

Manual de:

CSS



cipsa.net

DISTRIBUIDO POR: CENTRO DE INFORMÁTICA PROFESIONAL S.L.

**RESERVADOS TODOS LOS DERECHOS. QUEDA PROHIBIDO TODO TIPO DE REPRODUCCIÓN
TOTAL O PARCIAL DE ESTE MANUAL SIN PREVIO CONSENTIMIENTO POR EL ESCRITOR DEL
EDITOR**

MANUAL de CSS3 v. abril de 2018

DISTRIBUIDO POR: CENTRO DE INFORMÁTICA PROFESIONAL S.L.

**RESERVADOS TODOS LOS DERECHOS. QUEDA PROHIBIDO TODO TIPO DE REPRODUCCIÓN
TOTAL O PARCIAL DE ESTE MANUAL SIN PREVIO CONSENTIMIENTO POR EL ESCRITOR DEL
EDITOR**

CIPSA.NET

Contenido

| | | |
|-----|---|----|
| 1 | INTRODUCCIÓN: Presentación de CSS | 4 |
| 2 | SINTAXIS..... | 9 |
| 2.1 | Cómo declarar los estilos | 9 |
| 2.2 | Selectores..... | 10 |
| 2.3 | Unidades | 15 |
| 2.4 | Ejercicio 1 | 16 |
| 3 | FUENTES | 19 |
| 3.1 | COLORES | 21 |
| 3.2 | Fondo | 22 |
| 3.3 | Ejercicio 2 | 24 |
| 4 | TEXTO | 26 |
| 4.1 | Ejercicio 3 | 29 |
| 5 | LISTAS y TABLAS | 30 |
| 5.1 | Listas | 30 |
| 5.2 | TABLAS | 35 |
| 5.3 | Ejercicio 4 | 37 |
| 5.4 | Ejercicio 5 | 37 |
| 6 | MODELO DE CAJAS | 39 |
| 7 | MAQUETACION | 43 |
| 7.1 | POSITION..... | 45 |
| 7.2 | FLOAT | 48 |
| 7.3 | CLEAR..... | 49 |
| 8 | FINAL..... | 53 |

1 INTRODUCCIÓN: Presentación de CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas Web complejas.

Las CSS o *Cascading Style Sheets* permiten solucionar en gran medida los problemas que da el HTML a la hora de maquetar correctamente o al intentar obtener un determinado formato para nuestras páginas.

La separación de contenidos de su presentación con CSS ha presentado varias ventajas: mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento, permite visualizar el mismo contenido en diferentes dispositivos.

Tenemos varias posibilidades para definir un estilo: especificarlo directamente en la etiqueta en la que queremos usarlo, definirlo aparte y aplicarlo en las etiquetas que queramos, o definir estilos globales para las etiquetas (que podrán ser cambiados en las que no se desee aplicarlos).

FUNCIONAMIENTO BÁSICO DE CSS

Para aplicar un estilo a una etiqueta concreta

Sintaxis:

```
<etiqueta style="propiedad1:valor;...;propiedadN:valor;"> ...
```

```
</etiqueta>
```

Etiqueta es la etiqueta de HTML en la que queremos dar una apariencia concreta (<p>, <h3>, , ...). *STYLE* es el parámetro que indica que vamos a aplicar el estilo definido a continuación a la etiqueta en la que se encuentra. La definición del estilo se representa mediante pareja/s "propiedad:valor" separados por punto y coma.

Propiedad será la característica de la etiqueta que queramos modificar (el color, el tamaño de la fuente, el tipo de letra, ...) y valor será el valor que queremos darle (color negro, 8 puntos de tamaño de letra, ...).

Por ejemplo, si tenemos un texto en negrita y queremos que salga con un tamaño de letra 14 y en color rojo, haremos:

La negrita que vemos `<b style="font-size:14pt;color:red;">`es más grande y está en rojo``.

Insertar comentarios:

Es conveniente añadir comentarios a las CSS para recordar qué hacen los estilos que estamos creando, para posible consultas posteriormente; ya sea para modificaciones o borrado de estilos.

Para agregar estos comentarios en la hoja de estilos comenzamos escribiendo `/*`, escribimos el comentario y lo terminamos escribiendo `*/`.

Ejemplo: `/* Los textos con letra Arial */`

Para aplicar un estilo Global utilizamos la etiqueta `<style> ...</style>`. Dentro de esta etiqueta escribimos los estilos que utilizaremos en todo el código (Bloques de estilos). En un bloque de estilo global podremos definir cuantos estilos queramos, separados por ";" y todo agrupados dentro de {}.

Sintaxis:

```
<style>
/*
etiqueta1, etiqueta2 {
    propiedad1 : valor;
}
etiqueta3 {
    propiedad1 : valor;
    ...;
    propiedadS:valor
}
.clase1 : {
    propiedad1 : valor;
    ...;
```

```
    propiedadT:valor} */  
</style>
```

NOTA: Los símbolos "/*" y "*/" indican que lo escrito entre ellos es un comentario, explicado anteriormente.

Antes de la utilización de CSS, los diseñadores de página HTML debían definir el estilo y el aspecto de los elementos HTML en el propio documento.

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Ejemplo de estilos con CSS</title>  
<meta charset="utf-8">  
<style>  
h1 {  
    color:red;  
    font-family:Arial;  
    font-size:large;  
}  
p {  
    color:grey;  
    font-family:Verdana;  
    font-size:medium;  
} </style>  
</head>  
<body>  
<h1>Titular de la página</h1>  
<p>Un párrafo de texto no muy largo</p>  
</body>  
</html>
```

En este ejemplo básico, podemos comprobar que la estructura utilizada es la misma que (X)HTML, pero la etiqueta utilizada para incluir CSS en (X)HTML es **<style>**. En ella hemos creado dos estilos que utilizaremos para el texto escrito de la etiqueta **<h1>** y el texto incluido en la etiqueta **<p>**.

Para el texto de la etiqueta **<h1>** escrito dentro de **<body>**, "Titular de la página"

le hemos asignado un color rojo, un tipo de fuente Arial y un tamaño grande. Para el texto de la etiqueta

`<p>` “Un párrafo de texto no muy largo” le hemos indicado un estilo con un color de letra gris, tipo de fuente Verdana y un tamaño de letra media.

NOTA: Para comprobar e ir manejando las CSS te recomendamos que realices las prácticas. Recuerda utilizar un editor de Texto (Bloc de notas, Sublime y guardar el documento con extensión HTML.

De todas las formas de aplicar CSS a los documentos (X)HTML, esta es la más utilizada. También podemos recurrir a la opción de incluir estilos externos para ello se utiliza la etiqueta `<link>`. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas (X)HTML, por lo que se garantiza la aplicación de los mismos estilos a todos los documentos que forman un sitio web.

Sintaxis:`<link rel="stylesheet" type="text/css" href="URL_Hoja.css">`

La principal razón por la que es el método más utilizado es que es el más sencillo de mantener, ya que un solo cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todos los documentos (X)HTML que enlazan ese archivo. Cuando utilizemos hojas de estilos externas, antes que hay que crearlas y tener en cuenta que debemos guardarlas con extensión .css, para ello podemos utilizar cualquier programa editor de texto.

Otra posibilidad para utilizar una hoja de estilos es exportarla, para importar una hoja estilo externa, dentro del elemento **style**, pero antes de cualquier regla de estilo individual, escriba **@import “externa.css”**, donde externa.css es el nombre de su hoja de estilo CSS.

También la regla **@import** se puede escribir como **@importurl(externa.css);** o **@importurl(“externa.css”);**. sitúela siempre delante de cualquier otra regla en el elemento **style** y no olvidar poner el punto y coma.

Ejemplo:

```
<html>
<head>
<title>ejemplo de importación de CSS</title>
<style type="text/css">
```

```
@import "hoja_estilos_externa.css";  
</style>  
</head>  
<body>  
    Ejemplo  
</body>  
</html>
```


2 SINTAXIS

2.1 Cómo declarar los estilos

Con tantas formas de aplicar estilos, es normal que se aplique más de una regla de estilo al mismo elemento. La regla básica es, con todo lo demás igual, cuanto más al final aparezca un estilo, más importancia tendrá.

Por lo tanto, los estilos aplicados localmente tienen mayor preferencia y anularán cualquier estilo con conflicto aplicado con anterioridad.

En un elemento `style`, cualquier regla **@import** presente saldrá pendiente frente a cualquier regla de estilo individual que también aparezca en el elemento `style`.

Las reglas de hoja de estilos importada tienen prioridad sobre cualquier regla que venga delante de la regla **@import**.

Ejemplo:

```
<head>
  <title>Ejemplo de preferencia de estilos</title>
  <link rel="stylesheet" type="text/css" href="sitio.css">
  <style>img {border-style:dashed} </style>
</head>
```

El elemento **<style>** aparece el último, por lo tanto, tendrá preferencia sobre las reglas en la hoja de estilos `sitio .css`.

2.2 Selectores

El selector determina los elementos sobre los que se aplica una regla de estilo, es *a quién* le decimos qué hacer. Por ejemplo, si desea cambiar la fuente de los elementos `<p>` a un tipo de fuente Courier, 10 píxeles, necesitamos crear un selector que identifique sólo los elementos `<p>`.

CSS nos permite modificar el aspecto de **id's, clases o etiquetas de HTML**. El nombre de la regla determinará que es lo que vamos a modificar con respecto a la visualización por defecto del navegador. Así, tendremos los siguientes casos:

- Si queremos dar estilo a un identificador único (ID) colocaremos una **almohadilla (#)** y a continuación el nombre del ID.
- Si queremos dar estilo a una clase (class) colocaremos un **punto (.)** y a continuación el nombre de la clase. **ES LO MÁS RECOMENDADO.**
- y si lo que queremos es modificar una etiqueta (por defecto ya tendrán un aspecto) simplemente colocamos el nombre de la etiqueta.

Un selector puede definir hasta cinco criterios diferentes para elegir los elementos que se deberían cambiar:

1. **Selector de tipo o etiqueta:** El tipo más sencillo de selector es

simplemente el nombre del tipo de elemento y la propiedad. Ejemplo: `h1`

```
{color:red;}
```

En este ejemplo los elementos `h1` cambiar a color rojo.

2. **Cambiar un elemento dentro de otro elemento:**

Ejemplo:

```
h1 em {color:red;}
```

Cambiar el elemento `em` dentro de `h1` a color rojo.

3. **Selector de Clase:**

```
.cambio {color:red;}
```

Cambiará cualquier elemento cuyo atributo class sea igual a “cambio”. Se utiliza para dar unos valores determinados a diferentes elementos.

Ejemplo:

```
<p>Primer texto </p>
<p class="cambio">Segundo texto</p>
<p>tercer texto </p>
```

La segunda línea del código es la que obedecería al Selector CSS, el color del texto sería negro (predeterminado) y con la etiqueta class lo cambiaríamos a rojo, manteniéndose en negro los textos dentro de las otras etiquetas <p>.

También podemos restringir el alcance de los selectores, aplicando el selector solamente a los elementos de un tipo.

Ejemplo:

```
p.mensaje {
  color: #9907AA;
  border: 1px
solid;
  background: #AAFF00;
}
```

Únicamente se aplicará el selector a los elementos de tipo <p> con un atributo class de nombre “mensaje”.

4. Selector ID:

Se utiliza para aplicar un estilo a un solo elemento específico. Sólo se puede declarar una vez en toda la página, por lo tanto, NO ES REUTILIZABLE y es difícil de mantener.

```
#identificador {color:red;}
```

Cambiará cualquier elemento cuyo atributo ID sea igual a “identificador”.

Ejemplo:

```
<h1>Ejemplo de Identificador </h1>
<div id="identificador">
    <p>Texto</p>
</div>
```

**Diferencia entre estilos de clase y estilos de ID:*

- En una página el valor ID debe ser único, dos elementos diferentes no pueden tener el mismo valor ID.
- El atributo class puede ser compartido por elementos diferentes.

**Cuidado con:*

- **p#estilo** : quiere decir , todos los elementos de tipo "p" con atributo "id"= estilo.
- **p #estilo**: quiere decir, todos los elementos con atributo "id"= estilo que estén dentro de cualquier elemento de tipo "p".
- **p, #estilo**: quiere decir, todos los elementos de tipo "p" y todos los elementos con atributo "id"=estilo.

5. Selector adyacente:

Funciona cambiando los elementos adyacentes a otro elemento. Utilizamos la jerarquía padrehijo*.

Ejemplo:

h1 + h2 {-----}

```
<body>
<h1>Texto 1</h1>
<h2>Texto
2</h2> .....
<h2>Texto3</h2>
```

```
.....  
</body>
```

Selecciona todos los tipos h2 cuyo elemento padre sea el mismo que el de h1 (body es el elemento padre) y además h2 tiene que estar inmediatamente después que h1.

Es decir, en este ejemplo, los estilos h1+h2 se aplican al primer elemento h2 de la página, pero no al segundo h2

**Ejemplo: <p>span (donde p es padre y span hijo). El elemento hijo tiene que estar inmediatamente después del padre en el estilo.*

SELECCIONAR PARTE DE UN ELEMENTO

Puede seleccionar sólo la primera letra o primera línea de un elemento y luego aplicarle formato.

Ejemplo:

```
p:first-line  
{color: red;}
```

La primera línea de la etiqueta <p> la mostrará de color rojo.

SELECCIONAR LA PRIMERA LETRA DE UN ELEMENTO

El selector elegirá sólo la primera letra de cada elemento.

Ejemplo:

```
p:first-letter  
{color: red;}
```

La primera letra de la etiqueta <p> la mostrará de color rojo.

DAR FORMATO A ELEMENTOS DE VÍNCULO SEGÚN SU ESTADO

Los enlaces pueden tener diferentes estados, llamados PSEUDOCASES, según haya sido visitado o no este enlace. Los estados pueden ser los siguientes:

- **Link** : los vínculos aún no visitados
- **Visited**: vínculos ya visitados
- **Focus**: Vínculo seleccionado por medio del teclado y está a punto de ser activado.
- **Hover**: cambia la apariencia cuando se señala el vínculo.
- **Active**: cuando se hace clic sobre el vínculo.

Seguramente ya los conocíamos de X(HTML).

Por ello podemos utilizar estilo para cada uno de los estados de los vínculos.

Ejemplo:

```
a:link {color: red;}
a:visited {color: orange;}
a:focus {color: purple;}
a:hover {color: green;}
a:active {color:blue;}
```

Los estilos para vínculos se deberían definir siempre en este orden para evitar que las propiedades se anulen cuando un vínculo esté en más de un estado.

GRUPOS DE ELEMENTOS

A veces es necesario aplicar las mismas reglas de estilo a más de un elemento. Puede repetir las reglas para cada elemento, o puede combinar selectores y aplicar las reglas de un solo golpe.

Para ellos debemos especificar los nombres de los elementos, uno detrás de otro, separados por una coma.

Ejemplo:

```
h1, h2 {color: blue;}
```

En este ejemplo la propiedad de color se aplicará a los dos elementos h1 y h2.

2.3 Unidades

Normalmente los valores de las propiedades serán cadenas de texto. Pero en los casos en los que nos refiramos a cantidades o medidas, tenemos que tener en cuenta que hay varias formas de hacerlo. Las clasificaremos en relativas y absolutas. Pero comodidad y compatibilidad usaremos sólo las relativas. Las siguientes unidades relativas están disponibles:

- **em** (ems, la altura de la fuente de los elementos)
- **ex** (x-height, altura de la letra "x")
- **px** (pixels, relativa a la resolución del lienzo)

Absolutas

- **%** (porcentaje, relativa generalmente al tamaño de fuente. No llevan espacios)
- **In** Pulgada.
- **cm** Centímetros
- **pt** Puntos
- **mm** Milímetros
- **pc** Picas

Para entender mejor las opciones de medida, comentar que **em** suele equivaler al tamaño de fuente del elemento, por ejemplo 2em significa "el doble del tamaño de fuente". La unidad **ex** sería igual a la altura x de la fuente, es decir, a la altura de la letra x de la fuente, pero no tiene buen soporte. Los píxeles (**px**) son relativos a la resolución del monitor (pero no a otras reglas de estilo). La mayoría de los monitores de hoy en día muestran unos 80 píxeles por pulgada (aunque varían entre 72 y 96 ppp), así que 16 píxeles es, aproximadamente, 1/5 de pulgada o 0,65 cm.

2.4 . Ejercicio 1 .

Abre el archivo ejercicio01.html con tu editor.

A partir del código HTML y CSS que se muestra, añadir los selectores CSS que faltan para aplicar los estilos deseados. Cada regla CSS incluye un comentario en el que se explica los elementos a los que debe aplicarse:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <title>Ejercicio de selectores</title>

    <style type="text/css">

      /* Todos los elementos de la pagina */

      { font: 1em/1.3 Arial, Helvetica, sans-serif; }

      /* Todos los parrafos de la pagina */

      { color: #555; }

      /* Todos los párrafos contenidos en #primero */

      { color: #336699; }

      /* Todos los enlaces de la pagina */

      { color: #CC3300; }

      /* Los elementos "em" contenidos en #primero */

      { background: #FFFFCC; padding: .1em; }

      /* Todos los elementos "em" de clase "especial" en toda la
pagina */

      { background: #FFCC99; border: 1px solid #FF9900; padding: .1em;
}

      /* Elementos "span" contenidos en .normal */
```



```
{ font-weight: bold; }

</style>

</head>

<body>

    <div id="primero">

        <p>Lorem ipsum dolor sit amet, <a href="#">consectetuer
adipiscing elit</a>. Praesent blandit nibh at felis. Sed nec diam in
dolor vestibulum aliquet. Duis                ullamcorper, nisi non
facilisis molestie, <em>lorem sem aliquam nulla</em>, id
    lacinia velit mi vestibulum enim.

        </p>

    </div>

    <div class="normal">

        <p>

            Phasellus eu velit sed lorem sodales egestas. Ut feugiat.
<span>    <a href="#">Donec porttitor</a>, magna eu varius luctus,</span>
metus massa tristique massa, in imperdiet est velit vel magna. Phasellus
erat. Duis risus.<a href="#">Maecenas dictum</a>, nibh vitae pellentesque
auctor, tellus velit consectetur tellus, tempor pretium felis tellus at
metus.

            </p>

            <p>

                Cum sociis natoque <em class="especial">

penatibus et magnis</em> dis parturient montes, nascetur
ridiculus mus. Proin aliquam convallis ante.Pellentesque
```

habitant morbi tristique senectus et netus et malesuada fames
ac turpis egestas. Nunc aliquet. Sed eu metus. Duis justo.

</p>

<p>

Donec facilisis blandit velit. Vestibulum nisi.
Proin volutpat, <em class="especial">enim id iaculis
congue, orci justo ultrices tortor, quis lacinia eros libero in eros. Sed
malesuada dui vel quam.

Integer at eros.

</p>

</div>

</body>

</html>

3 FUENTES

En este apartado veremos las diferentes propiedades que podemos añadir a nuestros estilos.

Estas propiedades harán que podamos variar más en nuestros estilos y realizar variados cambios. **Valores para las propiedades**

Piensa que cuando abres un documento de Word, por defecto, el tamaño de fuente que aparece es de 12pt. Estos 12pt equivalen a 16px.

Esto es un estándar, una predeterminación de los ordenadores y los navegadores.

Este es el motivo por el que se igualan estas medidas 12pt a 16px y a su vez al 100% y a 1em. Estos son los valores base para nuestro trabajo.

Observa esta tabla, estos son los valores predeterminados en los que se basan los navegadores:

| Puntos | Píxeles | Em's | % |
|--------|---------|---------|--------|
| 6pt | 8px | 0.5em | 50% |
| 7pt | 9px | 0.55em | 55% |
| 7.5pt | 10px | 0.625em | 62.5% |
| 8pt | 11px | 0.7em | 70% |
| 9pt | 12px | 0.75em | 75% |
| 10pt | 13px | 0.8em | 80% |
| 10.5pt | 14px | 0.875em | 87.5% |
| 11pt | 15px | 0.95em | 95% |
| 12pt | 16px | 1em | 100% |
| 13pt | 17px | 1.05em | 105% |
| 13.5pt | 18px | 1.125em | 112.5% |
| 14pt | 19px | 1.2em | 120% |
| 14.5pt | 20px | 1.25em | 125% |
| 15pt | 21px | 1.3em | 130% |
| 16pt | 22px | 1.4em | 140% |
| 17pt | 23px | 1.45em | 145% |
| 18pt | 24px | 1.5em | 150% |
| 20pt | 26px | 1.6em | 160% |
| 22pt | 29px | 1.8em | 180% |
| 24pt | 32px | 2em | 200% |
| 26pt | 35px | 2.2em | 220% |
| 27pt | 36px | 2.25em | 225% |
| 28pt | 37px | 2.3em | 230% |
| 29pt | 38px | 2.35em | 235% |
| 30pt | 40px | 2.45em | 245% |
| 32pt | 42px | 2.55em | 255% |

| | | | |
|------|------|--------|------|
| 34pt | 45px | 2.75em | 275% |
| 36pt | 48px | 3em | 300% |

¿Qué haría si quiero modificar ese valor base? Supongamos que quiero que la fuente de mi proyecto web sea más pequeña que 16px, 10px por ejemplo.

Si sigo las equivalencias de la tabla, vemos que sería 0.625em o 62.5%. Si establezco en el estilo para el body lo siguiente:

```
body{font-size:0.652em}
```

```
/*a partir de este momento todo el contenido de texto del documento se verá a  
10px*/
```

Si a continuación digo:

```
p{ font-size:1em}
```

```
/*el párrafo se verá a un tamaño de 10px*/
```

¿Por qué ocurre esto? Si en la tabla de equivalencias hemos visto que 1em son 16px. Pero la

<p> está dentro de <body>, por lo tanto, el tamaño de fuente de <p> depende del de <body>, es decir, para <p> el tamaño de fuente de <body> es 1em, es la base.

Otro ejemplo:

```
h1{ font-size:2em}
```

/*el tamaño de fuente del encabezado será el doble del de párrafo*/

3.1 COLORES

También tendremos que hacer referencia a los colores. Para ello, tenemos varias formas:

- **#rrggbb** (por ej., #00cc00)
- **#rgb** (por ej., #0c0)
- **rgb(x,x,x)** donde x es un entero entre 0 y 255 inclusive (por ej., rgb(0,204,0))

Las combinaciones que pueden conseguir y su explicación ya lo vimos en el manual de HTML. Dales un repaso.

También podemos utilizar una de las opciones más sencillas, que es utilizar el nombre de los 16 colores predefinidos. En la tabla de ejemplo se incluye su código Hexadecimal. Cuando necesitemos alguno de estos colores podemos utilizar su nombre para indicar el color a utilizar sin código Hexadecimal.

| Nombre | Código | Nombre | Código | Nombre | Código |
|--------|---------|--------|---------|--------|---------|
| Aqua | #00ffff | Black | #000000 | Blue | #0000ff |
| Gray | #808080 | Green | #008000 | Lime | #00ff00 |
| Maroon | #800000 | Navy | #000080 | Olive | #808000 |
| Purple | #800080 | Red | #ff0000 | Silver | #c0c0c0 |
| Teal | #008080 | White | #ffffff | Yellow | #ffff00 |

NOTA: Dispones de una tabla completa de colores con sus códigos hexadecimales en:

https://www.w3schools.com/colors/colors_names.asp

Para identificar mejor las propiedades de cada objeto las dividimos en:

- bloques • Texto
- Fuentes

- Color
- Listas.

3.2 Fondo

Esta tabla contiene diferentes propiedades dirigidas a los fondos:

| PROPIEDAD | NOMBRE | DESCRIPCIÓN |
|-----------------------|--------------------------------------|---|
| Background-color | Color de fondo | Establece un color de fondo para los elementos |
| Background-image | Imagen de fondo | Establece una imagen como fondo para los elementos. |
| Background-repeat | Repetición de la imagen de fondo | Controla la forma en la que se repiten las imágenes de fondo. |
| Background-position | Posición de la imagen de fondo | Controla la posición en la que se muestra la imagen en el fondo del elemento. |
| Background-attachment | Comportamiento de la imagen de fondo | Controla la forma en la que se visualiza la imagen de fondo, permanece fija cuando se hace scroll o en la ventana del navegador o se desplaza con la ventana. |
| Background | Fondo del elemento | Establece todas las propiedades del fondo de un elemento. |

Cuando utilizamos una imagen de fondo, podemos indicarle las veces que queremos que se repita dicha imagen. Se escribe **repeat** para disponer la imagen en mosaico tanto horizontal como verticalmente, **repeat-x** para que solamente sea horizontalmente, **repeat-y** para que sólo sea verticalmente, y **no-repeat** para que no se repita en mosaico.

Ejemplo:

```
body { background-image:
url(imagen.jpg) repeat-x; }
```

También se puede escribir así:

```
body {  
  background: url(imagen.jpg);  
  backgroundrepeat: repeat-x;  
}
```

La imagen colocada como fondo sólo se repetirá en el eje X.

Podemos indicar un fondo transparente:

```
background: transparent;
```

En la propiedad **background-attachment**, se puede dar los valores **fixed** o **scroll** para determinar si el fondo tiene que desplazarse con el lienzo.

3.3 . Ejercicio 2 .

Realiza un menú vertical con 5 elementos, los cuales mostrarán las siguientes características y comportamientos.

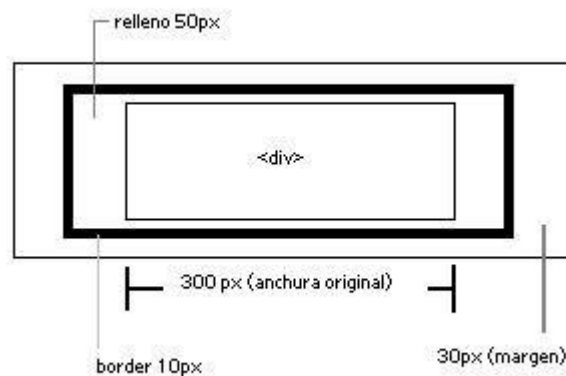
- ⌚ Los elementos tendrán un color de fondo azul.
- ⌚ Los enlaces de color Blanco, los enlaces visitados se muestran de color Rojo.
- ⌚ Cuando se pasa el ratón por encima de uno de los elementos, se debe mostrar un color de letra alternativo.

Nota: Los enlaces por ejemplo deben estar apuntando a las siguientes Webs :

<https://www.cipsa.net/> , www.cipsaempresas.net, www.maps.google.com,
<https://www.google.es/maps/place/CIPSA++Formaci%C3%B3+a+Barcelona> y
<https://www.google.es/>

En un elemento hay que determinar mas propiedades que la altura (height) y anchura (width), también hay que determinar el relleno, bordes, márgenes, etc...

En este dibujo podrás verlo mejor:



Otro elemento que podemos añadir a un elemento caja es el fondo. El fondo puede ser un color o una imagen. El fondo se visualizar únicamente en el área ocupada por el contenido y su relleno, ya que los bordes tienen su propiedad.

Para color de fondo se utiliza la propiedad **Background-color** y para añadir imagen **Backgroundimage**.

Ejemplos:

```
body {  
background-color:  
#FF0000;  
}
```

El fondo de la página web sería rojo. (Recuerda que utilizamos el método hexadecimal para indicar los colores).

```
body {  
background-  
image:  
url("imágenes/fondo_imagen.jpg") }
```

Cuando trabajamos con imágenes es recomendable crear una carpeta a parte para almacenarla, así nos resultará más fácil su localización.

En ocasiones, no necesitamos que la imagen se repita horizontal y verticalmente. Para que esto ocurra utilizamos la propiedad **background-repeat** que permite controlar la forma de repetición de las imágenes de fondo.

Esta propiedad tiene distintos valores:

- repeat (la imagen se repetirá en todas direcciones)
- repeat-x (la imagen se repite de forma horizontal)
- repeat-y (la imagen se repite de forma vertical)
- no-repeat (se repite una sola vez)

4 TEXTO

En este apartado veremos los estilos que podemos agregarle a los textos de nuestra página Web.

Ya hemos visto en los anteriores apartados como indicar las propiedades de cada elemento, así que directamente nos centraremos en las propiedades dirigidas a texto.

PROPIEDADES

- **Font-family**: los navegadores que tenemos en nuestro ordenador, cuando muestra las páginas Web, utilizan los tipos de letra ya instalados en el ordenador del usuario. Por ello si el diseñador puede indicar que muestra la fuente que indica en esta propiedad. Empieza por el primer tipo de letra, si no lo encuentra en el equipo, busca la segunda y si no la encuentra prueba con la tercera; En caso de no encontrar ninguna de las indicadas el navegador mostrará la más parecida que tenga a las indicadas.

Las más utilizadas son:

font-family: Arial, Helvetica, sans-serif; font-family: "Courier new", courier, monospace; font-family: "Times new roman", Times, serif; font-family: Georgia, "Times new roman", Times, serif; font-family: Verdana, Arial, Helvetica, sans-serif;

- **Font-size:** Tamaño de la letra. Al comienzo de este tema se explicó las diferentes formas de indicar unidades de medida.
- **Font-weight:** Peso de la fuente. Podemos dar un peso medio en negrita al texto. También podemos escribir **bolder** o **lighter** para utilizar un valor relativo al peso actual. O escribir un múltiplo de 100 entre 100 y 900, donde 400 representa un peso normal y 700 representa negrita.
- **Font-style:** Estilo de la fuente.
- **Font:** Permite indicar de forma directa todas las propiedades de la tipografía de texto.

Ejemplo:

```
font: bold 1em Verdana, Arial, Helvetica, sans-serif;
```

En este ejemplo hemos indicado que el formato de la letra será negrita con 1em de tamaño, y la familia de tipografías a utilizar.

```
font-weight: Bold;
```

El siguiente estilo también indica que la letra es en negrita, y hemos indicado una negrita normal, pero también se puede indicar como los siguientes ejemplos:

```
font-weight: 500;
```

Podemos indicar la cantidad de peso en negrita que tendrá la fuente, cómo hemos indicado anteriormente 400 indica un peso normal, así que nuestra letra será ligeramente negrita.

Para eliminar el formato negrita: `font-weight: normal;`

CIPSA.NET

4.1 . Ejercicio 3 .

Vamos a realizar el ejercicio correspondiente al tema de Texto, por supuesto iremos repasando opciones de anteriores temas.

Has de realizar este ejemplo.

CAJA 1

Esta es la primera caja, que le añadiremos una alineación izquierda, un interlineado de 1.2em y letra Arial con tamaño 2em.

CAJA 2

Esta es la segunda caja, que le añadiremos una alineación central, un interlineado de 1.6em y letra Courier con un tamaño de 1.5em.

CAJA 3

Esta es la tercera caja, que le añadiremos una alineación derecha, un interlineado de 2em, letra Verdana con un tamaño de 0.8em.

El texto de dentro de las cajas tiene las indicaciones para las propiedades del texto. Las opciones comunes son las siguientes:

Tamaño de las cajas: width: 200px; height: 450px;

Visualización display: inline; float: left;

margin: 1em;

5 LISTAS y TABLAS

5.1 Listas

En el manual de HTML nos explican que son las listas, pero por si se te ha olvidado, te lo volvemos a explicar.

A menudo nos interesa presentar las cosas en forma de lista.

Ejemplo:

- Una cosa
- Otra cosa
- Otra cosa más

Como ves en este ejemplo es la secuencia de elementos indicados con un símbolo. Las etiquetas utilizadas en HTML son `` para indicar que es una lista, `` indica cada uno de los elementos de esa lista. Para cerrar la lista ``.

También podemos incluir lista numeradas:

Ejemplo:

1. Una cosa
2. Otra cosa
3. Otra cosa más

En código HTML se utilizan la etiqueta `` para indicar el comienzo de la lista,

`` indica cada uno de los elementos de la lista, por último cerrar la lista ``

Después de un pequeño repaso a las listas, vamos a ver los estilo que le podemos agregar con CSS.

En las listas del primer tipo, por defecto aparece un círculo negro como indicación; En las lista numeradas aparecen los números enteros, esto estilos podemos cambiarlos.

List-style-type

Cambia el tipo de viñeta, puede contener diferentes valore a escoger, son los siguientes:

| Valores Gráficos | | Valores alfanuméricos | |
|------------------|--|------------------------------|--------------------------|
| None | Sin valor, muestra listas sin símbolos de viñeta | Decimal | Numeración normal |
| Disc | Círculo relleno | Decimal-leading-zero | Numeración: 01,02,03... |
| Circle | Círculo vacío | lower-roman | Números romanos |
| Square | Cuadrado relleno | Lower-greek | Alfabeto griego: α,β,... |
| | | Upper-latin | Alfabeto latin: A, B,.. |
| | | Decimal-leading-zero 01, 02, | |

También podemos utilizar como viñetas imágenes externas, ya sean de creación propia o de Internet. Para poder insertas estas imágenes utilizamos la propiedad: **liststyle-image**.

Veamos un ejemplo de la utilización de imagen externa en una lista:

```
ul {list-style-image: url(imagenes/boton.jpg);}
```

Asi se indicaría el estilo para insertar una imagen como viñeta. Ahora veremos un ejemplo completo sobre esto, escribe el código y pruébalo. Puedes escoger una imagen de Internet.

```
<html>
<head>
<title>Ejemplo de estilos de viñetas</title>
<style
type="text/css"
> ul {
margin:0;
padding.left:
```

```
1.5em;
lineheight:
1.5em;}
ul.viñeta{list-style-image: url(imagenes/boton.jpg);}

</style>
</body>

<div>
<ul class="viñeta">
    <li>Elemento_1</li>
    <li>Elemento_2</li>
    <li>Elemtnos_3</li>
</ul>
</div>
</body>
</html>
```

Las listas también se suelen emplear para la creación de menús de navegación verticales y horizontales. Veamos un ejemplo de lista con vínculos en HTML:

```
<ul>
<li><a ref="?" title="Enlace">Enlace_1</a></li>
<li><a ref="?" title="Enlace">Enlace_2</a></li>
<li><a ref="?" title="Enlace">Enlace_3</a></li>
<li><a ref="?"
title="Enlace">Enlace_4</a></li> </ul>
```

Para transformar una lista en un menú, deberíamos realizar los siguientes cambios:

- Indicar la anchura del menú
- Quitar la viñetas
- Añadir un borde inferior en cada uno de los elementos del menú (Opcional)
- Podemos añadir cada uno de los enlaces dentro de una celda de una tabla.

Prueba el siguiente código para ver un menú básico vertical:

```
<html>
<head>
<title>Menu vertical sencillo</title>
```



```

<style type="text/css">
ul.menu{width:180px; list-style:none; margin:0;
padding:0; border:1px solid #7c7c7c;} ul.menu
li{border-bottom:1px solid
#7c7c7c; border-top:1px
solid #fff;
background:#f4f4f4;}
ul.menu li a:link{padding:.2em 0 .2em .5em;
display:block; text-decoration:none; color:green;}
ul.menu li a:hover{padding:.2em 0 .2em .5em;
display:block; text-decoration:none; color:red;}
ul.menu li a:visited{padding:.2em 0 .2em .5em;
display:block; text-decoration:none; color:blue;}
</style>
</head>
<body>
<ul class="menu">
<li class="menu"><a href="#" title="Enlace
generico">Enlace
1</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace
2</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace 3</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace
4</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace
5</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace 6</a></li>
</ul>
</body>
</html>

```

Este código mostrará un Menú básico horizontal:

```

<html>
<head>
<title>Menu vertical sencillo</title>
<style type="text/css"> ul.menu{clear:both; float:left;
width:100%; list-style:none; margin:0; padding:0; border-
left:1px solid #7c7c7c;} ul.menu

```

```
li{display:inline;      float:left;}      ul.menu      li
a:link{padding:.3em;      display:block;      text-
decoration:none;      color:green;      background:#f4f4f4;
border-bottom:1px solid #9c9c9c; borderright:1px solid
#9c9c9c; border-top:1px solid #7c7c7c;} ul.menu li
a:hover{padding:.3em;      display:block;      text-
decoration:none; color:red; background:#f4f4f4; border-
bottom:1px solid #9c9c9c; border-right:1px solid
#9c9c9c; border-top:1px solid #7c7c7c;} ul.menu li
a:visited{padding:.3em;      display:block;
textdecoration:none; color:blue; background:#f4f4f4;
borderbottom:1px solid #9c9c9c; border-right:1px solid
#9c9c9c; border-top:1px solid #7c7c7c;}
</style>
</head>

<body>
<ul class="menu">
<li class="menu"><a href="#" title="Enlace
generico">Enlace
1</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace 2</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace
3</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace
4</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace 5</a></li>
<li class="menu"><a href="#" title="Enlace
generico">Enlace 6</a></li>
</ul>
</body>
</html>
```

5.2 TABLAS

Este es el aspecto de una tabla normal realizada en X(HTML) con el siguiente código:

```
<html>
<head>
<title>Ejercicio de Tablas</title>
</head>
<body>
<table border>
<tr>
<td>c-1 f-1</td>
<td>c-2 f-1</td>
<td>c-3 f-1</td>
</tr>
<tr>
<td>c-1 f-2</td>
<td>c-2 f-2</td>
<td>c-3 f-2</td>
</tr>
</table>
</body>
</html>
```

| | | |
|---------|---------|---------|
| c-1 f-1 | c-2 f-1 | c-3 f-1 |
| c-1 f-2 | c-2 f-2 | c-3 f-2 |

Ahora veremos las propiedades que podemos añadir a las tablas con los estilos CSS.

border-spacing: Espaciado entre border.

Ejemplo:

```
.norma
1 {
width:
150
px;
Border: 1px solid #000000;
border-spacing: 0;
}
```

Aplicando este estilo indicamos: la anchura de la tabla, el borde de la tabla y quitamos el espacio del border.

| | | |
|---------|---------|---------|
| c-1 f-1 | c-2 f-1 | c-3 f-1 |
| c-1 f-2 | c-2 f-2 | c-3 f-2 |

Así es como aparecería la tabla si la clase creada en el estilo la aplicamos en la etiqueta **<table>**

| | | |
|---------|---------|---------|
| c-1 f-1 | c-2 f-1 | c-3 f-1 |
| c-1 f-2 | c-2 f-2 | c-3 f-2 |

Este sería el resultado de una tabla con border y sin espacios.

Escribe el siguiente código y compruébalo.

```
<html>
<head>
<title>Ejercicio de Tablas</title>
<style
type="text/css"
> .normal {
width: 150 px;
border: 1px
solid #000000;
border-spacing: 0;
}
.normal td, .normal tr {
border: 1px solid;}
</style>
</head>
<body>
<table class="normal">
<tr>
<td>c-1 f-1</td>
<td>c-2 f-1</td>
<td>c-3 f-1</td>
</tr>
<tr>
<td>c-1 f-2</td>
<td>c-2 f-2</td>
<td>c-3 f-2</td>
</tr>
</table>
```

```
</body>  
</html>
```

También podemos añadirle un título a la tabla y para controlar su posición utilizamos la propiedad ***caption-side***. Esta propiedad puede tener 2 valores diferentes:

- Top: establece el título en en la parte superior de tabla
- Bottom: indica que el título de la tabla se mostrará después de los contenido de la tabla, en la parte inferior.

Ejemplo:

5.3

Ejercicio 4

Comprueba el resultado modificando el código escrito anteriormente (TABLA 1) para que se muestre como se ve en el ejemplo.

| | | |
|---------|---------|---------|
| c-1 f-1 | c-2 f-1 | c-3 f-1 |
| c-1 f-2 | c-2 f-2 | c-3 f-2 |

TABLA 1

Esta tabla sería el resultado de la utilización de la propiedad ***caption-side*** con un valor de ***bottom***.

5.4

Ejercicio 5

Realiza un diseño tabla con el de la muestra.

VENTAS 2008

| VENTAS | ENERO | FEBRERO | MARZO |
|-----------|----------|----------|----------|
| BIZKAIA | 10.250 € | 15.560 € | 10.360 € |
| GUIPUZKOA | 12.360 € | 8.950 € | 18.500 € |
| ARABA | 9.635 € | 6.580 € | 11.250 € |

Datos adicionales: ○

Letra utilizada:

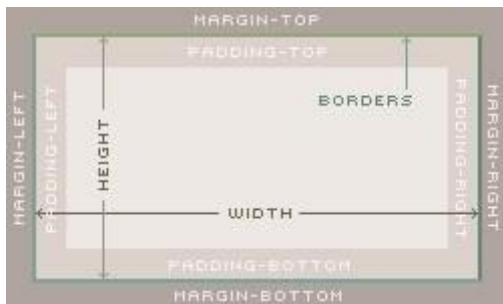
Arial, .9 em ○

Tabla centrada

6 MODELO DE CAJAS

Las cajas es probablemente la característica más importante del lenguaje de hojas de estilos, ya que con ellas organizamos el diseño de las páginas Web.

En este dibujo puedes ver la distribución con las propiedades correspondientes para tratar con cajas.



Están indicadas cada propiedad de separación y marcación (border) para las cajas. Luego las explicaremos más ampliamente.

Las cajas de la página se crean automáticamente cada vez que se inserta una etiqueta o elemento en la página.

Podemos posicionar nuestros elementos usando las propiedades **left** y **top** de

dos formas:

- Absoluta: Referida a la esquina superior izquierda
- Relativa: Referida a la posición que debería ocupar si siguiese el flujo normal (esta es la opción por defecto)

Para los bloques o cajas definiremos los espacios de separación entre la caja y su contenido.

| | Margen | Relleno | Bordes | Estilo |
|--------|--|--|--|--------------|
| NOMBRE | margin-left, margin-right, margin-top, margin-bottom, margin | padding-left, padding-right, padding-top, padding-bottom, padding | border-left-width, border-right-width, border-top-width, border-bottom-width, border-width | border-style |

El ancho de la caja será el resultante del ancho del original (contenido o content), los márgenes, los bordes y los rellenos.

Ej:

```
<!DOCTYPE html>
<html>
<head>
<title>Propiedades de Caja</title>
<meta charset="utf-8">
<style type="text/css">
.margen {
    margin-top:2em; margin-right:2em;
marginbottom:2em;marginleft:2em;
    border-style:solid;
}
.relleno {
    padding-top:2em; padding-right:2em; padding-
bottom:2em; paddingleft:2em;        border-
style:solid;
}
</style>
<body>
<p> A continuación el ejemplo de márgenes </p>
<p class="margen">En este ejemplo veremos las
separación producida por la propiedad
"margin", existe una separación indicada en el estilo,
de 2em, con respecto al texto.</p>

<p>A continuación el ejemplo de Relleno </p>
<p class="relleno">En este ejemplo veremos la
separación producida por la propiedad "padding", existe
una separación en el interior de la caja desde el borde
al texto.</p>

</body>
</html>
```


A continuación el ejemplo de márgenes

En este ejemplo veremos la separación producida por la propiedad "margin", existe una separación indicada en el estilo, de 2em, con respecto al texto.

A continuación el ejemplo de Relleno

En este ejemplo veremos la separación producida por la propiedad "padding", existe una separación en el interior de la caja desde el borde al texto.

Resultado final del código

Como vemos el margen hace la separación con respecto al margen de la página y **padding** la separación interior. Hemos añadido un **border** para ver mejor la diferencia.

Sabiendo esto, el título lo vamos a posicionar de manera absoluta. El menú será colocado en la parte derecha y con una anchura de 200px y el texto quedará en el hueco restante, a la izquierda del menú:

```
#contenedor{ /* el estilo elegido para el contenedor */
}

#titulo {position:absolute; left:100px; top: 50px; }

#menu {position: relative; float:right; width:200px; }

#textoPagina {position: absolute; top:100px; }
```

Resultado:



Este diseño, aunque en apariencia válido, tiene un problema relacionado con el posicionamiento absoluto. Si en `textoPagina` no especificamos la anchura por defecto usará toda la pantalla. Y eso incluye la parte del menú.

Por lo tanto quedarán solapados, entonces ¿Qué es mejor, utilizar posicionamiento absoluto o relativo? Pues depende. Si tenemos una anchura fija quizás sea más adecuado usar posicionamiento absoluto (teniendo en cuenta siempre las anchuras parciales). En caso de que tengamos anchura variable, lo mejor será usar posicionamiento relativo.

7 MAQUETACION

Tamaño de los elementos de la estructura principal <div>, <header>, <footer> ...

Supongamos que diseñamos una página web con un ancho de 900px. Creamos una

```
<div id="contenedor">
```

```
</div>
```

y le aplicamos un estilo

```
#contenedor{ width:900px; background-color:#ccc; margin:auto}/*el margin auto nos centra la  
página en la ventana, en el body*/
```

Pruébalo.

Para el monitor en el que esté trabajando esa medida es absoluta si no cambio la configuración del monitor. Lo que ocurrirá es que si hago más pequeña la ventana de navegación el ancho de la web no varía, perdería de vista parte de la página.

Pruébalo.

Si cambio la configuración del monitor entonces veré la página más pequeña o más grande en función de la configuración que haya establecido.

Pruébalo.

Ahora supongamos que en lugar de 900px le aplicamos un ancho del 80%. Creamos una

```
<div id="contenedor">  
</div>
```

y le aplicamos un estilo

```
#contenedor{ width:80%; background-color:#ccc; margin:auto}
```

La página tomará el 80% del elemento que contiene a la div, es decir, del body. Lo que ocurrirá en este caso es que al hacer más grande o pequeña la ventana del navegador, la página se redimensionará en función de la ventana.

Pruébalo.

Por lo tanto, el porcentaje es una unidad de medida relativa.

7.1 POSITION

La propiedad que define el posicionamiento del elemento es **position**.

Puede tener 4 valores:

- **Static:** Se tulipa por defecto y todos los elementos al principio se muestran de esta forma. No tienen en cuenta los valores de las propiedades: top, right, bottom y left que veremos posteriormente.
- **Relative:** la posición del elemento surge a partir de la posición que tendría si no se utilizara la propiedad position. La posición de los demás elementos no se ven afectados.
- **Absolute:** la posición se determina mediante las propiedades top, right, bottom y left.
- **Fixed:** La posición se calcula igual al posicionamiento absoluto, pero el elemento no se mueve cuando se desplaza la pantalla).

Los valores utilizados para la posición son:

- Top: superior
- Right: lateral derecho
- Bottom: inferior
- Left: Lateral izquierdo

Ejemplo:

```
img.posicion { position: relative; top: 7em;
}
```

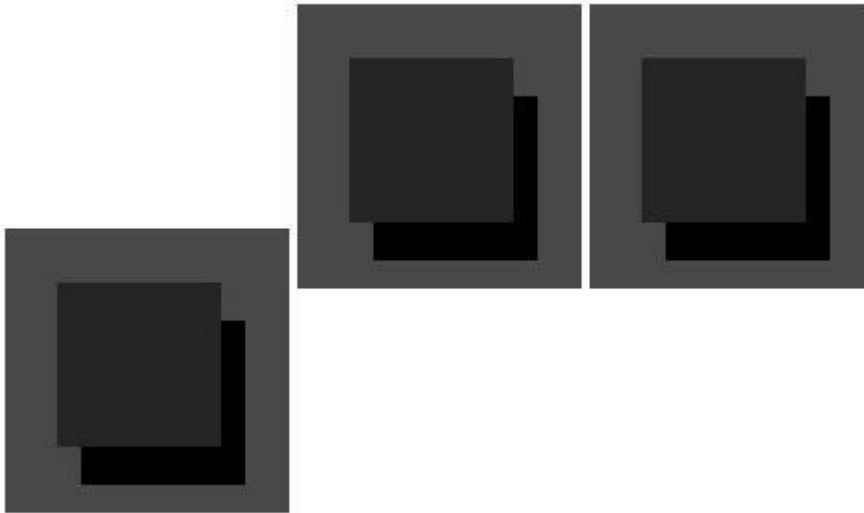
```



```

Así quedará.

EJEMPLO DE POSICIONAMIENTO



Aquí tienes el código completo, Pruébalo:

```
<!DOCTYPE html>
<html>
<head>
<title>Ejemplo de Posicionamiento</title>
<style type="text/css">

img.posicio
n {
position:
relative;
    top: 7em;
}

</style>
</head>

<body>
<p><B>EJEMPLO DE POSICIONAMIENTO</P>
<img src=imagenes/imagen1.jpg alt="Imagen ejemplo"
class="posicion">
<img src=imagenes/imagen1.jpg>
<img src=imagenes/imagen1.jpg>
</body>
</html>
```

Si queremos colocar una única imagen debemos controlar su posición para ello utilizaremos la propiedad ***background-position***.

Las palabras clave permitidas son equivalente a algunos porcentajes significativos: top=0%, left=0%, center=25%, bottom=50%, right=100%.

Se pueden mezclar porcentajes y palabras clave, como por ejemplo 50% 3cm, center 2cm, center 10%..

Si utilizamos sólo palabras clave, el orden es indiferente y por tanto, es equivalente a indicar **top left** y **left top**. Podemos utilizar palabras clave como **bottom left**, **right top**, etc.

Podemos indicar que el fondo permanezca fijo cuando la ventana del navegador se desplaza mediante las barras scroll, para ello utilizamos la propiedad ***backgroundattachment***. Los valores que puede tomar son los siguientes:

- scroll
- fixed (dejar la imagen de fondo fija)

7.2 FLOAT

Dentro de posicionamiento, decir que, uno de los más utilizados es **float**. Su función es desplazar una caja hasta la zona más a la izquierda o más a la derecha de la línea en la que se debería mostrar si no se desplazara.

Hay que destacar la propiedad **float**. El efecto de esta propiedad es bastante simple pero resulta muy útil para que el diseño de la página se adapte al monitor del usuario. Lo que hace la propiedad es quitar al elemento del flujo normal de la página y colocarlo a la izquierda o derecha del resto del contenido de su elemento padre (se lo hace "flotar" hacia uno u otro costado).

Los valores de esta propiedad son:

- left: Izquierda
- right: Derecha
- none: ninguno, elimina el posicionamiento y muestra el elemento en su posición original.

Los elementos que se encuentran alrededor de un elemento que tiene la propiedad *float*, se adaptan para que se coloquen alrededor del elemento con *float*.

Float permite que los textos se coloquen alrededor de las imágenes u otros elementos.

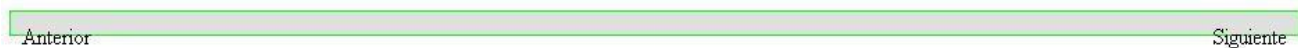
7.3 CLEAR

Dentro del posicionamiento tenemos la propiedad **clear**, esta permite resolver el comportamiento por defecto de los **float** y fuerza a un elemento a mostrarse debajo de otro que fue posicionado con **float**.

Tiene los siguientes valores:

- **right**: Se desplaza a la derecha de forma descendente has que pueda colocarse en una línea en la que no haya ningún elemento en el lado derecho.
 - **left**: Igual que el valor right pero a la izquierda
- **both**: Ambos lados.

En el ejemplo que ves a continuación, puede ver 3 elementos, una caja y dos texto, los cuales quedan descuadrados con respecto a la caja; En este caso no fue utilizada la propiedad **clear**.



En este ejemplo ves el resultado con la propiedad clear.



Prueba el siguiente código, para ver el resultado del ejemplo:

```
<html>
<head>
<title>Ejercicio de posicionamiento FLOAT</title>
<style type="text/css">
div#posicion { border: 1px solid #00ff00;
backgroundcolor:#E0E0E0;
padding: .5em; }
.derecha {
float:right;
}
.izquierda{
float:left;
```

```
}
div.clear{
  clear: both;
}
</style>
</head>
<body>
<div id="posicion">
<span class="izquierda">Anterior </span><span
class="derecha">Siguiente</span>
<div class="clear"></div>

</div>

</body>
</html>
```

VISUALIZACIÓN

Veremos las cuatro propiedades que controlan la visualización:

- **display**: hace visible o no un elemento
- **visibility**: igual que display pero mantiene la estructura.
- **Overflow**: Permite controlar los contenidos sobrantes de un elemento.
- **z-index**

Comencemos por las órdenes **display** y **visibility**, permiten hacer invisible cualquier elemento de la página. Se suelen utilizar junto con JavaScript para crear efectos y aplicaciones dinámicas.

La propiedad **display** se utiliza para no visualizar los elementos y para que el resto de elementos se visualicen ocupando el espacio del elemento oculto. La orden **visibility**, mantiene la estructura de la página pero no muestra el elemento oculto, así que los elementos se mantiene en su lugar.

Cuando creamos cajas en un página Web, puede que el contenido sea más grande que el tamaño de la caja, por lo que podemos indicar que hacer con, por ejemplo el texto, que sobresale de la caja.

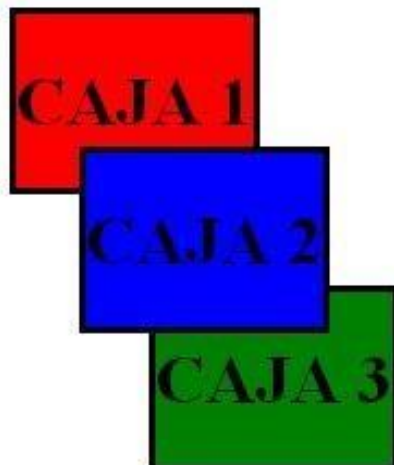
Para ello utilizamos la propiedad **overflow**, la cual tiene 3 valores:

- **visible**: el contenido que sobresale se muestra en su totalidad. No respeta el tamaño de la caja.
- **hidden**: sólo se muestra el contenido hasta los límites de la caja.
- **scroll**: se visualiza el contenido que cabe dentro de la caja, pero también se muestran barras de scroll, las cuales permiten visualizar el resto del contenido.

La propiedad **z-index** permite asignar una posición tridimensional a las cajas.

Ejemplo:

```
<html>
<head>
<title>Ejemplo de propiedad zindex</title> <style
type="text/css"> div
{
position:absolute;
} #caja1 {
z-index: 1; top:1em; left:7em; border:solid;
background:red;
} #caja2 { z-index: 3; top:5em; left:9em;
border:solid; background:blue; } #caja3 {
z-index: 2; top:9em; left:11em; border:solid;
background:green;
}
</style>
</head>
<body>
<div id="caja1"><h1>CAJA
1</h1></div>
<div id="caja2"><h1>CAJA
2</h1></div>
<div id="caja3"><h1>CAJA
3</h1></div>
</body>
</html>
```



Cómo se puede observar en el ejemplo, le hemos asignado a cada caja en su propiedad **z-index** un número, este número se refiere a la prioridad de caja, por ejemplo la caja1 tiene el número 1, por lo que la caja está por debajo de todas, la caja3 el número 3 por lo tanto es la que está por encima de todas. Con este ejemplo podemos ver claramente que estamos trabajando con capas y la propiedad **z-index** indica el orden de preferencia.

8 FINAL

Felicidades, ya sabes CSS!

Puedes ver todas un recopilatorio de las etiquetas de CSS y sus ejemplos en el siguiente enlace:

<https://www.w3schools.com/cssref/default.asp>

CIPSA.NET