - Su función es estructurar el texto en secciones, como tablas, listas, párrafos, etc., pero en cuanto a diseño visual resulta pobre.
- El diseñador de páginas Web se encontraba con archivos HTML liosos
  - Ya que en el mismo texto se agrupan contenidos, estructuras e instrucciones de formato.
  - Esto hacía que la modificación y elaboración resultase muy complicada.
- La modificación del formato utilizado en un sitio Web compuesto por un gran número de páginas resultaba muy complejo
  - Ya que se tenían que cambiar manualmente los estilos de cada una de las páginas que formaban el sitio.

# El Origen De CSS

- Para solucionar estos problemas se creó un sistema mediante el cual las instrucciones de formato se encuentren separadas del resto de elementos.
- Aparece **CSS**
  - También denominado hojas de estilo en cascada.
  - Es un lenguaje para dotar de formato a los elementos HTML y XHTML.
  - Mediante una simple instrucción, se puede aplicar un determinado formato a todo un sitio Web.
  - Permite que los sitios Web tengan un aspecto homogéneo sin necesidad de muchas horas de trabajo.
  - Ofrece más posibilidades que las etiquetas de HTML.

# Hojas De Estilo En Cascada

## ■ Hoja de estilo

- Se denomina así, ya que es un conjunto de instrucciones que definen los formatos de los elementos HTML a los que afecta.
- Son archivos de texto con un formato muy simple, por lo que se facilita la claridad del código generado.
- El código que compone la hoja de estilo está formado por una o más reglas de estilo.
  - Que identifica el elemento HTML que se desea seleccionar y la apariencia que se le quiere dar.

## ■ En cascada

- Hace referencia a que a la hora de crear un estilo las especificaciones que se realicen para un determinado elemento serán aplicables a todos aquellos elementos que se encuentren "por debajo" de él, atendiendo a un criterio de herencia.
  - Así, el elemento heredará el tipo de formato salvo que especifiquemos lo contrario, por lo que no necesitaríamos volver a indicar esa característica.

# Hojas De Estilo En Cascada

- En la actualidad hay 3 versiones de hojas de estilo:
  - CSS1
  - CSS2
    - Ofrece muchas mejoras respecto a CSS1.
    - Hay que tener cuidado con posibles problemas de compatibilidad, ya que los navegadores Web no son compatibles con todas las especificaciones de la versión CSS2.
  - CSS3 está dividida en varios documentos separados, llamados "módulos".

Cada módulo añade nuevas funcionalidades a las definidas en CSS2, los primeros borradores de CSS3 fueron liberados en junio de 1999, a fechas de noviembre de 2011, hay alrededor de cincuenta módulos publicados, tres de ellos se convirtieron en recomendaciones oficiales de la W3C en 2011: *"Selectores"*, *"Espacios de nombres"* y *"Color"*.

# Hojas De Estilo En Cascada

- Una hoja de estilos contiene datos de formato relativos a los elementos definidos en la página HTML.
  - Por tanto, ambos (documento HTML y hoja de estilo) deben estar relacionados de alguna manera.
- Existen tres formas de dotar de estilo a una página Web mediante CSS
  - Aplicar estilos de forma local.
  - Mediante hojas de estilo internas.
  - Mediante hojas de estilo externas.

# Las Hojas De Estilo Y Sus Clases

- Aplicar estilos **de forma local**

- Se utiliza para aplicar estilos a determinados elementos HTML directamente.
- Las hojas de estilo locales son las menos utilizadas porque no presentan casi ninguna ventaja frente al uso tradicional de HTML.
- Se escribe el código CSS dentro del atributo “style” de la etiqueta HTML en cuestión, de la forma:

`<P style="código_CSS;"> texto </P>`

- Ejemplo:

`<P style="color:green;"> HOLA </P>`

# Las Hojas De Estilo Y Sus Clases

- Aplicar estilos mediante **hojas de estilo internas**
  - Se utiliza cuando se pretende aplicar el estilo sólo a la página en la que se utiliza.
  - El código de la hoja de estilo interna se encontrará entre las etiquetas

```
<HTML>  
  <HEAD>  
    <STYLE type="text/css">  
      código_CSS  
    </STYLE>  
  </HEAD>  
  <BODY>  
    ...  
  </BODY>  
</HTML>
```



# Las Hojas De Estilo Y Sus Clases

- Aplicar estilos mediante **hojas de estilo externas**
  - Se usan para dar un aspecto común a varias páginas de un portal.
  - Todos los estilos se definen en una hoja de estilo externa para posteriormente hacer que todas las páginas del portal la consulten.
    - Se escribe su código en un documento sin formato y se guarda con la extensión “.css”
    - Luego se vincula a la página Web mediante la inclusión de la siguiente etiqueta en la cabecera del documento HTML en cuestión.

<HEAD>

<LINK rel=“stylesheet” type=“text/css” href=“url\_de\_la\_hoja\_de\_estilo.css”>

</HEAD>

- **Prioridad** entre los distintos estilos:
  - + Estilos locales.
  - Estilos definidos en la hoja de estilos interna.
  - Estilos definidos en la hoja de estilos externa.

# Las Hojas De Estilo Y Sus Clases

- Los archivos de hojas de estilo tienen extensión .css y están formados por un conjunto de reglas puestas una a continuación de otra:

```
body {  
    background: white; font-size: 11pt;  
    font-family: Arial, Helvetica, Sans-serif;  
    color: black; text-align: justify;  
}  
p {  
    text-indent: 2em;  
}
```

- Por tanto, las hojas internas serán exactamente igual a las externas pero estarán contenidas en el mismo documento HTML en vez de en un archivo aparte.

# Ejercicios

- 1) Aplica un estilo CSS de forma local a un párrafo. Dicho estilo consistirá en poner el texto de color azul y con tamaño de letra de 15 puntos.
- 2) Aplica un estilo CSS mediante una hoja de estilo interna a un párrafo. Dicho estilo consistirá en poner el texto de color verde y con tamaño de letra de 8 puntos.
- 3) Crea un nuevo archivo de texto llamado "estilo.css" y escribe en él el siguiente código css.

```
p{  
  color: blue;  
  text-indent: 2em;  
  text-align: right;  
}
```

A continuación, guarda el archivo y sitúalo en la misma carpeta que el documento HTML y modifica éste último para que utilice la hoja de estilos externa.

# Solución Ejercicios

1) `<P style="color:blue; font-size:15pt;"> este es mi párrafo </P>`

2) `<HEAD>`

`<STYLE type="text/css">`

`p{color:green; font-size:8pt;}`

`</STYLE>`

`</HEAD>`

- Todos los párrafos del documento se ponen de color verde con tamaño de letra 8, exceptuando el primero, que tiene un estilo local.
  - Esto se debe a la regla de prioridad de estilos.

3) `<HEAD>`

`<LINK rel="stylesheet" type="text/css" href="url_de_la_hoja_de_estilo.css">`

`</HEAD>`

- Observa que los párrafos se alinean a la derecha y que aparecen con sangría, pero el color del texto no se vuelve azul, tal y como hemos especificado en el archivo css. Esto es porque la hoja de estilos externa tiene menos importancia que la interna, la cual dice que el texto va en color verde y en tamaño 8.

# Estructura Genérica De Un Documento CSS

- Como podemos ver, las reglas de estilo tienen dos componentes:
  - El selector
    - Seleccionará los elementos sobre los que actuará la regla de estilo.
    - Sin él, el navegador no sabría a qué debe aplicarle el formato que a continuación viene encerrado entre llaves.
  - La declaración
    - Indica a qué propiedad (ó atributo) del elemento se le va a dar un estilo o formato concreto y establece el valor que se le da a dicha propiedad.

Selector { propiedad1:valor1; propiedad2:valor2;...;}

- A este tipo de selector se le denomina **selector por tipo de elemento**.
  - Aplica un determinado formato a todos los elementos seleccionados mediante la regla de estilo, sin distinción de clases, identificadores, contextos, etc.

# Selector Por Tipo De Elemento

## ■ Ejemplo:

```
IMG {  
    border-color: green;  
    border-width: 2px;  
    border-style: solid;}
```

- En este caso se aplicarían los estilos a todas las etiquetas <IMG> que aparezcan en el documento HTML, sin importar qué identificador lleven o a qué clase pertenezcan.
- Sin embargo, ¿sería posible actuar solamente en unas imágenes determinadas y dejar el resto con otro formato?

## ■ Para ello, se han desarrollado unos selectores especiales más específicos

- Identificadores.
- Clases.
- Pseudo-clases.
- Pseudo-elementos.

# Identificadores Y Clases

## ■ Identificadores

- Su finalidad es la de asignar una identificación unívoca al elemento.
  - De esta forma, con CSS podremos seleccionarlo y distinguirlo de los demás.
  - Dentro de una página Web no puede haber más de un elemento con el mismo identificador y un mismo elemento sólo puede tener un único identificador.
- Los valores de este atributo deben comenzar por una letra seguida de un número cualquiera de caracteres alfanuméricos.
  - Se recomienda que sean valores que describan o clasifiquen de forma breve al elemento.

## ■ Clases

- Su finalidad es juntar los elementos por grupos.
  - Permite la especificación de características concretas a grupos heterogéneos de elementos.
- Un elemento puede pertenecer a varias clases, ya que éstas no son excluyentes.
- Una vez que hemos asignado una clase a un elemento, podemos hacer referencia a todos los elementos pertenecientes a una clase.

## ■ Ambos selectores son opcionales.

# Selector Por Identificador O Clase Del Elemento

- Si se etiquetan elementos con una clase o identificador, se puede utilizar este criterio para aplicar formato solamente a dichos elementos.
- Selector por identificador del elemento
  - Si queremos asignar al elemento un nombre que lo distinga de los demás, haremos:

`<P id="carta"> texto <P>`

- Una vez identificado el elemento, si lo que queremos es modificar el estilo de los elementos identificados
  - Escribiremos el nombre del elemento seguido de un separador (#) y su identificador, dentro de la hoja de estilo interna o externa:

`P#carta {font-size: 14px;}`



# Selector Por Identificador O Clase Del Elemento

- Selector por clase del elemento
  - Para asignar un elemento a una clase especificaremos dentro del atributo “class” los nombres de las clases separados por espacios.

```
<IMG src="casa.jpg" class="muebles">
```

```
<IMG src="cama.jpg" class="muebles habitacion">
```

- Para hacer referencia a todos los elementos pertenecientes a una clase, se escribe en la hoja de estilo el nombre del elemento seguido de un punto (.) y el nombre de la clase

```
IMG.muebles { border-width: 1px; border-color: red; }
```

- En este ejemplo, se crea un borde rojo de un píxel en las imágenes que pertenezcan a la clase muebles.

# Selector Por Identificador O Clase Del Elemento

- Si se omite el nombre del elemento, se verán afectados todos los que pertenezcan a dicha clase, independientemente del tipo de elemento que sea (párrafo, imagen, encabezado, etc.).
  - Siempre y cuando dichos atributos se puedan aplicar a los elementos en cuestión.
- Ejemplo:

```
.habitacion { border-width: 2px; border-style: dashed; }
```

- De esta manera, se crea un borde de línea discontinua de 2 píxeles en todos los elementos de la clase habitacion.
- ¿Qué formato tomaría cama.jpg?

# Selector Por Identificador O Clase Del Elemento

## ■ Ejemplos:

*/\*Se seleccionan todos los elementos de la clase impares\*/*

```
.impares {  
    border-color: green;  
    border-width: 2px;  
    border-style: solid;} /*significa un borde de línea continua.*/
```

*/\*Se seleccionan las imágenes que pertenezcan a la clase impares\*/*

```
img.impares {  
    border-color: green;  
    border-width: 2px;  
    border-style: solid;}
```

*/\*Se selecciona el párrafo con identificador dos\*/*

```
p#dos {  
    border-color: green;  
    border-width: 2px;  
    border-style: solid;}
```

# Pseudo-Clases

- Clasifican a los elementos según el estado del mismo.
  - Los elementos son agrupados en función de los eventos que les hayan ocurrido:
    - Que el enlace haya sido visitado.
    - Que el cursor del ratón se encuentre encima o que deje de estar sobre él.
    - Etc.
  - Las pseudo-clases son dinámicas ya que un elemento puede adquirirla o perderla a medida que el usuario interactúa con el documento.
- De los diferentes elementos con pseudo-clases existentes, los más utilizados son los referentes a los vínculos.
- En los selectores por pseudo-clase, se escribe el nombre de la etiqueta seguido de dos puntos (:) y el estado de ésta:

etiqueta:estado

- Ejemplo:
  - Si lo que queremos es cambiar el formato de aquellos vínculos visitados, haremos:

A:visited {color:blue;}

# Pseudo-Clases

- Ejemplo:

- A:link { color: red } /\*enlace sin visitar\*/

- A:visited { color: blue } /\*enlace visitado\*/

- A:hover { color: yellow } /\*enlace con el ratón encima\*/

- A:focus { color: yellow } /\*enlace al que ha llegado con el teclado\*/

- A:active { color: lime } /\*enlace sobre el que se está haciendo  
“clic”\*/

- “A:hover” debe ir después de las reglas “A:link” y “A:visited”
  - de otro modo las reglas de cascada ocultarán la propiedad “color” de la regla A:hover.

# Pseudo-Elementos

- Son partes de elementos de una página a los que XHTML no otorga identidad propia y, por tanto, no pueden ser seleccionados por dicho lenguaje.
  - Sin embargo, sí que pueden ser identificados por las reglas de estilo y ser formateados de manera distinta a la de los elementos a los que pertenecen.
  - Por ejemplo, no existe ninguna etiqueta XHTML para marcar la primera letra o la primera línea de cada párrafo
    - En cambio, mediante los pseudo-elementos es posible acceder a ellas y dotarles de un estilo distinto al del párrafo.
  - Se hace referencia a ellas al igual que con las pseudo-clases.
    - Es decir, comienzan con dos puntos (:).
- Ejemplo:

```
p:first-letter { color:red;} //ponemos de color rojo la primera letra de todos los párrafos.
```

# Pseudo-Clases Y Pseudo-Elementos

Pseudo-Clase	Significado
:link	• Vínculos que todavía no se han visitado.
:visited	• Vínculos ya visitados.
:hover	• Vínculos sobre los que está el cursor del ratón.
:focus	• Vínculos sobre los que está pero que ha accedido por teclado.
:active	• Vínculos que han sido pinchados

Tipos de pseudo-clases que afectan a los vínculos

Pseudo-Elemento	Significado
:first-letter	• El primer carácter del elemento al que pertenece.
:first-line	• La primera línea del elemento al que pertenece.
:before	• El elemento anterior.
:after	• El elemento posterior.

Tipos de pseudo-elementos

# Selector Por Contexto Del Elemento

- De esta manera se pueden seleccionar los elementos en función de dónde se encuentran.
  - Por ejemplo, para indicar que se aplique una regla de estilo a los enlaces que se encuentren dentro de una determinada sección.
- Se pueden seleccionar elementos en función de:
  - Los ascendentes
  - Los padres
  - Los hermanos
- Vamos a ver cada uno de ellos:

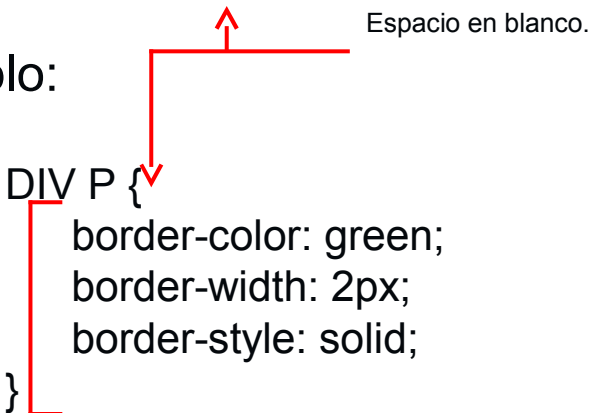


# Selector Por Contexto.

## Los Ascendentes

- Es el elemento que contiene al que se desea formatear.
- Puede haber más de un elemento entre uno y otro.

Ascendiente    Descendiente { propiedad: valor;}

- Ejemplo:

```
DIV P {  
  border-color: green;  
  border-width: 2px;  
  border-style: solid;  
}
```

Espacio en blanco.

- En este caso se aplicarían los estilos a todas los elementos <P> que se encuentren dentro de una etiqueta <DIV>.

# Selector Por Contexto.

## Los Padres

- Es el elemento que contiene directamente al que se desea formatear, al que llamaremos hijo.

Padre>Hijo { propiedad: valor;}

- Ejemplos:

```
DIV>IMG{  
border-color: green;  
border-width: 2px;  
border-style: solid;}
```

- En este caso se formatearían los elementos <IMG> que se encuentren dentro de una etiqueta <DIV>, sin que pudiera haber ninguna etiqueta en medio.

```
<DIV>  
  <P> párrafo... <IMG src="imagen">... </P>  
</DIV>
```

- En este caso, no se aplicaría el formato, ya que, aunque la etiqueta <IMG> está dentro de una etiqueta <DIV>, hay una etiqueta <P> entre medias, por lo que <IMG> no es hija de <DIV>, aunque sí descendiente.

# Selector Por Contexto.

## Los Hermanos

- Es el elemento que precede directamente, dentro del mismo elemento padre, al que se desea formatear.

HermanoPrecedente+HermanoQueSeSelecciona { propiedad: valor;}

- Ejemplos:

```
p+p {  
  border-color: green;  
  border-width: 2px;  
  border-style: solid;  
}
```

```
<div>  
  <p>párrafo1...</p> <!--Hermano precedente-->  
  <p>párrafo2...</p> <!--Hermano que recibe el formato-->  
</div>
```

- En este caso se formatearían los elementos <p> que se encuentren después de otro elemento <p>.

# Selector Por Los Atributos Del Elemento

- Se utiliza cuando se desea aplicar formato a aquellos elementos que tienen un determinado atributo o un valor de atributo dado.
- No son muy habituales.
- La forma de declararlo es:

elemento[atributo]

- Ejemplo:

td[bgcolor]{color:red;}

- Con éste, se seleccionan todas las celdas de una tabla (td) que tengan el atributo de color de fondo y se les aplica un color de texto rojo.

# Agrupar Y Combinar Selectores

## ■ Agrupar selectores

- Es aplicar las mismas reglas de estilo a más de un tipo de elemento.
- Para ello, se separan los selectores por una coma.
- Ejemplo:

```
td, p { color:blue;}
```

- Con esto se consigue formatear el texto de las celdas y los párrafos con color azul.

## ■ Combinar selectores

- Se trata de seleccionar aquellos elementos que cumplan más de un criterio a la vez, según las reglas de precedencia.
- Ejemplo:

```
div p.portada:first-letter {color:blue;}
```

- Con esto, se ponen de color azul la primera letra de los párrafos que pertenezcan a la clase “portada” y que estén dentro de una etiqueta <DIV>.

## ■ Ventajas

- Se consigue una selección más fina.
- Se ahorra código.
- Aumenta la legibilidad del código.

# La Precedencia

- Puede ocurrir que en un portal Web se hayan definido dos reglas de estilo para un mismo elemento. ¿Cuál se aplica?
  - El **principio de la cascada** decidirá cómo resolver conflictos generados por problemas de
    - Herencia.
    - Especificidad.
    - Ubicación.
- La Herencia
  - Hay propiedades CSS que afectan a los elementos definidos por el selector y también a sus descendientes.
  - Los elementos que heredan estas propiedades se ven afectados por ellos de manera implícita
    - Es posible que produzcan efectos indeseados en la apariencia de la página Web.

# La Precedencia

## ■ Especificidad

- Cuando se aplica más de una regla a un mismo elemento, existe un conflicto de especificidad.
- La ley de especificidad indica que cuanto más concreto es el selector, más valor tiene la regla.
- El navegador determina según un sistema de puntos cuál es el selector más importante.

Tipo de selector	Ejemplo	Puntos
■ Selector de tipo simple	p	1
■ Clase	.cuadro_info	10
■ Pseudoclase	:visited	10
■ ID	#navegacion	100

# La Precedencia

- La ubicación
  - Las reglas que aparecen en último lugar tienen más peso.
  - Por ejemplo: aquellas reglas que se aplican localmente en el elemento se consideran que aparecen después que las definidas en una hoja de estilo interna y, por tanto, que tienen más relevancia que éstas, y las que aparecen en una hoja de estilo interna tienen más relevancia que las que aparecen en una hoja de estilo externa.



# La Precedencia

## ■ Ejemplo:

```
<HTML>
  <HEAD>
    <TITLE>ejemplo</TITLE>
    <STYLE type="text/css">
      p {color: blue;}
      p.grupo {color: red;}
      p#tercero {color: black;}
    </STYLE>
  </HEAD>
  <BODY>
    <P> este párrafo se queda de color azul </P>
    <P class="grupo"> este se queda de color rojo </P>
    <P class="grupo" id="tercero"> este tendrá color negro </P>
    <P style="color: white"> y este último se quedará en color blanco </P>
  </BODY>
</HTML>
```

# Ejercicios

- **Ejercicio 1 →** Aplica un estilo CSS a los párrafos impares. Dicho estilo consistirá en poner el texto de color azul y con tamaño de letra de 7 puntos.
- **Ejercicio 2 →** Aplica un estilo CSS a los enlaces que aparezcan en el documento. Los enlaces normales deberán ser rojos (red), los visitados verdes (green), los activos amarillo (yellow) y los que tengan el ratón encima azules (blue).
- **Ejercicio 3 →** Aplica un estilo CSS para que la primera letra de cada párrafo tenga tamaño 18 puntos y la primera letra de cada encabezado h3 tenga un tamaño de 22 puntos.

# Solución Ejercicios

- Pondremos en la parte superior de la página HTML el código:  
`<LINK rel="stylesheet" type="text/css" href="estilo.css">`
- **Ejercicio 1** → utilizaremos clases.
  - En estilo.css, escribiremos:

```
.impares{  
  color: blue;  
  font-size: 7pt;  
}
```

- Y en la página HTML:  
  
`<p> párrafo par <P>`  
`<P class="impares"> párrafo impar <P>`  
`<p> párrafo par <P>`  
`<P class="impares"> párrafo impar <P>`  
`<p> párrafo par <P>`

# Solución Ejercicios

## ■ Ejercicio 2 → Utilizaremos las pseudo-clases.

### □ En estilo.css

```
a:link{color: red;}  
a:visited{color: green;}  
a:hover, a:focus{color: yellow;}
```

### □ En la página HTML

```
<a href="www.google.es"> esto es un hiperenlace </a>
```

## ■ Ejercicio 3 → Utilizaremos los pseudo-elementos.

### □ En estilo.css

```
p:first-letter{font-size: 18pt}  
h3:first-letter{font-size: 22pt}
```

# Introducir Comentarios

- Es conveniente añadir comentarios en la hoja de estilo, sobre todo si es extensa.
  - Así, es más fácil recordar qué es lo que hace cada parte cuando se vuelva a editar el código.
- Para añadir comentarios a las hojas de estilo, tanto internas como externas, se usan los caracteres `/* */`

`/*esto es un comentario*/`

# Presentación Con Estilos.

## Las Capas

- El modelo de capas suele utilizarse para realizar diseños vistosos de las páginas Web.
  - Son divisiones dentro del documento que dotan a éstas de un comportamiento relativamente independiente.
- Estas divisiones se obtienen mediante la inserción de elementos
  - Span → permite dividir el documento en pequeñas porciones.
  - Div → permite dividir el documento en grandes secciones.
- A cada división se le asigna un identificador para poder referirse a ella desde la hoja de estilos y así poder situarla en la página con sus características propias (color de fondo, borde, tipos de letra, etc.).

# Presentación Con Estilos.

## Span

- Fue introducido en HTML para permitir a los autores dar un estilo que no pudiera estar conectado a un elemento estructural HTML.
  - Puede usarse como un selector en una hoja de estilo.
  - O como un elemento.
    - Pero no tiene tal significado. Existe solamente para aplicar estilos, y por lo tanto, no tiene efecto cuando la hoja de estilo está desactivada.
- Acepta los atributos CLASS e ID.

```
<HEAD>
```

```
  <STYLE TYPE="text/css">
```

```
    .primeraspalabras { font-variant: small-caps };
```

```
  </STYLE>
```

```
</HEAD>
```

```
<BODY>
```

```
  <P> <SPAN CLASS=primeraspalabras>Las primeras palabras</SPAN> de un párrafo  
podrían estar en small-caps. El estilo también puede estar en línea, como para cambiar el  
estilo de una palabra como <SPAN STYLE="font-family: Arial;"> Arial</SPAN>
```

```
  </P>
```

```
</BODY>
```

# Presentación Con Estilos.

## Las Capas DIV

- Cosas como dónde colocar las imágenes, menús, etc. son fundamentales a la hora de crear un sitio Web.
  - Pero para ello primero debemos estructurar correctamente la información a nivel de código.
  - Esto se consigue usando etiquetas DIV y SPAN.
- La etiqueta DIV permite estructurar un documento HTML.
  - Se trata de un contenedor que incluye texto, imágenes, tablas, etc.
  - Lo que se encuentre entre las etiquetas <DIV> y </DIV> se considera una capa y puede posicionarse en el documento HTML mediante CSS.
- Dentro de la etiqueta DIV o SPAN **deberemos** colocar el atributo “id” ó “class”.
  - Colocaremos “id” si vamos a referenciar una parte única en todo el texto (encabezamientos, título principal, menú, etc.).
  - En los casos en los que queramos asociar un aspecto semejante a distintos elementos (que pueden o no repetirse) usaremos la etiqueta “class”.



# Presentación Con Estilos.

## Las Capas DIV

- La diferencia más evidente entre DIV y SPAN radica en el modo en que se incrustan los contenidos en la página que estamos creando.
  - Con **DIV** lograremos una división absoluta, algo así como una caja invisible que se extenderá tanto como pueda.
    - Por ello, en el ejemplo cada sección DIV llega hasta fin de línea y la siguiente capa comienza justo debajo.
    - Esto pasa con todas las etiquetas de tipo bloque.
  - **Span** se adaptará y permitirá dividir los contenidos sin saltar de línea forzosamente. Es una etiqueta de tipo inline.
    - Es útil para destacar porciones de texto, por ejemplo.

Uso de capas. ID y CLASS //tituloPagina	//contenedor
Web de ejemplo usando capas //descripcion	
Parrafo 1 //texto	
Parrafo 2 //texto	
Parrafo 3 //texto	

Ejemplo diferencia entre DIV y SPAN

# Presentación Con Estilos.

## Las Capas DIV

- Lo primero que se debe realizar es la planificación de la estructura de la página Web → lo haremos mediante un boceto.
- Después, creamos el código de la página Web.
  - Para ello, haremos uso de las etiquetas <DIV>.

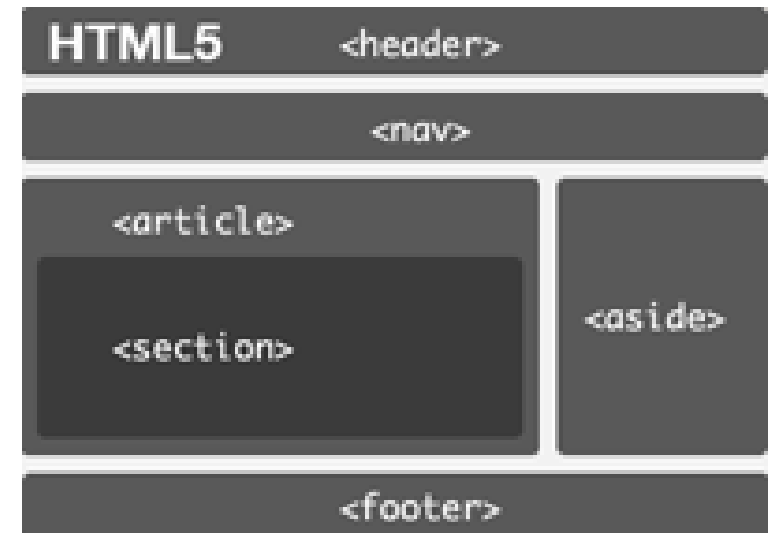
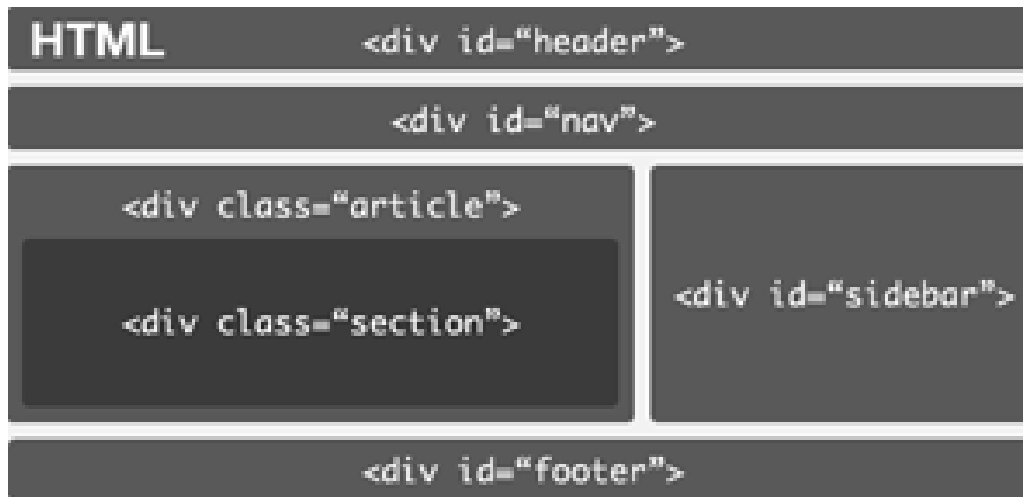
```
<HTML>
  <HEAD>
  </ HEAD>
  <BODY>
    <DIV id="cabecera"> <!-- Aquí vendría el código de la cabecera --> </DIV>
    <DIV id="índice"> <!-- Aquí vendría el código del índice --> </DIV>
    <DIV id="contenido"> <!-- Aquí vendría el código del cuerpo --> </DIV>
    <DIV id="foto1"> <!-- Aquí vendría el código de FOTO1 --> </DIV>
    <DIV id="foto2"> <!-- Aquí vendría código de FOTO2 --> </DIV >
  </ BODY>
</ HTML >
```

- Ahora añadiremos los contenidos para cada una de las secciones.
- Una vez completada la página Web, pasamos a definir la hoja de estilos.
  - En la hoja de estilos se definirá el posicionamiento de cada una de las secciones aquí definidas.

# ■ Presentación Con Estilos.

## Las Capas DIV

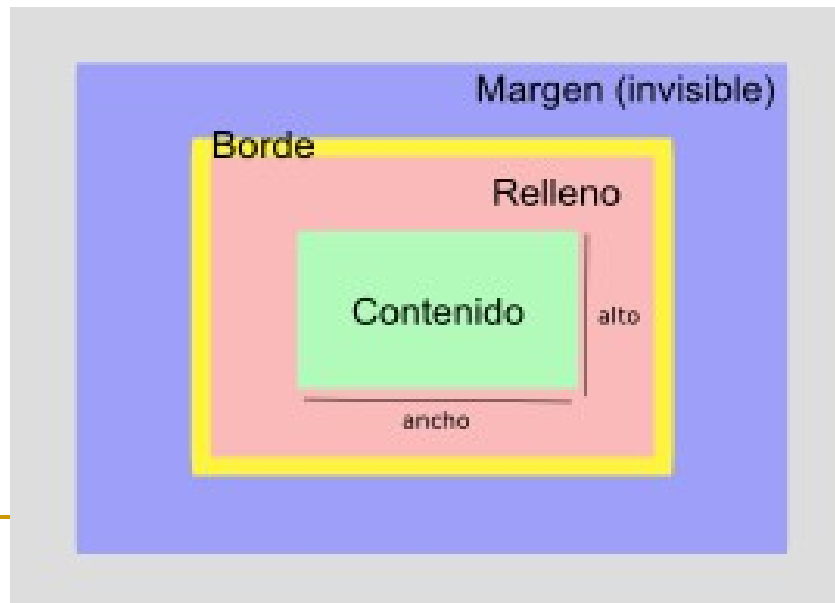
- La maquetación hasta ahora se hacía mediante capas con la etiqueta `<div>`. Ahora ya no es necesario con HTML5, porque se han incorporado una serie de etiquetas estructurales que permiten **definir las partes básicas de una página** y estructurar su contenido:



# Presentación Con Estilos.

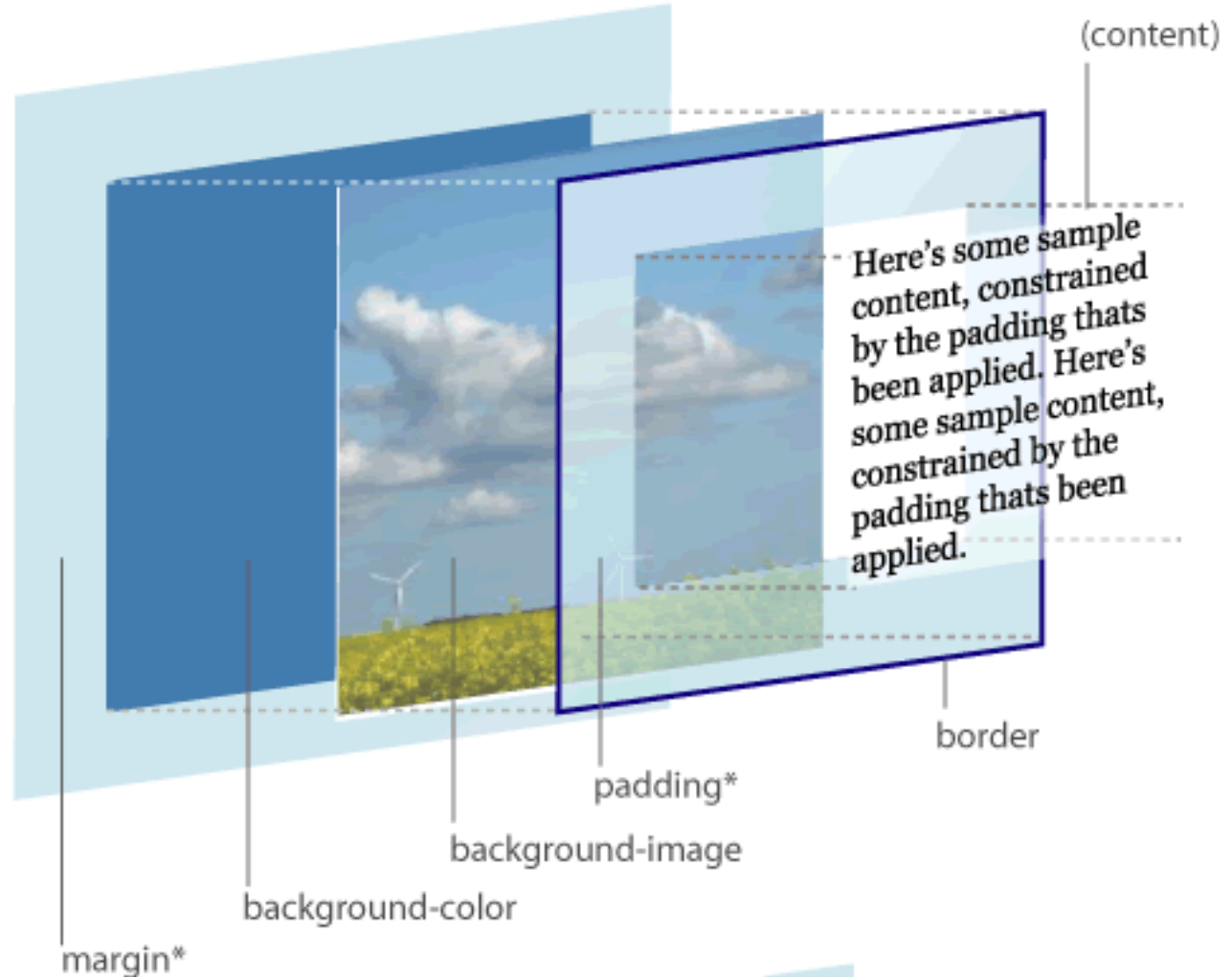
## Las Capas DIV

- La capa es un cuadro que se compone de:
  - Contenido
    - Texto, en el caso de un párrafo
    - Contenido de una celda, etc.
  - Alrededor del contenido hay un espacio de relleno (padding).
  - Alrededor del relleno hay un borde (border) que puede estar visible o no.
  - Y por último está el margen (margin), que es el espacio que se deja desde la caja a lo que le rodea (área exterior del bloque).
    - Nos sirve para dar espacio entre los elementos.



Partes de una capa

# THE CSS BOX MODEL HIERARCHY



\* transparent elements

# Presentación Con Estilos.

## Las Capas DIV

- El elemento *position* permite definir el comportamiento de la caja en el documento.
- Los valores que puede tomar son:
  - Static o normal
    - Las propiedades top, right, bottom y left no se aplican.
    - Esta es la opción por defecto.
  - Relative
    - La posición de la caja dependerá de los elementos HTML que tenga delante y detrás.
  - Absolute
    - La posición de la caja vendrá dada por las coordenadas que se definan, por lo que se podrá solapar con otros elementos de la página. La referencia es con respecto al siguiente elemento contenedor posicionado (relative, absolute o fixed).
  - Fixed
    - Este valor implica un posicionamiento absoluto (como el anterior) pero manteniendo la posición aunque se desplace la página. La referencia en las coordenadas es el navegador.
    - De esta manera, la caja estará **siempre visible**, por lo que se suele utilizar para los **índices de las páginas (menú)**.

# Presentación Con Estilos.

## Las Capas DIV

- Una vez determinada el comportamiento de la caja, se puede definir la posición de éstas en función de unas coordenadas.
  - Dichas coordenadas hacen referencia al elemento padre o a la ventana del navegador.
  - Los atributos son:
    - Top → Distancia de la caja respecto al límite superior de su elemento contenedor.
    - Right → Distancia de la caja respecto al límite derecho de su elemento contenedor.
    - Bottom → Distancia de la caja respecto al límite inferior de su elemento contenedor.
    - Left → Distancia de la caja respecto al límite izquierdo de su elemento contenedor.
- En el caso de que alguna caja se solape con otra, se puede determinar cuál estará delante utilizando el parámetro z-index.
  - Este parámetro toma valores enteros y se asigna un valor para cada caja.
  - La caja que tenga un valor mayor es la que aparecerá por delante de las demás.
- También se puede indicar la anchura (width) y la altura (height) de las cajas.

# Presentación Con Estilos.

## Las Capas DIV

### ■ Ejemplo:

```
#cabecera{  
    position: absolute;  
    top: 0px;  
    left: 0px;  
    width: 100%;  
    height: 60px; }
```

```
#indice {  
    position: absolute;  
    width: 200px;  
    left: 0px;  
    top: 64px; }
```

```
#foto1 {  
    position: absolute;  
    width: 200px;  
    left: 0px;  
    top: 200px; }
```

```
#cuerpo {  
    position: absolute;  
    left: 215px;  
    top: 65px;  
    padding: 15px; }
```



# Validación De Código

- Igual que en el caso de HTML, el organismo W3C dispone de una página donde comprobar que el código escrito en nuestra hoja de estilos se ajusta a los estándares definidos.
- La página es:

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

- Además, nos permite subir un archivo o indicar el código directamente.