

3 — Maquetación Web: CSS3

3.1 Introducción a CSS

Las hojas de estilo en cascada (CSS - *Cascading Style Sheets*) son un estándar W3C que define la presentación de los documentos Web y añade características dinámicas que junto a HTML5 permiten prescindir en gran medida de código JavaScript. Lo ideal es que la apariencia de un sitio Web pueda ser visualizada del mismo modo tanto en un monitor, teléfono móvil o tableta digital. Una de las principales características de CSS es que permite mantener separada la capa de presentación a la de estructuración (HTML) sin entremezclar las diferentes capas. Las principales versiones de CSS actuales son:

- **CSS Nivel 2, Revisión 1:** Es la versión que mayor repercusión ha tenido y ha sentado la base a la actual definición. No obstante, su especificación se queda corta a tiempos modernos puesto que fue desarrollada a lo largo de los años 1998 y 2004.
- **CSS Nivel 3:** Es la versión que mejor se ha integrado con HTML5 puesto que han nacido de la mano ambas tecnologías. Al margen de incorporar características de maquetación más amolde a los tiempos ha permitido elaborar interacción con el usuario pudiendo reducir en gran medida el código JavaScript. Además, tiene una mejor integración con tecnologías XML como son SVG o MathML.

Hoy en día no existe ninguna desventaja por utilizar CSS en la maquetación de sitios Web. De hecho, el único inconveniente actualmente es que no todos los navegadores Web cumplen con el estándar de modo completo. Entre las principales características positivas de utilizar CSS se pueden enumerar las siguientes:

1. **Mayor control en el diseño de las páginas:** Se pueden conseguir diseños que con los antiguos elementos de maquetación de HTML no se podía llegar.
2. **Menos trabajo:** Se puede modificar el estilo de todo un sitio Web modificando un solo archivo o se pueden tener diferentes estilos que se adapten a los usuarios o medios sin tener que reescribir la estructuración o los contenidos de la Web.
3. **Accesibilidad:** Al tener una buena estructuración y maquetación de los sitios Web permite construir sitios Web más accesibles tanto a usuarios como a dispositivos.

4. **Ampliamente estandarizado:** Casi todos los navegadores cumplen con la especificación 1.0, 2 y 2.1. No obstante, poco a poco se van estandarizando hacia el CSS3 y el camino será este puesto que ya está en preparación CSS4.

3.1.1 Aplicación de CSS en un documento HTML

Las hojas de estilos consisten en una o más reglas que describen la apariencia de un elemento de la estructura de HTML. Existen tres modos de aplicar las hojas de estilos en un documento HTML:

1. **Estilos en línea.** Se añade información del estilo directamente sobre un elemento HTML utilizando el atributo *style*. Esta manera de aplicarlo se hace directamente en el documento HTML. No obstante, no es recomendable hacer uso de este tipo de aplicación de CSS puesto que entremezcla los diferentes niveles (HTML y CSS). Este tipo de uso de hojas de estilos deriva de las primeras versiones de CSS y aunque se puede utilizar este modo de aplicación no está recomendada.

```
1 <p style="font-size:18px;_font-family:arial;_color:red">
2 Este texto tiene color azul
3 </p>
```

En el ejemplo anterior se muestra un párrafo en el que se escribe el texto *Este texto tiene color azul* pero su configuración de aspecto sería un tamaño de letra de 18px, tipo de fuente arial y color rojo.

2. **Hojas de estilos incrustados.** Este método es más compacto que el anterior puesto que se definen todas las reglas de estilos en la cabecera del documento bajo el elemento **<style>**. En este caso es obligatorio especificar el atributo *type* con el valor *text/css*. Además, se puede definir el atributo *media* el cual permite definir diferentes medios en los que se puede renderizar las reglas de CSS tales como pantalla (screen), dispositivo móvil, impresión, ...

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style type="text/css" media="screen">
5       h1 {font-size:18px;font-family:arial;color:red;}
6     </style>
7   </head>
8   <title> Página con estilos </title>
9   <body>
10    <h1> Título maquetado con estilos </h1>
11  </body>
12 </html>
```

En el ejemplo anterior se muestra una regla que especifica el formato que debe tener el elemento h1. En este caso hemos especificado las mismas reglas que en el ejemplo anterior para el párrafo. Este formato tampoco es recomendable puesto que los estilos no son reutilizables entre diferentes páginas del sitio Web.

3. **Hojas de estilos externas: vinculadas o importadas.** Esta manera de utilizar las hojas de estilos son las utilizadas hoy en día puesto que permiten tener fácilmente desacoplado los ficheros y pudiendo ser reutilizados sin complicaciones. Las hojas de estilos pueden ser importadas con la sentencia *@import*. Esta sentencia se puede utilizar en cualquier fichero de estilo o en el elemento **<style>**. Su utilización es mostrada en el siguiente código.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style type="text/css" media="screen">
5       <!--
6         @import url(dirección URL o local del archivo CSS);
7       -->
8     </style>
9   </head>
10  <!-- código html -->
11 </html>
```

Por otro lado, se pueden vincular o enlazar las hojas de estilos haciendo uso del elemento `<link>` de HTML. Este elemento debe ir en la cabecera del documento. Para vincular la hoja de estilos se deben utilizar los atributos `rel` y `href`. El atributo `rel` especifica la relación del documento vinculado que en este caso será una hoja de estilos (*stylesheet*). El atributo `href` especifica la URL del documento ya sea remoto o local. Además, utilizando este método se pueden vincular varias hojas de estilos siendo la más prioritaria la última en ser especificada puesto que las reglas serán sobreescritas por la última hoja de estilos.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="../styles/style1.css" type="text/
5       css"/>
6   </head>
7 </html>
```

3.1.2 Sintaxis de CSS

La sintaxis de todas las reglas de CSS están compuestas por varias partes que vamos a describir en base a varios ejemplos.

```
1 Selectores {
2   Propiedad1:Valor1;
3   Propiedad2:Valor2;
4   ....
5   PropiedadN:ValorN;
6 }
```

En el código anterior se describe la sintaxis básica de una regla de CSS. Cada una de las partes de esta regla es descrita a continuación:

- **Regla:** Es cada una de las especificaciones de formato que aparecen en un fichero CSS. Es decir, lo normal es que en un fichero de estilos existan varias reglas que describen el aspecto de un sitio Web. Las reglas en CSS están compuestas por un conjunto de selectores, que son las estructuras en las que se aplica la regla, entre las llaves () se describe una declaración de la regla.
- **Selector:** Los selectores son los elementos HTML a los que se aplican las reglas CSS. Estos selectores pueden estar combinados o ser en cascadas. en CSS3 se especifican nuevos selectores que no existían de modo natural en las anteriores versiones.
- **Declaración:** Especifica los estilos que se aplican a los elementos. Está compuesta siempre por un par de valores que son la propiedad que se quiere especificar y el valor.
- **Propiedad:** Es la característica que se quiere definir, por ejemplo el tamaño de la letra, el color, el borde, ...

- **Valor:** Establece el valor que debe tomar la propiedad por ejemplo, el color rojo, la tipografía Arial, ...

Es demasiado complicado para los diseñadores Web tener memorizadas todas las propiedades y valores que componen el estándar, en la versión 2.1. existían 115 propiedades y en CSS3 hay más de 240 propiedades, cada una con sus correspondientes posibles valores. Para aclarar la sintaxis anteriormente descrita se muestra un ejemplo aclaratorio:

```
1 p {font-size:12px; font-family:arial; color:#CC0066;}
2 h1 {font-size:18px; font-weight:bold; color:black;}
```

En este ejemplo se han especificado dos reglas relacionadas con los selectores p y h1 respectivamente. Posteriormente se puede observar que como para el selector p se ha especificado varias declaraciones como son el tamaño de la letra (*font-size*) a 12px, la tipografía (*font-family*) Arial y el color especificado en hexadecimal (#CC0066). Por otro lado, la otra regla especifica el formato para todos los elementos <h1>. En este caso, el tamaño de la letra será 18px, el formato de la letra será negrita y el color especificado el negro.

3.2 Selectores

La primera parte de cada una de las reglas CSS es definir sobre qué elementos se aplican las reglas. Esto es descrito utilizando los selectores. En las reglas CSS un mismo elemento puede estar implicado en varias reglas CSS y cada regla CSS puede aplicarse a un número ilimitado de elementos. Es decir, una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

Existen diferentes modos de determinar los selectores. De hecho en el estándar 2.1 existen hasta 12 formas diferentes de determinar los selectores. Además, el estándar 3 amplía el número posible de formas de determinar los selectores. A continuación se van a describir cada uno de estos métodos.

3.2.1 Selector universal

El primer selector es único y permite seleccionar todos los elementos de la página. Este selector puede ser utilizado para definir declaraciones generales y posteriormente ir refinando según los diferentes elementos. El selector universal se indica mediante un asterisco (*). En el siguiente elemento se elimina el margen (*margin*) y el relleno (*padding*) de todos los elementos HTML. En las siguientes secciones se describirán algunas de las principales propiedades de CSS.

```
1 * {
2   margin:0;
3   padding:0;
4 }
```

3.2.2 Selector de elementos

Los selectores de elementos se aplican sobre los diferentes elementos HTML que aparecen en la estructura del documento.

```
1 h1 {color:red;}
2 h2 {color:blue;}
3 p {color:black;}
```

En este ejemplo se han aplicado las reglas de que los titulares <h1> toman el color rojo, <h2> el color azul y el texto de los párrafos en negros.

Además, se pueden agrupar las reglas de varios elementos en una sola regla. Para ello, se incluyen todos los selectores que se quieren agrupar separados por una coma (,).

```
1 h1, h2, h3 {
2   color:#8A8E27;
3   font-weight:normal;
4   font-family:Helvetica;
5 }
```

Es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos.

```
1 h1, h2, h3 {
2   color:#8A8E27;
3   font-weight:normal;
4   font-family:Helvetica;
5 }
6
7 h1 { font-size:2em; }
8 h2 { font-size:1.5em; }
9 h3 { font-size:1.2em; }
```

El siguiente ejemplo establece en primer lugar las propiedades comunes de los títulos de sección (color y tipo de letra) y a continuación, establece el tamaño de letra de cada uno de ellos según cada selector.

3.2.3 Selector descendente

Una de las principales características de HTML es que los elementos se van encadenando uno dentro de otro en forma jerárquica. CSS3 suministra un mecanismo para poder seleccionar los selectores descendientes. De este modo se puede realizar una maquetación más específica. El modo de ir determinando los elementos hijos es separarlos por un espacio en blanco. Por ejemplo los párrafos que se encuentra dentro del elemento <article> quedarían descritos de la siguiente manera:

```
1 article p { color:_red; }
```

```
1 <article>
2   <p>Texto en rojo</p>
3   <a href=""> <p> Texto también en rojo </p> </a>
4 </article>
```

Observe que los dos textos serán afectados por la regla anterior. Esto es así porque los dos textos se encuentran en una estructura HTML jerárquica <article> ←<p>. Aunque la relación no es directa el selector descendente se aplica puesto que el elemento hijo se encuentra dentro del elemento padre aunque entre medio existan otros nodos. Además, si existieran otros elementos <p> pero no se encuentren dentro del elemento <article> no se les aplicará la regla de maquetación.

Ejercicio 3.1 Prueba y explica las diferencias de las siguientes tres reglas:

```
1 p table a em { text-decoration:_underline; }
2 p,table,a,em {text-decoration:underline;}
3 p * a em{text-decoration:underline;}
```

3.2.4 Selector de clase

Además de aplicar reglas sobre elementos del código HTML puede surgir la necesidad de diferenciar un conjunto de elementos frente a otros aunque estos elementos sean los mismos y estén en el mismo nivel de anidamiento. En ese caso se utilizan los selectores de clase, estos selectores requieren que en el código HTML se especifique el atributo *class*. En el siguiente ejemplo se muestran tres párrafos del cuerpo de la Web, en la que se desea destacar el primero frente a los demás. Al primer párrafo se le agrega el atributo class con un valor *destacado*. Para definir en una regla el estilo relativo a una clase se hace referencia indicándolo con un punto y seguido el nombre (*.nombreClase*).

```
1 <body>
2   <p class="destacado">Texto principal</p>
3   <p>Segundo texto sin resaltar</p>
4   <p>Tercer texto sin resaltar</p>
5 </body>
```

```
1 .destacado { color:red; }
```

En este caso la regla se aplica a cualquier elemento de la página cuyo atributo class sea igual a destacado. Este tipo de selectores son los más aplicados juntos a los selectores de identidad. La principal característica de este tipo de selectores frente a los de identidad es que en una misma página Web varios elementos pueden pertenecer a la misma clase. Un ejemplo un poco más complejo sería el siguiente.

```
1 .aviso {
2   padding: 0.5em;
3   border: 1px solid #98be10;
4   background: #f6feda;
5 }
6 .error {
7   color: #930;
8   font-weight: bold;
9 }
```

```
1 <body>
2   <p class="aviso">Texto de aviso.</p>
3   <p>Texto normal <a href="#" class="error"> aparecen errores</a> a
4     solucionar...</p>
5   <p>Otro texto <em class="aviso">con un aviso</em> importante</p>
6 </body>
```

En este caso se especificó se han definido elementos de clase aviso y error que se aplican cada vez que aparece un elemento de la clase especificada. No obstante, existen ocasiones que solamente se quiere aplicar una regla de una clase cuando se encuentra dicha clase a un tipo de elemento para eso se antepone el elemento a la especificación de la regla de clase.

```
1 p.aviso {color:yellow;}
```

En el ejemplo anterior se muestra el texto de color rojo cuando la clase aviso aparece en un elemento de párrafo (<p>). Así que, se podría deducir que cuando no se especifica el elemento de la clase es que en realidad se quiere especificar todos los elementos. Por lo tanto, las siguientes reglas son equivalentes.

```
1 .aviso {color:yellow;}
2 *.aviso {color:yellow;}
```


Hay que tener cuidado con los espacios y comas en la definición de las reglas puesto que en este caso se pueden combinar con las anteriores reglas descritas (descendencia y agrupamiento). A continuación se muestran tres claros ejemplos de reglas que parecen ser muy similares pero tienen un significado bastante diferente:

```
1 /* Todos los elementos de tipo "p" con atributo class="aviso" */
2 p.aviso { ... }
3 /* Todos los elementos con atributo class="aviso" que estén dentro
4   de cualquier elemento de tipo "p" */
5 p .aviso { ... }
6 /* Todos los elementos "p" de la página y todos los elementos con
7   atributo class="aviso" de la página */
8 p, .aviso { ... }
```

Finalmente se puede indicar que un elemento HTML puede recibir los estilos de varias reglas CSS sin ningún problema. Eso sí, se debe recordar que en caso de reglas que afectan sobre las mismas propiedades prevalecerá la que se aplica en un último lugar. En el ejemplo se muestra que un elemento puede pertenecer a varias clases siempre que se especifiquen por un espacio en blanco en el atributo class.

```
1 <p class="aviso_destacado_error">Párrafo de varias clases</p>
```

```
1 .error { color:red; }
2 .destacado { font-size:18px; }
3 .aviso { font-weight:bold;color:yellow; }
```

En el caso de que se quieran especificar los elementos que dispongan de más de una clase simultáneamente se debe especificar en la regla CSS ambas clases consecutivamente. A continuación se modifica la regla de estilos para que tenga la siguiente especificación:

```
1 .error { color:red; }
2 .error.destacado{color:blue;}
3 .destacado { font-size:15px; }
4 .aviso { font-weight:bold; }
```

El color que se tomará será el azul puesto que se ha utilizado un selector de clase múltiple que se puede interpretar como aquellos elementos de la página que dispongan de un atributo class con al menos los valores error y destacado.

3.2.5 Selectores de identidad

Los selectores de identidad se utilizan cuando se quiere especificar una regla para un único elemento de una página CSS. Es posible construir una regla para un único elemento haciendo uso de los selectores de clase pero se recomienda hacer uso del selector de identidad puesto que son más eficiente en su procesamiento.

Los selectores de identidad son especificados utilizando el valor del atributo id del elemento HTML. El elemento id no puede estar repetido en una página HTML. Esto es un error semántico, aunque los navegadores permiten leer dichas páginas los validadores no darán el sitio Web con una calificación apta.

El funcionamiento en las reglas CSS de los selectores de identidad es similar a la de clases pero el carácter para especificar una identidad es la amohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS.

```
1 #error{ color:red; }
```

```

1 <p>Primer párrafo</p>
2 <p id="error">Segundo párrafo</p>
3 <p>Tercer párrafo</p>

```

En el código anterior se muestra el selector `#error` con color rojo y este solamente se aplica al elemento `<p>` que corresponde con el segundo párrafo puesto que el atributo `id` tiene como valor `error`.

Las reglas de aplicación y selección múltiple de los selectores de clase también existen en los selectores de identidad. A veces es difícil apreciar la utilidad de aplicar el selector de identidad sobre un elemento pero esto es útil cuando el mismo fichero de estilos es utilizado para varias páginas y el identificador existe en las diferentes páginas (puesto que es único en cada una) pero aparece en diferentes elementos según la página HTML. En el siguiente código se muestran varias especificaciones del selector de identidad.

```

1 /* Todos los elementos de tipo "p" con atributo id="aviso" */
2 p#aviso { ... }
3 /* Todos los elementos con atributo id="aviso" que estén dentro
4    de cualquier elemento de tipo "p" */
5 p #aviso { ... }
6 /* Todos los elementos "p" de la página y todos los elementos con
7    atributo id="aviso" de la página */
8 p, #aviso { ... }

```

Ejercicio 3.2 Especifica a qué ámbitos de aplicación sobre un documento HTML afectan las siguientes especificaciones de reglas CSS y pon un ejemplo de código sobre dicha regla.

- `.aviso .alerta { ... }`
- `p.aviso div.alerta { ... }`
- `ul#menuPrincipal li.aviso a#contacto { ... }`

Ejercicio 3.3 Dado el siguiente cuerpo de una página HTML, se debe completar el fichero de especificaciones CSS según los criterios exigidos.

```

1 <body>
2   <div id="principal">
3     <p>The increase in the amount of solar data provided by <a href="#">
4       new satellites </a> makes it necessary to develop methods
5       to automate the detection of solar features. Here we present
6       a method for <em>automatically detecting active regions</em>
7       in solar extreme ultraviolet (EUV) images using a series of
8       steps.</p>
9   </div>
10  <div class="normal">
11    <p>Initially, the bright regions in the image are segmented using
12    seeded region growing. <span> <a href="#">In a second phase </a>
13    > these bright regions </span> are clustered into active
14    regions. <a href="#"> Partition-based clustering </a> (both
15    hard and fuzzy) and hierarchical clustering are compared in
16    this work. </p>
17    <p> The aim of the clustering phase <em class="resaltado"> is to
18    associate a group to each segmented region </em> in order to
19    reduce the total number of active regions. This facilitates the
20    documentation or subsequent monitoring of these regions.</p>
21    <p> We use two indicators to validate the partitioning: i) <em
22    class="resaltado">the number of detected clusters </em>
23    approximates the number of active regions reported by the
24    National Oceanic and Atmospheric Administration (NOAA) and ii)
25    the area that defines each cluster overlaps with the area of <a

```



```

        href="#">an active region of NOAA</a>. </p>
9   </div>
10  </body>

1  /* Todos los elementos de la pagina */
2  { font:_1em/1.3_Arial,_sans-serif; }
3  /* Todos los parrafos de la pagina */
4  { color:_#333; }
5  /* Todos los párrafos contenidos en #principal */
6  { color:_#336699; }
7  /* Todos los enlaces la pagina */
8  { color:_#CC3300; }
9  /* Los elementos "em" contenidos en #principal */
10 { background:_#FFFCC; padding:_.1em; }
11 /* Todos los elementos "em" de clase "resaltado" en toda la pagina */
12 { background:_#FFCC99; border:_1px_solid_FF9900; padding:_.1em; }
13 /* Elementos "span" contenidos en .normal */
14 { font-weight:_bold; }

```

3.2.6 Selector de hijos

Una de las características de los selectores descendientes es que no importa que el selector descendiente sea directo o tenga otros elementos intermedios. El selector de hijos es más específico en este sentido puesto que exige que el selector descendiente sea obligatoriamente el hijo sin elementos intermedios. El símbolo que se utiliza para definir esta característica es el signo mayor que (>). En el siguiente código se describe como los elementos que son hijos del elemento <p> tomarán el color azul. En el caso de que existan elementos intermedios estos no serán afectados por la regla CSS.

```

1  p > span {color:_blue; }
2
3  <p><span>Texto1</span></p>
4  <p><a href="#"><span>Texto2</span></a></p>

```

3.2.7 Selector adyacente

Los selectores adyacentes son reglas que se aplican a elementos que se encuentran en el mismo nivel (son hermanos) y uno a continuación del otro. Es decir, no puede haber nodos hijos en alguno de los adyacentes. El símbolo para especificar esta regla en CSS es el símbolo de suma (+). A continuación se muestra un ejemplo sencillo de aplicación sobre el uso de títulos y subtítulos, donde el subtítulo que aparece justo después de un título tendrá un color diferente a los subtítulos que aparecen en otras áreas de la página.

```

1  h2 { color:_green; }
2  h1 + h2 { color:_red; }
3
4  <body>
5    <h1>Titulo1</h1>
6    <h2>Subtítulo</h2>
7    <!-- Contenido de la Web -->
8    <h2>Otro subtítulo</h2>
9  </body>

```



Figura 3.1: CSS3: Selector de atributos.

Ejercicio 3.4 Explica las diferencias entre las siguientes reglas y expón un ejemplo de aplicación de cada uno de ellos:

```
1 p strong{color:_blue; }
2 p > strong{color:_blue; }
```

3.2.8 Selector de atributos

Otro interesante selector es la de especificar elementos por los atributos que tengan. Este tipo de selectores permiten por tanto seleccionar solamente los elementos HTML según sus atributos y/o valores. Este tipo de selectores tienen cuatro variantes en la versión 2.1. Este conjunto de selectores ha aumentado en la nueva especificación y por tanto, es en la Sección 3.9.1 donde se describen los nuevos selectores y algunos de los que ya existían en la versión 2.1. No obstante, los dos más básicos se presentan a continuación:

- **[nombre_atributo]:** Se seleccionan los elementos que tienen establecido el atributo *nombre_atributo* independientemente de cuál sea su valor.
- **[nombre_atributo=valor]:** Se seleccionan los elementos que tienen el atributo *nombre_atributo* con el valor asignado *valor*.

Para el segundo de los selectores de atributos se presenta el siguiente ejemplo:

```
1 <input type="text">
2 <input type="password">
3 <input type="submit">
4 <input type="reset">
```

```
1 input{background:white; }
2 input[type="text"]{border:2px_solid_#D667C2; }
3 input[type="password"]{border:2px_dashed_#71D1AF; }
4 input[type="submit"]{border:2px_dotted_#D1CE71; }
5 input[type="reset"]{border:2px_double_#2476E0; }
```

En este caso, los elemento `<input>` tienen un aspecto diferente en función del tipo de información que suministren. En la Figura 3.1 se muestra el resultado de los cuatros campos de input.

3.2.9 Pseudo-clases y pseudo-elementos

En HTML algunos elementos disponen de estados especiales. A estos estados especiales se les puede aplicar un formato específico. Así, las pseudo-clases son los estados de un elemento al que se les puede aplicar un estilo diferente. En principio existen hasta cuatro tipos diferentes de pseudo-clases:

- **Link:** Estas pseudo-clases se usan para definir el estilo al enlace, por ejemplo el del elemento `<a>`.
- **Dinámicas:** Estas pseudo-clases permiten modificar el aspecto a mostrar cuando el cursor está situado sobre estos elementos o se hace click en ellos.
- **Estructurales:** El efecto de una regla CSS es definida según la posición numérica que ocupe el elemento en la estructura HTML.

- **Otras:** Algunos elementos no encajan en las anteriores definiciones y disponen de sus propios estados o por el simple hecho de ser un tipo de elemento.

El tradicional ejemplo es el del elemento `<a>` (define hiperenlaces) el cual dispone de hasta cuatro estados diferentes:

- **normal:** Su estado normal y por defecto inicialmente.
- **visited:** El estado tras haber sido visitado.
- **hover:** Estado de tener el cursor sobre el enlace.
- **active:** Estado del momento en el que se está realizando la pulsación de click.

A continuación se va a describir un ejemplo en el cual se modifica la apariencia del elemento `<a>` según su estado.

```
1 a:link_{color:_red; text-decoration:_none;}
2 a:visited_{color:_blue; text-decoration:_none;}
3 a:hover_{color:_red; text-decoration:_underline;}
```

Los selectores de CSS, las pseudo-clases y todos los elementos HTML no son suficientes para poder aplicar estilos a algunos elementos especiales. Si se desea por ejemplo cambiar el estilo de la primera línea de texto de un elemento, no es posible hacerlo con las utilidades anteriores. La única forma de poder seleccionar estos elementos especiales es mediante los pseudo-elementos definidos por CSS para este propósito.

Por ejemplo, el pseudo-elemento `:first-line` permite seleccionar la primera línea de texto de un elemento. Así, la siguiente regla CSS muestra en mayúsculas la primera línea de cada párrafo.

```
1 p:first-line_{text-transform:_uppercase; }
```

En la Tabla 3.1 se muestran las pseudoclases y pseudoelementos aceptados por el estándar 2.1 de CSS. Además, se ha incorporado un ejemplo y una descripción de lo que consigue realizar dicho ejemplo.

Selector	Ejemplo	Descripción
:link	a:link	Selecciona todos los enlaces no visitados
:visited	a:visited	Selecciona todos los enlaces visitados
:active	a:active	Selecciona todos los enlaces activos
:hover	a:hover	Selecciona los enlaces con el ratón encima
:focus	input:focus	Selecciona los elementos <code><input></code> que tienen el foco
:first-letter	p:first-letter	Selecciona la primera letra de cada elemento <code><p></code>
:first-line	p:first-line	Selecciona la primera línea de cada elemento <code><p></code>
:before	p:before	Inserta contenido antes de cada elemento <code><p></code>
:after	p:after	Inserta contenido después de cada elemento <code><p></code>
:lang(language)	p:lang(es)	Selecciona todos los elementos <code><p></code> con el atributo <code>lang</code> es

Cuadro 3.1: Pseudoclases y Pseudoelementos

Ejercicio 3.5 Explica cómo se aplican los siguientes estilos en la página HTML.

```
1 p:first-child_i{color:blue; }
```

```
1 p > i:first-child{color:blue; }
```

```

1 <html>
2 <head> </head>
3 <body>
4 <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
5 <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
6 </body>
7 </html>

```

Ejercicio 3.6 Construye un sitio Web donde se haga uso de las pseudo-clases de la Tabla 3.1. Comprueba que entiendes el comportamiento de cada una de estas pseudo-clases.

Ejercicio 3.7 Al igual que el ejercicio anterior, construye un sitio Web en la que se haga uso de los pseudo-elementos de la Tabla 3.1.

3.3 Fundamentos de CSS

Antes de abordar las principales propiedades de los elementos de CSS. Se debe comprender los fundamentos de CSS, es decir, las unidades de medida, el modelo de trabajo de CSS (modelo de cajas) y cómo posicionar y visualizar los elementos utilizando las reglas CSS.

3.3.1 Unidades de medida

Un punto importante que se debe conocer son las unidades de medida para las propiedades CSS puesto que existen varios formatos para especificar las unidades de medida. Algunas son más recomendables según las circunstancias de cada diseño. Las medidas son tradicionalmente utilizadas para definir la altura, anchura, márgenes de los elementos o el tamaño de la tipografía. Todas las unidades van expresadas numéricamente bien por un valor entero o uno decimal seguido por la unidad de medida sin ningún espacio en blanco entre el número y la unidad de medida.

Lo primero que hay que comprender es que existen dos grandes grupos de unidades: Las absolutas y las relativas. Las medidas relativas definen su valor en relación a otra medida, por lo que para obtener el valor real, se debe realizar alguna operación de conversión. Por otro lado, las unidades absolutas establecen el valor real de la medida.

Unidades absolutas

Estas medidas indican el valor real puesto que no dependen de otro valor de referencia. Son varias las medidas absolutas que soporta CSS:

- **in**, pulgadas. Una pulgada equivale a 2.54 centímetros.
- **cm**, centímetros.
- **mm**, milímetros.
- **pt**, puntos. Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros.
- **pc**, picas. Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros.

Las unidades absolutas se visualizan del mismo modo en todas las pantallas, sean del tamaño que sea. Por lo tanto, no son recomendadas hoy en día puesto que existen multitud de dispositivos de diferentes tamaños y sería una ardua tarea construir hojas de estilos para cada uno de los tipos de pantallas que existen y existirán. En el siguiente código se muestra un ejemplo de aplicación de estas medidas.

```

1 /* El cuerpo de la página debe mostrar un margen de media pulgada */
2 body { margin: 0.5in; }
3 /* Los elementos <h1> deben mostrar un interlineado de 2 centímetros */
4 h1 { line-height: 2cm; }

```

```

5 /* Las palabras de todos los párrafos deben estar separadas 4 milímetros
   entre si */
6 p { word-spacing: 4mm; }
7 /* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos */
8 a { font-size: 12pt; }
9 /* Los elementos <span> deben tener un tamaño de letra de 1 pica */
10 span { font-size: 1pc; }

```

De todas las unidades absolutas, la única que se suele utilizar es el punto (pt) para el tamaño del texto en los documentos a imprimir. Es decir, para el medio print de CSS. Es un error grave en el diseño de interfaces establecer unidades absolutas en entornos de producción puesto que se pierde flexibilidad frente a otros medios como ya se ha comentado.

Unidades relativas

Las unidades relativas no están totalmente definidas sino que su valor es referenciado normalmente a otro valor. Esto aunque en un principio es más complejo que las unidades absolutas tiene como principal ventaja poder diseñar una sola interfaz de usuario que se adapte a las diferentes pantallas donde se visualiza. En principio se suministran tres unidades de medida en CSS:

- **em:** No tiene nada que ver con el elemento . Esta unidad equivale a la anchura de la letra M (eme mayúsculas) del tipo y tamaño de letra del elemento donde se define.
- **ex:** Igual que en el caso anterior pero la letra de referencia es la x minúsculas.
- **px:** Píxel, esta medida es relativa respecto a la resolución de la pantalla del dispositivo donde se visualiza la página HTML.
- **Porcentajes:** Se pueden definir los elementos según el porcentaje de ocupación del elemento donde se sitúan.

Las unidades de medida em y ex han existido durante décadas en la tipografía para adaptarlo a otros medios de comunicación. De este modo si se está utilizando una tipografía de 13 puntos en la región donde se define 1em, este tomará el valor de 13 puntos. El valor de 1 ex es aproximadamente 0.5 em.

Por defecto los navegadores Web definen el tamaño de la letra de los párrafos en 16píxel. En caso de que no se modifique en el estilo cuando se define 1em este tomará el valor de 16 píxeles. En el siguiente código se muestra la especificación de un margen de 1em o 16 píxeles en este caso.

```

1 p { margin: 1em; }

```

A continuación se muestra una especificación en la cual se ha modificado el tamaño de la letra a 32px en dicho caso el valor de 1em será el de 32px, si se quisiera modificar el margen a 16px habría que definirlo como 0.5em.

```

1 p { font-size: 32px; margin: 1em; }

```

Finalmente, las medidas en píxeles también son consideradas relativas puesto que su aspecto depende de la resolución del dispositivo donde se visualiza la Web. Si un elemento tiene una anchura de 400px, ocupará la mitad de una pantalla con una resolución de 800x600 pero ocupará menos de la tercera parte en una pantalla con resolución 1440x900. Es por esto que esta medida es bastante peligrosa puesto que los elementos pueden tener unas dimensiones no deseadas según el tamaño de la pantalla del usuario. En el siguiente ejemplo se visualiza la combinación de varias unidades de medida de un modo satisfactorio.

```

1 body { font-size: 12px; }
2 h1 { font-size: 3em; }

```

El tamaño de la letra que se define en el elemento `<body>` corresponde a 12 píxeles. A continuación, se asigna un tamaño de 3em al elemento `<h1>`, el cual se encuentra dentro del elemento `body` y por tanto el tamaño de la letra será 3 veces 12px = 36px.

La herencia en las medidas relativas no es directa, sino que se hereda el valor real después de haber sido calculado. Esto es importante puesto que puede conllevar a confusiones entre los principiantes en CSS. El siguiente código aclara esta situación.

```
1 body {
2   font-size: 12px;
3   text-indent: 3em;
4 }
5 h1 { font-size: 15px; }
```

La propiedad *text-indent* permite tabular la primera línea de un texto. Lo que nos interesa observar es que el valor que tiene asignado para ese espacio es de 3em. El elemento `<h1>` hereda las propiedades de sus nodos padre. En este caso será el elemento `<body>` el cual será *text-indent: 3em* para los elementos `<h1>`. El elemento `<h1>` redefine el tamaño de la fuente como 15px. Por lo tanto se podría deducir que el tamaño de *text-indent* del elemento `<h1>` debería ser 45px (3 x 15px) pero esto no es así porque el elemento que se hereda es primero calculado en el elemento padre y luego se hereda. Por lo tanto, en el elemento `<body>` el elemento *text-indent* toma el valor 36px (3 x 12px).

Los porcentajes también son unidades relativa, aunque CSS lo trata como unidades diferentes a las relativas su funcionamiento es similar a las relativas. De hecho, todos los elementos permiten definir sus tamaños con porcentajes. En el siguiente ejemplo se muestra como el tamaño de la letra del elemento `<body>` es 1em, al usar el valor por defecto será 16px. Por otro lado los tamaños de los titulares y subtulares toman los valores relativos de 200% y 150%, lo cual son equivalentes a 2em y 1.5em al encontrarse los elementos de titulares dentro del elemento `<body>`.

```
1 body { font-size: 1em; }
2 h1 { font-size: 200%; }
3 h2 { font-size: 150%; }
```

Los porcentajes son ampliamente utilizados para establecer la anchura de los elementos. En este caso, los elementos ocupan el porcentaje del elemento padre. Esto puede ser muy útil si se quieren construir interfaces que se adapten a diferentes pantallas de modo cómodo.

```
1 div#contenido { width: 600px; }
2 div.principal { width: 80%; }
```

```
1 <div id="contenido">
2   <div class="principal">
3     <!-- Contenido -->
4   </div>
5 </div>
```

3.3.2 Colores

Los colores en CSS pueden ser definidos de diversos modos: Colores en hexadecimal, colores en RGB (Red, Green, Blue), colores en RGBA (Red, Green, Blue, Alpha), colores en HSL (Hue, Saturation, Lightness), colores en HSLA (Hue, Saturation, Lightness, Alpha) y colores predefinidos por los navegadores.

Hexadecimal

Los colores en hexadecimal se definen siempre con la almohadilla (#) al principio y pueden tener 3 o 6 caracteres que cada uno de ellos representa un valor en hexadecimal. Cuando se utilizan 3 caracteres se hace uso de una paleta de colores reducida que permite crear colores básicos, en cambio, cuando se utilizan 6 caracteres se tiene una paleta de 24 bits (ocupa 32 bits en memoria pero como no hay transparencias realmente solo se utilizan 24).

Al usar 3 caracteres el primero corresponde al rojo, el segundo al verde y el tercero al azul. Cuando se usan 6 caracteres los colores se agrupan de 2 en 2 y siguen el mismo orden: rojo, verde y azul. Un valor en hexadecimal se expresa en base 16, esto quiere decir que disponemos en un solo carácter de un valor que llega de 0 a 15, y para representarlo además de números se pueden usar algunas letras : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Los valores cercanos a '0' son oscuros (negro), y los valores cercanos a 'F' son claros (blanco), por lo tanto para definir el color negro debemos especificar #000000, para el color blanco #FFFFFF, al estar dividido en tres canales (Red, Green, Blue) podemos especificar el color rojo con #FF0000, el verde con #00FF00, y el azul con #0000FF.

```
1 <p style='background-color:#FFFFFF;'> Blanco </p>
2 <p style='background-color:#000000;'> Negro </p>
3 <p style='background-color:#FF0000;'> Rojo </p>
4 <p style='background-color:#00FF00;'> Verde </p>
5 <p style='background-color:#0000FF;'> Azul </p>
```

Colores RGB (Red, Green, Blue)

Los colores RGB funcionan prácticamente igual que los colores hexadecimales, con la diferencia de que los representamos con un valor decimal, y que para crear el color debemos utilizar la función rgb. Cada canal de color puede ir de 0 a 255 (en hexadecimal 255 es FF), y al igual que con los colores hexadecimales los valores cercanos a cero son colores oscuros (negro) y los valores cercanos a 255 son colores claros (blanco). También es posible especificar cada canal de color por porcentajes del 0% al 100%.

```
1 <p style='background-color:rgb(255,255,255);'> Blanco </p>
2 <p style='background-color:rgb(0,0,0);'> Negro </p>
3 <p style='background-color:rgb(255,0,0);'> Rojo </p>
4 <p style='background-color:rgb(0,255,0);'> Verde </p>
5 <p style='background-color:rgb(0,0,255);'> Azul </p>
```

Colores RGBA (Red, Green, Blue, Alpha)

Los colores RGBA son una extensión del RGB y la única diferencia es que tienen un cuarto parámetro adicional, el cual indica su transparencia. La transparencia puede tener un valor desde 0.0 (transparente) a 1.0 (opaco). Esta característica existe solamente en los navegadores que son capaces de trabajar con CSS3.

```
1 <p style='background-color:rgba(255,255,255,0.5);'>Blanco translucido</p>
2 <p style='background-color:rgba(0,0,0,0.5);'> Negro translucido</p>
3 <p style='background-color:rgba(255,0,0,0.5);'> Rojo translucido</p>
4 <p style='background-color:rgba(0,255,0,0.5);'> Verde translucido</p>
5 <p style='background-color:rgba(0,0,255,0.5);'> Azul translucido</p>
```

Colores HSL (Hue, Saturation, Lightness)

Los colores HSL (Matriz, Saturación, Luminosidad) definen un modelo de color en términos de sus componentes constituyentes. El modelo HSL se representa gráficamente como un cono doble o un doble hexágono. Los dos vértices en el modelo HSL se corresponden con el blanco

y el negro, el ángulo se corresponde con el matriz, la distancia al eje con la saturación, y la distancia al eje blanco-negro se corresponde con la luminancia.

La matriz se define de 0 a 360 grados donde 0-360 son rojo, 120 es verde, y 240 es azul. La saturación es un porcentaje del 0% (gris) al 100% (color claro). La luminosidad es otro porcentaje del 0% al 100% donde el 0 es negro y el 100 es blanco. Se pueden especificar valores en HSL en todos los navegadores compatibles con CSS3.

```
1 <p style='background-color:hsl(0,0%,100%);'> Blanco </p>
2 <p style='background-color:hsl(0,0%,0%);'> Negro </p>
3 <p style='background-color:hsl(0,100%,50%);'> Rojo </p>
4 <p style='background-color:hsl(120,100%,50%);'> Verde </p>
5 <p style='background-color:hsl(240,100%,50%);'> Azul </p>
```

Colores HSLA (Hue, Saturation, Lightness, Alpha)

Los colores HSLA son una extensión de los colores HSL con el añadido del canal Alpha para realizar transparencias. Para ello se ha agregado un cuarto parámetro que especifica la transparencia y que puede ir de 0.0 (transparente) al 1.0 (opaco).

```
1 <p style='background-color:hsl(0,0%,100%,0.5);'>Blanco translucido</p>
2 <p style='background-color:hsl(0,0%,0%,0.5);'>Negro translucido</p>
3 <p style='background-color:hsl(0,100%,50%,0.5);'>Rojo translucido</p>
4 <p style='background-color:hsl(120,100%,50%,0.5);'>Verde translucido</p>
5 <p style='background-color:hsl(240,100%,50%,0.5);'>Azul translucido</p>
```

Ejercicio 3.8 Busca un chuletario a color con los colores ya definidos por los navegadores y sus especificaciones en diversas nomenclaturas. Este chuletario te servirá a partir de ahora para ser utilizado con las propiedades relativas a colores. ■

3.3.3 Modelo de cajas

El modelo de cajas es la característica más importante del diseño de interfaces Web ya que condiciona el diseño de todas las páginas del sitio Web. El modelo de cajas es el comportamiento que hace que todos los elementos de la página se represente mediante cajas rectangulares. Antes de la aparición de CSS, se utilizaban las tablas para estructurar un sitio Web. No obstante, las tablas no suministraban propiedades que permitiesen diseños modernos, así que fueron relegadas para solamente tabular/mostrar información.

Las cajas de una página se crean automáticamente cada vez que se inserta un elemento HTML. No es necesario que las cajas sean todas visibles por el usuario. De hecho, lo que suelen contener es todo el contenido del elemento. En la Figura 3.2 se muestra el modelo de cajas de la página CSS-tricks. Este resultado es fácilmente visible para cualquier página que estemos desarrollando en fase de desarrollo con el siguiente código CSS.

```
1 * { border:1px_solid_red!important; }
```

Cada una de las cajas está compuesta por seis partes. En la Figura 3.3 se muestra una representación gráfica del modelo de cajas para un elemento HTML. Estas partes se describen a continuación:

- **Border:** Define el borde del elemento, por defecto es 0 (sin borde).
- **Background:** El fondo del elemento, sus variaciones son un color solido o transparente (*background-color*), una imagen (*background-image*) o un gradiente con CSS3.
- **Padding:** Es un espacio que hay entre la caja y el contenido, se podría decir que es un margen interno.



Figura 3.2: CSS: Modelo de cajas de una Web.

- **Width/Height:** El width es la longitud del contenido, no de la caja en sí, la longitud real de la caja es la longitud del contenido + el padding y el border, y el height es el alto de la caja.
- **Margin:** Es el espacio que hay entre la caja y lo que está afuera, un margen externo.

En las siguientes subsecciones se van a presentar algunas de las propiedades y valores que se pueden ir adaptando en las reglas. No se van a especificar todas puesto que lo ideal es consultar la documentación oficial en <http://www.w3schools.com/cssref/> donde se va a encontrar una completa especificación de todas las propiedades y valores posibles, así como, una constante actualización que ningún manual puede ni debe cubrir.

Bordes

Cada uno de los bordes de cada caja pueden ser modificados. Las propiedades que pueden ser modificadas de los bordes de las cajas son la **anchura, color y estilo**. En la actualidad existen más de 20 propiedades relacionadas con los bordes.

La **anchura de los bordes** se controla con las propiedades: *border-top-width*, *border-right-width*, *border-bottom-width* y *border-left-width*. El siguiente código muestra un tamaño diferente para cada uno de los bordes del elemento <div>.

```

1 div {
2   border-top-width: 4em;
3   border-right-width: 1em;
4   border-bottom-width: 3em;
5   border-left-width: 2em;
6 }

```

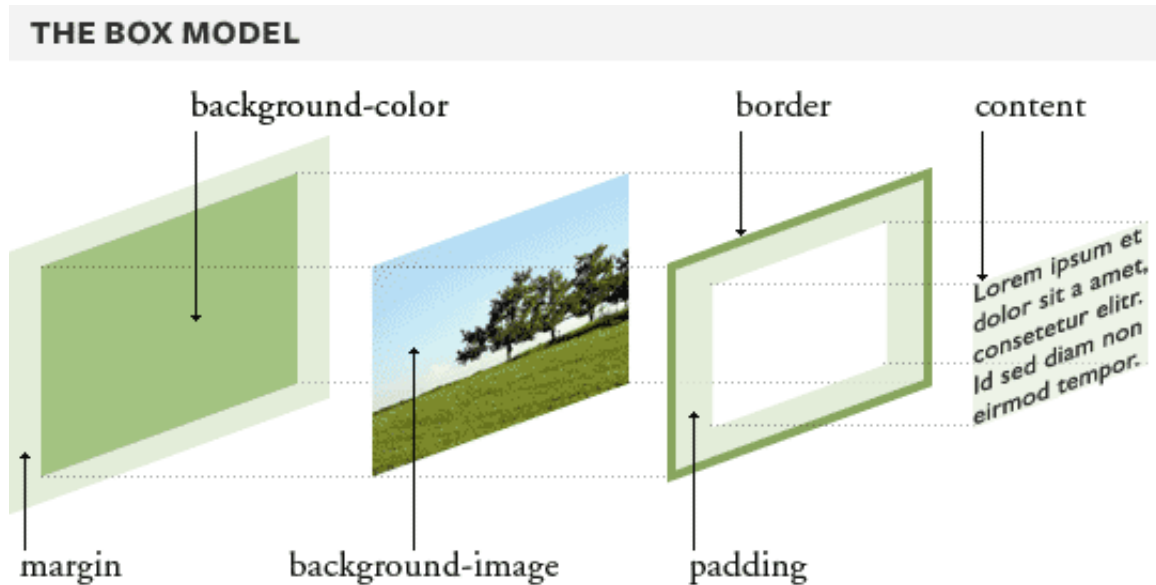


Figura 3.3: CSS: Modelo de cajas de un elemento HTML.

Además, existen propiedades denominadas *shorthand* en la cual se permiten configurar los cuatro elementos de una caja. En este caso el elemento que permite definir el borde de la caja es *border-width*. Las propiedades shorthand permiten definir desde 1 hasta los 4 elementos. Si se indica solo un valor este será idéntico para los cuatro elementos. Si se indican dos valores, el primero se aplica al borde superior e inferior y el segundo al borde izquierdo y derecho. Si se utilizan tres valores entonces el primero se aplica al borde superior el segundo al borde izquierdo y derecho y el tercero al borde inferior. Finalmente, si se definen los cuatros valores entonces cada valor corresponden al borde superior, derecho, inferior e izquierdo respectivamente. En el siguiente código se muestran los cuatro posibles usos de la propiedad *border-width*.

```
1 p { border-width:1em; }
2 p { border-width:1em_2em; }
3 p { border-width:1em_2em_3em; }
4 p { border-width:1em_2em_3em_4em; }
```

El **color de los bordes** funciona de un modo similar al borde puesto que existen cuatro atributos para definir cada uno de los colores de los bordes de la caja (*border-top-color*, *border-right-color*, *border-bottom-color*, *border-left-color*). Además, también existe una propiedad shorthand denominada *border-color*. En el siguiente código se muestra un borde diferente para cada uno de los bordes de cada elemento *div*.

```
1 div {
2   border-top-color: _#CC0000;
3   border-right-color: _blue;
4   border-bottom-color: _#00FF00;
5   border-left-color: _#CCC;
6 }
```

Por último, también se pueden establecer los **estilos de los bordes** mediante las propiedades *border-top-style*, *border-right-style*, *border-bottom-style* y *border-left-style*; y el shorthand *border-style*. Los estilos del borde solamente pueden ser definidos con las especificaciones (palabras reservadas) del estándar. El valor por defecto es *none* así que no se muestran los bordes

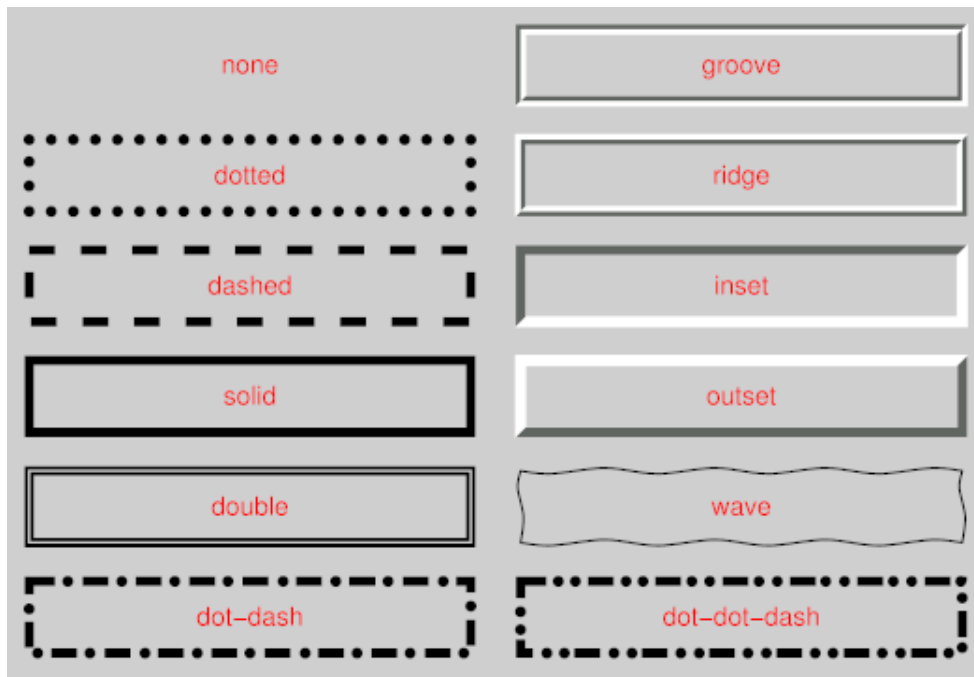


Figura 3.4: CSS: Estilos de los bordes.

de las cajas de los diferentes elementos. En la Figura 3.4 se muestran los diferentes estilos de los bordes que son aceptados en el estándar.

Además, se pueden definir todas las propiedades de los bordes de forma simultáneamente. Es decir, sería una propiedad shorthand global de los atributos del border denominada *border*. En el siguiente código se muestra como se ha especificado el tamaño de los cuatro bordes a 2px, el estilo sólido y el color azul para todos los elementos <div>.

```
1 div { border:2px_solid_blue; }
```

Ejercicio 3.9 Construye los estilos necesarios para obtener los resultados mostrados en las imágenes de la Figura 3.5. ■

Margen y relleno

Al igual que con los bordes de una caja, se definen cuatro propiedades para gestionar los espacios entre las cajas. Por ejemplo en el siguiente código se muestra como la clase espaciada especifica un margen a la izquierda de 2em.

```
1 .espaciada {margin-left:_2em;}
```

```
1 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam et elit.
2 Vivamus placerat lorem. Maecenas sapien. Integer ut massa. Cras diam
   ipsum,
3 laoreet non, tincidunt a, viverra sed, tortor.</p>
4
5 <p class="espaciado">Vestibulum lectus diam, luctus vel, venenatis
   ultrices,
6 cursus vel, tellus. Etiam placerat erat non sem. Nulla molestie odio non
7 nisl tincidunt faucibus.</p>
8
9 <p>Aliquam euismod sapien eu libero. Ut tempor orci at nulla. Nam in eros
```

Borders

Iste quidem veteres:

- inter ponetur honeste, qui vel mense brevi vel toto est iunior anno.
- Utor permissio, caudaeque pilos ut equinae paulatim vello unum, demo etiam unum, dum cadat elusus
- ratione ruentis acervi, qui redit in fastos et annis miraturque.

Quod si tam Graecis novitas

Ennius et sapines et fortis et alter Homerus:

- critici
- dicunt
- leviter curare
- videtur
- quo promissa

Hos ediscit et hos arto stipata theatro spectat Roma potens; habet hos numeratque poetas ad nostrum tempus Livi scriptoris ab aevo, si nimis antique, si dure.

Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et Iova iudicat aequo. Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilius dictare; sed emendata videri pulchraque et exactis minimum distantia miror. Inter quae verbum emicuit si forte decorum, et si versus paulo concinnior unus et alter, venditque poema.

(a)

Different colors on borders

(b)

Figura 3.5: CSS: Ejercicios de bordes.

```
10 egestas massa vehicula nonummy. Morbi posuere, nibh ultricies
    consectetuer tincidunt,
11 risus turpis laoreet elit, ut tincidunt risus sem et nunc.</p>
```

En el código anterior muestra el resultado de aplicar en un párrafo el margen relativo a la izquierda en 2em. En la Figura 3.6 se muestra una ilustración con la aplicación del padding, margin y border en una caja de texto.

La propiedad para controlar los espacios de relleno horizontales y verticales son *padding-top*, *padding-right*, *padding-bottom* y *padding-left*. En el siguiente código se muestra el efecto de aplicar relleno sobre un mismo párrafo.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       p{background-color:yellow;}
6       p.padding{padding-top:25px;
7                 padding-bottom:25px;
8                 padding-right:50px;
9                 padding-left:50px;}
10    </style>
11  </head>
12  <body>
13    <p>This is a paragraph with no specified padding.</p>
14    <p class="padding">This is a paragraph with specified paddings.</p>
15  </body>
16 </html>
```

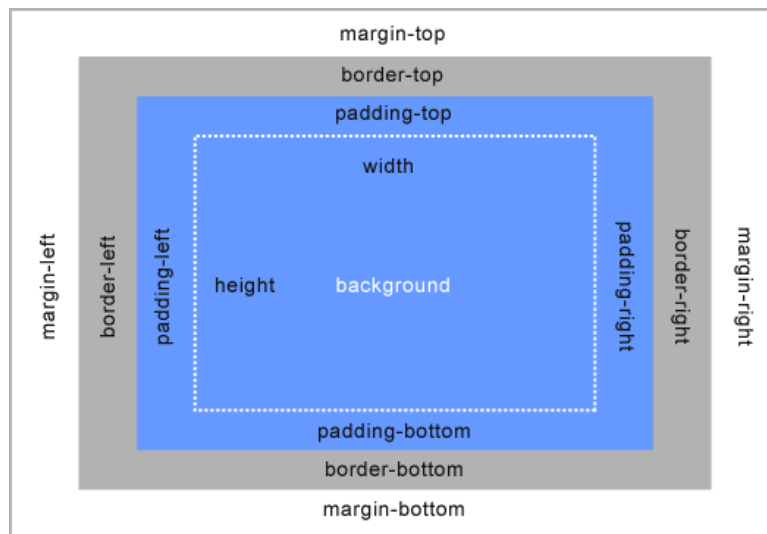


Figura 3.6: CSS: Padding, Márgenes y bordes.

Document with margins

Iste quidem veteres inter ponetur honeste, qui vel mense brevi vel toto est iunior anno. Utor permissio, caudaeque pilos ut equinae paulatim vello unum, demo etiam unum, dum cadat elusus ratione ruentis acervi, qui redit in fastos et annis miraturque.

Ennius et sapines et fortis et alter Homerus, ut critici dicunt, leviter curare videtur, quo promissa cadant et somnia Pythagorea. Naevius in manibus non est et mentibus haeret paene recens? Adeo sanctum est vetus omne poema. Ambigitur quotiens, sit prior, Pacuvius docti.

Hos ediscit et hos arto stipata

Hos ediscit et hos arto stipata theatro spectat Roma potens; habet hos numeratque poetas ad nostrum tempus Livi scriptoris ab aevo, si nimis antiquae, si dure.

Interdum volguis rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antiquae, si peraeque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et lova iudicat aequo. Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilius dictare; sed emendata videri pulchraque et exactis minimum distantia miror. Inter quae verbum emicuit si forte decorum, et si versus paulo concinnior unus et alter, venditque poema.

Indignor quicquam reprehendi, non quia crasse compositum illepedeve putetur, sed quia nuper, nec veniam antiquis, sed honorem et praemia posci. Recte necne crocum floresque perambulet Attae fabula si dubitem, clament periisse pudorem cuncti paene patres, ea cum reprehendere coner, quae gravis Aesopus, quae doctus Roscius egit; vel quia nil rectum, nisi quod placuit sibi, ducunt, vel quia turpe putant parere minoribus, et quae imberbes senes.

(a)

Headings and paddings

Ennius et sapines et fortis et alter Homerus, ut critici dicunt, leviter curare videtur, quo promissa cadant et somnia Pythagorea. Naevius in manibus non est et mentibus haeret paene recens? Adeo sanctum est vetus omne poema. Ambigitur quotiens, sit prior, Pacuvius docti.

Hos ediscit et hos arto stipata

Indignor quicquam reprehendi, non quia crasse compositum illepedeve putetur, sed quia nuper, nec veniam antiquis, sed honorem et praemia posci. Recte necne crocum floresque perambulet Attae fabula si dubitem, clament periisse pudorem cuncti paene patres, ea cum reprehendere coner, quae gravis Aesopus, quae doctus Roscius egit; vel quia nil rectum, nisi quod placuit sibi, ducunt, vel quia turpe putant parere minoribus, et quae imberbes senes.

Quod si tam Graecis novitas

Quod si tam Graecis novitas invisa fuisset quam nobis, quid nunc esset vetus? Aut quid haberet quod legeret tereretque viritum. Ut primum positus nugari Graecia bellis coepit et in vitium fortuna labier aequa, nunc athletarum studiis, nunc arsit equorum, marmoris aut eboris fabros aut aeris amavit, tibicinibus, nunc est gavis tragoedis; puella.

Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes; nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem aut ambulemus voluntatis esse, is non videt quae quamque rem res consequatur. Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes; nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem aut ambulemus voluntatis esse, is non videt quae quamque rem res consequatur. Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes; nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem aut ambulemus voluntatis esse, is non videt quae quamque rem res consequatur.

(b)

Figura 3.7: CSS: Ejercicios de márgenes y rellenos.

Anchura y altura

Las propiedades de CSS encargadas de especificar la anchura y altura de las cajas de los elementos son *width* y *height* respectivamente. La **propiedad width** no permite valores negativos y cuando se aplican porcentajes se tiene en cuenta el elemento padre del nodo al que se está aplicando la propiedad. Además, de especificar esta propiedad con valores numéricos se pueden definir dos palabras clave que son *auto* y *inherit*. El valor *auto* es el que se toma por defecto si no se especifica y es el navegador el encargado de definir la anchura del elemento en función de los contenidos y el espacio disponible. El valor *inherit* es utilizado para especificar que lo que se desea es utilizar el mismo espacio que está siendo utilizado por el elemento padre del nodo. En el Código 3.1 se especifica un tamaño de anchura de 250 píxeles para el lateral de un sitio Web.

Código 3.1: CSS3: Ejemplo de anchura de una caja.

```
1 aside { width: 250px; }
2
3 <aside>
4 ...
5 </aside>
```

La propiedad relacionada con la altura (*height* tiene un funcionamiento y los mismos atributos que la propiedad *width* . En el Código 3.2 se muestra un ejemplo en el cual se define una cabecera de 65 píxeles.

Código 3.2: CSS3: Ejemplo de altura de una caja.

```
1 header { height: 65px; }
2
3 <header>
4 ...
5 </header>
```

Ejercicio 3.10 Construye los documentos HTML y CSS necesarios para conseguir el mismo resultado que se muestra en la Figura 3.7

Ejercicio 3.11 Dada la siguiente plantilla HTML5 y la especificación básica CSS3 explique cómo se aplica cada una de las reglas CSS en el diseño.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8">
5     <link rel="stylesheet" type="text/css" href="styles.css" title="
      style">
6     <title>Mi primera página en HTML5/CSS3</title>
7   </head>
8   <body>
9     <header>
10      <div id="logo"><h1><span>LOGOTIPO</span></h1></div>
11      <div id="buscador">
12        <form action="#" method="post">
13          Buscar &nbsp; <input name="busqueda" type="text">
14        </form>
15      </div>
16      <div class="clear"></div>
17    </header>
18    <nav>
19      <ul id="menu_principal">
```



```

20     <li><a href="#">Inicio</a></li>
21     <li><a href="#">Productos</a></li>
22     <li><a href="#">Servicios</a></li>
23     <li><a href="#">Contacto</a></li>
24     <li><a href="#">Blog</a></li>
25 </ul>
26 <div class="clear"></div>
27 </nav>
28 <aside>
29     <article id="noticias">
30         <h3><a href="#">Noticias</a></h3>
31         <p><span class="fecha">22/03/2014</span> <a href="#">Noticia
32             1</a></p>
33         <p><span class="fecha">12/02/2014</span> <a href="#">Noticia
34             2</a></p>
35         <p><span class="fecha">30/12/2013</span> <a href="#">Noticia
36             3</a></p>
37         <p><span class="fecha">27/11/2013</span> <a href="#">Noticia
38             4</a></p>
39         <p><span class="fecha">24/10/2013</span> <a href="#">Noticia
40             5</a></p>
41         <h3><a href="#">Enlaces relacionados</a></h3>
42         <ul>
43             <li><a href="#">W3C</a></li>
44             <li><a href="#">Wikipedia</a></li>
45             <li><a href="#">Stackoverflow</a></li>
46         </ul>
47     </article>
48     <article id="publicidad">
49         <h3><a href="#">Publicidad</a></h3>
50         <p>El blog invierno cálido tiene entradas relacionadas con las
51             TI.</p>
52         <p>Si tomas zumo de bote no se te caerá el pelo.</p>
53         <p><a href="#">Seguir leyendo...</a></p>
54     </article>
55 </aside>
56 <main>
57     <section id="principal">
58         <article class="articulo">
59             <h2>Estructuración Web: HTML5</h2>
60             
61             <p>HTML5 (HyperText Markup Language, versión 5) es ...</p>
62             <p>Todavía se encuentra en modo experimental, lo ...</p>
63             <p><a href="#">Seguir leyendo...</a></p>
64         </article>
65         <article class="articulo">
66             <h2>Maquetación Web: CSS3</h2>
67             
68             <p>Las hojas de estilo en cascada o ...</p>
69             <p>La información de estilo puede ser ...</p>
70             <p><a href="#">Seguir leyendo...</a></p>
71         </article>
72     </section>
73     <section id="secundario">
74         <h2>JavaScript</h2>
75         <p>JavaScript es un lenguaje de...</p>
76         <p><a href="#">Seguir leyendo...</a></p>
77         <h2>jQuery</h2>
78         <p>jQuery es una biblioteca de...</p>
79         <p><a href="#">Seguir leyendo...</a></p>

```

```

74 </section>
75 </main>
76 <div class="clear"></div>
77 <footer>
78   <span class="enlaces">
79     <a href="#">Acerca de</a> |
80     <a href="#">Aviso legal</a> |
81     <a href="#">RSS</a> |
82     <a href="#">Otros</a> |
83     <a href="#">Enlace</a> |
84     <a href="#">Publicidad</a>
85   </span>
86   <span class="copyright">Copyright Carlos Caballero</span>
87 <div class="clear"></div>
88 </footer>
89 </body>
90 </html>

```

Ejercicio 3.12 A partir de la plantilla HTML5/CSS3 facilitada en el ejercicio anterior debe aplicar las siguientes reglas de bordes:

- La cabecera tiene un relleno de 1em en todos los lados.
- El menú de navegación tiene un relleno de 0.5em en todos los lados y un margen inferior de 0.5em.
- El resto de secciones (noticias, publicidad, principal, secundario) tienen un relleno de 0.5em a los cuatro lados.
- El pie tiene relleno en la zona superior e inferior (a los laterales no debe tener).
- Los artículos (clase en CSS y no en HTML5) deben tener una separación de 1em.
- Los elementos de publicidad están separados 1em de su elemento superior.
- El pie tiene un margen superior de 1em.

El resultado de de nuestra plantilla antes y después de estas modificaciones deben tener un aspecto similar al mostrado en la Figura 3.8(b).

Ejercicio 3.13 Continuando con la plantilla del ejercicio anterior ahora se van a realizar modificaciones en el borde. De hecho debe realizar los siguientes cambios:

- Eliminar el borde que muestran por defecto todos los elementos.
- El menú debe tener un borde inferior de 1 píxel y azul (#004C99).
- El elemento noticias debe tener un borde de 1 píxel y gris claro (#004C99).
- El elemento publicidad debe tener un borde discontinuo de 1 píxel y color (#CC6600).
- El elemento secundario debe tener un borde de 1 píxel y de color #CC6600.
- El pie debe mostrar borde superior e inferior de 1 píxel y color gris claro (#C5C5C5).

El resultado tras aplicar estos cambios es mostrado en la Figura 3.9(a)

Fondos

Los fondos son elementos muy importantes en el diseño de interfaces Web. En principio, los fondos se aplican sobre los diferentes elementos del modelo de caja. El fondo que se aplica puede ser un color o una imagen. El fondo es visualizado tanto en el área del ocupada por el contenido como por el relleno de la caja, mientras que, no se visualizará en el espacio del borde puesto que este tomará el fondo del propio borde. Los márgenes del modelo de caja no reciben ningún fondo, es decir, son transparentes.

Para la especificación del fondo se dispone de las siguientes propiedades en la versión 2.1. de CSS. Las novedades se verán en la sección destinada a las novedades de CSS3:

LOGOTIPO

[Inicio](#)
[Productos](#)
[Servicios](#)
[Contacto](#)
[Blog](#)

Noticias

22/03/2014 [Noticia 1](#)
12/02/2014 [Noticia 2](#)
30/12/2013 [Noticia 3](#)
27/11/2013 [Noticia 4](#)
24/10/2013 [Noticia 5](#)


Enlaces relacionados

[W3C](#)
[Wikipedia](#)
[Stackoverflow](#)

Publicidad


El blog invierno cálido tiene entradas relacionadas con las TI.
Si tomas zumo de bote no se te caerá el pelo.
[Seguir leyendo...](#)

Estructuración Web: HTML5



HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un clásico HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).^{1 2} Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.
[Seguir leyendo...](#)

Maquetación Web: CSS3



Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.
[Seguir leyendo...](#)

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.
[Seguir leyendo...](#)

jQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML.
[Seguir leyendo...](#)

[Acerca de](#) | [Aviso legal](#) | [RSS](#) | [Otros](#) | [Enlace](#) | [Publicidad](#)

© Copyright Carlos Caballero

(a)

LOGOTIPO

[Inicio](#)
[Productos](#)
[Servicios](#)
[Contacto](#)
[Blog](#)

Noticias

22/03/2014 [Noticia 1](#)
12/02/2014 [Noticia 2](#)
30/12/2013 [Noticia 3](#)
27/11/2013 [Noticia 4](#)
24/10/2013 [Noticia 5](#)

Enlaces relacionados

[W3C](#)
[Wikipedia](#)
[Stackoverflow](#)

Publicidad

El blog invierno cálido tiene entradas relacionadas con las TI.
Si tomas zumo de bote no se te caerá el pelo.
[Seguir leyendo...](#)

Estructuración Web: HTML5



HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un clásico HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).^{1 2} Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.
[Seguir leyendo...](#)

Maquetación Web: CSS3



Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.
[Seguir leyendo...](#)

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.
[Seguir leyendo...](#)

jQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML.
[Seguir leyendo...](#)

[Acerca de](#) | [Aviso legal](#) | [RSS](#) | [Otros](#) | [Enlace](#) | [Publicidad](#)

© Copyright Carlos Caballero

(b)

Figura 3.8: Ejercicio CSS Plantilla: (a) Plantilla original (b) resultado tras aplicar márgenes y rellenos.

LOGOTIPO

 Buscar
[Inicio](#)[Productos](#)[Servicios](#)[Contacto](#)[Blog](#)

Noticias

 22/03/2014 [Noticia](#)
[1](#)

 12/02/2014 [Noticia](#)
[2](#)

 30/12/2013 [Noticia](#)
[3](#)

 27/11/2013 [Noticia](#)
[4](#)

 24/10/2013 [Noticia](#)
[5](#)

Enlaces relacionados
[W3C](#)
[Wikipedia](#)
[Stackoverflow](#)

Publicidad

El blog invierno cálido tiene entradas relacionadas con las TI. Si tomas zumo de bote no se te caerá el pelo.
[Seguir leyendo...](#)

Estructuración Web: HTML5



HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un clásico HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).^{1 2} Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.

[Seguir leyendo...](#)

Maquetación Web: CSS3



Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.

[Seguir leyendo...](#)

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.

[Seguir leyendo...](#)

JQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML.

[Seguir leyendo...](#)

[Acerca de](#) | [Aviso legal](#) | [RSS](#) | [Otros](#) | [Enlace](#) | [Publicidad](#)

© Copyright Carlos Caballero

(a)

Logotipo

 Buscar
[Inicio](#)[Productos](#)[Servicios](#)[Contacto](#)[Blog](#)

Noticias

 22/03/2014 [Noticia](#)
[1](#)

 12/02/2014 [Noticia](#)
[2](#)

 30/12/2013 [Noticia](#)
[3](#)

 27/11/2013 [Noticia](#)
[4](#)

 24/10/2013 [Noticia](#)
[5](#)

Enlaces relacionados
[W3C](#)
[Wikipedia](#)
[Stackoverflow](#)

Publicidad

El blog invierno cálido tiene entradas relacionadas con las TI. Si tomas zumo de bote no se te caerá el pelo.
[Seguir leyendo...](#)

Estructuración Web: HTML5



HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un clásico HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).^{1 2} Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.

[Seguir leyendo...](#)

Maquetación Web: CSS3



Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.

[Seguir leyendo...](#)

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.

[Seguir leyendo...](#)

JQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML.

[Seguir leyendo...](#)

[Acerca de](#) | [Aviso legal](#) | [RSS](#) | [Otros](#) | [Enlace](#) | [Publicidad](#)

© Copyright Carlos Caballero

(b)

- **background-color.** Permite establecer un color de fondo. Sus valores pueden ser el color en hexadecimal/nombre, *transparent* y *inherit*. En el Código 3.3 se muestra como aplicar un color de fondo simple a toda una página Web completa a través de su elemento `<body>`.

Código 3.3: CSS3: Ejemplo de background-color.

```
1 body { background-color: #FF5F5F; }
```

- **background-image.** Permite especificar una imagen de fondo. Los valores que toman esta propiedad son *url*, *none* y *inherit*. En la URL se especifica la dirección donde se puede localizar la imagen de fondo, se puede establecer rutas absolutas o relativas. El valor *none* lo utilizamos para no utilizar ningún borde y finalmente el valor *inherit* es aplicado con las propiedades del nodo padre. En CSS se permite especificar simultáneamente una imagen de fondo y un color de fondo. Se muestra la imagen por delante del color fondo y en caso de que no se pueda cargar la imagen debido a un fallo de carga del servidor se muestra el color. En el Código 3.4 se muestra la aplicación de un fondo en la cabecera de un sitio Web.

Es importante aclarar que por defecto si la imagen es demasiado grande para el fondo del elemento solamente se muestra la parte de la imagen que se puede visualizar en el elemento. Por otro lado, si la imagen es más pequeña entonces se repetirá la imagen tanto en horizontal como en vertical hasta completar el fondo de una imagen. Este comportamiento al principio puede parecer algo molesto pero si queremos formar fondos a partir de texturas puede ser buena idea.

Código 3.4: CSS3: Ejemplo de background-image.

```
1 header { background-image: url("./img/header.png"); }
```

- **background-repeat.** El comportamiento por defecto es repetir las imágenes de fondo, pero esto puede no ser deseable según el tipo de imagen que se utilice de fondo para eso surge esta propiedad la cual toma los siguientes valores:

- **repeat:** En este caso la imagen se repite tanto en el eje de coordenadas x como y. Es el comportamiento por defecto.
- **repeat-x, repeat-y:** Se repite la imagen solamente en horizontal o vertical respectivamente. En el Código 3.5 se muestra la aplicación de repetir una imagen en el eje de coordenadas x para la definición de una imagen de fondo en la cabecera.

Código 3.5: CSS3: Ejemplo de repeat-x.

```
1 header { background: url("./img/head.png") repeat-x; }
```

- **no-repeat:** Al especificar esta propiedad se consigue que la imagen no se repita en ninguna coordenada.
- **background-position.** Se especifica la posición de la imagen en el fondo de la caja del elemento. Por defecto las imágenes de fondo se sitúan en la esquina superior izquierda. En caso de especificar dos porcentajes o unidades de medida el primero se refiere al desplazamiento horizontal y el segundo al vertical desde el origen. Si solamente se especifica un valor se considera que es el desplazamiento horizontal y al vertical se le asigna automáticamente el 50 %. Además, existen una serie de palabras reservadas que hacen referencia a porcentajes ampliamente utilizados: *top* = Las palabras clave permitidas son equivalentes a algunos porcentajes significativos: *top* = 0 %, *left* = 0 %, *center* = 50 %, *bottom* = 100 %, *right* = 100 %. En el Código 3.6 se muestra como una imagen de fondo se

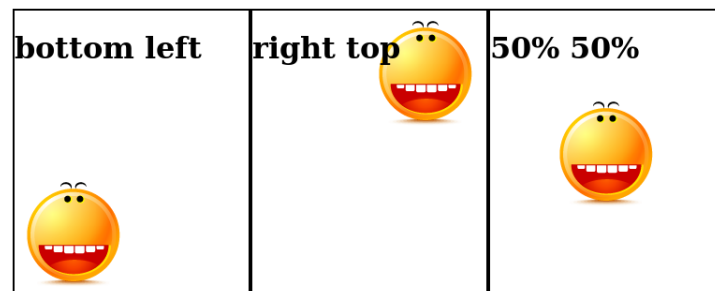


Figura 3.10: CSS3: Uso de background-position

puede situar en diferentes posiciones de su contenedor, en este caso un elemento `<div>`. En la Figura 3.10 se muestra el resultado de aplicación de este estilo.

Código 3.6: CSS3: Ejemplo de background-position.

```

1  div{
2    float:left;
3    border:2px_solid_black;
4    width:250px;
5    height:_300px;
6    background-image:_url("smile.png");
7  }
8  #caja1 {
9    background-repeat:_no-repeat;
10   background-position:_bottom_left;
11 }
12 #caja2 {
13   background-repeat:_no-repeat;
14   background-position:_right_top;
15 }
16 #caja3 {
17   background-repeat:_no-repeat;
18   background-position:_50%_50%;
19 }
20
21 <div id="caja1"><h1>bottom left</h1></div>
22 <div id="caja2"><h1>right top</h1></div>
23 <div id="caja3"><h1>50% 50%</h1></div>

```

- **background-attachment.** Permite especificar un fondo fijo, es decir, cuando el usuario se desplace en la Web a través de las barras de navegación quedará fijada la imagen de fondo. Los valores que se pueden especificar son *scroll*, *fixed* y *inherit*. Para que la imagen quede fija se hace uso del valor *fixed*.

Finalmente, también existe la propiedad *shorthand* denominada *background* que es la que generalmente se utiliza para definir las propiedades del fondo. El orden que se suele especificar es color, url de la imagen, repetición y posición. Esta propiedad no obliga a que se deban especificar todas las características sino que se pueden especificar aquellas que sean necesarias. En el Código 3.7 se muestra la especificación de una imagen de fondo a toda una página a través del elemento `<body>`.

Código 3.7: CSS3: Ejemplo de uso de background.

```
1 /* Color e imagen de fondo de la página mediante una propiedad shorthand
   */
2 body { background: _#222d2d_url(./graphics/colorstrip.gif)_repeat-x_0_0; }
```

Ejercicio 3.14 Continuando con el ejercicio de nuestra plantilla Web HTML5/CSS3 descrita en los ejercicios anteriores se debe mejorar el diseño agregando las siguientes características:

- Los elementos noticias y el pie de página tienen un color de fondo gris claro (#F8F8F8).
- El elemento publicidad tiene un color de fondo amarillo claro (#FFF6CD).
- Los elementos <h2> que se encuentran en el lateral de la clase secundario tendrán un color de fondo #DB905C y un padding de 0.2em.
- El fondo del menu es construido con una imagen llamada fondo_menu.gif (la cual deberá repetirse).
- El logotipo de la imagen es una imagen de fondo del elemento <h1> que se encuentra en la cabecera (imagen logo.gif).

El resultado de estas especificaciones deben dar un resultado similar al propuesto en la Figura 3.9(b).

3.4 Posicionamiento y visualización

CSS utiliza el flotado y el posicionamiento para tener el máximo control sobre el lugar que ocupa cada elemento en una página Web. En los siguientes apartados se presentan algunas propiedades de CSS 2.1 que se utilizan para controlar el posicionamiento de los elementos. Estas son: *float*, *clear*, *position*, *bottom*, *top*, *left*, *right*, *overflow*, *clip*, *visibility* y *z-index*.

El término *flujo normal* se entiende cuando los objetos de una página siguen el flujo normal del documento, es decir, los objetos ocupan el lugar indicado según el lugar que ocupan en el documento. El orden de lectura es de arriba a abajo y de izquierda a derecha. Utilizando las técnicas de posicionamiento y flotado se consigue que los elementos rompan con el flujo normal. Así puede ocurrir que un elemento que está escrito por detrás de otro elemento aparezca por encima a este.

Los elementos en HTML se pueden clasificar en dos grandes grupos:

- **Elementos de bloque.** Estos elementos empiezan siempre una nueva línea y ocupan todo el espacio necesario de la línea. Un claro ejemplo de estos elementos son los párrafos, los cuales comienzan una nueva línea cada vez que son definidos. Los elementos de bloque no pueden insertarse dentro de elementos en línea y tan solo pueden aparecer dentro de otros elementos de bloque.
- **Elementos en línea.** No comienzan una nueva línea y solo ocupan el espacio necesario para mostrar su contenido. El elemento de hipervínculos es un ejemplo de elemento en línea. Los elementos en línea pueden aparecer tanto dentro de un elemento de bloque como de otro elemento en línea.

Los navegadores son los encargados en primera instancia de crear y posicionar de forma automática las cajas de cada uno de los elementos de una página HTML. No obstante, es con CSS con el que se consigue un mayor control en la posición de cada una de las cajas. En un principio se definen hasta cinco modos diferentes para posicionar una caja:

1. **Normal o estático:** Se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario. Es decir, queda a disposición del navegador.

2. **Relativo:** En un principio se realiza el posicionamiento por defecto y en función a este se realiza un desplazamiento. Es decir, es relativo al normal.
3. **Absoluto:** Se especifica de un modo absoluto donde se debe situar el elemento respecto a su contenedor.
4. **Fijo:** Es una variante del absoluto y establece un elemento en inamovible. De este modo, cada elemento tiene una posición en pantalla y siempre es la misma independientemente del resto de elementos.
5. **Flotante:** Es el modelo más avanzado del posicionamiento ya que permite desplazar las cajas todo lo que el diseñador quiera hacia la izquierda o derecha.

3.4.1 Flotar los objetos

Las propiedades de flotar permiten mover una caja a la izquierda o derecha hasta el límite del borde del contenedor que la incluye o colisione con otra caja flotante. Para que un elemento pueda flotar debe tener definido implícitamente o explícitamente su tamaño. Las cajas flotantes no se encuentran en el *flujo normal* del documento por lo que las cajas que sí siguen el flujo normal se comportan como si las flotantes no estuviesen ahí.

La propiedad de flotar permite realizar diseños multicolumna, barras de navegación de listas, poner contenido en forma tabular sin necesidad de utilizar las tablas. La propiedad *float* puede tener los siguientes valores:

- **none:** El elemento no es flotante. Deshabilita esta opción.
- **left:** Hace que el elemento flote a la izquierda.
- **right:** Hace que el elemento flote a la derecha.
- **inherit:** Al igual que en otras ocasiones tome el valor de esta propiedad de su elemento padre.

Para comprender como los elementos flotan se va a presentar las diferentes opciones a lo largo de un ejemplo. Dado el Código 3.8 se muestran tres cajas con el flujo normal y lo que se muestran son las tres cajas una debajo de la otra ya que son elementos de bloque y cada uno ocupa una línea. El resultado se muestra en la Figura 3.11(a).

Código 3.8: CSS3: Elemento de flotar.

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8" />
5     <link rel="stylesheet" type="text/css" href="estilos.css" />
6     <title>Flotar</title>
7   </head>
8   <body>
9     <div id="contenido">
10      <div id="caja1"> Caja 1 </div>
11      <div id="caja2"> Caja 2 </div>
12      <div id="caja3"> Caja 3 </div>
13    </div>
14  </body>
15 </html>
16
17 body { font-weight: bold; color: white; }
18 #contenido {
19   width: 700px; margin: auto;
20   border: solid 1px black; background-color: #CCCCCC; }
21 #caja1 {

```

```

22 width:_100px; height:_100px; background-color:#FF0000;
23 border:_solid_1px_black; margin:_10px; }
24 #caja2 {
25 width:_130px; height:_130px; background-color:#00FF00;
26 border:_solid_1px_black; margin:_10px; }
27 #caja3 {
28 width:_160px; height:_160px; background-color:#0000FF;
29 border:_solid_1px_black; margin:_10px; }

```

A continuación se van a hacer varias operaciones sobre el código anterior. Imaginad que se quiere hacer flotar el elemento de la caja1 hacia la derecha. Solamente se deberá modificar el código anterior para que quede como el descrito en el Código 3.9. El resultado se puede observar en la Figura 3.11(b), observe que no llega hasta el borde de la caja que lo contiene porque se han definido márgenes de 10 píxeles.

Código 3.9: CSS3: Elemento de flotar.

```

1 #caja1{
2 width:_100px; height:_100px;
3 background-color:_#FF0000;
4 border:_solid_1px_black;
5 margin:_10px;
6 float:_right;
7 }

```

En el caso de que el elemento de la caja1 hubiera flotado hacia la izquierda lo que sucede es que el elemento de la caja1 ya no corresponde su comportamiento con el flujo normal mientras que la caja2 y caja3 siguen aún en el flujo normal. Esto produce que la caja1 y la caja2 se solapen como se muestra en la Figura 3.11(c).

Si queremos tener alineadas a la derecha las tres cajas sin que se encuentren solapadas entre sí se deben establecer las tres cajas como elementos flotantes. El resultado de aplicar este nuevo estilo se puede visualizar en el resultado mostrado en la Figura 3.11(d). Observe que la caja contenedora tendrá ahora un tamaño de 0 píxeles, esto es así porque este contenedor no tiene ningún elemento en su interior ya que las tres cajas ahora no están en el flujo normal sino que están flotando en el diseño. No obstante, las tres cajas siguen en el interior del elemento contenedor puesto que no se salen del espacio de sus dimensiones. Para conseguir que la caja contenido se vuelva a visualizar se dispone de tres opciones:

1. Se aplica altura al elemento contenido en la propiedad *height*.
2. Se hace flotar a la izquierda el elemento contenido. De esta forma volvería a aparecer pero flotando a la izquierda de su contenedor. En este caso sería el elemento `<body>`.
3. Añadir dentro del elemento contenido y a continuación de las tres cajas un elemento `<div>` vacío con la propiedad *clear* y valor *both*. Esta es la solución más utilizada y la que utilizaremos en un principio.

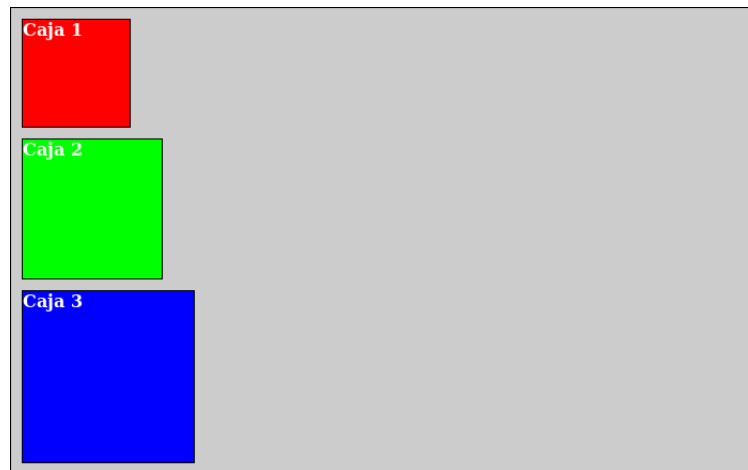
Como se ha expuesto anteriormente de las tres soluciones propuestas se va a utilizar la tercera. En el Código 3.10 se muestra el código modificado para conseguir el resultado mostrado en la Figura 3.11(e).

Código 3.10: CSS3: Elemento de flotar.

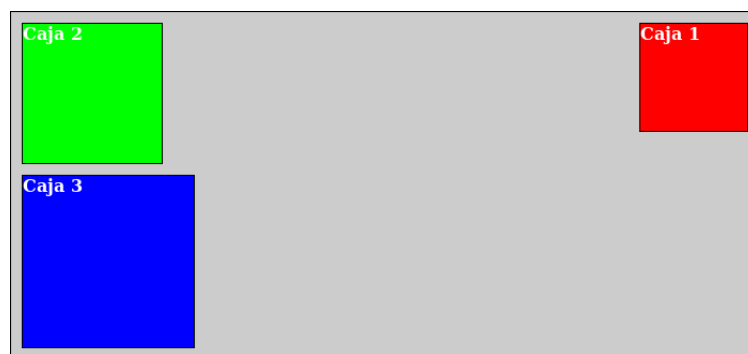
```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8" />

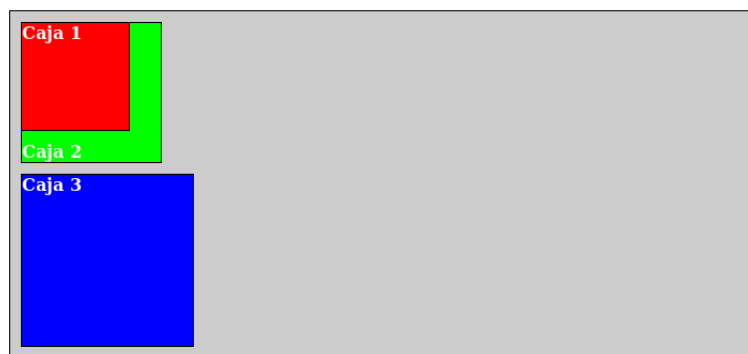
```



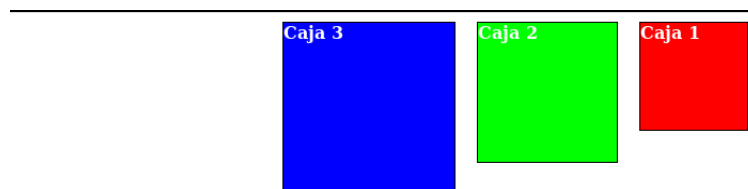
(a)



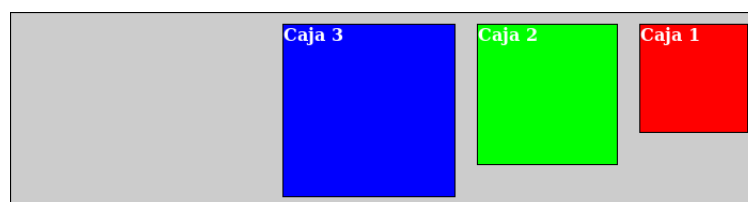
(b)



(c)



(d)



(e)

Figura 3.11: CSS3 Uso de flotar las imágenes: (a) flujo normal (b) caja1 flota hacia la derecha (c) caja1 flota hacia la izquierda (d) las tres cajas flotando hacia la derecha (e) uso de la propiedad clear.

```

5      <link rel="stylesheet" type="text/css" href="estilos.css" />
6      <title>Flotar</title>
7  </head>
8  <body>
9      <div id="contenido">
10         <div id="caja1"> Caja 1 </div>
11         <div id="caja2"> Caja 2 </div>
12         <div id="caja3"> Caja 3 </div>
13         <div class="clearboth"></div>
14     </div>
15 </body>
16 </html>
17
18 body { font-weight:_bold; color:_white; }
19 #contenido {
20     width:_700px; margin:_auto;
21     border:_solid_1px_black; background-color:_#CCCCCC; }
22 #caja1 {
23     width:_100px; height:_100px; background-color:#FF0000;
24     border:_solid_1px_black; margin:_10px; }
25 #caja2 {
26     width:_130px; height:_130px; background-color:#00FF00;
27     border:_solid_1px_black; margin:_10px; }
28 #caja3 {
29     width:_160px; height:_160px; background-color:#0000FF;
30     border:_solid_1px_black; margin:_10px; }
31 .clearboth{clear:_both;}

```

3.4.2 Posicionar los elementos

Una vez presentado el modelo de cajas y cómo éstas pueden flotar es necesario avanzar hacia los modelos de posicionamiento y visualizado para comprender cómo se organizan los elementos hoy en día en un diseño Web. Las principales propiedades son *display* y *position*.

La propiedad *display* permite transformar los elementos en su forma de visualizarse. Recordamos que existen dos grandes grupos: bloques y en línea. Esta propiedad permite establecer que un elemento bloque se visualice como un elemento en línea y viceversa.

Para establecer el posicionamiento de una caja se utiliza la propiedad *position* la cual puede tomar los siguientes valores:

- **static:** Corresponde con el posicionamiento estático. Si se especifica este valor se ignoran todos los demás valores de posicionamiento. Es el que se utiliza por defecto en los navegadores. Solamente se considera si el elemento es de bloque o en línea y sus principales propiedades son *width* y *height*.
- **relative:** Corresponde al posicionamiento relativo. El desplazamiento de la caja se controla con las propiedades *top*, *right*, *bottom* y *left*. Las cajas se desplazan en función de su posición original (la establecida en el posicionamiento normal). En el Código 3.11 se muestra como se ha desplazado la caja2 en modo relativo 8em desde la parte superior. El resultado se puede apreciar en la Figura 3.12(a). Aunque este tipo de posicionamiento puede parecer muy versátil tiene como principal problema de posicionar elementos de forma relativa es que se pueden producir solapamientos con otros elementos de la página.

Código 3.11: CSS3: Ejemplo de uso de position relative.

```

1 #caja3 {
2     width:_160px; height:_160px; background-color:#0000FF;
3     border:_solid_1px_black; margin:_10px;

```

```

4  position: _relative;
5  top: _8em;
6  float: right; }

```

- **absolute:** Corresponde al posicionamiento absoluto. El desplazamiento de la caja se controla con las mismas propiedades que *relative* pero las coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor. Este tipo de posicionamiento se utiliza para establecer de forma exacta la posición en la que se debe mostrar la caja. Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página se ven afectados y modifican su posición. Al igual que en el posicionamiento relativo, cuando se posiciona de forma absoluta una caja es probable que se produzcan solapamientos con otras cajas. El proceso para determinar la posición de un elemento absoluto sigue los siguientes pasos:

1. Se buscan todos los elementos contenedores de la caja hasta llegar al elemento <body> de la página.
2. Se recorren todos los elementos contenedores empezando por el más cercano a la caja y llegando hasta el <body>.
3. El primer elemento contenedor que esté posicionado de cualquier forma diferente a la posición estática se convierte en la referencia que determina la posición de la caja posicionada de forma absoluta.
4. Si ningún elemento contenedor está posicionado, la referencia es la ventana del navegador, que no debe confundirse con el elemento <body> de la página.

Una vez determinada la referencia del posicionamiento absoluto, se debe realizar la interpretación de los valores de las propiedades *top*, *right*, *bottom* y *left*.

- El valor de la propiedad *top* indica el desplazamiento desde el borde superior de la caja hasta el borde superior del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad *right* indica el desplazamiento desde el borde derecho de la caja hasta el borde derecho del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad *bottom* indica el desplazamiento desde el borde inferior de la caja hasta el borde inferior del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad *left* indica el desplazamiento desde el borde izquierdo de la caja hasta el borde izquierdo del elemento contenedor que se utiliza como referencia.

En el Código 3.12 se modifica el estilo para definir que la caja3 se debe situar desplazado 2em de la parte superior e izquierda de su contenedor. En el caso de que no se especifique en el contenedor la propiedad *position* se toma en cuenta su valor por defecto que es estático y por tanto ese elemento no sería su contenedor. Por defecto, si no existe ningún elemento padre que tenga definida la propiedad *position* el elemento contenedor será <body>. El resultado de realizar un posicionamiento absoluto se visualiza en la Figura 3.12(b).

Código 3.12: CSS3: Ejemplo de uso de *position relative*.

```

1 #caja3 {
2   width: _160px; height: _160px; background-color: #0000FF;
3   border: _solid_1px_black; margin: _10px;
4   position: _absolute;
5   top: _2em;
6   left: _2em;
7 }

```

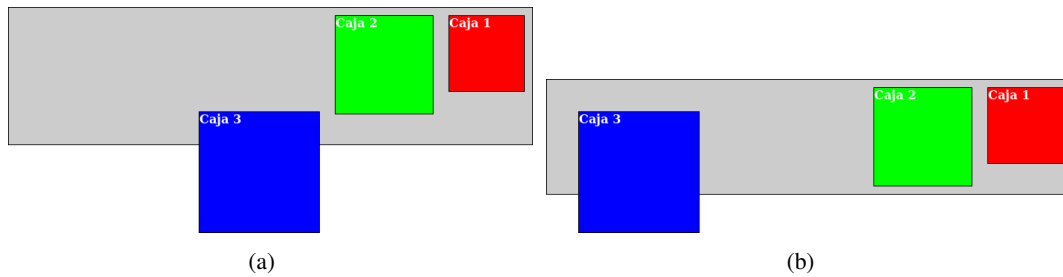


Figura 3.12: CSS3 Uso de posicionar las imágenes: (a) relativo en top a la caja 3 (b) absoluto en top y left a la caja3.

- **fixed:** Corresponde al posicionamiento fijo. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanecerá inamovible en la pantalla.

Ejercicio 3.15 Busca un chuletario con la clasificación de todos los elementos HTML5 según si son de bloque o de línea.

Ejercicio 3.16 Debe escribir el código HTML5 y CSS3 necesario para alcanzar los resultados mostrados en la Figura 3.13.

Ejercicio 3.17 Dada nuestra plantilla HTML5/CSS del código anterior debes conseguir que las imágenes de los artículos estén alineados a la derecha o a la izquierda. Se debe utilizar el elemento `<figure>` de HTML5. El resultado debe ser similar a la de la imagen 3.14.

Visibilidad y capas de los elementos

La propiedad *visibility* permite especificar si un elemento es visible o no. Los valores que se pueden especificar son *visible* y *hidden*. Es importante recordar que aunque un elemento no sea visible, éste continua ocupando su espacio en el flujo normal del documento. Para evitar este efecto se debe especificar en la propiedad *display* el valor *none*.

Se recuerda al lector que utilizando el posicionamiento puede ocurrir que ciertos elementos se solapen entre sí. La propiedad *z-index* permite controlar el orden en el que se presentan los elementos que quedan solapados. Así que, esta propiedad establece el orden en el eje de coordenadas *z* de los elementos, es decir, un orden de apilamiento entre diferentes capas. Por defecto, los elementos se apilan en el orden en el que aparecen en el documento, es decir, el elemento situado más abajo en el flujo normal del documento quedará encima. Los elementos que se solapan se orden por el orden de la propiedad *z-index* apareciendo en la parte superior aquellos que tienen un valor superior.

Ejercicio 3.18 Utilizando diversas capas consigue el efecto de tener un conjunto de cartas en las cuales se solapan las cartas. El resultado debe ser similar al mostrado en la Figura 3.15. ■

3.5 Propiedades de texto y tipografía

Las hojas de estilo permiten un control total sobre el formato del texto en los sitios Web. En esta sección se presentarán algunas de las principales propiedades de CSS utilizadas para formatear el texto.

By Hans Christian Andersen

A soldier came marching along the high road: "Left, right – left, right." He had his knapsack on his back, and a sword at his side; he had been to the wars, and was now returning home.

As he walked on, he met a frightful-looking old witch in the road. Her under-lip hung quite down on her breast, and she stopped and said, "Good evening, soldier; you have a very good sword, and a large knapsack, and you are a real soldier; so you shall have as much money as ever you like."

"Thank you, old witch," said the soldier.

"Do you see that large tree," said the witch, pointing to a tree which stood beside them. "Well, it is quite hollow inside, and you must climb to the top, when you will see a hole, through which you can let yourself down into the tree to a great depth. I will tie a rope round your body, so that I can pull you up again when you call out to me."

"But what am I to do, down there in the tree?" asked the soldier.

"Get money," she replied; "for you must know that when you reach the ground under the tree, you will find yourself in a large hall, lighted up by three hundred lamps; you will then see many doors, which can be easily opened, for the keys are in all the locks. On entering the first of the chambers, to which these doors lead, you will see a large chest in the middle of the floor, and upon it a dog seated, with a pair of eyes as large as teacups. But you need not be at all afraid of him; I will give you my word that the dog which you must spread upon the floor, and then boldly seize hold of the dog, and place him upon it. You can then open the chest, and take from it what you please, they are only copper pence; but if you would rather have silver money, you must go into the second chamber. Here you will find another chest as big as mill-wheels; but do not let that trouble you. Place him upon my apron, and then take what money you please. If, however, you like gold, go into the third chamber, where there is another chest full of it. The dog who sits on this chest is very dreadful; his eyes are as big as a tower, but do not mind him; place him upon my apron, he cannot hurt you, and you may take from the chest what gold you will."

"This is not a bad bargain," said the soldier; "but what am I to give you, you old witch? for, of course, you do not mean to tell me all this for nothing."

(a)

Box 1

Box 2

(b)

Box 4

Box 3

Figura 3.13: CSS3: Ejercicios de posicionar elementos.

3.5.1 Propiedades de la tipografía

Las propiedades de las fuentes en CSS son usadas para configurar la apariencia deseada para el texto de un documento.

- **font-family:** Permite especificar un nombre de fuente en concreto o bien una familia genérica de fuentes. Se puede especificar una lista de fuentes separadas por comas teniendo en cuenta que si el nombre de la fuente o familia tiene algún espacio habrá que encerrarlo entre comillas. En esta propiedad existe la limitación de que la fuente debe estar instalada en el equipo cliente, es por ello que habrá que utilizar las denominadas fuentes seguras presentadas en el Capítulo 1. No obstante, en las novedades de CSS3 se presenta un nuevo método que permite utilizar cualquier tipografía. Se define una lista de fuentes debido a que si no existe la primera fuente en el cliente se procederá a tratar de cargar la siguiente fuente y así sucesivamente.

```
1 font-family: _Arial, _Helvetica, _sans-serif;
2 font-family: _"Times_New_Roman", _Times, _serif;
3 font-family: _"Courier_New", _Courier, _monospace;
4 font-family: _Georgia, _"Times_New_Roman", _Times, _serif;
5 font-family: _Verdana, _Arial, _Helvetica, _sans-serif;
```

- **font-size:** Permite especificar el tamaño del texto. Se pueden especificar los tamaños utilizando la forma absoluta, relativa, con valores numéricos o con porcentajes.
- **font-weight:** Permite especificar el espesor o intensidad de las fuentes. Se pueden especificar diferentes valores cerrados: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800 y 900. Los valores que normalmente se utilizan son normal (el valor por



Figura 3.14: CSS3: Ejercicio de flotar elementos <figure>.

Royal Flush



Figura 3.15: CSS3: Ejercicio de usar diferetes capas.

defecto) y **bold** para los textos en negrita. El valor normal equivale al valor numérico 400 y el valor **bold** al valor numérico 700.

- **font-style:** Permite especificar el estilo de la fuente. Existen tres valores posibles: *normal* que no configura ningún estilo (es el valor por defecto); *italic* que equivale a tener una fuente *cursiva*; y *oblique* en este caso se tiene el efecto de tener una fuente cursiva. Es decir, en *oblique* solamente se inclinan las letras de la fuente pero no corresponden realmente a un tipo de fuente cursiva como en el caso anterior.
- **font-variant:** Permite especificar dos posibilidades de variantes de texto: *normal* y *small-caps*. Al especificar *normal* el texto no cambia de apariencia y con *small-caps* el texto se mostrará en mayúsculas pero de un tamaño inferior.
- **color:** Permite especificar el color de la fuente, utilizando cualquiera de los métodos ya presentados anteriormente.

Además, existe una propiedad *shorthand* denominada *font* que nos permite indicar de forma directa algunas o todas las propiedades de la tipografía de un texto. El orden en el que se deben indicar las propiedades del texto es el siguiente:

1. En primer lugar y opcionalmente se indican *font-style*, *font-variant* y *font-weight* en cualquier orden.
2. Posteriormente se indica obligatoriamente el valor de *font-size* y opcionalmente la altura de la línea (*line-height*). La propiedad *line-height* se presentará a continuación.
3. Finalmente, se indica obligatoriamente el tipo de letra a utilizar.

Algunos ejemplos de utilizar la propiedad *font* se presenta a continuación.

```
1 font: 76%/140%_Verdana,Arial,Helvetica,sans-serif;
2 font: normal_24px/26px_Century_Gothic", "Trebuchet_MS",Arial,Helvetica,
   sans-serif;
3 font: normal_.94em_Trebuchet_MS",Arial,Helvetica,sans-serif;
4 font: bold_1em_Trebuchet_MS",Arial,Sans-Serif;
5 font: normal_0.9em_Lucida_Grande",_Verdana,_Arial,_Helvetica,_sans-serif
   ;
6 font: normal_1.2em/1em_helvetica,_arial,_sans-serif;
7 font: 11px_verdana,_sans-serif;
8 font: normal_1.4em/1.6em_helvetica",_arial,_sans-serif;
9 font: bold_14px_georgia,_times,_serif;
```

3.5.2 Propiedades de textos

Las propiedades de texto permiten aplicar estilos a los textos espaciando sus palabras o sus letras, decorándolo, alineándolo, transformándolo, etcétera. Algunas de estas propiedades son:

- **text-align:** Esta propiedad especifica la alineación del texto. Los valores definidos por CSS permite alinear el texto según los valores: a la izquierda (*left*), derecha (*right*), centrado (*center*) y justificado (*justify*). La propiedad *text-align* no sólo alinea el texto que contiene un elemento, sino que también alinea todos sus contenidos, como por ejemplo las imágenes.
- **text-decoration:** Permite decorar el texto con subrayados y otros efectos. Los valores que se pueden especificar son: *none* (ninguno), *underline* (subrayado), *overline* (con una línea por encima), *line-through* (tachado), *blink* (parpadeante) e *inherit* (heredado).
- **text-transform:** Especifica la apariencia de las letras. Los valores que puede tener son: *none* (texto normal), *capitalize* (cada palabra comienza con mayúsculas), *uppercase* (todo el texto aparece en mayúsculas) y *lowercase* (todo el texto aparece en minúsculas).

```

1 <div style="text-transform:_none"><h1>Original</h1>Lorem ipsum dolor
2 sit amet...</div>
3 <div style="text-transform:_capitalize"><h1>text-transform:_
  capitalize</h1>
4 Lorem_ipsum_dolor_sit_amet...</div>
5 <div_style="text-transform:_lowercase"><h1>text-transform:_lowercase
  </h1>
6 Lorem_ipsum_dolor_sit_amet...</div>
7 <div_style="text-transform:_uppercase"><h1>text-transform:_uppercase
  </h1>
8 Lorem_ipsum_dolor_sit_amet...</div>

```

- **line-height:** Permite especificar el espacio que hay entre dos líneas consecutivas.

```

1 p { line-height:_1.4; } /* Establece un factor de multiplicación
  respecto al tamaño font-size */
2 p { line-height:_14pt; } /* Establece una distancia fija entre las l
  íneas de 14 puntos */
3 p { line-height:_140%; } /* Establece una distancia proporcional (%)
  respecto a font-size */

```

- **text-indent:** Permite especificar el sangrado de la primera línea del elemento. Lo tradicional es dejar un espacio sangrado en los párrafos.

```

1 p { text-indent: 20px; } /* Define un sangrado positivo. El valor
  por defecto es 0*/
2 p { text-indent: -12px; } /* Éste es negativo (hacia la izquierda)
  */
3 p { text-indent: 10%; } /* Aquí está como un porcentaje respecto al
  ancho del elemento padre*/

```

- **letter-spacing:** Especifica el espacio que hay entre los caracteres. Este valor aumenta o disminuye en relación con el espacio definido por la fuente utilizada en el elemento.

```

1 p { letter-spacing:_12px; } /* Expande los caracteres. Los separa */
2 p { letter-spacing:_-0.5px; } /* Contrae los caracteres. Los junta
  */

```

- **white-space:** Establece cómo se gestionan los espacios en blanco. Los valores que se pueden definir son: *normal* (los espacios en blanco adicionales son ignorados por el navegador), *pre* (los espacios en blanco adicionales son utilizados al igual que con el elemento <pre>), *nowrap* (no se produce el ajuste de línea automático por lo que el texto permanecerá en la misma línea hasta que se encuentre un elemento de salto de línea).
- **vertical-align:** Permite establecer la alineación vertical en una misma línea de varios elementos diferentes, ya sean imágenes o texto. Los valores que puede tomar esta propiedad son *baseline*, *sub*, *super*, *top*, *text-top*, *text-bottom*, unidades de medida y porcentajes. Para aclarar todos estos elementos se presentan estos valores en la imagen de la Figura 3.16. El valor por defecto es *baseline*.

Ejercicio 3.19 Se deben realizar las siguientes modificaciones al diseño de nuestra plantilla HTML5/CSS. El resultado debe ser similar al mostrado en la Figura 3.17.

- La fuente base de la página debe ser: color negro, tipo Arial, tamaño 0.9em, interlineado 1.4.
- Los textos del cuerpo del sitio Web estarán justificados, los textos de la parte izquierda estarán alineados a la izquierda y los de la parte derecha se encontrarán centrados.

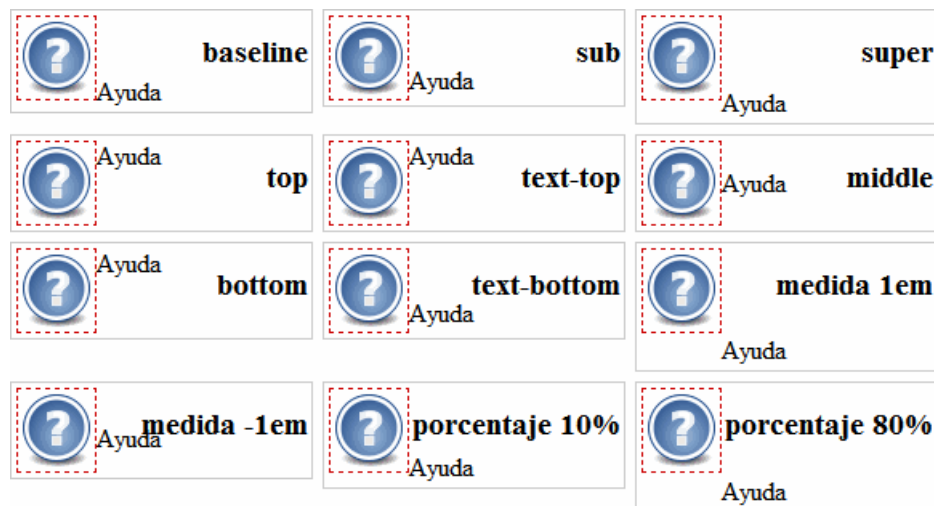


Figura 3.16: CSS3: Ejemplo de uso de la propiedad *vertical-align*.



Figura 3.17: CSS3: Ejercicio de maquetar textos en la plantilla HTML5/CSS3.

- Los elementos `<h2>` de `.articulo` se muestran en color `#CC6600`, con un tamaño de letra de 1.6em, un interlineado de 1.2 y un margen inferior de 0.3em.
- Los elementos del menu deben mostrar un margen a su derecha de 1em y los enlaces deben ser de color blanco y tamaño de letra 1.3em.
- El tamaño del texto de todos los contenidos del lateral debe ser de 0.9em. La fecha de cada noticia debe ocupar el espacio de toda su línea y mostrarse en color gris claro `#999`. El elemento `<h3>` de `#noticias` debe mostrarse de color `#003366`.
- El texto del elemento publicidad es de color gris oscuro `#555` y todos los enlaces de color `#CC6600`.
- Los enlaces contenidos dentro de `.articulo` son de color `#CC6600` y todos los párrafos muestran un margen superior e inferior de 0.3em.
- Añadir las reglas necesarias para que el contenido de secundario se vea como en la imagen que se muestra.
- Añadir las reglas necesarias para que el contenido de pie se vea como en la imagen que se muestra.

3.6 Propiedades de hiperenlaces

La especificación del tamaño, color y decoración de los hiperenlaces se realizan con las mismas propiedades que para los textos debido a que los hiperenlaces son una especialización de los textos. Es decir, se podrán utilizar propiedades tales como *color*, *text-decoration* o *font-weight*. A continuación se muestra un ejemplo en el cual se configuran las diferentes propiedades de los hiperenlaces.

```

1 a { margin:_1em_0; display:_block; }
2
3 .alternativo {color:_#CC0000; }
4 .simple {text-decoration:_none; }
5 .importante {font-weight:_bold; font-size:_1.3em; }
6 .raro {text-decoration:underline; }
7
8 <a href="#">Enlace con el estilo por defecto</a>
9 <a class="alternativo" href="#">Enlace de color rojo</a>
10 <a class="simple" href="#">Enlace sin subrayado</a>
11 <a class="importante" href="#">Enlace muy importante</a>
12 <a class="raro" href="#">Enlace con un estilo raro</a>

```

Al principio del tema se presentaron las pseudo-clases. Algunas de las pseudo-clases más importantes son las relacionadas con los hiperenlaces. Un hiperenlace puede tener diferentes estados y estos son descritos por los siguientes hiperenlaces:

- **:link.** Se aplica estilos a los enlaces que aún no han sido visitados por el usuario.
- **:visited.** Se aplica estilos a los enlaces que han sido visitados anteriormente por el usuario. El historial de enlaces visitados se borra automáticamente cada cierto tiempo y el usuario también puede borrarlo manualmente.
- **:hover.** Se aplica estilos al enlace sobre el que el usuario ha posicionado el puntero del ratón.
- **:active.** Se aplica estilos al enlace que está pinchando el usuario. Los estilos sólo se aplican desde que el usuario pincha el botón del ratón hasta que lo suelta, por lo que suelen ser unas pocas décimas de segundo.

Como se sabe, por defecto los navegadores muestran los enlaces no visitados de color azul y subrayados y los enlaces visitados de color morado. Las pseudo-clases anteriores permiten modificar completamente ese aspecto por defecto y por eso casi todas las páginas las utilizan.

El siguiente ejemplo muestra cómo ocultar el subrayado cuando el usuario pasa el ratón por encima de cualquier enlace de la página:

```
1 a:hover{text-decoration:none; }
```

El uso de las pseudo-clases cuando se aplica sobre clases o identificadores siguen las mismas reglas que se presentaron en los selectores. En el siguiente código se muestran unos cuantos ejemplos de aplicación de las pseudoclasas en el elemento `a` de diferentes partes de un documento Web.

```
1 a.importante:visited{...;}
2 a#principal:hover{...;}
3 div#menu ul li a.primerito:active{...;}

```

Otra de las características que puede resultar interesante según los diseños Web es la de incluir un pequeño icono al lado de un enlace para indicar el tipo de contenido que enlaza, como pueden ser iconos de archivos comprimidos, documentos .DOC o .PDF. Es relativamente sencillo conseguir este efecto utilizando la propiedad *background-image* de los enlaces.

```
1 a { margin:1em0; float:left; clear:left; }
2 .rss {
3   color:#E37529;
4   padding:000018px;
5   background:#FFF url(imagenes/rss.gif) no-repeat left center;
6 }
7 .pdf {
8   padding:000022px;
9   background:#FFF url(imagenes/pdf.png) no-repeat left center;
10 }

```

```
1 <a href="#">Enlace con el estilo por defecto</a>
2 <a class="rss" href="#" alt="Enlace a un archivo RSS"></a>
3 <a class="pdf" href="#" alt="Enlace a un documento PDF"></a>

```

Este efecto es mejorado en CSS3 con la utilización de nuevos selectores, los cuales nos permiten seleccionar un elemento según el comienzo o finalización de un texto.

Ejercicio 3.20 Definir el documento HTML y las reglas necesarias para los siguientes estados de enlaces:

- En su estado normal, los enlaces se muestran de color verde.
- Cuando el usuario pasa su ratón sobre el enlace, se muestra con un color de fondo verde y la letra de color blanco.
- Los enlaces visitados se muestran en color gris claro.

Ejercicio 3.21 Continuando con nuestra plantilla HTML5/CSS3. Definir los estilos para los diferentes estados de los enlaces para que se amolden al diseño. ¿Qué colores y por qué los escogerías?

Ejercicio 3.22 En nuestra plantilla HTML5/CSS3. Definir los estilos para sustituir los enlaces que creas oportunos por sus versiones en iconos.

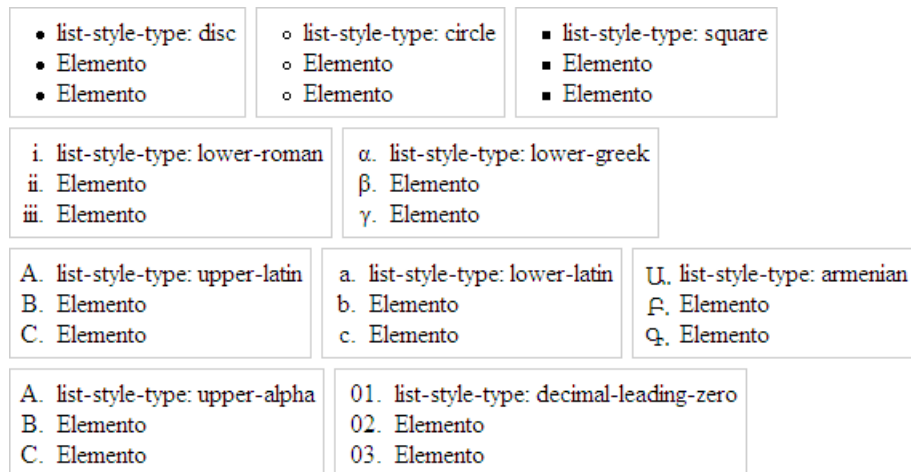


Figura 3.18: CSS3: Estilos de tipos de listas.

3.7 Propiedades de listas

Los navegadores han tomado como consenso para las listas no ordenadas mostrar un pequeño círculo de color negro y las listas ordenadas muestran la numeración decimal. No obstante, estos adornos pueden ser configurados utilizando CSS tanto en apariencia como en posición. La propiedad básica que controla el tipo de viñetas se denomina *list-style-type* y acepta los siguientes valores:

- **none.** Los elementos de la lista no tendrán adorno ninguno. Es ampliamente utilizado para crear los menús de navegación, puesto que se necesita eliminar las numeraciones.
- **disc, circle y square.** Los elementos de viñeta son un círculo relleno, círculo vacío o un cuadrado relleno respectivamente.
- **decimal, decimal-leading-zero, lower-roman, upper-roman, armenian y georgian.** Especifica los valores numéricos.
- **lower-latin, lower-alpha, upper-latin, upper-alpha y lower-greek.** Especifica los valores alfanuméricos.

En la Figura 3.18 se muestra un resumen de las diferentes opciones de listas en las que se modifica los elementos de la lista.

Otra de las propiedades de control de las listas en CSS es *list-style-position* la cual permite controlar la colocación de las viñetas. Los valores que se pueden asignar son *inside* y *outside*. El valor *inside* indica que las viñetas se situará dentro de la caja mientras que al utilizar el valor *outside* las viñetas se situarán fuera de la caja. La aplicación de esta propiedad es mostrada en la Figura 3.19.

Utilizando las propiedades anteriores no se puede personalizar el color y tamaño de los elementos de viñeta. La propiedad que nos va a permitir especificar estas características es *list-style-image*, la cual permite modificar una imagen propia en la viñeta. Las imágenes personalizadas se especifican mediante la URL de la imagen, en caso de que la imagen no se pueda cargar se muestra la viñeta automática. En la Figura 3.20 se muestra el resultado de aplicar los estilos descritos en el Código 3.13.

Código 3.13: CSS3: Ejemplo de uso de *list-style-image*.

```
1 ul.ok { list-style-image: url("imagenes/ok.png"); }
2 ul.flecha { list-style-image: url("imagenes/flecha.png"); }
3 ul.circulo { list-style-image: url("imagenes/circulo_rojo.png"); }
```

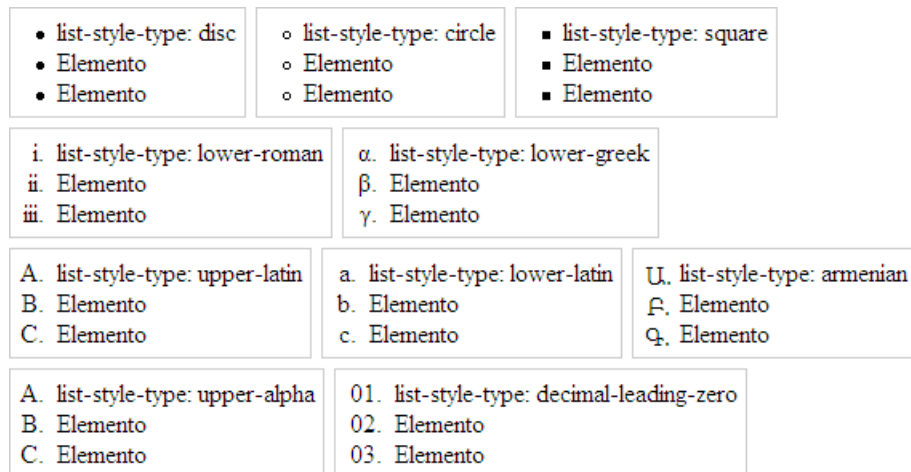


Figura 3.19: CSS3: Atributo *list-style-position* de las listas.

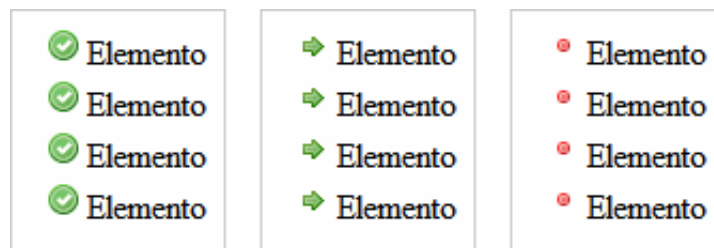


Figura 3.20: CSS3: Atributo *list-style-figure* de las listas.

Además, existe la propiedad *shorthand* denominada *list-style* que permite configurar las propiedades *list-style-type*, *list-style-position* y *list-style-image* en cualquier orden.

Creación de sistemas de navegación

En esta sección se van a describir las técnicas básicas de creación de sistemas de navegación tanto vertical como horizontal utilizando las listas dentro del elemento de navegación (<nav>).

Menú vertical

El menú vertical se construye al igual que el horizontal utilizando el elemento , el cual permite agrupar todas las opciones del menú y es a partir de CSS el que realiza toda la tarea de modificar y adaptar su aspecto tanto en horizontal como en vertical. A continuación se va a describir paso a paso como se puede confeccionar un menú vertical de navegación con diferentes enlaces. El primer paso consiste en construir una lista de elementos con los enlaces a las diferentes secciones como se describe en el siguiente código.

```

1 <ul>
2 <li><a href="#">Elemento 1</a></li>
3 <li><a href="#">Elemento 2</a></li>
4 <li><a href="#">Elemento 3</a></li>
5 <li><a href="#">Elemento 4</a></li>
6 <li><a href="#">Elemento 5</a></li>
7 <li><a href="#">Elemento 6</a></li>
8 </ul>

```

En la Figura 3.21 se muestran las imágenes del menú de navegación previa a la aplicación de los estilos y el resultado final del menú de navegación tras haber aplicado los estilos. Así que,

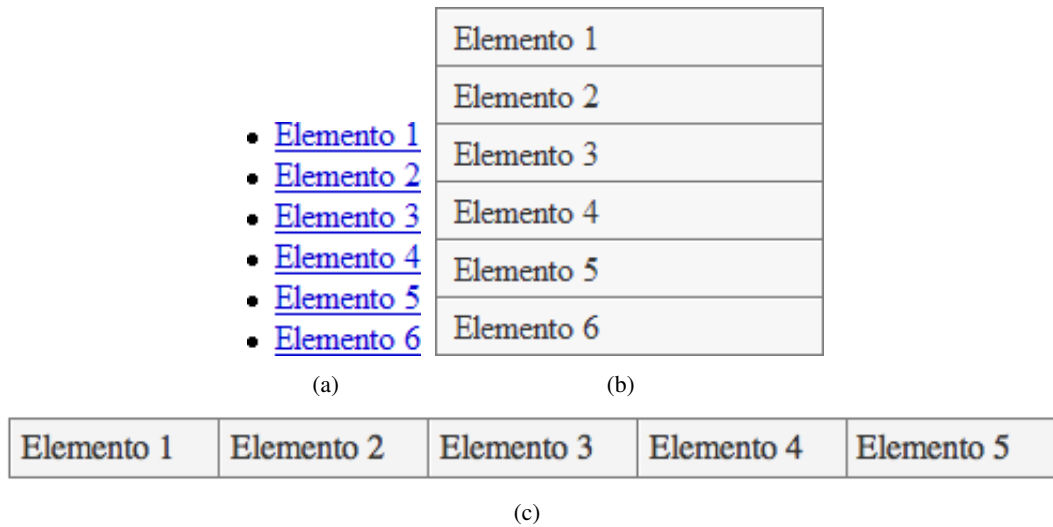


Figura 3.21: CSS3: (a) Menú original sin aplicar estilos (b) menú vertical (c) menú horizontal.

una vez que están definida la estructura del menú, se van a aplicar los siguientes estilos:

1. Definir la anchura del menú (180px).

```
1 ul.menu {
2   list-style: none;
3   margin: 0;
4   padding: 0;
5   width: 180px;
6 }
```

2. Eliminar las viñetas automáticas y todos los márgenes y espaciados aplicados por defecto.

```
1 ul.menu {
2   list-style: none;
3   margin: 0;
4   padding: 0;
5   width: 180px;
6 }
```

3. Añadir un borde al menú de navegación y establecer el color de fondo y los bordes de cada elemento del menú.

```
1 ul.menu {
2   border: 1px solid #7C7C7C;
3   border-bottom: none;
4   list-style: none;
5   margin: 0;
6   padding: 0;
7   width: 180px;
8 }
9 ul.menu li {
10  background: #F4F4F4;
11  border-bottom: 1px solid #7C7C7C;
12  border-top: 1px solid #FFF;
13 }
```

4. Aplicar estilos a los enlaces: mostrarlos como un elemento de bloque para que ocupen todo el espacio de cada del menú, añadir un espacio de relleno y modificar los colores y la decoración por defecto.

```

1 ul.menu li a {
2   color: #333;
3   display: block;
4   padding: .2em 0 .2em .5em;
5   text-decoration: none;
6 }

```

Ejercicio 3.23 A partir del menú anterior, añadir las reglas CSS necesarias para:

- Modificar el elemento de lista a un icono el cual cambiará cuando el ratón se encuentre sobre el elemento de la lista.
- El color de fondo del elemento cambiará cuando el ratón esté sobre el elemento de la lista.

Menú horizontal

En esta sección se va a construir un menú horizontal como el mostrado en la Figura 3.21. Se hará uso del mismo documento HTML utilizado para el menú vertical se van a aplicar las siguientes reglas CSS que van a permitir construir un menú horizontal:

1. Aplicar los estilos CSS básicos para establecer el estilo del menú (similares a los del menú vertical anterior):

```

1 ul.menu {
2   background: #F4F4F4;
3   border: 1px solid #7C7C7C;
4   list-style: none;
5   margin: 0;
6   padding: 0;
7 }
8 ul.menu li a {
9   color: #333;
10  display: block;
11  padding: .3em;
12  text-decoration: none;
13 }

```

2. Establecer la anchura de los elementos del menú. Como el menú es de anchura variable y contiene cinco elementos, se asigna una anchura del 20% a cada elemento. Si se quiere tener un control más preciso sobre el aspecto de cada elemento, es necesario asignar una anchura fija al menú. Además, se posiciona de forma flotante los elementos de la lista mediante la propiedad *float*. Esta es la clave de la transformación de una lista en un menú horizontal.

```

1 ul.menu li {
2   float: left;
3   width: 20%;
4 }

```

3. Después de posicionar de forma flotante a todos los elementos de la lista, el elemento es un elemento vacío ya que en su interior no existe ningún elemento posicionado de forma normal. Como ya se explicó en las secciones anteriores, la solución de este

problema consiste en aplicar la propiedad *overflow: hidden*; al elemento ``, de forma que encierre a todos los elementos posicionados de forma flotante.

```
1 ul.menu {
2   overflow: hidden;
3 }
```

4. Establecer los bordes de los elementos que forman el menú.

```
1 ul.menu li a {
2   border-left: 1px solid #FFF;
3   border-right: 1px solid #FFF;
4   border-bottom: 1px solid #7C7C7C;
5   color: #333;
6   display: block;
7   padding: .3em;
8   text-decoration: none;
9 }
```

5. Por último, se debe eliminar el borde derecho del último elemento de la lista, para evitar el borde duplicado. Esto se deja al lector para deducir el atributo y cómo aplicarlo.

Ejercicio 3.24 Agregar el mismo efecto del botón y cambio de color de fondo al pasar el ratón sobre cada elemento de la lista, del mismo modo que se hizo para el menú vertical.

Ejercicio 3.25 Desde nuestra plantilla HTML5/CSS3 se deben adaptar 2 sistemas de navegación de los descritos en la siguiente Web: <http://www.alvit.de/css-showcase/css-navigation-techniques-showcase.php>. Recuerde que debe adaptarlos a HTML5/CSS3.

3.8 Propiedades de tablas

El aspecto de las tablas es otro punto importante en la maquetación de sitios Web. Para controlar el aspecto de las tablas existen un conjunto de propiedades específicas se describen a continuación:

- **caption-side.** Esta propiedad permite especificar dónde se situará el título (*caption*) de la tabla. Los valores que se pueden asignar son *top*, *bottom*, *left* y *right*.
- **empty-cells.** Esta propiedad permite dibujar las celdas que están vacías, puesto que en principio si no tienen ningún contenido éstas no se mostrarán. Los valores posibles son *show* para mostrar los bordes y fondos de la celda y *hide* para ocultar tanto los bordes como el fondo.
- **border-collapse.** Permite establecer el modo en el que se dibujan los bordes de las tablas. Los valores posibles son *separate* y *collapse*. En el modo *separate*, cada celda está rodeada por su borde haciendo el efecto de un borde con una línea doble, mientras que, en el modo *collapse* las celdas contiguas comparten sus bordes. Es decir, el método *collapse* fusiona los bordes de las celdas adyacentes. El valor por defecto es *separate*.
- **border-spacing.** Permite establecer la separación entre celdas contiguas. Para hacerlos se indica el valor del espacio horizontal seguido del valor del espaciado vertical. Si se escribe un único valor, la separación horizontal y vertical serán iguales.
- **table-layout.** Permite definir el modo en el que el navegador dibujará la tabla ya que puede hacerse de dos formas. Los valores que admite son: *fixed* el cual dibuja la tabla basándose en las medidas establecidas en el código fuente. Con este valor se consigue que

Employee	Salary	Bonus	Supervisor
Stephen C. Cox	\$300	\$50	Bob
Josephin Tan	\$150	-	Annie
Joyce Ming	\$200	\$35	Andy
James A. Pentel	\$175	\$25	Annie

Figura 3.22: CSS3: Tabla tras aplicar estilos.

el sistema trabaje más rápido y *auto* el cual dibuja la tabla basándose en el contenido de sus celdas, es el valor por defecto.

A continuación se va a describir paso a paso un ejemplo de construcción de una tabla con estilos. En la Figura 3.22 se muestra el resultado de la tabla que se construirá. Para ello, lo inicial es construir la estructura de la tabla, en nuestro ejemplo se va a construir una tabla con 4 columnas: *Employee*, *Salary*, *Bonus* y *Supervisor*.

```

1 <table id="hor-minimalist-b" summary="Employee_Pay_Sheet">
2   <thead>
3     <tr>
4       <th scope="col">Employee</th>
5       <th scope="col">Salary</th>
6       <th scope="col">Bonus</th>
7       <th scope="col">Supervisor</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>Stephen C. Cox</td>
13      <td>$300</td>
14      <td>$50</td>
15      <td>Bob</td>
16    </tr>
17    <tr>
18      <td>Josephin Tan</td>
19      <td>$150</td>
20      <td>-</td>
21      <td>Annie</td>
22    </tr>
23    <tr>
24      <td>Joyce Ming</td>
25      <td>$200</td>
26      <td>$35</td>
27      <td>Andy</td>
28    </tr>
29    <tr>
30      <td>James A. Pentel</td>
31      <td>$175</td>
32      <td>$25</td>
33      <td>Annie</td>
34    </tr>
35  </tbody>
36 </table>

```

Comedy	Adventure	Action	Children
Scary Movie	Indiana Jones	The Punisher	Wall-E
Epic Movie	Star Wars	Bad Boys	Madagascar
Spartan	LOTR	Die Hard	Finding Nemo
Dr. Dolittle	The Mummy	300	A Bug's Life

(a)

Employee	Salary	Bonus	Supervisor
Stephen C. Cox	\$300	\$50	Bob
Josephin Tan	\$150	-	Annie
Joyce Ming	\$200	\$35	Andy
James A. Pentel	\$175	\$25	Annie

(b)

Comedy	Adventure	Action	Children
Scary Movie	Indiana Jones	The Punisher	Wall-E
Epic Movie	Star Wars	Bad Boys	Madagascar
Spartan	LOTR	Die Hard	Finding Nemo
Dr. Dolittle	The Mummy	300	A Bug's Life

(c)

Employee	Salary	Bonus	Supervisor
Stephen C. Cox	\$300	\$50	Bob
Josephin Tan	\$150	-	Annie
Joyce Ming	\$200	\$35	Andy
James A. Pentel	\$175	\$25	Annie

(d)

Figura 3.23: CSS3: Ejercicios de tablas.

Comedy	Adventure	Action	Children
Scary Movie	Indiana Jones	The Punisher	Wall-E
Epic Movie	Star Wars	Bad Boys	Madagascar
Spartan	LOTR	Die Hard	Finding Nemo
Dr. Dolittle	The Mummy	300	A Bug's Life

(a)

Company	Q1	Q2	Q3	Q4
Microsoft	20.3	30.5	23.5	40.3
Google	50.2	40.63	45.23	39.3
Apple	25.4	30.2	33.3	36.7
IBM	20.4	15.6	22.3	29.3

(b)

Figura 3.24: CSS3: Ejercicios de tablas (II).

El primer paso consiste en definir el tipo de fuente, fondo, márgenes, alineación y grosor del borde de la tabla de un modo general. De hecho, todas las propiedades que se utilizan son conocidas por el lector de secciones anteriores.

```

1 #hor-minimalist-b{
2   font-family: "Lucida_Sans_Unicode", "Lucida_Grande", Sans-Serif;
3   font-size: 12px;
4   background: #fff;
5   margin: 45px;
6   width: 480px;
7   border-collapse: collapse;
8   text-align: left;
9 }
```

Una vez definida los estilos genéricos de nuestra tabla se puede observar que la cabecera tiene un tamaño mayor, el estilo de la letra es negrita, el color es ligeramente diferente y que tiene detalles como el padding y el borde inferior diferente. Esto es fácilmente configurable como se muestra en el siguiente código.

```

1 #hor-minimalist-b th {
2   font-size: 14px;
3   font-weight: normal;
4   color: #039;
5   padding: 10px 8px;
6   border-bottom: 2px solid #6678b1;
7 }

```

Del mismo modo que para la cabecera, se va a especificar las propiedades de cada una de las celdas, es decir, el borde inferior, el color de las celdas y el *padding*.

```

1 #hor-minimalist-b td{
2   border-bottom: 1px solid #ccc;
3   color: #669;
4   padding: 6px 8px;
5 }

```

Finalmente, se puede observar que cuando el cursor se encuentra sobre una determinada celda ésta cambia de color a un tono más oscuro, esto se consigue aplicando un estilo concreto con la pseudo-clase *hover*.

```

1 #hor-minimalist-b tbody tr:hover td{
2   color: #009;
3 }

```

Ejercicio 3.26 Construya los ficheros HTML y CSS necesarios para conseguir los resultados de las tablas mostradas en las Figuras 3.23. y 3.24. ■

3.9 Novedades de CSS3

La nueva especificación de CSS3 no nace de cero desechando todo lo existente. En realidad es una expansión de la especificación 2.1 en el cual se han incorporado características relacionadas con la interacción y con carencias en el diseño de la anterior especificación. Esta especificación ha sido desarrollada conjuntamente con la especificación de HTML5 y es por ello que constantemente aparece la relación HTML5 y CSS3. Las nuevas características de CSS3 se engloban en los siguientes apartados:

- **Nuevos selectores.** Para poder hacer una selección más adecuada de los elementos sobre los cuales se aplican las reglas CSS.
- **Bordes y fondos.** Se resuelven problemas de manipulación de bordes y fondos que han arrastrado los diseñadores desde hace años.
- **Tratamiento de texto avanzado.** Permitiendo características avanzadas como es la selección de cualquier tipografía, multi-columnas o sombras en los textos sin tener que recurrir a códigos JavaScript o imágenes.
- **Colores RGBA, HSL y HSLA.** Los colores RGBA, HSL y HSLA han sido presentados en la sección anterior aunque son novedades incorporadas en CSS3.
- **Transformaciones y Animaciones en CSS3.** Tanto las transformaciones como las animaciones serán presentadas en el Capítulo 4 en el cual se presentará la incorporación de multimedia y animaciones en la Web.

En CSS 3 existen etiquetas nuevas que cada navegador interpreta de forma diferente al no haber un estándar por el que se tengan que regir. Es por ello que por ejemplo, que a la hora de hacer un degradado, o con algunas de las etiquetas nuevas, tengamos que usar diferentes prefijos.

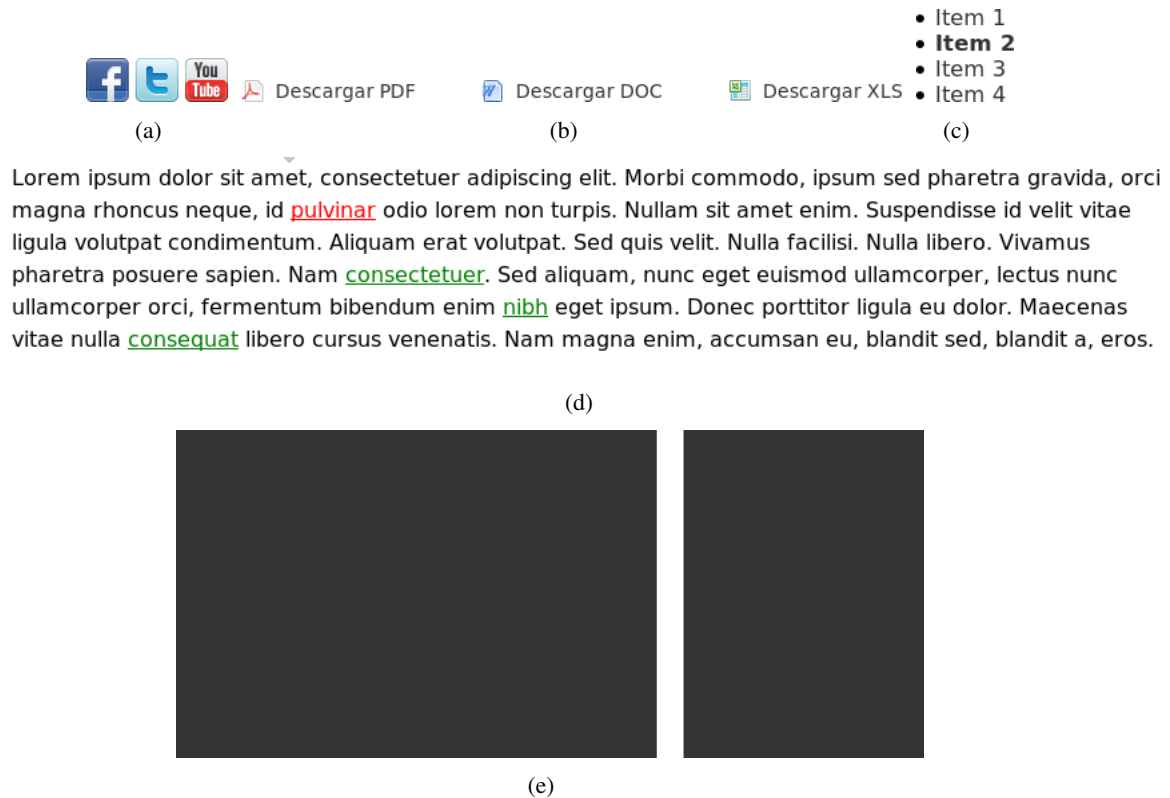


Figura 3.25: Nuevos selectores CSS3: (a)

Estos prefijos nuevos son los siguientes:

- **-webkit.** Chrome, Safari. Por ejemplo `-moz-linear-gradient`.
- **-moz.** Firefox. Por ejemplo `-webkit-gradient`.
- **-ms.** Microsoft. Por ejemplo `-o-linear-gradient`.
- **-o.** Opera. Por ejemplo `-ms-linear-gradient`.
- **Estándar.** Según se vayan adaptando no hace falta ningún prefijo. Por ejemplo, `linear-gradient`.

3.9.1 Nuevos selectores

Los selectores son una pieza fundamental en las reglas CSS puesto que permiten establecer sobre qué elementos se aplicará dicha regla. Aunque con CSS 2.1 se pueden especificar perfectamente los diferentes elementos en la nueva especificación (CSS3) se han incorporado nuevos selectores que facilitan aún más la tarea de los diseñadores Web.

Así que al igual que se presentó en la primera sección de este capítulo se van a describir los nuevos selectores acompañado de ejemplos de aplicación:

- **[attr=valor].** Esta variación seleccionará a los elementos que tengan ese atributo y cuyo valor declarado comience con lo que se indica. Un uso práctico sería por ejemplo ponerles iconos a una serie de enlaces a redes sociales, donde detectamos a que red social esta enlazando (el enlace debe empezar con `http://face...`) y de acuerdo a ello ponemos el icono respectivo. El resultado se muestra en la Figura 3.25(a).

```
1 <a href="http://facebook.com/webtursos"></a>
2 <a href="http://twitter.com/webtursos"></a>
```

```
3 <a href="http://youtube.com/webtursos"></a>
```

```
1 a{
2   display:inline-block;
3   height:32px;
4   width:32px;
5 }
6 a[href^="http://facebook"]{
7   background:url(http://cdn4.iconfinder.com/data/icons/
8     vector_social_media_icons/32px/facebook.png);
9 }
10 a[href^="http://twitter"]{
11   background:url(http://cdn4.iconfinder.com/data/icons/
12     vector_social_media_icons/32px/twitter.png);
13 }
14 a[href^="http://youtube"]{
15   background:url(http://cdn1.iconfinder.com/data/icons/
16     vector_social_media_icons/32px/youtube.png);
17 }
```

- **[attr\$=valor]**. Este selector es lo invertido al anterior, seleccionará a los elementos que tengan ese atributo y cuyo valor declarado termine con lo que se indica. El uso mas común que se le da a este tipo de selector es para estilizar enlaces de descarga de archivos, en el cual la ruta termina en el formato de archivo (http://.../documento.doc), el cual usamos para hacer una selección individual. El resultado se muestra en la Figura 3.25(b).

```
1 <a href="archivo.pdf">Descargar PDF</a>
2 <a href="archivo.doc">Descargar DOC</a>
3 <a href="archivo.xls">Descargar XLS</a>
```

```
1 a{
2   display:inline-block;
3   text-decoration:none;
4   color:#333;
5   font-family:sans-serif;
6   padding:10px 25px;
7   font-size:14px;
8   border-radius:4px;
9   margin:10px;
10 }
11 a[href$=pdf]{
12   background:url(http://cdn1.iconfinder.com/data/icons/silk2/
13     page_white_acrobat.png) no-repeat left center;
14 }
15 a[href$=doc]{
16   background:url(http://cdn1.iconfinder.com/data/icons/silk2/
17     page_word.png) no-repeat left center;
18 }
19 a[href$=xls],a[href$=csv]{
20   background:url(http://cdn3.iconfinder.com/data/icons/fugue/
21     icon_shadowless/table-excel.png) no-repeat left center;
22 }
```

- **[attr*=valor]**. Este selector es el que tiene mas alcance, ya que como ven lleva el *, lo que indica que es global, va a seleccionar el elemento que tenga el valor indicado este donde este. Un ejemplo práctico es cuando trabajamos con los menús de WordPress y queremos aplicar estilo al elemento activo o current, pero WordPress nos añade un montón

de clases y al final no logramos lo cometido o terminamos con un CSS muy desordenado, para estos casos podemos usar el selector de atributo global *. El resultado se muestra en la Figura 3.25(c).

```
1 <ul>
2   <li><a href="#">Item 1</a></li>
3   <li class="menu-item_menu-item-type-taxonomy_menu-item-object-
      category_current-category-ancestor_current-menu-ancestor_
      current-menu-parent_current-category-parent"><a href="#">Item
      2</a></li>
4   <li><a href="#">Item 3</a></li>
5   <li><a href="#">Item 4</a></li>
6 </ul>
```

```
1 a{
2   text-decoration: none;
3   font-family: sans-serif;
4   color: #333;
5 }
6 ul li[class*="curr"] a{
7   font-weight: bold;
8 }
```

- **[attr =valor]**. Esta variación seleccionará a los elementos que tengan ese atributo y cuyo valor declarado este entre espacios. Esta tipo de selector se usa casi siempre para seleccionar elementos que tengan atributos con multiples valores, tales como el atributo rel de un a, en este ejemplo tendremos un par de parrafos en los cuales habran varios enlaces a los que les pondremos multiples valores rel, pero colorearemos de verde solo los que sean external. El resultado se muestra en la Figura 3.25(d).

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi
  commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id
  <a href="#">pulvinar</a> odio lorem non turpis. Nullam sit amet
  enim. Suspendisse id velit vitae ligula volutpat condimentum.
  Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla
  libero. Vivamus pharetra posuere sapien.
2
3 Nam <a href="#" rel="external">consectetur</a>. Sed aliquam, nunc
  eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum
  bibendum enim <a href="#" rel="nofollow_external">nibh</a> eget
  ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla <a
  href="#" rel="external_friend">consequat</a> libero cursus
  venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a,
  eros.
```

```
1 *{font: 16px/24px sans-serif;}
2 a{color: red;}
3 a[rel~="external"]{color: green;}
```

- **[attr|valor]**. Esta variación seleccionará a los elementos que tengan ese atributo y cuyo valor declarado este entre guiones. Para este selector haremos un ejemplo un poco mas complicado: lo usaremos para hacer una pequeña grilla CSS, en la cual las propiedades se aplicaran a las columnas solo por el hecho de que la clase de estas contenga la palabra grid-, seguido ira el numero que indique el ancho de esa columna. El resultado se muestra en la Figura 3.25(e).

```

1 <div class="wrap">
2   <div class="grid-8"></div>
3   <div class="grid-4"></div>
4 </div>

```

```

1 .wrap{
2   width:_560px;
3   margin:_20px_auto;
4 }
5 .wrap [class|="grid"]{
6   background:_#333;
7   height:_320px;
8   float:_left;
9   margin-right:_20px;
10 }
11 .wrap [class|="grid"]:last-child{margin-right:_0;}
12 .wrap .grid-4{width:_180px;}
13 .wrap .grid-8{width:_360px;}

```

3.9.2 Bordos y fondos

La manipulación de los bordes y fondos ha sido un quebradero de cabeza para los diseñadores Web durante muchos años. Se han utilizado toda clase de trucos para conseguir bordes redondeado o sombras a los mismos. Es con la llegada de CSS3 cuando se implementan propiedades propias de los bordes y fondos.

Bordes

Las novedades en la manipulación de bordes son tres nuevas propiedades que van a facilitar en gran medida la tarea de los diseñadores.

- **border-radius.** Esta propiedad permite establecer bordes redondeados del modelo de cajas. Se especifica cada una de las esquinas con el siguiente orden: esquina superior izquierda, esquina superior derecha, esquina inferior derecha y esquina inferior izquierda.

```

1 border-radius: 40px 60px 100px 100px;
2 -moz-border-radius: 40px 60px 100px 100px;
3 -webkit-border-radius: 40px 60px 100px 100px;

```

Además, existen propiedades específicas para cada uno de los bordes. Los siguientes códigos son equivalentes entre ellos.

```

1 border-radius:2em;
2
3 es equivalente a
4
5 border-top-left-radius:2em;
6 border-top-right-radius:2em;
7 border-bottom-right-radius:2em;
8 border-bottom-left-radius:2em;

```

Un ejemplo de aplicación de aplicar bordes redondeados es mostrado en la Figura 3.26 el cual corresponde al Código 3.14.

Código 3.14: CSS3: Ejemplo de uso de border-radius.

```

1 #Example_A {
2   height:_65px;

```

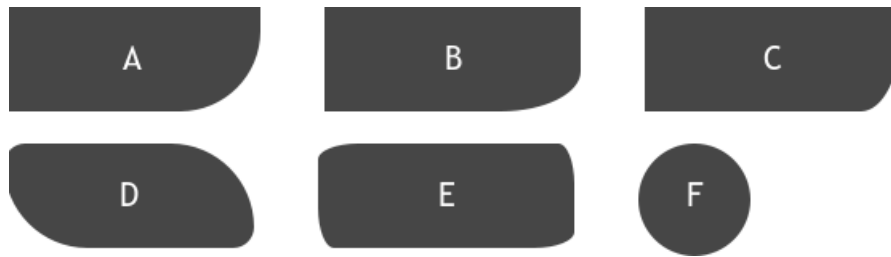


Figura 3.26: CSS3: Border-radius

```

3  width:160px;
4  -moz-border-radius-bottomright:50px;
5  border-bottom-right-radius:50px;
6  }
7  #Example_B {
8  height:65px;
9  width:160px;
10 -moz-border-radius-bottomright:50px_25px;
11 border-bottom-right-radius:50px_25px;
12 }
13 #Example_C {
14 height:65px;
15 width:160px;
16 -moz-border-radius-bottomright:25px_50px;
17 border-bottom-right-radius:25px_50px;
18 }
19 #Example_D {
20 height:5em;
21 width:12em;
22 -moz-border-radius:1em_4em_1em_4em;
23 border-radius:1em_4em_1em_4em;
24 }
25 #Example_E {
26 height:65px;
27 width:160px;
28 -moz-border-radius:25px_10px_10px_25px;
29 border-radius:25px_10px_10px_25px;
30 }
31 #Example_F {
32 height:70px;
33 width:70px;
34 -moz-border-radius:35px;
35 border-radius:35px;
36 }

```

- **box-shadow.** Esta propiedad permite definir sombras a los diferentes elementos de una página. Esta propiedad recibe algunos parámetros que permiten especificar la sombra, el difuminado, separación de la sombra y la caja y el color. La sintaxis sería la siguiente:

```

1 box-shadow "desplazamiento_horizontal_de_la_sombra" "desplazamiento_
   vertical_de_la_sombra" "difuminado" "color_de_la_sombra".
2 box-shadow: 5px -9px 3px #000;

```

- **Desplazamiento horizontal de la sombra:** La sombra de un elemento suele estar un poco desplazada con respecto al elemento que la produce y su posición será en función del ángulo con el que llegue la luz. En el caso de este ejemplo el primero de los valores, 5px, quiere decir que la sombra aparecerá 5 píxeles a la derecha. Si la



Figura 3.27: CSS3: Border-shadow (I)

sombra apareciera un poco hacia la izquierda del elemento original que la produce, aparecería un valor negativo a este atributo. Cuanto más desplazamiento tenga una sombra, el elemento que la produce parecerá que está más separado del lienzo de la página.

- **Desplazamiento vertical de la sombra:** El segundo valor es el desplazamiento vertical de la sombra con respecto a la posición del elemento que la produce. Este valor es similar al desplazamiento horizontal. Valores positivos indican que la sombra aparecerá hacia abajo del elemento y valores negativos harán que la sombra aparezca desplazada un poco hacia arriba. En el caso del anterior ejemplo, con `-9px` estamos indicando que la sombra aparecerá desplazada 9 píxeles hacia arriba del elemento.
- **Difuminado:** El tercer valor indica cuánto queremos que esté difuminado el borde de la sombra. Si el valor es mayor que cero, como en nuestro ejemplo `3px`, quiere decir que la sombra tendrá un difuminado de esa anchura, 3 píxeles en el ejemplo.
- **Color de la sombra:** El último atributo que se indica en el atributo `box-shadow` es el color de la sombra. Generalmente las sombras en el mundo real tienen un color negro o grisáceo, pero con CSS3 podremos indicar cualquier gama de color para hacer la sombra, lo que nos dará bastante más versatilidad a los diseños gracias a la posible utilización de sombras en distintos colores, que puedan combinar mejor con nuestra paleta. En el ejemplo anterior habíamos indicado una sombra con color negro.

En el código siguiente se muestra unos cuantos ejemplos de aplicación de sombras sobre un elemento. Los resultados se pueden visualizar en la Figura 3.27.

```

1 #Example_A {
2   -moz-box-shadow: -5px -5px #888;
3   -webkit-box-shadow: -5px -5px #888;
4   box-shadow: -5px -5px #888;
5 }
6 #Example_B {
7   -moz-box-shadow: -5px -5px 5px #888;
8   -webkit-box-shadow: -5px -5px 5px #888;
9   box-shadow: -5px -5px 5px #888;
10 }
11 #Example_C {
12   -moz-box-shadow: -5px -5px 0 5px #888;
13   -webkit-box-shadow: -5px -5px 0 5px #888;
14   box-shadow: -5px -5px 0 5px #888;
15 }
16 #Example_D {
17   -moz-box-shadow: -5px -5px 5px 5px #888;
18   -webkit-box-shadow: -5px -5px 5px 5px #888;
19   box-shadow: -5px -5px 5px 5px #888;
20 }

```



Figura 3.28: CSS3: Border-shadow (II)

```

21 #Example_E {
22   -moz-box-shadow: 0 0 5px #888;
23   -webkit-box-shadow: 0 0 5px #888;
24   box-shadow: 0 0 5px #888;
25 }
26 #Example_F {
27   -moz-box-shadow: 0 0 5px 5px #888;
28   -webkit-box-shadow: 0 0 5px 5px #888;
29   box-shadow: 0 0 5px 5px #888;
30 }

```

Además, se pueden construir sombras internas al elemento utilizando la palabra reservada *inset*. En el siguiente código se muestra un ejemplo de aplicación y los correspondientes resultados se pueden visualizar en la Figura 3.28.

```

1 #Example_G {
2   -moz-box-shadow: inset -5px -5px #888;
3   -webkit-box-shadow: inset -5px -5px #888;
4   box-shadow: inset -5px -5px #888;
5 }
6 #Example_H {
7   -moz-box-shadow: inset -5px -5px 5px #888;
8   -webkit-box-shadow: inset -5px -5px 5px #888;
9   box-shadow: inset -5px -5px 5px #888;
10 }
11 #Example_I {
12   -moz-box-shadow: inset -5px -5px 0 5px #888;
13   -webkit-box-shadow: inset -5px -5px 0 5px #888;
14   box-shadow: inset -5px -5px 0 5px #888;
15 }
16 #Example_J {
17   -moz-box-shadow: inset -5px -5px 5px 5px #888;
18   -webkit-box-shadow: inset -5px -5px 5px 5px #888;
19   box-shadow: inset -5px -5px 5px 5px #888;
20 }
21 #Example_K {
22   -moz-box-shadow: inset 0 0 5px #888;
23   -webkit-box-shadow: inset 0 0 5px #888;
24   box-shadow: inner 0 0 5px #888;
25 }
26 #Example_L {
27   -moz-box-shadow: inset 0 0 5px 5px #888;
28   -webkit-box-shadow: inset 0 0 5px 5px #888;
29   box-shadow: inset 0 0 5px 5px #888;
30 }

```

Son muchas las características de las sombras de CSS3. Una de ellas es permitir las múltiples capas de sombras como se visualiza en la Figura 3.29 y en el siguiente código.

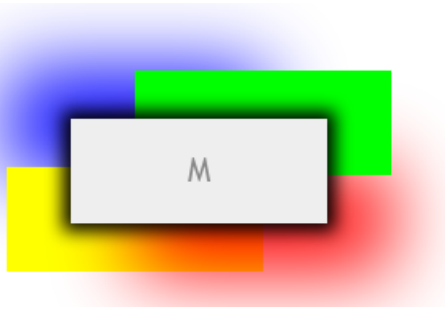


Figura 3.29: CSS3: Border-shadow (III)

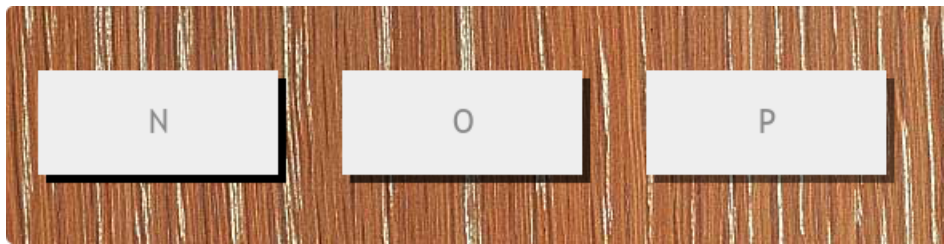


Figura 3.30: CSS3: Border-shadow (IV)

```

1 #Example_M {
2   -moz-box-shadow: 0 0 10px 5px black, 40px -30px lime, 40px 30px 50
      px red, -40px 30px yellow, -40px -30px 50px blue;
3   -webkit-box-shadow: 0 0 10px 5px black, 40px -30px lime, 40px 30px
      50px red, -40px 30px yellow, -40px -30px 50px blue;
4   box-shadow: 0 0 10px 5px black, 40px -30px lime, 40px 30px 50px
      red, -40px 30px yellow, -40px -30px 50px blue;
5 }

```

Si se combinan las sombras con características ya vistas como son los colores en el canal alfa o con los bordes redondeados se pueden ganar incluso más realismo tal y como se muestran en las imágenes de las Figuras 3.30 y 3.31.

```

1 #Example_N {
2   -moz-box-shadow: 5px 5px rgb(0,0,0);
3   -webkit-box-shadow: 5px 5px rgb(0,0,0);
4   box-shadow: 5px 5px rgb(0,0,0);
5 }
6 #Example_O {
7   -moz-box-shadow: 5px 5px rgba(0,0,0,0.7);
8   -webkit-box-shadow: 5px 5px rgba(0,0,0,0.7);
9   box-shadow: 5px 5px rgba(0,0,0,0.7);
10 }
11 #Example_P {
12   -moz-box-shadow: 5px 5px rgba(0,0,0,0.5);
13   -webkit-box-shadow: 5px 5px rgba(0,0,0,0.5);

```



Figura 3.31: CSS3: Border-shadow (V)

```
14 box-shadow: 5px 5px rgba(0,0,0,0.5);
15 }
```

```
1 #Example_Q {
2   -moz-border-radius: 5px;
3   border-radius: 5px;
4   -moz-box-shadow: 5px 5px black;
5   -webkit-box-shadow: 5px 5px black;
6   box-shadow: 5px 5px black;
7 }
8 #Example_R {
9   -moz-border-radius: 5px;
10  border-radius: 5px;
11  -moz-box-shadow: 5px 5px 5px black;
12  -webkit-box-shadow: 5px 5px 5px black;
13  box-shadow: 5px 5px 5px black;
14 }
```

- **border-image.** Esta propiedad añade un borde gráfico con el soporte de una imagen a cualquier tipo de elemento. Su sintaxis es la siguiente.

```
1 border-image: _url(dirección_imagen) _recorte_repetición;
```

Los parámetros de esta propiedad se pueden resumir en los siguientes:

- **dirección_imagen:** La dirección de la imagen que servirá como marco; la imagen de fondo.
- **recorte:** La porción de imagen de fondo que tomaremos en píxeles, medida desde sus extremos.
- **repetición:** La manera en que va a ser tratada la imagen de fondo para rellenar el espacio reservado para el borde cuando las medidas de esta y de la imagen principal no coincidan; a priori, casi siempre. Puede tomar estos cuatro valores:
 - **stretch:** Se escala la imagen de fondo (deformándose).
 - **repeat:** Se repite la imagen de fondo igual que hace background.
 - **round:** Idem pero si con un número entero de repeticiones no se rellena del todo, los trozos se redimensionarán.
 - **space:** Idem pero si con un número entero de repeticiones no se rellena del todo, se aumentarán los espacios, no deformando la imagen de fondo.

En la Figura 3.32 se muestra un borde de forma de marco que se puede aplicar a un elemento <figure>. El siguiente código muestra la utilización de la nueva propiedad.

```
1 .borderimg1 img {
2   border: _60px_solid_black;
3   border-image: _url(./img/marco.png) _36_stretch;
4 }
```

Ejercicio 3.27 Utilizando las reglas CSS3 necesarias construye los elementos mostrados en la Figura 3.33.

Ejercicio 3.28 Utilizando la herramienta online: <http://thany.nl/apps/BoxShadows/> para conseguir configurar los mismos bordes que el ejercicio anterior y además trata de crear dos bordes innovadores (que se adapten bien a tu diseño).



Figura 3.32: CSS3: Aplicación de la propiedad *border-image*.



Figura 3.33: CSS3: Ejercicio de aplicación de bordes.

Ejercicio 3.29 Construye un borde con partes de imágenes utilizando la propiedad *border-image*. Una vez hecho el efecto, prueba a utilizar la herramienta online: <http://border-image.com> con la cual permite construir bordes utilizando la propiedad descrita anteriormente.

Ejercicio 3.30 Estudia como funcionan las siguientes propiedades y plantea ejemplos prácticos para las siguientes propiedades: *border-image-source*, *border-image-slice*, *border-image-width*, *border-image-outset* y *border-image-repeat*.

Fondos

En el tratamiento de los fondos CSS3 incorpora tres nuevas propiedades que se describen a continuación:

- **background-size.** Esta propiedad se puede ajustar o escalar las imágenes utilizadas como fondo en los elementos HTML. En principio se puede establecer el ancho y alto, o permitir que el sistema adapte el fondo a la resolución de la pantalla. En el siguiente código se establece un tamaño de 400x635 píxeles para el tamaño del fondo establecido.

```
1 background-size: 400px 635px;
```

Además, se puede definir con porcentaje o palabras reservadas como son *cover* o *contain*.

- **Porcentaje:** Establece el porcentaje que ocupará el fondo en función del elemento padre donde se está estableciendo el fondo.
- **Cover:** Escala la imagen para el tamaño más pequeño de tal manera que su anchura y su altura puede caber dentro del área de contenido.

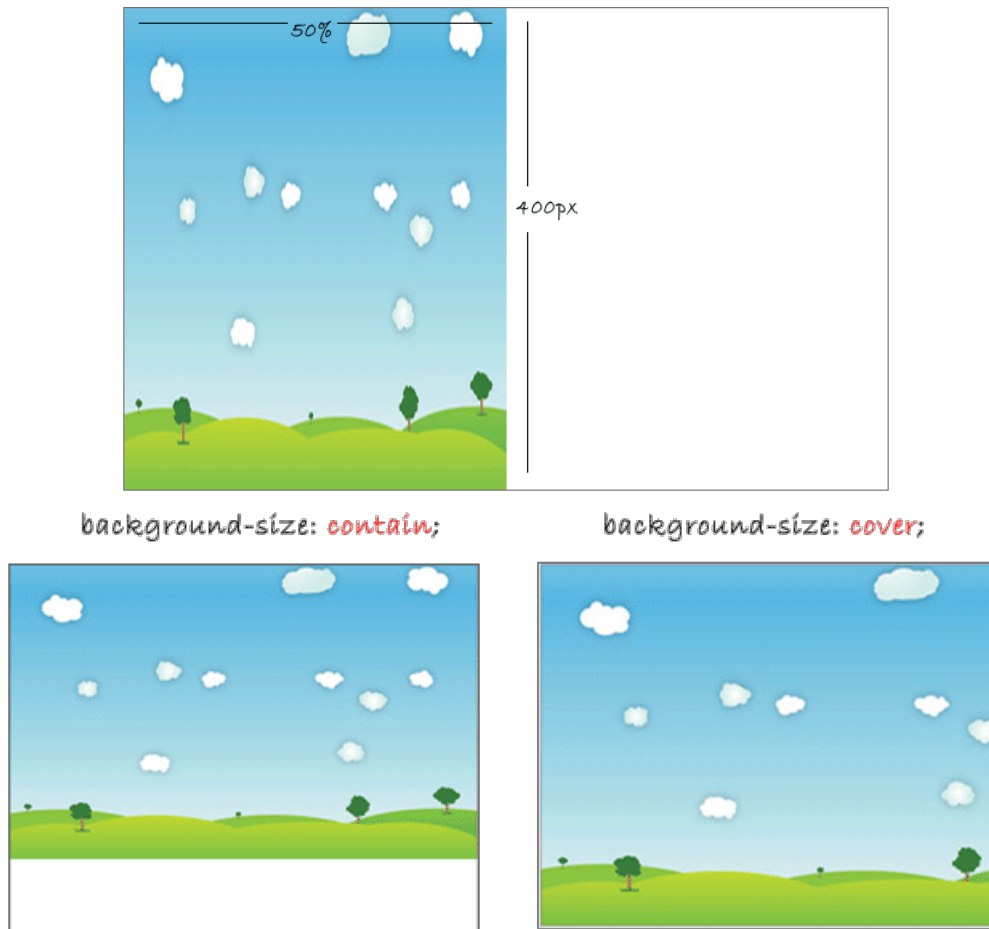


Figura 3.34: CSS3: Background-size.

- **Contain:** Escala la imagen al tamaño más grande de tal manera que su anchura y su altura puede caber dentro del área de contenido.

A continuación se muestra un ejemplo de aplicación de background-size. En la Figura 3.34 se muestran varias imágenes de aplicar el código anterior y de utilizar las palabras reservadas *cover* y *contain*.

```

1 div{
2     height:400px;
3     width:_635px;
4     background:url(paisaje.jpg)_0_0_no-repeat;
5     background-size:_50%_400px;
6     border:_1px_solid_#777;
7 }

```

- **background-origin.** Esta propiedad permite posicionar el fondo en relación del contenido o área de la caja contenedora. Este elemento puede ser relativa al *padding* de la caja utilizando el valor *padding-box*. Además, puede ser relativa al borde de la caja utilizando el valor *border-box* y finalmente, se puede ubicar relativamente al contenido de la caja utilizando el valor *content-box*. En el siguiente código se muestra un ejemplo de utilización de esta nueva propiedad y en la Figura 3.35 las diferentes opciones de configuración.

```

1 div{
2     height:300px;
3     width:_435px;

```



Figura 3.35: CSS3: Background-origin.

```

4  background:url(sol.png)_0_0_no-repeat;
5  background-origin:_content-box;
6  border:_5px_solid_#777;
7  padding:_15px;
8  }

```

- **background-clip.** Es una de las nuevas propiedades que permite recortar los fondos. Se pueden definir tres valores posibles:

- **border-box:** Con este valor el fondo se pega al inicio del borde de la caja (esta es la opción por defecto). Básicamente el navegador hace lo mismo que siempre, ocupar todo el ancho y alto de una contenedor. En el siguiente ejemplo se utiliza un color de fondo rojo y y el borde de color blanco y punteado para que pueda apreciarse la colocación del mismo al inicio del borde la caja (Figura 3.36(a)).

```

1  .fondonormal{
2      background-clip: border-box;
3      background-color: red;
4      border: 20px dotted white;
5  }

```

- **padding-box:** Con este valor el fondo inicia después del ancho del borde. Este valor lo único que hace, es que si se define un borde con CSS (`border`) en el contenedor, el fondo inicie con él después del grueso del mismo `border-width`, esto no lo podemos apreciar cuando usamos una línea solida, sino de puntos (*dotted*). En el ejemplo de

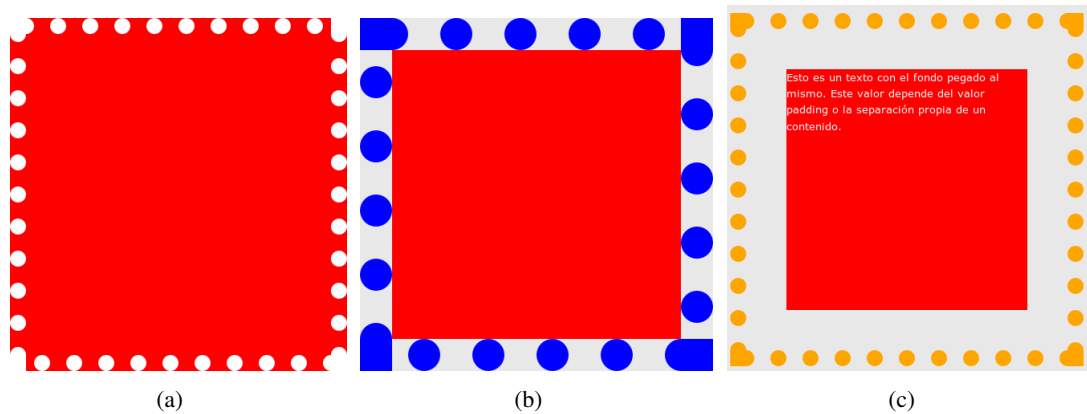


Figura 3.36: CSS3: Background-clip.

la Figura 3.36(b) se puede apreciar que la línea azul es el borde pero en el fondo empieza después dejando espacio sin fondo de color.

```
1 .fondoalborde{
2   background-clip: padding-box;
3   background-color: red;
4   border: 40px dotted blue; /* ojo, esta regla está resumida */
5 }
```

- **content-box:** Con este valor el fondo rodea al contenido con un *padding* asignado. Esta propiedad hace que el borde se genere donde el contenido está puesto, si es un texto, este seguirá las dimensiones asignado a los límites de ese texto, pero no es lo mismo que cuando se asigna como fondo a un texto, en el que un fondo de color solo rellena el contorno del texto en sí, pues en este caso se ocupa todo el ancho y alto disponible para el mismo. Si usan la propiedad *Padding* de espaciado de relleno, este definirá el inicio de background respecto al borde del contenedor. En el ejemplo de la Figura 3.36(c) se muestra un ejemplo de aplicación de esta propiedad.

```
1 .fondo a la caja{
2   background-clip: content-box;
3   background-color: red;
4   padding: 50px;
5 }
```

- **Uso de varios fondos en capas.** Una de las características más interesantes en el tratamiento de varios fondos en diferentes capas. En versiones anteriores había que componer las imágenes de fondo en una sola imagen. La sintaxis es bastante simple, puesto que consiste en especificar utilizando la función *url* la lista de imágenes que se quieren utilizar de fondo.

```
1 background-image:
2   url(sol.jpg), url(vaca.png), url(paisaje.jpg);
```

Utilizando esta nueva característica se pueden crear composiciones compuestas por varias imágenes y ubicarlas en diferentes posiciones para lograr lo deseado. En el siguiente ejemplo se utilizan 3 imágenes para componer un paisaje en un elemento `<div>`. El resultado se muestra en la Figura 3.37

```
1 div{
2   height: 400px;
```

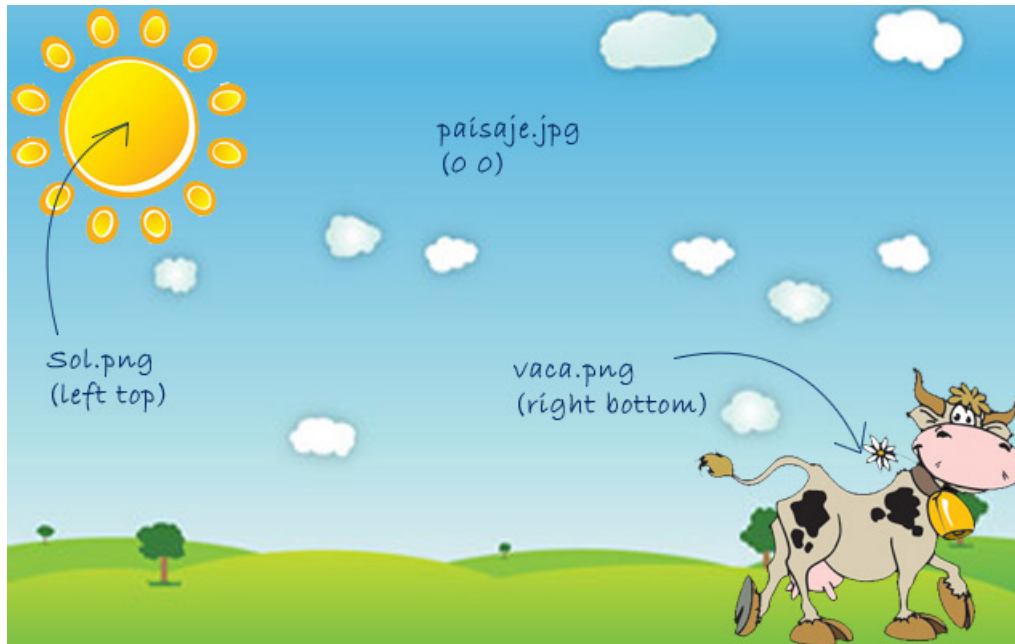


Figura 3.37: CSS3: Varios fondos.

```

3  width: 635px;
4  background-image: url(sol.png), url(vaca.png), url(paisaje.jpg);
5  background-repeat: no-repeat;
6  background-position: left top, right bottom, 0 0;
7  }
8
9  es equivalente a
10
11 div{
12     background:
13         url(sol.png) left top no-repeat,
14         url(vaca.png) right bottom no-repeat,
15         url(paisaje.jpg) 0 0 no-repeat;
16 }

```

Ejercicio 3.31 Para nuestro proyecto HTML/CSS que se ha ido realizando a lo largo del Capítulo debe incorporar una imagen de cabecera como fondo y sustituir los fondos de algunas secciones por imágenes utilizando las nuevas propiedades presentadas.

Ejercicio 3.32 Debe componer un paisaje de varias imágenes utilizando la composición de varios fondos en una sola imagen. Debe buscar las imágenes que crea más adecuada para su implementación y realizar el código HTML y CSS.

3.9.3 Fuentes

El tratamiento de las fuentes es uno de los puntos fuerte de CSS3. CSS3 permite mostrar cualquier tipografía independientemente de que el visitante tenga instalada en su máquina o no la tipografía. Además, esta característica no es innovadora en CSS3 sino que ya existía en el año 1998 en la especificación de CSS2 e incluso Microsoft lo hacía posible en su navegador Internet Explorer 4 pero esta característica fue desechada en la especificación de CSS2.1 y quedó relegada a tener una mínima cobertura entre los desarrolladores. No fue así por los navegadores

Web, a continuación se describen algunos soportes de ficheros y versiones de navegadores:

- **Internet Explorer:** Desde el IE4 soporta Embedded OpenType (EOT).
- **Mozilla/firefox:** Desde Firefox 3.5 soporta TrueType/OpenType TT (TTF), OpenType PS (OTF) y WOFF(Desde Firefox 3.6).
- **Google Chrome:** Desde la versión 4 soporta TrueType/OpenType TT(TTF) y OpenType PS (OTF).
- **Opera:** Desde la versión 10 y la 9.7 en el móvil soportan TrueType/OpenType TT (TTF), OpenType PS (OTF) y SVG.
- **Safari/Webkit:** Desde la versión 3.1 soporta TrueType/OpenType TT (TTF), OpenType PS (OTF); SafariMobile (iPhone/iPad) sólo SVG.
- **Netscape:** Desde la versión 4 solamente soporta Portable Font Resource (PFR).
- **Móviles:** Safari es compatible con SVG WebFonts a partir del iOS 3.1 y Android soporta otf/ttf a partir de la versión 2.2. Firefox móvil en Android soporta WOFF desde la versión 5. Webkit presenta problemas, pues no muestra el texto (es invisible) hasta no haber descargado los archivos de las fuentes. El iPhone y el iPad sólo co el formato SVG.

El uso de cualquier tipografía es establecido utilizando la regla **@font-face**. Lo primero que se debe hacer es descargar los ficheros de las tipografías a utilizar. La regla @font-face sigue la sintaxis expuesta en el Código 3.15.

Código 3.15: CSS3: Ejemplo de uso de @font-face.

```

1 @font-face{
2   font-family: 'CrimsonRoman';
3   src: url('Crimson-Roman.eot');
4   src: url('Crimson-Roman.eot?#iefix') format('embedded-opentype'),
5         url('Crimson-Roman.woff') format('woff'),
6         url('Crimson-Roman.ttf') format('truetype'),
7         url('Crimson-Roman.svg#CrimsonRoman') format('svg');
8   font-weight: normal;
9   font-style: normal;
10 }
```

La explicación de la regla anterior es la siguiente. En primer lugar aparece *font-family* que es el nombre de la fuente. Esta declaración se debe incluir en aquellos elementos que se quiera utilizar la tipografía. En nuestro ejemplo se ha tomado como nombre *CrimsonRoman*. A continuación se le indica al navegador dónde encontrar el archivo de la tipografía. Se puede observar que en primer lugar aparece la ruta de la fuente en formato *eot* (válido para el navegador Internet Explorer). Además, se incorpora una versión que corrige un bug de IE9. Las siguientes url son para indicarle a cada navegador la ruta del archivo de la fuente que maneja cada uno. Finalmente se establecen las propiedades *font-weight* y *font-style* ya estudiadas previamente. Estas propiedades se establecen para que los navegadores no modifiquen el aspecto de la tipografía (aunque la mayoría de los navegadores hacen uso de estas especificaciones no se garantiza para todos los navegadores).

Una vez que han sido establecidos los archivos de las fuentes y la regla @font-face se deben aplicar la tipografía sobre los diferentes elementos. En el siguiente ejemplo se va a utilizar la fuente descrita para los títulos h1. Se definen más tipografías por si por alguna circunstancia es imposible cargar la fuente.

```

1 h1 { font-family: _CrimsonRoman, _Georgia, _Garamond, _serif; }
```

En la Figura 3.38 se muestra la aplicación de fuentes propias en una Web personal.

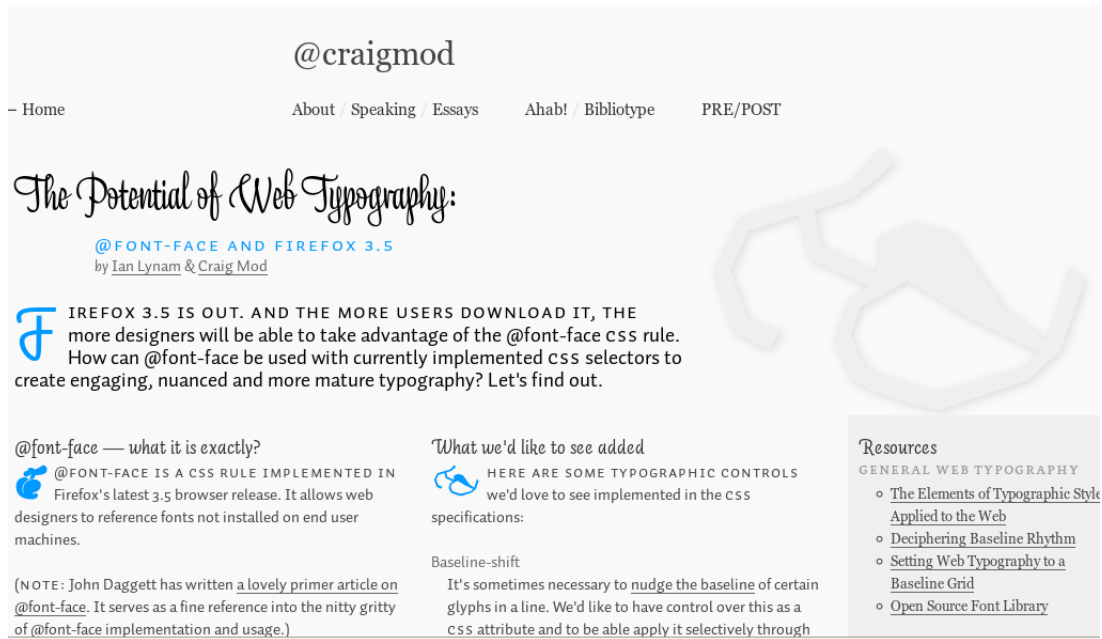


Figura 3.38: CSS3: @font-face.

Ejercicio 3.33 Busca fuentes libres para descargar y construye en tu sitio Web la aplicación de al menos 3 fuentes personalizadas. Recuerda que debes establecer fuentes seguras en caso de que no se cargue satisfactoriamente tu fuente personalizada.

Ejercicio 3.34 Google Fonts (<http://www.google.com/fonts>) tiene disponible un servicio libre y gratuito para incluir tipografías en las páginas. Google Fonts permite utilizar fuentes remotas a través de sus servicios (al margen de descargarlas). Utiliza fuentes de Google fonts en tu proyecto Web.

Ejercicio 3.35 Utilice la Web <http://www.fontsquirrel.com/> para descargar un conjunto de fuentes que crea interesantes (al menos 3) e indique en qué zonas de una Web y para qué tipo de Web las utilizaría.

Ejercicio 3.36 Utilice la Web anterior para generar reglas @font-face para ser utilizada en los casos anteriores.

Ejercicio 3.37 Busca un método para convertir una tipografía (un formato dato) en cualquier otro formato. Puedes usar herramientas online o *stand-alone*.

3.9.4 Efectos de textos

Text-shadow

Otra interesante característica de CSS3 es la de poder añadir una sombra a cada letra de un texto sin tener que recurrir a imágenes o código JavaScript. La sintaxis más sencilla es la siguiente:

```
1 h3 {text-shadow: 0.1em 0.1em #333;}
```

Esto añade una sombra gris oscura (#333) desplazada a la derecha (0.1em) y abajo (0.1em) en relación con el texto normal. El resultado se muestra en la Figura 3.39(a).

Además, también se puede incorporar borrosidad a la sombra. Esto se consigue añadiendo un nuevo valor a la propiedad entre el color y el desplazamiento abajo. En el siguiente código se

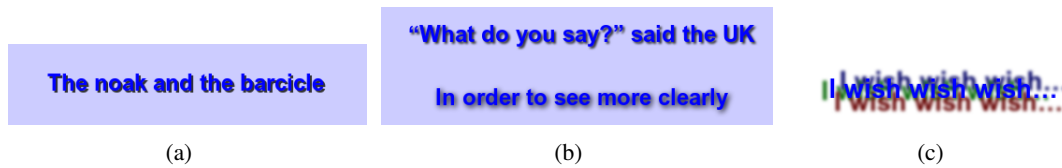


Figura 3.39: CSS3: Text-shadow.

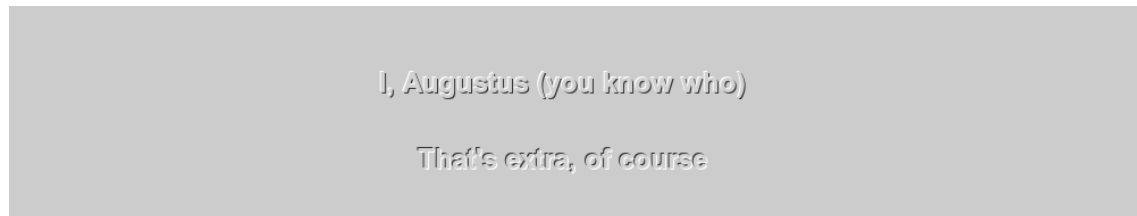


Figura 3.40: CSS3: Text-shadow (II).

muestran dos líneas, una con un poco de imprecisión (0.05em) y la otra con mucha imprecisión (0.2em). El resultado se muestra en la Figura 3.39(b).

```
1 h3.a {text-shadow: 0.1em 0.1em 0.05em #333;}
2 h3.b {text-shadow: 0.1em 0.1em 0.2em black;}
```

Se pueden lograr efectos más interesantes agregando más de una sombra. Esto se consigue indicando cada una de las sombras separadas por una coma. El resultado se muestra en la Figura 3.39(c).

```
1 h3 {text-shadow: 0.2em 0.5em 0.1em #600,
2      -0.3em 0.1em 0.1em #060,
3      0.4em -0.3em 0.1em #006}
```

El efecto no es muy adecuado, pero si se combinan dos sombras pueden obtener satisfactorios como el mostrado en el siguiente código (Figura 3.41).

```
1 h3.a {text-shadow: -1px -1px white, 1px 1px #333;}
2 h3.b {text-shadow: 1px 1px white, -1px -1px #333;}
```

Con esta idea básica se pueden conseguir efectos tan interesantes como el mostrado en la Figura 3.41. El código correspondiente a estos efectos es el siguiente.

```
1 #vintage h1 {
2   text-align: center;
3   margin: 200px auto;
4   font-family: "League-Gothic", Courier;
5   font-size: 200px; text-transform: uppercase;
6   color: #707070;
7   text-shadow: 5px 5px 0px #eee, 7px 7px 0px #707070;
8 }
9 #neon h1 {
10  text-align: center;
11  margin: 200px auto;
12  font-family: "Museo";
13  font-size: 200px; text-transform: uppercase;
14  color: #fff;
15  text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #fff, 0 0 40px #
    ff00de, 0 0 70px #ff00de, 0 0 80px #ff00de, 0 0 100px #ff00de, 0 0 1
    50px #ff00de;
```



Figura 3.41: CSS3: text-shadow (III).

```

16 }
17 #inset h1 {
18     text-align: center;
19     margin: 200px auto;
20     font-family: "League-Gothic", Courier;
21     font-size: 200px; text-transform: uppercase;
22     color: #222;
23     text-shadow: 0px 2px 3px #666;
24 }
25 #anaglyphic h1 {
26     text-align: center;
27     margin: 200px auto;
28     font-family: "League-Gothic", Courier;
29     font-size: 200px; text-transform: uppercase;
30     color: rgba(0,168,255,0.5);
31     text-shadow: 8px 8px 0 rgba(255,0,180,0.5);
32 }
33 #fire h1 {
34     text-align: center;
35     margin: 200px auto;
36     font-family: "League-Gothic", Courier;
37     font-size: 200px; text-transform: uppercase;
38     color: #fff;
39     text-shadow: 0 0 20px #fefcc9, 10px -10px 30px #feec85, -20px -20px 40
40                 px #ffae34, 20px -40px 50px #ec760c, -20px -60px 60px #cd4606, 0
41                 -80px 70px #973716, 10px -90px 80px #451b0e;
42 }
43 #boardgame h1 {
44     text-align: center;
45     margin: 200px auto;
46     font-family: "League-Gothic", Courier;
47     font-size: 200px; text-transform: uppercase;
48     color: #fff;
49     text-shadow: 10px 10px 0 #ffd217, 20px 20px 0 #5ac7ff, 30px 30px 0 #
50                 ffd217, 40px 40px 0 #5ac7ff;
51 }

```

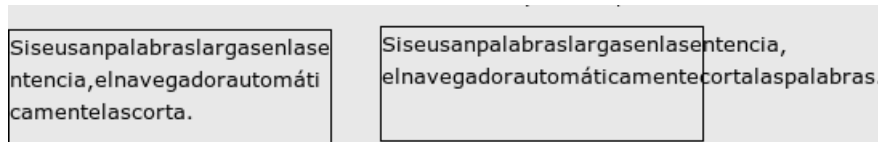


Figura 3.42: CSS3: word-wrap.

Word-wrap

Word-wrap es la propiedad que permite establecer que las palabras largas se corten para no sobrepasar el ancho del tamaño del contenedor definido con la propiedad *width*. El principal defecto que tiene es que cuando se realiza la división de palabras estas no respetan la división de sílabas de los idiomas, ni añade el guión de corte de palabras.

Inicialmente fue ideado por Microsoft pero hoy en día está estandarizado por la W3C. Su única funcionalidad es cuando previamente ha sido establecido un ancho (*width*) al elemento contenedor. Sus dos posibles valores son:

- **normal:** Es el comportamiento por defecto y no divide las palabras.
- **break-word:** Realiza el particionado de las palabras (recuerde sin respetar sílabas).

En la Figura 3.42 se muestra el resultado de utilizar los valores *break-word* o *normal*,

Ejercicio 3.38 Añade las reglas necesarias para añadir sombras en los textos de títulos de tu proyecto Web. Además, observa si puedes combinarlo con las reglas *@font-face*.

Ejercicio 3.39 Utiliza un generador automático de sombras para ver los efectos que se van generando (<http://css3gen.com/text-shadow/>). Apicalos efectos en tu proyecto Web.

Ejercicio 3.40 Establece en tu proyecto Web la propiedad word-wrap en los diferentes textos para conseguir una división de palabras sin que sobresalgan de los elementos.

3.9.5 Múltiples columnas

La estructuración del texto en columnas ha sido una tarea complicada en las versiones anteriores del estándar, teniendo que recurrir a trucos. A partir de ahora existen propiedades específicas que se describen a continuación.

- **column-count:** Permite especificar en cuántas columnas se divide el texto.
- **column-width:** Permite especificar el ancho de cada una de las columnas.
- **column-gap:** Especifica el espacio entre columnas.
- **column-rule:** Especifica si existirá una línea entre las columnas.

Las propiedades *column-count* y *column-width* se utilizarán de manera alternativa, es decir, se utilizará una de las dos para especificar el número de columnas en las que se divide el texto. La principal diferencia entre estas propiedades es que la primera tendrá un número fijo de columnas sin tener en cuenta el tamaño de la pantalla del usuario. En cambio, en la segunda se definirán tantas columnas como permita el tamaño de la pantalla del usuario. En el siguiente código se muestra un ejemplo de aplicación utilizando columnas con dimensiones variables (cada 200 px) y además una separación entre las columnas de 20px.

```
1 .columnas {
2   -moz-column-width: 200px;
3   -moz-column-gap: 20px;
4   -moz-column-rule: 1px dashed #000000;
5   -webkit-column-width: 200px;
```




Figura 3.43: CSS3: Multicolumnas.

```

6  -webkit-column-gap: 20px;
7  -webkit-column-rule: 1px dashed #000000;
8  text-align: justify;
9  }

```

Por otro lado, si se quiere fijar que el número de columnas sea fijo bastaría con utilizar la propiedad *column-count*. En el siguiente código se muestra un código en el cual existen 4 columnas, con una separación de 20px y una línea separadora.

```

1  .columnas4 {
2    -moz-column-count: 4;
3    -moz-column-gap: 20px;
4    -moz-column-rule: 1px dashed #000000;
5    -webkit-column-count: 4;
6    -webkit-column-gap: 20px;
7    -webkit-column-rule: 1px dashed #000000;
8    text-align: justify;

```

Un ejemplo de aplicación del texto multicolumna se muestra en la imagen de la Figura 3.43.

Ejercicio 3.41 Escribe un texto largo en el cual lo dividas en 5 columnas utilizando las dos propiedades anteriormente descritas y exista una línea entre las columnas. ¿Es posible hacerlo de manera estable con las dos técnicas? ¿Por qué?

3.9.6 Degradados

Un degradado es una imagen que se desvanece de un color a otro. En CSS se utiliza esta propiedad para aplicarla a los fondos o cajas, en CSS3 se introducen dos nuevas propiedades para conseguir construir degradados sin necesidad de recurrir a imágenes. No obstante, hoy en día es de las propiedades menos estandarizadas y se pueden encontrar serias diferencias entre las diferentes implementaciones en los navegadores Web. Según los estándares W3C, crear gradientes se compone de:

```
1 linear-gradient( [<punto> || <angulo>,<stop>, <stop> [, <stop>]* )
2 radial-gradient( [<posicion> || <angulo>,<stop> [<forma> || <tamaño>,<stop>
  >,<stop>[, <stop>]* )
```

Una explicación un poco más extensa es la siguiente:

- **linear-gradient:** Esto permite construir un degradado lineal. Sus valores son los siguientes:

- **Orientación X e Y:** Es la orientación de los ejes de coordenadas X e Y desde el cual comenzará el degradado.
- **Color 1:** Es el color de partida.
- **Color 2:** Es el color de finalización.
- **Otros colores:** Se pueden especificar más colores para que los degradados varíen entre dichos colores.

Así que un ejemplo de aplicar un gradiente desde la parte izquierda superior y pasando por varios colores.

```
1 .degrade{
2   background-color: #FFFFFFF;
3   background-image: linear-gradient(left top , #11324b, #f4faff);
4   width:500px;
5   height:200px;
6 }
```

Un ejemplo de crear un gradiente lineal que pueda ser utilizado en diferentes navegadores Web es el siguiente.

```
1 .grad {
2   background-color:#F07575; /* fallback color*/
3   background-image:_webkit-linear-gradient(top, hsl(0,80%,70%), #
   bada55); /*Chrome and Safari*/
4   background-image:-moz-linear-gradient(top, hsl(0,80%,70%), #bada55);
   /*Fx (3.6 to 15)*/
5   background-image:-ms-linear-gradient(top, hsl(0,80%,70%), #bada55);
   /*pre-releases of IE 10*/
6   background-image:-o-linear-gradient(top, hsl(0,80%,70%), #bada55); /*
   old Opera (11.1 to 12.0)*/
7   background-image:linear-gradient(to_bottom, hsl(0,80%,70%), #bada55)
   ; /*Standard syntax; must be last*/
8 }
```

Un ejemplo de aplicación de un degradado desde arriba hacia abajo y cambiando del color azul al blanco es el mostrado en la Figura 3.44(a).

- **radial-gradient:** Los degradados radiales son más complejo que los lineales puesto que se van formado según un radial. La sintaxis básica de estos gradientes es la siguiente:

```
1 -webkit-gradient(radial, centro_interno, radio_interno,
   centro_externo, radio_externo, parada[, parada]+)
```

Donde los diferentes argumentos significan:

- **centro_interno:** Es el centro del círculo interno del degradado. Se puede utilizar las palabras claves *left*, *center* y *right* para designar la componente horizontal y *top*, *center* y *bottom* para la componente vertical. Además, también se pueden utilizar porcentajes.

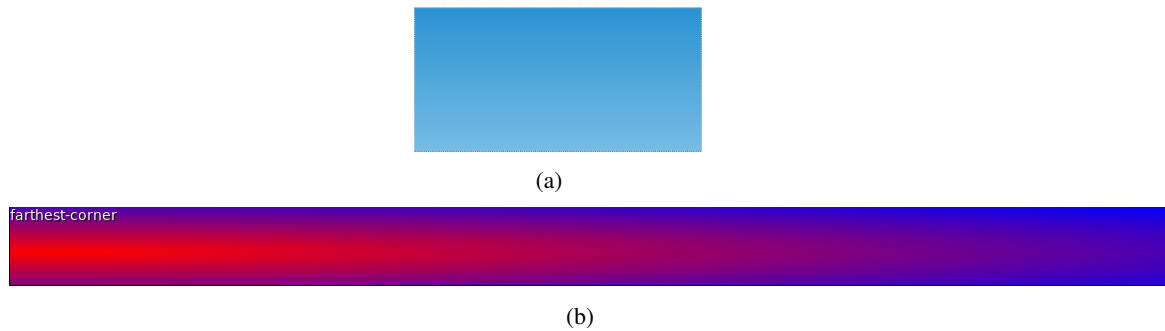


Figura 3.44: CSS3: (a) Gradiente lineal (b) Gradiente Radial.

- **centro_externo:** Es el centro del círculo externo del degradado.
- **radio_interno y radio_externo:** Son los radios de los dos círculos anteriormente definidos.
- **Parada:** Se dispone de una serie de paradas (colores) que son utilizados al igual que para los degradados lineales.

Para profundizar en la comprensión de los degradados radiales se recomienda la lectura de la Web: <https://developer.mozilla.org/en-US/docs/Web/CSS/radial-gradient>. Finalmente, un ejemplo de aplicación del degradado radial es mostrado en la Figura 3.44(b).

```
1 background: -webkit-radial-gradient(45px 45px, farthest-corner, rgb
  (255, 0, 0) 0%, rgb(0, 0, 255) 100%);
2 background: radial-gradient(45px 45px, farthest-corner, rgb
  (255, 0, 0) 0%, rgb(0, 0, 255) 100%);
```

Ejercicio 3.42 Utilizando la herramienta online:

<http://www.colorzilla.com/gradient-editor/> construye varios gradientes y utilízalos como color de fondo.

Ejercicio 3.43 Observa los patrones de degradados de la Web <http://lea.verou.me/css3patterns/>. Basándote en alguno de los ahí mostrados ponlo en aplicación en tu Web.

Ejercicio 3.44 Investiga cómo definir los degradados radiales la W3C. Haz ejemplos de su aplicación.

3.10 Puesta en práctica

En esta sección se describe un conjunto de prácticas que deben realizarse utilizando todas las técnicas presentadas en este capítulo y anteriores.

3.10.1 Esquinas redondeadas, degradados y sombras

En esta práctica se debe hacer uso de las técnicas para conseguir esquinas redondeadas para crear un marco para una fotografía de un lindo gatito. Además, el elemento donde se aloja la fotografía tiene un color degradado con unas lindas sombras. El resultado debe ser el mostrado en la Figura 3.45.



Marco para foto redondo

Figura 3.45: CSS3: Práctica de esquinas redondeadas, degradados y sombras.

3.10.2 Contruir botones con gradientes

En esta práctica se va a construir un conjunto de botones con diferentes formas (cuadrados, redondeados) de diferentes colores (degradados) que al cambiar sus estados cambien el color del degradado. Además, deben ser botones que funcionen en todos los navegadores posibles y en caso de que el navegador no soporte alguna característica mostrará un botón clásico. Así que en la Figura 3.46 muestra el aspecto de los botones en caso de que los navegadores acepten CSS3 o no.

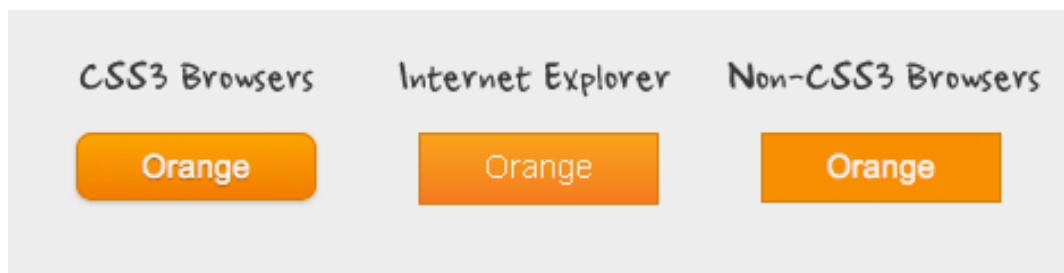


Figura 3.46: CSS3 Práctica de construir botones con gradientes: Aspecto de botones.

El comportamiento de los botones debe responder a las siguientes características:

- **normal state:** Gradiente con bordes y sombras.
- **hover:** Gradiente oscuro.
- **active:** Gradient es revertido, es decir 1px hacia abajo, y color de la fuente oscura.

En la Figura 3.47 se muestra el comportamiento del aspecto de los botones según el estado de los mismos. Además, debe crear un conjunto de botones utilizando estilos (piense que solamente debe modificarlo a través de los estilos) para tener un paquete de botones similar al mostrado en la Figura 3.48.

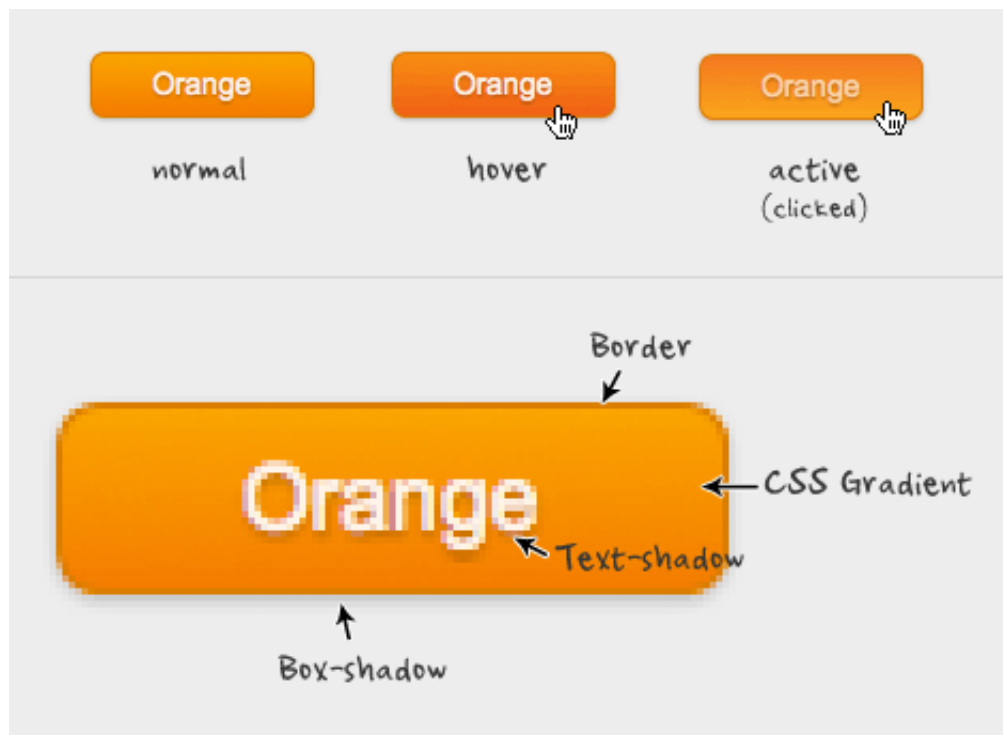


Figura 3.47: CSS3 Práctica de construir botones con gradientes: Estado de botones.

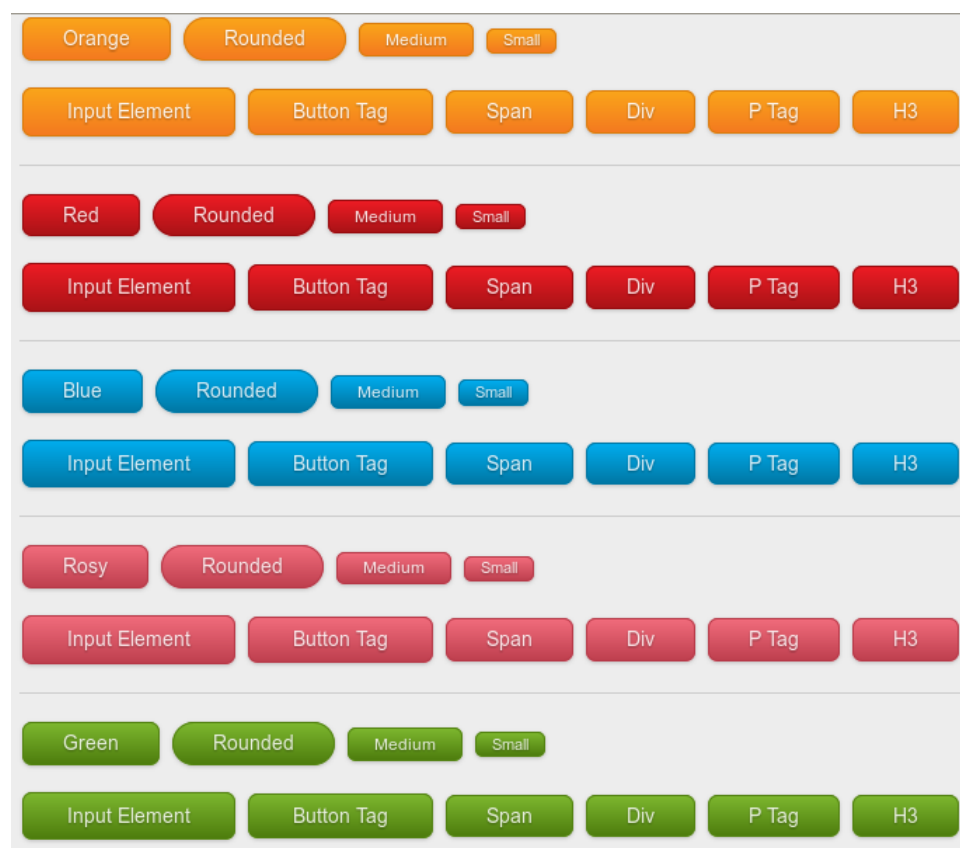


Figura 3.48: CSS3 Práctica de construir botones con gradientes: Juego de botones.

3.10.3 Menú estilo Google

En esta práctica se debe conseguir construir un menú de navegación similar al propuesto por Google. En la Figura 3.49 se muestra el resultado que debe conseguirse en esta práctica. Las características que se deben cumplir:

- Se debe establecer el color de fondo.
- Los enlaces tienen un comportamiento diferente según esté activo, el cursor sobre el enlace o pulsado. Además, cambia el fondo de cada selección a un tono gris cuando está el cursor sobre el bloque que contiene la información.
- Todos los elementos están alineados a la izquierda, si se quisiera simular la identificación de un usuario con una imagen debe ir alineado a la derecha.
- El cursor del ratón cambia a una mano cuando se sitúa sobre una de las opciones posibles.
- El borde superior de la opción seleccionada tiene un tono rojizo.
- Los enlaces de los botones deben abrir enlaces en otras pestañas diferentes.
- Al situar el ratón sobre la opción *more* se debe crear un menú desplegable con más opciones. En caso de no conseguir este efecto ahora mismo no se preocupe puesto que puede utilizar el propio de la siguiente práctica (recuerde NO utilizar JavaScript o JQUERY para ésta práctica).

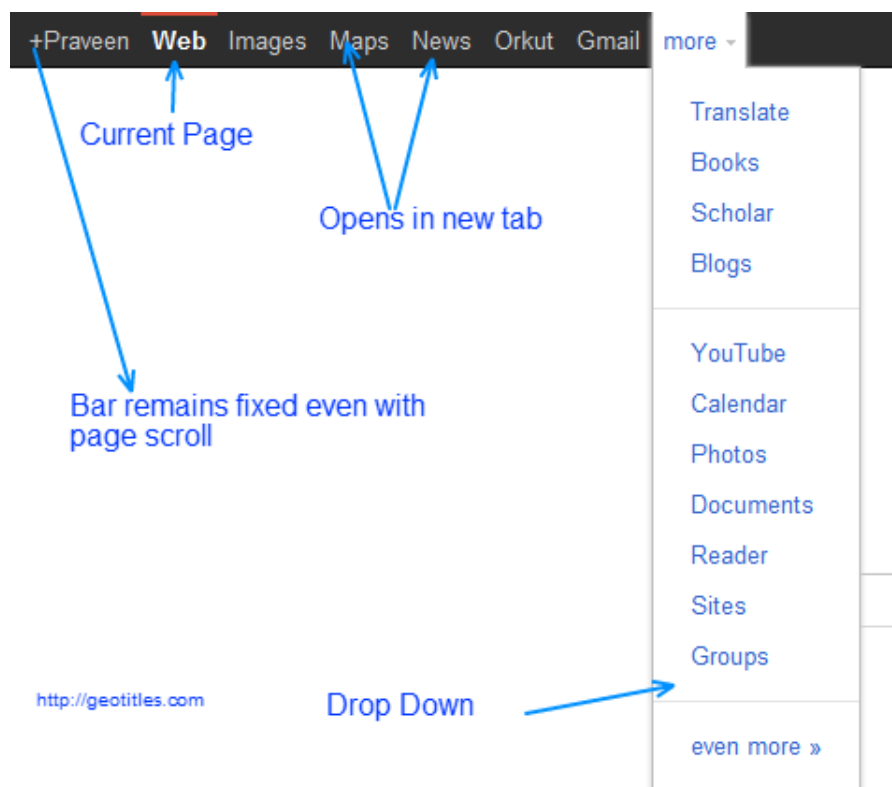


Figura 3.49: CSS3 Práctica: Menú estilo Google.

3.10.4 Menú desplegable multinivel

En esta práctica se debe conseguir construir un menú desplegable multinivel de tal manera que cada vez que se pulsa el ratón sobre las diferentes opciones se desplegarán. Los elementos que tienen subelementos mostrarán una flecha indicando que hay un submenú. Los diferentes

submenús de tercer nivel irán apareciendo de izquierda a derecha. En la Figura 3.50 se muestra el resultado que debe ofrecer ésta práctica.

Como pista debe utilizar la propiedad *display:none* para esconder los elementos del menú que no quiere que se visualicen hasta que llegue su momento puesto que inicialmente en el documento HTML deben aparecer todos los elementos listados. Se le facilita al alumno la especificación por defecto de los elementos de la lista de diferentes niveles que aparecen en el menú de navegación:

```
1 nav li ul {
2   display:none;
3   position:absolute;
4   min-width:140px;
5 }
```

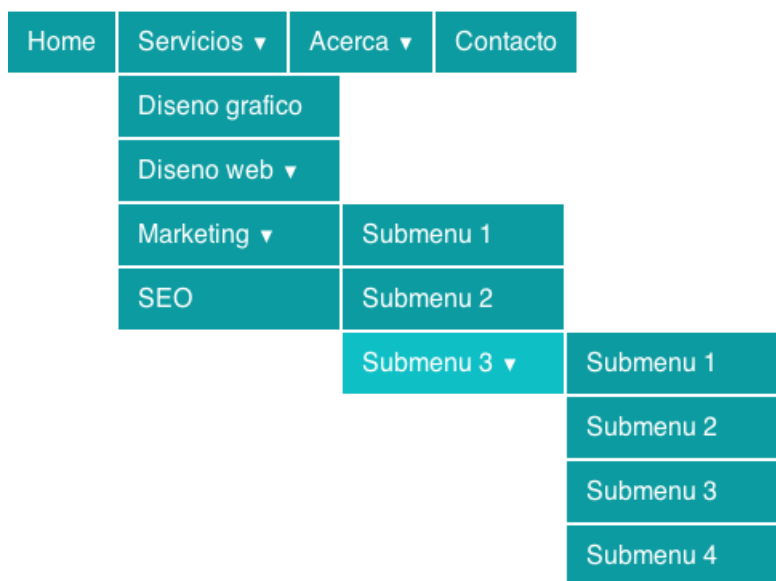


Figura 3.50: CSS3 Práctica: Menú desplegable.

3.10.5 Menú con pestañas

En esta práctica se debe construir un menú de pestañas en las que según qué pestaña esté seleccionada está tendrá un color u otro. Además, también cambiará el tono del borde. El resultado debe ser similar al mostrado en la Figura 3.51.

De modo de ayuda a esta práctica se facilita el fragmento de código HTML que corresponde con las pestañas y el contenido de las mismas. Recuerde que debe aplicar todas las técnicas presentadas hasta el momento.

```
1 <section class="tabs">
2   <input id="tb-1" type="radio" name="radio-set" class="t-selector-1"
3     checked="checked"/>
4   <label for="tb-1" class="tb-label-1">About us</label>
5   <input id="tb-2" type="radio" name="radio-set" class="t-selector-2"/>
6   <label for="tb-2" class="tb-label-2">How we work</label>
7   <input id="tb-3" type="radio" name="radio-set" class="t-selector-3"/>
8   <label for="tb-3" class="tb-label-3">References</label>
9   <input id="tb-4" type="radio" name="radio-set" class="t-selector-4"/>
10  <label for="tb-4" class="tb-label-4">Contact us</label>
```



```

10
11 <div class="clear-shadow"></div>
12 <div class="content">
13   <div class="content-1"><p>Some content</p></div>
14   <div class="content-2"><p>Some content</p></div>
15   <div class="content-3"><p>Some content</p></div>
16   <div class="content-4"><p>Some content</p></div>
17 </div>
18 </section>

```



Figura 3.51: CSS3 Práctica: Menú con pestañas.

3.10.6 Crear bonitos tooltips

En esta práctica se deben crear bonitos tooltips con iconos. Un tooltip es un texto que aparece sobre un texto al pasar un texto sobre él. El alumno debe diseñar 5 tooltips diferentes según las siguientes características:

- Un texto sin resaltar.
- Texto de advertencia.
- Texto de ayuda.
- Texto de información.
- Texto de error.

En la Figura 3.52 se muestran los diferentes tipos de tooltips que existen. Los pasos que debe el alumno realizar son los siguientes:

- Crear el documento HTML necesario con la estructuración necesaria para poder construir un elemento CSS que genere el tooltip.
- Establezca los estados de los elementos en los que aparece el tooltip.
- Construya las reglas necesarias para generar una caja con una imagen a modo de tooltip.
- Incluya sombras y degradados a cada una de las cajas.

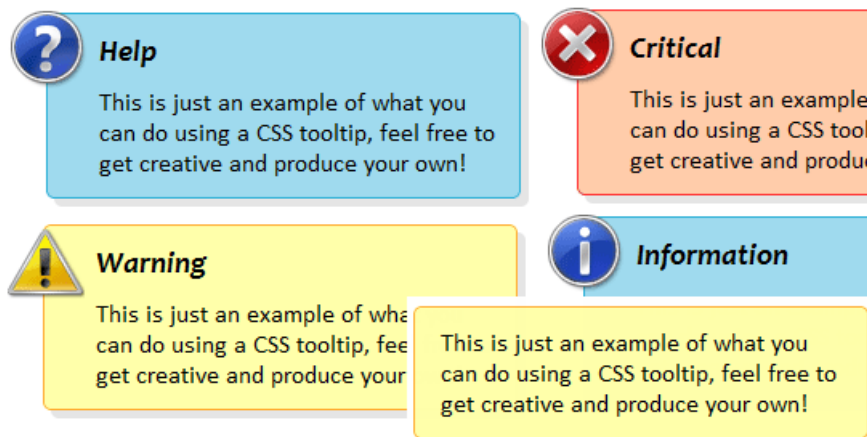


Figura 3.52: CSS3 Práctica: Generar bonitos tooltips.

3.10.7 Diseñar una caja de búsqueda con características de CSS3

En esta práctica el alumno deberá diseñar una caja de búsqueda con características de CSS3 y HTML5. La apariencia del resultado final debe ser similar a la mostrada en la Figura 3.53. A modo de aclaración se debe tener en cuenta:

- El elemento input utiliza placeholder para representar el texto inicialmente.
- El elemento input está enmarcado en un elemento con bordes redondeados, degradados y sombra.
- El botón cambia el aspecto al ser pulsado (con un tono diferente de degradado).

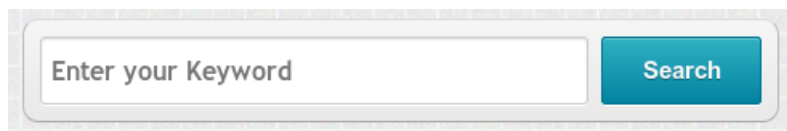


Figura 3.53: CSS3 Práctica: Caja de búsqueda.

3.10.8 Formularios y CSS3

En esta práctica el alumno creará un completo formulario utilizando HTML5 y CSS3. Para conseguir efectos sorprendentes en CSS existen varias pseudo-clases para manipular los efectos en los formularios. En primer lugar existe la pseudo-clase **:invalid** la cual representa cualquier cambio inválido que sea resultado de una validación, lo opuesto ocurre con su contraparte **:valid**. Esto facilita que si un campo contiene un valor inválido pueda adoptar una apariencia acorde, de esta forma ayudando al usuario a identificar los campos que se necesitan ser verificados.

```
1  input:invalid_{border:_1px_solid_red; }
2
3  /* Estilo por defecto */
4  input:valid_{border:_1px_solid_green; }
```

Por defecto todos los campos son válidos, pero si un campo es requerido, entonces ocurre lo contrario, este por defecto es inválido.

```
1  /* Estilo por defecto */
2  input:required:invalid_{border:_1px_solid_red; }
3
```

```
4 input:required:valid_{border:_1px_solid_green;}
```

Así que el resultado del formulario debe ser similar al mostrado en la Figura 3.54 y donde se debe destacar las siguientes características:

- Elementos HTML5 propios para validar formularios (required, email, ...).
- Utilizar placeholder.
- Utilizar pseudo-clases para remacar errores o campos válidos.
- El aspecto debe tener degradados, sombras y alineación como la mostrada.
- El botón debe tener diferente estados.

The image shows a web form titled "Contáctanos" in green text. The form is divided into four sections by horizontal lines. The first section is labeled "Nombre:" and contains a text input field with the value "Johan", which has a green border. The second section is labeled "Email:" and contains a text input field with the value "j@ozu.es", also with a green border. The third section is labeled "Sitio Web:" and contains a text input field with the value "http://tutosytips.com", which has a red border. The fourth section is labeled "Mensaje:" and contains a large text area. Below the text area is a button labeled "Enviar".

Figura 3.54: CSS3 Práctica: Formulario Web.