Unidad - CSS

Introducción

Antes del uso de **CSS** , los diseñadores de páginas web debían definir el aspecto de cada elemento dentro de las etiquetas HTML de la página. El principal problema de esta forma de definir el aspecto de los elementos es que hay que definir el formato de cada uno de los que forman la página, lo cual, hace que sea muy difícil de actualizar, mantener y arreglar fallos.

CSS permite separar los contenidos de la página y su aspecto. Para ello se define en una zona reservada el formato de cada uno de los elementos de la web. Cualquier cambio en el estilo marcado para un elemento en el CSS afectará a todas las páginas vinculadas a ella en las que aparezca ese elemento. Las hojas de estilo están compuestas por una o más reglas de estilo aplicadas a un documento HTML o XML.

Introducción

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, cabecera, texto destacado, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el formato de cada elemento.

CSS obliga a crear documentos semánticos HTML/XHTML, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

CSS ayudó a solucionar varios de los problemas que hasta ese momento tenían programadores de HTML. Vamos a verlos:

Introducción

- Cada etiqueta de HTML podía incorporar atributos diferentes para introducir el diseño. Esto hacía que el código HTML, fuese farragoso y difícil de comprender.

- Si se quería añadir el mismo atributo a todas las etiquetas del mismo tipo que hubiese en la página, había que repetir el atributo tantas veces como apareciese dicha etiqueta. Por lo que había que escribir varias veces el mismo código.

- Debido a que muchos de los atributos no eran estándar en un principio, no se representaba igual el diseño de la página en un navegador que en otro.

Evolución del CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar estilos a los documentos electrónicos.

La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos que tuvieran igual apariencia en distintos navegadores.

Desde que aparecieron los primeros navegadores gráficos sobre 1993, distintas empresas de desarrollo de SW, entre ellas Netscape Communications (con Netscape Comunicator) y Microsoft (con Internet Explorer), añadían continuamente nuevas etiquetas y funcionalidades a sus navegadores con intención de atraer usuarios, pero sin realizar mejoras profundas en ellos.

Esto hizo que fuese complicado desarrollar páginas HTML que se pudiesen visualizar correctamente en todos los navegadores.

Evolución del CSS

El organismo W3C propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML.

En 1995, el W3C añadió a su grupo de trabajo de HTML el desarrollo y estandarización de CSS.

- CSS 1, se publicó en 1996, es la primera recomendación oficial.
- CSS 2, publicada en 1998, es la segunda recomendación oficial.
- CSS 3, continúa en desarrollo desde 1998, es la tercera recomendación oficial

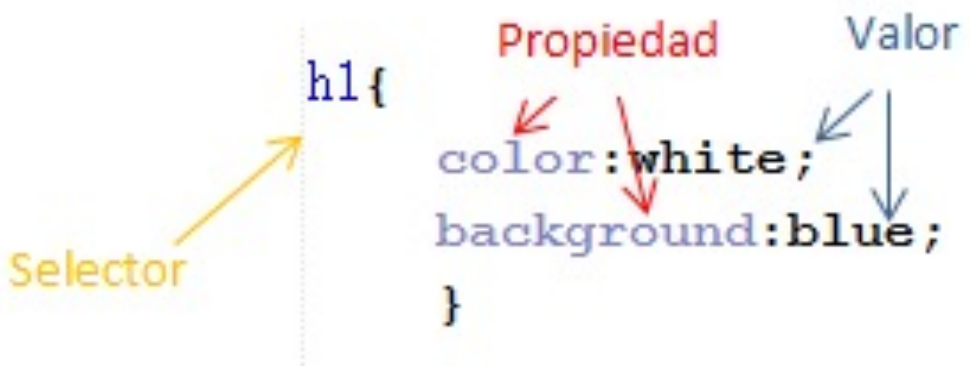
Actualmente la versión más utilizada es la CSS 3

Reglas CSS

Cualquier regla de estilo tiene dos partes importantes:
El selector que indica a qué elementos se aplicarán dichos elementos
La declaración que a su vez se compone de dos partes:

- **La propiedad:** Qué es lo que vamos a cambiar del selector (color, tamaño, …)
- **Valor:** El valor que le vamos a asignar (rojo, 20px, Arial…)

Una misma regla puede tener varias declaraciones **separadas por punto y coma**.



Un archivo CSS puede contener infinitas reglas CSS, cada regla puede contener varios selectores y cada declaración puede estar formada por diferentes declaraciones.

Comentarios CSS

Conviene añadir **comentarios** a las hojas de estilo para recordar qué hacen las reglas de estilo, sobre todo aquellas que sean más complejas.

Para añadir comentarios, se comienza con los caracteres **/*** y todo lo escrito a continuación se considerará comentario hasta que aparezcan los caracteres ***/**

```
/* Regla para párrafos en que el texto es blanco, el fondo azul,
   el tipo de letra Arial y el tamaño 30 pízeles*/
p{color:white; background:blue; font-family:Arial; font-size:30px;}
```

Inclusión de CSS en HTML

Existen tres opciones para incluir CSS en un documento HTML o XHTML:

- CSS inline
- CSS en el head
- CSS en un archivo externo

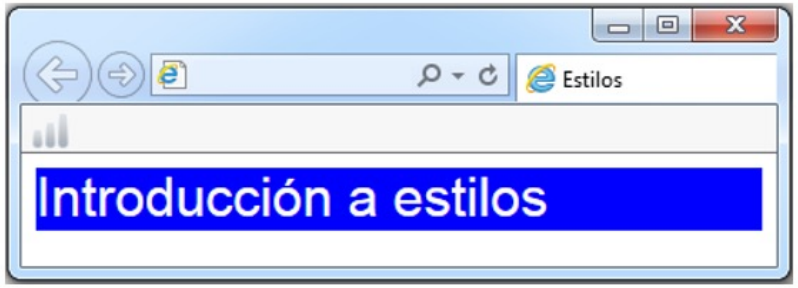
Inclusión de CSS en HTML

CSS inline:

Este método es el peor y el menos utilizado, ya que para modificar un formato hay que cambiar todos los elementos que estén asociados a él.

Solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

```
<p style="color:white; background-color:blue; font-family:Arial; font-size:30px;">
    Introducción a estilos
</p>
```



Inclusión de CSS en HTML

CSS en el head:

Este método se emplea cuando se definen pocos estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos globales de todas las páginas del sitio web.

Tiene el inconveniente de que, para modificar los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a cambiar.

```
<html>
<head>
     <title>Estilos</title>
     <meta http-equiv="content-type" content="text/html"; charset="UTF-8">
     <style type="text/css">
        p{color:white; background:blue; font-family:Arial; font-size:30px;}
      </style>
</head>
<body>
    <p>
     Primer párrafo
    </p>
    <p>
     Segundo párrafo
    </p>
</body>
```

Inclusión de CSS en HTML

CSS en un archivo externo

El archivo externo, es la forma de incluir CSS en las páginas HTML más utilizada. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Además, el mantenimiento del sitio web se simplifica al máximo, ya que el cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas HTML asociadas.

Se puede realizar de dos formas:

1. Enlazándolas mediante el elemento <link> de la cabecera del fichero HTML con sus correspondientes atributos
2. Importándolo con la etiqueta <style> y el símbolo @ delante del nombre del fichero .css

Inclusión de CSS en HTML

1. Enlazándolas mediante el elemento <link> de la cabecera del fichero HTML con sus correspondientes atributos

```
<head>
     <title>Estilos</title>
     <meta http-equiv="content-type" content="text/html"; charset="UTF-8">
     <link rel="stylesheet" type="text/css" href="estilos.css">
</head>
```

2. Importándolo con la etiqueta <style> y el símbolo @ delante del nombre del fichero .css

```
<head>
     <title>Estilos</title>
     <meta http-equiv="content-type" content="text/html"; charset="UTF-8">
     <style>@import "estilos.css"</style>
</head>
```

Selectores

Existen diferentes tipos de selectores

- Selector universal: Representa a todos los elementos de la página y se simboliza mediante el carácter *
- Selector de tipo: Es el selector que coincide con algún elemento html
- Selector descendiente: Un selector es descendiente de otro si el primero se encuentra “encerrado” entre las etiquetas del segundo.
- Selector hijo: Es un selector descendiente, pero en este caso sólo el que está justo un nivel por debajo.

Selectores de clase

Supón que en una página tienes varios párrafos marcados con sus respectivas etiquetas <p>. Deseas que los párrafos tengan estilos diferentes, los pares un estilo y los impares otro estilo ¿Cómo podrías resolverlo?

Las **clases** nos permiten solucionarlo asignando un identificador de clase a la etiqueta concreta a la que deseamos aplicar dicho estilo de la forma:

. identificador { declaraciones }

Para asociar un elemento HTML a una clase habrá que usar el atributo CLASS al usar dicho elemento en el documento HTML del siguiente modo:

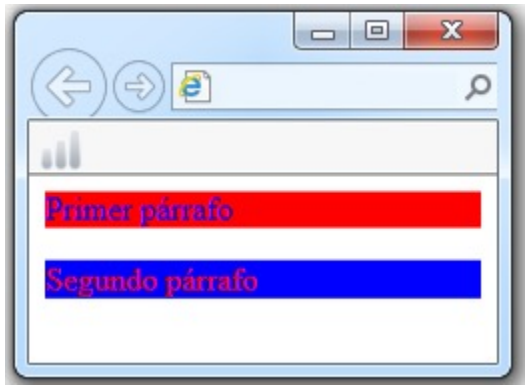
<h3 class="clase_azul">El encabezado de tercer nivel es ahora azul</h3>

Para restringir la clase a un determinado elemento basta poner el elemento delante del punto al definir la regla. Por ejemplo, para restringir el uso de la clase a párrafos tendremos:

p.clase_azul{color:blue}

Selectores de clase

```
<html>
<head>
     <title>Estilos</title>
     <meta http-equiv="content-type" content="text/html"; charset="UTF-8">
     <style type="text/css">

       p.clase1{background-color:red; color:blue;}
       p.clase2{background-color:blue;color:red;}

     </style>
</head>
<body>
    <p class="clase1">
     Primer párrafo
    </p>
    <p class="clase2">
     Segundo párrafo
    </p>
</body>

</html>
```



Actividad

Realiza la siguiente tabla en html y aplícale los estilos para que su aspecto sea el siguiente:

Selectores de pseudoclase

Son selectores que permiten añadir estilos a un mismo elemento dependiendo de su estado.
Se aplica a la etiqueta HTML, separada por dos puntos de su estado.
Un ejemplo muy significativo son los distintos **estados de un enlace**. Un enlace puede estar en uno de estos estados:

- Normal, no visitado
- Situados con el ratón por encima (hover)
- Haciendo click, sobre el enlace
- Visitado

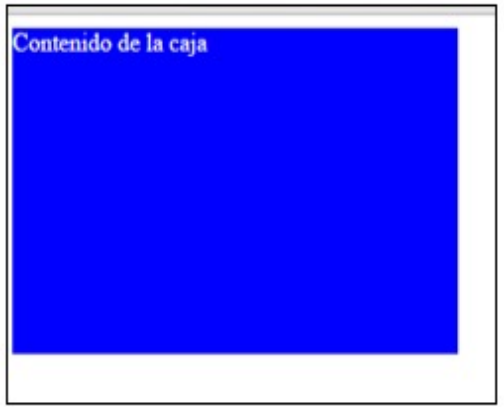
```
<style type="text/css">

   a:link {color:#FF8000}
   a:visited {color:gray}
   a:hover {color:#FF8000;font-size:20px;}
   a:active {color:red;}

</style>
```

Selectores de identificador

El selector permite seleccionar un elemento por su atributo identificador (id)
El selector comienza con el carácter #

El id es un atributo que identifica de forma única un elemento, por lo tanto, el selector id se utiliza normalmente para asignar propiedades a un único elemento.

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <style>
        #caja1 {background-color:blue; color:white;
                width:300px; height:200px}
    </style>
</head>
<body >
   <div id="caja1">
    Contenido de la caja
   </div>
</body>
</html>
```



Propiedades principales

Las propiedades de color y fondo son los que enumeramos a continuación:

- color: Indica el color del texto. Lo admiten casi todas las etiquetas de HTML. El valor de este atributo es un color, con su nombre o su valor RGB.

- background-color: Indica el color de fondo del elemento. El valor de este atributo es un color, con su nombre o su valor RGB.

- backgroun-image: Permite colocar una imagen de fondo del elemento. El valor que toma es el nombre de la imagen con su ruta relativa o absoluta.

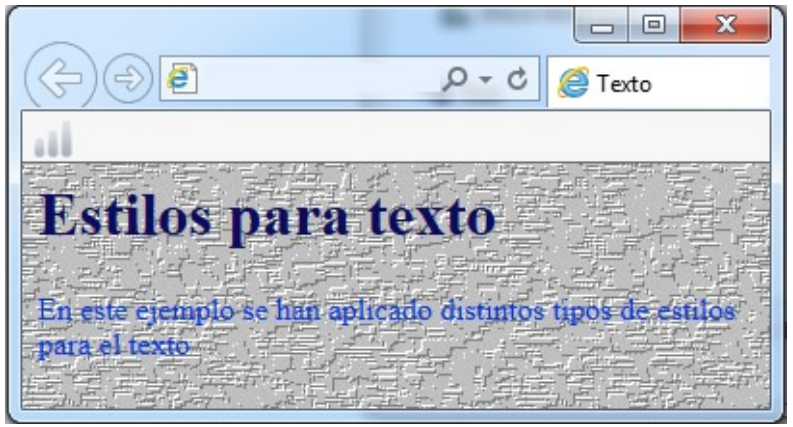
- background-repeat: Indica si ha de repetirse la imagen de fondo y, en ese caso, si debe ser horizontal o verticalmente

- background-attachment: Especifica si la imagen ha de permanecer fija o realizar un scroll.

- backround-position: Es una medida, porcentaje o el posicionamiento vertical u horizontal con los valores establecidos que sirve para posicionar una imagen

Propiedades principales

```
<html>
<head>
     <title>Tamaños</title>
     <meta http-equiv="content-type" content="text/html"; charset="UTF-8">
     <style type="text/css">
        body  {color:#8A0829; background-image:url("../images/gris02.jpg")}
        h1 {color:#FE2E64;}
     </style>
</head>
<body>
         <h1>Estilos colores y fondos</h1>
         <p> En este ejemplo se ha establecido una imagen de fondo
         y se han aplicado colores al texto</p>
</body>
</html>
```



Propiedades principales

En este apartado vamos a ver los distintos atributos que podemos utilizar referentes a las fuentes de nuestro documento y que son:

- font-size: Indica el tamaño de la fuente. Puede ser un tamaño absoluto, relativo o en porcentaje. Toma valores de unidades de CSS.

- font-family: Establece la familia a la que pertenece la fuente. Si el nombre de una fuente tiene espacios se utilizan comillas para que se entienda bien. El valor es el nombre de la familia fuente.

- font-weight: Define el grosor de los caracteres. Los valores que puede tomar son: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800 o 900.

- font-style: Determina si la fuente es normal o cursiva.

- font-variant: Determina si la fuente es normal o mayúsculas pequeñas

- line-height: El alto de una línea y por tanto, el espaciado entre líneas.

Propiedades principales

En el apartado anterior vimos los atributos relacionados con las fuentes y en este vamos a ver los relacionados con el texto en sí y son los siguientes:

- text-decoration: Establece si el texto está subrayado, sobre rayado o tachado.
- text-align: Indica la alineación del texto
- text-indent: Determina la tabulación del texto. Los valores que toma son una longitud, en unidades CSS, o un porcentaje de la establecida
- text-transform: Nos permite transformar el texto, haciendo que tenga la primera letra en mayúsculas de todas las palabras, todo en mayúsculas o minúsculas.
- word-spacing: Determina el espaciado entre las palabras
- letter-spacing: Determina el espaciado entre las palabras
- vertical-align: Establece la alineación vertical del texto
- line-height: Altura de la línea. Puede establecerse mediante un tamaño o un porcentaje.

Propiedades principales

Ahora vamos a ver otras propiedades muy importantes y que utilizaremos muy a menudo con los divs o tablas:

- margin: Permite establecer los márgenes de una vez. Se pueden subdividir en margin-left, margin-right, margin-top y margin-bottom

- padding: Establece el espacio entre los bordes y el contenido de una sola vez. Hay que respetar el orden superior, derecho, inferior e izquierdo. Puede usarse una longitud, en unidades CSS, o un porcentaje. Se puede dividir en padding-left, padding-right, padding-top y padding-bottom

- border-color: Establece el color de los bordes del elemento de una sola vez. Su valor es un color RGB o el nombre del color.

- border-style: Establece el estilo del borde

Propiedades principales

Ahora vamos a ver otras propiedades muy importantes y que utilizaremos muy a menudo con los divs o tablas:

- border-width: Establece el grosor de los bordes del elemento al que lo aplicamos. Sus valores posibles son: thin, medium, thick o un tamaño
- border-radius: Proporciona un diseño redondeado a los bordes. Se utilizan unidades de medidas o porcentaje
- width: Establece el ancho del contenido del elemento. El valor es un porcentaje o un tamaño
- height: Establece la altura del contenido del elemento. El valor es un porcentaje o un tamaño
- float: Sirve para alinear un elemento a la izquierda o la derecha haciendo que el texto se agrupe alrededor de dicho elemento.

Posicionamiento

Todos los elementos al insertarlos en una página, se adaptan a una posición, que se denomina **posición natural**.

Mover al elemento de dicha posición, se denomina posicionamiento, y existen 5 formas de realizarlo con CSS.

Es la posición en el que el navegador coloca los distintos elementos navegadores por defecto. Tanto si son elementos en línea o en bloque

Aunque es el posicionamiento que viene por defecto, se puede especificar indicando en los estilos la propiedad: **{position:static;}**

Los elementos de línea no ocuparán todo el ancho de la página, sólo lo que ocupe el contenido. (Ejemplos: <a>, <img>,…)

Los elementos de bloque se sitúan al principio de la línea y ocupan todo el ancho de la página. (Ejemplo: <p>)

Posicionamiento

Si deseamos mover un elemento con respecto a su ubicación natural, es decir, respecto al elemento en el que se encuentra incluido, a este tipo de posicionamiento se le denomina **posicionamiento relativo**.

Ejemplo:
Probemos a añadir el siguiente estilo al ‘a’ de un ‘p’.

```
a {position:relative; top:10px;}
```

Posicionamiento

Con el **posicionamiento absoluto**, no se mueve al elemento respecto a su posición natural, sino con el primer contenedor que esté posicionado de forma diferente a static.
Para localizarlo, se recorren todos los elementos contenedores empezando por el más cercano al elemento que deseamos mover. Si no hay ninguno, la posición se realizará respecto a la ventana del navegador.
Ejemplo:
Mover el enlace 40 píxeles desde la parte superior de la ventana y 40 desde la parte de la izquierda.

```
a {position:absolute; top:40px; left:40px}
```

Posicionamiento

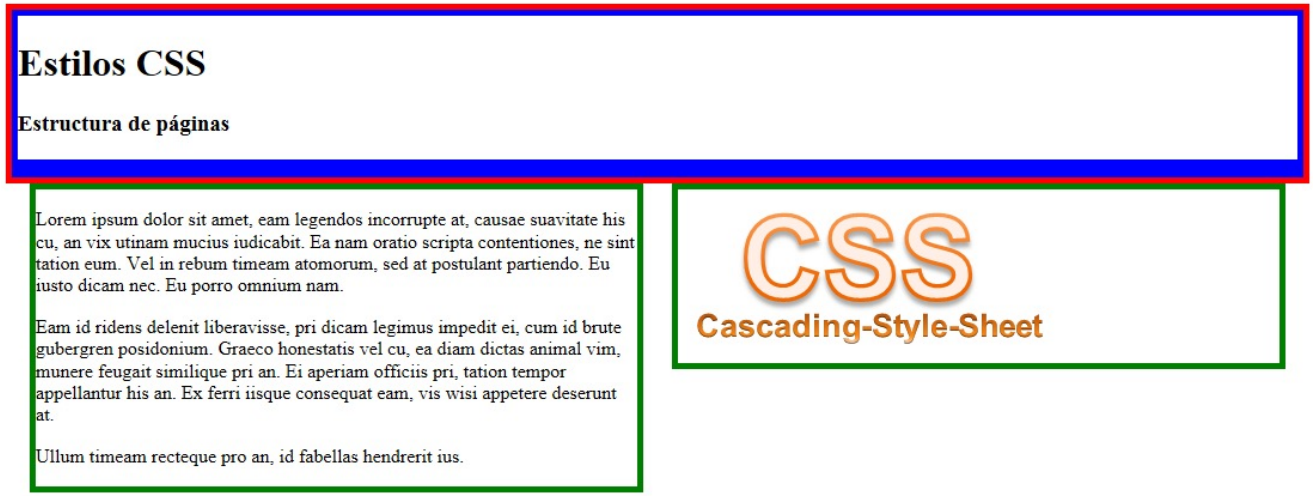
El posicionamiento estático es similar al posicionamiento absoluto, pero al desplazarse la barra de desplazamiento, de la página, el elemento no se desplaza y siempre ocupa la misma posición en la pantalla.

La línea en los estilos sería:

```
a {position:fixed; top:40px; left:40px}
```

Float

Con los posicionamientos vistos hasta ahora, no pueden ponerse dos cajas en la misma línea. Sólo una debajo de otra. Esto se puede solucionar con la propiedad float.

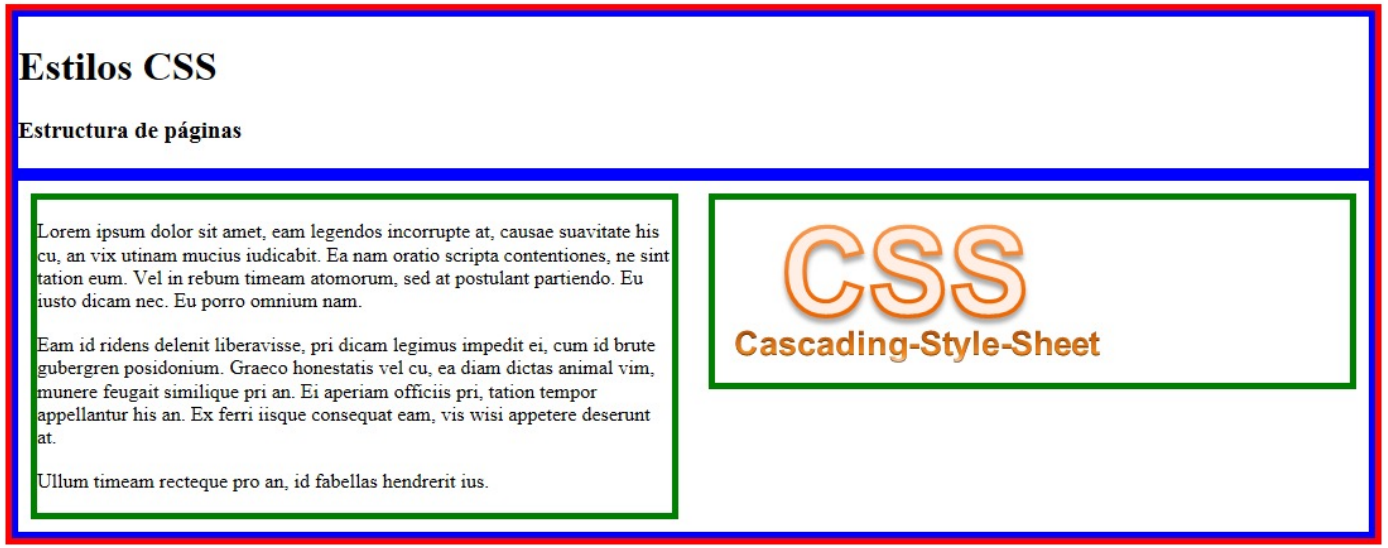


En la imagen anterior, las cajas verdes están colocadas dentro de la roja, debajo de la azul, pero al poner la propiedad float en ellas, y “flotar” se han salido de dichas cajas.

Para solucionar dicho problema utilizamos la propiedad float

Float

La propiedad clear nos permite “romper”, la propiedad float.
Es una propiedad que nos ayuda a solucionar el problema de que, al flotar las cajas, se salgan de las cajas que las contenían.
Así solucionaríamos el problema anterior



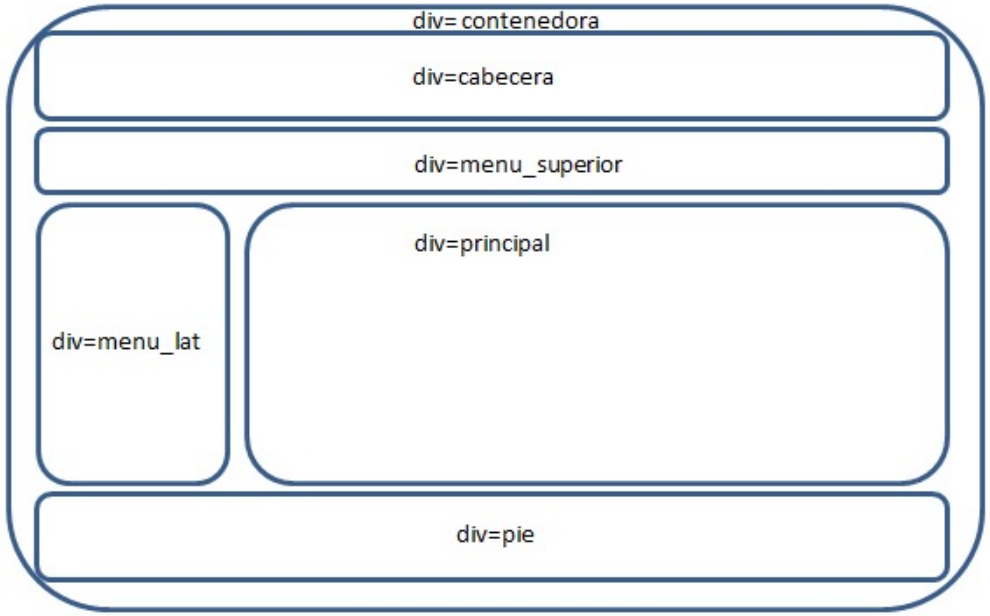
Float

```html
<div id="contenedora">
    <div id="encabezado">
        <h1> Estilos CSS </h1>
        <h3> Estructura de páginas </h3>

    </div> <!-- fin de encabezado-->
    <div id="contenido">
        <div id="texto">

            <p> Lorem ipsum dolor sit amet, eam legendos incorrupte at, causae suavitate his cu, an vix utinam mucius iudicabit.
                Ea nam oratio scripta contentiones, ne sint tation eum. Vel in rebum timeam atomorum, sed at postulant partiendo.
                Eu iusto dicam nec. Eu porro omnium nam.
            </p>
            <p>Eam id ridens delenit liberavisse, pri dicam legimus impedit ei, cum id brute gubergren posidonium.
            Graeco honestatis vel cu, ea diam dictas animal vim, munere feugait similique pri an. Ei aperiam officiis pri,
            tation tempor appellantur his an. Ex ferri iisque consequat eam, vis wisi appetere deserunt at.
            </p>
            <p>Ullum timeam recteque pro an, id fabellas hendrerit ius.</p>
        </div>
        <div id="imagen">
            <img src="../images/css.png">
        </div>
        <div id="ultimo" class="romper"></div>
    </div>
</div>
```

```css
#contenedora{border: solid red 5px;
             width:80%;margin:0 auto;}
#encabezado{border: solid blue 5px;}
#contenido{border: solid blue 5px;}
#texto{border: solid green 5px;
       margin:10px;
       float:left; width:47%}
#imagen{border: solid green 5px;
       margin:10px;
       float:right;width:47%;}
.romper{clear:both;}
```

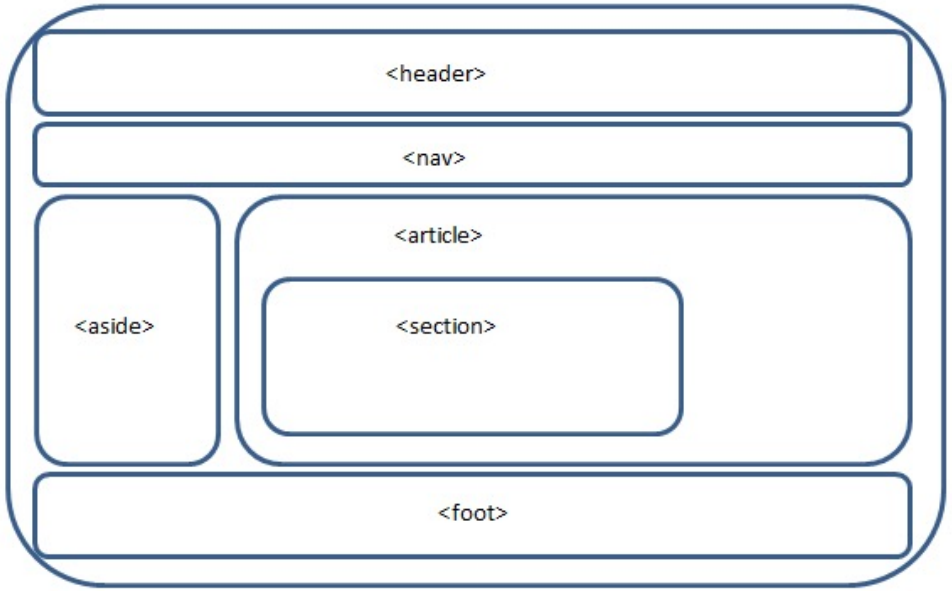
Layout

La estructura de la página se suele conocer como **layout.**
La página se divide en distintos bloques formados por cajas, con distintos posicionamientos.
Un ejemplo de layout podría ser el siguiente:

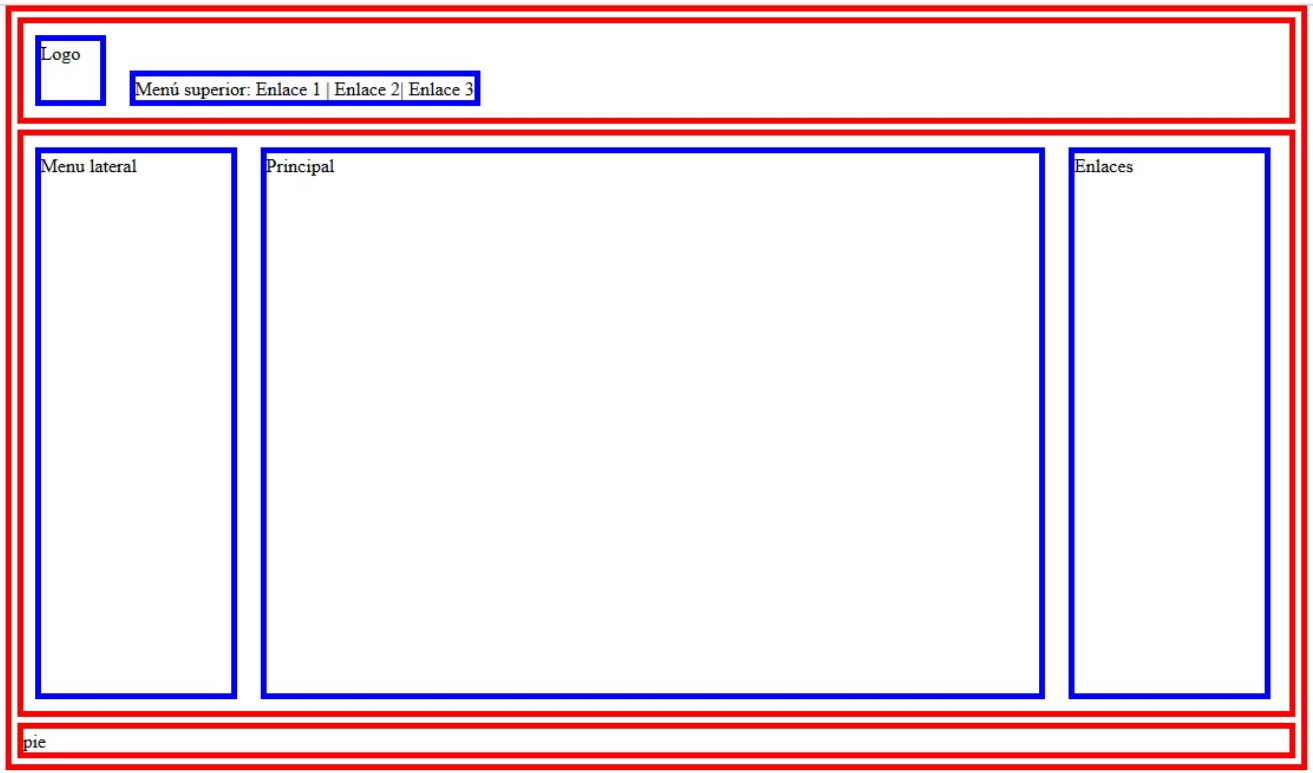


Layout

En html5, en lugar de utilizar todas las etiquetas div con su respectivo identificador, se han desarrollado etiquetas propias para cada zona



Actividad


Logo
Menú superior: Enlace 1 | Enlace 2| Enlace 3
Menu lateral
Principal
Enlaces
pie