

Todos los elementos en CSS son cuadrados o rectángulos y aparte de eso, estos elementos tienen un comportamiento que se define a través de la propiedad display. Los display más comunes y usados son: block, inline, inline-block, none, table, flex y grid. Veamos de qué se tratan:

Tipos de display	Descripción
Block	<p>Quiero que pensemos en este display como el flujo que tenemos cada vez que escribimos. Por ejemplo, nosotros escribimos de izquierda a derecha, pero cuando se nos acaba el espacio horizontalmente, seguimos abajo. Este es el flujo de bloque en nuestra escritura (de arriba a abajo) y es exactamente el que obedece display block, lo que quiere decir que los elementos se visualizan uno debajo del otro.</p> <p>Algunos elementos que por defecto tienen este tipo de display son: &lt;div&gt;, &lt;p&gt;, &lt;h1&gt;, &lt;section&gt; y &lt;ul&gt;.</p>
Inline	<p>Quiero que pensemos en este display como la línea base que seguimos para escribir, que en el caso del latín es de izquierda a derecha. Por lo tanto, todos los elementos con este display se visualizarán uno al lado del otro de izquierda a derecha.</p> <p>Algunos elementos que por defecto tienen este tipo de display son: &lt;span&gt;, &lt;em&gt; y &lt;b&gt;.</p>
Inline-block	<p>Este display es muy similar al display inline, sin embargo este respeta el width y el height que se le asigne a los elementos.</p>
None	<p>Este display es utilizado para ocultar elementos sin eliminarlos por completo del DOM.</p>
Table	<p>Los elementos con este display se comportan como los elementos &lt;table&gt;.</p>
Flex	<p>Centrar elementos, alinearlos a la derecha o a la izquierda, siempre ha sido una tarea difícil. Sin embargo, llega flexbox para ayudarnos con este dilema. Para usar flexbox es indispensable la propiedad display: flex en el elemento padre que por defecto alineará los elementos hijos de izquierda a derecha. También, este display suele acompañarse por otras propiedades como justify-content y align-items.</p> <p>Trabajar con flexbox no basta sólo con escribir display: flex, también es muy importante que entendamos cómo funciona y para ello te dejo la siguiente documentación:</p> <p><a href="https://css-tricks.com/snippets/css/a-guide-to-flexbox/">https://css-tricks.com/snippets/css/a-guide-to-flexbox/</a></p>
Grid	<p>Después de flexbox, llega el más poderoso sistema de layout: CSS Grid y para implementarlo debemos hacer uso de display: grid en el elemento contenedor o elemento padre.</p> <p>Al igual que con flexbox, CSS Grid requiere de un estudio adicional ya que no basta con colocar este display. Para ello, te dejo la siguiente guía:</p> <p><a href="https://css-tricks.com/snippets/css/complete-guide-grid/">https://css-tricks.com/snippets/css/complete-guide-grid/</a></p>