



Universidad Politécnica de Madrid

Repaso CSS

Enrique Barra Arias
Álvaro Alonso González

El **HTML** controla el aspecto gráfico del documento mediante **estilos CSS** (*cascading style sheets*), que indican al navegador como se deben visualizar los elementos de un documento HTML.

Así se consigue separar el aspecto del contenido.

[Volver a la página anterior](#)
[Arrastre hacia abajo para ver el historial](#)

The Beauty of CSS Design

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only thing that has changed is the external CSS file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

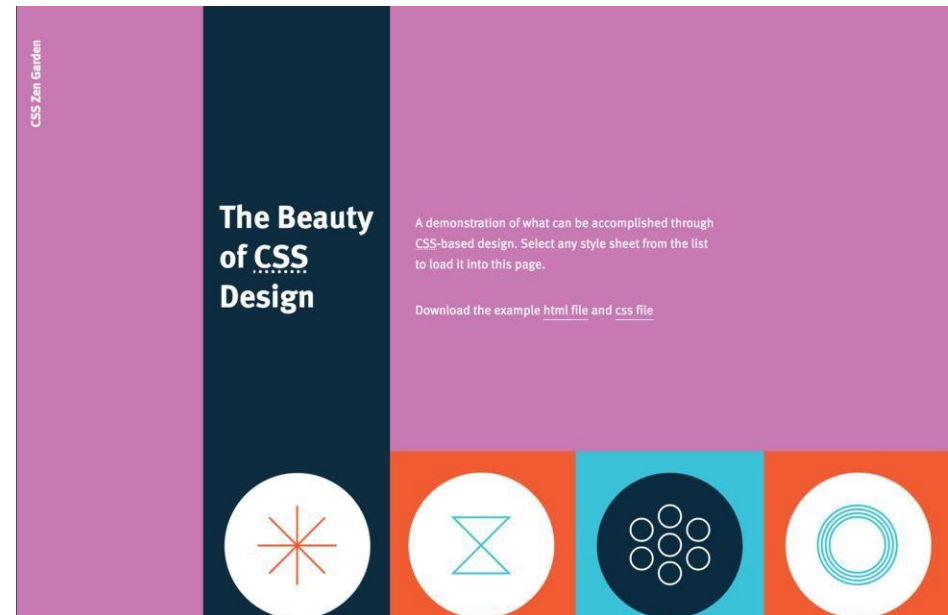
Participation

Strong visual design has always been our focus. You are modifying this page, so strong CSS skills are necessary too, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample [HTML](#) and [CSS](#) to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your CSS file to a web server under your control. [Send us a link](#) to an archive of that file and all associated assets, and if we choose to use it we will download it and place it on our server.

Benefits



Proyecto **css Zen Garden**:

<http://www.mezzoblue.com/zengarden/alldesigns/>

```
h1 { color : green; }
```

Norma de estilo

h1	{	color	:	green	;	...	}
Selector		Propiedad		Valor		Propiedad : Valor ...	
		Declaración			Declaración ...		
	Bloque de declaración						

CSS - propiedades

CSS

background

background-attachment

background-color

background-image

background-position

background-repeat

border

border-bottom

border-bottom-color

border-bottom-style

border-bottom-width

border-color

border-left

border-left-color

border-left-style

border-left-width

border-right

border-right-color

border-right-style

border-right-width

border-style

border-top

border-top-color

border-top-width

border-width

clear

clip

color

cursor

display

filter

float

font

font-family

font-size

font-variant

font-weight

height

left

letter-spacing

line-height

list-style

list-style-image

list-style-position

list-style-type

margin

margin-bottom

margin-top

overflow

padding

padding-bottom

padding-left

padding-right

padding-top

page-break-after

page-break-before

position

stroke-dasharray

stroke-dashoffset

stroke-width

text-align

text-decoration

text-decoration: blink

text-decoration: line-through

text-decoration: none

text-decoration: overline

text-decoration: underline

text-indent

text-transform

top

vertical-align

visibility

width

z-index

Para aplicar un estilo de presentación, los selectores que tenemos:

Selector **universal** `*` afecta a todos los elementos.

```
*{ margin: 0; padding: 0; }
```

El selector de **etiqueta** aplica a todos los elementos HTML de la página con esa etiqueta (p).

```
<p>Párrafo rojo</p> -> p{ color: red; }
```

Las **clases** se usan para aplicar estilos a un elemento determinado.

```
<div class="rojo">Texto rojo</div> -> .rojo{color: red;}
```

También se pueden aplicar estilos a un **id**.

```
<p id="texto">Párrafo azul</p> -> #texto {color: blue;}
```

- El **selector múltiple** de CSS, incluye varios selectores separados por coma (,), para aplicar propiedades comunes.

➤ `p, .rojo, #texto {color: red;}`
- El selector **descendente** puede incluir etiquetas **separadas solo por espacios**. Se aplicará solo a elementos que estén dentro de otros anteriores, ancestros, en el ejemplo, la clase **.caja** deberá ser ancestro de **nav** y **nav** deberá ser ancestro de **ul**.

➤ `.caja nav ul {color: red;}`
- `p.texto{color: red;}`

El estilo CSS, se puede añadir mediante un bloque marcado como **<style>** en la **cabecera (<head>)** y aplica el estilo a los elementos de esa página.

...

<head>

<style type="text/css">

body{

background-color: red;

}

.class{

color: black;

}

</style>

</head>

...

Los estilos CSS se pueden añadir escribiendo directamente las propiedades CSS **en línea**, es el método más sencillo, se añade un **atributo style** en el elemento concreto dentro de la página. No se pueden reutilizar en otros elementos que comparten las mismas propiedades.

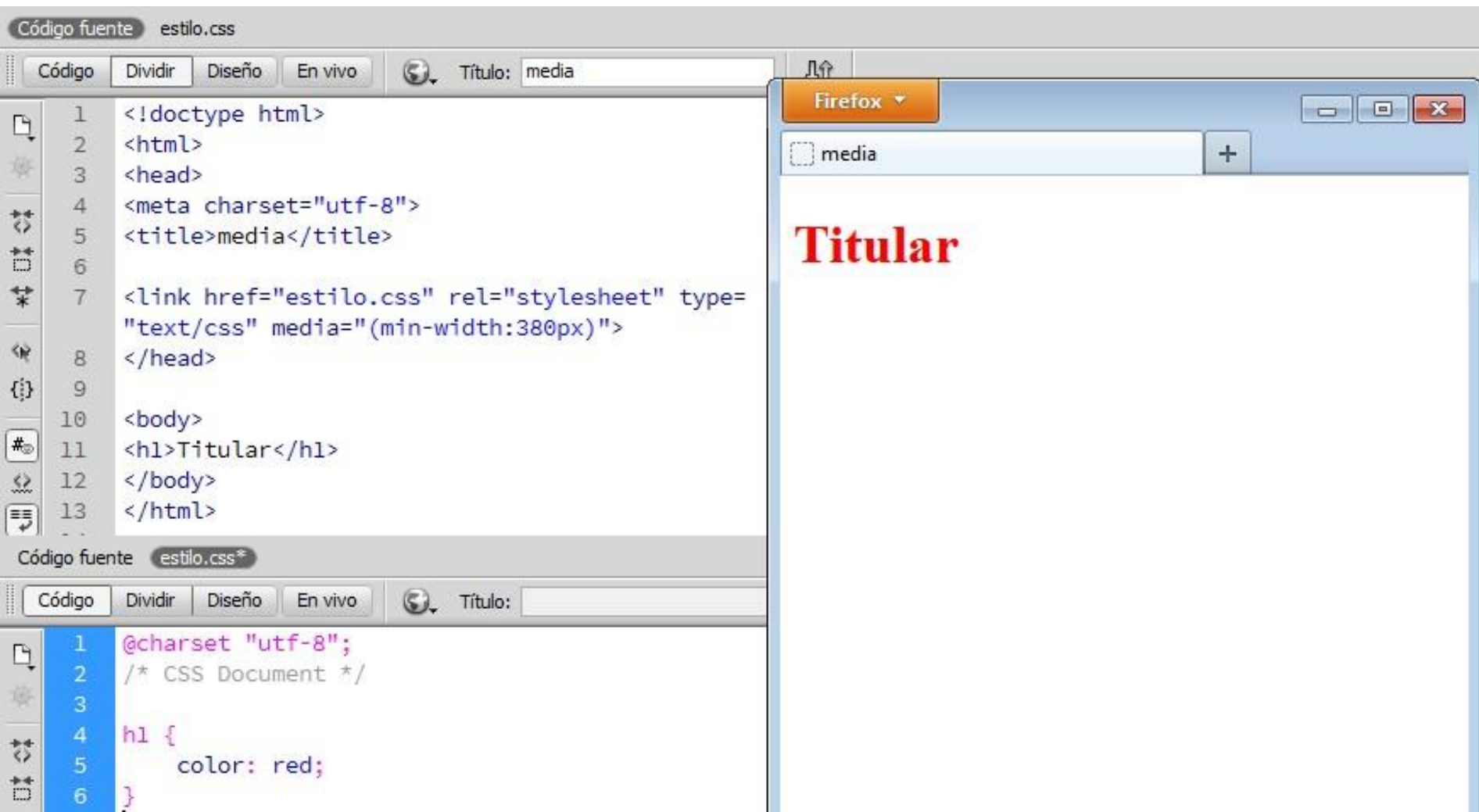
Se escriben las propiedades del estilo en las marcas del HTML como "*nombre: valor*" separadas por punto y coma.

```
<body style="color: red;">
```

```
<p style="font-size: 16px; color: blue; font-family: Arial,  
Helvetica, sans-serif;">
```

```
Mi primera página</p>
```


Lo más aconsejable para mantener la separación entre contenido y presentación, en una **hoja de estilo CSS** separada del HTML, que se importa con un elemento **<link>** en la cabecera.



The screenshot displays a web development environment with two main panels. The top panel shows the HTML source code for a file named 'estilo.css'. The code includes a DOCTYPE declaration, a meta charset declaration, a title 'media', and a link to 'estilo.css' with attributes 'rel="stylesheet"', 'type="text/css"', and 'media="(min-width:380px)"'. The body contains an h1 element with the text 'Titular'. The bottom panel shows the CSS source code for 'estilo.css*', which includes a charset declaration, a comment '/* CSS Document */', and a rule for h1 with the color set to red. To the right, a Firefox browser window is open, showing the rendered page with the title 'media' in the address bar and the word 'Titular' in large red font on a white background.

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>media</title>
6
7 <link href="estilo.css" rel="stylesheet" type=
  "text/css" media="(min-width:380px)">
8 </head>
9
10 <body>
11 <h1>Titular</h1>
12 </body>
13 </html>
```

```
1 @charset "utf-8";
2 /* CSS Document */
3
4 h1 {
5     color: red;
6 }
```

CSS Permite adaptar la presentación a PCs, móviles, tabletas o impresoras con el atributo **media** que activará el estilo específico de cada dispositivo.

```
<link rel="stylesheet" type="text/css" media="screen"
href="sans-serif.css">
```

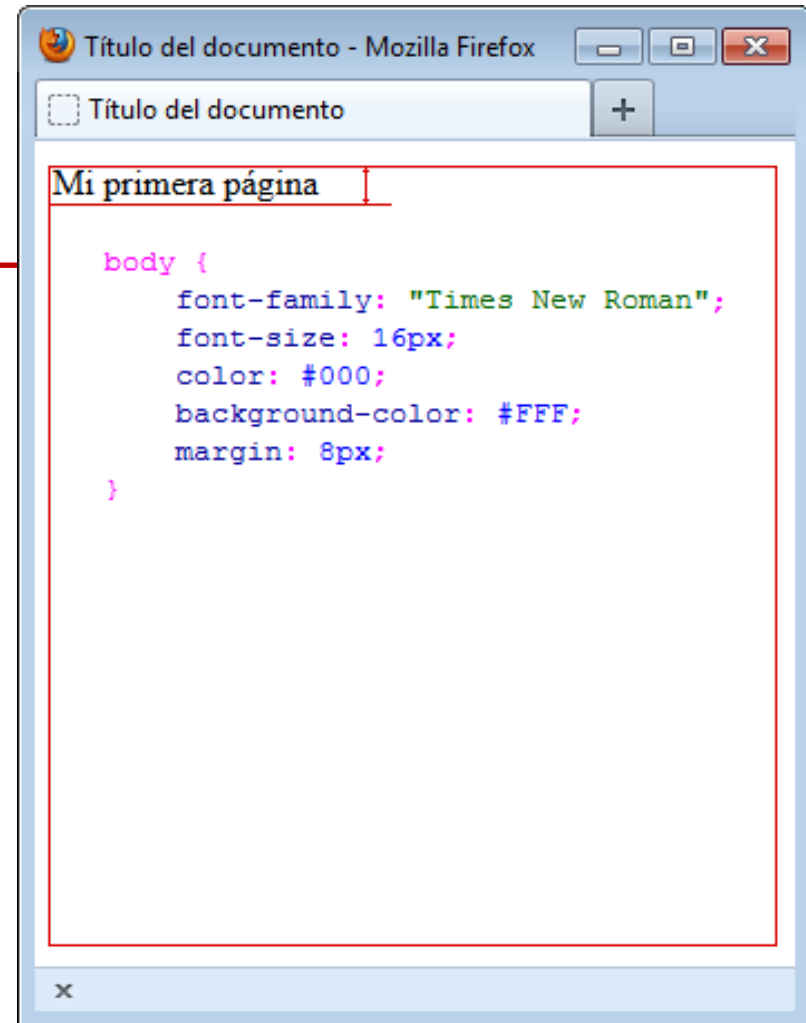
```
<link href="estilo.css" rel="stylesheet" type="text/css"
media="(min-width:380px)">
```

<http://www.w3.org/TR/css3-mediaqueries/>

Cada etiqueta HTML tiene unos valores de CSS por defecto, que pueden variar según el navegador, que tendremos que modificar con nuestros estilos personales.

Visualización valores por defecto CSS en el navegador

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Título del documento</title>
6 </head>
7
8 <body>
9 Mi primera página
10 </body>
11 </html>
12
```



Ejemplo valores CSS por defecto del navegador, de: **h1**, **p**, **strong**, **em**.

```
Code Split Design Live
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Marcas de texto</title>
6 </head>
7
8 <body>
9 <h1>Titular nivel 1</h1>
10 <p>En este párrafo hay rexta marcado
11 como <em>importante</em> y otro texto
12 marcado como <strong>muy importante.</strong></p>
13 <p>Este es el segundo párrafo.</p>
14 </body>
15 </html>
16
```

h1 {
display: block;
font-size: 2em;
margin-before: 0.67em;
margin-after: 0.67em;
margin-start: 0;
margin-end: 0;
font-weight: bold; }

p {
display: block;
margin-before: 1em;
margin-after: 1em;
margin-start: 0;
margin-end: 0; }

Titular nivel 1

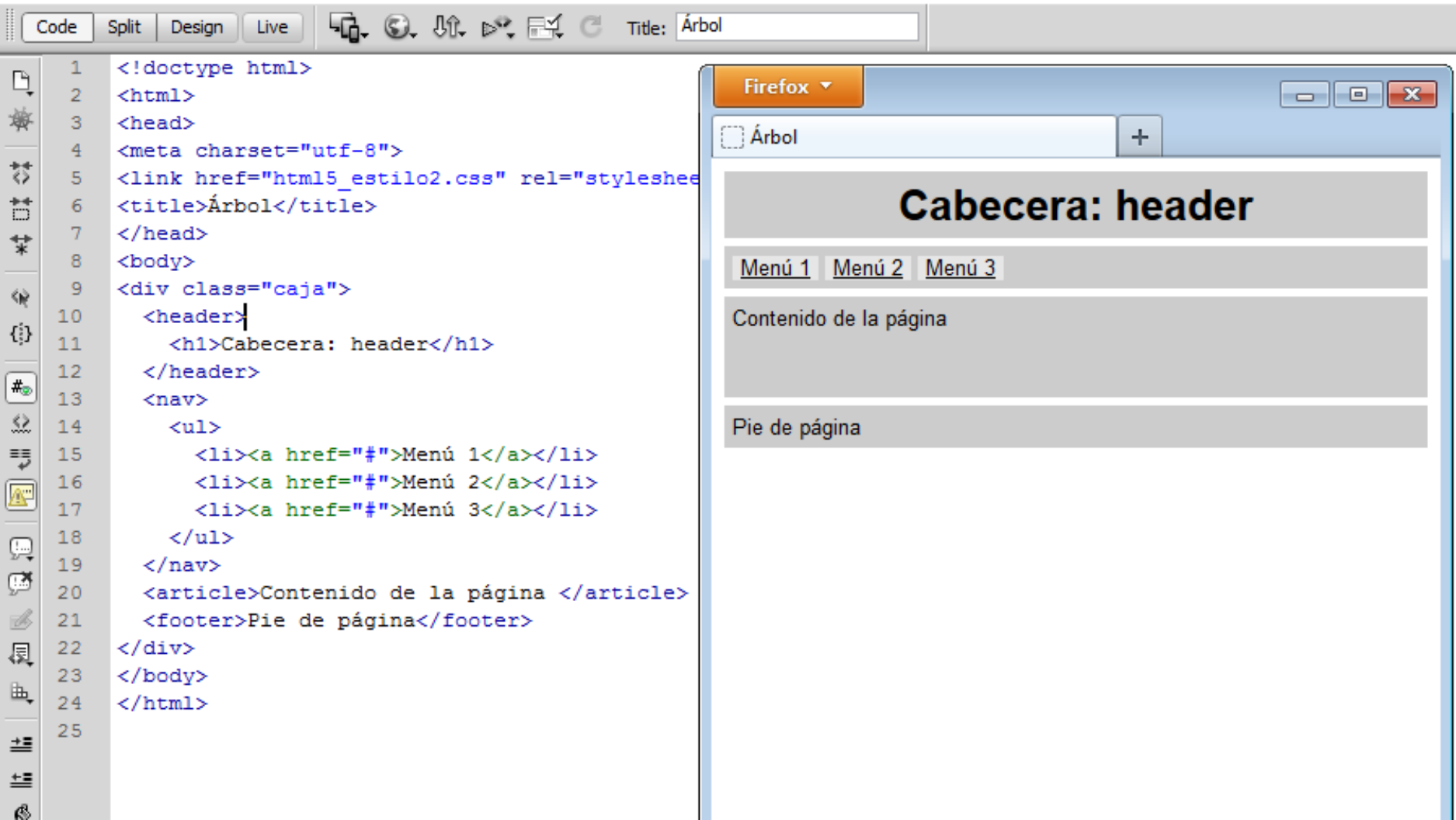
En este párrafo hay rexta marcado como *importante* y otro texto marcado como **muy importante**.

Este es el segundo párrafo.

strong {
font-weight: bold; }

em {
font-style: italic; }

Para entender como funcionan los selectores y la herencia CSS es necesario entender qué es el **árbol del documento**.



The image shows a web editor interface with two main panels. The left panel displays the HTML source code, and the right panel shows the rendered page in a Firefox browser window.

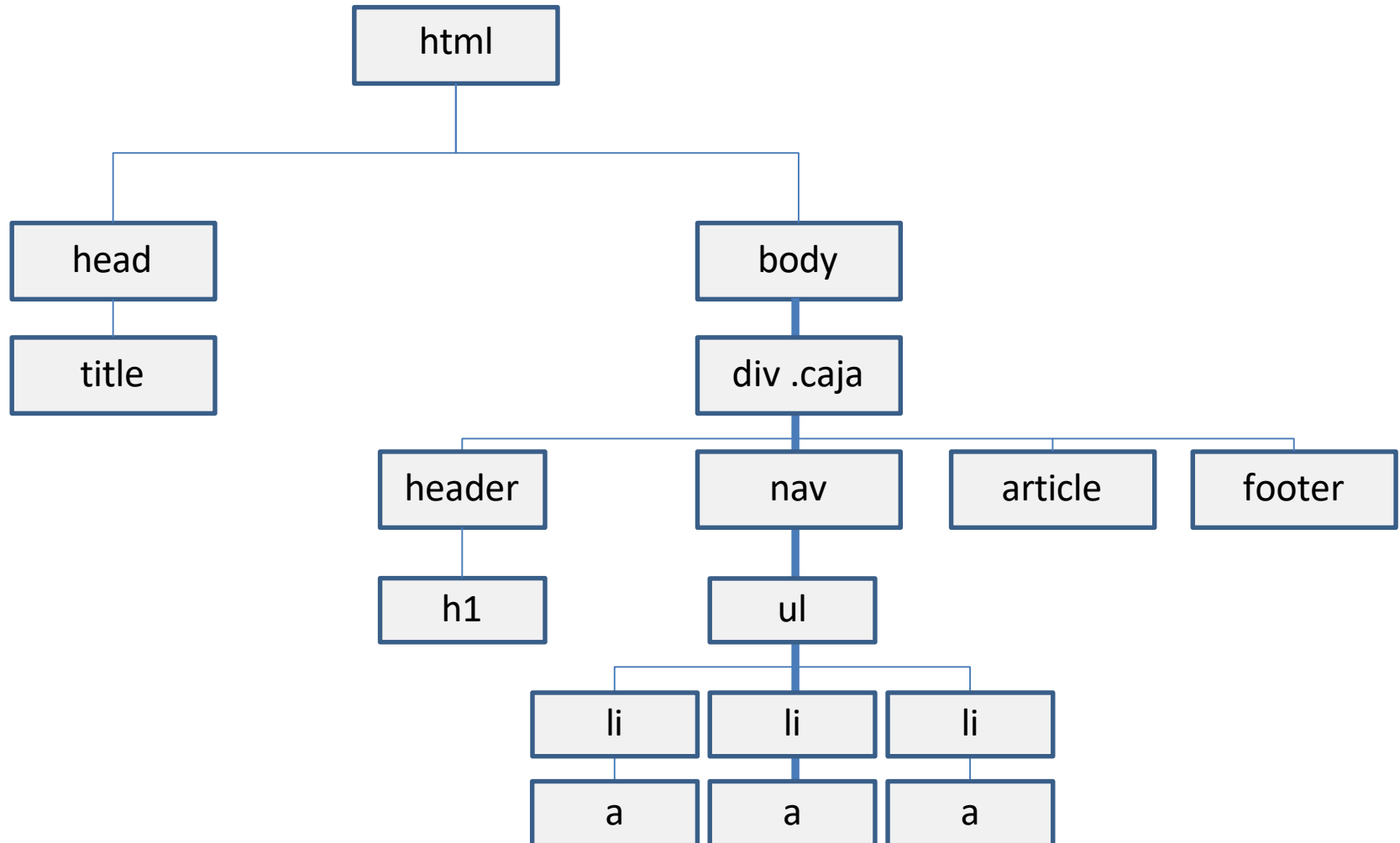
HTML Source Code (Left Panel):

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link href="html5_estilo2.css" rel="stylesheet">
6 <title>Árbol</title>
7 </head>
8 <body>
9 <div class="caja">
10   <header>
11     <h1>Cabecera: header</h1>
12   </header>
13   <nav>
14     <ul>
15       <li><a href="#">Menú 1</a></li>
16       <li><a href="#">Menú 2</a></li>
17       <li><a href="#">Menú 3</a></li>
18     </ul>
19   </nav>
20   <article>Contenido de la página </article>
21   <footer>Pie de página</footer>
22 </div>
23 </body>
24 </html>
25
```

Rendered Page (Right Panel - Firefox):

- Browser title: **Árbol**
- Page content structure:
 - Cabecera: header** (H1)
 - Menú 1 Menú 2 Menú 3 (Navigation links)
 - Contenido de la página (Main content area)
 - Pie de página (Footer)

el árbol del documento del ejemplo anterior:



Si definimos un estilo para la etiqueta **body** y esa propiedad es “heredable”, todos los elementos situados debajo en el árbol del documento la heredan.

...

```
<style type="text/css">  
    body{ color: red; }  
</style>
```

...

```
<body>  
    <h1>Título</h1>  
</body>
```

...

El orden de **prioridad en CSS** se establece según la siguiente lista (siendo 1 lo que más prioridad tiene, y 4 lo que menos)

1 = Estilo en línea

2 = IDs

3 = Clases

4 = Por defecto en marcas HTML

Además de esta regla, los selectores descendentes también añaden prioridad, es decir, **cuanto más específica sea una norma de estilo, más prioridad tendrá**

```
.caja nav ul {color: red;} -> más prioridad  
ul {color: orange;}
```


MUY COMPLETO, MUY ÚTIL COMO REFERENCIA Y PARA EL ESTUDIO:

<https://thevalleyofcode.com/css/>

Curso en MiriadaX de Desarrollo Frontend con HTML, CSS y JavaScript:

<https://miriadax.net/curso/desarrollo-frontend-con-html-css-y-javascript-3-a-edicion/>

W3C: Tutorial CSS

<https://www.w3schools.com/css/>

Kseso CSS: CSS básico: Cascada, especificidad y herencia

<http://ksesocss.blogspot.com/2012/05/css-basico-cascada-especificidad-y.html>

Dev.Opera: Recorriendo el árbol DOM

<http://dev.opera.com/articles/view/traversing-the-dom-es/>

LIBROS WEB: Árbol de nodo

http://librosweb.es/javascript/capitulo_5/arbol_de_nodos.html

UOC: Herencia y cascada

<http://mosaic.uoc.edu/ac/le/es/m6/ud2/>

LIBROSWEB: Colisiones de estilos

http://librosweb.es/css/capitulo_2/colisiones_de_estilos.html

CSS: Specificity Wars

http://www.stuffandnonsense.co.uk/archives/css_specificity_wars.html

CSS - Posicionamiento

CSS3 permite posicionar los **div** en la página mediante el uso de varias propiedades.

- **float y clear** permiten que los elementos floten a la derecha o a la izquierda
- **position**, que puede tener valor: static, absolute, relative o fixed

Esto es antiguo conviene empezar a utilizar flex y grid que son modernos y muy potentes

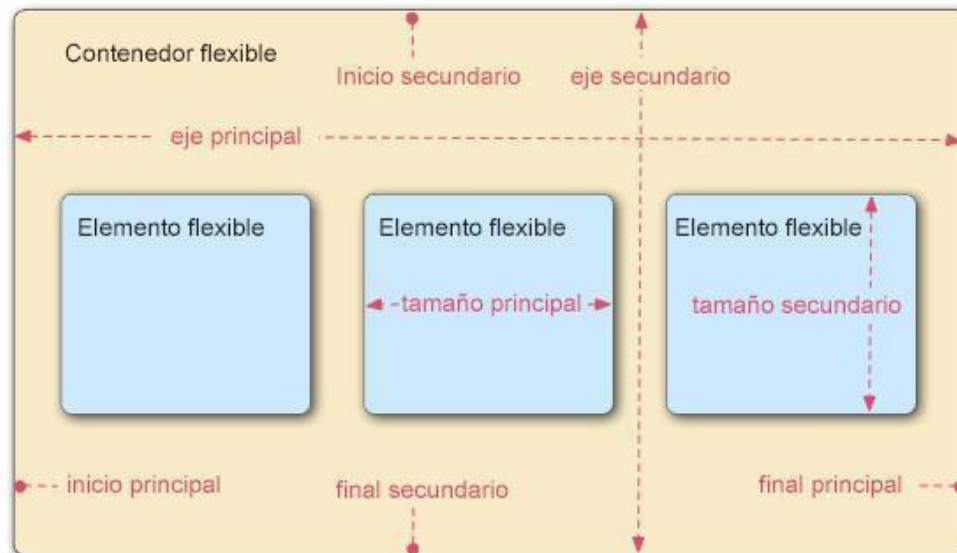
```
<style type="text/css">
div {
  float: left;
  padding: 5px;
  margin-right: 5px;
  margin-bottom: 5px;
  border: 1px solid #666;
}
</style>
```

```
<style type="text/css">
.caja1 {
  height: 100px;
  width: 150px;
  background-color: #FF9;
  float: left;
}
.caja2 {
  height: 100px;
  width: 150px;
  background-color: #CCF;
  position: absolute;
  left: 250px;
  top: 40px;
}
.caja3 {
  height: 100px;
  width: 150px;
  float: left;
  background-color: #CFC;
}
.caja {
  position: absolute;
  left: 100px;
  top: 100px;
}
.caja4 {
  height: 50px;
  width: 450px;
  background-color: #FC6;
  clear: both;
}
</style>
```

Las **cajas flexibles** se consiguen con un nuevo valor de la propiedad display, **(display: flex;)** de la caja padre.

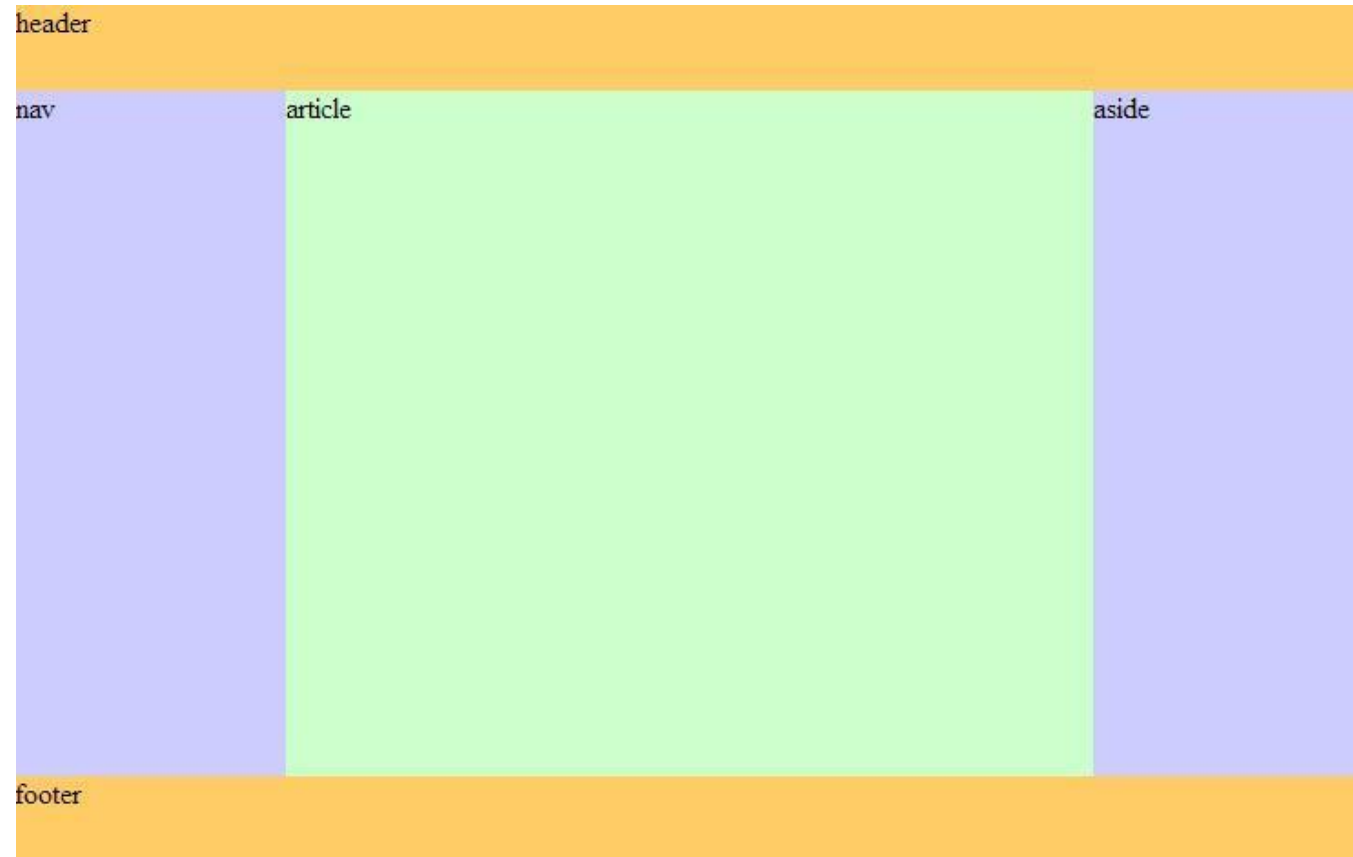
La orientación se define con **flex-direction** y puede ser horizontal o vertical, según sea fila o columna.

Los elementos flexibles tienen diferentes formas de alinearse y distribuirse **justify-content** y **align-items**.



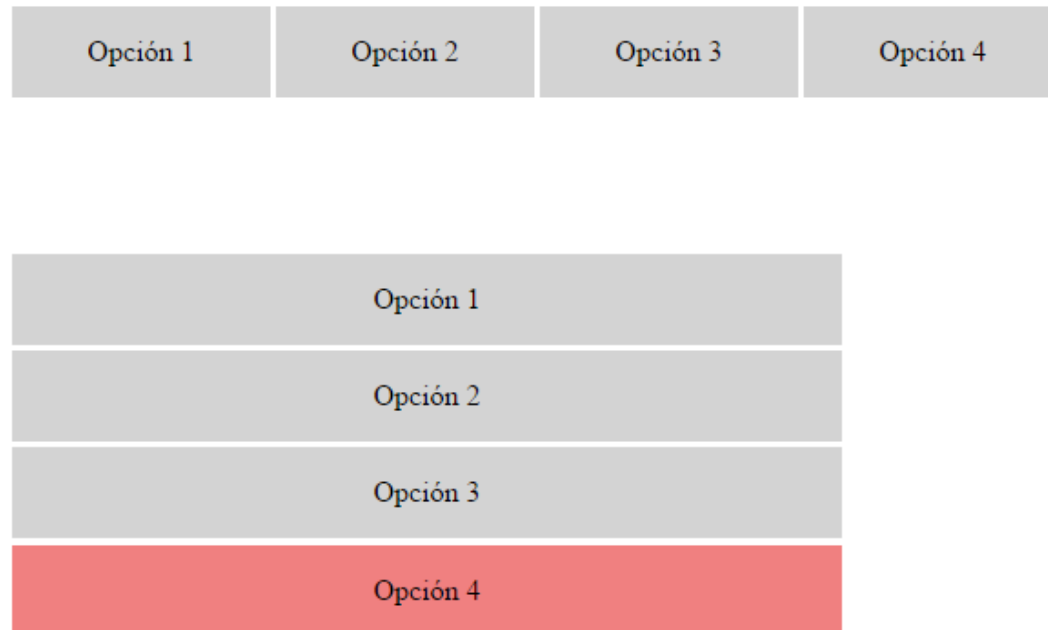
La propiedad de **cajas flexibles**, es la forma más sencilla de controlar el aspecto de las cajas div.

```
<style>
body, html {
  height: 100%;
  width: 100%;
  margin: 0px;
  padding: 0px;
}
#main {
  height: 80%;
  display: -webkit-flex;
  display: flex;
  flex-flow: row;
}
#main article {
  background: #CFC;
  flex: 3 1 60%;
  order: 2;
}
#main nav {
  background: #CCF;
  flex: 1 6 20%;
  order: 1;
}
#main aside {
  background: #CCF;
  flex: 1 6 20%;
  order: 3;
}
header, footer {
  display: block;
  min-height: 10%;
  background: #FC6;
}
</style>
```



Un menú vertical y horizontal con FLEX, tendría este formato.

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>MENÚ HORIZONTAL FLEX</title>
6 <style type="text/css">
7   ul {
8     list-style-type:none;
9     margin:0;
10    padding:0;
11    display:flex;
12    flex-direction: row; flex-direction: column;
13  }
14  li { flex-grow: 1; }
15  a {
16    display:block;
17    padding:1em;
18    background-color:LightGray ;
19    text-align:center;
20    margin-right:0.2em; margin-bottom:0.2em;
21    text-decoration:none;
22    color:#000000;
23  }
24  a:hover {background-color:LightCoral; }
25 </style>
26 </head>
27 <body>
28 <nav>
29   <ul>
30     <li><a href="#">Opción 1</a></li>
31     <li><a href="#">Opción 2</a></li>
32     <li><a href="#">Opción 3</a></li>
33     <li><a href="#">Opción 4</a></li>
34   </ul>
35 </nav>
36 </body>
```



Tutorial completo y sirve de referencia:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Flexbox Froggy Un juego para aprender CSS flexbox

<http://flexboxfroggy.com/#es>

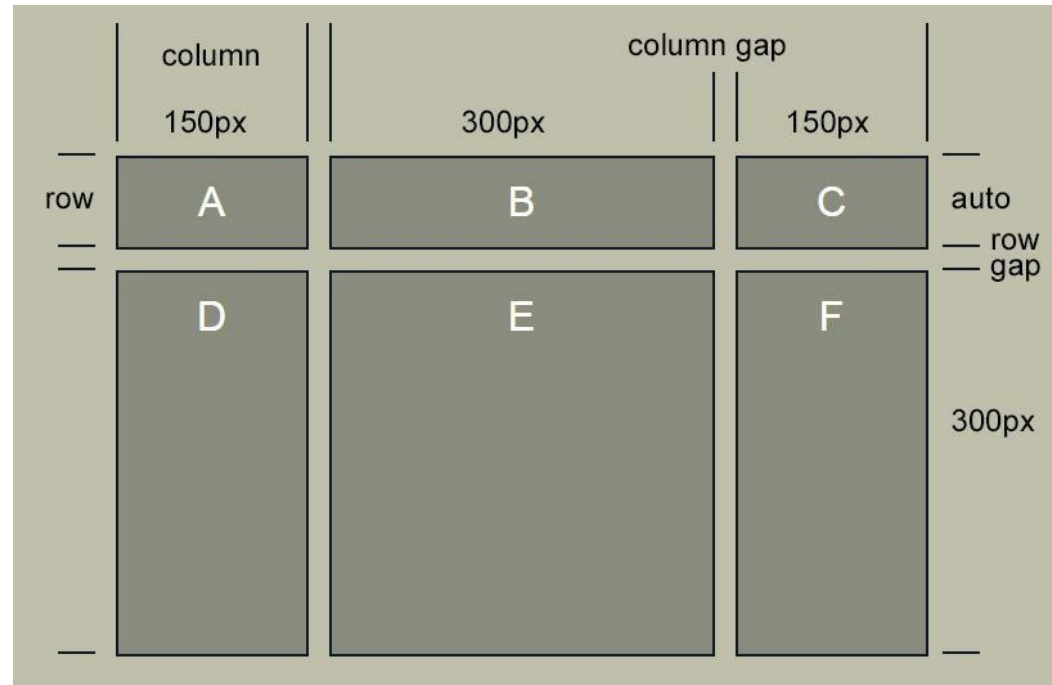
La nueva etiqueta **Grid**, es el mejor sistema para estructurar la Web mediante la creación de rejillas de composición, muchos framework ya utilizan sistemas de grid para estructurar la web y esta etiqueta CSS tiene importantes mejoras que permite disponer fácilmente los elementos en una página web.

HTML

```
<div class="container">  
  <div>A</div>  
  <div>B</div>  
  <div>C</div>  
  <div>D</div>  
  <div>E</div>  
  <div>F</div>  
</div>
```

CSS

```
.container {  
  display: grid;  
  grid-template-columns: 150px 300px 150px;  
  grid-template-rows: auto 300px;  
  grid-gap: 1rem;  
}
```



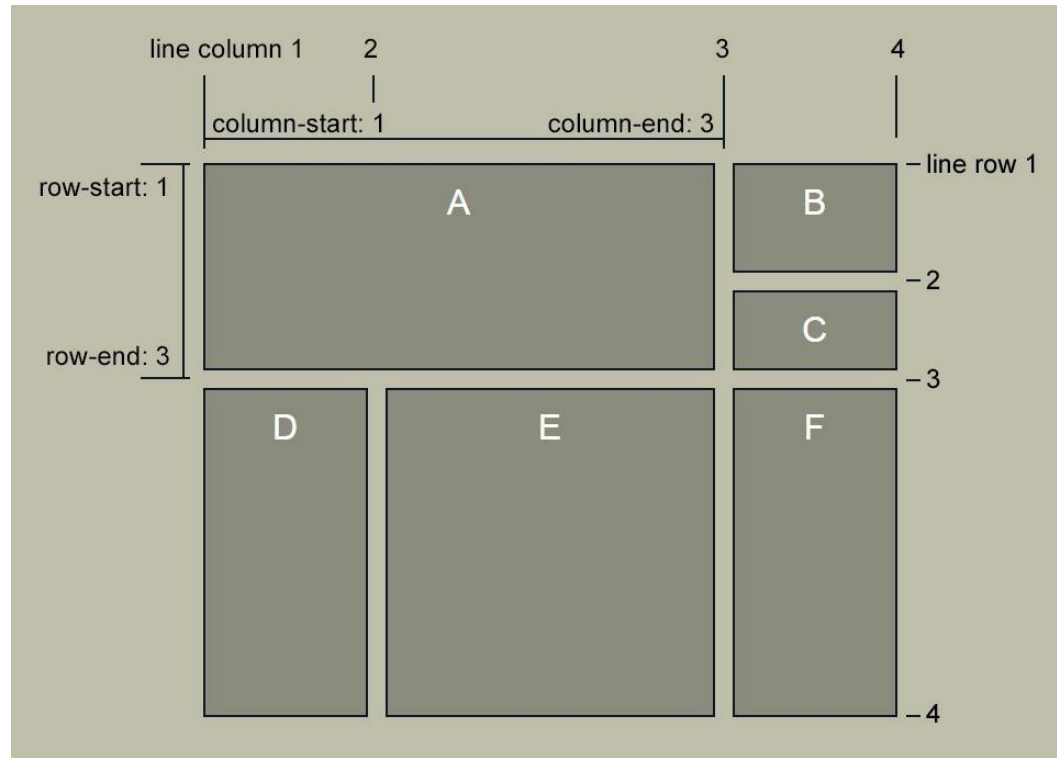
La línea es el separador horizontal (row) y vertical (column), se puede emplear para indicar el principio y final de una celda.

HTML

```
<div class="container">  
  <div class="item1">A</div>  
  <div>B</div>  
  <div>C</div>  
  <div>D</div>  
  <div>E</div>  
  <div>F</div>  
</div>
```

CSS

```
.container {  
  display: grid;  
  grid-template-columns: 150px 300px 150px;  
  grid-template-rows: 100px auto 300px;  
  grid-gap: 1rem;  
}  
.item1 {  
  grid-row-start: 1;  
  grid-row-end: 3;  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```



Tutorial completo y sirve de referencia:

<https://css-tricks.com/snippets/css/complete-guide-grid/>

Grid Garden Un juego para aprender CSS grid

<https://cssgridgarden.com/#es>

CSS-TRICKS: Auto-Sizing Columns in CSS Grid: `auto-fill` vs `auto-fit`

<https://css-tricks.com/auto-sizing-columns-css-grid-auto-fill-vs-auto-fit/>