

# Lenguajes de programación HTML y CSS

## Parte 3: Diseño del sitio

Hemos aprendido a crear páginas con HTML básico, para cambiar el formato con CSS, ahora vamos al diseño de nuestro sitio. Al final de esta sección, vamos a llegar a nuestro primer sitio completo, diseñado como lo queremos.

### Estructurar la página

Nos estamos acercando más y más a la meta. Si nuestras páginas web no se parecen bastante a los sitios web que conocemos es que nos falta el conocimiento para hacer el diseño.

En general, una página web consta de una cabecera (arriba), los menús de navegación (arriba, a los lados) las diferentes secciones en el centro y un pie de página (foto inferior).

En este capítulo vamos a buscar nuevas etiquetas de HTML dedicadas a la estructuración del sitio. Estas etiquetas se han introducido por HTML5 (no existían antes) y nos van a permitir decir: "Esta es mi cabecera", "Esta es mi navegación ", etc.

Por ahora, ni siquiera harás el diseño de página. Estamos en realidad preparando nuestro documento HTML para descubrir el diseño en los siguientes capítulos.

### Etiquetas estructurales de HTML5

Voy a presentar las nuevas etiquetas introducidas en HTML5 para estructurar nuestras páginas. Esto no va a cambiar tanto el aspecto de nuestro sitio todavía, pero estará bien construido y listo para entonces ser formateado.

#### **<header>: cabecera**

La mayoría de los sitios web suelen tener un encabezado, el encabezado se llama header en Inglés. Normalmente hay un logotipo, bandera, el lema de su sitio ... Tendrás que colocar esta información en el interior de la etiqueta <header>

Código : HTML

```
<header>  
<!-- Coloca aquí el contenido de la cabecera -->  
</header>
```

**<footer>: pie de página**

A diferencia de la cabecera, el pie de página es generalmente en la parte inferior del documento. Contiene información tal como enlaces de contacto, el nombre del autor, legales, etc.

```
<footer>
<- Colocar aquí el contenido de pie de página ->
</ footer>
```

**<nav>: Enlaces de navegación principales**

La etiqueta <nav> debe consolidar todos los enlaces primarios de navegación en el sitio. Por ejemplo puedes colocar el menú principal del sitio.

En general, el menú está diseñado como una lista con viñetas en el interior de la etiqueta <nav>

Código : HTML

```
<nav>
  <ul>
    <li><a href="index.html">Inicio</a></li>
    <li><a href="forum.html">Foro</a></li>
    <li><a href="contacto.html">Contacto</a></li>
  </ul>
</nav>
```

**<section>: una sección de la página**

La etiqueta <section> se utiliza para agrupar el contenido en función de su tema. Por lo general, comprende una porción de contenido en el centro de la página.

```
<section>
<h1>Mi sección de la página</h1>
<p>Bla bla bla bla</p>
</section>
```

Cada sección puede tener su título de nivel 1 partida (<h1>), así como la cabecera también puede contener un <h1>. Cada uno de estos bloques son independientes de los demás, no es ilógico encontrar varios títulos en el código de <h1>.

## **<aside>: Información adicional**

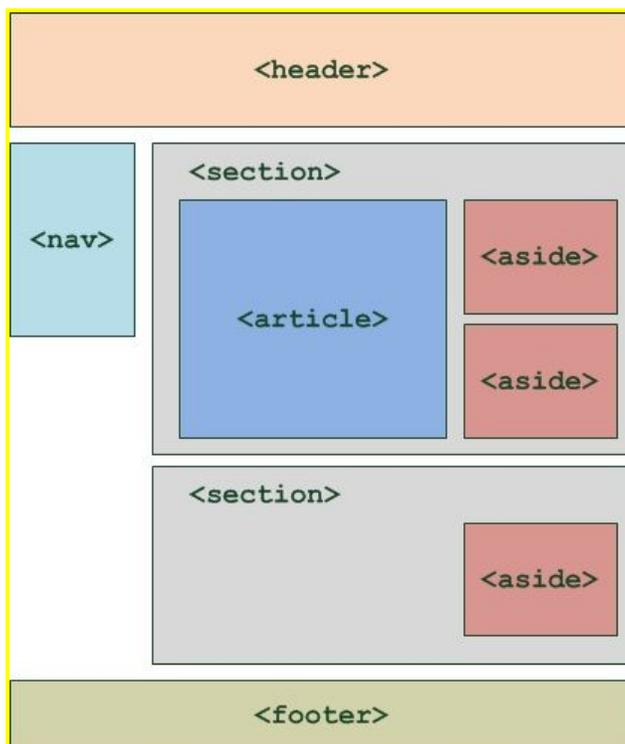
La etiqueta <aside> está diseñado para contener información adicional al documento que se visualiza. Esta información se coloca generalmente en el lado (aunque esto no es un requisito).

```
<aside>  
<!-- -Colocar aquí información complementaria-->  
</aside>
```

## **<article>: un artículo independiente**

La etiqueta se usa para abarcar una porción <article> generalmente autónoma de la página. Esta es parte de la página que podría por lo tanto reanudarse en otro sitio. Este es el caso por ejemplo de noticias (artículos de periódicos o blogs).

```
<article>  
<h1>Mi artículo</h1>  
<p>Bla bla bla bla</p>  
</article>
```



## **Un ejemplo concreto de la utilización de etiquetas**

Trata de usar las etiquetas que hemos descubierto para estructurar nuestro sitio web. El código siguiente muestra todas las etiquetas que hemos visto en una página web completa:

Código : HTML

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Página web de Rafael Barzanallana</title>
</head>
<body>
<header>
  <h1>Universidad de Murcia</h1>
  <h2>Apuntes de las asignaturas que imparto</h2>
</header>
<nav>
  <ul>
    <li><a href="#">DAWeb</a></li>
    <li><a href="#">IAT</a></li>
    <li><a href="#">Master Informática II</a></li>
  </ul>
</nav>
<section>
  <aside>
    <h1>Algo sobre mi</h1>
    <p>Profesor de la Universidad de Murcia</p>
  </aside>
  <article>
    <h1>Yo soy un gran escéptico</h1>
    <p>Soy contrario a todo tipo de creencias</p>
  </article>
</section>
<footer>
  <p>Bajo licencia Creative Commons<br />
  <a href="#">Contacto</a></p>
</footer>
</body>
</html>

```

## Resumen

- Se introdujeron varias etiquetas con HTML5 para delimitar las diferentes áreas que conforman la página web:

- <header>: cabecera;
- <footer>: pie de página;
- <nav>: enlaces principales de navegación;
- <section>: sección de la página;
- <aside>: la información;
- <article>: artículo independiente.

- Estas etiquetas se pueden anidar unas dentro de otras. Así, una sección puede tener su propio encabezado.

- Estas etiquetas no les importa el diseño. Sólo sirven para indicar al ordenador el texto que contienen. Se podría muy bien colocar la cabecera en la parte inferior de la página, si se desea.

## El modelo de cajas

Una página web puede ser vista como una sucesión de cajas apiladas, denominadas "bloques". La mayor parte de los elementos vistos en el capítulo previo son bloques: <header>, <article>, <nav> ... Pero ya sabíamos de otros bloques: párrafos <p>, títulos <h1>...

En este capítulo vamos a aprender a manipular estos bloques como cajas reales. Les daremos dimensiones, las ordenaremos para jugar con sus márgenes, también aprenderemos a manejar su contenido, por ejemplo para evitar que el texto de estos bloques se exceda.

Estos son los fundamentos que se necesitan para diseñar nuestro sitio web .

### Etiquetas de tipo block e inline

En HTML, la mayoría de las etiquetas se pueden clasificar en cualquiera de las dos categorías:

Etiquetas inline: este es el caso de los enlaces <a> </ a>.

Etiquetas block: este es el caso por ejemplo de los párrafos </ p>

Pero, ¿cómo puedo distinguir una etiqueta inline de una etiqueta de block?

En realidad es bastante fácil:

**block:** una etiqueta de tipo de bloque en la página web crea automáticamente un salto de línea antes y después. Es fácil imaginar un bloque. Una página web, de hecho está constituida por una serie de bloques cada uno a continuación de otro. Pero se verá que, además, es posible poner un bloque dentro de otro, lo que aumentará significativamente nuestra capacidad de crear el diseño de nuestro sitio.

**inline:** una etiqueta en línea está necesariamente dentro de una etiqueta de bloque. Una etiqueta en línea no crea una nueva línea, el texto que está dentro se escribe como un resultado del texto precedente en la misma línea (que es por eso hablamos de la etiqueta "inline").

A continuación se visualiza el concepto, donde las etiquetas de bloque se muestran con fondo azul y las inline se muestran con fondo blanco.

```
<h1>Título (bloque)</h1>
```

```
<p>Párrafo (bloque)Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi sit amet metus ac sem pharetra aliquam. <a>enlace (en línea)</a>Duis metus tortor, dignissim porttitor dolor ac, tincidunt tristique dolor. </p>
```

### Algunos ejemplos

Como ayuda para saber qué etiquetas son en línea y cuales son de bloque, una tabla sobre algunas etiquetas comunes.

<b>etiqueta block</b>	<b>etiqueta inline</b>
<p>	<em>
<footer>	<strong>
<h1>	<mark>
<h2>	<a>
<article>	<img>

### **Etiquetas universales**

Estas son las etiquetas que no tienen significado en particular (a diferencia de <p> que significa "párrafo" <strong> "importante", etc.).

El principal interés de estas etiquetas es que se pueden aplicar una class (o id) de CSS cuando no hay otra etiqueta adecuada.

Hay dos etiquetas genéricas y la única diferencia entre las dos es que una de ellas es en línea y la otra es de bloque:

- <span> </ span> (en línea);
- <div> </ div> (bloque).

### **Respetar la semántica**

Las etiquetas universales son prácticas en algunos casos, sí, pero ten cuidado de no abusar de ellas. Muchos *webmasters* hacen uso de <div> y <span> demasiado a menudo, olvidamos que hay otros marcadores más apropiados.

Dos ejemplos:

Un <span> inútil. Nunca debes ver esto <span class="importante">en el código mientras que exista la etiqueta <strong> utilizada para indicar importante.

Un <div> innecesario: <div class="título">. Esto es completamente absurdo, ya que hay etiquetas hechas especialmente para los títulos (<h1>, <h2> ...).

El resultado final (visual) es el mismo, sin embargo, las etiquetas genéricas no proporcionan ningún sentido a la página y no pueden ser entendidas por el ordenador. Siempre usa otras etiquetas más adaptadas en lo posible. Google lo aconseja para ayudarte a mejorar el posicionamiento dentro de las páginas de resultados de la búsqueda.

## Dimensiones

Aquí solo vamos a trabajar en las etiquetas de estilo de bloque. Para empezar, echemos un vistazo a el tamaño del bloque. A diferencia de un bloque en línea que tiene unas dimensiones precisas, tiene anchura y altura. Lo que hace, que se tengan dos propiedades CSS:

`width`: la anchura del bloque. Expresada en píxeles (px) o porcentaje (%).

`height`: la altura del bloque. También se expresa en píxeles (px) o porcentaje (%).

Ahora, vamos a añadir un poco de CSS para cambiar el ancho de los párrafos. El siguiente código CSS, indica, "quiero que mis párrafos tengan una anchura del 60%".

Código CSS

```
p
{
width: 60%;
}
```

Los porcentajes serán de utilidad para crear un diseño que se adapta automáticamente a la resolución de pantalla del visitante. Sin embargo, puede ser que necesites crear bloques con un tamaño específico en píxeles:

Código CSS

```
p
{
width: 350px;
}
```

## Mínimo y máximo

Podemos indicar que un bloque tenga las dimensiones mínimas y máximas. Esto es conveniente ya que permite definir dimensiones "límites" para que nuestro sitio se adapte a las diferentes resoluciones de pantalla de los visitantes:

`min-width`: ancho mínimo;

`min-height`: altura mínima;

`max-width`: anchura máxima;

`max-height`: altura máxima.

Por ejemplo, puedes solicitar que los párrafos ocupen el 50% de la anchura y obligarlos a tener

por lo menos 500 píxeles de ancho en todos los casos:

Código CSS

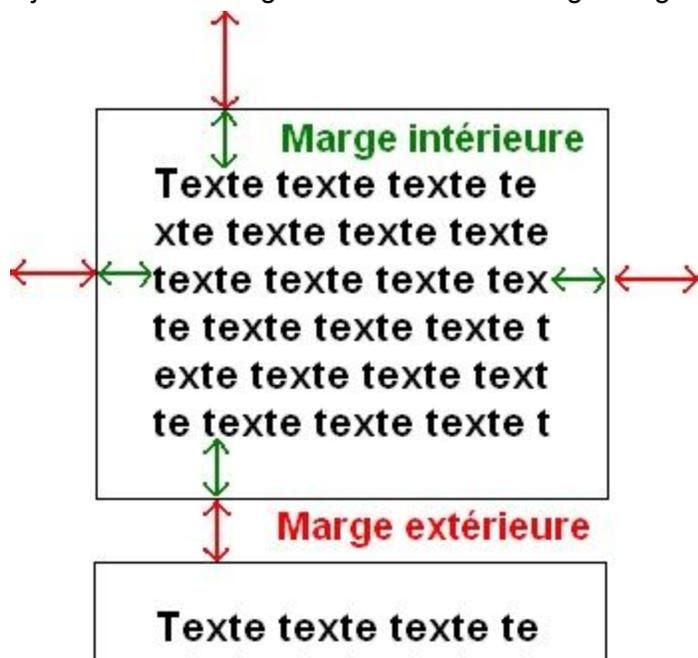
```
p
{
width: 50%;
min-width: 500px;
}
```

## Márgenes

Debes saber que todos los bloques tienen márgenes. Hay dos tipos de márgenes:

Márgenes interiores;  
Márgenes exteriores.

Fíjate bien en el diagrama mostrado en la figura siguiente.



Este bloque tiene un borde para que podamos solucionar mejor sus fronteras.

El espacio entre el texto y el borde es el margen interior (verde).

El espacio entre el borde y el siguiente bloque es el margen exterior (en rojo).

En CSS, se puede cambiar el tamaño de los márgenes con las siguientes propiedades:

- padding: especifica el tamaño de la separación. Se suele expresar en píxeles (px).
- margin: indica el tamaño del margen exterior. Se utilizan con frecuencia los píxeles.

Las etiquetas de tipo en línea también tienen márgenes. Así que también se pueden tratar estas manipulaciones en etiquetas de este tipo.

Para ver los márgenes, toma dos párrafos a la que se aplica un pequeño borde.

Código CSS

```
p
{
width: 350px;
border: 1px solid black;
text-align: justify;
}
```

Como se puede ver, no hay relleno por defecto (padding). Sin embargo, existe un margen exterior (margin). Este es el margen que hace que los dos párrafos no estén juntos y que parezca "que saltan una línea."

Los márgenes predeterminados no son los mismos para todas las etiquetas tipo bloque. Trata de aplicar este CSS a etiquetas <div> que contienen texto, por ejemplo: verás que en este caso, no hay por defecto margen interior o exterior.

Supongamos que deseas agregar un relleno de 10 píxeles en los párrafos:

Código CSS

```
p
{
width: 350px;
border: 1px solid black;
text-align: justify;
padding: 10px; /* Margen interior de 10px */
}
```

Ahora quiero que mis párrafos estén más separados entre sí. Puedo añadir la propiedad margin para pedir que haya un margen de 50 píxeles entre los párrafos:

```
p
{
```

```

width: 350px;
border: 1px solid black;
text-align: justify;
padding: 10px;
margin: 50px; /* Margen exterior de 50px */
}

```

¿Pero, un margen se añade a la izquierda también? Sí, margin (a semejanza de padding en otras partes) se aplica a los cuatro lados del bloque. Si deseas especificar diferentes márgenes superior, inferior, izquierdo y derecho, tendrás que utilizar más propiedades. El principio es el mismo que la propiedad border.

### **Arriba, derecha, izquierda, abajo ... Y otra vez!**

Lo ideal es retengas los siguientes vocablos en Inglés:

```

top: arriba;
bottom: abajo;
left: izquierda;
right: derecha.

```

Así que puedes memorizar todas las propiedades. La lista de propiedades de margin y padding es:

Lista para margin:

```

margin-top: margen exterior por arriba;
margin-bottom: margen exterior por debajo;
margin-left: margen exterior a izquierda;
margin-right: margen exterior a derecha.

```

Lista para padding:

```

padding-top: margen interior superior;
padding-bottom: margen interior inferior;
padding-left: margen interior izquierdo;
padding-right: margen exterior derecho.

```

Hay otras maneras de especificar los márgenes con las propiedades margin y padding. Por ejemplo: margin: 1px 3px 2px 0, lo que significa "dos px de margen superior, 0px a la derecha (px es opcional en este caso), 3px abajo, 1px a la izquierda". Otra notación abreviada: margin: 2px 1px; significa "2px margen superior e inferior, 1px margen izquierdo px y derecho. "

## Centrado de bloques

Es conveniente hacer un diseño centrado cuando no se sabe la resolución del visitante.

Para centrar se deben cumplir las siguientes reglas:

- dar una longitud al bloque (con la propiedad width);
- indicar que deseas los márgenes exteriores de forma automática, así: margin: auto;.

Prueba esta técnica en nuestros pequeños párrafos (líneas 3 y 4):

Código CSS

```
p
{
width: 350px; /* Se indica un ancho (obligatorio) */
margin: auto; /* Se puede centrar el bloque con auto */
border: 1px solid black;
text-align: justify;
padding: 12px;
margin-bottom: 20px;
}
```

No es posible centrar verticalmente un bloque con esta técnica. Sólo está permitido el centrado horizontal .

## Cuando se sobrepasa ...

Cuando comenzamos a definir dimensiones exactas de los bloques, como acabamos de hacer, pueden llegar a ser demasiado pequeñas para el texto que contienen.

Las propiedades CSS que vamos a ver aquí han sido creadas precisamente para control de desbordamientos de ... y decidir qué hacer si esto sucede.

- overflow: cortar un bloque

Supón que tienes un párrafo largo y queremos que (por una razón que depende de ti) que tenga 250 px de ancho y 110 de alto. Vamos a añadir un borde y rellenar con texto ... hasta el tope.

```
p
{
width: 250px;
```

```

height: 110px;
text-align: justify;
border: 1px solid black;
}

```

Pero el texto supera los límites del párrafo. Pediste dimensiones exactas, las has establecido, pero el texto no cabe en el interior de un pequeño bloque.

Si deseas que el texto no sea superior a los límites del párrafo, tendrás que utilizar la propiedad `overflow`, que puede aceptar los valores:

- visible (por defecto): Si el texto supera el límite de tamaño, sigue siendo visible y supera el bloque.
- hidden: si el texto supera los límites, simplemente se cortan. No podemos ver todo el texto.
- scroll: de nuevo, el texto se ajustará si excede los límites. Excepto que esta vez, el navegador establecerá barras de desplazamiento para que podamos leer el texto completo. Es un poco como un marco dentro de la página.
- auto: este modo es tipo "piloto automático". Básicamente, es el navegador el que decide si debe o no prohibir desplazarse (que lo hará si es necesario). Este es el valor que recomiendo utilizar con más frecuencia.

Con `overflow: hidden`, el texto se corta (no se puede ver a continuación).

Ahora intenta `overflow: auto`; con el siguiente código CSS:

```

p
{
width: 250px;
height: 110px;
text-align: justify;
border: 1px solid black;
overflow: auto;
}

```

Las barras de desplazamiento que aparecen permiten ver el contenido que no aparecía. Hay una antigua etiqueta HTML, `<iframe>`, que da casi el mismo resultado. Sin embargo, el uso de esta etiqueta, no se recomienda en la actualidad. Permite cargar todo el contenido de otra página dentro de la página HTML.

- `word-wrap`: cortar texto demasiado grande

Si tienes que colocar una palabra muy larga en un bloque, lo que no es posible por la anchura,

te será de utilidad `word-wrap`. esta propiedad sirve para forzar la rotura de palabras muy largas (por lo general direcciones).

Por ejemplo el navegador no puede "cortar" la dirección porque no hay espacio y no tiene guión. No puede hacer guiones.

Con el siguiente código de separación silábica será obligada cuando el texto sobrepase.

```
p
{
word-wrap: break-word;
}
```

## Resumen

- Hay dos tipos principales de etiquetas en HTML:

Tipo bloque (`<p>`, `<h1>` ...): estas etiquetas crean una nueva línea de forma predeterminada y ocupa todo el ancho disponible. Siguen de arriba a abajo.

Tipo inline (`<a>` `<strong>` ...): estas etiquetas definen el texto en el medio de una línea. Se siguen de izquierda a derecha.

- Puedes cambiar el tamaño de una etiqueta de tipo bloque con las propiedades `width` y `height`.
- Puedes definir mínimo y máximo para la anchura y la altura: `min-width`, `max-width`, `min-height`, `max-height`.
- Los elementos de las páginas tienen cada uno de márgenes interior (`padding`) y exterior (`margin`).
- Si hay demasiado texto dentro de un bloque de dimensiones fijas, hay riesgo de desbordamiento. En este caso, puede ser útil añadir barras de desplazamiento con la propiedad `overflow` o forzar la ruptura con `word-wrap`.

## Posicionamiento en CSS

Llegó el momento tan esperado: vamos a aprender cómo cambiar la posición de los elementos en una página. La teoría que vemos aquí vamos a necesitarla en el próximo capítulo, en el que vamos paso a paso a lograr el diseño de nuestro primer sitio.

Hay varias técnicas para hacer el diseño del sitio. Cada uno tiene sus ventajas y defectos, serás tú el que elijas elegir lo que mejor se adapte dependiendo del caso.

- Posicionamiento flotante

La técnica aquí presentada es con fines informativos. Es utilizado por la mayoría de los sitios pero tiene defectos. Una técnica mejor, el posicionamiento inline-block se presentará posteriormente animo a utilizarlo tanto como sea posible.

¿Te acuerdas de la propiedad float? Nos acostumbramos a flotar una imagen alrededor del texto.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vitae lorem imperdiet lacus molestie molestie. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu purus. Phasellus metus lorem, blandit et, posuere quis, tincidunt vitae, ante. Vivamus consequat mauris a diam. Vivamus nibh erat, hendrerit nec, aliquet ut, hendrerit quis, nunc. Vestibulum et turpis et elit tempor euismod.

Resulta que esta propiedad ahora es utilizada por la mayoría de los sitios web para para el diseño. En efecto, si deseas colocar el menú a la izquierda y el contenido de la página a la derecha, es a *priori* una buena forma. Digo a *priori*, ya que la base. Esta propiedad no ha sido diseñada para hacer el diseño y vamos a ver que tiene algunas pegas.

Retomemos la estructura HTML vista unos cuantos capítulos más atrás:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
```

```

        <title>Página web de Rafael Barzanallana</title>
</head>
<body>
<header>
    <h1>Universidad de Murcia</h1>
    <h2>Apuntes de las asignaturas que imparto</h2>
</header>
<nav>
    <ul>
        <li><a href="#">DAWeb</a></li>
        <li><a href="#">IAT</a></li>
        <li><a href="#">IAGP</a></li>
    </ul>
</nav>
<section>
    <aside>
        <h1>Algo sobre mi</h1>
        <p>Profesor de la Universidad de Murcia</p>
    </aside>
    <article>
        <h1>Yo soy un gran escéptico</h1>
        <p>Soy contrario a todo tipo de creencias</p>
    </article>
</section>
<footer>
    <p>Bajo licencia Creative Commons<br />
    <a href="#">Contacto</a></p>
</footer>
</body>
</html>

```

Sin CSS aparecería la página como una secuencia lineal de los contenidos.

Vamos a tratar de colocar el menú a la izquierda y el resto del texto a la derecha. Para ello, flotará en el menú a la izquierda y dejará el resto del texto en el lugar correcto. Queremos que el menú ocupe 150 píxeles de ancho. También vamos a añadir un borde negro alrededor del menú y un borde azul alrededor del cuerpo (etiqueta <section>) para distinguirlos:

Código CCS

```

nav
{
}

```

```

float: left;
width: 150px;
border: 1px solid black;
section
{
border: 1px solid blue;
}

```

Hay dos defectos (aparte del hecho de que es todavía feo):

- El cuerpo del texto de la página toca el borde del menú. Le falta un pequeño margen.
- Aún más molesto: el resto del texto sigue debajo del menú

Queremos que el pie de página esté en la parte inferior del menú, por contra, nos gustaría que todos el cuerpo de la página esté construido de un solo bloque situado en la derecha.

Para resolver ambos problemas a la vez, añadir un margen exterior a la izquierda de nuestra `<section>` que debe ser mayor que la anchura del menú. Si nuestro menú es de 150 píxeles, por ejemplo, vamos a dar un margen externo a la izquierda de 170 píxeles a nuestra sección de página, por la línea 10.

Código CSS

```

nav
{
float: left;
width: 150px;
border: 1px solid black;
}
section
{
margin-left: 170px;
border: 1px solid blue;
}

```

Por el contrario, puede que prefieras un elemento a establecer obligatoriamente bajo el menú. En este caso, habrá que usar `clear: both;`, que ya habíamos empleado, que obliga al resto del texto que se coloque bajo el elemento flotante.

## Transforma tus elementos con display

Voy a enseñar aquí a cambiar las leyes de CSS. Hay una instrucción CSS muy potente: display. Puede transformar cualquier elemento de la página a partir de un tipo a otro. Con esta propiedad “mágica”, por ejemplo puedo poner en mis enlaces (originalmente de tipo en línea) como bloques:

### Código CSS

```
a
{
  display: block;
}
```

En este momento, los enlaces se colocan unos debajo de los otros (como bloques normales) y se hace posible cambiar su tamaño.

Algunos de los principales valores que puede tomar la propiedad de presentación en CSS (hay más):

Valor	Ejemplos	Descripción
inline	<a>, <em>,<span>...	Elementos en una línea. Se colocan uno al lado del otro.
block	<p>, <div>,<section>...	Elementos en forma de bloque. Se encuentran uno debajo del otro y se pueden redimensionar.
inline-block	<select>,<input>	Elemento posicionados uno al lado del otro (como elementos en línea), pero que se pueden redimensionar (como bloques).
none	<head>	Elementos no mostrados.

Por lo tanto, puedes decidir ocultar completamente un elemento de página con esta propiedad. Por ejemplo, si deseas ocultar elementos que tienen la clase "secreto", voy a escribir:

Código CSS

```
.secreto
{
display: none;
}
```

¿Y qué es este extraño nuevo tipo, inline-block? ¿Es una mezcla?

Sí, ese tipo de elemento es en realidad una combinación de inline y bloques. Es un poco lo mejor de ambos mundos: elementos que se muestran al lado del otro y pueden cambiar de tamaño.

Pocas etiquetas se muestran de forma predeterminada, se trata de elementos de formulario (como <input>) que descubriremos un poco más tarde. Por contra, con la propiedad de presentación, podemos transformar otras etiquetas inline-block, lo que nos ayudará a alcanzar nuestro diseño.

- Posicionamiento inline-block

Las manipulaciones necesarias para el posicionamiento flotante a veces resultan un poco difíciles en sitios complejos. Cuando haya algo más de un menú sencillo para diseñar, corremos el riesgo de tener que recurrir a `clear: both;` que complica el código.

El posicionamiento flotante sigue siendo, de lejos, el modo más utilizado en la actualidad, existen otras técnicas, poco conocidas los diseñadores de páginas web. Una de ellas, sorprendentemente poderosa, pasa desapercibida a los diseñadores, aunque ha existido desde CSS 2.1, es decir, durante más de diez años. Consiste en la transformación de los elementos inline-block con la propiedad `display`.

Algunos recordatorios sobre temas como inline-block:

- Se colocan uno junto al otro (exactamente como deseas colocar, el menú y el cuerpo de la página).
- Podemos darles las dimensiones exactas (de nuevo, exactamente lo que deseas).

Vamos a transformar en inline-block los elementos que queremos colocar, uno al lado del otro: el menú de navegación y la sección del centro de la página.

## Código CSS

```

nav
{
display: inline-block;
width: 150px;
border: 1px solid black;
}

section
{
display: inline-block;
border: 1px solid blue;
}

```

No es exactamente lo que quería. Y, de hecho, esto es normal: los elementos inline-block se colocan en una misma línea base (llamada línea de base), en la parte inferior.

Afortunadamente, el hecho de haber transformado los elementos inline-block nos permite usar una nueva propiedad, vertical-align. Esta propiedad permite cambiar la alineación vertical de los elementos. Aquí algunos de los posibles valores para esta propiedad:

- baseline: alinea la base del elemento con la del elemento principal (por defecto);
- top: alinea a la parte superior;
- middle: centra verticalmente;
- bottom: alinea a la parte inferior;
- (valor Px o %): alinea a una cierta distancia de la línea de base (baseline).

Nos queda por adaptar nuestros elementos principales (líneas 6 y 13), y resulta:

## Código CSS

```

nav
{
display: inline-block;
width: 150px;
border: 1px solid black;
vertical-align: top;
}

section
{
display: inline-block;
}

```

```
border: 1px solid blue;
vertical-align: top;
}
```

Notarás que el cuerpo (la <section>) no ocupa toda la anchura. En efecto, esto no es un bloque. La sección ocupa sólo el espacio que necesita. Si esto no te conviene para el diseño, cambia el tamaño de la sección con width.

## Posicionamiento absoluto, fijo y relativo

Hay algunas otras técnicas específicas para colocar con precisión los elementos de la página:

- Posicionamiento absoluto: nos permite colocar un objeto en cualquier lugar de la página (arriba, a la izquierda, abajo, en el centro, etc.).
- Posicionamiento fijo: Igual que el posicionamiento absoluto, pero esta vez, el elemento está siempre visible, incluso si se desciende más profundamente en la página. Es un poco el mismo principio de background-attachment: fixed; (ya visto).
- Posicionamiento relativo: sirve para desplazar el elemento en relación a su posición normal.

En cuanto a los flotantes, la posición absoluta, relativa y fija también trabajan en las etiquetas de línea. Sin embargo, verás que se utiliza con más frecuencia en etiquetas de bloque que en las etiquetas de línea.

En primer lugar debemos hacer una elección entre los tres modos de posicionamiento disponibles. Para ello, utilizamos la propiedad CSS position, a la que se le da uno de estos valores:

```
absolute: posicionamiento absoluto;
fixed: posicionamiento fijo;
relative: posicionamiento relativo.
```

Vamos a estudiar cada uno de estas posiciones.

### - Posicionamiento absoluto

El posicionamiento absoluto permite colocar un elemento (de verdad) en cualquier lugar de la página. Para hacer un posicionamiento absoluto, debemos escribir:

```
element
{
position: absolute;
```

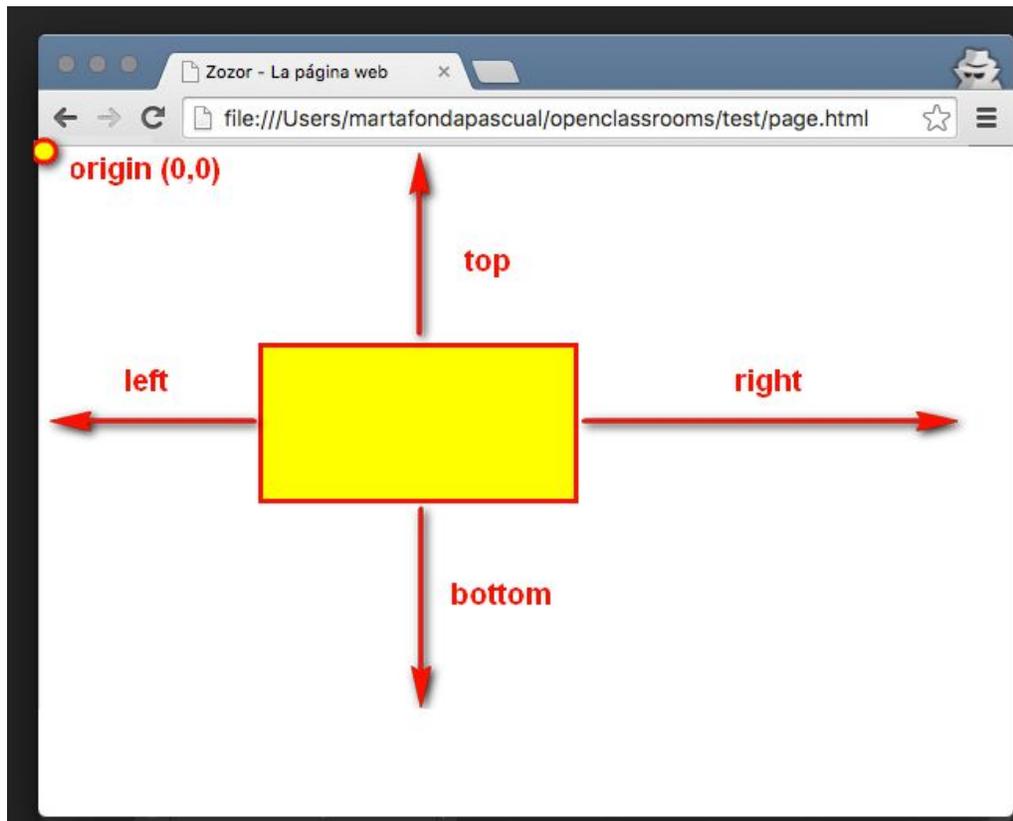
}

Pero esto no es suficiente. Dijimos que queríamos posicionamiento absoluto, pero lo que dice que se quiere es que el bloque sea posicionado en la página.

Para ello, utilizaremos cuatro propiedades CSS:

left: posición relativa a la izquierda de la página;  
 right: posición relativa a la derecha de la página;  
 top: posición relativa a la parte superior de la página  
 bottom: posición relativa a la parte inferior de la página.

Se les puede dar un valor de píxeles como 14px, o un valor de porcentaje, tal como 50%. Si no está claro para algunos, la siguiente figura te ayudará a entender.



Con esto, deberías ser capaz de colocar correctamente el bloque. Así que utiliza la propiedad position y al menos una de las cuatro propiedades anteriores (top, left, right o bottom). Si se escribe por ejemplo:

Código CSS

```
element
{
position: absolute;
right: 0px;
bottom: 0px;
}
```

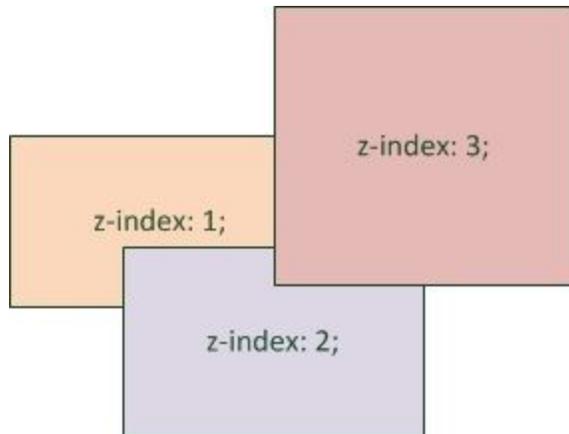
Esto significa que el bloque se debe colocar en la parte inferior derecha (0 píxel desde la derecha de la página, 0 píxel desde la parte inferior de la página).

Los elementos con posición absoluta se colocan sobre los restantes elementos de la página. Además, si pones dos elementos absolutos en el mismo lugar, pueden solaparse. En este caso, utiliza la propiedad z-index para indicar, qué elemento debe aparecer encima del otro.

Código CSS

```
element
{
position: absolute;
right: 0px;
bottom: 0px;
z-index: 1;
}
element2
{
position: absolute;
right: 30px;
bottom: 30px;
z-index: 2;
}
```

El elemento con el valor del índice z mayor será colocado en la parte superior del otro, como se muestra en la siguiente figura.



Una pequeña precisión técnica que es importante: la posición absoluta no necesariamente siempre en relación a la esquina superior izquierda de la ventana. Si configuras un bloque absoluto A que se encuentra en otro bloque B, posicionado absoluto (en relativo o fijo), entonces el bloque A se posicionará respecto a la esquina superior izquierda del bloque B. Haz la prueba y verás.

### Posicionamiento fijo

El principio es exactamente el mismo que en el posicionamiento absoluto, excepto que esta vez el bloque se fija en su posición, incluso si desciende más abajo de la página.

Código CSS

```

element
{
  position: fixed;
  right: 0px;
  bottom: 0px;
}

```

### Posicionamiento relativo

Más difícil, el posicionamiento relativo puede convertirse rápidamente en difícil de usar. Este posicionamiento permite "ajustes": El elemento se desplaza respecto de su posición inicial.

Tomemos, por ejemplo, un texto importante, entre dos etiquetas `<strong>`. Para empezar, ponemos un fondo rojo para poderlo identificar mejor:

Código CSS

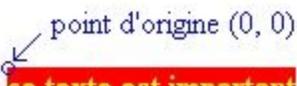
```

strong
{
background-color: red; /* Fondo rojo */
color: yellow; /* Texto de color amarillo */
}

```

Esta vez, el modelo que te mostré anteriormente para las posiciones absolutas y fijas ya no funcionan ¿Por qué?, porque el origen ha cambiado las coordenadas (0, 0) y ya no está en la parte superior izquierda de la ventana, como fue en todos los casos anteriores. No, esta vez el origen se encuentra en la parte superior izquierda de la posición actual del elemento.

Twisted ¿no es así? Este es el principio de la posición relativa. El siguiente diagrama esquemático debería ayudar a entender dónde están los puntos de origen.

Pas de doute,  point d'origine (0, 0) si on veut comprendre corre

Así que si hace una `position: relative`, y al que aplicas las propiedades `top`, `left`, `right` o `bottom`, el texto sobre fondo rojo se moverá con respecto a la posición en la que se encuentra.

Un ejemplo: Quiero que mi texto se mueva 55 píxeles a la derecha y 10 píxeles hacia abajo. Voy a pedir que se compense con 55 píxeles desde la "izquierda" y 10 píxeles desde el "extremo superior" (líneas 6-8):

Código CSS

```

strong
{
background-color: red;
color: yellow;
position: relative;
left: 55px;
top: 10px;
}

```

El texto se desplaza de su posición inicial, como se muestra en la siguiente figura.

Pas de doute,  si on veut com

## Resumen

- El diseño de un sitio web se realiza mediante CSS. Varias técnicas están disponibles para nosotros.
- El posicionamiento flotante (con la propiedad float) es uno de los más utilizados en la actualidad. Permite, por ejemplo, la colocación de un menú en el lado izquierdo o derecho de la página. Sin embargo, esta característica no fue diseñada originalmente para este fin y es preferible, si es posible, evitar esta técnica.
- Posicionamiento inline-block es asignar una serie de elementos inline-block con nuestra propiedad display. Se comportan como inline (colocación de izquierda a derecha), pero pueden ser redimensionados como bloques (con ancho y altura). Esta técnica es preferible al posicionamiento flotante.
- Posicionamiento absoluto, permite colocar un elemento en el sitio que se desea en la página, con una precisión de píxel.
- Posicionamiento fijo es similar al posicionamiento absoluto, pero el tema seguirá siendo visible incluso si cae más abajo en la página.
- El posicionamiento relativo permite desplazar un bloque en relación a su posición normal.
- Un elemento A con posición absoluta dentro de otro elemento B (el mismo en posición absoluta, fija o relativa) se posicionarse en relación con el elemento B, y no desde la esquina superior izquierda de la página.