

-CSS vendor prefixes

son una forma de que los fabricantes de navegadores agreguen soporte para nuevas funciones de CSS antes de que esas funciones sean totalmente compatibles con todos los navegadores. Esto se puede hacer durante una especie de período de prueba y experimentación en el que el fabricante del navegador está determinando exactamente cómo se implementarán estas nuevas características de CSS.

<https://www.lifewire.com/css-vendor-prefixes-3466867>

prefix	Browser
-webkit-	Chrome, Safari, Android, iOS
-moz-	Firefox
-o-	Opera
-ms-	Microsoft Internet Explorer

<https://codepen.io/paqui-molina/pen/PyyZgV>

Styles

Columns

Multiple columns

https://www.w3schools.com/css/css3_multiple_columns.asp

<https://codepen.io/paqui-molina/pen/NWvYQae>

CSS lists

https://www.w3schools.com/css/css_list.asp

CSS Table

Tables are used to arrange data in tabular format - rows and columns of cells. You can put a variety of data like text, images, forms, links and even other tables in your table.

You may hear experienced Web developers decrying the use of tables, giving the impression that tables should be avoided at all costs. It might seem off-putting at first, but they're not really talking about using tables for tabular information. In earlier days, when layout options were limited, many developers resorted to tables as a means of layout. There's really no need to do that anymore, there are plenty of layout capabilities in CSS3. You really don't want to use tables that way, but there absolutely nothing wrong with using them for their intended purpose --making tables.

Separating content and style

Tables are **semantically incorrect** for layout because they represent presentation and not content. It puts presentation data in your content making your **HTML larger**. The user must download this unnecessary presentation data for every page they visit.

Accessibility: tables are not very screen reader friendly. Using tables for layout will clutter your HTML making it harder for assistive technology to parse your Web page.

Redesigns are harder when your HTML is cluttered with presentational code that should go into CSS. To change the layout of the page, you shouldn't be editing your content. Instead, you should just have to make CSS related changes.

Using CSS (one or two external stylesheets for your whole Web site) is easier to maintain consistency among pages.

Tables were not intended as a layout tool, so it is best to stick to them only for tabular data.

Table elements:

Type	Element
	<table>
	<caption>
Row groups	<thead>, <tfoot>,<tbody>
Col groups	<colgroup>,<col>
Table row	<tr>
Table cells	<th>,<td>

We will use these elements to build our table

Standards CSS for some elements:

```
table { display: table; border-collapse: separate; border-spacing: 2px; border-color: gray; }
```

```
tr { display: table-row; vertical-align: inherit; border-color: inherit;}
```

```
td { display: table-cell; vertical-align: inherit; }
```

```
th { display: table-cell; vertical-align: inherit; font-weight: bold; text-align: center; }
```

<https://codepen.io/paqui-molina/pen/yLLvvgj>

<https://codepen.io/paqui-molina/pen/eYYVajb>

The <colgroup> and <col> tags

<https://codepen.io/paqui-molina/pen/MWWQdxz>

<colgroup>

This tag allows you to group columns in a table. Grouping columns is useful if you want to specify properties for a group of columns like applying styles to the whole column instead of repeating it for each cell.

Attribute:

- `span` - takes a positive integer value. It specifies the number of columns you want your `colgroup` to span (cover). The `colgroup` element shares its attributes like style and width with all the columns it spans. Essentially it allows a single cell to stretch to cover multiple columns on a particular row.

<col>

Used within `<colgroup>`, the `<col>` tag specifies the column property for each column within a `colgroup`. The only element a `<colgroup>` can contain is `<col>`.

Attribute:

- `span` - takes a positive integer value. It specifies the number of columns you want the `col` element to span (cover).

Consider the table above we created using `<tr>`, `<th>` and `<td>`. Let's say I want the 'name' column to be in green and the 'age' column to be orange. You need to use the `<colgroup>` and `<col>` tags to achieve styling effects specific to a column.

The `thead`, `tbody` and `tfoot` tags

Similar to an HTML document, a table in HTML can be split into header, body and footer. We use these three tags - `<thead>`, `<tbody>` and `<tfoot>` - to specify parts of a table.

<https://codepen.io/paqui-molina/pen/yLLvvgi>

https://www.w3schools.com/html/html_tables.asp

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table>

Styles in your table

https://www.w3schools.com/css/css_table.asp

//w3css_tables

https://www.w3schools.com/w3css/w3css_tables.asp

Formularios HTML5

<http://www.mclibre.org/consultar/htmlcss/html/html-formularios.html>

https://www.w3schools.com/html/html_forms.asp

<https://www.html.am/tags/html-output-tag.cfm>

<http://desarrolloweb.dlsi.ua.es/cursos/2012/nuevos-estandares-desarrollo-sitios-web/html5-formularios>

Style forms

https://www.w3schools.com/css/css_form.asp

Elementos de formulario

<https://codepen.io/paqui-molina/pen/VwjdpYj?editors=1100>

Formulario sencillo

<https://codepen.io/paqui-molina/pen/RwReejq>

<https://codepen.io/paqui-molina/pen/jOLKPxb>

Estilizando radio, checkbox

<https://codepen.io/paqui-molina/pen/zYBaomb>

Mezclar pseudoclases

<https://codepen.io/paqui-molina/pen/NOewOd>

Align forms without tables:

- legend
- fieldset
- label

pseudoclasses to forms

Focus: when the input has the focus

Invalid: while the content of the field is not valid

Valid: while the content of the field is valid

Required: When the attribute required has been used

Optional: the opposite, podemos seleccionar todos los campos que no tengan aplicado el atributo "required" mediante la pseudo-clase "optional"

In-range: dates or number field. While the value is between the range, min and max.

Out-of-range: When the value is out of the rang

;checked any radio, checkbox and option that is selected

:read-only elements that the user can edit

:read-write elements that the user can not edit

we can mix pseudoclass how we want

/ the input has the focus, it's required and, for the moment, it's no valid */*

```
input:focus:required:invalid {  
    background-color:red;  
}
```

CSS3 fonts

[http://www.mclibre.org/consultar/amaya/css/css-fuentes-web.html#Cualquier tipo de fuente](http://www.mclibre.org/consultar/amaya/css/css-fuentes-web.html#Cualquier%20tipo%20de%20fuente)

<https://fonts.google.com/>

Fuentes de forma local

<https://canonicalized.com/local-font/>

<https://codepen.io/paqui-molina/pen/JjjvowL>

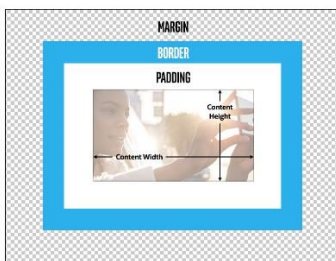
<https://codepen.io/paqui-molina/pen/GRRdJgV>

<https://codepen.io/paqui-molina/pen/WNNJvoo>

Box model

El modelo de caja controla 3 características importantes de espaciado de CSS. Anteriormente estudiamos margin como el espacio entre elementos. Hay otras dos nociones similares, padding y border.

Todos los elementos de un documento html terminan siendo tratados como rectángulos en algún lugar de la vista. El contenido(content) de cada rectángulo corresponde al rectángulo más interno de la imagen. Justo fuera del contenido está el relleno(padding). Lo que significa que separa el contenido del borde(border). Básicamente, el borde traza los lados del rectángulo de relleno.



El **margen**, especifica la posición del elemento en relación con lo que está adyacente a él, ya sea a la derecha o izquierda, o arriba o abajo. El margen es siempre transparente y cada lado se puede configurar individualmente. Lo único del margen es que los valores de cualquiera de los lados pueden ser negativos, incluso si eso significa que se superpone con otro elemento de la página. Esto puede resultar útil cuando queremos controlar dónde se coloca un elemento en una página.

<https://codepen.io/paqui-molina/pen/rNLQmMe>

Modelo de caja estándar o tradicional

En el modelo de cajas estándar, cuando estableces los atributos width y height para una caja, defines el ancho y el alto del *contenido de la caja*. Cualquier área de relleno y borde se añade a ese ancho y alto para obtener el tamaño total que ocupa la caja. Esto se muestra en la imagen que encontrarás a continuación.

Si suponemos que la caja tiene el CSS siguiente,

El espacio que ocupa nuestra caja, usando el modelo de cajas estándar, es

Ancho: $350 + 25 + 25 + 5 + 5 = 410\text{px}$

Alto: $150 + 25 + 25 + 5 + 5 = 210\text{px}$

```
.box {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  border: 5px solid black;  
}
```

El **margen no se cuenta para el tamaño real de la caja**; por supuesto, afecta al espacio total que la caja ocupa en la página, pero solo al espacio de fuera de la caja. El área de la caja se termina en el borde, no se extiende hasta el margen.

Modelo de caja alternativo

Con este modelo, cualquier ancho es el ancho de la caja visible en la página, por lo tanto, el ancho del área de contenido es ese ancho menos el ancho para el relleno y el borde.

En el ejemplo anterior, el ancho es 350px y el alto es 150px.

Por defecto, los navegadores usan el modelo de cajas estándar. Si deseas activar el modelo de cajas alternativo para un elemento, hazlo configurando box-sizing: border-box.

<https://codepen.io/paqui-molina/pen/LYZXLRX>

https://developer.mozilla.org/es/docs/Learn/CSS/Introduction_to_CSS/Modelo_cajas

box model (background-clip): <https://codepen.io/paqui-molina/pen/aRrVzO>

Tipos de cajas (line, block, inline-block): <https://codepen.io/paqui-molina/pen/OBYdOW>

Outline: https://www.w3schools.com/css/css_outline.asp

<https://codepen.io/paqui-molina/pen/RemdMx>

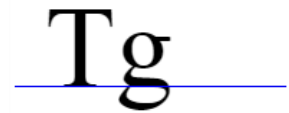
Display: https://www.w3schools.com/css/css_display_visibility.asp

Overflow: https://www.w3schools.com/css/css_overflow.asp

<https://codepen.io/paqui-molina/pen/OBYrXG>

Elementos de línea y elementos de bloque

Baseline (línea de base o línea de guía) es un concepto clave para comprender cómo el navegador toma sus decisiones de diseño. Baseline determina cómo y dónde se colocan los elementos. Gobierna la ubicación de todos los elementos en línea.



Por ejemplo, el navegador le dará una nueva línea a una etiqueta `<p>` pero NO a una etiqueta `<a>`. Ésta es la distinción clave entre los elementos de nivel "bloque" (como la etiqueta `<p>`) y los elementos "en línea" (como la etiqueta `<a>`). Aquí hay una tabla rápida de los valores predeterminados para algunas de las etiquetas que ya hemos aprendido.

Diferencias entre elementos de bloque(block) y elementos en línea(inline)

Elementos de bloque

- aparece debajo y a la izquierda de sus vecinos de nivel de bloque
- se expandirá para llenar el ancho del contenedor principal de forma predeterminada
- respeta todas las propiedades de los márgenes
- puede tener configurada su propiedad `width`, lo que lo hará más estrecho y hará que sus hijos se envuelvan, pero no se recorten.
- toma la altura de todos sus hijos (en espera de ciertas excepciones) siempre que su propia altura no esté definida.
- ignora la propiedad `vertical-align`

Elementos en línea

- Aparece a la derecha de su vecino en línea anterior. No pasan a la siguiente línea a menos que deban "ajustar".
- Por defecto, el ancho es simplemente el ancho del contenido del elemento, más cualquier relleno
- Ignoran la configuración de los márgenes superior e inferior
- Ignoran las propiedades de `width` y `height`
- Están sujetos a la propiedad `vertical-align`, así como a la configuración de espacios en blanco CSS • Admiten `padding`, pero cualquier parte superior o inferior no contribuye al cálculo de la altura de la línea de texto sobre la que se asienta
- Se adhieren a la línea de base donde se colocan

Centrado horizontal y vertical

Centrado horizontal

Inline

Para centrar un elemento inline usamos la propiedad `text-align: center` en el elemento padre

```
p {text-align: center; } /* the text and any inline children of this element will be centered */
```

Block

La forma más sencilla de centrar un elemento de bloque es aplicar auto a los márgenes izquierdo y derecho, además de dimensionar el elemento.

```
div {width: 200px; margin:auto}
```

Centrado vertical

Inline

Los elementos en línea respetan la propiedad de `vertical-align`. Determina cómo se alinea el elemento en línea en relación con la línea guía o baseline sobre la que se coloca.

Block

No hay `margin:auto` para un centrado vertical. Esto lo resolveremos más adelante con flexbox.

Propiedad position

Propiedades: left, top, right, and bottom

En CSS, hay cuatro propiedades que se pueden usar para ajustar o establecer la posición de un elemento. Esas propiedades son: `left`, `top`, `right`, and `bottom`.

Sin embargo, el uso de esas propiedades por sí mismas no tendrá ningún efecto en ningún elemento. La razón por la que estas propiedades no funcionan de forma predeterminada es que solo funcionan cuando se aplican a elementos posicionados. Y los elementos posicionados son aquellos a los que se les ha cambiado la propiedad de posición de la predeterminada.

Position

Permite establecer la posición de un elemento

https://www.w3schools.com/css/css_positioning.asp

<https://developer.mozilla.org/es/docs/Web/CSS/position>

Tipos de posicionamiento:

- **Static:** Posicionamiento normal, por defecto.
- **Relative:** Permite que un elemento se desplace respecto a lo que hubiera sido su posición normal; el resto de elementos continúan en su posición ignorando al que se desplaza, esto puede crear superposiciones. El espacio que deja queda libre.
- **Absolute:** permite que un elemento se desplace respecto al origen de coordenadas del primer elemento contenedor posicionado o respecto a la esquina superior izquierda de la ventana de visualización; el resto de elementos actúan como si el desplazado no existiera, por lo que su espacio será ocupado por otros elementos; puede crear superposiciones.
- **Fixed:** permite fijar un elemento en una posición respecto al origen de coordenadas del primer elemento contenedor posicionado ó respecto a la esquina superior de la ventana de visualización; el elemento se mantendrá en la ventana de visualización o viewport, siempre en una misma posición aunque el usuario se desplace por la web haciendo scroll
- **Sticky:** es un elemento con posición relativa hasta que llega a un límite, (por ejemplo: puede estar establecido con top), a partir de ese momento pasa a posición fija

z-index property

https://www.w3schools.com/cssref/pr_pos_z-index.asp
<https://developer.mozilla.org/es/docs/Web/CSS/z-index>

colors and transparency

https://www.w3schools.com/css/css_colors.asp
<http://w3.unpocodetodo.info/css3/hsl.php>

Backgrounds

https://www.w3schools.com/css/css_background.asp
background: color image repeat/no-repeat fixed/scroll position

Multiple backgrounds

https://www.w3schools.com/css/css3_backgrounds.asp
background-size
<https://developer.mozilla.org/es/docs/Web/CSS/background-size>

lists that don't look like it

DropDown

https://www.w3schools.com/css/css_dropdowns.asp

https://www.w3schools.com/cssref/pr_class_visibility.asp

[display:none <> visibility:hidden](#)

<https://codepen.io/paqui-molina/pen/zMvNaE>

Bordes

https://www.w3schools.com/css/css_border.asp

https://www.w3schools.com/css/css3_borders.asp

https://www.w3schools.com/css/css3_border_images.asp

<https://developer.mozilla.org/en-US/docs/Web/CSS/border-radius>

<https://www.w3.org/TR/css-backgrounds-3/#border-radius>

<https://codepen.io/paqui-molina/pen/mdEodwz>

Shadows:

https://www.w3schools.com/css/css3_shadows.asp

<https://developer.mozilla.org/es/docs/Web/CSS/text-shadow>

shadow: desplazamiento vertical desplazamiento horizontal desenfoque color
si número negativo: desplazamiento derecha, arriba

<https://codepen.io/paqui-molina/pen/PxZKEP>

Gradients

https://www.w3schools.com/css/css3_gradients.asp

Repetición de gradientes

https://www.w3schools.com/cssref/func_repeating-linear-gradient.asp

Gradientes radiales

<https://developer.mozilla.org/es/docs/Web/CSS/radial-gradient>

Transformaciones

https://www.w3schools.com/css/css3_2dtransforms.asp

https://www.w3schools.com/css/css3_3dtransforms.asp

Listas y transformaciones

<https://codepen.io/paqui-molina/pen/QWwExYE>

Transitions

https://www.w3schools.com/css/css3_transitions.asp

Animaciones

https://www.w3schools.com/css/css3_animations.asp

<https://codepen.io/paqui-molina/full/VVeoMg>