TEMA 8. POSICIONAMIENTO, MAQUETACIÓN Y VISUALIZACIÓN

1 POSICIONAMIENTO

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible diseñar estructuras de páginas que de otra forma no serían posibles.

El estándar de CSS define cinco modelos diferentes para posicionar una caja:

- ✓ **Posicionamiento normal o estático**: se trata del posicionamiento que utilizan los navegadores por defecto, es el que tienen todos los elementos al incluirlos en nuestro documento. Aunque hablemos de posicionamiento estático, no se considera que el elemento está posicionado.
- ✓ **Posicionamiento relativo**: La posición de la caja se establece en función (relativa) a la posición pantalla donde se ha colocado el elemento cuando ha sido creado y a partir de ahí, desplazamos el elemento respecto de su posición original donde fue posicionado cuando se creó.
- ✓ **Posicionamiento absoluto**: la posición de la caja se establece respecto de su elemento contenedor y el resto de los elementos de la página ignoran la nueva posición del elemento, pierde su espacio en pantalla y pueden colocarse encima de otro elemento.
- ✓ **Posicionamiento fijo:** variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- ✓ **Posicionamiento pegatina**: se trata del modelo más especial de posicionamiento, ya que desplaza las cajas hasta una posición y a partir de ahí el posicionamiento es absoluto, es decir, ya no se mueve, permanece fijo.

El posicionamiento de una caja se establece mediante la propiedad position, asignándole alguno de los siguientes valores:

position	
Valores	relative | absolute | fixed | sticky
Descripción	Selecciona el posicionamiento con el que se mostrará el elemento

El significado de cada uno de los posibles valores de la propiedad position es el siguiente:

- ✓ ***relative:*** corresponde al posicionamiento relativo. El desplazamiento de la caja se controla con las propiedades ***top, right, bottom y left.***
- ✓ ***absolute***: corresponde al posicionamiento absoluto. El desplazamiento de la caja también se controla con las propiedades ***top, right, bottom y left***, pero su interpretación es mucho más compleja, ya que el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.
- ✓ **fixed**: corresponde al posicionamiento fijo. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.
- ✓ **sticky**: corresponde al posicionamiento pegatina. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.

Cuando un elemento está posicionado, es decir, le hemos dado valor a la propiedad position, podemos utilizar las propiedades top, left, right, bottom y z-index. Si el elemento no está posicionado, es decir, tiene posicionamiento estático, estás propiedades no van a funcionar y no las vamos a poder utilizar.

top, right, bottom, left	
Valores	unidad de medida | porcentaje | auto | inherit
Descripción	Indican el desplazamiento horizontal y vertical del elemento respecto de su posición original

En el caso del posicionamiento relativo, cada una de estas propiedades indica el desplazamiento del elemento desde la posición original de su borde superior/derecho/inferior/izquierdo. Si el posicionamiento es absoluto, las propiedades indican el desplazamiento del elemento respecto del borde superior/derecho/inferior/izquierdo de su primer elemento padre posicionado.

En cualquiera de los dos casos, si el desplazamiento se indica en forma de porcentaje, se refiere al porcentaje sobre la anchura (propiedades right y left) o altura (propiedades top y bottom) del elemento.

1.1 POSICIONAMIENTO ESTÁTICO

El posicionamiento normal, es el que se aplica por defecto a los elementos cuando se añaden a una página. En este posicionamiento, sólo se tiene en cuenta si el elemento es de bloque o de línea y su contenido.

Cuando un elemento se coloca en pantalla tiene posicionamiento estático, pero como decíamos al principio se considera que no está posicionado, para estar posicionado tenemos que darle valor a la propiedad position.

Si un elemento se encuentra dentro de otro, el elemento padre se llama *"elemento contenedor"* y determina tanto la posición como el tamaño de todas sus cajas interiores.

Si un elemento no se encuentra dentro de un elemento contenedor, entonces su elemento contenedor es el elemento <body> de la página.

1.2 POSICIONAMIENTO RELATIVO

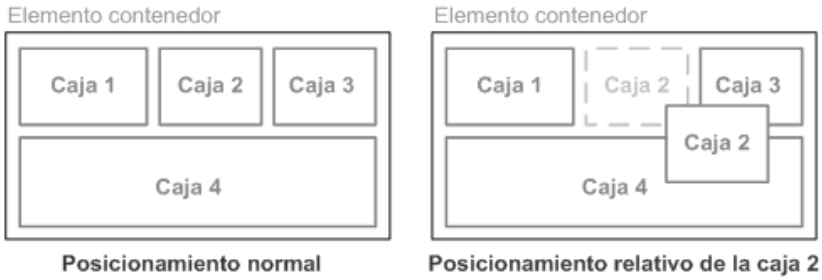
Una caja con posicionamiento relativo ocupa la posición de pantalla en la que fue creada. Pero cuando queremos mover la caja con posicionamiento relativo, se desplaza respecto de su posición original. El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.

El valor de la propiedad top se interpreta como el desplazamiento entre el borde superior de la caja en su posición final y el borde superior de la misma caja en su posición original.

De la misma forma, el valor de las propiedades left, right y bottom indica respectivamente el desplazamiento entre el borde izquierdo/derecho/inferior de la caja en su posición final y el borde izquierdo/derecho/inferior de la caja original.

Por tanto, la propiedad top se emplea para mover las cajas de forma descendente, la propiedad bottom mueve las cajas ascendentemente, la propiedad left se utiliza para desplazar las cajas hacia la derecha y la propiedad right mueve las cajas hacia la izquierda. Este comportamiento parece poco intuitivo y es causa de errores cuando se empiezan a diseñar páginas con CSS. Si se utilizan valores negativos en las propiedades top, right, bottom y left, su efecto es justamente el inverso.

El desplazamiento relativo de una caja no afecta al resto de cajas adyacentes, que se muestran en la misma posición que si la caja desplazada no se hubiera movido de su posición original.



Ejemplo de posicionamiento relativo de un elemento

En la imagen anterior, la caja 2 se ha desplazado lateralmente hacia la derecha y verticalmente de forma descendente. Como el resto de cajas de la página no modifican su posición, se producen solapamientos entre los contenidos de las cajas.

El siguiente ejemplo muestra tres imágenes posicionadas de forma normal:

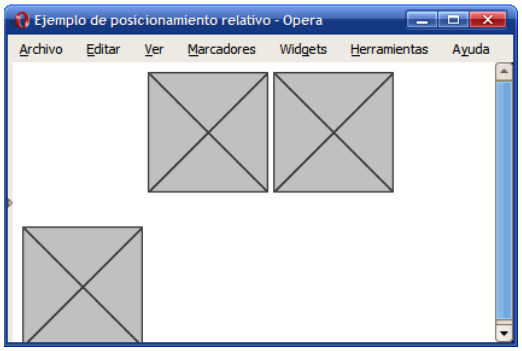
Elementos posicionados de forma normal

Aplicando el posicionamiento relativo, se desplaza la primera imagen de forma descendente:

```
img.desplazada {
  position: relative;
  top: 8em;
}
```

```
<img class="desplazada" src="imagenes/imagen.png" alt="Imagen genérica" />
<img src="imagenes/imagen.png" alt="Imagen genérica" />
<img src="imagenes/imagen.png" alt="Imagen genérica" />
```

El aspecto que muestran ahora las imágenes es el siguiente:



Elemento posicionado de forma relativa

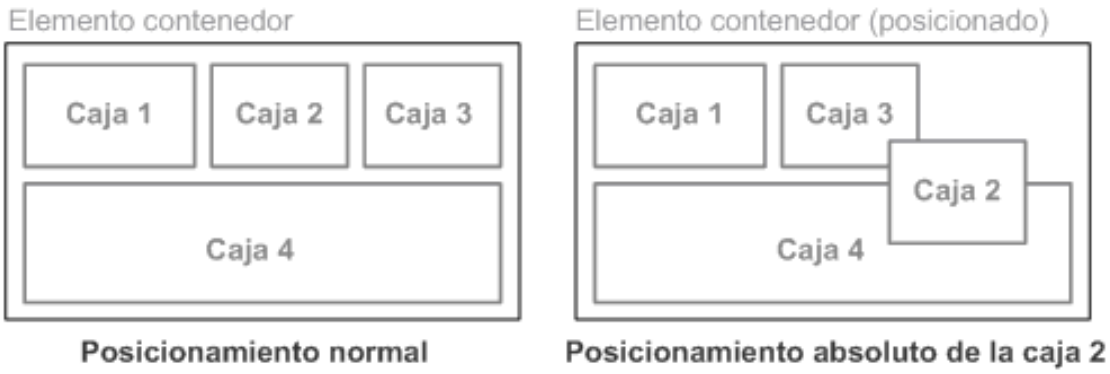
1.3 POSICIONAMIENTO ABSOLUTO

El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades top, right, bottom y left. La interpretación de los valores de estas propiedades es mucho más compleja

que en el posicionamiento relativo, ya que en este caso dependen del posicionamiento del elemento contenedor.

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 se pueden producir solapamientos con otras cajas.

En el siguiente ejemplo, se posiciona de forma absoluta la caja 2:

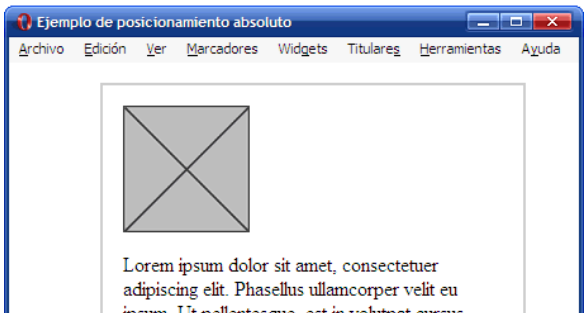


Ejemplo de posicionamiento absoluto de un elemento

La caja 2 está posicionada de forma absoluta, lo que provoca que el resto de elementos de la página modifiquen su posición. En concreto, la caja 3 deja su lugar original y pasa a ocupar el hueco dejado por la caja 2.

Las cajas posicionadas de forma absoluta *"salen del flujo normal de la página"*, lo que provoca que el resto de los elementos de la página se muevan y en ocasiones, ocupen la posición original en la que se encontraba la caja.

Por otra parte, el desplazamiento de una caja posicionada de forma absoluta se controla mediante las propiedades top, right, bottom y left.



Situación original antes de modificar el posicionamiento

A continuación, se muestra el código HTML y CSS de la página original:

```
div {
  border: 2px solid #CCC;
  padding: 1em;
  margin: 1em 0 1em 4em;
  width: 300px;
}

<div>
  <img src="imagenes/imagen.png" alt="Imagen genérica" />
```

```
 <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellusullamcorper velit eu ipsum. Ut pellentesque, est in volutpat cursus, risus mi viverra augue, at pulvinar turpis leo sed orci. Donec ipsum. Curabitur felis dui, ultrices ut, sollicitudin vel, rutrum at, tellus.</p>
</div>
```

En primer lugar, se posiciona de forma absoluta la imagen mediante la propiedad position y se indica su nueva posición mediante las propiedades top y left:

```
div img {
 position: absolute;
 top:  50px;
 left: 50px;
}
```

El resultado *visual* se muestra en la siguiente imagen:

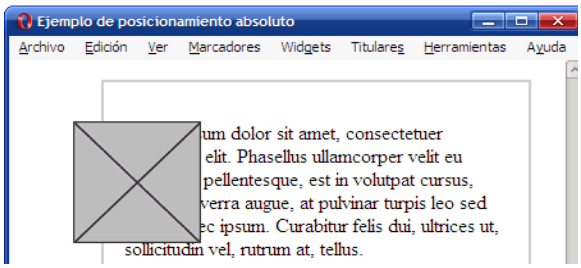
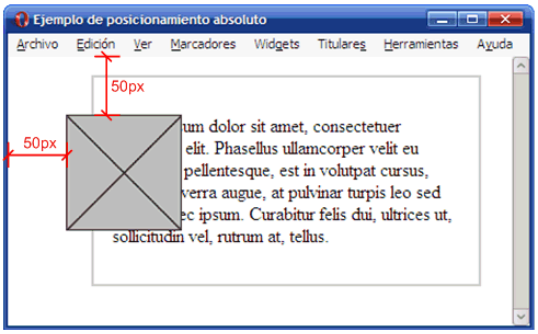


Imagen posicionada de forma absoluta

La imagen posicionada de forma absoluta no toma como referencia su elemento contenedor , sino la ventana del navegador, tal y como demuestra la siguiente imagen:



La referencia del posicionamiento absoluto es la ventana del navegador

Como la imagen se posiciona de forma absoluta, el resto de elementos de la página se mueven para ocupar el lugar libre dejado por la imagen. Por este motivo, el párrafo sube hasta el principio del y se produce un solapamiento con la imagen posicionada que impide ver parte de los contenidos del párrafo.

A continuación, se modifica el ejemplo anterior posicionando de forma relativa el elemento que contiene la imagen y el párrafo. La única propiedad añadida al es position: relative por lo que el elemento contenedor se posiciona, pero no se desplaza respecto de su posición original:

```
div {
 border: 2px solid #CCC;
 padding: 1em;
 margin: 1em 0 1em 4em;
 width: 300px;
 position: relative;
}
```

```
div img {
  position: absolute;
  top: 50px;
  left: 50px;
}
```

La siguiente imagen muestra el resultado obtenido:

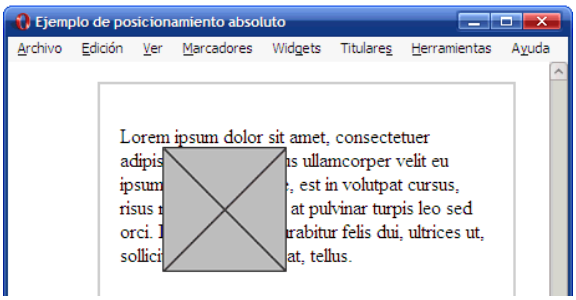
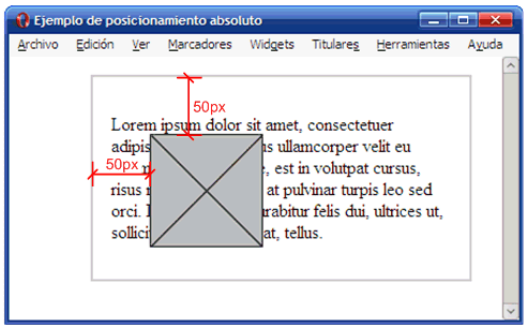


Imagen posicionada de forma absoluta

En este caso, como el elemento contenedor de la imagen está posicionado, se convierte en la referencia para el posicionamiento absoluto. El resultado es que la posición de la imagen es muy diferente a la del ejemplo anterior:



La referencia del posicionamiento absoluto es el elemento contenedor de la imagen

Por tanto, si se quiere posicionar un elemento de forma absoluta respecto de su elemento contenedor, es imprescindible posicionar este último. Para ello, sólo es necesario añadir la propiedad `position: relative`, por lo que no es obligatorio desplazar el elemento contenedor respecto de su posición original.

1.4 POSICIONAMIENTO FIJO

El estándar CSS considera que el posicionamiento fijo es un caso particular del posicionamiento absoluto, ya que sólo se diferencian en el comportamiento de las cajas posicionadas.

Cuando una caja se posiciona de forma fija, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es idéntica al posicionamiento absoluto. De hecho, si el usuario no mueve la página HTML en la ventana del navegador, no existe ninguna diferencia entre estos dos modelos de posicionamiento.

La principal característica de una caja posicionada de forma fija es que su posición es inamovible dentro de la ventana del navegador. El posicionamiento fijo hace que las cajas no modifiquen su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.

Si la página se visualiza en un medio paginado (por ejemplo en una impresora) las cajas posicionadas de forma fija se repiten en todas las páginas. Esta característica puede ser útil para crear encabezados o pies de página en páginas HTML preparadas para imprimir.

El posicionamiento fijo apenas se ha utilizado en el diseño de páginas web hasta hace poco tiempo porque el navegador Internet Explorer 6 y las versiones anteriores no lo soportan.

1.5 POSICIONAMIENTO STICKY

hace que el elemento se comporte como en position:relative hasta que, debido al scroll de la página, el elemento es alcanzado por la parte superior de la pantalla. En ese momento se comporta como fixed, y para que funcione correctamente hay que especificar al menos un valor top, bottom, left o right, en función de si va a ser sticky en un scroll horizontal o vertical.

Los casos más comunes en los que se aplica esto suelen ser para fijar un menú en la parte superior de la página, resaltar banners, formularios u otras llamadas a la acción.

Con la propiedad top podemos contralar la posición en la que queremos que el elemento pase de sticky a fixed.

2 MAQUETACIÓN

En esta parte del tema vamos a ver la forma en la que vamos a visualizar y distribuir los elementos en pantalla, es decir, como se van a maquetar.

La forma en la que vamos a realizar la maquetación es como se realizaba la maquetación hasta la llegada de flex que veremos más adelante.

Antes de la llegada de flex teníamos dos formas de maquetar, una era utilizando la propiedad float y la otra utilizando la propiedad display.

2.1 FLOAT

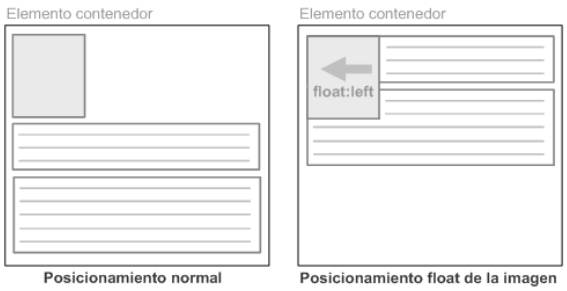
Float nos permite colocar una caja al lado de otra, aunque sea un elemento en bloque.

float	
Valores	left | right | none
Descripción	Establece el tipo de posicionamiento flotante del elemento

Si se indica un valor left, la caja se desplaza hasta el punto más a la izquierda posible en esa misma línea (si no existe sitio en esa línea, la caja baja una línea y se muestra lo más a la izquierda posible en esa nueva línea). El resto de los elementos adyacentes se adaptan y *fluyen* alrededor de la caja flotante.

El valor right tiene un funcionamiento idéntico, salvo que, en este caso, la caja se desplaza hacia la derecha. El valor none permite anular el posicionamiento flotante de forma que el elemento se muestre en su posición original.

Los elementos que se encuentran alrededor de una caja flotante adaptan sus contenidos para que fluyan alrededor del elemento posicionado:



Elementos que fluyen alrededor de un elemento posicionado mediante float

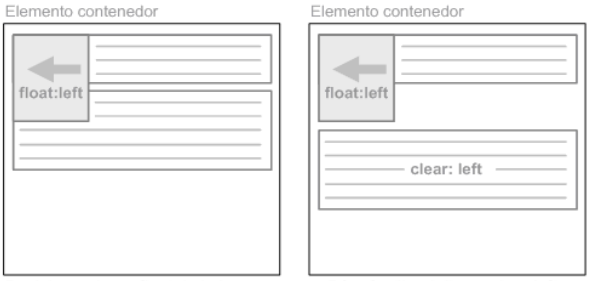
La regla CSS que se aplica en la imagen del ejemplo anterior es:

```
img {
```

```
 float: left;
}
```

Uno de los principales motivos para la creación del posicionamiento float fue precisamente la posibilidad de colocar imágenes alrededor de las cuales fluye el texto.

CSS permite controlar la forma en la que los contenidos fluyen alrededor de los contenidos posicionados mediante float. De hecho, en muchas ocasiones es admisible que algunos contenidos fluyan alrededor de una imagen, pero el resto de los contenidos deben mostrarse en su totalidad sin fluir alrededor de la imagen:



Posicionamiento float de la imagen — Párrafo "limpio" con clear:left

Forzando a que un elemento no fluya alrededor de otro elemento posicionado mediante float

La propiedad clear permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante. La regla CSS que se aplica al segundo párrafo del ejemplo anterior es la siguiente:

```
<p style="clear: left;">...</p>
```

La definición formal de la propiedad clear se muestra a continuación:

clear	
Valores	none | left | right | both
Descripción	Indica el lado del elemento que no debe ser adyacente a ninguna caja flotante

La propiedad clear indica el lado del elemento HTML que no debe ser adyacente a ninguna caja posicionada de forma flotante. Si se indica el valor left, el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ninguna caja flotante en el lado izquierdo.

La especificación oficial de CSS explica este comportamiento como *"un desplazamiento descendente hasta que el borde superior del elemento esté por debajo del borde inferior de cualquier elemento flotante hacia la izquierda"*.

Si se indica el valor right, el comportamiento es análogo, salvo que en este caso se tienen en cuenta los elementos desplazados hacia la derecha.

El valor both despeja los lados izquierdo y derecho del elemento, ya que desplaza el elemento de forma descendente hasta que el borde superior se encuentre por debajo del borde inferior de cualquier elemento flotante hacia la izquierda o hacia la derecha.

Como se verá más adelante, la propiedad clear es imprescindible cuando se crean las estructuras de las páginas web complejas.

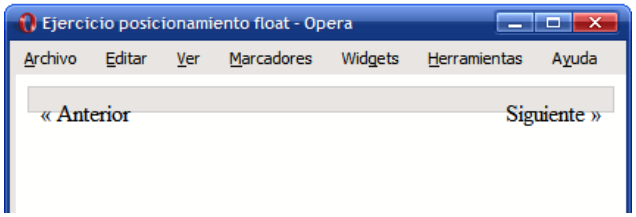
Si se considera el siguiente código CSS y HTML:

```
#paginacion {
 border: 1px solid #CCC;
 background-color: #E0E0E0;
 padding: .5em;
```

```
}

.derecha   { float: right; }
.izquierda { float: left;  }
<div id="paginacion">
  <span class="izquierda">&laquo; Anterior</span>
  <span class="derecha">Siguiente &raquo;</span>
</div>
```

Si se visualiza la página anterior en cualquier navegador, el resultado es el que muestra la siguiente imagen:



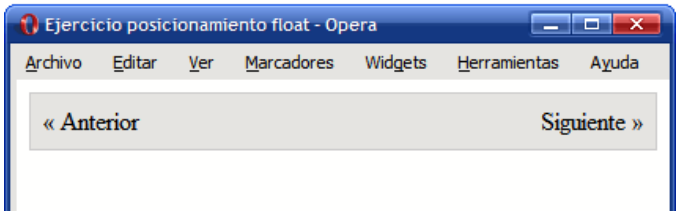
Visualización incorrecta de dos elementos posicionados mediante float

Los elementos Anterior y Siguiente *se salen* de su elemento contenedor y el resultado es visualmente incorrecto. El motivo de este comportamiento es que un elemento posicionado de forma flotante ya no pertenece al flujo normal de la página HTML. Por tanto, el elemento en realidad no encierra ningún contenido y por eso se visualiza incorrectamente.

La solución consiste en utilizar la propiedad overflow sobre el elemento contenedor:

```
.paginacion {
  border: 1px solid #CCC;
  background-color: #E0E0E0;
  padding: .5em;
  overflow: auto;
}
 .derecha   { float: right; }
.izquierda { float: left;  }
```

Si se visualiza de nuevo la página anterior en cualquier navegador, el resultado ahora sí que es el esperado:



Visualización correcta de dos elementos posicionados mediante float

2.2 DISPLAY

Display puede que sea la propiedad CSS con más funcionalidades, ahora vamos a ver una de ellas que es la maquetación. Decimos que la propiedad display sirve para maquetar debido a que puede convertir un elemento en bloque en un elemento en línea.

Lo que no permite hacer display es lo mismo que hacemos con float, pero en este caso el elemento no pierde el flujo de la página y por tanto sin la necesidad de utilizar overflow.

Al darle el valor inline-block a la propiedad display conseguimos que un elemento de bloque se convierta en un elemento en línea, pero con todas las características de un elemento de bloque (ancho, alto y margen vertical)

```
div {
  display: inline-block;
}
```

Vamos a tener un problema con el margen vertical. Cuando las cajas tengan diferentes alturas y necesitemos que todas empiecen en el mismo top, tendremos que colocarlo manualmente, para ello utilizaremos la propiedad vertical-align.

```
div {
  display: inline-block;
 vertical-align: top;
}
```

3 VISUALIZACIÓN

Para poder controlar cuando debe aparecer un elemento en pantalla y cuando debe desaparecer, además del comportamiento del elemento cuando desaparece CSS nos ofrece las siguientes propiedades.

3.1 PROPIEDADES DISPLAY Y VISIBILITY

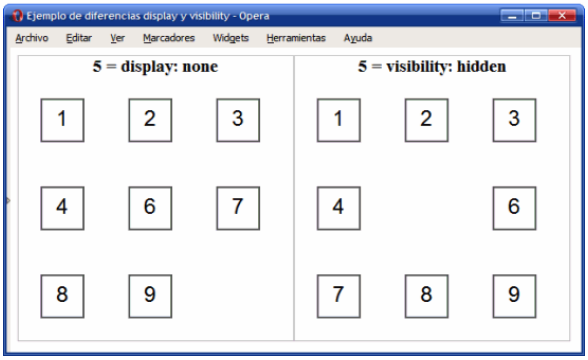
Como ya hemos visto display nos permite realizar diferentes acciones, ahora vamos a ver otra funcionalidad.

Las propiedades display y visibility controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página. Las podemos utilizar junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes.

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:



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.

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

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.

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.

Para volver a visualizar un elemento le daremos valor block o inline a la propiedad display y el elemento volverá a mostrarse en pantalla.

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

visibility	
Valores	visible | hidden
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 los 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.

3.2 RELACIÓN ENTRE DISPLAY, FLOAT Y POSITION

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

1. Si display vale none, se ignoran las propiedades float y position y la caja no se muestra en la página.
2. 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.
3. 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.

3.3 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.

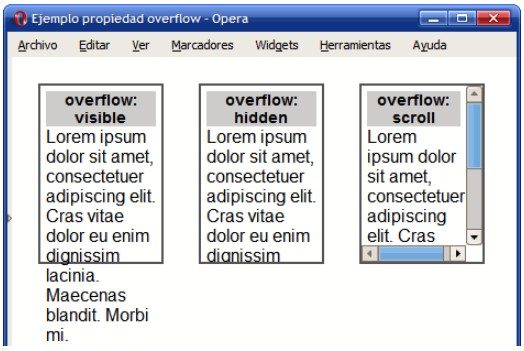
CSS define la propiedad overflow para controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.

overflow	
Valores	visible | hidden | scroll | auto
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:



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;
 font: 1em Arial, Helvetica, sans-serif;
}
```

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

3.4 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.

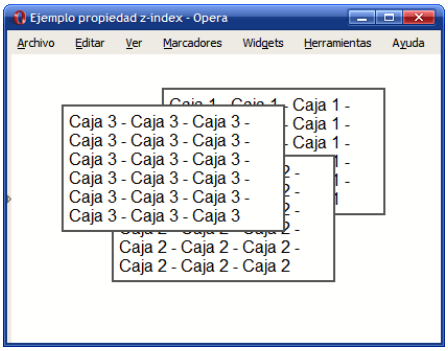
A continuación, se muestra la definición formal de la propiedad z-index:

z-index	
Valores	auto | numero
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:



Ejemplo de propiedad z-index

El código HTML y CSS del ejemplo anterior es el siguiente:

```
div { position: absolute; }
#caja1 { z-index: 5;  top: 1em; left: 8em;}
#caja2 { z-index: 15;  top: 5em; left: 5em;}
#caja3 { z-index: 25;  top: 2em; left: 2em;}

<div id="caja1">Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 -
Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1</div>

<div id="caja2">Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 -
Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2</div>

<div id="caja3">Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 -
Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3</div>
```

La propiedad z-index sólo tiene efecto en los elementos posicionados, por lo que es obligatorio que la propiedad z-index vaya acompañada de la propiedad position. Si debes posicionar un elemento, pero no quieres moverlo de su posición original ni afectar al resto de elementos de la página, puedes utilizar el posicionamiento relativo (position: relative).