

Material Imprimible

Curso de CSS para desarrollo Web

Módulo 6: Posicionamiento

Contenidos:

- Layout. Display. Visibility.
- Position Static. Relative. Fixed. Absolute. Sticky.
- Texto en imagen. Overflow. Scroll. Float. Clear
- Clearfix. Menú de Navegación. Flexbox. Barra de navegación
- Divisores Fronterizos. Navegación desplegable. Responsive.

Diseño (Layout)CSS : la propiedad de visualización

La `display` propiedad es la propiedad CSS más importante para controlar el diseño.

La propiedad de visualización

La `display` propiedad especifica si / cómo se muestra un elemento.

Cada elemento HTML tiene un valor de visualización predeterminado según el tipo de elemento que sea. El valor de visualización predeterminado para la mayoría de los elementos es `block` o `inline`.

Elementos a nivel de bloque

Un elemento de nivel de bloque siempre comienza en una nueva línea y ocupa todo el ancho disponible (se extiende hacia la izquierda y hacia la derecha tanto como sea posible).

El elemento `<div>` es un elemento de nivel de bloque.

Ejemplos de elementos a nivel de bloque:

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- `<form>`
- `<encabezado>`
- `<footer>`
- `<sección>`

Elementos en línea

Un elemento en línea no comienza en una nueva línea y solo ocupa el ancho que sea necesario.

Este es un elemento `` en línea dentro de un párrafo.

Ejemplos de elementos en línea:

- ``
- `<a>`
- ``

Pantalla: ninguna;

`display: none;` se usa comúnmente con JavaScript para ocultar y mostrar elementos sin eliminarlos ni recrearlos.

El `<script>` elemento se usa `display: none;` por defecto.

Anular el valor de visualización predeterminado

Como se mencionó, cada elemento tiene un valor de visualización predeterminado. Sin embargo, puede anular esto.

Cambiar un elemento en línea a un elemento de bloque, o viceversa, puede ser útil para hacer que la página se vea de una manera específica y seguir los estándares web.

Un ejemplo común es hacer `` elementos en línea para menús horizontales:

```
<!DOCTYPE html>
<html>
<head>
<style>
li {
  display: inline;
}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>

</body>
</html>
```

El siguiente ejemplo muestra elementos `` como elementos de bloque:

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
span {
  display: block;
}
</style>
</head>
<body>
```

``A display property with a value of "block" results in`` ``a line break between the two elements.``

```
</body>
</html>
```

El siguiente ejemplo muestra elementos `<a>` como elementos de bloque:

```
<!DOCTYPE html>
<html>
<head>
<style>
a {
  display: block;
}
</style>
</head>
<body>
```

`<p>`Display links as block elements:`</p>`

```
<a href="/html/default.asp" target="_blank">HTML</a>
<a href="/css/default.asp" target="_blank">CSS</a>
<a href="/js/default.asp" target="_blank">JavaScript</a>

</body>
```

```
</html>
```

Se puede ocultar un elemento estableciendo la `display` propiedad en `none`. El elemento estará oculto y la página se mostrará como si el elemento no estuviera allí

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1.hidden {
```

```
  display: none;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a visible heading</h1>
```

```
<h1 class="hidden">This is a hidden heading</h1>
```

```
<p>Notice that the h1 element with display: none; does not take up any space.</p>
```

```
</body>
```

```
</html>
```

`visibility:hidden`; También esconde un elemento.

Sin embargo, el elemento seguirá ocupando el mismo espacio que antes. El elemento estará oculto, pero aún afectará el diseño:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1.hidden {
```

```
  visibility: hidden;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>

</body>
</html>
```

Más ejemplos

Diferencias entre pantalla: ninguna; y visibilidad: oculta;

Este ejemplo muestra display: none; versus visibilidad: oculto;

```
<!DOCTYPE html>
<html>
<head>
<style>
.imgbox {
  float: left;
  text-align: center;
  width: 120px;
  border: 1px solid gray;
  margin: 4px;
  padding: 6px;
}

button {
  width: 100%;
}
</style>
</head>
<body>

<h3>Difference between display:none and visibility: hidden</h3>
```

<p>visibility:hidden hides the element, but it still takes up space in the layout.</p>

<p>display:none removes the element from the document. It does not take up any space.</p>

```
<div class="imgbox" id="imgbox1">Box 1<br>
  
  <button onclick="removeElement()">Remove</button>
</div>
```

```
<div class="imgbox" id="imgbox2">Box 2<br>
  
  <button onclick="changeVisibility()">Hide</button>
</div>
```

```
<div class="imgbox">Box 3<br>
  
  <button onclick="resetElement()">Reset All</button>
</div>
```

```
<script>
function removeElement() {
  document.getElementById("imgbox1").style.display = "none";
}

function changeVisibility() {
  document.getElementById("imgbox2").style.visibility = "hidden";
}

function resetElement() {
  document.getElementById("imgbox1").style.display = "block";
  document.getElementById("imgbox2").style.visibility = "visible";
}
</script>
```

```
</body>  
</html>
```

Uso de CSS junto con JavaScript para mostrar contenido

Este ejemplo muestra cómo usar CSS y JavaScript para mostrar un elemento al hacer clic.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#panel, .flip {  
  font-size: 16px;  
  padding: 10px;  
  text-align: center;  
  background-color: #4CAF50;  
  color: white;  
  border: solid 1px #a6d8a8;  
  margin: auto;  
}  
  
#panel {  
  display: none;  
}  
</style>  
</head>  
<body>  
  
<p class="flip" onclick="myFunction()">Click to show panel</p>  
  
<div id="panel">  
  <p>Este panel contiene un elemento div, que está oculto de forma predeterminada  
(pantalla: ninguno). </p>  
  <p> Está diseñado con CSS y usamos JavaScript para mostrarlo (display: block). </p>
```

<p> Cómo funciona: Tenga en cuenta que el elemento p con class = "flip" tiene un atributo onclick adjunto. Cuando el usuario hace clic en el elemento p, se ejecuta una función llamada myFunction (), que cambia el estilo del div con id = "panel" de display: none (hidden) para mostrar: block (visible). </p>

<p> Aprenderá más sobre JavaScript en nuestro Tutorial de JavaScript.</p>
</div>

```
<script>  
function myFunction() {  
  document.getElementById("panel").style.display = "block";  
}  
</script>  
  
</body>  
</html>
```

Posicionar texto en una imagen

Cómo colocar texto sobre una imagen

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.container {  
  position: relative;  
}
```

```
.topleft {  
  position: absolute;  
  top: 8px;  
  left: 16px;  
  font-size: 18px;  
}
```

```
.topright {  
  position: absolute;
```

```
top: 8px;
right: 16px;
font-size: 18px;
}
.bottomleft {
  position: absolute;
  bottom: 8px;
  left: 16px;
  font-size: 18px;
}
.bottomright {
  position: absolute;
  bottom: 8px;
  right: 16px;
  font-size: 18px;
}
.center {
  position: absolute;
  left: 0;
  top: 50%;
  width: 100%;
  text-align: center;
  font-size: 18px;
}
```

```
img {
  width: 100%;
  height: auto;
  opacity: 0.3;
}
```

```
</style>
</head>
<body >
```

```
<h2>Image Text</h2>
```

```
<p>Add some text to an image in the top left corner:</p>
```

```
<div class="container">
```

```
  
```

```
  <div class="topleft">Top Left</div>
```

```
<div class="topright">Top Right</div>
```

```
<div class="center">Centered</div>
```

```
<div class="bottomright">Bottom Right</div>
```

```
<div class="bottomleft">Bottom Left</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Todas las propiedades de posicionamiento CSS

Propiedad	Descripción
bottom	Establece el borde del margen inferior para un cuadro posicionado
clip	Recorta un elemento absolutamente posicionado
left	Establece el borde del margen izquierdo para un cuadro posicionado
position	Especifica el tipo de posicionamiento para un elemento
right	Establece el borde de margen derecho para un cuadro posicionado
top	Establece el borde del margen superior para un cuadro posicionado
z-index	Establece el orden de apilamiento de un elemento

CSS desbordamiento

La `overflow` propiedad especifica si se debe recortar el contenido o agregar barras de desplazamiento cuando el contenido de un elemento es demasiado grande para caber en el área especificada.

La `overflow` propiedad tiene los siguientes valores:

- `visible` - Defecto. El desbordamiento no está recortado. El contenido se representa fuera del cuadro del elemento.
- `hidden` - El desbordamiento se recorta y el resto del contenido será invisible.
- `scroll` - El desbordamiento se recorta y se agrega una barra de desplazamiento para ver el resto del contenido
- `auto` - Similar a `scroll`, pero agrega barras de desplazamiento solo cuando es necesario

desbordamiento: visible

Por defecto, el desbordamiento es `visible`, lo que significa que no está recortado y se representa fuera del cuadro del elemento:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  background-color: #eee;
```

```
  width: 200px;
```

```
  height: 50px;
```

```
  border: 1px dotted black;
```

```
  overflow: visible;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>CSS Overflow</h2>
```

```
<p>By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box:</p>
```

```
<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>
```

```
</body>
```

```
</html>
```

desbordamiento: oculto

Con el `hidden` valor, el desbordamiento se recorta y el resto del contenido se oculta

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  background-color: #eee;
```

```
  width: 200px;
```

```
  height: 50px;
```

```
  border: 1px dotted black;
```

```
  overflow: hidden;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>CSS Overflow</h2>
```

```
<p>With the hidden value, the overflow is clipped, and the rest of the content is hidden:</p>
```

```
<p>Try to remove the overflow property to understand how it works.</p>
```

```
<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>
```

```
</body>  
</html>
```

desbordamiento: desplazamiento

Al establecer el valor en `scroll`, el desbordamiento se recorta y se agrega una barra de desplazamiento para desplazarse dentro del cuadro. Tenga en cuenta que esto agregará una barra de desplazamiento tanto horizontal como verticalmente (incluso si no la necesita)

desbordamiento: auto

El `auto` valor es similar a `scroll`, pero agrega barras de desplazamiento solo cuando es necesario

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  background-color: #eee;
```

```
  width: 200px;
```

```
  height: 50px;
```

```
  border: 1px dotted black;
```

```
  overflow: auto;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>CSS Overflow</h2>
```

```
<p>The auto value is similar to scroll, only it add scrollbars when necessary:</p>
```

```
<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>
```

```
</body>  
</html>
```

Diseño CSS : flotante(float) y claro(clear)

La `float` propiedad CSS especifica cómo debe flotar un elemento.

La `clear` propiedad CSS especifica qué elementos pueden flotar junto al elemento borrado y de qué lado

La propiedad flotante

La `float` propiedad se usa para posicionar y formatear contenido, por ejemplo, dejar que una imagen flote hacia la izquierda en el texto en un contenedor.

La `float` propiedad puede tener uno de los siguientes valores:

- `left`: el elemento flota a la izquierda de su contenedor
- `right`: el elemento flota a la derecha de su contenedor
- `none`: el elemento no flota (se mostrará justo donde aparece en el texto). Esto es por defecto
- `heredar`: el elemento hereda el valor flotante de su padre

En su uso más simple, la `float` propiedad se puede usar para ajustar el texto alrededor de las imágenes.

Ejemplo: flotante:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.right {  
  float: right;  
}  
.left {  
  float: left;  
}  
.none {  
  float: none;
```

```
}  
</style>  
</head>  
<body bgcolor="black">
```

<p>In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.</p>

```
<p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>

<p>In this example, the image will float to the left in the paragraph, and the text in the paragraph will wrap around the image.</p>

```
<p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>


```
<p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>

```
<p>In this example, the image will float to the left in the paragraph, and the text in the paragraph will wrap around the image.</p>
```

```
</body>
```

```
</html>
```

Diseño CSS : clear y clearfix

La propiedad limpiar

La **clear** propiedad especifica qué elementos pueden flotar al lado del elemento borrado y de qué lado.

La **clear** propiedad puede tener uno de los siguientes valores:

- none: permite elementos flotantes en ambos lados. Esto es por defecto
- left: no se permiten elementos flotantes en el lado izquierdo
- right- No se permiten elementos flotantes en el lado derecho
- ambos: no se permiten elementos flotantes ni en el lado izquierdo ni en el derecho
- heredar: el elemento hereda el valor clear de su padre

La forma más común de usar la **clear** propiedad es después de haber usado una **float** propiedad en un elemento.

Al despejar flotadores, debe hacer coincidir el despeje con el flotador: si un elemento flota a la izquierda, entonces debe despejar a la izquierda. Su elemento flotante continuará flotando, pero el elemento borrado aparecerá debajo de él en la página web.

El siguiente ejemplo borra el flotador a la izquierda. Significa que no se permiten elementos flotantes en el lado izquierdo (del div)

```
<!DOCTYPE html>
<html>
<head>
<style>
.div1 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #73AD21;
}

.div2 {
  border: 1px solid red;
}

.div3 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #73AD21;
}

.div4 {
  border: 1px solid red;
  clear: left;
}
</style>
</head>
<body>
```

```
<h2>Without clear</h2>
```

```
<div class="div1">div1</div>
```

```
<div class="div2">div2 - Notice that div2 is after div1 in the HTML code. However, since  
div1 floats to the left, the text in div2 flows around div1.</div>
```

```
<br><br>
```

```
<h2>With clear</h2>
```

```
<div class="div3">div3</div>
```

```
<div class="div4">div4 - Here, clear: left; moves div4 down below the floating div3. The  
value "left" clears elements floated to the left. You can also clear "right" and "both".</div>
```

```
</body>
```

```
</html>
```

El clearfix Hack

Si un elemento es más alto que el elemento que lo contiene y está flotando, se "desbordará" fuera de su contenedor:

Luego podemos agregar `overflow: auto;` al elemento contenedor para solucionar este problema:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  border: 3px solid #4CAF50;
```

```
  padding: 5px;
```

```
}
```

```
.img1 {
```

```
  float: right;
```

```
}
```

```
.clearfix {
```

```
  overflow: auto;
```

```
}  
  
.img2 {  
  float: right;  
}  
</style>  
</head>  
<body>
```

<p>In this example, the image is taller than the element containing it, and it is floated, so it overflows outside of its container:</p>

```
<div>  
    
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et  
  dictum interdum...  
</div>
```

<p style="clear:right">Add a clearfix class with overflow: auto; to the containing element, to fix this problem:</p>

```
<div class="clearfix">  
    
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et  
  dictum interdum...  
</div>
```

```
</body>  
</html>
```

El `overflow: auto` clearfix funciona bien, siempre y cuando usted es capaz de mantener el control de sus márgenes y el relleno (otra cosa que pueda ver las barras de desplazamiento). El **nuevo y moderno corte clearfix** sin embargo, es más seguro de usar, y el siguiente código se utiliza para la mayoría de las páginas web:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 3px solid #4CAF50;
  padding: 5px;
}

.img1 {
  float: right;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

.img2 {
  float: right;
}
</style>
</head>
<body>
```

<p>In this example, the image is taller than the element containing it, and it is floated, so it overflows outside of its container:</p>

```
<div>
  <h2>Without Clearfix</h2>
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et
  dictum interdum...
```

```
</div>
```

```
<p style="clear:right">Add the clearfix hack to the containing element, to fix this  
problem:</p>
```

```
<div class="clearfix">  
  <h2>With Clearfix</h2>  
    
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et  
dictum interdum...  
</div>
```

```
</body>
```

```
</html>
```

Menú de Navegación

Úselo `float` con una lista de hipervínculos para crear un menú horizontal

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
ul {
```

```
  list-style-type: none;
```

```
  margin: 0;
```

```
  padding: 0;
```

```
  overflow: hidden;
```

```
  background-color: #333;
```

```
}
```

```
li {
```

```
  float: left;
```

```
}
```

```
li a {  
  display: inline-block;  
  color: white;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}
```

```
li a:hover {  
  background-color: #111;  
}
```

```
.active {  
  background-color: red;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a href="#home" class="active">Home</a></li>
```

```
<li><a href="#news">News</a></li>
```

```
<li><a href="#contact">Contact</a></li>
```

```
<li><a href="#about">About</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

Ejemplo de diseño web

También es común hacer diseños web completos utilizando la `float` propiedad:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
* {
  box-sizing: border-box;
}

.header, .footer {
  background-color: grey;
  color: white;
  padding: 15px;
}

.column {
  float: left;
  padding: 15px;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

.menu {
  width: 25%;
}

.content {
  width: 75%;
}

.menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
```

```
}
```

```
.menu li {  
  padding: 8px;  
  margin-bottom: 8px;  
  background-color: #33b5e5;  
  color: #ffffff;  
}
```

```
.menu li:hover {  
  background-color: #0099cc;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="header">
```

```
  <h1>Chania</h1>
```

```
</div>
```

```
<div class="clearfix">
```

```
  <div class="column menu">
```

```
    <ul>
```

```
      <li>The Flight</li>
```

```
      <li>The City</li>
```

```
      <li>The Island</li>
```

```
      <li>The Food</li>
```

```
    </ul>
```

```
  </div>
```

```
<div class="column content">
```

```
  <h1>The City</h1>
```

```
  <p>Chania is the capital of the Chania region on the island of Crete. The city can  
be divided in two parts, the old town and the modern city.</p>
```

```
<p>You will learn more about web layout and responsive web pages in a later chapter.</p>
```

```
</div>
```

```
</div>
```

```
<div class="footer">
```

```
<p>Footer Text</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

Todas las propiedades de flotación CSS

Propiedad	Descripción
box-sizing	Define cómo se calculan el ancho y el alto de un elemento: si deben incluir relleno y bordes, o no
clear	Especifica qué elementos pueden flotar al lado del elemento borrado y de qué lado
float	Especifica cómo debe flotar un elemento
overflow	Especifica qué sucede si el contenido desborda el cuadro de un elemento
overflow-x	Especifica qué hacer con los bordes izquierdo / derecho del contenido si desborda el área de contenido del elemento
overflow-y	Especifica qué hacer con los bordes superior / inferior del contenido si desborda el área de contenido del elemento

Barras de navegación

Tener una navegación fácil de usar es importante para cualquier sitio web.

Con CSS puedes transformar menús HTML aburridos en atractivas barras de navegación.

Barra de navegación = Lista de enlaces

Una barra de navegación necesita HTML estándar como base.

En nuestros ejemplos construiremos la barra de navegación a partir de una lista HTML estándar.

Una barra de navegación es básicamente una lista de enlaces, por lo que usar los elementos `` y `` tiene mucho sentido:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<ul>
```

```
<li><a href="#home">Home</a></li>
```

```
<li><a href="#news">News</a></li>
```

```
<li><a href="#contact">Contact</a></li>
```

```
<li><a href="#about">About</a></li>
```

```
</ul>
```

```
<p>Note: We use href="#" for test links. In a real web site this would be URLs.</p>
```

```
</body>
```

```
</html>
```

Divisores fronterizos

Agregue la `border-right` propiedad a `` para crear divisores de enlaces:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
ul {
```

```
list-style-type: none;
```

```
margin: 0;
```

```
padding: 0;
overflow: hidden;
background-color: #333;
}

li {
float: left;
border-right: 1px solid #bbb;
}

li:last-child {
border-right: none;
}

li a {
display: block;
color: white;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}

li a:hover:not(.active) {
background-color: #111;
}

.active {
background-color: #4CAF50;
}
</style>
</head>
<body>

<ul>
```

```
<li><a class="active" href="#home">Home</a></li>
<li><a href="#news">News</a></li>
<li><a href="#contact">Contact</a></li>
<li style="float:right"><a href="#about">About</a></li>
</ul>
```

```
</body>
```

```
</html>
```

Barra de navegación desplegable

Cómo agregar un menú desplegable dentro de una barra de navegación.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: left;
}

li a, .dropbtn {
  display: inline-block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
```

```
li a:hover, .dropdown:hover .dropbtn {
  background-color: red;
}

li.dropdown {
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;
}

.dropdown-content a:hover {background-color: #f1f1f1;}

.dropdown:hover .dropdown-content {
  display: block;
}
</style>
</head>
<body>
```

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li class="dropdown">
    <a href="javascript:void(0)" class="dropbtn">Dropdown</a>
    <div class="dropdown-content">
      <a href="#">Link 1</a>
      <a href="#">Link 2</a>
      <a href="#">Link 3</a>
    </div>
  </li>
</ul>
```

```
<h3>Dropdown Menu inside a Navigation Bar</h3>
<p>Hover over the "Dropdown" link to see the dropdown menu.</p>
```

```
</body>
</html>
```

Desplegables CSS

Crea un menú desplegable hoverable con CSS.

Demostración: ejemplos desplegados

Mueva el mouse sobre los ejemplos a continuación:

```
<!DOCTYPE html>
<html>
<head>
<style>
.dropdown {
  position: relative;
  display: inline-block;
}

```

```
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: #f9f9f9;  
  min-width: 160px;  
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);  
  padding: 12px 16px;  
  z-index: 1;  
}
```

```
.dropdown:hover .dropdown-content {  
  display: block;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Hoverable Dropdown</h2>
```

```
<p>Move the mouse over the text below to open the dropdown content.</p>
```

```
<div class="dropdown">
```

```
  <span>Mouse over me</span>
```

```
  <div class="dropdown-content">
```

```
    <p>Hello World!</p>
```

```
  </div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Ejemplo explicado

HTML) Use cualquier elemento para abrir el contenido desplegable, por ejemplo, un elemento `` o `<button>`.

Use un elemento contenedor (como `<div>`) para crear el contenido desplegable y agregar lo que desee dentro de él.

Envuelva un elemento `<div>` alrededor de los elementos para colocar el contenido desplegable correctamente con CSS.

La `.dropdown` clase usa `position:relative`, lo cual es necesario cuando queremos que el contenido desplegable se coloque justo debajo del botón desplegable (usando `position:absolute`).

La `.dropdown-content` clase contiene el contenido desplegable real. Está oculto de forma predeterminada y se mostrará al pasar el mouse (ver más abajo). Tenga en cuenta que `min-width` se establece en 160px. Siéntase libre de cambiar esto.

Sugerencia: Si desea que el ancho del contenido desplegable sea tan ancho como el botón desplegable, configure el `width:100%` (y `overflow:auto` habilite el desplazamiento en pantallas pequeñas).

En lugar de usar un borde, hemos usado la `box-shadow` propiedad CSS para hacer que el menú desplegable se vea como una "tarjeta".

El `:hover` selector se usa para mostrar el menú desplegable cuando el usuario mueve el mouse sobre el botón desplegable.

Menú desplegable

Cree un menú desplegable que permita al usuario elegir una opción de una lista:

Este ejemplo es similar al anterior, excepto que agregamos enlaces dentro del cuadro desplegable y los diseñamos para que se ajusten a un botón desplegable con estilo:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.dropbtn {
```

```
  background-color: #4CAF50;
```

```
  color: white;
```

```
  padding: 16px;
```

```
  font-size: 16px;
```

```
  border: none;
```

```
    cursor: pointer;
}

.dropdown {
    position: relative;
    display: inline-block;
}

.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
}

.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}

.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content {
    display: block;
}

.dropdown:hover .dropbtn {
    background-color: #3e8e41;
}
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Dropdown Menu</h2>
```

```
<p>Move the mouse over the button to open the dropdown menu.</p>
```

```
<div class="dropdown">
```

```
  <button class="dropbtn">Dropdown</button>
```

```
  <div class="dropdown-content">
```

```
    <a href="#">Link 1</a>
```

```
    <a href="#">Link 2</a>
```

```
    <a href="#">Link 3</a>
```

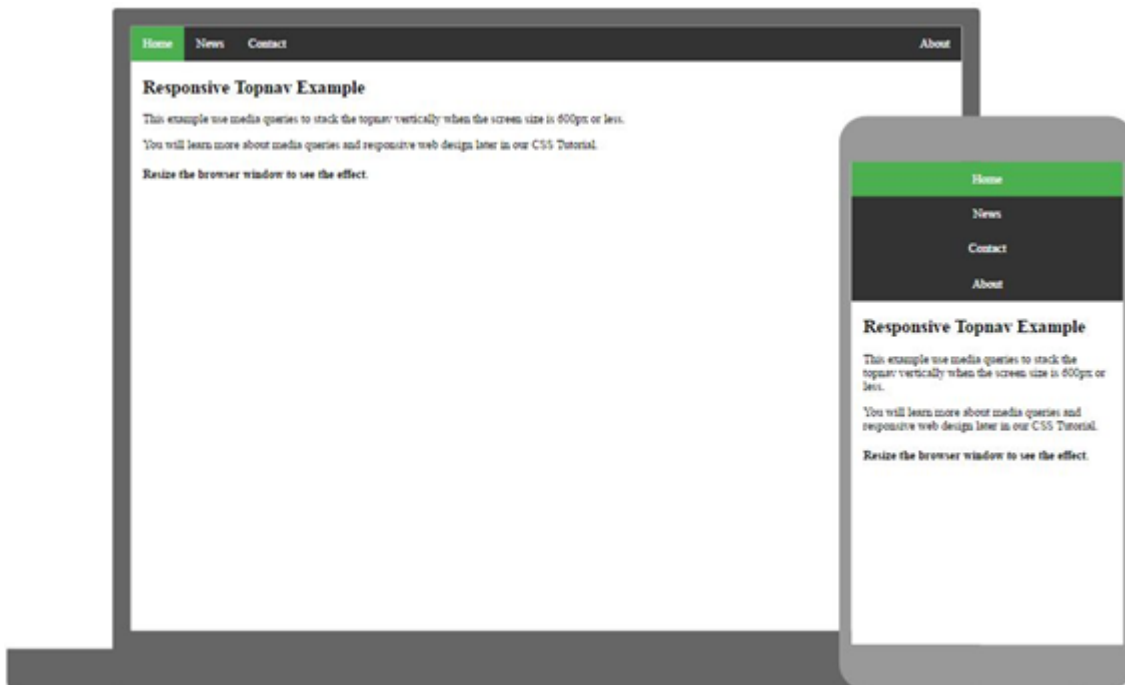
```
  </div>
```

```
</div>
```

```
<p><strong>Note:</strong> We use href="#" for test links. In a real web site this would  
be URLs.</p>
```

```
</body>
```

```
</html>
```



```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body {margin: 0;}

ul.topnav {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

ul.topnav li {float: left;}

ul.topnav li a {

```

```
display: block;
color: white;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}
```

```
ul.topnav li a:hover:not(.active) {background-color: #111;}
```

```
ul.topnav li a.active {background-color: #4CAF50;}
```

```
ul.topnav li.right {float: right;}
```

```
@media screen and (max-width: 600px) {
  ul.topnav li.right,
  ul.topnav li {float: none;}
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul class="topnav">
```

```
  <li><a class="active" href="#home">Home</a></li>
```

```
  <li><a href="#news">News</a></li>
```

```
  <li><a href="#contact">Contact</a></li>
```

```
  <li class="right"><a href="#about">About</a></li>
```

```
</ul>
```

```
<div style="padding:0 16px;">
```

```
  <h2>Responsive Topnav Example</h2>
```

```
  <p>This example use media queries to stack the topnav vertically when the screen size is 600px or less.</p>
```

```
  <p>You will learn more about media queries and responsive web design later in our CSS Tutorial.</p>
```

```
<h4>Resize the browser window to see the effect.</h4>  
</div>  
  
</body>  
</html>
```

Fuentes: http://www.w3bai.com/es/css/css_positioning.html
http://www.w3bai.com/es/css/css_display_visibility.html