

Estructura con CSS

Diseño de Interfaces Web. Grid



CSS

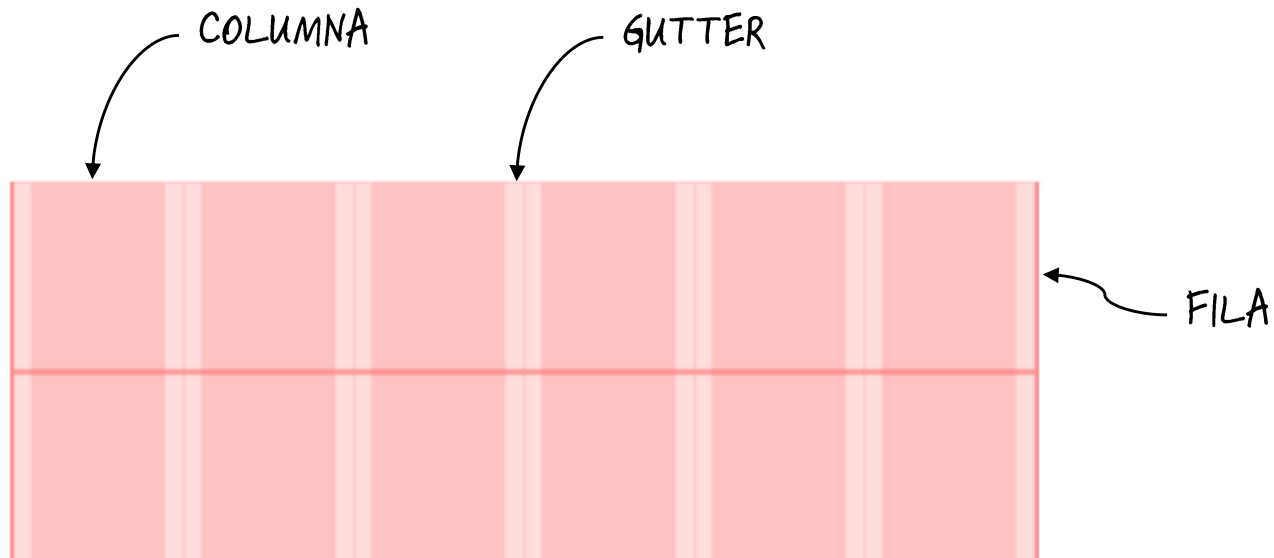
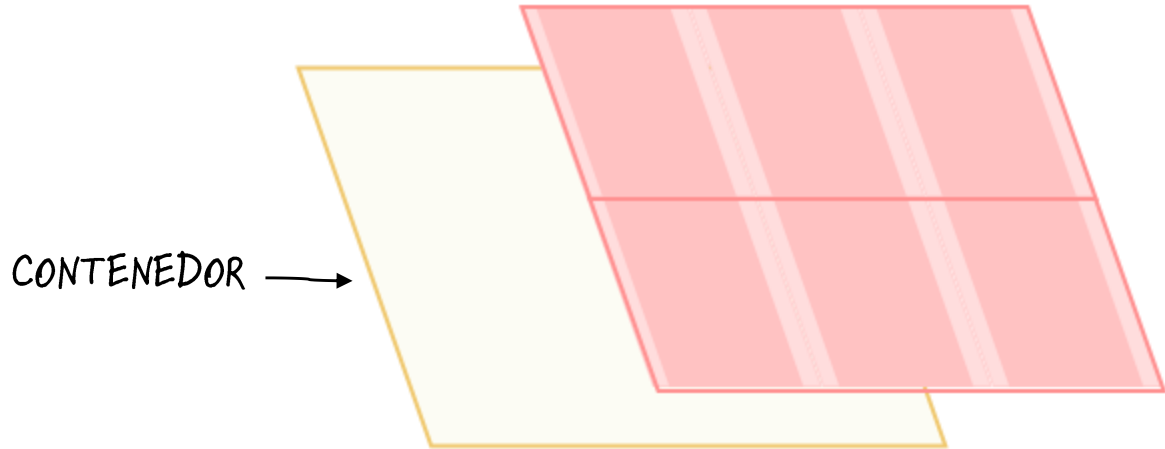




GRID HTML

Rejilla HTML

El diseño de páginas con rejilla se realiza mediante un contenedor, filas y columnas donde se colocan los contenidos. El ancho de las columnas no varía para adaptarse al contenido sino al dispositivo.



Rejilla HTML

Contenedor

El propósito del contenedor es ajustar el ancho total de la parrilla. El ancho del contenedor es normalmente del 100% pero se suele limitar, también, con `max-width` y `min-width`. Para simplificar los cálculos de los elementos del contenedor también se puede cambiar la propiedad `box-sizing`

```
.contenedor {  
  width : 100%;  
  max-width : 76em;  
  margin: 0 auto;  
}
```

```
.contenedor > * {  
  box-sizing: border-box;  
}
```

Fila

El propósito de las filas es mantener las columnas dentro de las mismas y no se pasen a otras filas. Para ello, se suele utilizar un *clearfix*

```
.fila:before,  
.fila:after {  
  content : '';  
  display : table;  
  clear: both;  
}
```

Columna

Es el elemento más difícil. Hay diferentes formas de posicionarlas: *responsive*, fijas, flotantes, inline-block, flex, etc.

Rejilla HTML

Columnas flotantes

Es el método menos propenso al error y uno de los más utilizados

Ancho

Para encontrar el ancho de una columna debemos dividir el ancho del contenedor por el número de columnas. Por ejemplo, con 6 columnas: $100/6 = 16,66\%$

Para calcular los diversos anchos, tendremos

```
.col-1 { width: 16.66%; }  
.col-2 { width: 33.33%; }  
.col-3 { width: 50%;    }  
.col-4 { width: 66.66%; }  
.col-5 { width: 83.33%; }  
.col-6 { width: 100%;  }
```

```
[class|='col'] {  
  float: left;  
  min-height : 1px;  
}
```

```
[class|='col'] {  
  float: left;  
  min-height : 1px;  
  width: 16.66%;  
}
```

Gutter

Es el *padding* de las columnas teniendo en cuenta que el modelo de caja utilizado es el de **border-box**

```
[class|='col'] {  
  float: left;  
  min-height : 1px;  
  width: 16.66%;  
  padding: 12px;  
}
```

Rejilla HTML

Rejilla de 12 columnas

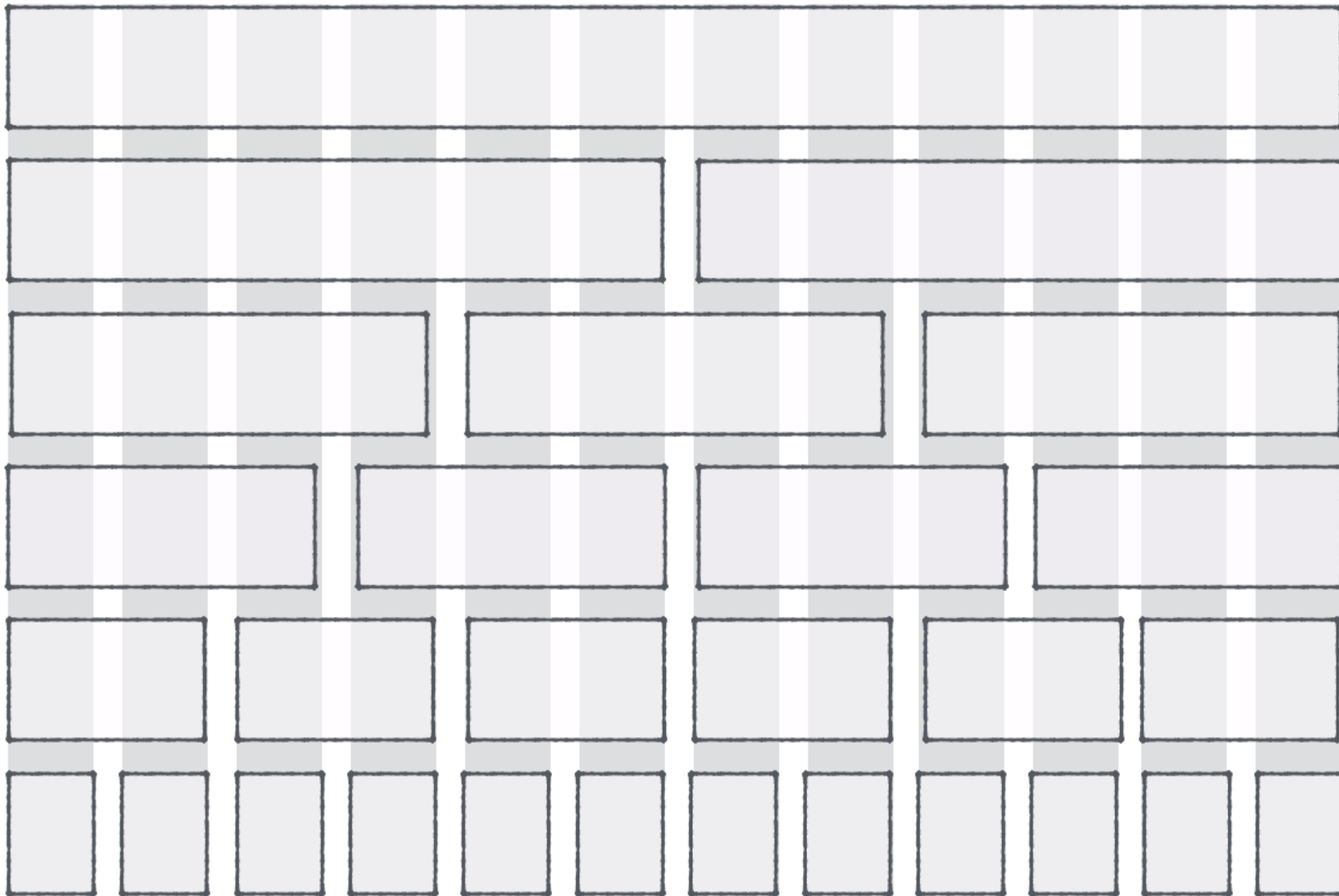
El tamaño hoy en día de las rejillas suelen ser de 12 columnas. Haciendo los cálculos, nos quedan los siguientes valores:

```
[class|='col'] {  
  float: left;  
  min-height : 1px;  
  width: 8.33%;  
  padding: 6px;  
}
```

```
.col-1 { width: 8.33%; }  
.col-2 { width: 16.66%; }  
.col-3 { width: 25%; }  
.col-4 { width: 33.33%; }  
.col-5 { width: 41.66%; }  
.col-6 { width: 50%; }  
.col-7 { width: 58.33%; }  
.col-8 { width: 66.66%; }  
.col-9 { width: 75%; }  
.col-10 { width: 83.33%; }  
.col-11 { width: 91.66%; }  
.col-12 { width: 100%; }
```

Rejilla HTML

Rejilla de 12 columnas



Rejilla HTML

Rejilla de 12 columnas

Rejilla HTML

Gutters con márgenes

En vez de fijar el *gutter* con un *padding* se puede fijar con un margen porcentual que facilita la respuesta *responsive* del modelo aunque complica el cálculo de los valores de los anchos de las columnas.

```
[class|='col']+[class|='col']{  
  margin-left: 1.6%;  
}
```

Ancho

$\text{tamaño_márgenes\%} = (\text{num_cols} - 1) * \text{tamaño_margen}$

$\text{espacio_fila} = 100\% - \text{tamaño_márgenes\%}$

$\text{ancho_col} = \text{espacio_fila} / \text{num_cols}$

$\text{col_x} = (\text{ancho_col} * x) + (\text{mar} * (x - 1))$

$\text{ancho_col} \rightarrow$ ancho columna
 $\text{num_cols} \rightarrow$ columnas totales

$\text{col_x} \rightarrow$ columna x
 $x \rightarrow$ el número de columna

Rejilla HTML

Gutters con márgenes

Para 12 columnas con gutter = 1.6%

$\text{tamaño_márgenes\%} = 1.6\% * 11 = 17.6\%$

$\text{espacio_fila} = 100\% - 17.6\% = 82.4\%$

$\text{ancho_col} = 82.4\% / 12 = 6.86666667\%$

$\text{col}_1 = (6.86667\% * 1) + (1.6\% * (1 - 1)) = 6.86666667\%$

$\text{col}_2 = (6.86667\% * 2) + (1.6\% * (2 - 1)) = 15.33333333\%$

$\text{col}_5 = (6.86667\% * 5) + (1.6\% * (5 - 1)) = 40.73\%$

```
.col-1 { width: 6.86%; }  
.col-2 { width: 15.33%; }  
.col-3 { width: 23.8%; }  
.col-4 { width: 32.26%; }  
.col-5 { width: 40.73%; }  
.col-6 { width: 49.20%; }  
.col-7 { width: 57.66%; }  
.col-8 { width: 66.13%; }  
.col-9 { width: 74.60%; }  
.col-10 { width: 83.06%; }  
.col-11 { width: 91.53%; }  
.col-12 { width: 100%; }
```

Rejilla HTML - Responsive

La solución más adoptada es la de definir diferentes tamaños de columnas en función del ancho del dispositivo. Para diferenciarlas, a la clase se le agrega un prefijo. Por ejemplo:

Prefijo	Tamaño del dispositivo
<code>.col-xs-</code>	<768px
<code>.col-sm-</code>	768px to 991px
<code>.col-md-</code>	992px to 1199px
<code>.col-lg-</code>	≥ 1200px

La orientación adoptada suele ser la de *mobile-first* por lo que las `.col-xs` funcionan para todos los tamaños.

Rejilla HTML – Anidado

Por regla general, solo pueden tener contenido las columnas y hay que respetar la jerarquía **contenedor** → **fila** → **columna** por lo que para anidar una rejilla se debe hacer en una columna ya definida. Ésta hará, a su vez, de **contenedor**. Sólo necesitamos, entonces, definir una fila y sus correspondientes columnas.

```
<div class="contenedor">
  <div class="fila"></div>
  <div class="col-6"></div>
  <div class="col-6">
    <div class="fila">
      <div class="col-9"></div>
      <div class="col-3"></div>
    </div>
  </div>
</div>
```

Rejilla HTML – Casos especiales

Columnas offset

Permiten dejar 'huecos' en una fila.
Para conseguirlo, se emplea la propiedad `margin` en vez de `width`

```
.col-offset-1 { margin-left: 8.33%; }
.col-offset-2 { margin-left: 16.66%; }
.col-offset-3 { margin-left: 25%; }
.col-offset-4 { margin-left: 33.33%; }
.col-offset-5 { margin-left: 41.66%; }
.col-offset-6 { margin-left: 50%; }
.col-offset-7 { margin-left: 58.33%; }
.col-offset-8 { margin-left: 66.66%; }
.col-offset-9 { margin-left: 75%; }
.col-offset-10 { margin-left: 83.33%; }
.col-offset-11 { margin-left: 91.66%; }
.col-offset-12 { margin-left: 100%; }
```

```
.col-pull-1 { right: 8.33%; }
.col-pull-2 { right: 16.66%; }
.col-pull-3 { right: 25%; }
.col-pull-4 { right: 33.33%; }
.col-pull-5 { right: 41.66%; }
.col-pull-6 { right: 50%; }
...
.col-push-1 { left: 8.33%; }
.col-push-2 { left: 16.66%; }
.col-push-3 { left: 25%; }
.col-push-4 { left: 33.33%; }
.col-push-5 { left: 41.66%; }
.col-push-6 { left: 50%; }
```

Columnas push/pull

Consisten en columnas que se colocan al inicio/final de una fila utilizando, respectivamente, las propiedades `left/right`

Rejilla HTML – Otro enfoque

Manteniendo la jerarquía **contenedor** → **fila** → **columna** se da más importancia a las filas que son las que determinan el tamaño indicando cuantas columnas entran en una de ellas.

```
[class]='col' {  
  float: left;  
  min-height : 1px;  
  padding: 6px;  
  box-sizing: border-box;  
}
```

```
[class]='fila' { box-sizing: border-box; }  
  
.fila-de1 > * { width: 100%; }  
.fila-de2 > * { width: 50%; }  
.fila-de3 > * { width: 33.33%; }  
.fila-de4 > * { width: 25%; }  
.fila-de5 > * { width: 20%; }  
.fila-de6 > * { width: 16.66%; }  
.fila-de7 > * { width: 14.28%; }  
.fila-de8 > * { width: 12.50%; }  
.fila-de9 > * { width: 11.11%; }  
.fila-de10 > * { width: 10%; }  
.fila-de11 > * { width: 9.10%; }  
.fila-de12 > * { width: 6.66%; }
```

Para el caso de que necesitemos columnas de distinto tamaño, hay que definir clases específicas en función del contenido

```
.col-contenido { width: 80%; }  
.col-cabecera { width: 20%; }  
.col-articulo { width: 50%; }
```

