



## HINTS

### QUÉ ES XML

Acrónimo de *Extensible Markup Language*, es un metalenguaje (lenguaje que se utiliza para decir algo acerca de otro) extensible de etiquetas que fue desarrollado por la **W3C**. Se considera que **XML** es un sistema que permite definir lenguajes de acuerdo con las necesidades.

### DESCENDENCIA

Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento. En el ejemplo, **option** es descendiente de **select**.

```
1 <select name="" id="">
2   <option value=""></option>
3 </select>
```

### ALT SHIF F

Cuando estamos escribiendo nuestro código fuente, en reiteradas ocasiones este se desordenará, haciendo compleja la lectura del documento y no siendo claro en el orden jerárquico de los elementos (es decir, que elemento se encuentra dentro de que elemento).

```
1 <ol>
2   <li>a</li>
3   <li>b</li></ol>
4
5   <div><p>hola</p></div>
6   <table><thead></thead><tbody>
7
8   </tbody></table>
```

Para solucionar este problema y ordenar el código, podemos hacer clic derecho con el *mouse* y seleccionar “dar formato al documento” o el atajo de teclado **alt + shif + f** que cumple la misma función.

```
1  <ol>
2      <li>a</li>
3      <li>b</li>
4  </ol>
5
6  <div>
7      <p>hola</p>
8  </div>
9  <table>
10     <thead></thead>
11     <tbody>
12
13     </tbody>
14 </table>
```

## COMENTARIOS

Sintaxis para escribir un comentario en **CSS**

```
1 /*soy
2 un
3 comentario*/
```

## CENTRALIZAR TEXTO

existe en **HTML** una etiqueta llamada **<center>** que permite centralizar el texto que coloquemos entre sus etiquetas. Esta es una práctica obsoleta y se recomienda usar, en el archivo CSS el atributo **text-align: center;**

## CÓDIGOS DE COLORES

**CSS** te ofrece una amplia gama de colores para elegir, sin embargo, puedes hacer uso de este enlace: <https://htmlcolorcodes.com/es/> para acceder a los códigos de colores que desees utilizar.

## PROPIEDADES DE CSS

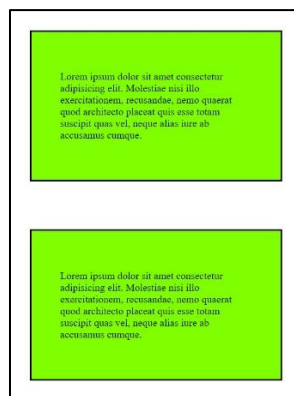
**CSS** tiene una enorme cantidad de propiedades y elementos decorativos que se pueden utilizar, por ejemplo, dentro de la lectura de hoy vimos los colores de fondo, de letra, el uso de márgenes, de padding, de bordes y los posicionamientos en línea o en bloque, pero mencionar todas sus propiedades y elementos uno por uno puede ser muy extenso, por eso los invito a visitar páginas como la de **mozilla** o **w3school** para conocer más detalles de este lenguaje. Acá se comparten interesantes enlaces:

- [CSS básico](#)
- [CSS](#)
- [Referencia CSS](#)
- [CSS Tutoriales](#)
- [Tutorial CSS](#)

## POSICIÓN DE LOS ELEMENTOS

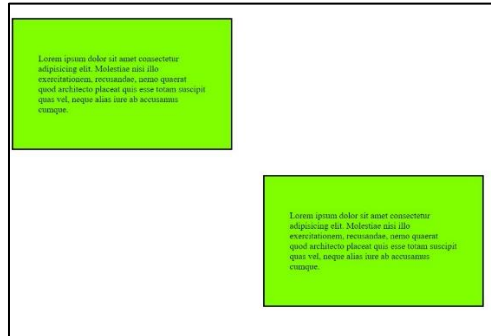
Como mencionamos, los elementos en bloque siempre ocupan todo el espacio disponible y luego tienen un salto de línea, posicionando el siguiente elemento siempre debajo.

Esta posición se llama **static** y es la posición original para esta clase de elementos.

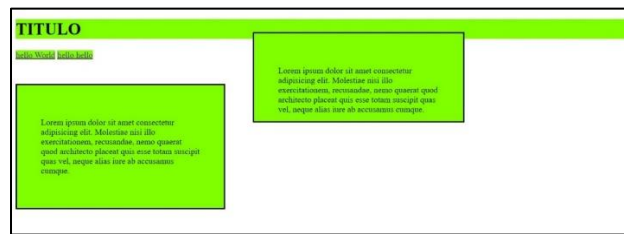


Pero existen otras posiciones que nombraremos acá:

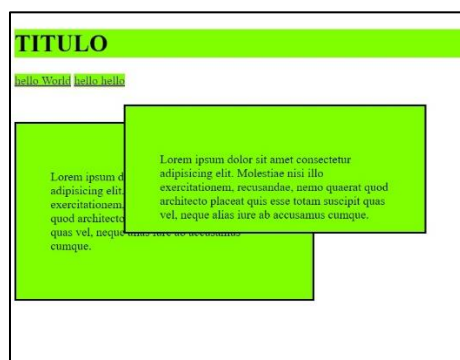
**Posición Relative:** Esta posición trabaja en conjunto con las propiedades *top*, *bottom*, *right* y *left*.



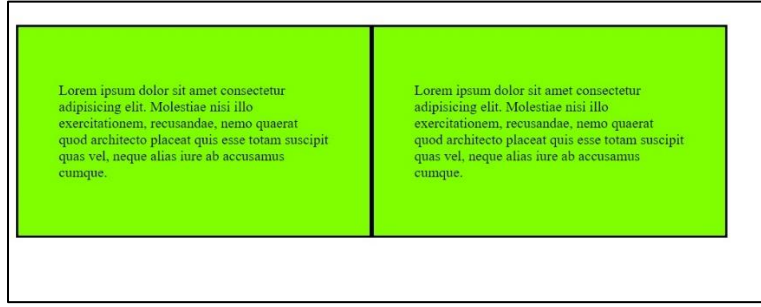
**Posición Absolute:** Trabaja con las mismas propiedades de posicionamiento de **relative**, pero lleva el elemento hasta la parte más alta de la página, perdiendo por completo el espacio en bloque.



**Posición Fixed:** Esta posición hace que el elemento quede fijo en alguna parte de la página y no se mueva de ahí, independiente de si tiene otros elementos debajo de él (es decir, queda superpuesto) o se mueve la página.



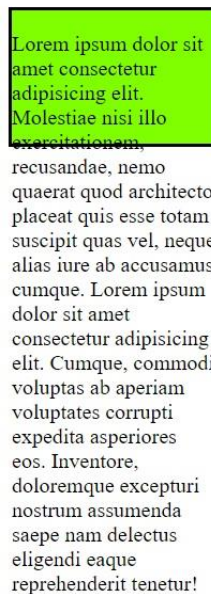
**Posición float:** esta posición no utiliza la propiedad position para indicarla ( a diferencia de las anteriores posiciones vistas) y es la base para crear diseños con columnas.



## CONTENIDO QUE NO CABE DENTRO DE LA CAJA

Cuando damos tamaño a una caja, puede ser que achiquemos demasiado el tamaño en comparación al contenido que posee, para eso existe la propiedad **overflow**. Con esta propiedad especificamos los que ocurre cuando el contenido no cabe en la caja. Los posibles valores son:

- **Visible:** Valor predefinido. Aunque el contenido supere el tamaño de la caja, se mostrará como tal.



Lorem ipsum dolor sit  
amet consectetur  
adipiscing elit.  
Molestiae nisi illo  
exercitationem,  
recusandae, nemo  
quaerat quod architecto  
placeat quis esse totam  
suscipit quas vel, neque  
alias iure ab accusamus  
cumque. Lorem ipsum  
dolor sit amet  
consectetur adipiscing  
elit. Cumque, commodi  
voluptas ab aperiam  
voluptates corrupti  
expedita asperiores  
eos. Inventore,  
doloremque excepturi  
nostrum assumenda  
saepe nam delectus  
eligendi eaque  
reprehenderit tenetur!

- **Hidden:** Sólo aparece el contenido que no supere el límite de la caja.



Lorem ipsum dolor sit  
amet consectetur  
adipiscing elit.  
Molestiae nisi illo  
exercitationem

- **Scroll:** Aparece una barra de desplazamiento.

accusamus cumque.  
Lorem ipsum dolor  
sit amet consectetur  
adipiscing elit.  
Cumque, commodi

- **Auto:** Dejamos a criterio del navegador el comportamiento.

Lorem ipsum dolor  
sit amet consectetur  
adipiscing elit.  
Molestiae nisi illo  
exercitationem

## BUENAS PRACTICAS

Cuando utilizamos hojas de estilos, al igual que cuando utilizamos cualquier herramienta en programación, se recomienda tener presente ciertas prácticas consideradas buenas y que permiten tener un código más legible y mantenible en el tiempo. Para el caso de CSS enumeraremos algunas prácticas que pueden darte “un punto extra” al momento de desarrollar tus proyectos.

- 1.- **Organizar la hoja de estilos:** es decir, procura que tenga un orden “de arriba hacia abajo” para que sea más fácil de leer y mantener. Los primeros estilos dados serán aquellos entregados a los primeros elementos del HTML y los últimos estilos en dados en la hoja de estilo, serán aquellos entregados a los últimos elementos del HTML.



2.- **Crea primero el HTML:** si tienes la estructura previamente creada y un diseño pensado, será mucho más fácil dar el estilo sin tener que hacerlo a medida que se crea la estructura.

3.- **Combina elementos:** en reiteradas ocasiones, más de un elemento tendrá el mismo estilo. Para esos casos no redundes en instrucciones y, en una sola línea, declara todos los elementos que llevarán el mismo estilo:

```
1 h1, h2, .clase{  
2   text-align: center;  
3   margin-top: 10%;  
4 }
```

4.- **Realiza comentarios:** nunca es bueno excederse en los comentarios, porque si tu código requiere ser explicado línea por línea para ser entendido, no es un buen código, sin embargo, siempre es bueno ir dejando ciertos comentarios que guíen al resto en el propósito de ciertos bloques.

5.- **código legible:** esto se repite a cada momento, pero es importante que el código este escrito de manera tal que leerlo y mantenerlo sea simple para cualquier persona que pueda llegar a trabajar en él y no sea un código que tú y solo tú podrás entender por su desorden. En este aspecto podríamos decir que “menos es más”.