CSS – Modelo de Caja y Posicionamiento

Lenguajes de Marcas y SGI

Profesora: Ines Menendez

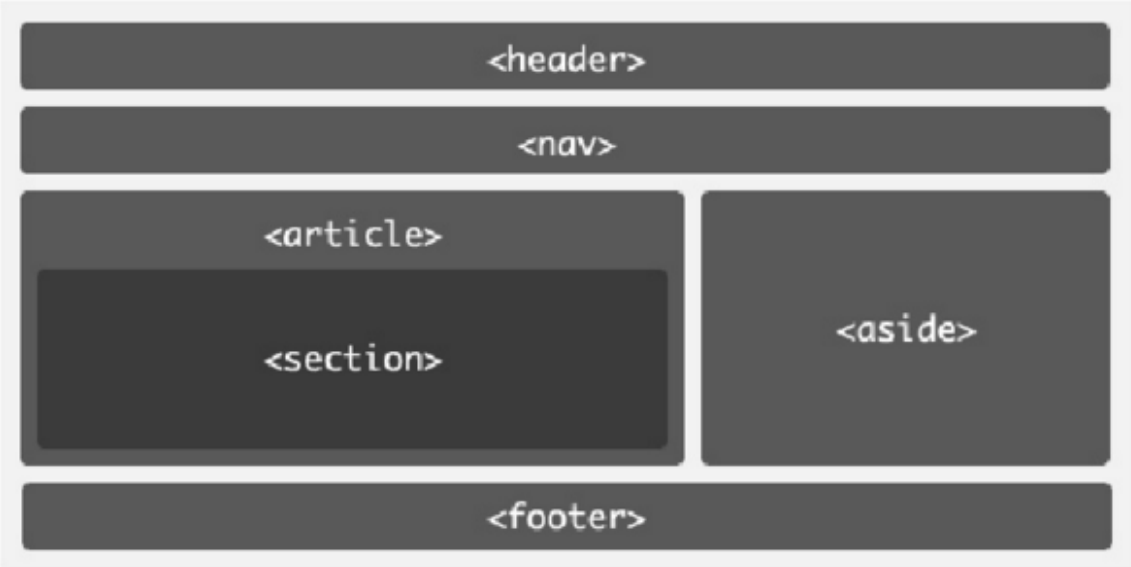
Bibliografía:

- Lenguajes de Marcas y SGI, Editorial Garceta
- HTML&CSS, Curso práctico, Sergio Luján, Editorial Altaria
- Tutorial CSS - w3schools, https://www.w3schools.com/css/default.asp
- Introducción a CSS, http://librosweb.es/libro/css/

1 Ejemplo de una estructura básica de un sitio

Esta es una estructura bastante común de distribución del contenido. Ninguno de los 'divs' tiene *significado semántico*:

HTML5 incorpora significado semántico a la misma estructura, con unas nuevas marcas 'estructurales' que hacen que ya no sea necesario el uso de ***divs***:

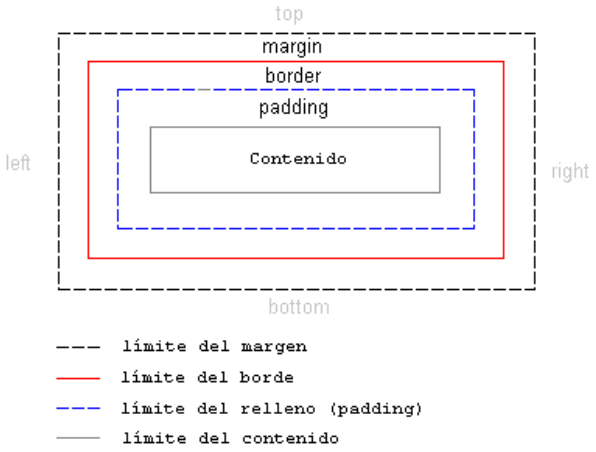


A lo largo de este curso vamos a ver cómo conseguir una estructura similar a la anterior para nuestro sitio web.

2 El modelo de caja

El modelo de cajas es el elemento fundamental de CSS porque determina la composición de la página web. TODOS los elementos que se insertan en el documento HTML se representan automáticamente mediante cajas rectangulares.

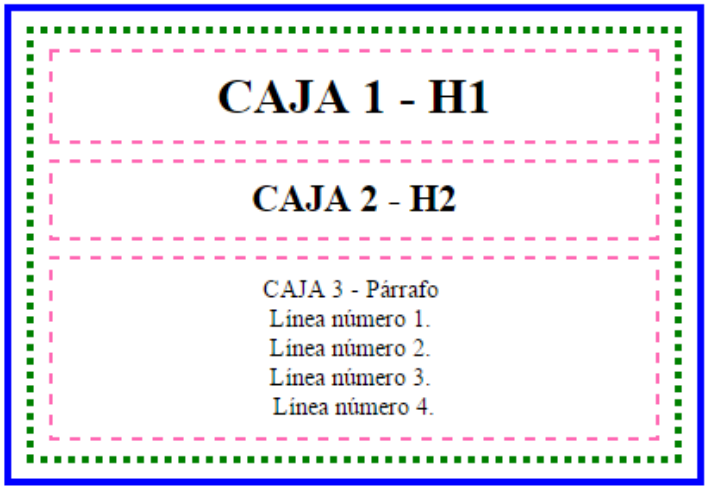
El modelo de caja en CSS describe las cajas que se generan a partir de los elementos HTML. El modelo de caja también contiene opciones detalladas en lo referente al ajuste de márgenes, bordes, relleno (padding) y contenido de cada elemento. La siguiente imagen muestra cómo se construye el modelo de caja:



Las "cajas" son invisibles salvo que les asignemos bordes o color de fondo.

Un ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Ejemplo de Cajas</title>
  <style>
    html {
      border: 5px solid blue;
      margin: 10px;
    }
    body {
      border: 5px dotted green;
      text-align: center;
      margin: 10px;
    }
    p,
    h1,
    h2 {
      border: 3px dashed hotpink;
      text-align: center;
      margin: 10px;
      padding: 10px;
    }
  </style>
</head>

<body>
  <h1>CAJA 1 - H1</h1>
  <h2>CAJA 2 - H2</h2>
  <p>CAJA 3 - Párrafo
    <br/>Línea número 1.
    <br/>Línea número 2.
    <br/>Línea número 3.
    <br/>Línea número 4.
  </p>

</body>
```



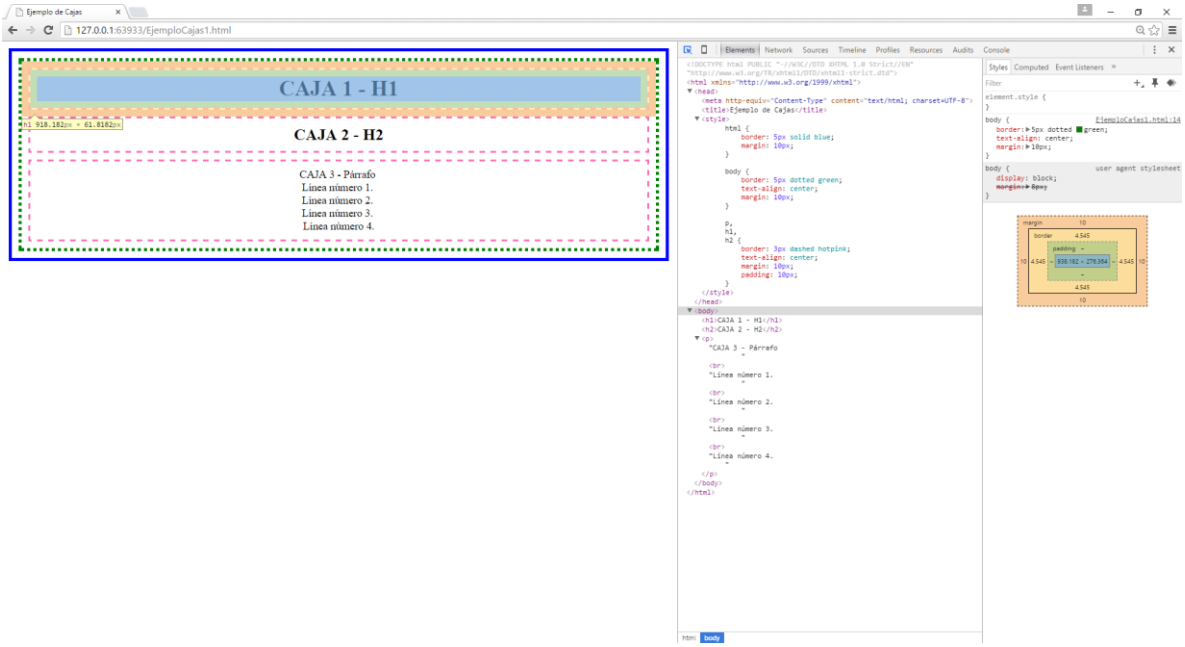
Las partes que componen cada caja son las siguientes:

- Contenido: texto, imágenes, etc. del elemento.
- Relleno (padding): espacio entre el contenido y el borde.

- Borde (border): línea que encierra completamente el contenido y su relleno.
- Imagen de fondo (background-image): imagen que se muestra por detrás del contenido y el espacio de relleno.
- Color de fondo (background-color): color que se muestra por detrás del contenido y el espacio de relleno.
- Margen (margin): espacio libre entre una caja y las cajas adyacentes.

✓ Padding y margin son transparentes.
✓ En el espacio ocupado por 'padding' se muestra el color o imagen de fondo.
✓ En el espacio ocupado por 'margin' se muestra el color o imagen de fondo de su elemento padre.
✓ Si ningún elemento padre tiene definido un color o imagen de fondo, se muestra el color o imagen de fondo de la página.
✓ Si una caja define tanto un color como una imagen de fondo, la imagen tiene más prioridad y se visualiza antes que el color.
✓ Si la imagen de fondo no cubre totalmente la caja del elemento, o si la imagen tiene zonas transparentes, entonces se visualiza el color de fondo en dichas zonas.

Los navegadores Chrome y Firefox, entre otros, cuentan con '*Herramientas para el desarrollador*' que permiten examinar el modelo de cajas y hacer cambios 'en tiempo real' sobre los estilos aplicados a los diversos elementos.

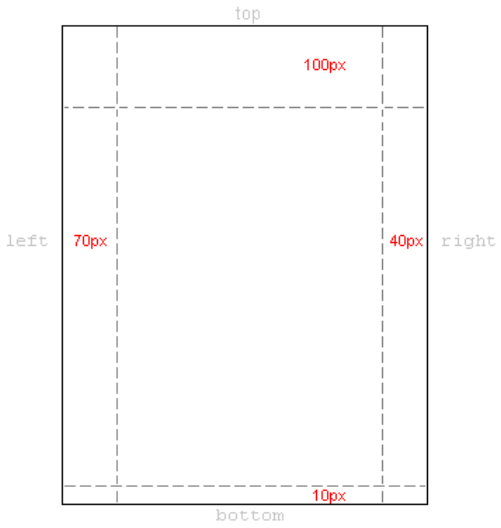


2.1 Establecer el margen de un elemento

Todo elemento tiene cuatro lados: derecho, izquierdo, superior e inferior. La propiedad *margin* hace referencia a la distancia desde cada lado respecto al elemento colindante (o respecto a los bordes del documento).
En un primer ejemplo, veremos cómo definir los márgenes del documento en sí, es decir, del elemento *<body>*. La imagen siguiente muestra cómo queremos que sean los márgenes de nuestras páginas.

El código CSS necesario para esto es el siguiente:

```
body {
        margin-top: 100px;
        margin-right: 40px;
        margin-bottom: 10px;
        margin-left: 70px;     }
```



O podrías elegir usar la versión abreviada de *margin*:

```
body {  margin: 100px 40px 10px 70px;}
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
body {
            margin: 100px 40px 10px 70px;
      }

</style>
</head>

<body>
<h1>Documento Con Márgenes</h1>

<p>Iste quidem veteres entre ponetur honeste, qui vel mense Brevi
vel toto est iunior anno. Utor permisso, caudaeque pilos ut unum
equinae paulatim vello, demo etiam unum, dum cadat elusus ratione
ruentis acervi, qui REDIT en Fastos et annis miraturque.</p>

<p>Ennio et sapines et Fortis et alter Homero, ut criticada dicunt,
leviter curare videtur, quo promissa Cadant et insomnio
Pythagorea.Naevius en manibus non est et mentibus haeret recens
paene? Adeo sanctum est vetus omne poema. Quotiens Ambigitur, antes
de sentarse, Pacuvio docti.</p>

<h2>Os ediscit et hos arto stipata</h2>
```

Your Result:

Documento Con Márgenes

Iste quidem veteres entre ponetur honeste, qui vel mense Brevi vel toto est iunior anno. Utor permisso, caudaeque pilos ut unum equinae paulatim vello, demo etiam unum, dum cadat elusus ratione ruentis acervi, qui REDIT en Fastos et annis miraturque.

Ennio et sapines et Fortis et alter Homero, ut criticada dicunt, leviter curare videtur, quo promissa Cadant et insomnio Pythagorea.Naevius en manibus non est et mentibus haeret recens paene? Adeo sanctum est vetus omne poema. Quotiens Ambigitur, antes de sentarse, Pacuvio docti.

Os ediscit et hos arto stipata

Os ediscit et hos arto stipata Teatro Roma spectat potens; habet hos numeratque poetas ad tempus nostrum Livi scriptoris ab AEVO, si nimis antiguos, si miento.

Se puede establecer los márgenes de casi todos los elementos del mismo modo. Por ejemplo, podemos elegir definir márgenes para todos los párrafos de texto marcados con el elemento *<p>:*

```
body { margin: 100px 40px 10px 70px;     }
p {margin: 5px 50px 5px 50px;     }
```

2.2 Establecer el relleno de un elemento

La propiedad *padding* puede entenderse como "relleno". Esto tiene sentido puesto que el relleno *(padding)* no afecta a la distancia de un elemento respecto a otros elementos, sino que sólo define la distancia interior entre el borde y el contenido del elemento.

El uso de la propiedad *padding* se puede ilustrar viendo un sencillo ejemplo en el que todos los títulos tienen diferentes colores de fondo:

```
h1 {
        background: yellow;
}

h2 {
        background: orange;
}
```

Encabezados y relleno

Ennio et sapines et Fortis et alter Homero, ut criticada dicunt, leviter curare videtur, quo promissa Cadant et insomnio Pythagorea.Naevius en mar omne poema. Quotiens Ambigitur, antes de sentarse, Pacuvio docti.

Os ediscit et hos arto stipata

Indignor quicquam reprehendi, no quia crasse compositum illepedeve putetur, sed quia nuper ncop veniam antiquis, sed honorem et praemia posc periisse pudorem cuncti paene patres, ea cum reprehendere coner, quae grave Aesopus, quae DOCTUS Roscio EGIT; vel recto quia nula, nisi q imberbes senes.

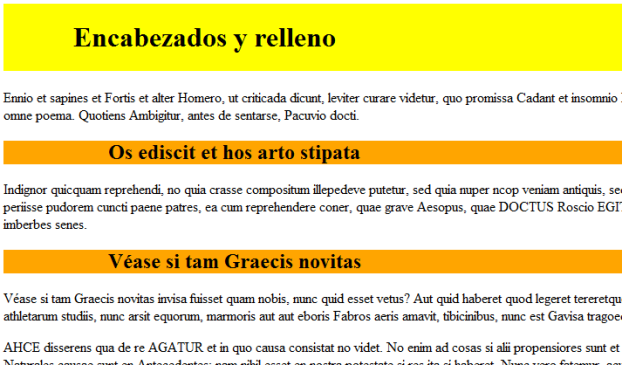
Véase si tam Graecis novitas

Véase si tam Graecis novitas invisa fuisset quam nobis, nunc quid esset vetus? Aut quid haberet quod legeret tereretque viritim.Ut primum positis athletarum studiis, nunc arsit equorum, marmoris aut aut eboris Fabros aeris amavit, tibicinibus, nunc est Gavisa tragoedis; Puella.

AHCE disserens qua de re AGATUR et in quo causa consistat no videt. No enim ad cosas si alii propensiores sunt et propter Causas Naturales Naturales causae sunt en Antecedentes; nam nihil esset en nostra potestate si res ita si haberet. Nunc vero fatemur, acuti hebetesne, Valente imbe ut sedeamus quidem aut ambulemus voluntatis esse, no es videt quae quamque rem res consequatur. AHCE disserens qua de re AGATUR et in q propter Causas Naturales Antecedentes, idciro etiam nostrarum voluntatum atque appetitionum Naturales causae sunt en Antecedentes; nam nihil hebetesne, Valente imbecilline simus, no esse Identificación en nobis, qui autem ex eo COGI putat ne ut sedeamus quidem aut ambulemus volunta

Al definir el *padding* para los títulos, cambiamos la cantidad de "relleno" que habrá alrededor del texto en cada uno de ellos:

```
h1 {
        background: yellow;
        padding: 20px 20px 20px 80px;
}

h2 {
        background: orange;
        padding-left:120px;}
```



2.3 Bordes

Los bordes se pueden usar para muchas cosas, por ejemplo, como elemento decorativo o para subrayar la separación entre dos cosas. CSS te ofrece opciones sin fin a la hora de usar bordes en tus páginas. En esta lección vamos a examinar las siguientes propiedades CSS:

- border-width
- border-color
- border-style

2.3.1 Anchura del borde [border-width]

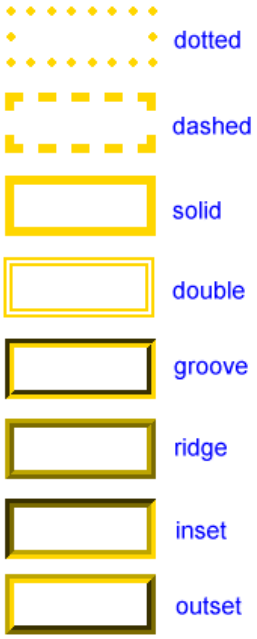
La anchura del borde se define por medio de la propiedad *border-width*, que dispone de los valores *thin*, *medium* y *thick*, o de un valor numérico indicado en píxeles. La siguiente imagen ilustra cómo funciona el sistema:

2.3.2 Color del borde [border-color]

La propiedad *border-color* define el color del borde. Los valores de esta propiedad son los valores de color normales, por ejemplo, "#123456" (en notación hexadecimal), "rgb(123,123,123)" (en notación RGB) o *"yellow"* (por nombre del color).

2.3.3 Estilo de borde [border-style]

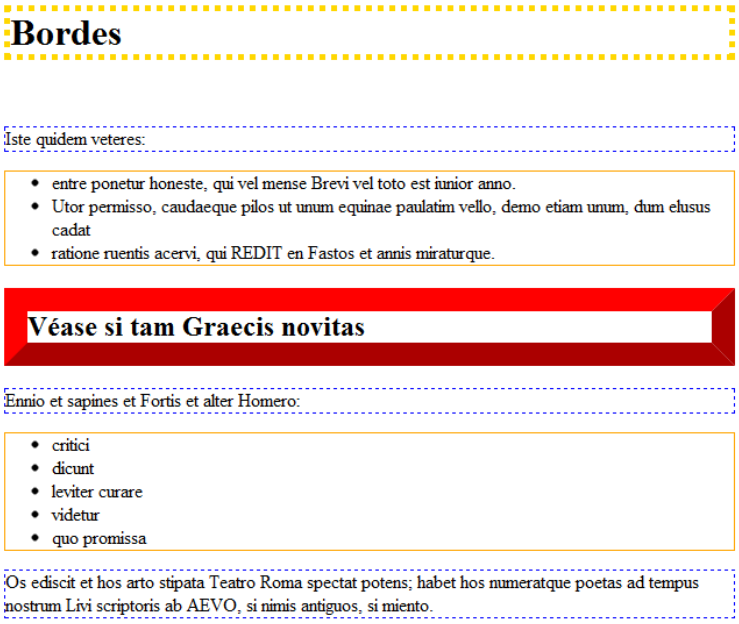
Se puede elegir entre diferentes estilos de borde. Al lado se muestran 8 estilos de borde. Todos los ejemplos se muestran con el valor del color a *"oro"* y el valor de la anchura a *"thick"*, pero se pueden mostrar, por supuesto, en otros colores y grosores.
Si no queremos mostrar ningún borde, se puede usar los valores *none* o *hidden*.



2.4 Ejemplos de definición de bordes

Las tres propiedades descritas anteriormente se pueden unir para cada elemento y así producir diferentes bordes. Para ilustrar esto, veremos un documento en el que definimos diferentes bordes para los elementos <h1>, <h2>, <ul> y <p>. El resultado puede que no sea demasiado bonito, pero ilustra gráficamente algunas de las muchas posibilidades:

```
h1 {
        border-width: thick;
        border-style: dotted;
        border-color: gold;
}

h2 {
        border-width: 20px;
        border-style: outset;
        border-color: red;
}

p {
        border-width: 1px;
        border-style: dashed;
        border-color: blue;
}

ul {
        border-width: thin;
        border-style: solid;
        border-color: orange;
}
```



También es posible declarar propiedades especiales para el borde superior (top), inferior (bottom), derecho (right) e izquierdo (left). Por ejemplo:

```
h1 {
        border-top-width: thick;
        border-top-style: solid;
        border-top-color: red;

        border-bottom-width: thick;
        border-bottom-style: solid;
        border-bottom-color: blue;

        border-right-width: thick;
        border-right-style: solid;
        border-right-color: green;

        border-left-width: thick;
        border-left-style: solid;
        border-left-color: orange;
}
```

2.5 Combinación de propiedades [border]

Como ocurre con muchas otras propiedades, usando la propiedad *border* se pueden combinar otras muchas propiedades en una sola. Veamos un ejemplo:

```
p {
        border-width: 1px;
        border-style: solid;
        border-color: blue;
}
```

La declaración anterior se puede combinar (abreviar) así:

```
p {
        border: 1px solid blue;
}
```

1.1. Altura y anchura

Hasta ahora, no hemos prestado demasiada atención a las dimensiones de los elementos con los que hemos estado trabajando. En esta lección examinaremos lo fácil que es definir la altura y anchura de un elemento. Para lo cual usaremos las propiedades:

- width
- height

2.5.1 Estableciendo la propiedad width

Con la propiedad *width* se puede definir la anchura concreta de un elemento.
El sencillo ejemplo que sigue nos proporciona una caja en la que se puede introducir texto:

```
div.box {
        width: 200px;
        border: 1px solid black;
        background: orange;
}
```

```
<div class="box">volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et Iova iudicat aequo.Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilium </div>
```

volgus videt recto, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis Comparet, errat. Si quaedam nimis antiguos, si peraque miento dicere crédito eos, ignave Multa fatetur, et SAPIT et cum et facit Iova iudicat aequo.Non equidem Insector delendave Carmina Livi esse reorganización, Memini quae mihi plagosum parvo Orbilium

2.5.2 Estableciendo la propiedad height

Fíjate cómo en el ejemplo anterior la altura de la caja queda establecida por el contenido de la misma. Se puede influir en la altura de un elemento con la propiedad *height*. Por ejemplo, probemos a fijar la altura de la caja en *500px*:

```
div.box {
        height: 500px;
        width: 200px;
        border: 1px solid black;
        background: orange;
}
```

volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et Iova iudicat aequo.Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilium

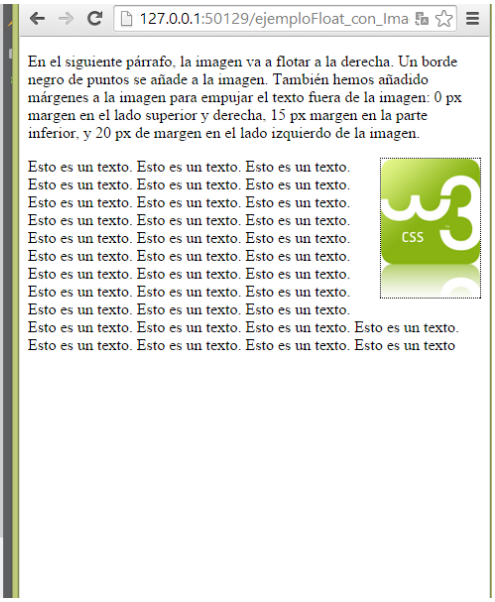
2.6 La propiedad *float*

La propiedad ***float*** establece el esquema de posicionamiento flotante para un elemento. Cuando existe un elemento flotante, los elementos que se encuentran a continuación del elemento flotante fluyen a lo largo de él, salvo que haya un elemento que tenga establecido la propiedad ***clear***.
Las propiedades ***float*** y ***clear*** se pueden aplicar a cualquier elemento de una página web.

Cuando se posiciona una caja de forma flotante: *La caja deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante.* La caja flotante se posiciona lo más a la izquierda o lo más a la derecha posible de la posición en la que se encontraba originalmente.

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

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/>
    <title>Imagen float</title>
</head>
<style>
img {
    float: right;
    border: 1px dotted black;
    margin: 0px 0px 15px 20px;
}
</style>
</head>
<body>

<p>En el siguiente párrafo, la imagen va a flotar a la derecha. Un borde
negro de puntos se añade a la imagen. También hemos añadido márgenes a
la imagen para empujar el texto fuera de la imagen: 0 px margen en el
lado superior y derecha, 15 px margen en la parte inferior, y 20 px de
margen en el lado izquierdo de la imagen.
</p>

<p><img src="w3css.gif" width="100" height="140">
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto.
Esto es un texto. Esto es un texto. Esto es un texto. Esto es un texto
</p>

</body>
</html>
```



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

Un ejemplo de una imagen que flota a la derecha del texto:

La propiedad ***clear*** permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante.

Propiedad	clear
Valores	none | left | right | both | inherit
Se aplica a	Todos los elementos de bloque
Valor inicial	none
Descripción	Indica el lado del elemento que no debe ser adyacente a ninguna caja flotante

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

La especificación oficial de CSS explica este comportamiento como *"un desplazamiento descendente hasta que el borde superior del elemento esté por debajo del borde inferior de cualquier elemento flotante hacia la izquierda"*.
Si se indica el valor *right*, el comportamiento es análogo, salvo que en este caso se tienen en cuenta los elementos desplazados hacia la derecha.

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

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

2.7 La propiedad display

Propiedad	display
Valores	inline | block | none | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | inherit
Se aplica a	Todos los elementos
Valor inicial	inline
Descripción	Permite controlar la forma de visualizar un elemento e incluso ocultarlo

La propiedad ***display*** modifica la forma en la que se visualiza un elemento.

Los valores más utilizados son *inline, block* y *none*. El valor *block* muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate. El valor *inline* visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate.

El valor *none* oculta un elemento y hace que desaparezca de la página. El resto de elementos de la página se visualizan como si no existiera el elemento oculto, es decir, pueden ocupar el espacio en el que se debería visualizar el elemento.

2.8 La propiedad overflow.

Normalmente, los contenidos de un elemento se pueden mostrar en el espacio reservado para ese elemento. Sin embargo, en algunas ocasiones el contenido de un elemento no cabe en el espacio reservado para ese elemento y se desborda.

La situación más habitual en la que el contenido sobresale de su espacio reservado es cuando se establece la anchura y/o altura de un elemento mediante la propiedad *width* y/o *height*. Otra situación habitual es la de las líneas muy largas contenidas dentro de un elemento *<pre>*, que hacen que la página entera sea demasiado ancha.

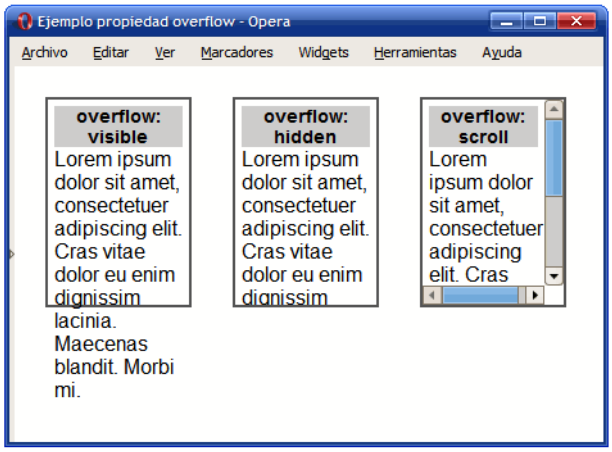
CSS define la propiedad ***overflow*** para controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.

Propiedad	overflow
Valores	visible | hidden | scroll | auto | inherit
Se aplica a	Elementos de bloque y celdas de tablas
Valor inicial	visible
Descripción	Permite controlar los contenidos sobrantes de un elemento

Los valores de la propiedad overflow tienen el siguiente significado:

- visible: el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.
- hidden: el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.
- scroll: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de *scroll* que permiten visualizar el resto del contenido.
- auto: el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.

La siguiente imagen muestra un ejemplo de los tres valores típicos de la propiedad overflow:




```
div {
  display: inline;
  float: left;
  margin: 1em;
  padding: .3em;
  border: 2px solid #555;
  width: 100px;
  height: 150px;
  font: 1em Arial, Helvetica, sans-serif;
}

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

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

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

3 Estuctura HTML y CSS

- Los elementos HTML pueden visualizarse como bloque o en línea.
- Los elementos de bloque ***block*** ocupan todo el ancho de la página y fuerzan a una nueva línea antes y después.
- Son elementos de bloque:

```
<address> <article> <aside> <audio>  <blockquote> <canvas>
<dd> <div> <dl> <fieldset> <figcaption> <figure> <footer>
<form> <h1>, <h2>, <h3>, <h4>, <h5>, <h6> <header> <hgroup>
<hr> <noscript> <ol> <output>  <p> <pre> <section>  <table>
<tfoot> <ul> video>
```

- Los elementos en línea ***inline*** sólo ocupan el ancho necesario y no fuerzan nuevas líneas.
- Son elementos de línea:

```
<b> <big> <i> <small> <tt>
<abbr> <acronym> <cite> <code> <dfn> <em> <kbd> <strong>
<samp> <var>
<a> <bdo> <br> <img> <map> <object> <q> <script> <span> <sub>
<sup>
<button> <input> <label> <select> <textarea>
```

3.1 El elemento div

- El elemento *** *** es un elemento de bloque que se emplea para contener en su interior otros elementos HTML.
- El elemento *** *** no tiene ningún significado, es semánticamente neutro.
- Empleando CSS, el elemento *** *** se puede utilizar para estructurar un documento en bloques de contenido y para poder ser formateados.
- Los *** *** se nombran mediante un identificador **ID** o una clase **class**.
- Una etiqueta div puede tener instrucciones de formato (de la familia de fuente, color, bordes, etc.), atributos de altura y anchura y posicionamiento.
- Hay que tener cuidado al emplear el tamaño, el contenido según el navegador puede desbordar el tamaño de un div. Evitar definir 'altos' si no son necesarios.

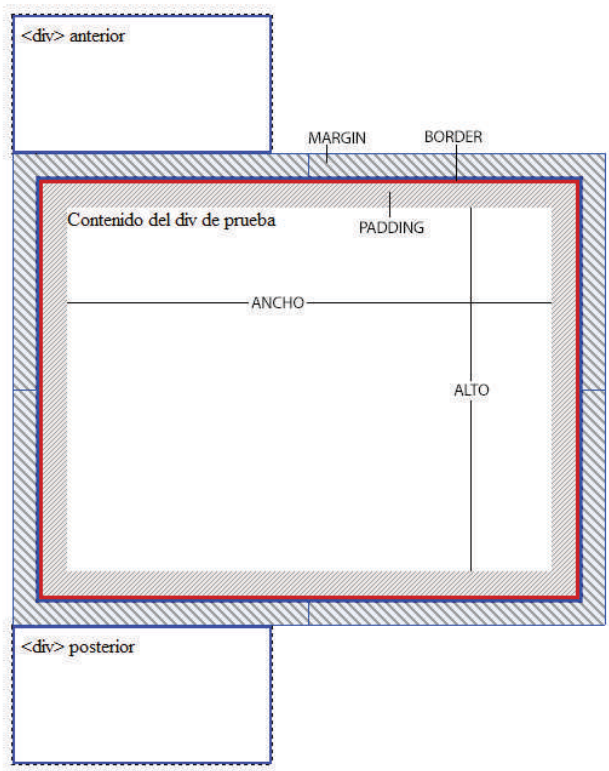
La estructura HTML debe realizarse en el sentido lógico de lectura del documento, para que cuando la página se vea sin aplicar los estilos no pierda su significado:

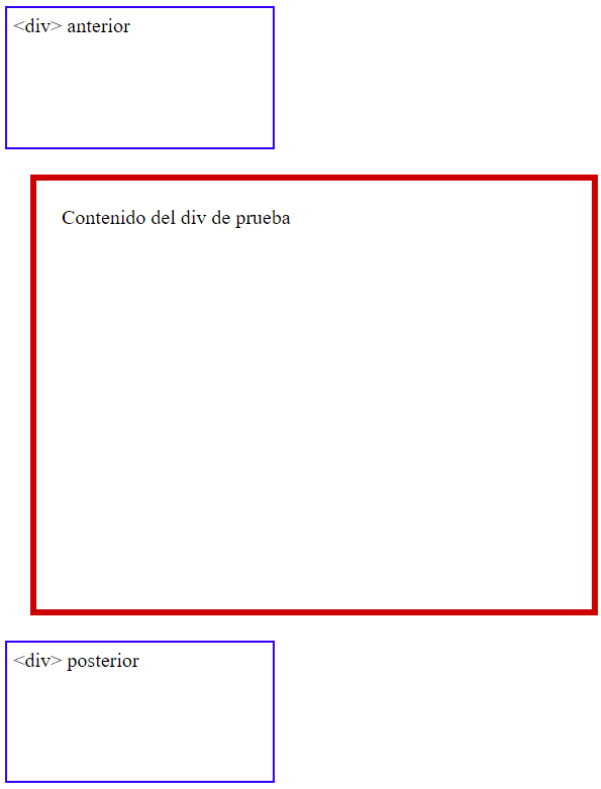
4 Ejemplos de estructura, divs, float y clear

Código del ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Ejemplo de estructura con divs</title>
    <style type="text/css">
        .muestra {
            height: 300px;
            width: 400px;
            border: 5px solid #C00;
            margin: 20px;
            padding: 20px;
        }

        .anterior,
        .posterior {
            height: 100px;
            width: 200px;
            border: 2px solid #30F;
            padding: 5px;
        }
    </style>

</head>

<body>
    <div class="anterior">&lt;div&gt; anterior</div>
    <div class="muestra">Contenido del div de prueba</div>
    <div class="posterior">&lt;div&gt; posterior</div>
</body>
```

La explicación y cómo se ve en el navegador:





CSS permite posicionar los ***div*** en la página, con ***float*** y ***clear***.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Ejemplo Float 1</title>

    <style type="text/css">
        .muestra {
            height: 300px;
            width: 400px;
            background-color: darkorange;
        }

        .anterior {
            height: 100px;
            width: 200px;
            float: left;
            background-color: aquamarine;
        }

        .posterior {
            height: 100px;
            width: 200px;
            background-color: blueviolet;
        }
    </style>

</head>

<body>
    <div class="anterior">&lt;div&gt; anterior con float</div>
    <div class="muestra">Contenido del div muestra</div>
    <div class="posterior">&lt;div&gt; posterior</div>
</body>
```

Con float el div 'flota' a una posición relativa.

Con ***clear*** rompe el esquema del ***float***: *(fijarse en el uso de porcentaje para definir anchos)*


```
<style type="text/css">
    .muestra {
        height: 300px;
        width: 60%;
        background-color: darkorange;
        float: left;
    }

    .anterior {
        height: 100px;
        width: 20%;
        float: left;
        background-color: aquamarine;
    }

    .posterior {
        height: 100px;
        width: 80%;
        background-color: blueviolet;
        clear: both;
    }
</style>
```

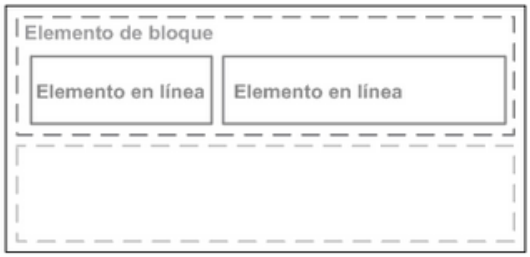
6 Posicionamiento

CSS tiene 5 formas de establecer la posición en pantalla de una caja:
Estático, relativo, absoluto, fijo y ***flotante.***

Para establecer la posición se dispone de las propiedades:
position, top, bottom, right, left y ***float***

6.1 Posicionamiento estático.

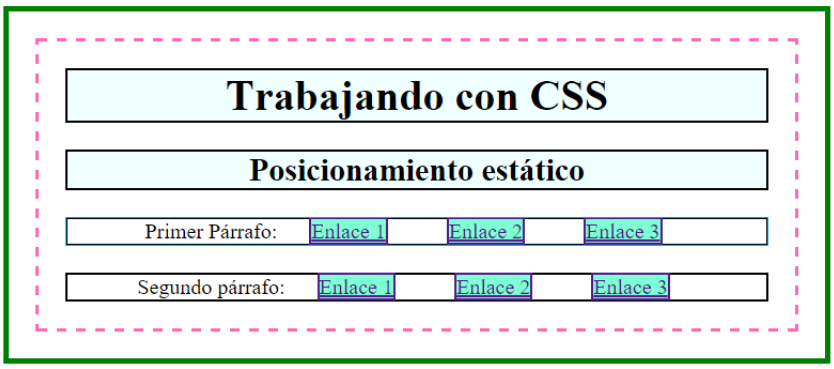
position: static es el valor por defecto que utilizan los navegadores.
Cuando definimos la posición de un elemento como estática indicamos que siga el flujo normal de la página, y que se sitúe en el lugar que le corresponde acorde con el modelo de formato visual de la página.



Elemento contenedor

Un ejemplo con **posicionamiento estático**:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>posicionamiento estático</title>
    <style type="text/css">
        * {
            border: 2px solid;
            text-align: center;
            margin: 20px;
        }

        html {
            border: 4px solid green;
        }

        body {
            border: dashed hotpink;
        }

        h1,
        h2 {
            background-color: azure;
        }

        a {
            background-color: aquamarine;
        }
    </style>
</head>

<body>
    <h1>Trabajando con  CSS</h1>
    <h2>Posicionamiento estático</h2>
    <p>Primer Párrafo:
        <a href="#">Enlace 1</a>
        <a href="#">Enlace 2</a>
        <a href="#">Enlace 3</a>
    </p>
    <p>Segundo párrafo:
        <a href="#">Enlace 1</a>
        <a href="#">Enlace 2</a>
        <a href="#">Enlace 3</a>
    </p>


</body>

</html>
```



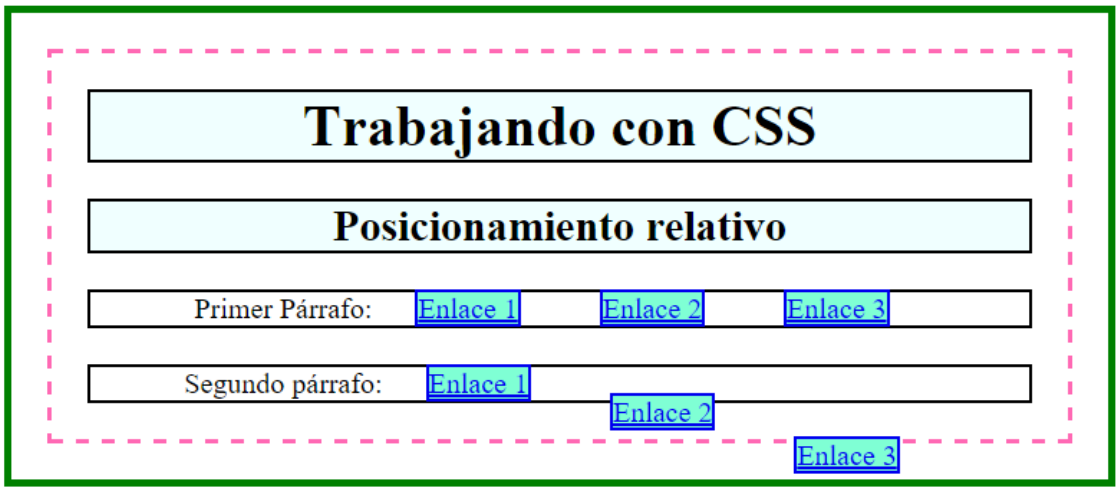
6.2 Posicionamiento relativo.

```css
a.abajo {
    position: relative;
    top: 1em;
}

a.superBajo {
    position: relative;
    top: 2.5em;
}
```

```html
<p>Segundo párrafo:
    <a href="#">Enlace 1</a>
    <a href="#" class="abajo">Enlace 2</a>
    <a href="#" class="superBajo">Enlace 3</a>
</p>
```

Un elemento posicionado con esta propiedad se situará respecto a la posición que debería tener siguiendo el flujo normal de la página, es decir, toma como base de sus coordenadas de posición, la posición en la que debería estar situado. Esto no afecta a los demás elementos, ya que el elemento se sale del flujo. Siguiendo con el ejemplo anterior, ponemos dos de los enlaces con las clases *abajo* y *superBajo*, y les damos las propiedades ***position: relative*** y un desplazamiento:

El resultado:



6.3 Posicionamiento absoluto.

Al posicionar elementos de manera absoluta rompemos el flujo de la página, y situamos los elementos en un nuevo plano de la página. Con este valor posicionamos elementos respecto a su padre, o 'elemento contenedor'. Los elementos a continuación de un elemento posicionado de manera absoluta no se situarán respecto a éste, porque ha abandonado el flujo de la página.

Para posicionar un elemento de manera absoluta debemos concretar la distancia respecto al borde superior e izquierdo del elemento contenedor usando las propiedades *left* y *top*, como en el siguiente ejemplo.

```
    <style type="text/css">
        * {
            border: 2px solid;
            text-align: center;
            margin: 20px;
        }

        html {
            border: 4px solid green;
        }

        body {
            border: dashed hotpink;
        }

        h1,
        h2 {
            background-color: azure;
        }

        div {
            border: 2px solid blue;
        }

        a {
            background-color: aquamarine;
        }

        a.absoluto {
            background-color: chocolate;
            position: absolute;
            top: 20px;
            left: 20px;
        }

        p {
            border: none;
        }
    </style>
</head>

<body>
    <h1>Trabajando con  CSS</h1>
    <h2>Posicionamiento absoluto</h2>
    <div>
        <p>Caja con posición estática</p>
        <a href="#">Enlace 1</a>
        <a href="#" class="absoluto">Enlace 2</a>
        <a href="#">Enlace 3</a>

    </div>
```

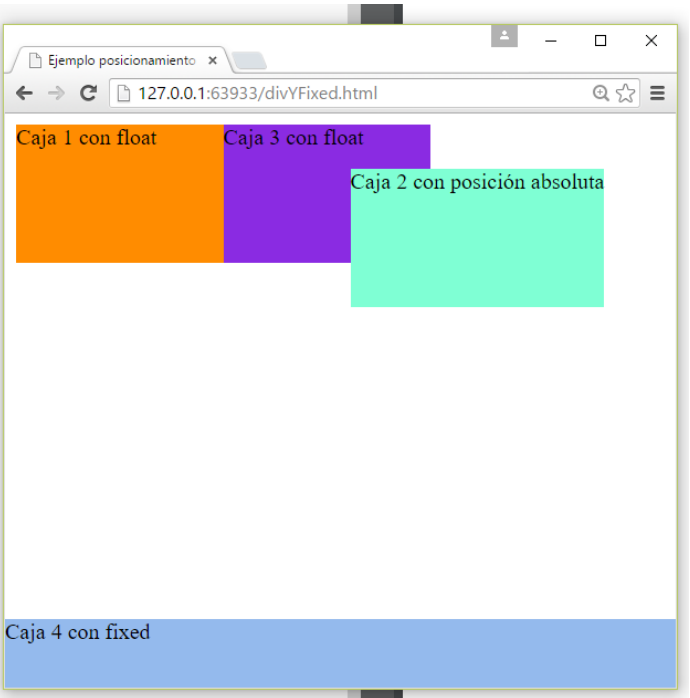
En este caso 'Enlace2' se desplaza 20px desde arriba (top) y desde la izquierda (left), partiendo desde la esquina superior izquierda de la página, porque *div* tiene posición normal y el siguiente contenedor que no es **static** es *body*.
La caja 'Enlace 3' ocupa el hueco dejado por 'Enlace 1'.

En este otro ejemplo de ***posicionamiento absoluto*** cambiamos el tipo de posicionamiento de * * a relativo. Esto hará que la posición de la caja 'Enlace 2' ahora se calcule con respecto a * *, que es la primera caja con posicionamiento no estático.
[Hay que tener en cuenta tb. el margen]

6.4 Posicionamiento fijo.

Es similar al posicionamiento absoluto, pero al desplazar la página las cajas se mantienen fijas en el mismo lugar.
Esto resulta útil para encabezados o pies de página que se mantienen inalterables a lo largo de un documento extenso.
También es posible aplicar el posicionamiento ***fixed*** a un menú de navegación lateral que se necesite que esté siempre presente, para los widgets sociales, o para menús de ayuda.

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

        .caja1 {
            height: 100px;
            width: 150px;
            background-color: darkorange;
            float: left;
        }

        .caja2 {
            height: 100px;
            position: absolute;
            left: 250px;
            top: 40px;
            background-color: aquamarine;
        }

        .caja3 {
            height: 100px;
            width: 150px;
            background-color: blueviolet;
            float: left;
        }

        .caja4 {
            height: 50px;
            width: 100%;
            background-color: #94baed;
            position: fixed;
            bottom: 0;
            left: 0;
        }
    </style>

</head>

<body>
    <div class="caja1">Caja 1 con float</div>
    <div class="caja2">Caja 2 con posición absoluta</div>
    <div class="caja3">Caja 3 con float</div>
    <div class="caja4">Caja 4 con fixed</div>
</body>
```



7 Relación entre *display*, *float* y *position*.

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

- Si `display` vale `none`, se ignoran las propiedades `float` y `position` y la caja no se muestra en la página.

- Si `position` vale `absolute` o `fixed`, la caja se posiciona de forma absoluta, se considera que `float` vale `none` y la propiedad `display` vale `block` tanto para los elementos en línea como para los elementos de bloque. La posición de la caja se determina mediante el valor de las propiedades `top`, `right`, `bottom` y `left`.

- En cualquier otro caso, si `float` tiene un valor distinto de `none`, la caja se posiciona de forma flotante y la propiedad `display` vale `block` tanto para los elementos en línea como para los elementos de bloque.