

ÍNDICE

7. POSICIONAMIENTO	2
7.1 INTRODUCCIÓN	2
7.2 POSICIONAMIENTO “PEGAJOSO” O STICKY	2
ACTIVIDAD 12. POSICIONAMIENTO STICKY	2
7.3 POSICIONAMIENTO FLOTANTE	3
▪ FLOAT.....	3
▪ CLEAR	6
ACTIVIDAD 13. POSICIONAMIENTO FLOTANTE	7
7.4 DISEÑO LÍQUIDO.....	7
ACTIVIDAD 14. DISEÑO LÍQUIDO	8
8. DISEÑO WEB RESPONSIVO ADAPTATIVO	9
8.1 DESCRIPCIÓN	9
8.2 MEDIA QUERY	9
ACTIVIDAD 15. MEDIA QUERY	10

7. Posicionamiento

7.1 Introducción

Las cajas CSS pueden colocarse en distintas posiciones en la pantalla. Existen varios tipos de posicionamiento distintos: estático, relativo, absoluto, fijo, flotante, flexible... Para configurar el posicionamiento se usan propiedades CSS como: `position`, `top`, `bottom`, `right`, `left`, `float` y `clear`.

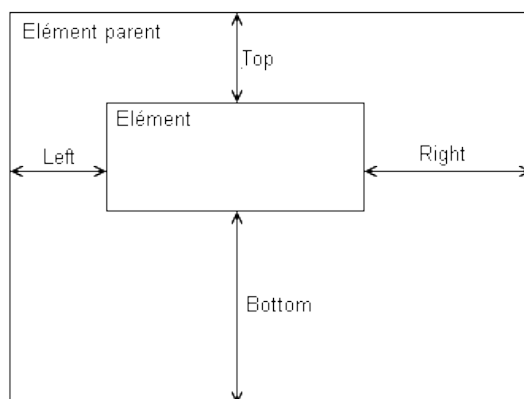
7.2 Posicionamiento “Pegajoso” o Sticky

El posicionamiento inicial del elemento es el habitual en la página. Sin embargo, al hacer scroll, el elemento queda fijo en la pantalla a la distancia del borde superior indicada por la propiedad `top`.

Se emplea especialmente para títulos.

Las propiedades CSS que se utilizan son:

```
position: sticky;      /* Se está utilizando un posicionamiento sticky */
top: ...px;            /* Distancia del elemento al borde superior */
```



- Descarga el fichero **pruebas_posicionamiento_sticky.html**. Ábrelo con un navegador web. Modifica el valor de la propiedad `top` y comprueba el resultado.

Algunos navegadores no soportan este valor de la posición. En ese caso se puede añadir `position: -webkit-sticky`; por compatibilidad.

Actividad 12. Posicionamiento sticky

12.1 Formato general

12.2 Posición de los encabezados

7.3 Posicionamiento flotante

▪ float

Cuando una caja se posiciona con el modelo de posicionamiento flotante, **automáticamente se convierte en una *caja flotante*, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.**

La propiedad que se emplea para utilizar el posicionamiento flotante es `float`. Posibles valores:

<code>float: left;</code>	la caja se desplaza a la izquierda
<code>float: right;</code>	la caja se desplaza a la derecha
<code>float: none;</code>	no hay desplazamiento flotante

Características relevantes:

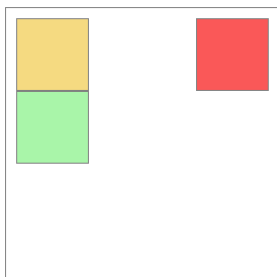
- ➔ Un elemento flotante se desplaza a izquierda o derecha, no arriba y abajo (sólo se desplaza verticalmente si no tiene espacio a los lados)
- ➔ Cuando un elemento se define como flotante “sale del flujo normal de la página”. Es decir, el resto de elementos no lo tienen en cuenta para definir su posición, es “invisible” para ellos. Sin embargo, eso no significa que no les afecte.
- ➔ Cuando hay varios elementos flotantes en la página, el resto de elementos “no los ven”, sin embargo, entre los elementos flotantes “sí se ven”. Es como si existiesen dos dimensiones: la de los elementos flotantes y la de los no flotantes.



- Descarga el fichero [pruebas_posicionamiento_flotante.html](#) y analiza su contenido.
Situación inicial:



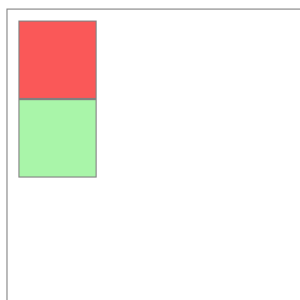
Añade a `#rojo` un posicionamiento flotante a la derecha (`float:right;`) y observa qué ocurre:



```
#rojo    {background-color: #FA5858;
          float:right;}
#ambar   {background-color: #F5DA81;}
#verde   {background-color: #A9F5A9;}
```

¿Qué ha ocurrido? La caja roja se desplaza de forma flotante a la derecha, por lo que se hace “invisible” para las otras dos. Por eso ámbar ocupa su lugar.

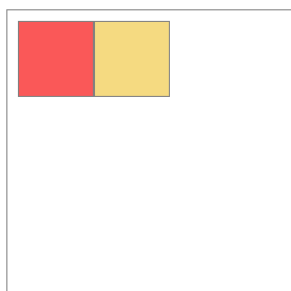
Ahora modifica el posicionamiento de #rojo a la izquierda (float:left;) y observa qué ocurre:



```
#rojo    {background-color: #FA5858;
          float:left;}
#ambar   {background-color: #F5DA81;}
#verde   {background-color: #A9F5A9;}
```

¿Qué ha ocurrido? La caja roja se desplaza de forma flotante a la izquierda, por lo que se hace “invisible” para las otras dos. Por eso ámbar ocupa su lugar y la caja roja queda superpuesta tapándola.

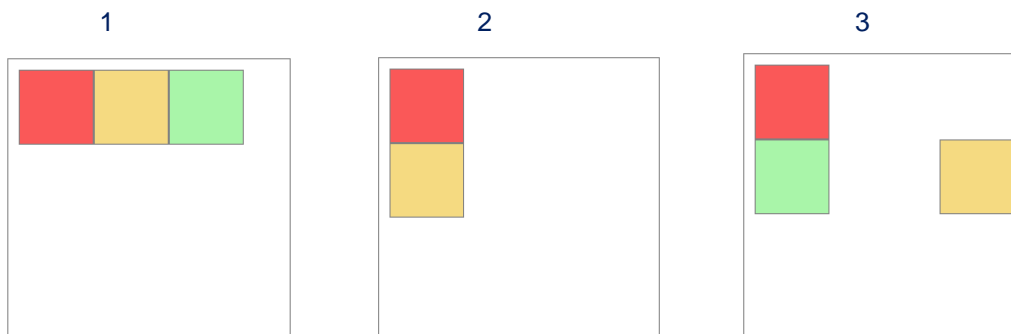
Ahora configura el posicionamiento de #rojo y de ámbar a la izquierda (float:left;) y observa qué ocurre:



```
#rojo    {background-color: #FA5858;
          float:left;}
#ambar   {background-color: #F5DA81;
          float:left;}
#verde   {background-color: #A9F5A9;}
```

¿Qué ha ocurrido? Las cajas roja y ámbar se desplazan de forma flotante a la izquierda, por lo que se hacen “invisibles” para la verde, que se sitúa debajo. Entre las cajas flotantes sí se ven, por lo que respetan su sitio. La roja queda más a la derecha porque es la primera.

Configura el posicionamiento flotante para obtener los siguientes resultados:



¿Qué ocurre cuando no hay espacio suficiente para los elementos flotantes?

- Utiliza de nuevo el fichero **pruebas_posicionamiento_flotante.html** pero modifica el ancho del <div> para que pase de 250 a 200px (width:200px;) Situación inicial:



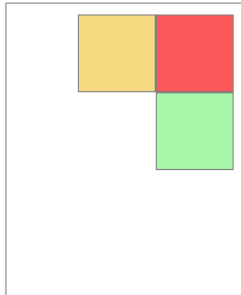
Añade a #rojo y #ambar un posicionamiento flotante a la derecha (float:right;) y observa qué ocurre:



```
#rojo    {background-color: #FA5858;
float:right;}
#ambar   {background-color: #F5DA81;
float:right;}
#verde    {background-color: #A9F5A9;}
```

¿Qué ha ocurrido? Las cajas roja y ámbar se desplazan a la derecha. La verde se sitúa en la primera posición, ya que “deja de ver” a las otras y queda parcialmente cubierta por la caja ámbar.

Ahora convierte las tres cajas en flotantes a la derecha (`float:right;`) y observa qué ocurre:



```
#rojo    {background-color: #FA5858;
          float:right;}
#ambar   {background-color: #F5DA81;
          float:right;}
#verde   {background-color: #A9F5A9;
          float:right;}
```

¿Qué ha ocurrido? Las tres cajas se desplazan a la derecha. Como no hay sitio suficiente, la verde hace un salto de línea y se sitúa a la derecha, pero abajo.

■ clear

La propiedad `float` hace que los elementos “salgan del flujo normal de la página”. De esta forma se crean dos dimensiones paralelas: la de los elementos flotantes y la de los elementos no flotantes.

En ocasiones, se necesita que un elemento no flotante sea consciente de los elementos flotantes. Con la propiedad `clear`, un elemento no flotante puede “ver” en la dimensión de los elementos flotantes, sin ser uno de ellos.

Esta propiedad establece que un elemento se sitúa siempre por debajo de otro que esté definido como flotante.

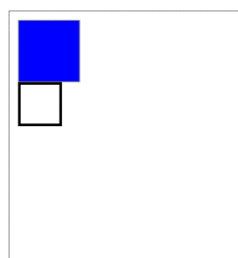


Posibles valores:

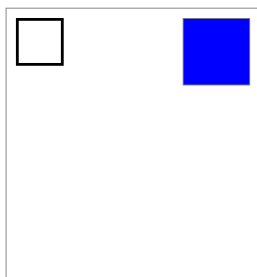
<code>clear: left;</code>	se sitúa por debajo de un elemento que es flotante a la izquierda
<code>clear: right;</code>	se sitúa por debajo de un elemento que es flotante a la derecha
<code>clear: both;</code>	se sitúa por debajo de un elemento flotante de cualquier tipo

- Descarga el fichero **pruebas_posicionamiento_flotante_clear.html** y analiza su contenido.

Situación inicial:



Añade a #azul un posicionamiento flotante a la derecha (float:right;). Observa qué ocurre:



```
#azul {background-color: blue;
float:right;}

#blanca {background-color: white;
width:25px;
height:25px;
border: 3px solid black;}
```

¿Qué ha ocurrido? La caja azul se hace flotante, desplazándose a la derecha.

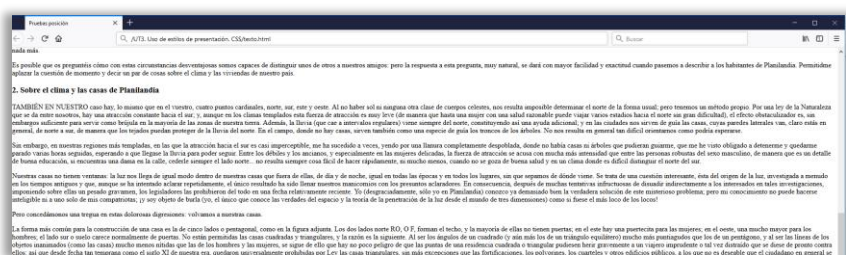
Ahora añade a #blanca la propiedad clear y prueba los tres valores posibles (left, right y both). Observa y analiza qué ocurre.

Actividad 13. Posicionamiento flotante

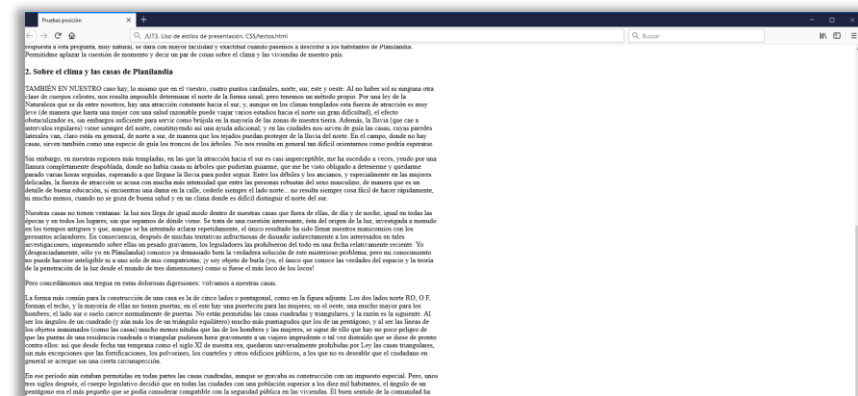
7.4 Diseño líquido

Las pantallas en que se visualizan páginas web pueden tener diferentes tamaños y resoluciones. Pueden darse situaciones indeseables como:

La pantalla es muy alargada y es incómodo leer el texto de lado a lado:



El texto se sitúa en la parte izquierda dando sensación de vacío:

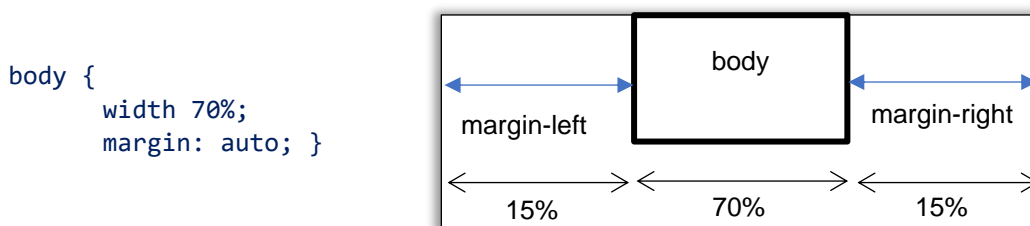


Normalmente, para dar un formato más estético a una página web, se centra el texto mediante un diseño llamado **líquido, fluido o dinámico**.

Para ello, todo el contenido que se desea centrar se inserta en un contenedor como, por ejemplo: `<div id="contenedor">` al que se le asignan dos propiedades. Si se quiere aplicar a todo el documento, pueden asignarse directamente al elemento `<body>`. Las propiedades son:

```
body {
  width: 70%;
  margin: auto;
}
```

De esta forma el texto ocupará el % de la pantalla especificado, y estará siempre centrado. Por ejemplo:



El valor auto significa que el navegador calculará el valor más óptimo:

- Asignará a margin-top y margin-bottom el valor 0
- Asignará a margin-left y margin-right los valores necesarios para que el contenido sea el 70%

Al hacer más o menos ancha la ventana, se respetará el % del contenido, quedando siempre centrado



Vista optimizada usando un diseño líquido

Actividad 14. Diseño líquido

14.1 Emplear diseño líquido

14.2 Asignar tamaño mínimo

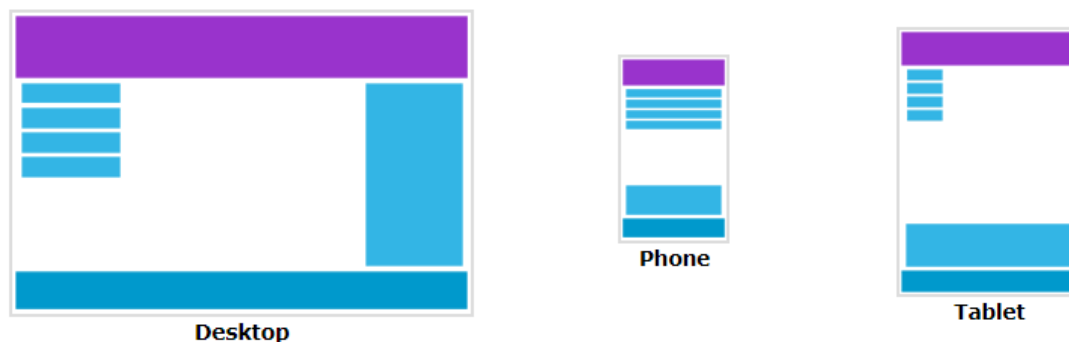
Ejercicios: Ejercicio 2

8. Diseño web responsivo adaptativo

8.1 Descripción

El diseño “web responsive” o adaptativo es una técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles.

Se trata de redimensionar y colocar los elementos de la web de forma que se adapten al ancho de cada dispositivo permitiendo una correcta visualización y una mejor experiencia de usuario. Se caracteriza porque los layouts (contenidos) e imágenes son fluidos y se usa código media-queries de CSS3.



8.2 Media query

Media query es una técnica disponible en CSS que permite hacer un **diseño web adaptativo**. Introduce reglas @media (breakpoints) que **permite aplicar distintas reglas CSS, dependiendo de las características del dispositivo de visualización**, entre ellas:

- Tipo de dispositivo
- Ancho y alto de la ventana o del dispositivo
- Relación de aspecto
- Orientación
- Resolución
- Número máximo de colores

La sintaxis básica es:

```
@media [not|only] mediatype and (mediafeatures) {
  CSS-Code;
}
```

Tipo de dispositivo genérico:

- all
- print (impresoras)
- screen (ordenador, tableta, móvil...)
- speech (lectura con screenreader)

Características (pueden combinarse varias empleando operadores lógicos):

- orientation (landscape /portrait)
- width
- resolution
- aspecto ratio

(ver [listado completo](#))

Ejemplo:

```
@media screen and (min-width: 400px) {  
  body { background-color: lightgreen; }  
  p { color: blue; }  
}
```

Para cambiar el formato de presentación en función del dispositivo (por ejemplo, que se visualice distinto en un móvil o en un ordenador) habrá que jugar con las distintas características.

Nota: cuando utilizamos la etiqueta html <picture> para mostrar una imagen, como ya hemos visto, puede utilizarse el atributo “media” en <source>. Los posibles valores en este caso difieren de las media queries y se pueden consultar en la documentación de [media](#).

Actividad 15. Media query

- 15.1 Colores en función de tamaño
- 15.2 Elementos ocultos en función del tamaño

Ejercicios: Ejercicio 3