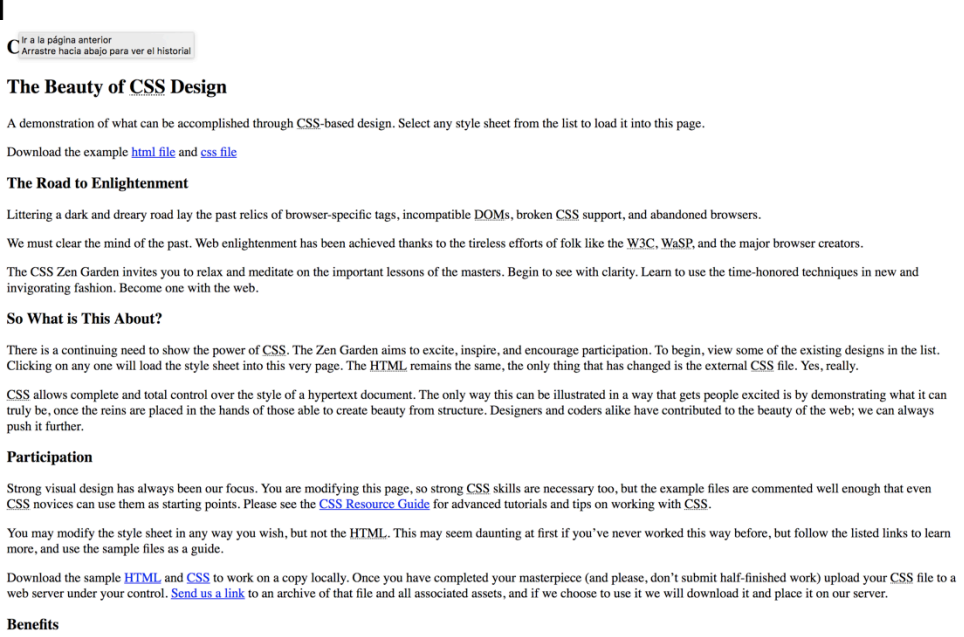
CSS

Estructura y herencia

e/s/d/
madrid

Qué es CSS

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.

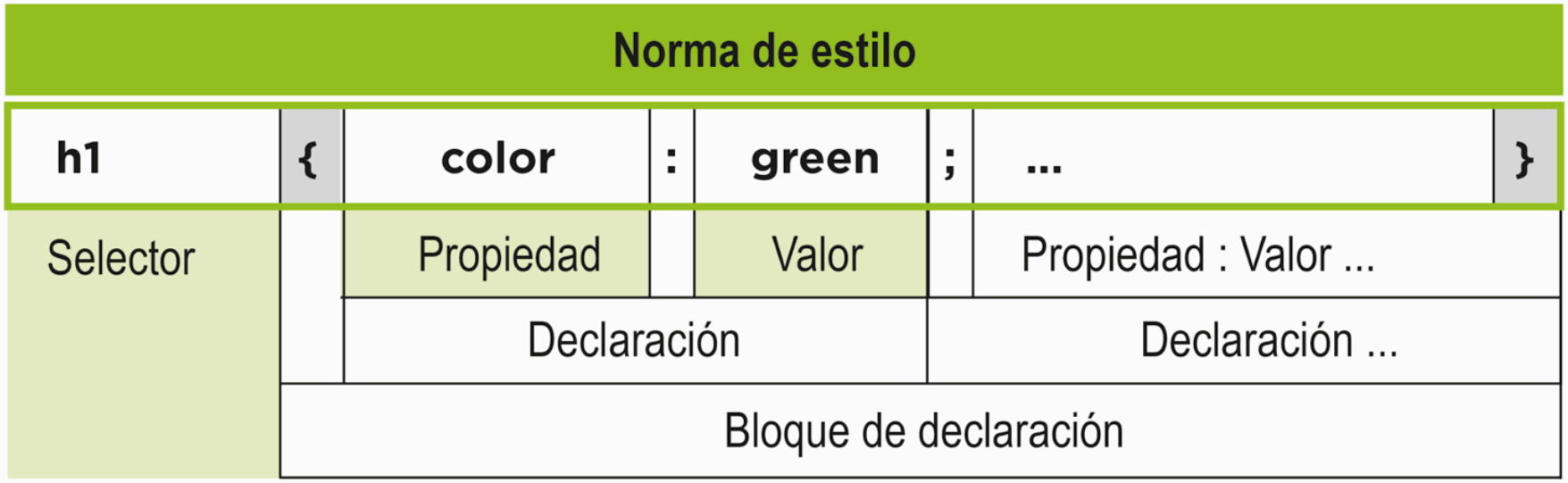


Proyecto **css Zen Garden**:
http://www.mezzoblue.com/zengarden/alldesigns/

e/s/d/ madrid

CSS - estructura

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



e/s/d/
madrid

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-style
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-left
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

e/s/d/
madrid

Selectores CSS

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;}
```

e/s/d/
madrid

Selectores CSS

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;}
```

e/s/d/
madrid

CSS - <style>

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>
...
```

CSS – en línea

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>
```

e/s/d/
madrid

CSS - <link>

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.

Código fuente estilo.css

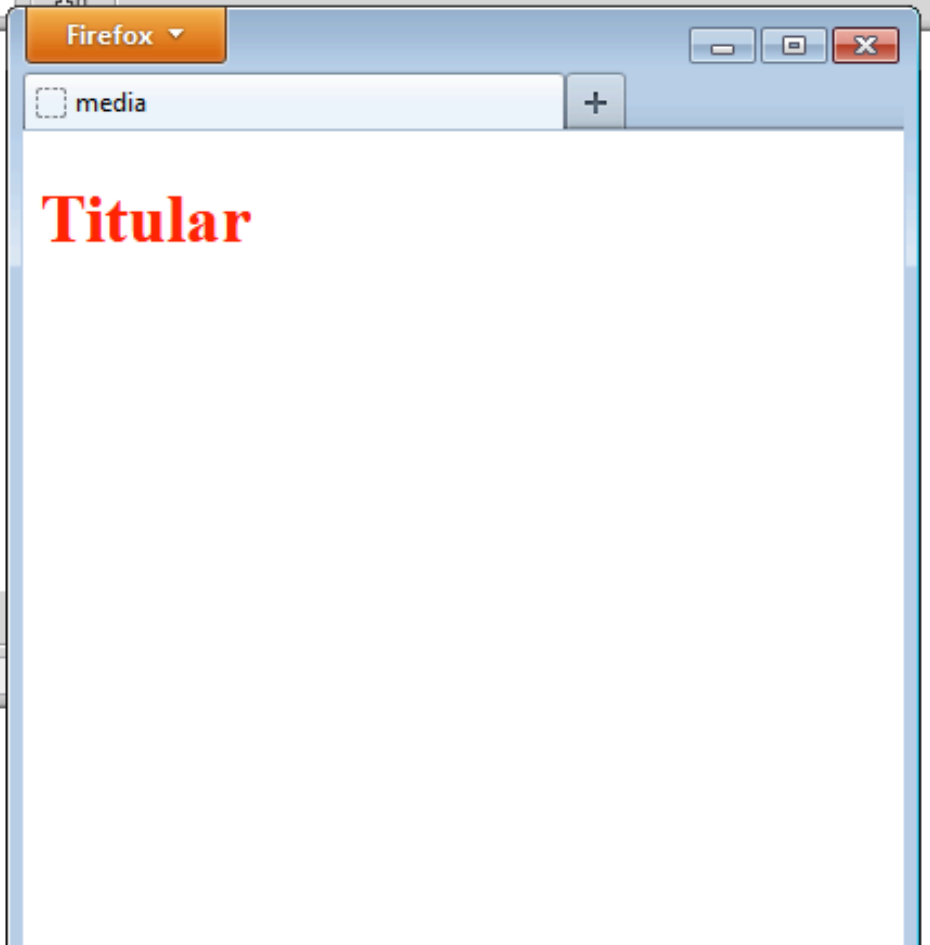
Código | Dividir | Diseño | En vivo | Título: media

```
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>
```

Código fuente estilo.css*

Código | Dividir | Diseño | En vivo | Título:

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



CSS - media

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/

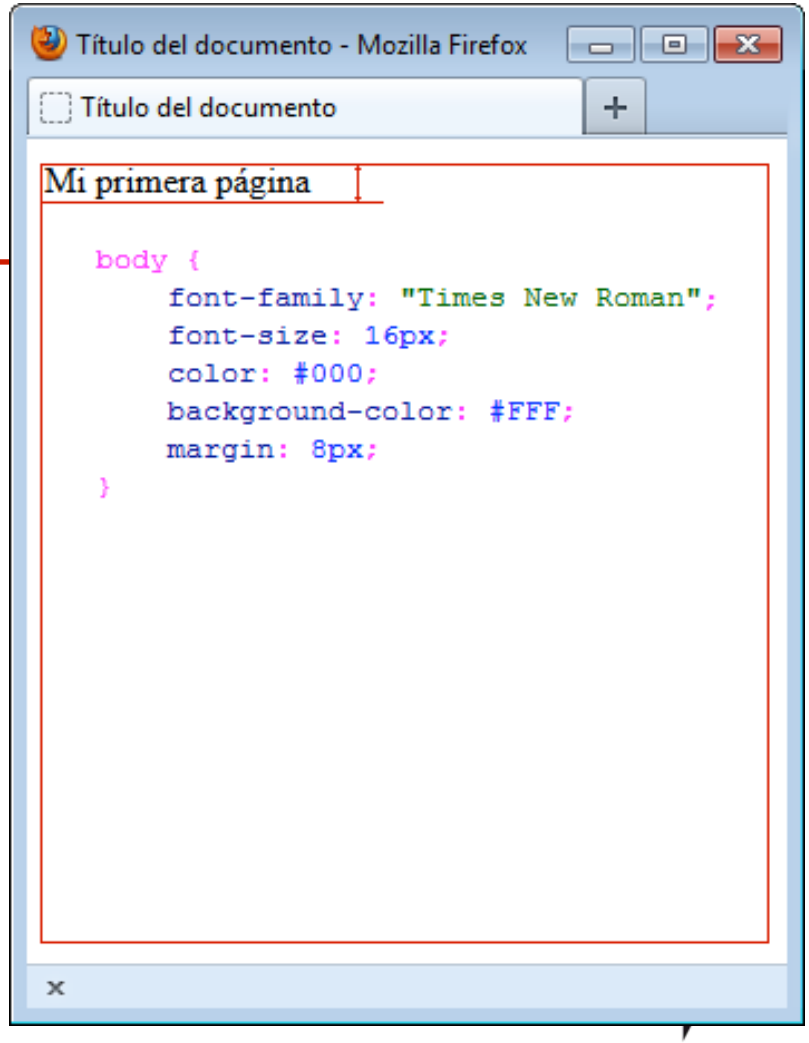
e/s/d/
madrid

CSS – por defecto

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
```



CSS – por defecto

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

```
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; }
```

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 rexto 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
```

Marcas

Titular nivel 1

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

Este es el segundo párrafo.

```
strong {
font-weight: bold; }
```

```
em {
font-style: italic; }
```

CSS

Más información valores por defecto:

W3C:
http://dev.w3.org/html5/markup/elements.html

Firefox:
https://dxr.mozilla.org/mozilla-central/source/layout/style/res/html.css

Chrome y Opera:
https://chromium.googlesource.com/chromium/blink/+/master/Source/core/css/html.css

Safari:
https://trac.webkit.org/browser/trunk/Source/WebCore/css/html.css

bit soft code: A look at CSS Resets in 2018
https://bitsofco.de/a-look-at-css-resets-in-2018/

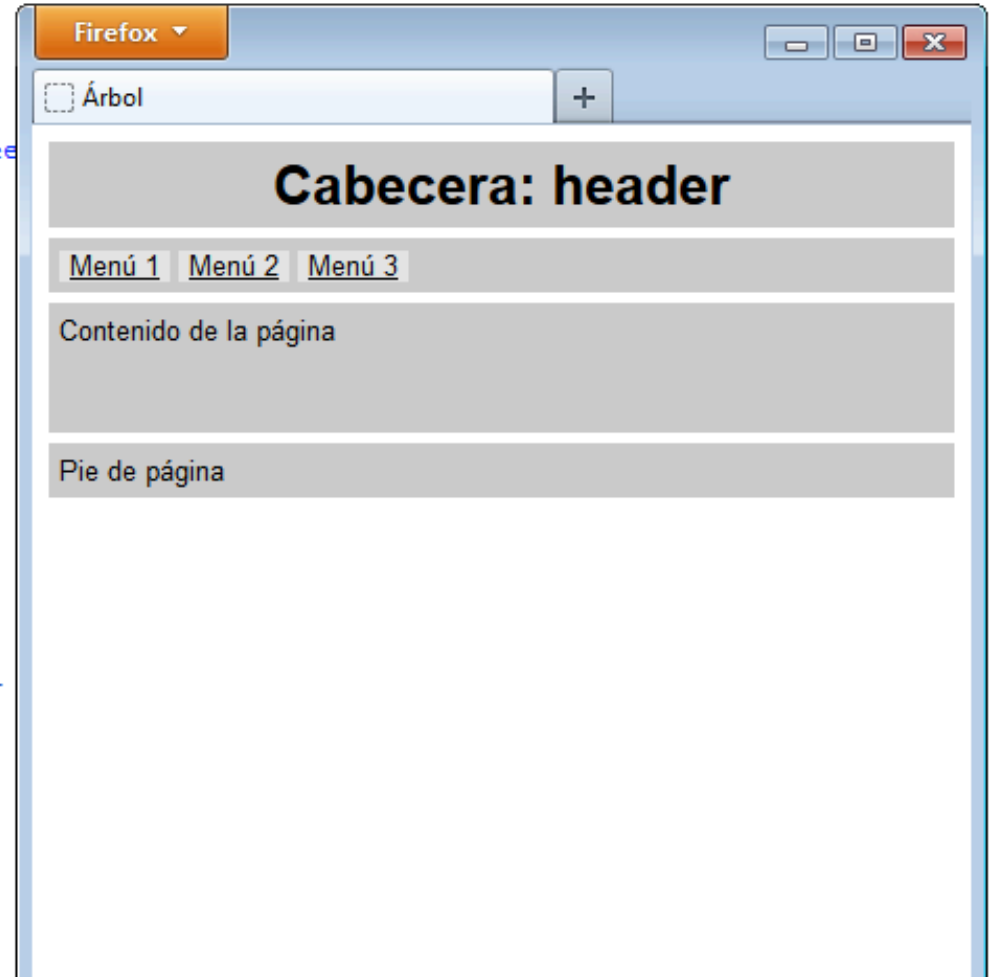
e/s/d/
madrid

Herencia CSS

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

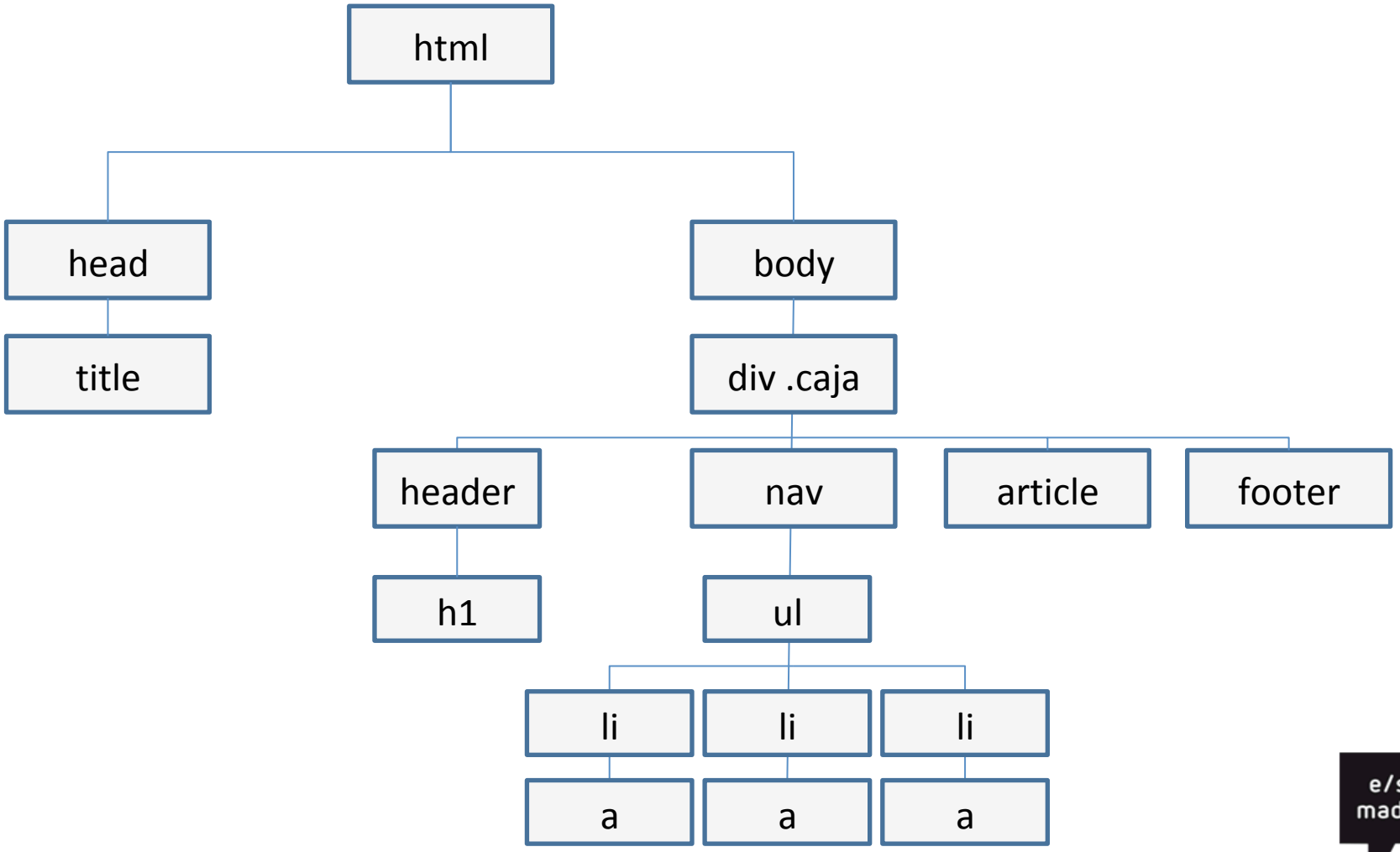
Code | Split | Design | Live | Title: Árbol

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<link href="html5_estilo2.css" rel="styleshee
<title>Árbol</title>
</head>
<body>
<div class="caja">
  <header>
    <h1>Cabecera: header</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">Menú 1</a></li>
      <li><a href="#">Menú 2</a></li>
      <li><a href="#">Menú 3</a></li>
    </ul>
  </nav>
  <article>Contenido de la página </article>
  <footer>Pie de página</footer>
</div>
</body>
</html>
```



Herencia CSS

el árbol del documento del ejemplo anterior:



e/s/d/
madrid

Herencia CSS

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>
...
```

Prioridad CSS

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 = 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**

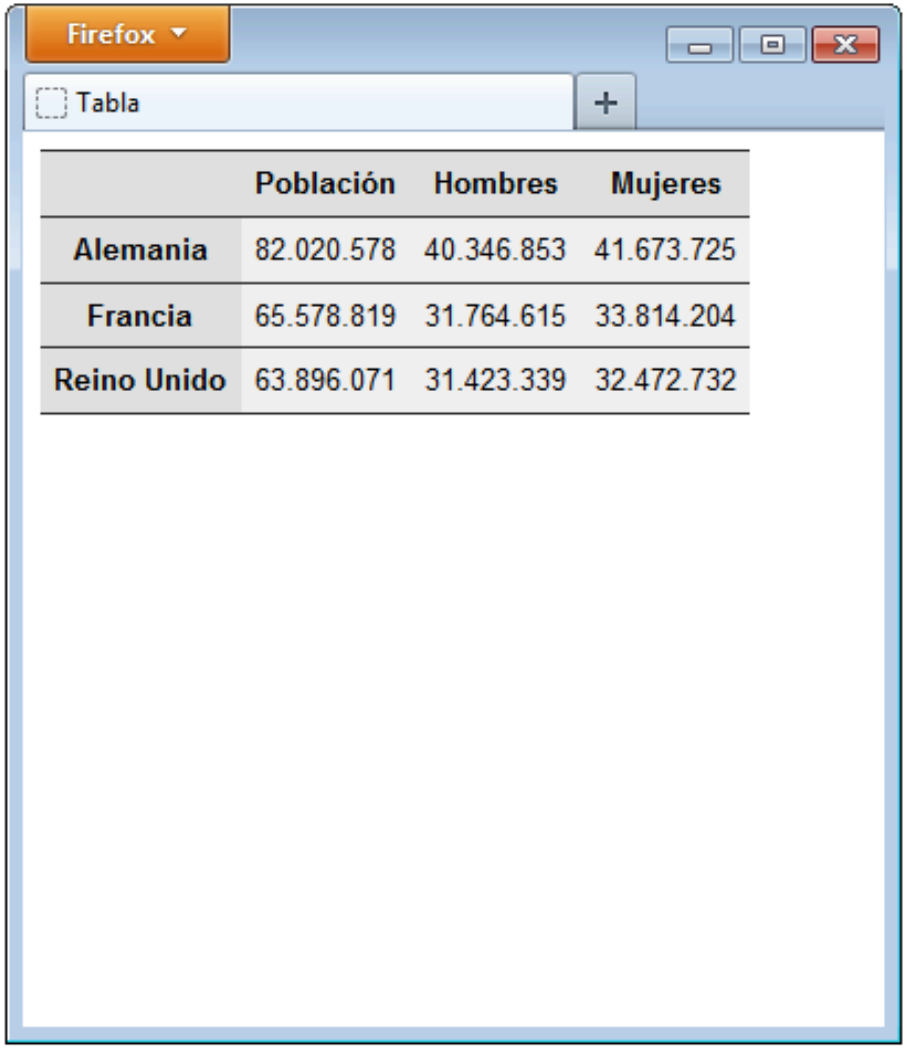
```
.caja nav ul {color: red;} -> más prioridad
ul {color: orange;}
```

e/s/d/
madrid

Tabla CSS

Código | Dividir | Diseño | En vivo | Título: Tabla

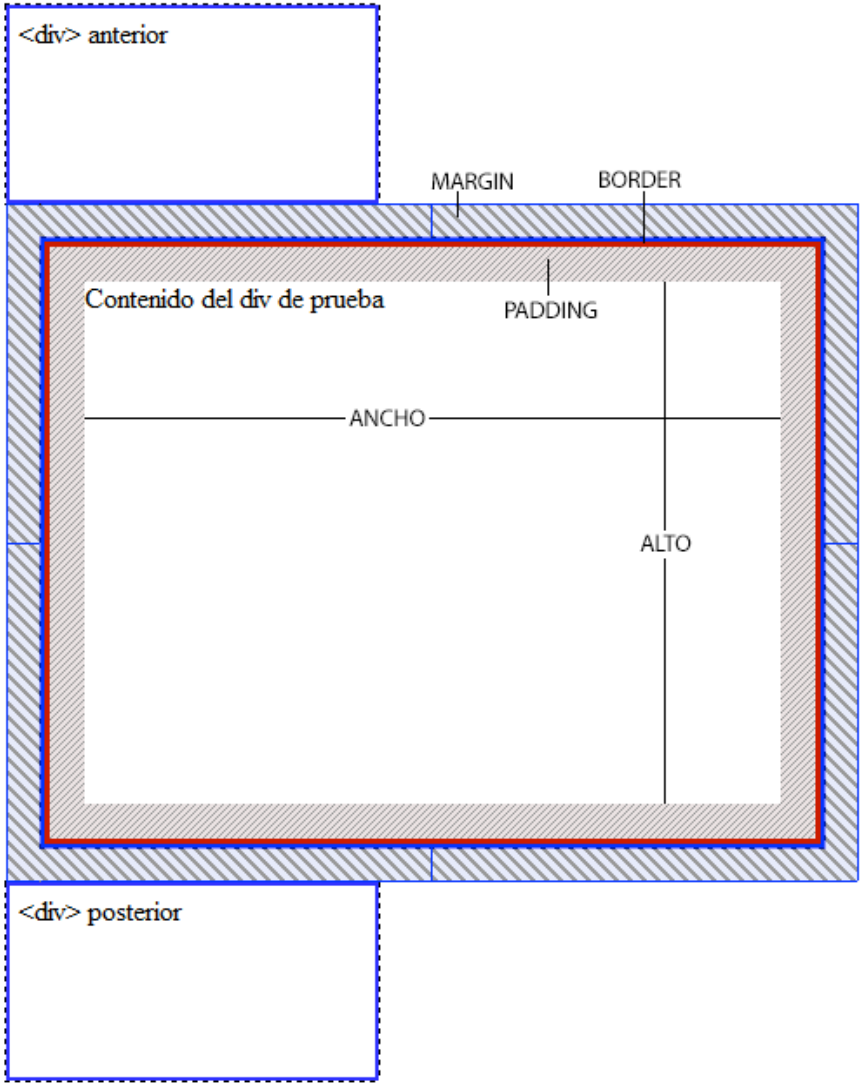
```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Tabla</title>
<style type="text/css">
table {
    padding: 0px;
    margin: 0px;
    border-spacing: 0px;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 0.8em;
    border-top-width: 1px;
    border-top-style: solid;
    border-top-color: #333;
}
th, td {
    padding: 0.5em;
    border-bottom-width: 1px;
    border-bottom-style: solid;
    border-bottom-color: #333;
}
th {
    background-color: #E2E2E2;
}
td {
    background-color: #F0F0F0;
}
```



	Población	Hombres	Mujeres
Alemania	82.020.578	40.346.853	41.673.725
Francia	65.578.819	31.764.615	33.814.204
Reino Unido	63.896.071	31.423.339	32.472.732

div CSS

Padding, margin y border




```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>div</title>
<style type="text/css">
.muestra {
    height: 300px;
    width: 400px;
    border: 5px solid #C00;
    margin: 20px;
    padding: 20px;

}
.anterior, .posterior {
    height: 100px;
    width: 200px;
    border: 2px solid #30F;
    padding: 5px;
}
</style>
</head>

<body>
<div class="anterior">&lt;div&gt; anterior</div>
<div class="muestra">Contenido del div de prueba</div>
<div class="posterior">&lt;div&gt; posterior</div>
</body>
</html>
```

e/s/d/
madrid

Recursos 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

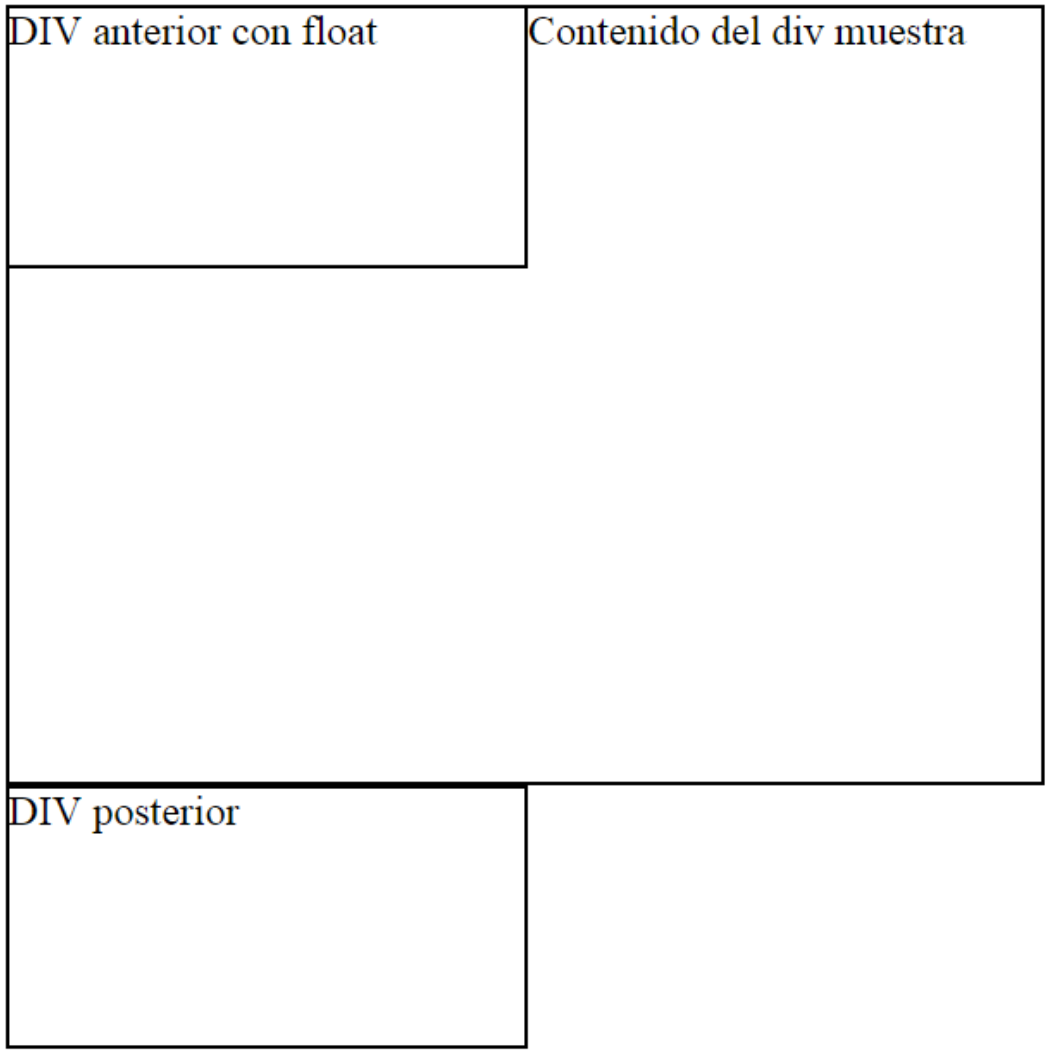
CSS: Specificity Wars
http://www.stuffandnonsense.co.uk/archives/css_specificity_wars.html

e/s/d/
madrid

e/s/d/
madrid

Estructura CSS - float y clear

CSS3 permite posicionar los **div** en la página, **float** y **clear**
Con **float** el div “flota” a una posición relativa.

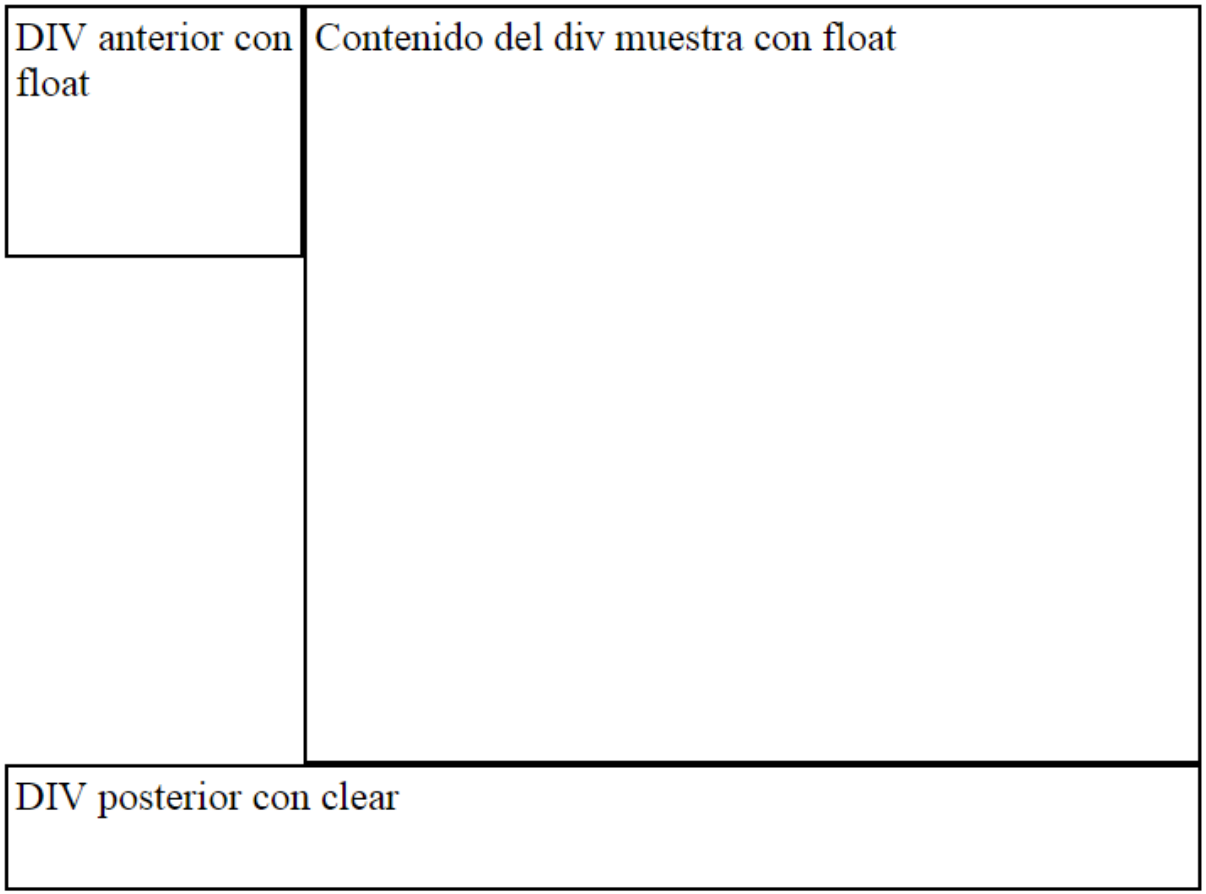


CSS

```
.muestra {
height: 300px;
width: 400px;
border:solid 1px black;
}
.anterior {
height: 100px;
width: 200px;
float: left;
border:solid 1px black;
}
.posterior {
height: 100px;
width: 200px;
border:solid 1px black;
}
```

Estructura CSS - float y clear

CSS3 permite posicionar los **div** en la página, **float** y **clear**
Con **clear** rompe el esquema del float.



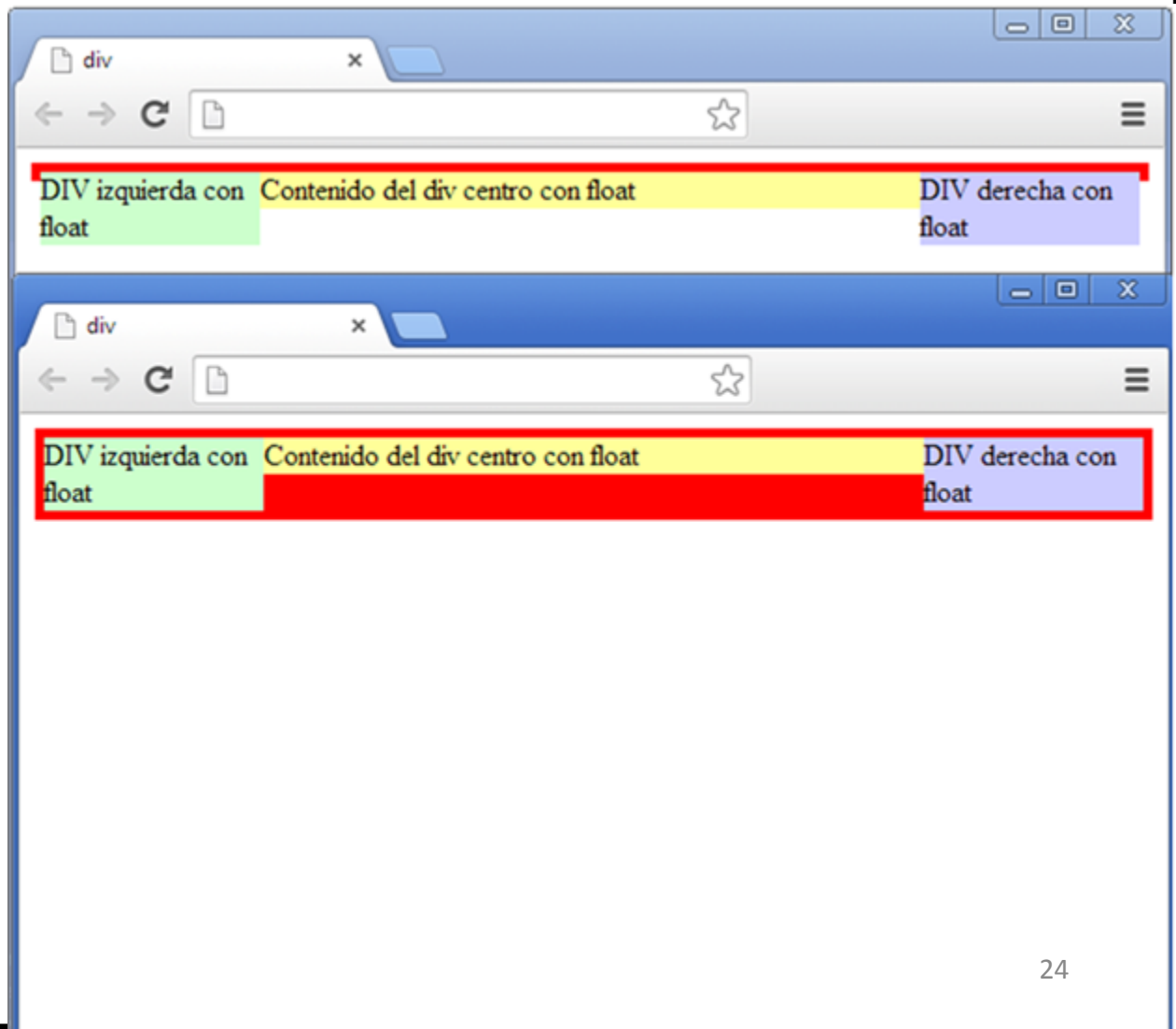
CSS

```
div {
border:solid 1px black;
box-sizing:border-box;
padding:0.2em;
}
.muestra {
height: 300px;
width: 60%;
float: left;
}
.anterior {
height: 100px;
width: 20%;
float: left;
}
.posterior {
height: 50px;
width: 80%;
clear: both;
}
```

Overflow:hidden

Uso de **float** y **overflow:hidden** para que la caja contenedora conozca la altura de sus bloques internos.

```
.caja{
        background-color: #F00;
        padding: 5px;
        overflow: hidden;
}
.caja .centro {
        width: 60%;
background-color: #FF9;
        float: left;
}
.caja .izq {
        width: 20%;
        float: left;
        background-color: #CFC;
}
.caja .dcha {
        width: 20%;
        background-color: #CCF;
        float: left;
}
```

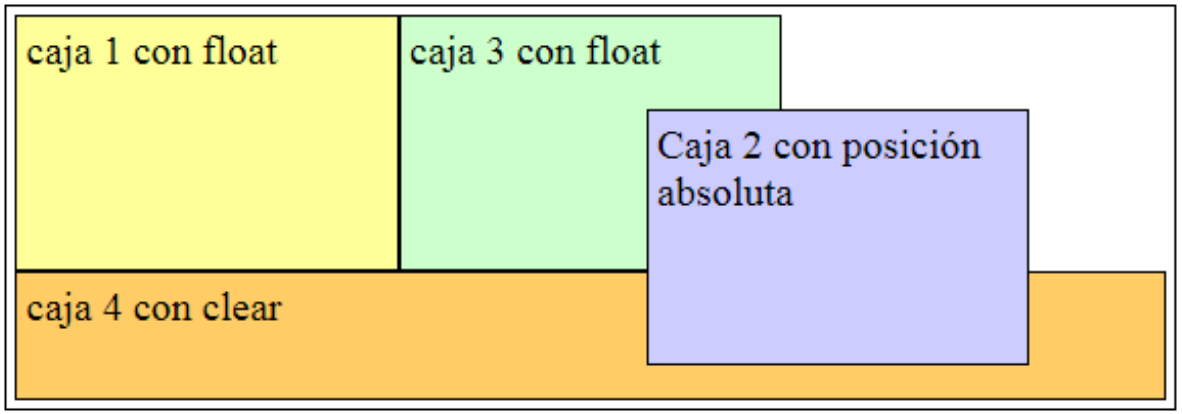


Estructura CSS - position

CSS3 permite posicionar los **div** en la página, mediante position, absoluta, relativa o fija.

Con posición absoluta el div sale del flujo del contenido.

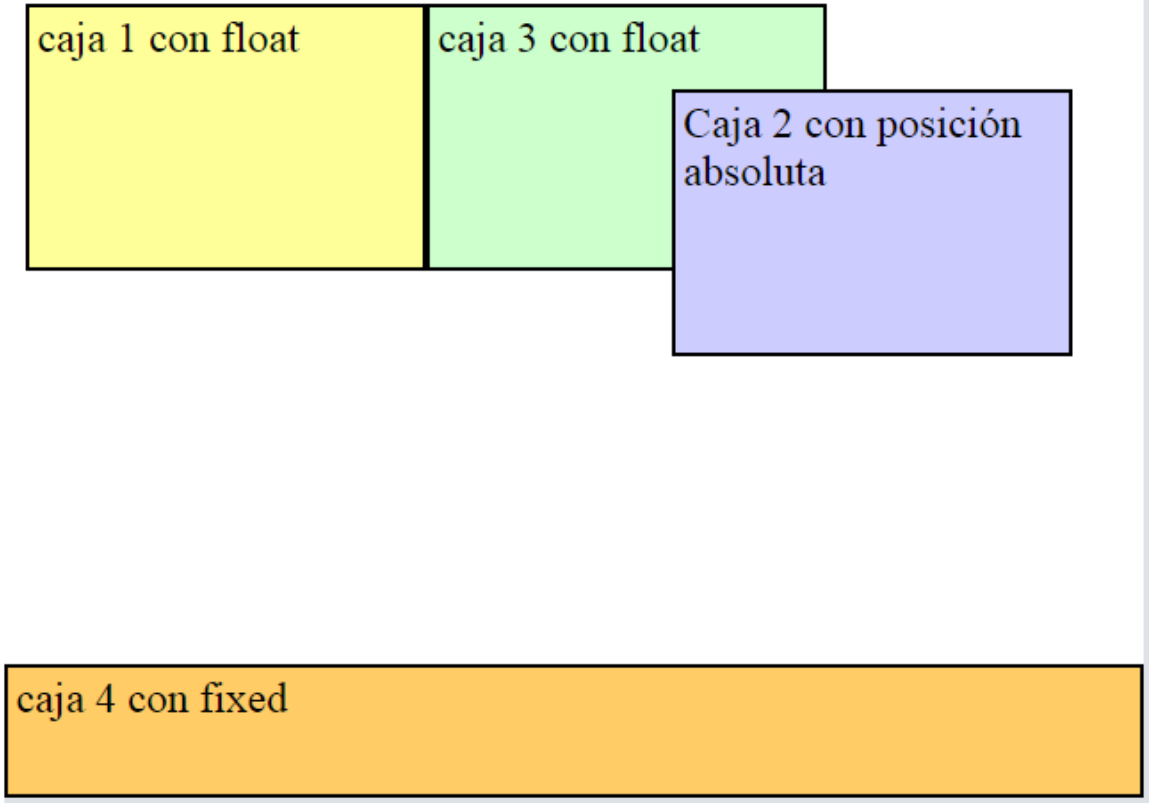
```
.caja {      position: absolute;
             left: 100px;
             top: 100px;    }
.caja1 {     height: 100px;
             width: 150px;
             background-color: #FF9;
             float: left;    }
.caja2 {     height: 100px;
             width: 150px;
             background-color: #CCF;
             position: absolute;
             left: 250px;
             top: 40px;    }
.caja3 {     height: 100px;
             width: 150px;
             float: left;
             background-color: #CFC;    }
.caja4 {     height: 50px;
             width: 450px;
             background-color: #FC6;
             clear: both;    }
```



Estructura CSS - position

Con posición **fija** el **div** no cambia de posición en el escalado del navegador o el scroll en la página.

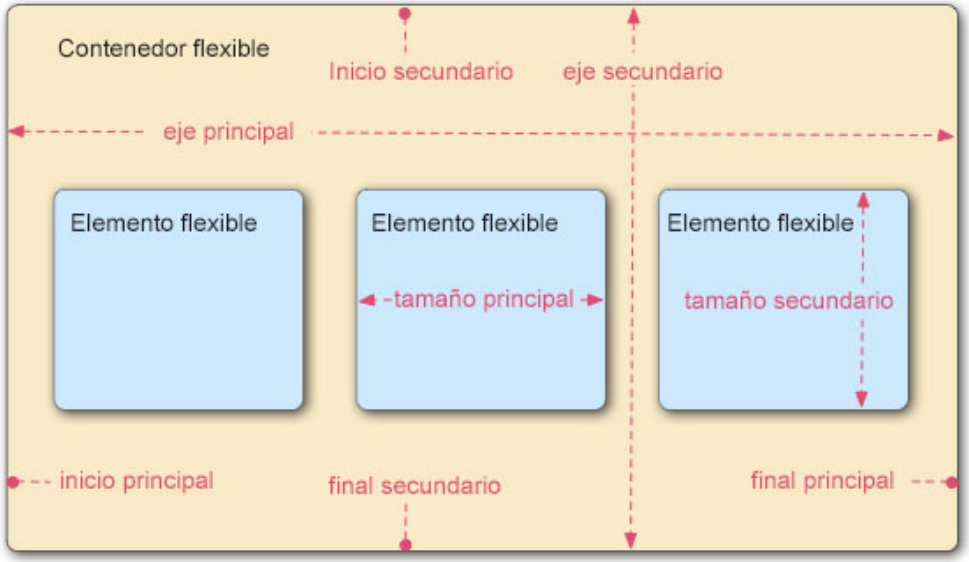
```
.caja1 {    height: 100px;
            width: 150px;
            background-color: #FF9;
            float: left; }
.caja2 {    height: 100px;
            width: 150px;
            background-color: #CCF;
            position: absolute;
            left: 250px;
            top: 40px; }
.caja3 {    height: 100px;
            width: 150px;
            float: left;
            background-color: #CFC;  }
.caja4 {    height: 50px;
            background-color: #FC6;
            clear: both;
            position: fixed;
            bottom: 0px;
            left: 0px;
            width: 100%;  }
```



Estructura CSS - FLEX

Las **cajas flexibles**, se consigue 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.

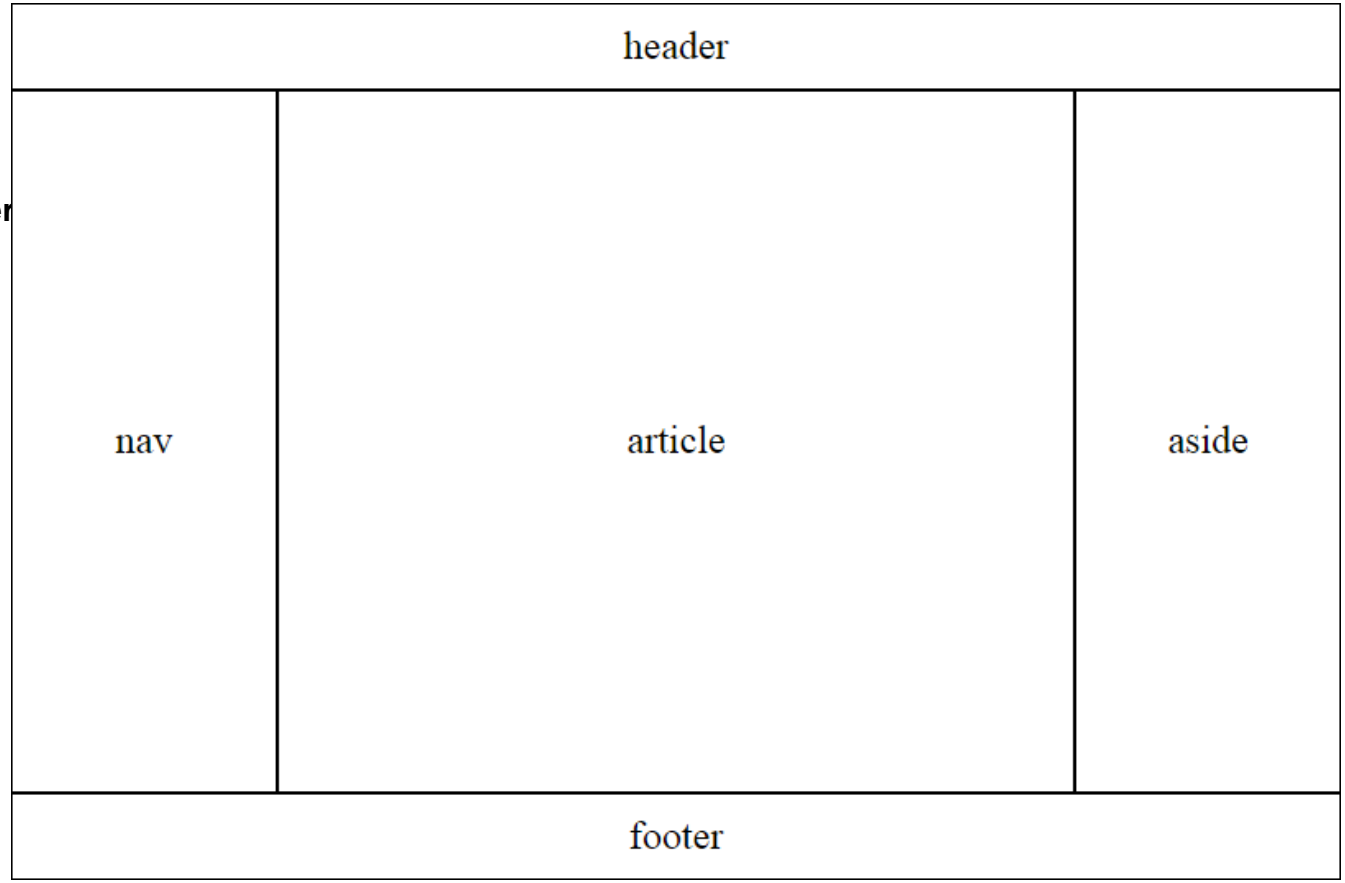
Cada uno de los elementos puede ordenarse o los diferentes modos crecer o Reducirse para ocupar el espacio.



Estructura CSS - FLEX

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

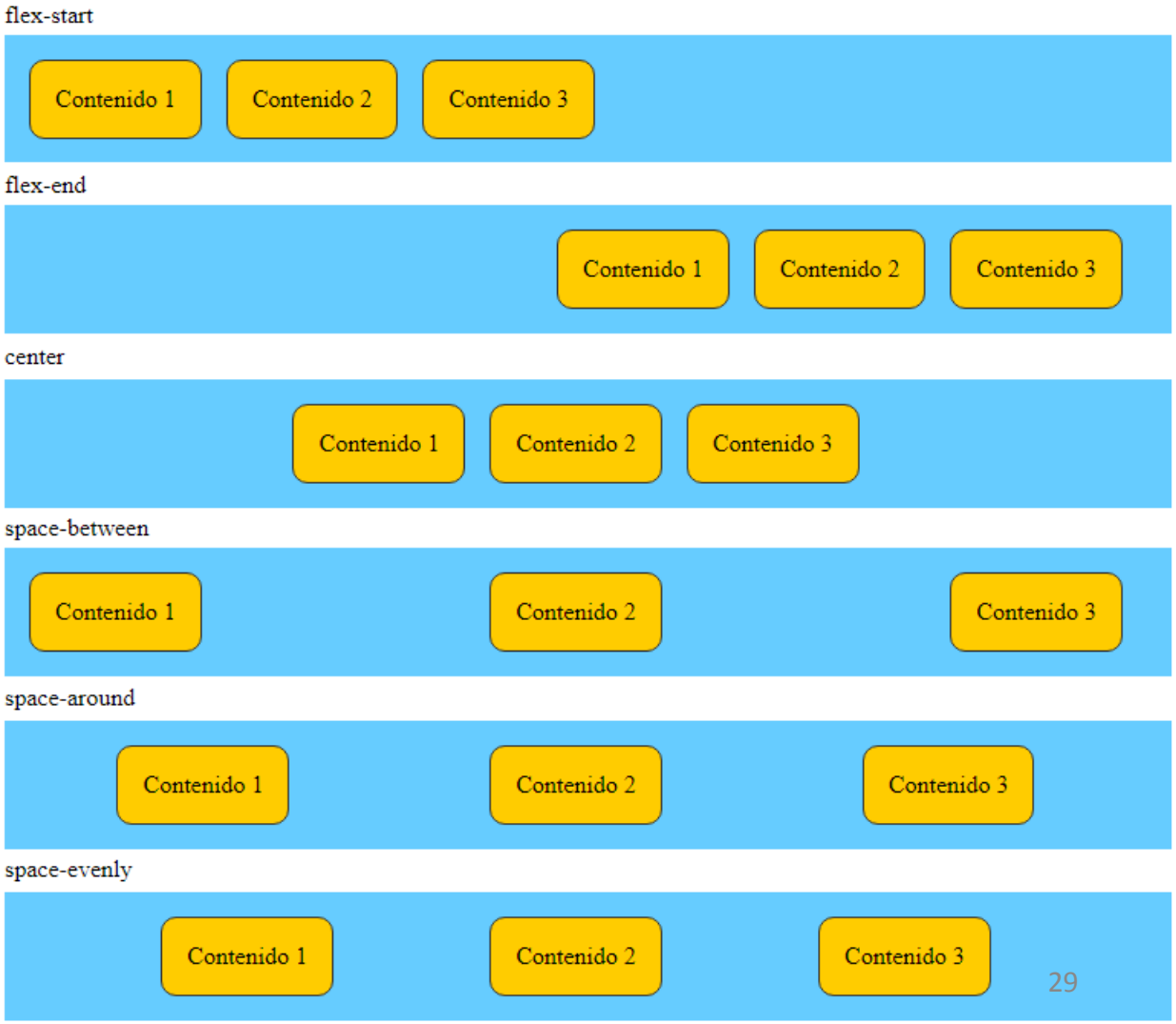
```
#main {
height: 80%;
 display: flex;
 flex-flow: row;
}
article, nav, aside, header, footer
display: flex;
justify-content: center;
align-items: center;
font-size:1.5em;
border:solid 1px black;
box-sizing:border-box;
}
#main article  {
flex: 3 1 60%;
order: 2;
}
#main nav  {
flex: 1 6 20%;
order: 1;
}
#main aside  {
flex: 1 6 20%;
order: 3;
}
header, footer  {
min-height: 10%;
}
```



Estructura CSS - FLEX – justify-content

Flex, facilita la alineación de los objetos en horizontal con **justify-content** y en vertical con **align-items**

```
.flex {          display: flex;
                 background-color: #6CF;
                 padding:1em;
}
.flex > div {    background-color: #FC0;
                 border: 1px solid #333;
                 margin-right:1em;
                 padding:1em;
                 border-radius:10px;
}
.start {         justify-content: flex-start;
}
.end {           justify-content: flex-end;
}
.center {        justify-content: center;
}
.between {       justify-content: space-
between;
}
.around {        justify-content: space-
around;
}
.evenly {        justify-content: space-
evenly;
}
```



Estructura CSS - FLEX – align-item

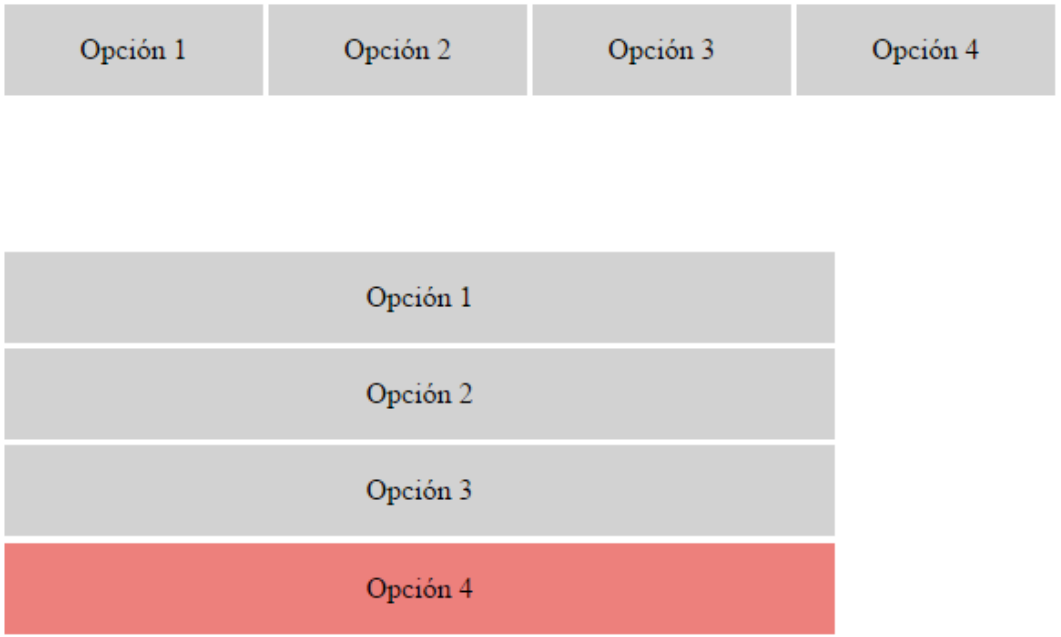
Flex, facilita la alineación de los objetos en horizontal con **justify-content** y en vertical con **align-items**

```
.flex {
        display: flex;
        background-color: #6CF;
        padding:1em;
}
.flex > div {
        background-color: #FC0;
        border: 1px solid #333;
        margin-right:1em;
        padding:1em;
        border-radius:10px;
}
img {       vertical-align:bottom;
}
.start {    align-items: flex-start;
}
.end {      align-items: flex-end;
}
.center {   align-items: center;
}
.baseline { align-items: baseline;
}
.stretch {  align-items: stretch;
}
```

Estructura Menú Flex

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

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>MENÚ HORIZONTAL FLEX</title>
<style type="text/css">
ul {
    list-style-type:none;
    margin:0;
    padding:0;
    display:flex;
    flex-direction: row; flex-direction: column;
}
li { flex-grow: 1; }
a {
    display:block;
    padding:1em;
    background-color:LightGray ;
    text-align:center;
    margin-right:0.2em;  margin-bottom:0.2em;
    text-decoration:none;
    color:#000000;
}
a:hover {background-color:LightCoral; }
</style>
</head>
<body>
<nav>
  <ul>
    <li><a href="#">Opción 1</a></li>
    <li><a href="#">Opción 2</a></li>
    <li><a href="#">Opción 3</a></li>
    <li><a href="#">Opción 4</a></li>
  </ul>
</nav>
</body>
```



Recursos CSS FLEX

MDN Usando las cajas flexibles CSS
https://developer.mozilla.org/es/docs/Web/Guide/CSS/Cajas_flexibles

CSS-TRICKS A Complete Guide to Flexbox
http://css-tricks.com/snippets/css/a-guide-to-flexbox/

Flexbox Froggy Un juego para aprender CSS flexbox
http://flexboxfroggy.com/#es

Estructura CSS - GRID

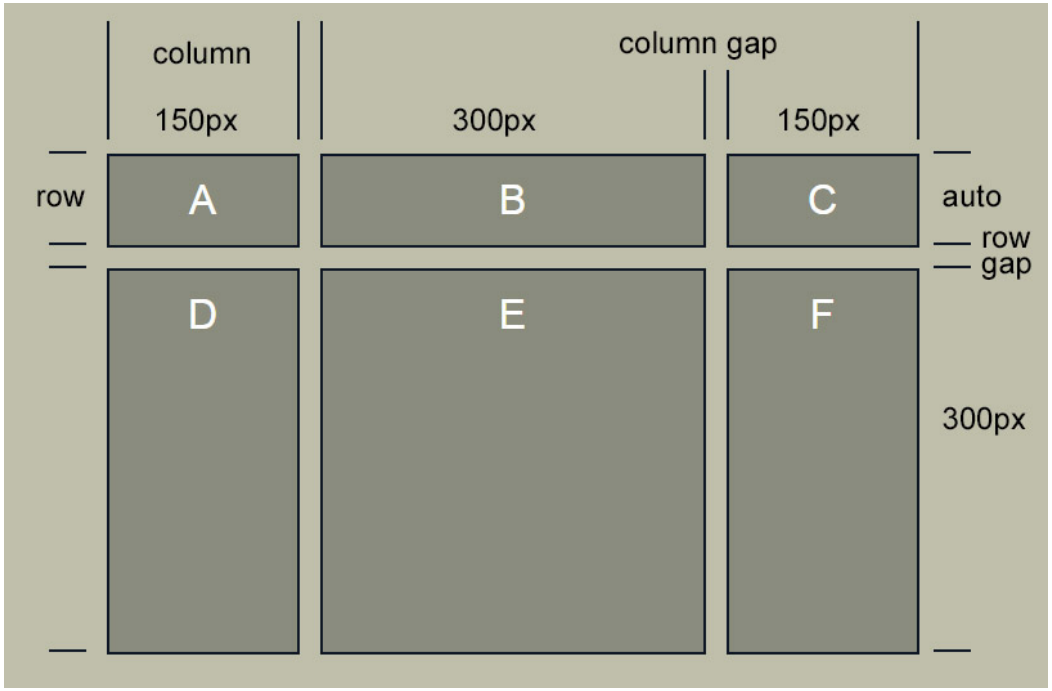
La nueva etiqueta de display **Grid**, es el mejor sistema para estructurar la Web mediante la creación de rejillas de composición, muchos framework ya utilizan sistemas de grid, esta etiqueta CSS tiene importantes mejoras.

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;
}
```



Estructura CSS - GRID

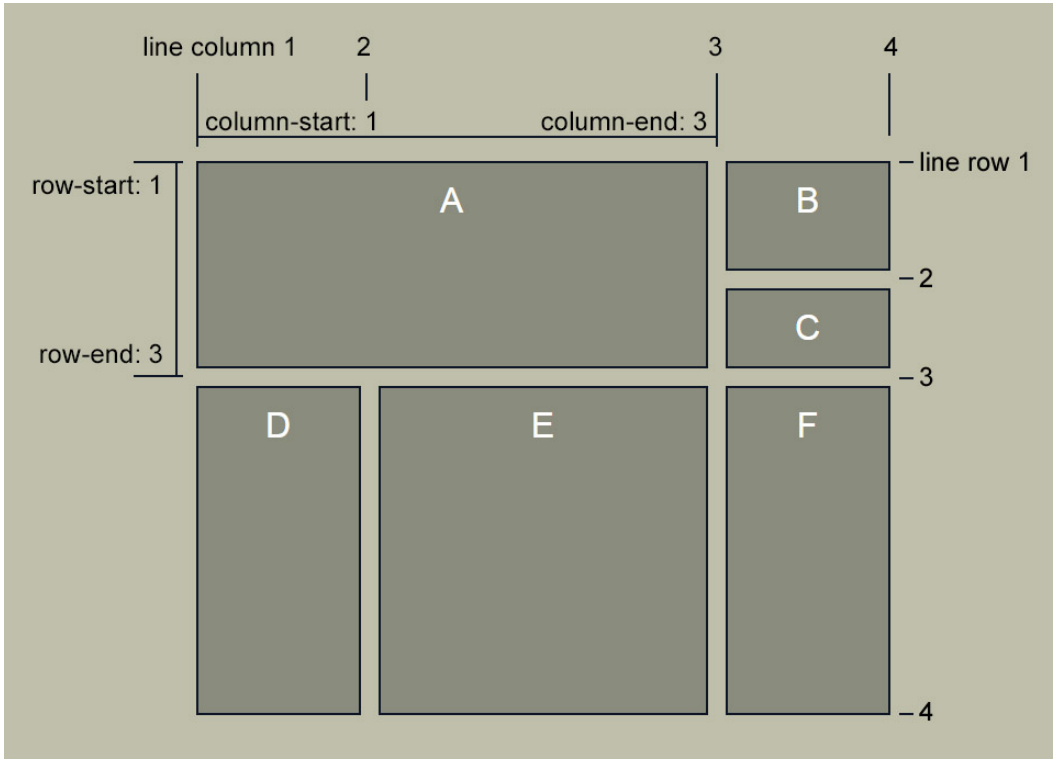
La línea es el separador horizontal (row) y vertical (column), se puede emplear para indicar el principio y final de una celda.

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;
}
```



Estructura CSS - GRID

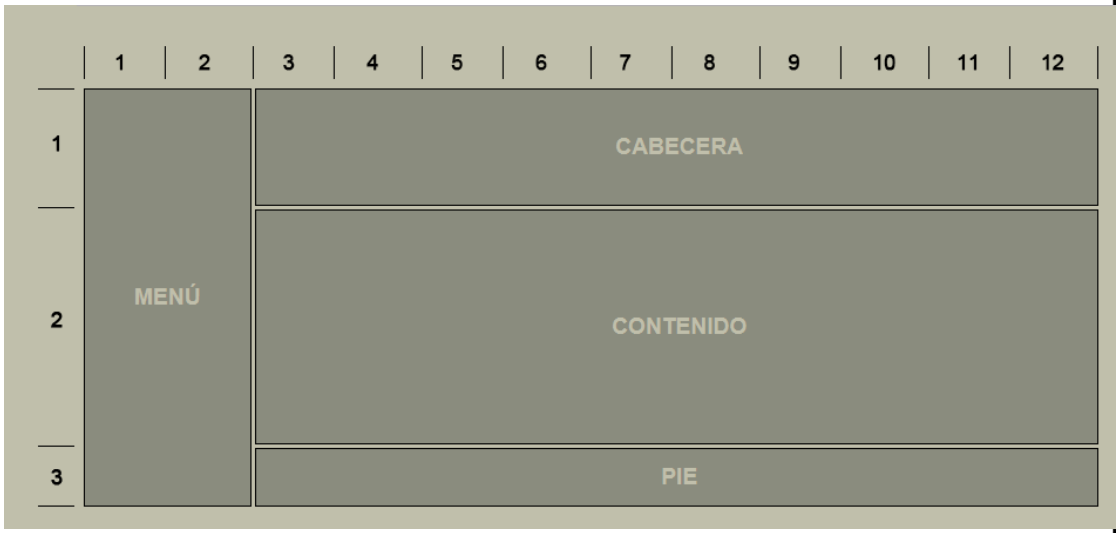
Podemos trabajar con una cuadrícula y asociar cada zona a un área, que puede abarcar varias celdas.

HTML

```
<div class="container">
 <div class="header">CABECERA</div>
 <div class="menu">MENÚ</div>
 <div class="content">CONTENIDO</div>
 <div class="footer">PIE</div>
</div>
```

CSS

```
.container {
   display: grid;
   grid-gap: 3px;
   grid-template-columns: repeat(12, 1fr);
   grid-template-rows: 100px 200px 50px;
   grid-template-areas:
      "m m h h h h h h h h h h"
      "m m c c c c c c c c c c"
      "m m f f f f f f f f f f";
}
 .header { grid-area: h;}
.menu { grid-area: m;}
.content { grid-area: c;}
.footer { grid-area: f;}
```



Estructura CSS - GRID

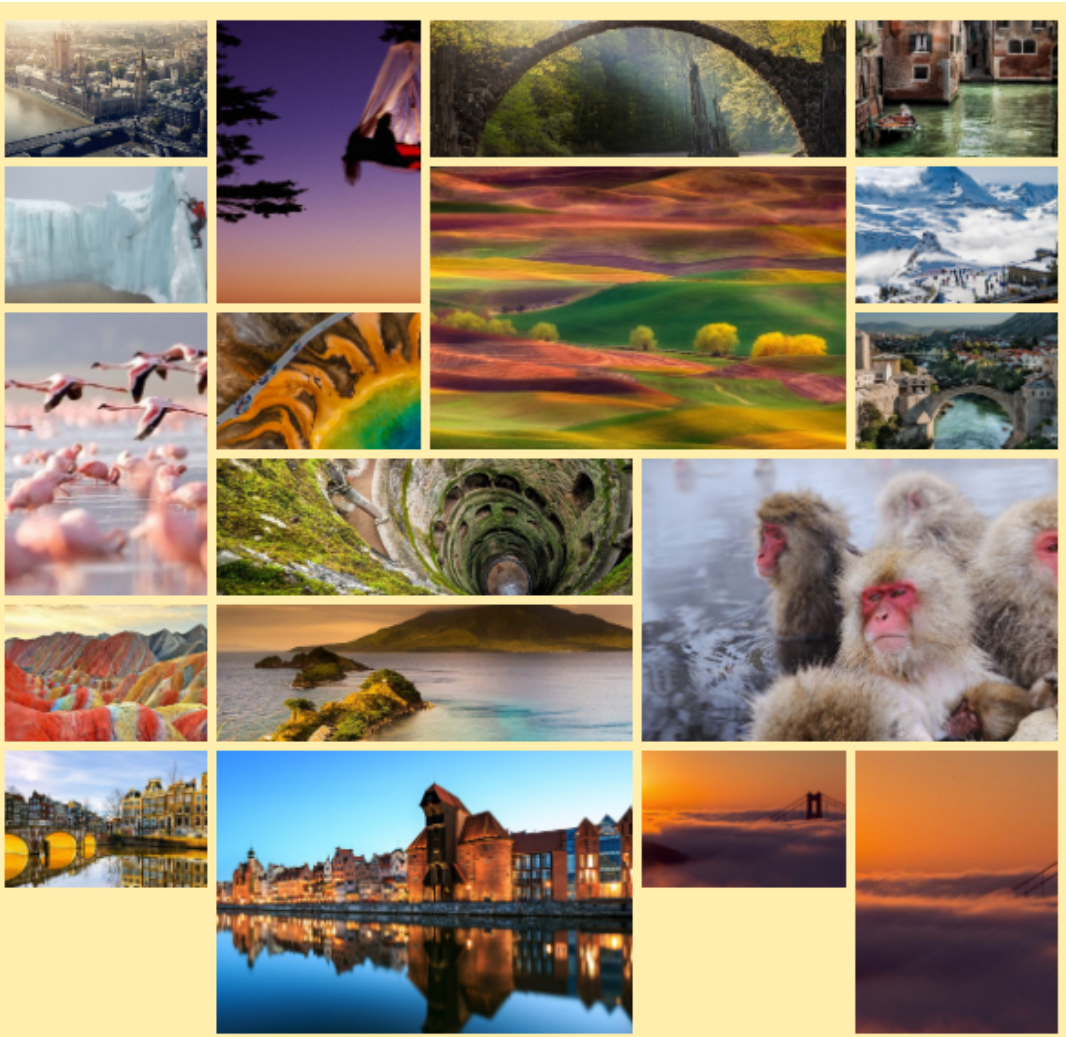
grid-auto-flow, controla la posición automática de las cajas contenidas en un grid.

HTML

```
<div class="container">
<div><img src="img… /></div><div
class="vertical"><img … /></div>
<div class="horizontal"><img… /></div>
<div class="big"><img … /></div>
… …
</div>
```

CSS

```
.container {
display: grid;
grid-gap: 5px;
grid-template-columns: repeat(auto-fit,
minmax(100px, 1fr));
grid-auto-rows: 75px;
grid-auto-flow: dense;
}
.horizontal { grid-column: span 2;  }
.vertical { grid-row: span 2; }
.big {
    grid-column: span 2;
    grid-row: span 2;
}
```



Estructura – menú GRID

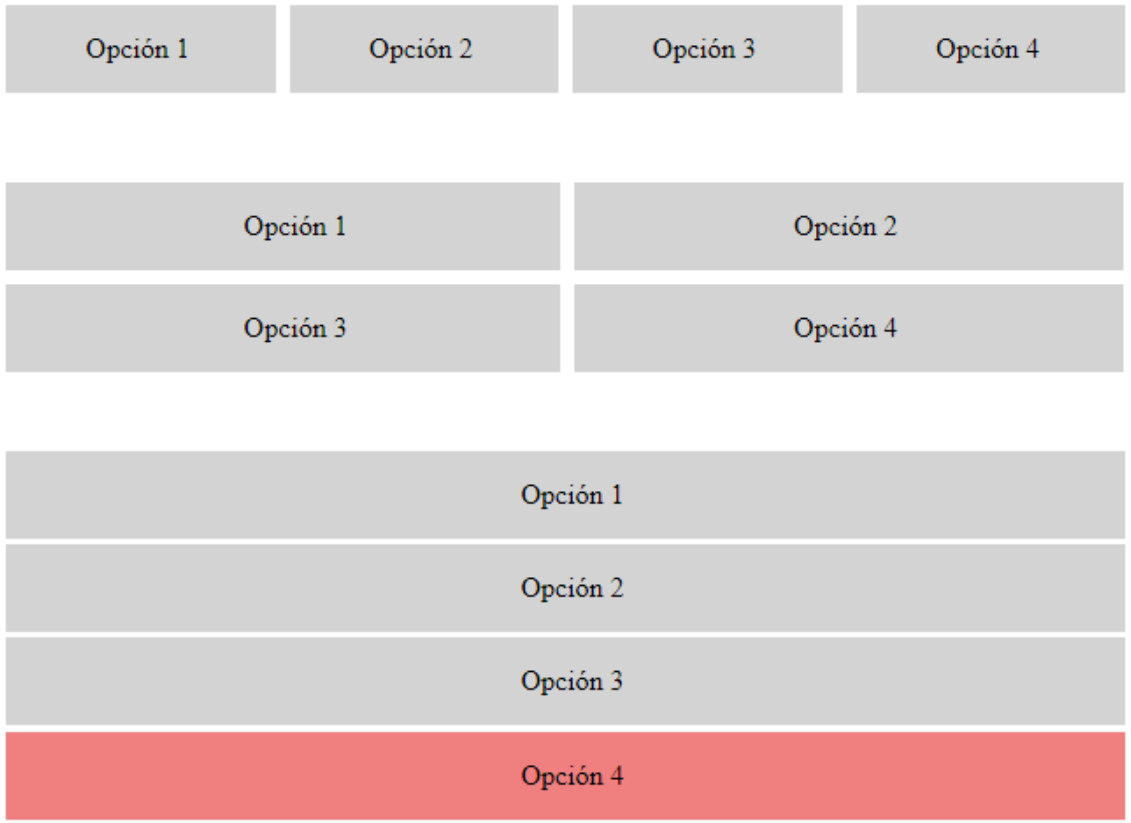
Un menú vertical y horizontal con GRID, tendría este formato.

HTML

```
<nav><ul>
   <li><a href="#">Opción 1</a></li>
   <li><a href="#">Opción 2</a></li>
   <li><a href="#">Opción 3</a></li>
   <li><a href="#">Opción 4</a></li>
  </ul></nav>
```

CSS

```
ul {  list-style-type:none;
      margin:0;
      padding:0;
      display: grid;
      grid-template-columns: repeat(4, 1fr);
      grid-template-rows: auto;
      gap: 0.2em;
}
a {   display:block;
       padding:1em;
       background-color:LightGray ;
       text-align:center;
       text-decoration:none;
       color:#000000;
}
a:hover {  background-color:LightCoral; }
```



Recursos CSS GRID

CSS-TRICKS A Complete Guide to Grid
https://css-tricks.com/snippets/css/complete-guide-grid/

M | Pavel Laptev: Learning CSS grid layout with the Swiss
https://medium.com/@PavelLaptev/learning-css-grid-with-the-swiss-2bd02e913fa

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/

GRID GARDEN Un juego para aprender CSS Grid
http://flexboxfroggy.com/#es