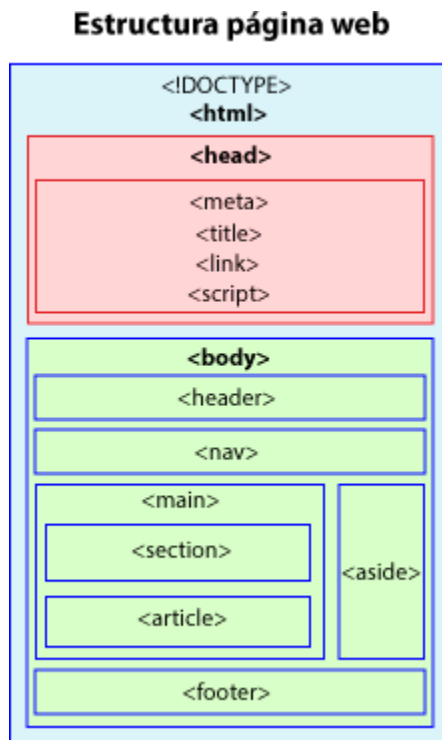


Estructura de una página web

Al hablar de la estructura de una página web `<html>` podemos distinguir dos partes principales, la cabecera `<head>` y el cuerpo `<body>`. La cabecera contiene información importante para la propia página pero que el usuario no ve directamente (salvo el título), y en el cuerpo está el contenido visible de la página.

Vamos a ver las zonas más importantes que hay dentro de ambas partes, para ello utilizaremos el siguiente esquema:



En la cabecera tenemos etiquetas para definir los metadatos `<meta>` como el juego de caracteres (charset) y el viewport, a continuación el título `<title>` y enlaces a archivos de hojas de estilo `<link>` y Javascript `<script>`. Algunas de estas etiquetas son obligatorias y otras opcionales, pero con reglas en cuanto a donde deben colocarse. Englobando la cabecera y el cuerpo tenemos la etiqueta `<html>`. Para definir el tipo de documento se utiliza la declaración `!DOCTYPE`.

Dentro del cuerpo `<body>` tenemos todo el contenido de la página web, por lo tanto podemos incluir en él prácticamente todas las etiquetas de HTML, sin ninguna regla en cuanto al orden de las etiquetas, sin embargo, es conveniente dar una estructura lógica a la página mediante el uso adecuado de las etiquetas semánticas. Este tipo de etiquetas permiten definir cabeceras `<header>`, cuerpo principal `<main>`, secciones `<section>`, artículos `<article>`, contenido adicional `<aside>` y pies `<footer>`. Dentro de cada una de estas partes podemos colocar las etiquetas que incorporan el contenido, párrafos `<p>`, cabeceras (h1,h2,...), listas, tablas, imágenes, vídeos, ... Si necesitamos hacer algún tipo de agrupación de etiquetas que no está contemplado por las etiquetas semánticas disponemos de la etiqueta `<div>` que crea un contenedor sin significado semántico.

Las etiquetas semánticas no conllevan ninguna característica de formato ni de ubicación, pero sí tienen un significado semántico que puede ser interpretado por los buscadores y por otro tipo de programas (lectura por voz, accesibilidad, ...)

Cada etiqueta se representa como un nodo del que cuelgan las etiquetas que contine, de esta forma vemos que una página web tiene una estructura de un árbol. Mediante esta representación podemos ver los diferentes niveles en que se encuentran las etiquetas, cuales son hermanas, cuales hijas, descendientes, etc.

En el siguiente ejemplo hemos representado una página web con una estructura similar a la que estamos viendo. El uso de las etiquetas para el contenido principal, la sección y del artículo no está claramente determinado y cada uno lo puede adaptar a sus necesidades, por ejemplo, en este caso el cuerpo principal contiene a las secciones, y estas están formadas por artículos. A lo largo del curso iremos viendo algunas de las combinaciones posibles con las etiquetas semánticas.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Título </title>
    <link rel="stylesheet" href="mi_hoja_estilo.css">
    <script src="mi_archivo_utilidades.js"></script>
    ...
  </head>
  <body>
    <header> Cabecera </header>
    <nav> Menú </nav>
    <main>
      <section>
        <h2> Cabecera de sección 1 </h2>
        <article>
          <h3> Cabecera del artículo 1 </h3>
          <div class="fondo_art_1">
            <p> Primer párrafo del artículo 1 </p>
            <p> Segundo párrafo del artículo 1 </p>
          </div>
        </article>
        <article>
          <h3> Cabecera del artículo 2 </h3>
          <p> Párrafo del artículo 2 </p>
        </article>
        ...
      </section>
      <section>
        <h2> Cabecera de sección 2</h2>
        ...
      </section>
    </main>
    <aside>
      <p> Publicidad </p>
    </aside>
```

```
<footer>
  <p> ©Copyright 2021 </p>
</footer>
  </body>
</html>
```

También podemos ver una forma de usar el contenedor div, en este caso sirve para agrupar párrafos a los que se le quiere dar un formato específico, pero por una cuestión estética, sin significado semántico, por ejemplo, para darle un color de fondo diferente, <div class="fondo_art_1">.

A continuación, vamos a ver con más detalle cada una de estas etiquetas.

Cabecera de una página web

Como decíamos la cabecera contiene etiquetas necesarias para la página web pero que normalmente no producen contenido visible para el usuario.

Aunque no forman parte estricta de la cabecera vamos a incluir aquí, la declaración !DOCTYPE y la etiqueta html. A continuación, veremos las etiquetas head, meta, title, link y script.

Declaración !DOCTYPE

La declaración !DOCTYPE se utiliza para indicar el tipo de documento HTML, de esta forma se ayuda a que los navegadores analicen correctamente el documento. Debe figurar en primer lugar, antes de la etiqueta HTML. Como vemos nos estamos refiriendo a !DOCTYPE como una declaración ya que no es una etiqueta..

A continuación vamos a ver la forma de esta declaración para varios tipos de versiones de HTML.

Para la versión 5 de HTML, debemos incluir:

```
<!DOCTYPE html>
```

El lenguaje HTML deriva en algunos aspectos de XML, de ahí viene la utilización del DTD (Document Type Definition) en esta declaración.

Etiqueta <html>

La etiqueta html indica que es un documento de tipo página web, es la raíz del documento, debe ir en primer lugar (salvo si se indica !Doctype). Entre la etiqueta de apertura <html> y la de cierre </html> deben ir todas las etiquetas de la página web.

Es recomendable incluir el atributo lang para indicar el idioma del documento. Este atributo toma como valor el identificativo del lenguaje, por ejemplo, para el español, <html lang="es">

Etiqueta <head>

La etiqueta head sirve para agrupar los datos de la cabecera, entre la etiqueta de inicio <head> y la de fin </head> se pueden colocar las etiquetas meta, title, link, script y base.

Este conjunto de etiquetas aporta información para que el navegador pueda interpretar mejor la página web. La mayor parte de esta información no es visible directamente para el usuario, salvo el title.

Etiqueta <meta>

La etiqueta meta sirve para aportar información adicional al navegador a la hora de procesar la página, es lo que se conoce como metadatos. Esta etiqueta no tiene etiqueta de cierre, se cierra con >. La forma de la etiqueta es <meta contenido="valor" >, donde contenido y valor pueden ser cualquier texto, en principio no hay restricciones estrictas al respecto, podemos crear una pareja contenido/valor según nuestras necesidades, por ejemplo, al crear un servicio web.

Los metadatos no se muestran al usuario, pero son analizados por el navegador, los buscadores o servicios web.

Algunos de los datos más frecuentes que se suelen especificar son los siguientes:

- <meta charset="utf-8">, indica el juego de caracteres empleado por la página, se recomienda usar utf-8, anteriormente se usaba ISO-8859-1, para los idiomas occidentales.

- <meta name="viewport" content="width=device-width, initial-scale=1.0">, especifica las características del tamaño de la pantalla para ajustar mejor las páginas con diseño adaptable, sobre todo para dispositivos móviles.

- <meta name="author" content="Colsubsidio">, indica el autor de la página web.

- <meta name="description" content="Cursos de informática gratis" >, indica la descripción del contenido de la página web.

- <meta name="keywords" content="curso, base datos, ofimática, diseño" >, es una lista de palabras claves relacionadas con el contenido de la página web. Esta información, junto con las dos anteriores es utilizada fundamentalmente por los buscadores.

- <meta name="generator" content="Joomla" >, indica el programa que ha generado la página, la utilizan los generadores de contenido como WordPress, Joomla, etc.

- <meta name="robots" content="index, follow" >, da información para los buscadores acerca del uso de programas robot que indexan las páginas web.

<meta name="geo.country" content="co"> Código del país, se utiliza únicamente dos letras (código ISO del país)

<meta name="geo.placename" content="Bogotá"> Ciudad o lugar específico del país

- <meta http-equiv="refresh" content="50">, con http-equiv se proporciona información que va a ser tratada como cabecera por la página web, por ejemplo, para indicar que la página web debe ser cargada de nuevo (refresh) cada 50 segundos, por ejemplo, para una página con datos de la bolsa que debe ser actualizada frecuentemente.

- <meta http-equiv="refresh" content="10;URL='http://nuevodomínio.com/'" >, indica que al cabo de x sg, se redirija a la página indicada. Es útil si hemos cambiado la dirección de la página y queremos redirigir al usuario a otra página de forma automática al cabo de unos segundos, para que pueda leer un mensaje de aviso. Si no queremos que lea nada y la redirección sea inmediata, escribiremos un cero en los segundos.

Etiqueta <title>

Esta etiqueta es obligatoria y se muestra en la pestaña del navegador, no forma parte del cuerpo de la página. Sólo se puede escribir texto y es importante para que los buscadores conozcan el tema de la página. Los resultados de las búsquedas muestran como primera línea lo que pongamos en esta etiqueta title, por lo tanto, es muy importante elegir bien este texto ya que puede hacer que los usuarios visiten o no nuestra página.

Por ejemplo: <title> Cursos de informática gratis y de calidad. Curso on-line de Excel, Access, Word y PowerPoint</title>

Etiqueta <link>

La etiqueta link sirve para enlazar la página web con otros archivos. Esta etiqueta no tiene etiqueta de cierre, se cierra con >. Puede haber varias etiquetas link en la cabecera de una página web, pero no puede aparecer en el cuerpo.

El uso más frecuente es para enlazar con las hojas de estilos, por ejemplo:

```
<link rel="stylesheet" href="css/mi_hoja_estilo.css">
```

Los parámetros son los siguientes:

- href, igual que para los hiperenlaces, nos dice la ruta donde se encuentra el archivo que estamos enlazando.
- rel, indica el tipo de recurso o relación, puede tomar varios valores (alternate, author, help, icon, license, next, stylesheet, ...).

Se utiliza stylesheet para enlazar con una hoja de estilo CSS. Y icon para enlazar con el gráfico que se utilizará como icono en las pestañas de los navegadores y en otros lugares. Con alternate podemos indicar que se trata de una hoja de estilo alternativa.

- type, indica el tipo de media del archivo, por ejemplo, type="text/css", para hojas de estilo, aunque no es obligatorio.

- media, permite especificar una condición Media Queries, por ejemplo, si queremos que la hoja de estilo se enlace sólo para las pantallas que como máximo tengan 800 px de ancho:

```
<link rel="stylesheet" media="(max-width: 800px)" href="pantalla_800.css" >
```

Etiqueta <script>

La etiqueta script permite incluir código de Javascript en la página web. Se puede incluir un atributo src con el archivo que contiene el código, por ejemplo:

```
<script src="mi_archivo_utilidades.js"></script>
```

También se puede escribir directamente el código entre la etiqueta de inicio <script> y la de fin </script>, por ejemplo:

```
<script>document.getElementById("prueba").innerHTML = "Hola amigo!"; </script>
```

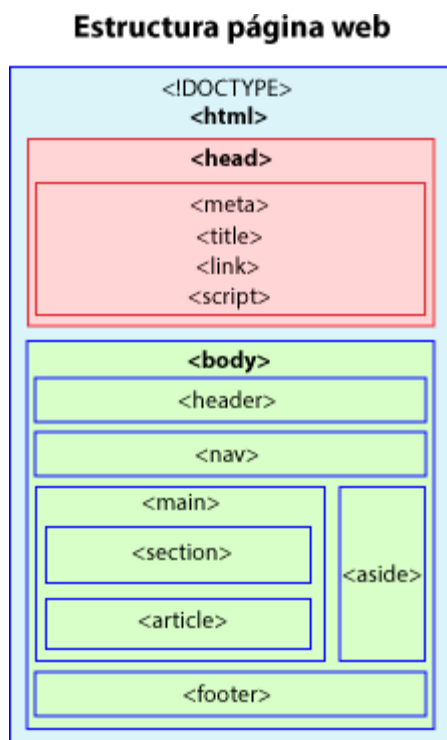
La etiqueta script puede ir en la cabecera y en el cuerpo de la página.

Etiqueta <body>

La etiqueta body engloba todo el cuerpo de la página web, el cuerpo contiene la información que verá el usuario, al contrario que la cabecera cuya información es para uso interno de la página web.

Todo el contenido de la página está entre el inicio de la etiqueta <body> y el fin de la etiqueta </body>

Vamos a explicar las partes del cuerpo de una página web sobre este esquema que ya vimos al principio de esta unidad,



Este es sólo un ejemplo de la estructura del cuerpo <body> ya que tenemos total libertad para utilizar las etiquetas que deseemos, y en el orden que prefiramos. De hecho, hasta la versión 5 de HTML no existían las etiquetas semánticas (section, article, ...) y las páginas se estructuraban utilizando los contenedores div, sin significado semántico. Todavía, actualmente (2019) muchas páginas no utilizan las etiquetas semánticas.

Por ejemplo, el siguiente fragmento de código del cuerpo de una página web que utiliza etiquetas semánticas, puede escribirse de forma equivalente sin utilizarlas como vemos en el párrafo más abajo:

```
<body>
  <header> Cabecera </header>
  <nav> Menú </nav>
  <main>
    <section>
      <h2> Cabecera de sección 1 </h2>
    <article>
      <h3> Cabecera del artículo 1 </h3>
      <div class="fondo_art_1">
```

```

                <p> Primer párrafo del artículo 1 </p>
                <p> Segundo párrafo del artículo 1 </p>
            </div>
        </article>
    </section>
</main>
<aside>
    <p> Publicidad </p>
</aside>
<footer>
    <p> ©Copyright 2021</p>
</footer>
</body>

```

Utilizando contenedores div con clases podemos obtener el mismo resultado en cuanto a estructura y formato, aunque sin el componente semántico:

```

<body>
    <div class="cabecera"> Cabecera </div>
    <div class="navegar"> Menú </div>
    <div class="principal">
        <div class="seccion">
            <h2> Cabecera de sección 1 </h2>
            <div class="articulo">
                <h3> Cabecera del artículo 1 </h3>
                <div class="fondo_art_1">
                    <p> Primer párrafo del artículo 1 </p>
                    <p> Segundo párrafo del artículo 1 </p>
                </div>
            </div>
        </div>
    </div>
    <div class="lateral">
        <p> Publicidad </p>
    </div>
    <div class="pie">
        <p> ©Copyright 2018</p>
    </div>
</body>

```

Sin embargo, es recomendable utilizar las etiquetas semánticas, por lo tanto, vamos a verlas con más detalle a continuación.

Etiqueta <header>

Se utiliza como una introducción del elemento que la contiene. Si colocamos un header en el body, a primer nivel, será una introducción o presentación de toda la página. Si colocamos un header dentro de un section, contendrá una introducción a esa sección. También se pueden colocar dentro de un article.

El header suele llevar el título, la descripción corta y el logo de la página. El header por si mismo no separa contenidos, no se debe utilizar aislado, sino dentro de otra etiqueta semántica, o a primer nivel en el body.

El header no es equivalente a la parte superior de la página, aunque frecuentemente está en esa posición. Si un sitio web tiene una cabecera que se repite a lo largo de todas las páginas del sitio, esa zona será candidata a ocupar una etiqueta header. La etiqueta hgroup que se solía utilizar junto con el header ha quedado obsoleta.

Es bastante frecuente utilizar las etiquetas h1, h2, ... para escribir los títulos dentro del header.

Etiqueta <main>

La etiqueta main sirve para agrupar el contenido principal del cuerpo de una página web. Es decir, la parte de la página donde se trata el tema primordial de la página. Lo normal es que sólo hay una etiqueta main en cada página, y esta se subdivide en secciones, artículos, ...

Etiqueta <section>

Una section agrupa contenido con un tema común, por ejemplo, las diferentes partes de un periódico: política, sociedad, deportes, ...

Un sitio de cocina podría estar organizado en secciones. Una section podría ser recetas de carne, otras recetas de pescado y otra de verduras. Dentro de cada section tendríamos un article por cada receta.

Cada section puede estar encabezada por una etiqueta h1, h2,... o puede tener su propia etiqueta header. Dentro de una section suelen haber etiquetas article, y también etiquetas div y p.

Una section no debe ser usada con el único fin de dar formato, para eso se utiliza la etiqueta div.

Etiqueta <article>

Representa una unidad de contenido, es decir, contenido que responde a un tema concreto. El ejemplo más claro es un artículo dentro de una revista, por ejemplo, en una página de cocina, un article sería cómo elaborar un determinado plato. En un foro, un article sería un post, o entrada en el foro. Cada article puede estar encabezado por una etiqueta h1, h2,... o puede tener su propia etiqueta header.

Se pueden anidar un article dentro de otro article. Dentro de un article el contenido se suele estructurar con las etiquetas div y p. También está permitido poner dentro de un article una section, siempre que el artículo sea largo y tenga diferentes secciones, pero lo más natural es que el article vaya dentro de una section. Para crear estructuras de páginas complejas, con tres niveles, podemos utilizar etiquetas div a tercer nivel, o anidar article y section.

Etiqueta <nav>

Esta etiqueta está pensada para contener la barra de navegación o los enlaces a las páginas del sitio web.

Normalmente va en la parte superior de la página o en un lateral. Suele estar formada por una lista de enlaces con las etiquetas ul y li. Aunque hay multitud de diseños diferentes de barras de navegación y menús.

Aunque no es lo habitual, la etiqueta nav también puede ir dentro de las etiquetas section y article.

Etiqueta <aside>

Esta etiqueta está diseñada para contenido adicional, es decir, menos importante que el cuerpo de la página, y que suele ser de un tema relacionado indirectamente (o no relacionado) con el tema principal. Se suele colocar

en los laterales de la página. Pero no todo lo que va en los laterales es aside. Si la página tiene publicidad en el lateral derecho, puede colocarse en una etiqueta aside, también una pequeña reseña sobre el autor podría ir en esta etiqueta.

Etiqueta <footer>

Normalmente va al final de la página y contiene información del tipo: autor, copyright, contacto, mapa del sitio, etc. Se puede poner footer dentro de section y article, aunque no es lo más normal. No hay nada prohibido, pero debe tener cierta lógica o sentido semántico. Por ejemplo para un article muy extenso puede tener sentido utilizar header y footer dentro, si tienen unas zonas que lo requieran semánticamente.

Estas son las etiquetas obligatorias en toda página web HTML 5, si no existen la página dará un error de validación (aunque el usuario no lo perciba) a continuación las veremos con detalle. También hay otras etiquetas que son muy importantes y es conveniente utilizar casi siempre, como las etiquetas <link> y <script>.

Etiqueta <div>

La etiqueta <div> es un contenedor genérico y neutro (sin significado semántico). Sirve para contener otras etiquetas y poder hacer un tratamiento conjunto, por ejemplo, darles un estilo, asignarles un identificador o estructurar el contenido. Normalmente se utiliza para agrupar etiquetas de bloque como párrafos, tablas, listas, nav, section, article, etc.

Si aplicamos una etiqueta <div> sin más, no producirá ningún efecto. Los navegadores no le asignan ningún formato por defecto. Tampoco los buscadores le darán ningún significado.

● Uso de la etiqueta <div>.

Hasta la aparición de las etiquetas semánticas (nav, section, article, etc.), se ha venido utilizando la etiqueta div para estructurar las páginas web. Ahora la recomendación oficial es no utilizar la etiqueta div cuando sea posible utilizar una etiqueta semántica, dejando el uso de la etiqueta div como último recurso para dar estructura a la página web. La razón para este cambio es que la página web tendrá un mejor significado semántico.

Por lo tanto, el uso de la etiqueta div se centrará en cuestiones de formato. Otro uso importante de la etiqueta div es para identificar partes de la página a efectos de programación. Por ejemplo, para ocultar/mostrar contenido, menús desplegables, animaciones, etc.

Las etiquetas div no pueden utilizarse dentro de un párrafo (p), ni encabezados (h1, h2, ...), ni texto preformateado (pre). Tampoco dentro de elementos en-línea como (strong, em, span).

No tiene mucho sentido utilizar la etiqueta div para agrupar una única etiqueta, en ese caso es mejor dar el formato directamente a esa etiqueta.

Por ejemplo, esto no sería recomendable:

```
<div><p>En internet hay muchos cursos gratuitos.</p></div>
```

**Etiqueta **

La etiqueta se utiliza para delimitar un texto corto con el fin de darle formato, o realizar otro tratamiento sobre él. Esta etiqueta no tiene significado semántico, por lo tanto, no indica que el texto tenga más o menos importancia. Los navegadores no dan ningún formato por defecto. Como siempre, para dar formato se utilizan

las hojas de estilo. La etiqueta `` es de tipo `display inline` por lo que no introduce ningún salto de línea, mientras que la etiqueta `<div>` si lo hace.

Por ejemplo, para poner el fondo de color utilizaremos un estilo `.fondo_amarillo { background-color: yellow }`

`<p> El futbolista es el deportista mejor`

El futbolista es el deportista mejor pagado.

La etiqueta `span` está pensada para delimitar un texto corto dentro de un párrafo u otro elemento, si el texto es largo y va a ocupar varias líneas, lo más lógico es utilizar una etiqueta `p`, ya que ese texto será un párrafo en sí mismo.

Posicionamiento.

Diseñar una página web es una tarea primordial a la hora de crear páginas web. El posicionamiento es lo que nos permite colocar los elementos de una página web para crear un diseño. Los modos de posicionamiento son una característica técnica, el diseño es una actividad creativa o incluso artística. Pero sin conocer bien el posicionamiento no tendremos herramientas para diseñar. Por esto vamos a estudiar primero el posicionamiento en esta unidad, y en las siguientes veremos diseño y maquetación.

El posicionamiento se establece, principalmente, con tres propiedades: `position`, `float` y `display`.

Posicionamiento relativo

El posicionamiento relativo coloca el elemento en su posición normal, y a partir de ahí lo podemos desplazar la distancia que le indiquemos. Al desplazarlo deja un "hueco" donde estaría su posición normal. El desplazamiento siempre es respecto al elemento que lo contiene.



Para aplicarlo, lo primero es declararlo en el estilo, utilizando `position: relative;`.

Para indicar el desplazamiento horizontal, utilizamos `left` para desplazarlo a la izquierda y `right` para hacerlo a la derecha. Utilizar valores negativos, produce el efecto opuesto. Por ejemplo, para desplazar un elemento 40px a la derecha, podríamos poner `right: 40px;` o `left: -40px;`.

Del mismo modo, podemos desplazarlo verticalmente utilizando `top` para desplazarlo hacia arriba o `bottom` para hacerlo hacia abajo.

Si no establecemos valores de desplazamiento el elemento no cambia su posición, por eso es muy común asignar el posicionamiento relativo a elementos para poder utilizar una propiedad que requiere posicionamiento, como puede ser `z-index`.

Posicionamiento absoluto

Al utilizar el posicionamiento absoluto indicamos de forma precisa la posición del elemento en la página.

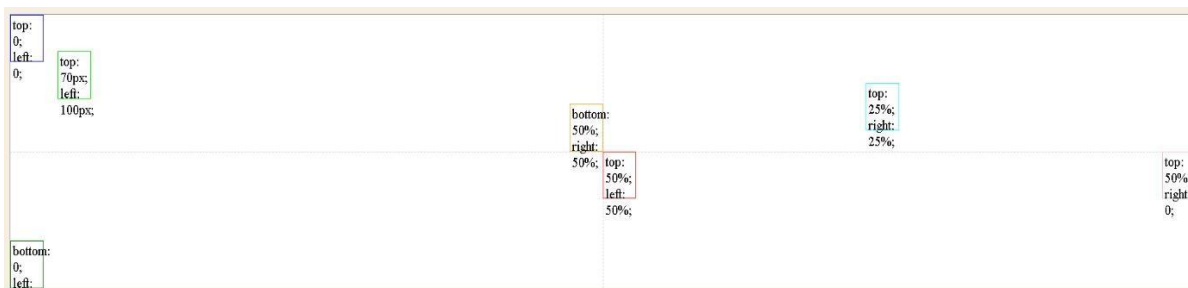
Al contrario que con el posicionamiento relativo, el elemento al que le hemos asignado un posicionamiento absoluto no deja un hueco en la página. Podemos decir que el resto de elementos lo ignoran, y se colocan como si no existiese.

Para aplicarlo, lo primero es declararlo en el estilo, utilizando `position: absolute;`.

La posición y tamaño del elemento se indica siempre en relación a otro elemento que lo contiene. El elemento contenedor será el más cercano que haya con posicionamiento no estático. Si no hay ninguno, se utilizará el body.

Para definir la posición utilizamos `top` para referirnos a la distancia entre el borde superior del elemento posicionado y el borde superior del elemento contenedor. Por lo tanto, `left` será la distancia entre los lados izquierdos, `right` entre los lados derechos y `bottom` entre los lados inferiores.

Para posicionar un elemento del que conocemos sus dimensiones, basta con indicar un punto en vertical (`top` o `bottom`) y un punto en horizontal (`right` o `left`). En el siguiente elemento, todas las cajas tienen `position: absolute;` `width: 60px;` `height: 80px` y un borde. Para cada una hemos cambiado la posición. Además, a la caja que contiene a todas, le hemos dado `position: relative;` para que las cajas de dentro tomen su posición a partir de ella.



Utilizando este posicionamiento podemos colocar un elemento en una posición concreta, pero tiene algunos inconvenientes. Un inconveniente es que los elementos pueden solaparse al cambiar el ancho de la ventana. Por otro lado, el posicionamiento absoluto no es muy útil si queremos diseñar páginas adaptables a todo tipo de anchos. No obstante, el posicionamiento absoluto puede ser interesante en algunas situaciones.

Posicionamiento flotante

Este posicionamiento es un poco distinto al resto. Para empezar, no se define con la propiedad `position`, si no con la propiedad `float`. Puede tomar estos valores:

`left` : flotante a la izquierda (`float: left;`). `right` :

flotante a la derecha (`float: right;`). `none` : sin

flotante. (`float: none;`)

Lo que hace este posicionamiento es desplazar el elemento todo lo que pueda hacia la dirección indicada, hasta encontrar el límite del elemento contenedor, u otro elemento también flotante. Además, hace que todos los elementos fluyan alrededor de él.

Es muy común utilizarlo en imágenes, para que el texto fluya alrededor de ellas. En el siguiente ejemplo, todas las cajas contienen un párrafo con una imagen al principio (<p> Texto</p>). Sólo cambia el float de la imagen.



Observa que el texto puede quedar demasiado pegado a la imagen. Para solucionarlo, no hay más que aplicar un pequeño margin a la imagen, hacia el lado que está el texto.

La propiedad display

La propiedad display es muy importante a la hora de posicionar los elementos en una página web ya que determina el espacio que va a ocupar un elemento. Existen dos tipos básicos de display, el bloque (block) y en línea (inline). Cada elemento o etiqueta de HTML tiene asignado por defecto un tipo de display, pero podemos cambiarlo. Las etiquetas con display en línea más usadas son la etiqueta span, la etiqueta a (hiper enlace), etiqueta img y las etiquetas de formato de palabras (strong, b, code, s, i, ...). Las etiquetas de bloque más usadas son p, div, table, ol, ul, h1, h2.