

# Diseño y Programación de Páginas Web

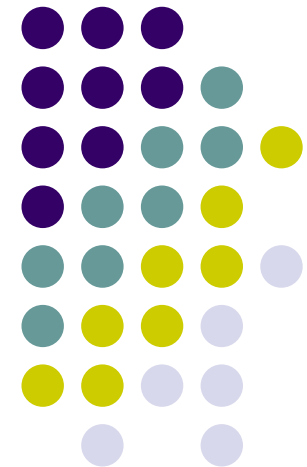
---

## 1.2. Hojas de Estilo CSS

Juan Ruiz de Miras ([demiras@ujaen.es](mailto:demiras@ujaen.es))

Departamento de Informática

Universidad de Jaén





# Contenidos

1. Introducción
2. Sintaxis de las hojas de estilo
3. Uso de hojas de estilo
4. Especificación de reglas CSS
5. Cascada y herencia
6. Propiedades CSS



# 1. Introducción

- Evolución HTML:
  - HTML (1994): fácil, estructurado y muy pocas etiquetas
  - Navegadores ganan en visualización gráfica y HTML 4 (1997) añade muchas etiquetas de formato (<font>, ...) y deja de ser un lenguaje orientado a la estructura
- Objetivo de las hojas de estilo:
  - Separar el contenido de las páginas web de su apariencia
    - **contenidos**: se especifican en ficheros HTML
    - **formato**: se especifica en ficheros CSS
  - CSS 1 (1996, 1999): fuentes, márgenes, colores, ...
  - CSS 2 (1998, 2005): posición absoluta, numeración automática, ...
  - CSS 3 (en desarrollo): interacción usuario, ...
  - <http://www.w3.org/Style/CSS/>



# 1. Introducción

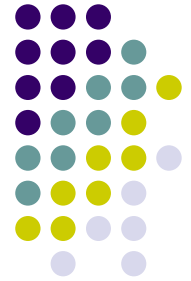
- Principales ventajas y características:
  - Permite edición más fácil y rápida:

```
<h1 align="center">Apartado 1</h1>
```

Texto del primer apartado

```
<h1 align="center">Apartado 2</h1>
```

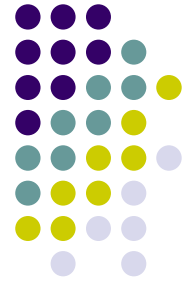
Texto del segundo apartado
  - Mayor posibilidad de formato que HTML
  - Reutilización de estilos
  - Aplicación y combinación en cascada
  - Reduce el tamaño del código a enviar
- Inconvenientes:
  - Tratamiento desigual por los distintos navegadores



# Contenidos

1. Introducción
2. **Sintaxis de las hojas de estilo**
3. Uso de hojas de estilo
4. Especificación de reglas CSS
5. Cascada y herencia
6. Propiedades CSS

## 2. Sintaxis de las hojas de estilo



- Los estilos se especifican utilizando **reglas**
- Cada regla consta de:
  - **selector**: identifica los elementos de la página web a los que se les aplicará el estilo
  - **declaraciones**: especifican el estilo a aplicar, cada declaración consiste en un par:
    - propiedad CSS a aplicar
    - valor para la propiedad

• Ejemplo: `h1 {text-align: center;}`

Diagram illustrating the components of the CSS rule:

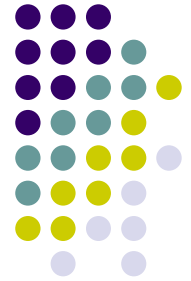
- selector**: `h1`
- declaraciones**: `{text-align: center;}`
- regla**: The entire rule `h1 {text-align: center;}`
- Inside the declarations:
  - propiedad CSS**: `text-align`
  - valor**: `center`



# Contenidos

1. Introducción
2. Sintaxis de las hojas de estilo
- 3. Uso de hojas de estilo**
4. Especificación de reglas CSS
5. Cascada y herencia
6. Propiedades CSS

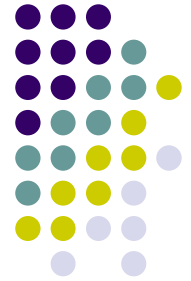
# 3. Uso de hojas de estilo



- En línea
- Incrustación
- Vinculación
- Varias hojas de estilo



## 3.1 En línea



- Se utiliza la propiedad **STYLE** de la etiqueta HTML
- El valor de la propiedad STYLE es la especificación del estilo a aplicar (declaraciones de la regla)
- Ejemplo:

```
<h1 style="text-align:center">Apartado 1</h1>
```
- Características:
  - No es la forma más práctica de aplicar estilos
  - Da más posibilidades de formato que utilizando sólo HTML
  - Tiene preferencia sobre el formato aplicado con HTML

```
<h1 align="right" style="text-align:center">
  Apartado 1
</h1>
```

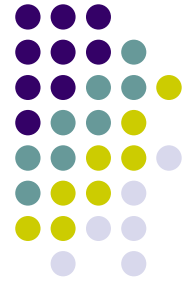
## 3.2. Incrustación



- Se utiliza la etiqueta **STYLE** en la cabecera de la página web
- Entre `<style type="text/css">` y `</style>` se especifican las reglas
- Ejemplo:

```
<html>
  <head>
    <style type="text/css">
      h1 {text-align:center;}
    </style>
  </head>
  <body>
    <h1>Apartado 1</h1>
    Texto del primer apartado
  </body>
</html>
```

## 3.2. Incrustación



- La hoja de estilos sólo afecta a la página donde se ha incrustado
- Interesante si en el web hay unas pocas páginas con un formato distinto al resto
- Para evitar problemas en navegadores que no soportan CSS (IE antes de la versión 3 y Netscape antes de la versión 4) se pueden poner las reglas entre comentarios html:

```
<head>
  <style type="text/css">
    <!--
      h1 {text-align:center;}
    -->
  </style>
</head>
```

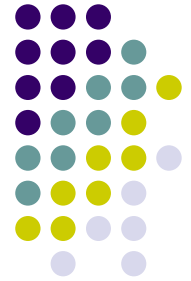


## 3.3 Vinculación

- Separa realmente los contenidos (fichero .html) de su formato (fichero .css)
- El fichero de texto .css contiene las reglas de estilo
- En la cabecera del fichero .html se establece la vinculación con la hoja de estilos del fichero .css utilizando la etiqueta:

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

- Es la forma más flexible de aplicar las hojas de estilo:
  - Permite cambiar el estilo sin tocar el código html
  - La misma hoja de estilo puede ser compartida por varias páginas



## 3.3 Vinculación

- Fichero "**estilo.css**":

```
h1 {text-align: center;}  
h2 {text-indent: 20pt;}
```

- Fichero html:

```
<html>  
  <head>  
    <link rel="stylesheet" href="estilo.css" type="text/css">  
  </head>  
  <body>  
    <h1>Apartado 1</h1>  
    Texto del primer apartado  
    <h2>SubApartado 1.1</h1>  
    Texto del subapartado  
  </body>  
</html>
```



## 3.4 Varias hojas de estilo

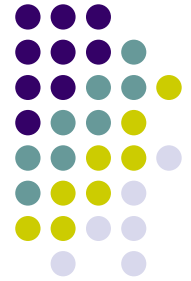
- Combinar vinculación, incrustación y estilos en línea:
  - vinculación: para los estilos comunes a todo el web
  - incrustación: para personalizar varias páginas
  - estilos en línea: para detalles concretos



## 3.4 Varias hojas de estilo

- Vincular varias hojas en función del destino
  - se utiliza la propiedad **MEDIA** de la etiqueta LINK, posibles valores:
    - screen: visualización en el monitor, valor por defecto
    - print: cuando la página se imprime
    - projection: proyección
    - braille: dispositivos braille
    - speech: dispositivos de habla
    - all: para todos los dispositivos
  - Muy útil para diferenciar vista en pantalla y vista impresora:

```
<head>
  <link rel="stylesheet" href="pantalla.css"
        type="text/css" media="screen">
  <link rel="stylesheet" href="impresora.css"
        type="text/css" media="print">
</head>
```



## 3.4 Varias hojas de estilo

- Vincular varias hojas y la selección la hace el usuario
  - se utiliza la propiedad **NAME** de la etiqueta LINK para dar nombre a las hojas de estilo que se cargan por defecto pero que se pueden desactivar
  - se da el valor "alternate stylesheet" a la propiedad REL para especificar hojas de estilo alternativas que el usuario podrá activar (también hay que especificar la propiedad NAME) pero que no se cargan por defecto
  - **NO está soportado por los navegadores actuales:** debería aparecer un menú donde se permitiera activar/desactivar las distintas hojas de estilo:
    - IE permite cargar una hoja de estilo propia del usuario
    - Mozilla permite visualizar la página eliminando todas las hojas de estilos



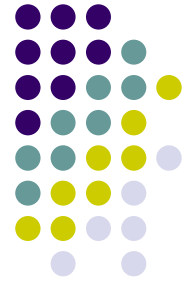


## 3.4 Varias hojas de estilo

- Vincular varias hojas y la selección la hace el usuario

```
<head>  
  <link rel="stylesheet" href="hoja1.css">  
  <link rel="stylesheet" href="hoja2.css" name="Por defecto">  
  <link rel="alternate stylesheet" href="hoja3.css" name="Alternativa">  
</head>
```

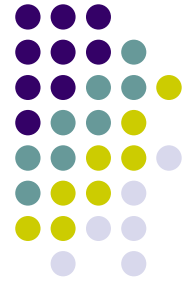
- hoja1.css se aplicará siempre y no es posible desactivarla
- hoja2.css se aplicará inicialmente y es posible desactivarla
- hoja3.css no se aplicará inicialmente pero es posible activarla



# Contenidos

1. Introducción
2. Sintaxis de las hojas de estilo
3. Uso de hojas de estilo
- 4. Especificación de reglas CSS**
5. Cascada y herencia
6. Propiedades CSS

# Especificación de reglas CSS



- Agrupación de selectores y declaraciones
- Especificación de selectores
  - Selectores de clase
  - Selectores "ID"
  - Selectores de contexto
  - Selectores de propiedades
  - Seudo-clases
  - Seudo-elementos

## 4.1. Agrupación de selectores y declaraciones



- Agrupación de selectores
  - Los selectores que comparten las mismas declaraciones se pueden poner juntos en la misma regla, separados por comas:

H1, H2 {color:blue;} equivale a:

```
H1 {color:blue;}  
H2 {color:blue;}
```

## 4.1. Agrupación de selectores y declaraciones



- Agrupación de declaraciones
  - Las declaraciones que afectan al mismo selector se pueden poner juntas en la misma regla, separadas por puntos y comas:

H1 {color:blue; text-align:center;} equivale a:

```
H1 {color:blue;}
```

```
H1 {text-align:center;}
```

## 4.2. Especificación de selectores

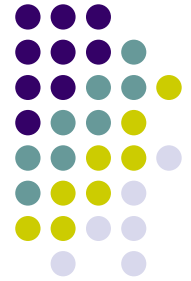


- Especificación de selectores es variada y muy flexible
- Permite sacar el máximo partido a las hojas de estilo
- Los editores CSS no suelen aprovechar al máximo estas capacidades
- Selector universal: `*`
  - `* {font-family:arial}`
- Selectores de tipo
  - Corresponden a las etiquetas HTML
  - Cambiando la hoja de estilo cambia el aspecto de la etiqueta html en todos los sitios que aparezca y en todas las páginas vinculadas
  - Problema: cambiar el estilo en distintos sitios para la misma etiqueta

## 4.2.1. Selectores de clase



- Selectores de clase:
  - Permiten crear clases de estilos para aplicar a cualquier elemento o como particularización de elementos concretos
  - Especificación:
    - Particularizar elemento:
      - `elemento.id_clase`
      - Ejemplo: `h1.centrado {text-align:center;}`
    - Clase genérica:
      - `.id_clase`
      - Ejemplo: `.resaltado {color:red;}`



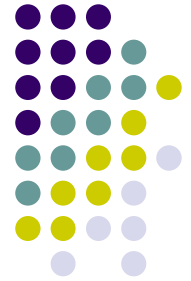
## 4.2.1. Selectores de clase

- Selectores de clase:
  - Utilización:
    - `<etiqueta class="id_clase1 id_clase2 ...">`
    - Ejemplo:

```
h1.centrado {text-align:center;}  
.resaltar {color:red;}  
  
<h1 class="centrado">Apartado 1</h1>  
Texto del apartado 1  
<p class="resaltar">Párrafo importante</p>  
<p class="centrado">Párrafo centrado</p>
```
  - Implica modificar el código html, pero
  - Proporciona flexibilidad para crear estilos particularizados



## 4.2.2. Selectores “ID”



- Selectores "ID":
  - La propiedad "ID" de una etiqueta HTML permite identificar ese elemento de manera única dentro de la página
  - El selector "ID" permite definir un estilo para aplicar a ese único elemento de la página
  - Los navegadores NO controlan que el identificador sea único en la página
- Especificación:
  - `elemento#identificador`
  - Ejemplo: `p#primero {text-align:center;}`
- Utilización:
  - `<etiqueta id="identificador">`
  - Ejemplo: `<p id="primero">El primer párrafo</p>`

## 4.2.3. Selectores de contexto



- Selectores de contexto:

- Permiten diferenciar estilos para un elemento en función de su situación respecto a otros elementos
- Tipos de situación entre elementos:

```
<table>
  <ul>
    <li> ... </li>
  </ul>
</table>
<p> ... </p>
```

- Descendientes: <ul> desciende de <table>, <li> de <table>
- Hijos: <ul> es hijo de <table>, <li> de <ul>
- Hermanos adyacentes: <p> es hermano de <table>  
<table> **NO** es hermano de <p>

## 4.2.3. Selectores de contexto



- Selectores de contexto

- Especificación:

- Descendientes:

- elementos separados por espacios en blanco
      - Ejemplo: `table ul li {color:red;}`

- Hijos:

- padre ">" hijo
      - Ejemplo: `table > ul {color:red;}`

- Hermanos:

- elemento + elemento
      - Ejemplo: `table + p {color:red;}`

No funciona en I. Explorer

Sí funciona en Mozilla

## 4.2.4. Selectores de propiedades



- Selectores de propiedades: (No funciona en I. Explorer)
  - Permiten especificar estilos a elementos de la página en función de sus propiedades o de valores que tengan asignadas
  - Posibles especificaciones:
    - elemento[**propiedad**] { ... }
      - se ha definido la propiedad del elemento
    - elemento[**propiedad="valor"**] { ... }
      - se ha asignado ese valor a la propiedad del elemento
    - elemento[**propiedad~="valor"**] { ... }
      - la propiedad del elemento tiene incluido ese valor separado por espacios
    - elemento[**propiedad|="valor"**] { .. }
      - la propiedad del elemento tiene incluido ese valor separado por guiones
    - elemento[...][...].... { ... }
      - es posible especificar varias propiedades

## 4.2.4. Selectores de propiedades



- Selectores de propiedades:

```
p[align] {color:red;}
```

```
p[align="center"] {color:blue;}
```

```
img[alt~="logo"] {border:solid red}
```

```
img[alt|="logo"] {border:solid blue}
```

```
<p align="center"> párrafo centrado </p>
```

```
<p align="right"> párrafo a la derecha </p>
```

```
<p> párrafo normal </p>
```

```

```

```
<br>
```

## 4.2.5. Seudo-clases



- Seudo-clases

- Permiten especificar estados de elementos de la página web que no se reflejan en el código HTML

- Especificación:

- `selector:seudo-clase { ... }`

- seudo-clases:

- `link`: enlace no visitado

- `visited`: enlace visitado

No en todos  
elementos

I. Expl.

Sí en Moz.

- `hover`: cuando el cursor pasa por encima del elemento

- `active`: cuando se activa el elemento (ej: mientras se pulsa el ratón)

- `focus`: cuando el elemento recibe el foco

No en I. Expl.

Sí en Moz.

- `first-child`: identifica al primer hijo en una relación padre - hijo

- `lang(id_lenguaje)`: cuando el lenguaje del elemento es `id_lenguaje`

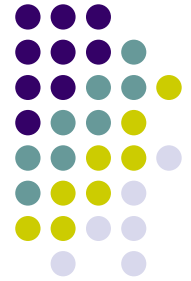
## 4.2.5. Pseudo-classes



- Pseudo-classes

```
a:link {color:red;}
a:visited {color:yellow;}
a:hover {color:green;}
a:active {color:blue;}
input:focus {border:2px solid;}
input:hover {color:red;}
ul > li:first-child {color:red;}
p:lang(en) {color:red;}
```

```
<a href="fichero.html" target="_blank">Enlace</a><br>
<input></input><br>
<ul>
  <li>primer hijo</li>
  <li>segundo hijo</li>
</ul>
<p lang="es">en español</p>
<p lang="en">en inglés</p>
```



## 4.2.6. Seudo-elementos

- Seudo-elementos

- Permiten especificar estilos a ciertas partes relevantes de elementos de la página web

- Especificación:

- `selector:seudo-elemento { ... }`

- seudo-elementos:

- `first-line`: identifica la primera línea

- `first-letter`: identifica la primera letra

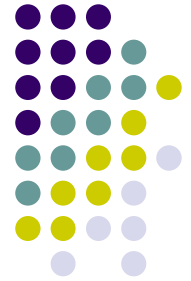
No en I. Expl  
Sí en Moz.

- `before {content:"contenido"; ...}`: inserta contenido, con sus propios estilos, antes del elemento especificado

- `after {content:"contenido"; ...}`: inserta contenido, con sus propios estilos, después del elemento especificado



## 4.2.6. Seudo-elementos



- Seudo-elementos

```
h1:first-letter {font-size:200%;}
```

```
p:first-line {color:red;}
```

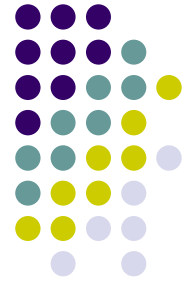
```
cite:before {content:"Cita: ";text-decoration:italic;}
```

```
<h1>una cabecera</h1>
```

```
<p>La primera línea en rojo<br>
```

```
a partir de aquí en el color por defecto</p>
```

```
<cite>Esto es una cita</cite>
```



# Contenidos

1. Introducción
2. Sintaxis de las hojas de estilo
3. Uso de hojas de estilo
4. Especificación de reglas CSS
- 5. Cascada y herencia**
6. Propiedades CSS

# 5. Cascada y Herencia



- Una misma página web puede utilizar varias hojas de estilo
  - Útil, por ejemplo, para tener estilos comunes a varias páginas y otros estilos personalizados dependiendo de la página
  - Combinar vinculación, incrustación y estilos en línea
  - Vincular varias hojas
  - Importación de hojas de estilo:
    - `@import "fichero.css";` ó `@import url("url_fichero.css");`
    - la importación se hace desde una hoja de estilo y antes de cualquier regla
- Los estilos de cada hoja, antes de aplicarse a la página, se combinan en un proceso denominado Cascada:
  - este proceso resuelve los conflictos que aparecen cuando varias reglas afectan al mismo elemento

# 5. Cascada y Herencia



- Orden de cascada (de mayor a menor preferencia):
  1. Procedencia de la hoja de estilos:
    1. Hoja de estilo del autor de la página web
    2. Hoja de estilo del usuario
      - los navegadores permiten cargar hojas de estilo propias del usuario
        - IE: Opciones de internet → General → Accesibilidad
        - Mozilla: Herramientas → Preferencias → Tipografía y colores
      - Reglas !important:
        - prevalecen sobre las reglas del autor al mismo nivel
        - selector { ... !important }
    3. Estilos por defecto del navegador
  2. Especificidad
  3. Orden

# 5. Cascada y Herencia



- Orden de cascada (de mayor a menor preferencia):
  1. Procedencia de la hoja de estilos:
  2. Especificidad:
    - Cálculo de especificidad:
      - A. número de atributos ID (#) en el selector
      - B. número de especificación de clases (.) en el selector
      - C. número de etiquetas en el selector
    - la especificidad la da el número ABC
    - en caso de conflicto prevalece la regla con mayor especificidad
  3. Orden
    - En caso de conflicto y a igualdad de especificidad, prevalece la última regla especificada
    - vinculación anterior a incrustación anterior a estilos en línea

# 5. Cascada y Herencia



- Ejemplos Orden de cascada:

<code>li {color:red}</code>	→ A=0 B=0 C=1: 001
<code>li.primerio {color:blue}</code>	→ A=0 B=1 C=1: 011
<code>ul li.primerio {color:green}</code>	→ A=0 B=1 C=2: 012
<code>#primerio {color:yellow}</code>	→ A=1 B=0 C=0: 100
<code>#primerio {color:magenta}</code>	→ A=1 B=0 C=0: 100

```
<ul>
  <li class="primerio" id="primerio">primerio</li>
  <li class="primerio">segundo</li>
  <li>tercero</li>
</ul>
```

# 5. Cascada y Herencia



- Herencia:
  - Un elemento de una página hereda el estilo del padre (elemento que lo contiene)
    - `html {color:red;}`, esta regla afecta a todos los elementos
  - No todas las propiedades son heredables
    - márgenes, bordes, ...
  - Al asignar un estilo a un elemento se sustituye el que hubiera heredado

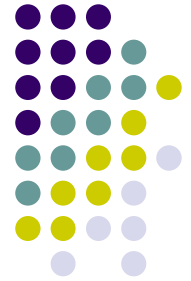


# Contenidos

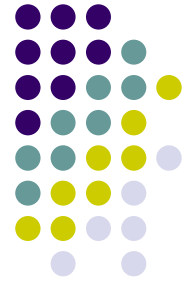
1. Introducción
2. Sintaxis de las hojas de estilo
3. Uso de hojas de estilo
4. Especificación de reglas CSS
5. Cascada y herencia
6. **Propiedades CSS**



# 6. Propiedades CSS



- Clasificación visual de los elementos HTML
- Etiquetas SPAN y DIV
- Unidades
- Principales propiedades CSS



## 6.1. Clasificación visual de los elementos HTML

- Elementos de bloque
  - Visualmente se tratan como un bloque separado de los elementos adyacentes
  - Fuerzan un salto de línea al inicio y al final del elemento
  - Ejemplos: `<p>`, `<h1>`, ...
- Elementos en línea
  - Visualmente se no se tratan como bloque y por tanto sus contenidos se sitúan a nivel de línea
  - No fuerzan salto de línea
  - Ejemplos: `<a>`, `<em>`, ...
- Elementos de lista
  - Elementos de bloque que contienen un elemento en línea asociados a viñetas o un carácter de orden
  - Fuerzan salto de línea al inicio y final del elemento
  - Ejemplo: `<li>`



## 6.1. Clasificación visual de los elementos HTML

- Cambio del tipo de visualización:
  - Propiedad CSS `DISPLAY`
  - No se hereda
  - Valores posibles: `block`, `inline`, `list-item` y `none`
  - Ejemplo:

```
.bloque {display:block}
.enlinea {display:inline;}
.lista {display:list-item;}
.ninguno {display:none;}
```

Antes del párrafo...

```
<p class="enlinea">párrafo en línea</p>
<ul> <p class="lista">párrafo lista</p> </ul>
<p class="ninguno">párrafo sin visualización</p>
<a class="bloque">enlace formato bloque</a>
```

Después de enlace bloque...

## 6.2. Etiquetas SPAN y DIV



- Las etiquetas SPAN y DIV permiten aplicar estilos a partes de la página independientemente de las etiquetas HTML
- Las propiedades que admiten estas etiquetas son STYLE, CLASS e ID
- SPAN es un elemento en línea
- DIV es un elemento de bloque
  - Puede contener cualquier otro elemento HTML
  - Característica fundamental que permite definir divisiones lógicas o capas (layers) en la página HTML y formatearlas con hojas de estilos

## 6.2. Etiquetas SPAN y DIV

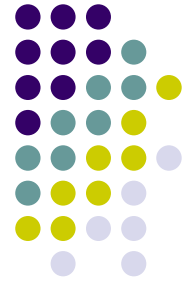


```
.resaltar {color:red;}  
div#menu {width:120px; background-color:green}  
div#menu a {display:block;}
```

```
<h1>Esto es una <span class="resaltar"> cabecera </span></h1>
```

```
<div id="menu">  
  <a>Inicio</a>  
  <a>Opción 1</a>  
  <a>Opción 2</a>  
</div>
```

## 6.3. Unidades



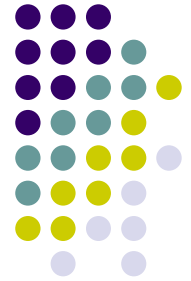
- Longitud:
  - Especificación:
    - número unidad
    - todo seguido, sin espacios en blanco
  - Absolutas:
    - in (pulgadas), cm, mm, pt (puntos), pc (picas)
  - Relativas:
    - em (altura fuente), ex (altura letra x), px (pixel)
  - Ejemplo:
    - `<span style="font-size:2em">`

## 6.3. Unidades



- Porcentaje:
  - Especificación:
    - número %
    - todo seguido, sin espacios en blanco
  - Ejemplo:
    - `<div style="width:50%;background-color:red">`
- Colores:
  - palabra clave: red, blue, green, grey, ...
  - #rrggbb: rr, gg y bb en hexadecimal
  - `rgb(x,x,x)`: x entre 0 y 255
  - `rgb(x%,x%,x%)`: x entre 0 y 100
  - Ejemplo:
    - #00cc00, `rgb(0,204,0)` y `rgb(0%,80%,0%)` es el mismo color

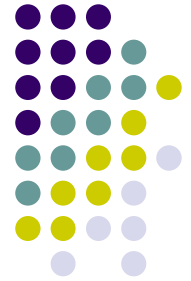
## 6.4. Principales propiedades CSS



- Formato de texto
- Fuente
- Listas
- Borde, margen y relleno
- Colores y fondo
- Posición y dimensiones
- Descripción detallada de las propiedades CSS:
  - <http://www.w3.org/TR/CSS21/propidx.html>
  - <http://www.htmlhelp.com/es/reference/css/properties.html>



## 6.4.1. Formato de texto



- Formato de texto:
  - Alineación horizontal: `text-align`
    - se aplica a elementos de bloque
    - valores posibles: `left`, `right`, `center`, `justify`
  - Alineación vertical: `vertical-align`
    - se aplica a elementos en línea
    - valores posibles: `baseline`, `top`, `middle`, `bottom` o porcentaje
    - útil para texto en tablas o alinear imágenes:

```
texto 
```
  - Sangría: `text-indent`
    - se aplica a elementos de bloque
    - valores posibles: medida o porcentaje
    - establece sangría de la primera línea (para todas utilizar `margin`)

## 6.4.1. Formato de texto

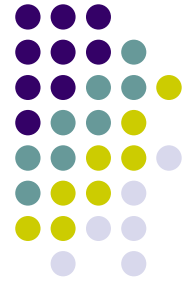


- Formato de texto:
  - Decoración del texto: `text-decoration`
    - valores posibles: `none`, `underline` (subrayado), `overline` (suprarrayado), `line-through` (tachado), `blink` (parpadeo)
    - ejemplo:

```
<a href="f.html" style="text-decoration:none">
  Enlace sin subrayar
</a>
```
  - Transformación del texto: `text-transform`
    - valores posibles: `none`, `capitalize` (primera letra en mayúsculas), `uppercase` (mayúsculas), `lowercase` (minúsculas)
    - ejemplo:

```
<span style="text-transform:capitalize">
  cada primera letra en mayúsculas
</span>
```

## 6.4.2. Fuente



- Fuente:
  - Familia de fuente: `font-family`
    - valores posibles: nombres de fuentes o de familias genéricas separadas por comas
    - familias genéricas: `serif`, `sans-serif`, `cursive`, `fantasy`, `monospace`
    - útil poner alguna fuente genérica por si las específicas no están instaladas
    - ejemplo:

```
<p style="font-family:arial,fantasy;">
  Si hay Arial, si no familia fantasy
</p>
```

## 6.4.2. Fuente



- Fuente:
  - Fuentes descargables:
    - problemas si fuentes seleccionadas no están disponibles en el cliente
    - es posible descargar la fuente junto con la página web
    - requiere que el servidor web reconozca los ficheros **.PFR**:
      - formato propietario de Netscape
      - mapa de bits con los caracteres que se utilizan → tamaño reducido
      - IE lo detecta como un control ActiveX
    - sólo se puede ver si la página está en un servidor

```
<link rel="fontdef" src="url_fichero.PFR">
```

```
<p style="font-family:nombre_fuente;">Ejemplo fuente</p>
```

## 6.4.2. Fuente



- Fuente:
  - Tamaño de la fuente: `font-size`
    - valores posibles:
      - valor absoluto: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`
      - valor relativo: `larger`, `smaller`
      - longitud o porcentaje (respecto al elemento padre)
    - ejemplo:

```
<body>
  <h1 style="font-size:2em;">
    Texto de tamaño 2 veces el de la fuente de <body>
  </h1>
</body>
```

## 6.4.2. Fuente



- Fuente:
  - Peso de la fuente: `font-weight`
    - valores posibles:
      - relativos: `normal`, `bolder`, `lighter`
      - absolutos: `bold`, `100`, `200`, ..., `900`
  - Estilo de la fuente: `font-style`
    - valores posibles: `normal`, `italic`, `oblique`
    - ejemplo:

```
<p style="font:bold italic;">
  Texto en negrita y cursiva
</p>
```

## 6.4.3. Listas



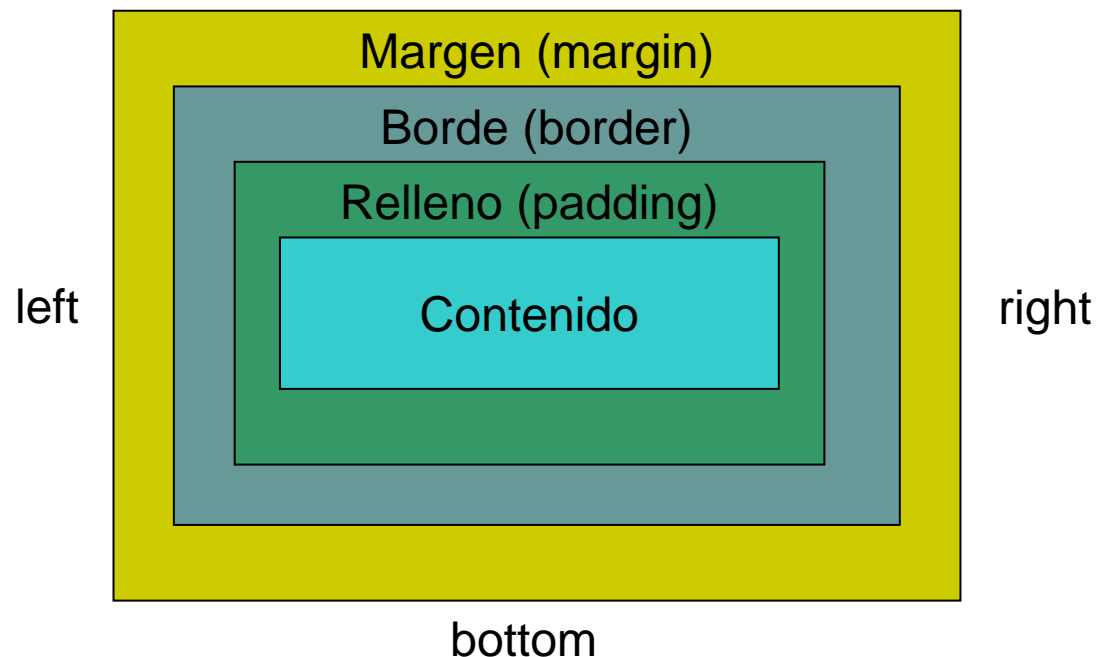
- Listas:
  - Tipo de lista: `list-style-type`
    - valores posibles: `disc`, `circle`, `square`, `decimal`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`, `none`
  - Imagen de la lista: `list-style-image`
    - valores posibles: `none`, `url(url_imagen)`
  - Posición de la lista: `list-style-position`
    - valores posibles: `outside` (sangría francesa), `inside`
    - ejemplo:

```
ol {list-style:upper-roman inside;}
ol ol {list-style:lower-alpha outside}
ul {list-style:url("ugr.gif")}
```

## 6.4.4. Borde, margen y relleno



- Borde, margen y relleno:
  - El aspecto visual de CSS se basa en un modelo de cajas
  - Este modelo permite aplicar todas las propiedades CSS a cualquier elemento de la página (bloque o en línea): imagen de fondo, relleno, posición en la pantalla, ...

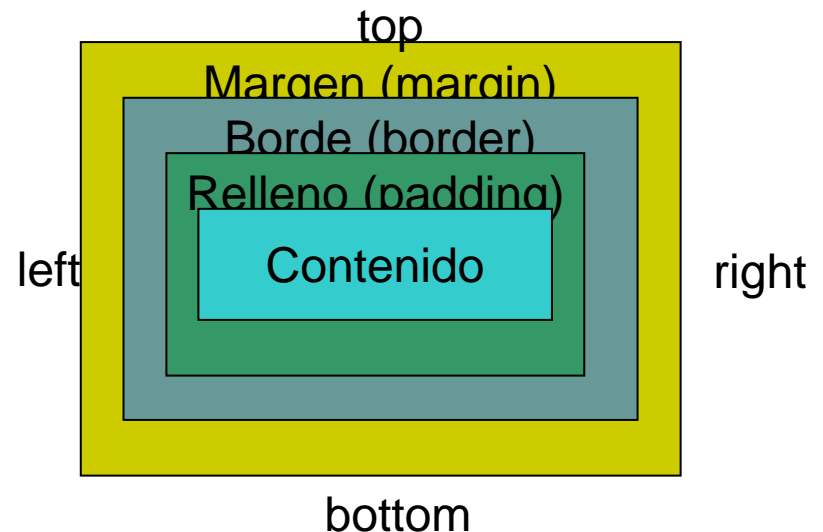




## 6.4.4. Borde, margen y relleno



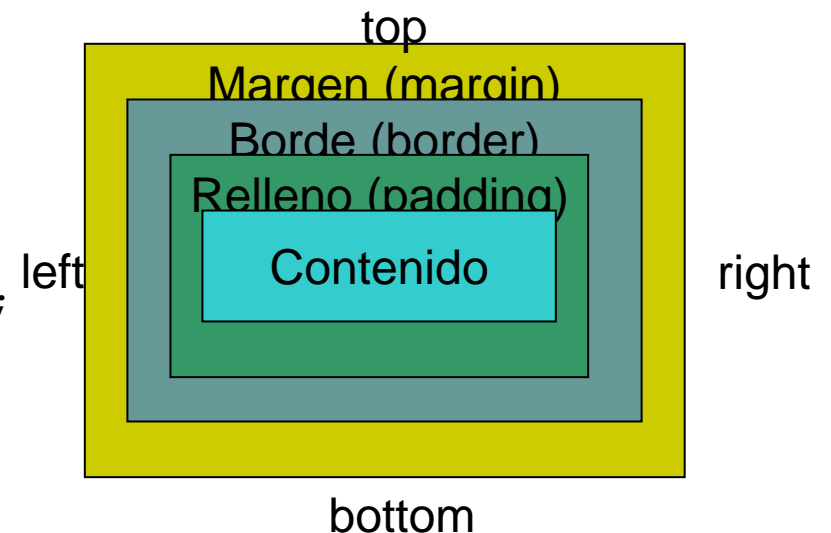
- Borde, margen y relleno:
  - Distinguiendo entre top, left, right o bottom:
    - `border-right`, `padding-left`, ...
      - `padding-left:10px;`
      - `border-top-color:blue;`
  - Tratamiento uniforme: `border`, `margin` o `padding`
    - Mismo valor:
      - `border-style:solid;`
    - Valores distintos:
      - 1 a 4 valores posibles
        - 1: top, bottom, left y right
        - 2: (top, bottom), (left, right)
        - 3: (top), (left, right), (bottom)
        - 4: (top), (right), (bottom), (left)



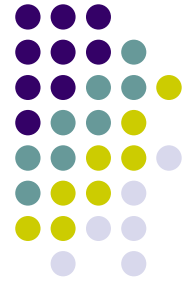
## 6.4.4. Borde, margen y relleno



- Borde, margen y relleno:
  - Características configurables:
    - grosor: medida o porcentaje
      - `padding: 1%;`
    - color: sólo para borde (margen es transparente y relleno tiene el color del fondo del contenido)
      - `border-color: red;`
      - `border-bottom-color: blue;`
    - estilo: sólo para borde
      - `none, solid, dotted, inset, outset, ...`
      - `border-style: inset dotted;`
      - `border-top-style: none;`



## 6.4.5. Colores y fondo



- Colores y fondo:
  - A diferencia de HTML, en CSS todos los elementos pueden tener color de primer plano y color e imagen de fondo
  - Para la imagen de fondo se puede especificar su posición y cómo debe repetirse
- Color de primer plano:
  - propiedad `color`:

```
#resaltar {color:red;}
```

```
<span id="resaltar">texto importante</span>
```

## 6.4.5. Colores y fondo



- Colores y fondo:
  - Propiedades del fondo (background):
    - color: `background-color`
      - valor de color o `transparent`
    - imagen de fondo: `background-image`
      - `none` o `url( "imagen" )`
    - repetición de la imagen: `background-repeat`
      - `repeat`: tipo mosaico
      - `repeat-x`, `repeat-y` o `no-repeat`

## 6.4.5. Colores y fondo



- Colores y fondo:
  - Propiedades del fondo (background):
    - posición de la imagen: `background-position`
      - uno (horizontal) o dos (horizontal vertical) valores de porcentaje o de medida, respecto de la esquina izquierda-superior
        - 0% 0%: esquina izquierda-superior de la imagen alineada con la esquina izquierda-superior de la caja
        - 30px 20px
      - [{top, center, bottom}] [{left, center, right}]
        - bottom → 50% 100%
        - right → 100% 50%
        - left center → 0% 50%
    - desplazamiento de la imagen: `background-attachment`
      - desplazamiento respecto al documento
      - `fixed` o `scroll`

## 6.4.5. Colores y fondo



- Colores y fondo:

- Propiedades del fondo (background):

```
#imagen {  
    padding:50%;  
    background-color:yellow;  
  
    background-image:url("ugr.gif");  
    background-repeat: no-repeat;  
    background-position: center center;  
  
    background-attachment: fixed;  
}
```

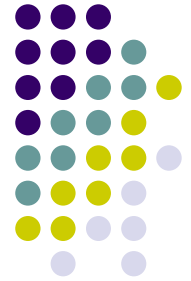
```
<div id="imagen"> Imagen de fondo </div>
```

## 6.4.6. Posición y dimensiones



- Posición y dimensiones:
  - Estas propiedades CSS permiten formatear la página web al estilo de los programas de maquetación
  - Posicionamiento de los elementos:
    - Se establece el tipo y los valores de posicionamiento
    - Tipo de posicionamiento: `position`
      - static: la de HTML, no es posible posicionar el elemento
      - `relative`: el elemento se posiciona respecto a su posición normal en la página, el elemento siguiente se sitúa respecto a la posición original (sin posicionar) del elemento anterior
      - `absolute`: el elemento se posiciona libremente en la página, el elemento se ignora para posicionar el elemento siguiente
      - `fixed`: igual que `absolute` salvo que al hacer scroll no se mueve junto con el contenido de la página. (Internet Explorer considera `fixed` igual que `static`)

## 6.4.6. Posición y dimensiones



- Posición y dimensiones:
  - Posicionamiento de los elementos:
    - Valores de posicionamiento: `top`, `left`, `right`, `bottom`
    - Se especifican en medida o porcentaje
    - Valores respecto a los márgenes superior, izquierdo, derecho o inferior de:
      - la caja que contiene al elemento: si posición es `absolute` o `fixed`
      - la posición original (sin posicionar) de la caja del propio elemento: si posición es `relative`
    - `top` y `left` tienen prioridad sobre `right` y `bottom`



## 6.4.6. Posición y dimensiones



- Posición y dimensiones:
  - Posicionamiento de los elementos:

```
#posicion_absoluta {  
  position: absolute;  
  bottom: 100px;  
  right: 100px; }  
#posicion_relativa {  
  position: relative;  
  top: 75px;  
  left: 75px; }
```

```
<div> Caja normal </div>  
<div id="posicion_absoluta"> Caja posición Absoluta </div>  
<div id="posicion_relativa"> Caja posición Relativa </div>
```

## 6.4.6. Posición y dimensiones



- Posición y dimensiones:
  - Dimensiones: `width`, `height`
  - Se especifica en medida o porcentaje
  - Se aplican a los elementos de bloque

```
#posicion_absoluta {  
    position: absolute;  
    bottom: 100px;  
    right: 100px;  
  
    background-color: green;  
  
    width: 50%;  
    height: 100px; }
```

```
<div id="posicion_absoluta"> Caja posición Absoluta </div>
```

## 6.4.6. Posición y dimensiones



- Posición y dimensiones:
  - Capas: propiedad `z-index:nº_entero`
    - Los elementos posicionados con `position` pueden solaparse
    - Por defecto, el navegador los muestra según aparecen en el código
    - z-index permite especificar la profundidad en el solapamiento
    - a mayor z-index más arriba en el solapamiento
    - esta propiedad se hereda:
      - elemento A: z-index vale 3
      - elemento B: z-index vale 2
      - elemento C: hijo de A y z-index vale 1
      - elemento D: hijo de A y z-index 0
    - A sobre B, **C y D sobre B**, C sobre D

## 6.4.6. Posición y dimensiones



- Posición y dimensiones:

- Capas:

```
#caja_A { position: absolute; top: 100px; left: 100px; width:120px; height:100px;
          background-color:green;
          z-index:3; }
#caja_B { position: absolute; top: 75px; left: 75px; width:100; height:100px;
          background-color:red;
          z-index:2; }
#caja_C { position: absolute; top: 0px; left: 50px; width:50; height:50px;
          background-color:blue;
          z-index:1; }
#caja_D { position: absolute; top: 50px; left: 50px; width:50; height:50px;
          background-color:yellow;
          z-index:0; }
```

```
<div id="caja_A"> Caja A
  <div id="caja_C">Caja C</div>
  <div id="caja_D">Caja D</div>
</div>
<div id="caja_B"> Caja B </div>
```