

Por último, indicar que cuando se utiliza el posicionamiento flotante es común encontrarse el problema de que un determinado elemento tenga en su interior solamente elementos flotados y el resultado visual no sea el esperado. Cuando esto ocurre los elementos pueden “salirse” del contenedor, para solucionar este problema, aplica la propiedad overflow con el valor auto sobre ese elemento contenedor.

4.13 Visualización

Además de las propiedades que controlan el posicionamiento de los elementos, CSS define otras cuatro propiedades para controlar su visualización: display, visibility, overflow y z-index.

Utilizando algunas de estas propiedades es posible ocultar y/o hacer invisibles las cajas de los elementos, por lo que son imprescindibles para realizar efectos avanzados y animaciones.

4.13.1 Propiedades display y visibility

Las propiedades display y visibility controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página. Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.

La propiedad display permite ocultar completamente un elemento haciendo que desaparezca de la página. Como el elemento oculto no se muestra, el resto de los elementos de la página se mueven para ocupar su lugar.

Por otra parte, la propiedad visibility permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento, pero no la muestra. En este caso, el resto de los elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.

La siguiente imagen muestra la diferencia entre ocultar la caja número 5 mediante la propiedad display o hacerla invisible mediante la propiedad visibility:

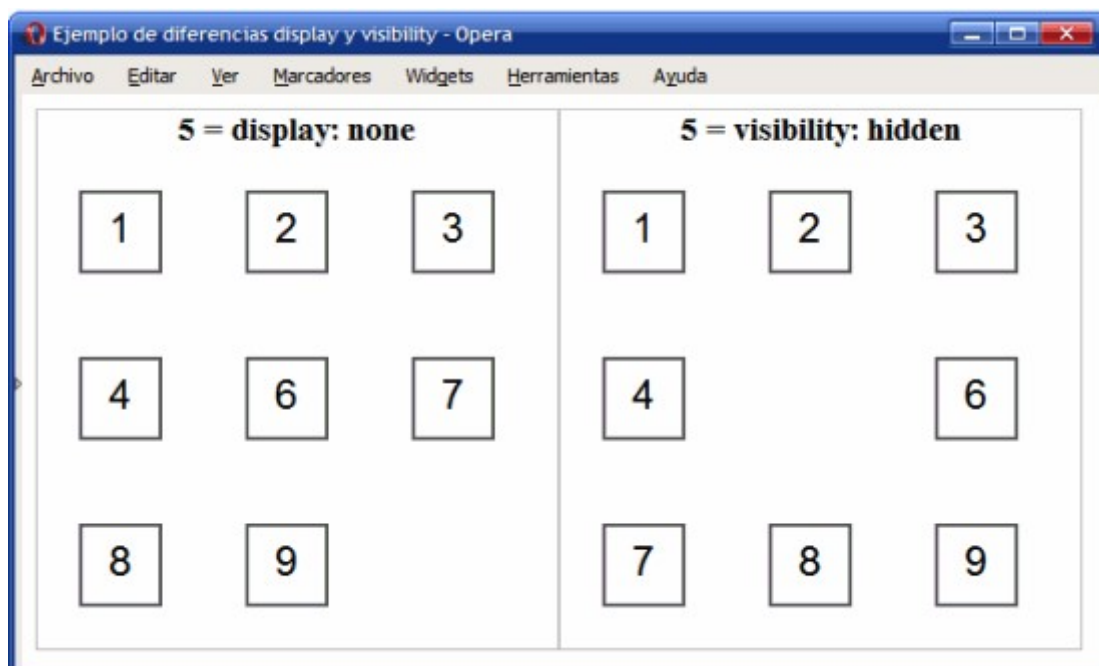


Figura 5.21 Diferencias visuales entre las propiedades display y visibility

En general, cuando se oculta un elemento no es deseable que siga ocupando sitio en la página, por lo que la propiedad display se utiliza mucho más que la propiedad visibility.

A continuación, se muestra la definición completa de la propiedad display:

Propiedad	display
Valores	inline block none list-item run-in inline-block table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption inherit
Se aplica a	Todos los elementos
Valor inicial	inline
Descripción	Permite controlar la forma de visualizar un elemento e incluso ocultarlo

Las posibilidades de la propiedad display son mucho más avanzadas que simplemente ocultar elementos. En realidad, la propiedad display modifica la forma en la que se visualiza un elemento.

Los valores más utilizados son inline, block y none. El valor block muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate. El valor inline visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate.

El valor none oculta un elemento y hace que desaparezca de la página. El resto de los elementos de la página se visualizan como si no existiera el elemento oculto, es decir, pueden ocupar el espacio en el que se debería visualizar el elemento.

El siguiente ejemplo muestra el uso de la propiedad display para mostrar un elemento de bloque como si fuera un elemento en línea y para mostrar un elemento en línea como si fuera un elemento de bloque:

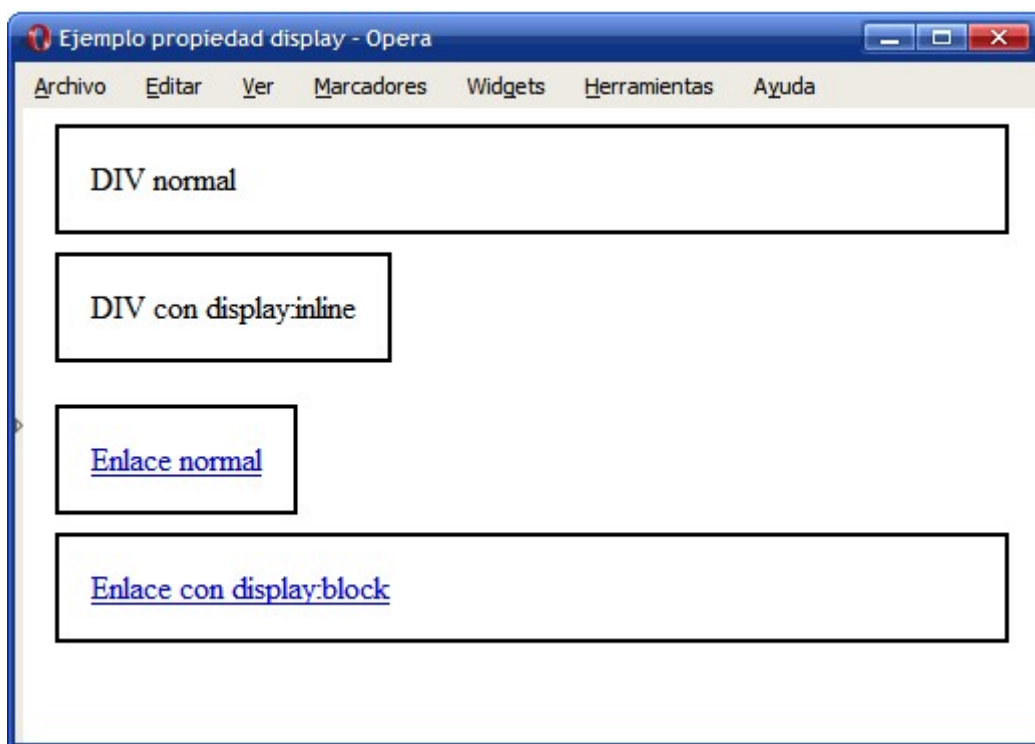


Figura 5.22 Ejemplo de propiedad display

Las reglas CSS del ejemplo anterior son las siguientes:

```
<div>DIV normal</div>

<div style="display:inline">DIV con display:inline</div>

<a href="#">Enlace normal</a>

<a href="#" style="display:block">Enlace con display:block</a>
```

Como se verá más adelante, la propiedad display: inline se puede utilizar en las listas (,) que se quieren mostrar horizontalmente y la propiedad display: block se emplea frecuentemente para los enlaces que forman el menú de navegación.

Por su parte, la definición completa de la propiedad visibility es mucho más sencilla:

Propiedad	visibility
Valores	visible hidden collapse inherit
Se aplica a	Todos los elementos
Valor inicial	visible
Descripción	Permite hacer visibles e invisibles a los elementos

Las posibilidades de la propiedad visibility son mucho más limitadas que las de la propiedad display, ya que sólo permite hacer visibles o invisibles a los elementos de la página.

Inicialmente todas las cajas que componen la página son visibles. Empleando el valor hidden es posible convertir una caja en invisible para que no muestre sus contenidos. El resto de

elementos de la página se muestran como si la caja todavía fuera visible, por lo que en el lugar donde originalmente se mostraba la caja invisible, ahora se muestra un hueco vacío.

Por último, el valor **collapse** de la propiedad visibility sólo se puede utilizar en las filas, grupos de filas, columnas y grupos de columnas de una tabla. Su efecto es similar al de la propiedad display, ya que oculta completamente la fila y/o columna y se pueden mostrar otros contenidos en ese lugar. Si se utiliza el valor collapse sobre cualquier otro tipo de elemento, su efecto es idéntico al valor hidden.

4.13.2 Relación entre display, float y position

Cuando se establecen las propiedades display, float y position sobre una misma caja, su interpretación es la siguiente:

- Si display vale none, se ignoran las propiedades float y position y la caja no se muestra en la página.
- Si position vale absolute o fixed, la caja se posiciona de forma absoluta, se considera que float vale none y la propiedad display vale block tanto para los elementos en línea como para los elementos de bloque. La posición de la caja se determina mediante el valor de las propiedades top, right, bottom y left.
- En cualquier otro caso, si float tiene un valor distinto de none, la caja se posiciona de forma flotante y la propiedad display vale block tanto para los elementos en línea como para los elementos de bloque.

4.13.3 Propiedad overflow

Normalmente, los contenidos de un elemento se pueden mostrar en el espacio reservado para ese elemento. Sin embargo, en algunas ocasiones el contenido de un elemento no cabe en el espacio reservado para ese elemento y se desborda.

La situación más habitual en la que el contenido sobresale de su espacio reservado es cuando se establece la anchura y/o altura de un elemento mediante la propiedad width y/o height. Otra situación habitual es la de las líneas muy largas contenidas dentro de un elemento <pre>, que hacen que la página entera sea demasiado ancha.

CSS define la propiedad overflow para controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.

Propiedad	overflow
Valores	visible hidden scroll auto inherit
Se aplica a	Elementos de bloque y celdas de tablas
Valor inicial	visible
Descripción	Permite controlar los contenidos sobrantes de un elemento

Los valores de la propiedad overflow tienen el siguiente significado:

- **visible:** el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.

- **hidden:** el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.
- **scroll:** solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de scroll que permiten visualizar el resto del contenido.
- **auto:** el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.

La siguiente imagen muestra un ejemplo de los tres valores típicos de la propiedad overflow:



Figura 5.23 Ejemplo de propiedad overflow

El código HTML y CSS del ejemplo anterior se muestran a continuación:

div {

```
display: inline;
float: left;
margin: 1em;
padding: .3em;
border: 2px solid #555;
width: 100px;
height: 150px; /*espacio reservado*/
font: 1em Arial, Helvetica, sans-serif;
```

}

```
<div><h1>overflow: visible</h1> Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Cras vitae dolor eu enim dignissim lacinia. Maecenas
blandit. Morbi mi.</div>
```

```
<div style="overflow:hidden"><h1>overflow: hidden</h1> Lorem ipsum dolor  
sit amet, consectetur adipiscing elit. Cras vitae dolor eu enim dignissim  
lacinia. Maecenas blandit. Morbi mi.</div>
```

```
<div style="overflow:scroll"><h1>overflow: scroll</h1> Lorem ipsum dolor sit  
amet, consectetur adipiscing elit. Cras vitae dolor eu enim dignissim  
lacinia.
```

```
Maecenas blandit. Morbi mi.</div>
```

4.13.4 Propiedad z-index

Además de posicionar una caja de forma horizontal y vertical, CSS permite controlar la posición tridimensional de las cajas posicionadas. De esta forma, es posible indicar las cajas que se muestran delante o detrás de otras cajas cuando se producen solapamientos.

La posición tridimensional de un elemento se establece sobre un tercer eje llamado Z y se controla mediante la propiedad `z-index`. Utilizando esta propiedad es posible crear páginas complejas con varios niveles o capas.

A continuación, se muestra la definición formal de la propiedad `z-index`:

Propiedad	<code>z-index</code>
Valores	auto numero inherit
Se aplica a	Elementos que han sido posicionados explícitamente
Valor inicial	auto
Descripción	Establece el nivel tridimensional en el que se muestra el elemento

El valor más común de la propiedad `z-index` es un número entero. Aunque la especificación oficial permite los números negativos, en general se considera el número 0 como el nivel más bajo.

Cuanto más alto sea el valor numérico, más cerca del usuario se muestra la caja. Un elemento con `z-index: 10` se muestra por encima de los elementos con `z-index: 8` o `z-index: 9`, pero por debajo de elementos con `z-index: 20` o `z-index: 50`.

La siguiente imagen muestra un ejemplo de uso de la propiedad `z-index`:

5 Capítulo 6. Enlaces

5.1 Estilos básicos

5.1.1 Tamaño, color y decoración

Los estilos más sencillos que se pueden aplicar a los enlaces son los que modifican su tamaño de letra, su color y la decoración del texto del enlace. Utilizando las propiedades `text-decoration` y `font-weight` se pueden conseguir estilos como los que se muestran en la siguiente imagen:



Figura 7.1 Ejemplo de enlaces con estilos diferentes

A continuación, se muestran las reglas CSS del ejemplo anterior:

```
a { margin: 1em 0; display: block;}

.alternativo {color: #CC0000;}
.simple {text-decoration: none;}
.importante {font-weight: bold; font-size: 1.3em;}
.raro {text-decoration:overline;}

<a href="#">Enlace con el estilo por defecto</a>
<a class="alternativo" href="#">Enlace de color rojo</a>
<a class="simple" href="#">Enlace sin subrayado</a>
<a class="importante" href="#">Enlace muy importante</a>
<a class="raro" href="#">Enlace con un estilo raro</a>
a:visited:hover { ... }
```

5.2 Estilos avanzados

5.2.1 Decoración personalizada

La propiedad **text-decoration** permite añadir o eliminar el subrayado de los enlaces. Sin embargo, el aspecto del subrayado lo controla automáticamente el navegador, por lo que su

color siempre es el mismo que el del texto del enlace y su anchura es proporcional al tamaño de letra.

No obstante, utilizando la propiedad **border-bottom** es posible añadir cualquier tipo de subrayado para los enlaces de las páginas. El siguiente ejemplo muestra algunos enlaces con el subrayado personalizado:

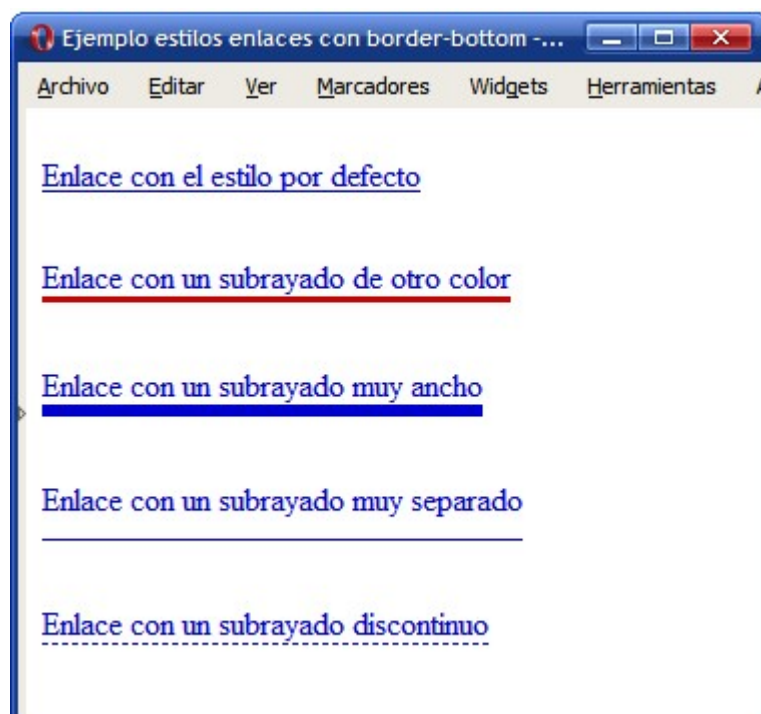


Figura 7.2 Enlaces con subrayado personalizado mediante la propiedad border

Las reglas CSS definidas en el ejemplo anterior se muestran a continuación:

```
a { margin: 1em 0; float: left; clear: left; text-decoration: none;}
.simple {text-decoration: underline;}
.color { border-bottom: medium solid #CC0000;}
.ancho {border-bottom: thick solid;}
.separado {border-bottom: 1px solid; padding-bottom: .6em;}
.discontinuo {border-bottom: thin dashed;}
```

```
<a class="simple" href="#">Enlace con el estilo por defecto</a>
```

```
<a class="color" href="#">Enlace un subrayado de otro color</a>
```

```
<a class="ancho" href="#">Enlace con un subrayado muy ancho</a>
```

```
<a class="separado" href="#">Enlace con un subrayado muy separado</a>
```

```
<a class="discontinuo" href="#">Enlace con un subrayado discontinuo</a>
```

5.2.2 Imágenes según el tipo de enlace

En ocasiones, puede resultar útil incluir un pequeño icono al lado de un enlace para indicar el tipo de contenido que enlaza, como por ejemplo un archivo comprimido o un documento con formato PDF.

Este tipo de imágenes son puramente decorativas en vez de imágenes de contenido, por lo que se deberían añadir con CSS y no con elementos de tipo ``. Utilizando la propiedad `background` (y `background-image`) se puede personalizar el aspecto de los enlaces para que incluyan un pequeño icono a su lado.

La técnica consiste en mostrar una imagen de fondo sin repetición en el enlace y añadir el padding necesario al texto del enlace para que no se solape con la imagen de fondo.

El siguiente ejemplo personaliza el aspecto de dos enlaces añadiéndoles una imagen de fondo:



Figura 7.3 Personalizando el aspecto de los enlaces en función de su tipo

Las reglas CSS necesarias se muestran a continuación:

```
a { margin: 1em 0; float: left; clear: left; }

.rss {
  color: #E37529;
  padding: 0 0 0 18px;
  background: #FFF url(imagenes/rss.gif) no-repeat left center;
}

.pdf {
  padding: 0 0 0 22px;
  background: #FFF url(imagenes/pdf.png) no-repeat left center;
}

<a href="#">Enlace con el estilo por defecto</a>

<a class="rss" href="#">Enlace a un archivo RSS</a>

<a class="pdf" href="#">Enlace a un documento PDF</a>
```

5.2.3 Mostrar los enlaces como si fueran botones

Las propiedades definidas por CSS permiten mostrar los enlaces con el aspecto de un botón, lo que puede ser útil en formularios en los que no se utilizan elementos específicos para crear botones.

El siguiente ejemplo muestra un enlace simple formateado como un botón:

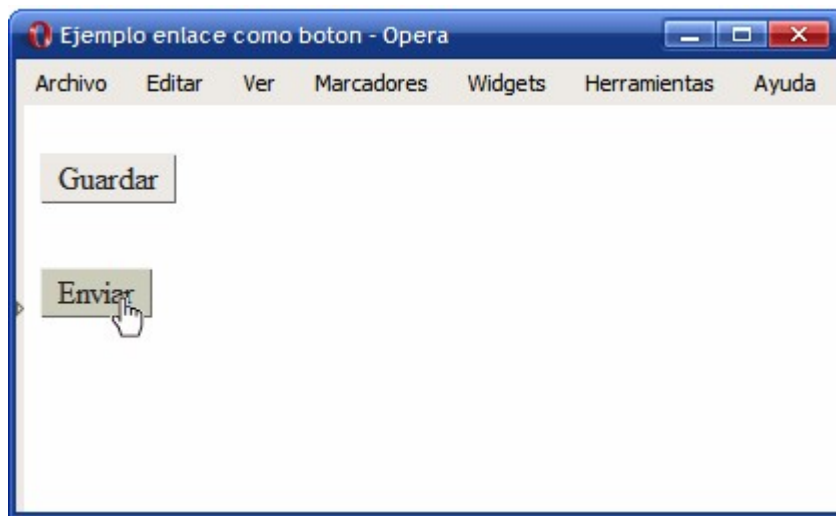


Figura 7.4 Mostrando un enlace como si fuera un botón

Las reglas CSS utilizadas en el ejemplo anterior son las siguientes:

```
a { margin: 1em 0; float: left; clear: left; }
a.boton {
    text-decoration: none;
    background: #EEE;
    color: #222;
    border: 1px outset #CCC;
    padding: .1em .5em;
}
a.boton:hover {
    background: #CCB;
}
a.boton:active {
    border: 1px inset #000;
}
```

```
<a class="boton" href="#">Guardar</a>
<a class="boton" href="#">Enviar</a>
```

6 Validador CSS

La validación del código CSS y de las reglas que lo forman es un concepto similar a la validación de documentos XHTML.

La validación es un mecanismo que permite comprobar que el código CSS creado cumple las reglas de la sintaxis del lenguaje CSS y que por tanto es una hoja de estilos válida para aplicarla a cualquier documento XHTML.

La validación suele ser útil cuando se producen errores en los estilos definidos o comportamientos no deseados al aplicar las reglas CSS. En muchas ocasiones, los errores se producen porque el navegador está ignorando algunas reglas que contienen propiedades mal definidas o errores de sintaxis.

El W3C (World Wide Web Consortium) dispone de un validador online que permite validar reglas CSS sueltas, páginas XHTML con CSS incluido y archivos CSS independientes. El validador se puede acceder en <http://jigsaw.w3.org/css-validator/>