



UNIVERSIDAD
DE GRANADA



Tecnologías Web

Grado en Ingeniería Informática

Tema 2 – Diseño y desarrollo de páginas web CSS: Distribución de los elementos (layout)

Este documento está protegido por la Ley de Propiedad Intelectual (Real Decreto Ley 1/1996 de 12 de abril).
Queda expresamente prohibido su uso o distribución sin autorización del autor.

© Javier Martínez Baena
jbaena@ugr.es

Departamento de Ciencias de la
Computación e Inteligencia Artificial
<http://decsai.ugr.es>



UNIVERSIDAD
DE GRANADA

Tecnologías Web

3º Grado en Ingeniería Informática

Diseño y desarrollo de páginas web

1. Introducción
2. Estructura y contenido de la página: HTML
3. Diseño de la página: CSS
 1. Fundamentos
 2. Distribución de los elementos (layout)
 1. Modelo de cajas (box model)
 2. Posicionamiento de los elementos
 3. El módulo Flexible box
 4. El módulo Grid
 5. Políticas de diseño de la página
 6. Diseño adaptable
 7. Bibliografía

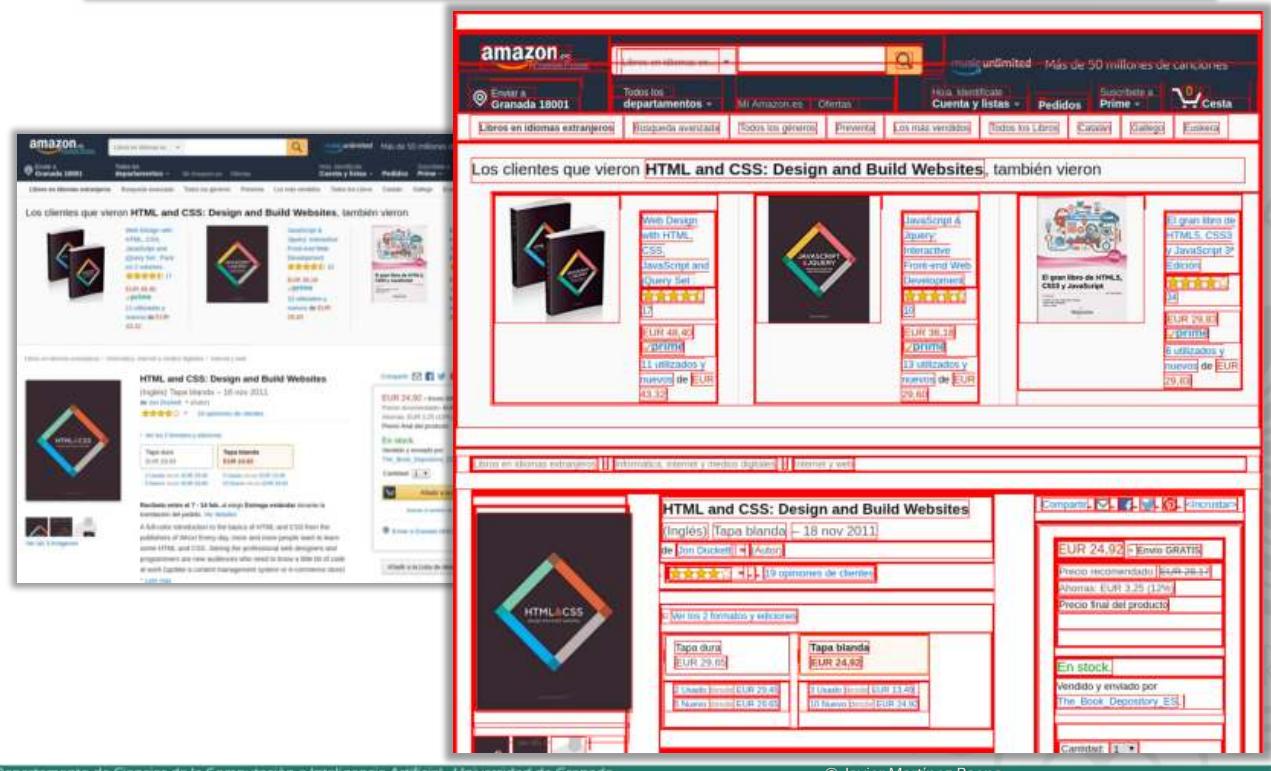


CSS: Modelo de cajas

El modelo de cajas (Box model)

El modelo de cajas (Box model)

Todo elemento de un documento es considerado una “caja” (box)



Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

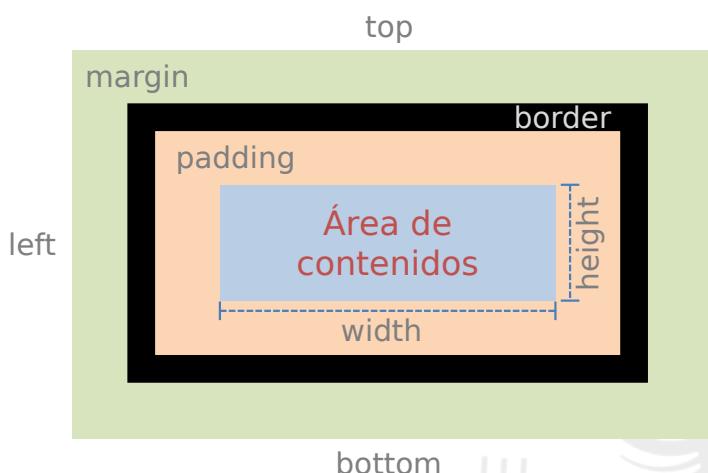
3

CSS: Modelo de cajas

El modelo de cajas (Box model)

El modelo de cajas (Box model)

Todo elemento de un documento es considerado una “caja” (box)



Margin: área de separación entre la caja y otras cajas (margen externo)
 Padding: área de separación entre el contenido y el borde (margen interno)

Margin/padding/border pueden diferir en top/bottom/left/right

CSS: Modelo de cajas
El modelo de cajas (Box model)

Alan Turing

Alan Mathison Turing, OBE (Paddington, Londres, 23 de junio de 1912-Wilmslow, Cheshire, 7 de junio de 1954), fue un:

- matemático,
- científico de la computación,
- filósofo

La Máquina de Turing

La máquina de Turing modela matemáticamente a una máquina que opera mecánicamente sobre una cinta. Está formada por:

1. Cinta
2. Cabezal
3. Registro de estado
4. Tabla de instrucciones
 - o Borra o escribe un símbolo
 - o Mueve el cabezal
 - o Cambia de estado

```

<body>
  <h1>Alan Turing</h1>
  <p><strong>Alan Mathison Turing</strong></p>
  Londres, 23 de junio de 1912-Wilmslow, Cheshire, 7 de junio de 1954), fue un:</p>
  <ul>
    <li>matemático,</li>
    <li>científico de la <strong>computación</strong>,</li>
    <li>filósofo</li>
  </ul>
  <h2>La Máquina de <em>Turing</em></h2>
  <p>La máquina de Turing modela matemáticamente a una máquina que opera mecánicamente sobre una cinta. Está formada por:</p>
  <ol>
    <li>Cinta</li>
    <li>Cabezal</li>
    <li>Registro de estado</li>
    <li>Tabla de instrucciones
      <ul>
        <li>Borra o escribe un <strong>símbolo</strong></li>
        <li>Mueve el cabezal</li>
        <li>Cambia de estado</li>
      </ul>
    </li>
  </ol>
</body>

```

CSS: Modelo de cajas
El modelo de cajas (Box model)

Contenido	Margin	Padding
width height	margin margin-top margin-right margin-bottom margin-left	padding padding-top padding-right padding-bottom padding-left

```

h1 {
  width: 300px;
  padding-left: 30px;
  margin: 30px;
}
h2 {
  width: 175px;
  height: 100px;
  margin-left: 60px;
  padding-right: 50px;
}
p {
  width: 50%;
  padding: 30px;
}

```

CSS: Modelo de cajas

El modelo de cajas (Box model)



Propiedades del borde

Ancho

border-width
border-top-width
border-right-width
border-bottom-width
border-left-width

Color

border-color
border-top-color
border-right-color
border-bottom-color
border-left-color

Estilo

border-style
border-top-style
border-right-style
border-bottom-style
border-left-style

Propiedades del borde (todas en una)

border, border-top, border-right, border-bottom, border-left

```
h1 {  
    border: orange solid;  
    border-width: 5px 15px 25px 35px;  
}  
  
h2 {  
    border: blue solid;  
    border-width: 5px 25px;  
}  
  
p {  
    border: 3px dotted;  
    border-top: 10px double red;  
    border-right-color: lightgreen;  
}
```

Título de la página**Subtítulo de la página**

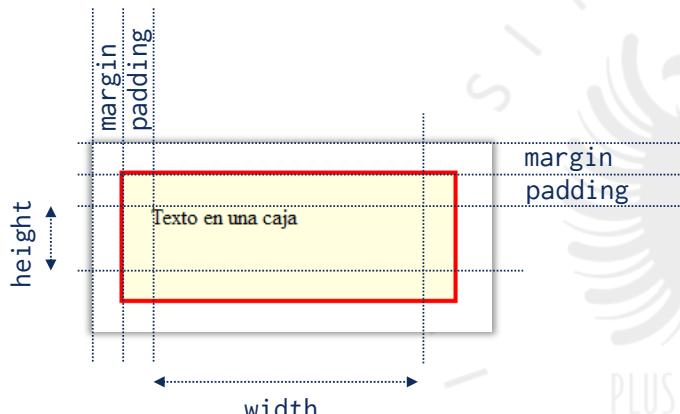
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras pharetra odio ex, nec tincidunt urna sagittis sit amet. Etiam ac nunc aliquam, feugiat mauris non, semper elit.

CSS: Modelo de cajas

El modelo de cajas (Box model)



```
border-color: red;           /* Borde del contenido */  
border-style: solid;        /* Borde del contenido */  
border-width: 3px;          /* Borde del contenido */  
width: 200px;               /* Tamaño (ancho) del contenido */  
height: 50px;                /* Tamaño (alto) del contenido */  
background: lightyellow;    /* Color de fondo del contenido */  
padding: 20px;              /* Separación interna contenido/borde */  
margin: 20px;                /* Separación externa borde/resto */
```



CSS: Modelo de cajas
El modelo de cajas (Box model)

```

border: green dotted 6px; /* Propiedades del borde */
width: 200px; /* Tamaño (ancho) del contenido */
height: 50px; /* Tamaño (alto) del contenido */
background: lightyellow; /* Color de fondo del contenido */
padding: 20px 40px 30px 10px; /* top, right, bottom, left */
margin-top: 20px; /* Separación externa top */
margin-left: 50px; /* Separación externa left */

border: blue dashed 6px; /* Propiedades del borde */
border-bottom-width: 20px; /* solo para el bottom */
border-top-color: red; /* solo para el top */
border-right-style: solid; /* solo para el right */
width: 200px; /* Tamaño (ancho) del contenido */
height: 80px; /* Tamaño (alto) del contenido */
background: lightyellow; /* Color de fondo del contenido */
padding-top: 20px;

```

CSS: Modelo de cajas
Ejemplo

Añadimos 2 nuevas reglas

```

p {
  padding: 20px;
  margin-left: 60px;
  margin-right: 20px;
  border-style: none;
  border-left-style: dotted;
  border-color: SlateGray;
  border-width: 6px;
}

h1 {
  padding-left: 60px;
  margin-right: 80px;
  margin-top: 60px;
}

```

ALAN TURING

Alan Mathison Turing, OBE (Paddington, Londres, 23 de junio de 1912-Wilmslow, Cheshire, 7 de junio de 1954), fue un matemático, lógico, científico de la computación, criptógrafo y también un filósofo. Además de eso le gustaba el **deporte de maratón**.

Es considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna.

La **máquina de Turing** modela matemáticamente a una máquina que opera mecánicamente sobre una cinta. Está formada por una cinta, un cabezal, un registro de estado y una tabla de instrucciones.

ADA LOVELACE

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la **Máquina analítica**.

Entre sus notas sobre la máquina se encuentra lo que se reconoce hoy como el primer algoritmo destinado a ser procesado por una máquina.



Box sizing

En ocasiones la forma de medir width y height puede ser confusa

```
h1 {  
    width: 200px;  
    border: 1px solid;  
}  
  
h2 {  
    width: 200px;  
    border: 1px solid;  
    padding: 30px;  
}
```

Título de la página

Subtítulo de la página

Aun teniendo el mismo width se ven de distinto tamaño

```
h1 {  
    width: 200px;  
    border: 1px solid;  
    box-sizing: border-box;  
}  
  
h2 {  
    width: 200px;  
    border: 1px solid;  
    padding: 30px;  
    box-sizing: border-box;  
}
```

Título de la página

Subtítulo de la página

width y height incluyen el padding y el border

Por defecto:
box-sizing: content-box



Tipos de elementos en la página

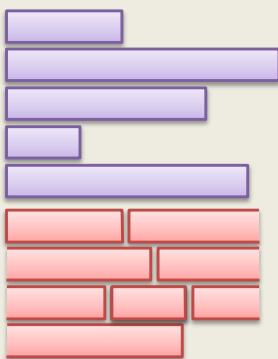
- **Nonreplaced**. Elementos cuyo contenido está en el propio documento
`<p>Texto de prueba</p>`
- **Replaced**. Elementos que sirven de contenedor para alojar contenido que no está en el documento
``

CSS: Modelo de cajas

Tipos de cajas

Tipos de cajas generadas por los elementos

- **Block.** Elementos cuyas cajas se disponen en vertical, cada caja se dibuja sola en una nueva línea
 - p, h1, h2, ..., footer, div, ol, ul, form, table, ...
- **Inline.** Elementos cuyas cajas se disponen en horizontal. No generan nuevas líneas
 - span, strong, a, em, img, input, label, ...



ALAN TURING

Inline

Block

CSS: Modelo de cajas

Tipos de cajas: block

El tamaño de la caja (block)

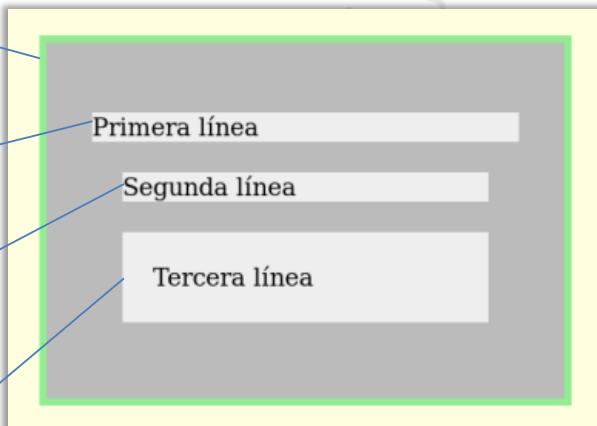
Por defecto, el ancho total de una caja (margen+borde+padding+width) coincide con el ancho del área de contenidos del bloque en el que está contenida. (Idem para el alto de una caja)

```
div {
    margin: 20px;
    padding: 30px;
    border: solid lightgreen 5px;
    background-color: #bbbbbb;
}

p#id1 {
    background-color: #eeeeee;
}

p#id2 {
    background-color: #eeeeee;
    margin: 20px;
}

p#id3 {
    background-color: #eeeeee;
    margin: 20px;
    padding: 20px;
}
```



CSS: Modelo de cajas**Tipos de cajas: block****El tamaño de la caja (block)****Width, margin-left, margin-right pueden tomar el valor auto**

```

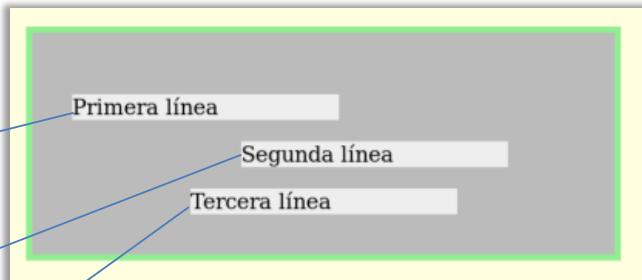
div {
    margin: 20px;
    padding: 30px;
    border: 5px;
}

#id1 {
    margin-right: 50px;
    width: 200px;
}

#id2 {
    margin-right: 50px;
    margin-left: auto;
    width: 200px;
}

#id3 {
    margin: auto;
    width: 200px;
}

```



- Horizontal (width,margin-left,margin-right)
auto significa que se ajuste el valor para completar
- Vertical (height,margin-bottom,margin-top)
Margin auto lo pone a cero

CSS: Modelo de cajas**Sobre los márgenes****Comportamiento especial de los márgenes**

- Los márgenes top y bottom de dos cajas adyacentes colapsan, es decir, se unen en un único margen con el tamaño del mayor de ambos
- En los elementos inline non-replaced los márgenes top y bottom no tienen efecto. [El elemento img es inline y replaced]
- Se pueden indicar márgenes negativos

margin: 50px;

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica. Entre sus notas sobre la máquina se encuentra lo que se reconoce hoy como el primer algoritmo destinado a ser procesado por una máquina.

margin: 20px;

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica. Entre sus notas sobre la máquina se encuentra lo que se reconoce hoy como el primer algoritmo destinado a ser procesado por una máquina.

margin: 30px;

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica. Entre sus notas sobre la máquina se encuentra lo que se reconoce hoy como el primer algoritmo destinado a ser procesado por una máquina.

CSS: Modelo de cajas

Tipos de cajas: inline

**Elementos inline**

Al calcular el tamaño de la caja no tienen efecto estas propiedades:

- Márgenes verticales
- Padding vertical
- Bordes verticales
- Height, Width

```
h1 {
    border-width: 10px;
    padding: 20px;
    margin: 20px;
    border-color: lightblue;
    border-style: solid;
}

em {
    border-width: 10px;
    padding: 10px;
    margin: 20px;
    border-color: lightgreen;
    border-style: solid;
}
```

Ada Lovelace

Augusta Ada King, *Condesa de Lovelace*, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

CSS: Modelo de cajas

La propiedad overflow

**La propiedad de desbordamiento (overflow)**

¿Qué ocurre si el contenido de un elemento no cabe en la caja?

La propiedad overflow indica qué hacer:

- visible: lo muestra (aunque solape con otros)
- hidden: solo muestra lo que cabe en la caja
- scroll: añade barras de scroll
- auto: deja decidir al navegador cómo ha de presentarlo

<code>overflow:visible</code>	<code>overflow:hidden</code>	<code>overflow:scroll</code>	<code>overflow:auto</code>
Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.	Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.	Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.	Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

CSS: Modelo de cajas**Tipos de cajas: display****La propiedad display****Permite cambiar el tipo de caja (inline, block, none, ...)**

```
<body>
<h1>Ada Lovelace</h1>
<p>Augusta Ada King, <em>Condesa de Lovelace</em>, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una <em>matemática</em> y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de <em>Charles Babbage</em>, la Máquina analítica.</p>
</body>
```

Ada Lovelace

Augusta Ada King, *Condesa de Lovelace*, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una *matemática* y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de *Charles Babbage*, la Máquina analítica.

CSS: Modelo de cajas**Tipos de cajas: display****La propiedad display****Permite cambiar el tipo de caja (inline, block, none, ...)****Ada Lovelace**

→ Augusta Ada King,
Condesa de Lovelace
(nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una
→ *matemática*
y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de
Charles Babbage
, la Máquina analítica.

```
em {
  display: block;
}
```

Ada Lovelace

→ Augusta Ada King, , (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de , la Máquina analítica.

```
em {
  display: none;
}
```

Ada Lovelace

→ Augusta Ada King,
Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una *matemática* y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de *Charles Babbage*, la Máquina analítica.

```
h1, p {
  display: inline;
}
```

CSS: Modelo de cajas

Tipos de cajas: inline-block

**Elementos inline-block****Elementos con comportamiento mixto:**

- Respecto a su contenedor son inline+replaced
- Su contenido se comporta como block

```
p {
    border: solid 10px lightgreen;
    margin: 10px;
    padding: 10px;
    background-color: rgba(20, 100, 200, 0.2);
    height: 50px;
    width: 150px;
    text-align: center;
}
```

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina mecánica de uso general de Charles Babbage, la máquina analítica.

display: inline

Augusta Ada King, Condesa de Lovelace

, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una

matemática

y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de

Charles Babbage

, la máquina analítica.

display: block

Augusta Ada King, Condesa de Lovelace

, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una

matemática

y escritora británica

conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de

Charles Babbage

, la máquina analítica.

display: inline-block

CSS: Modelo de cajas

La propiedad display

**La propiedad display en CSS3**

El estándar tiende a cambiar la forma de usarla definiendo distintas categorías que aluden a distintos comportamientos que se pueden combinar:

- Comportamiento respecto a la caja contenedora
block, inline
- Comportamiento del contenido
flow, flex, grid, table, ...
- ... otros

Esta nueva característica tienen poco soporte aun

Tradicional
block
inline
inline-block

Nuevo estándar
block flow
inline flow
inline flow-root



UNIVERSIDAD
DE GRANADA

Tecnologías Web

3º Grado en Ingeniería Informática

Diseño y desarrollo de páginas web

»

- 1. Introducción**
- 2. Estructura y contenido de la página: HTML**
- 3. Diseño de la página: CSS**
 - 1. Fundamentos**
 - 2. Distribución de los elementos (layout)**
 - 1. Modelo de cajas (box model)**
 - 2. Posicionamiento de los elementos**
 - 3. El módulo Flexible box**
 - 4. El módulo Grid**
 - 5. Políticas de diseño de la página**
 - 6. Diseño adaptable**
 - 7. Bibliografía**

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena

CSS: Posicionamiento

Flujo normal

Flujo normal (por defecto)

Se formatean las cajas según su contexto:

- Contexto “Block-level”: las cajas se disponen una tras otra, **verticalmente**, desde el comienzo del bloque que las contiene.
- Contexto “Inline-level”: las cajas se disponen una tras otra, **horizontalmente**, formando líneas. *Cada línea forma una nueva caja llamada “line box”.*

Esquemas de posicionamiento

- Normal
- Absoluto
- Relativo
- Fijo
- Pegajoso
- Flotante

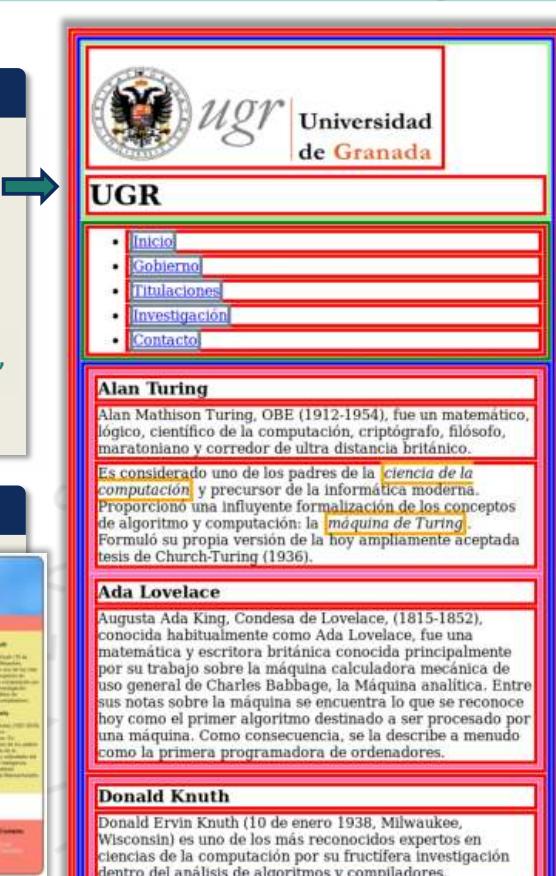


Diagram illustrating the normal flow of CSS positioning. It shows a website layout with various elements like the header, menu, and content blocks, each represented by a red-bordered box. An arrow points from the 'Flujo normal' text to this diagram.

CSS: Posicionamiento**Posicionamiento relativo****Posicionamiento relativo**

Tras acomodar una caja según el flujo normal, se puede desplazar de forma relativa a la posición en donde se había colocado.
Se considera una variante del flujo normal puesto que el elemento sigue ocupando un sitio en la distribución de cajas.

```
position: relative  
top:  
bottom:  
right:  
left:
```

```
#mat {  
    position: relative;  
    left: -60px;  
    top: -28px;  
    border-color: lightsalmon;  
}
```

Ada Lovelace

Augusta Ada King, *Condesa de Lovelace*, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una *matemática* y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de *Charles Babbage*, la Máquina analítica.

CSS: Posicionamiento**Posicionamiento absoluto****Posicionamiento absoluto**

- El elemento se saca por completo del flujo de cajas normal.
- La posición es relativa al bloque contenedor dentro del que está. Ese bloque contenedor será el primer ancestro de tipo block-level que tenga una propiedad position de tipo absolute, relative o fixed

```
position: absolute
```

```
#mat {  
    position: absolute;  
    left: 20px;  
    top: 40px;  
    border-color: lightsalmon;  
}
```

Ada Lovelace

matemática

Augusta Ada King, *Condesa de Lovelace*, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de *Charles Babbage*, la Máquina analítica.

Ada Lovelace

Augusta Ada King, *Condesa de Lovelace*, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de *Charles Babbage*, la Máquina analítica.

CSS: Posicionamiento

Posicionamiento fijo

Posicionamiento fijo

- El elemento se saca por completo del flujo de cajas normal.
- La posición es relativa al viewport (pantalla). Si se hace scrolling se mantiene la posición relativa en la pantalla.

`position: fixed;`

```
position: fixed;
top: 0;
left: 20px;
right: 20px;
```

```
position: fixed;
left: 20px;
top: 150px;
```

```
position: fixed;
bottom: 40px;
left: 20px;
```

The screenshot shows a web page with a sidebar on the left containing biographical information about Donald Knuth, Marvin Minsky, and Alan Turing. The sidebar has a fixed position with the CSS rule `position: fixed; top: 0; left: 20px; right: 20px;`. The main content area on the right also has a sidebar with the same fixed position rule. The footer of the page includes the University of Granada logo and navigation links.

This screenshot shows a web page with a sidebar on the right side containing information about Ada Lovelace. The sidebar has a fixed position with the CSS rule `position: fixed; right: 0; top: 150px; height: 400px; overflow: auto;`.

```
position: fixed;
right: 0;
top: 150px;
height: 400px;
overflow: auto;
```

```
position: fixed;
bottom: 0;
left: 20px;
right: 20px;
```

CSS: Posicionamiento

Flujo normal

Posicionamiento “pegajoso”

- Comportamiento mixto relativo/absoluto
- El contenedor ha de ser “scrollable”
- Cuando el borde de la caja toca el borde de la caja contenedora su posición se queda fijada (relativa/pegada a ese borde)
- Aun tiene poco soporte

`position: sticky;`

This screenshot shows a web page with a sidebar on the left containing information about Alan Turing. The sidebar has a sticky position with the CSS rule `position: sticky; top: 0;`. As the page is scrolled down, the sidebar remains attached to the top of the screen.

This screenshot shows a web page with a sidebar on the right containing information about Ada Lovelace. The sidebar has a sticky position with the CSS rule `position: sticky; bottom: 0;`. As the page is scrolled up, the sidebar remains attached to the bottom of the screen.

CSS: Posicionamiento**Posicionamiento flotante****Cajas flotantes**

- Una caja flotante no sigue el flujo normal.
- El contenido de la caja flotante se desplaza a la izquierda/derecha de la caja en la que está contenida.
- El resto de cajas se distribuyen verticalmente sin tener en cuenta a la caja flotante.
- La actual “line box” y siguientes que se crean junto a la caja flotante, se recortan para “fluir” a su alrededor.

```
float: left  
float: right
```


<p>Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.</p>

```
img {  
    float: right;  
    width: 20%;  
    margin: 10px;  
}
```

Ada Lovelace

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

**CSS: Posicionamiento**
Posicionamiento flotante

<p>Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.</p>

```
img {  
    float: left;  
    width: 20%;  
    margin: 10px;  
}
```



Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

```
img {  
    float: right;  
    width: 20%;  
    margin: 10px;  
}
```



Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

CSS: Posicionamiento**Posicionamiento flotante**

```

```

```
<h3>Ada Lovelace</h3>
```

pAugusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

```
img {  
    float: left;  
    width: 20%;  
    margin: 10px;  
}
```

El efecto de float afecta a todos los elementos que siguen

```
p {  
    clear: both;  
}
```

clear: impide que haya elementos float a izdq/dcha del elemento

**Ada Lovelace**

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

**Ada Lovelace**

Augusta Ada King, Condesa de Lovelace, (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la Máquina analítica.

**CSS: Posicionamiento**
Posicionamiento flotante**Ejemplo****Elementos float para conseguir layout multicolumna**

```
#contenedor {  
    background-color: lightgreen;  
}  
  
#item1 {  
    float: left;  
    margin: 1%;  
    width: 44%;  
    background-color: gold;  
}  
  
#item2 {  
    float: right;  
    margin: 1%;  
    width: 44%;  
    background-color: lightsalmon;  
}  
  
#item3 {  
    background-color: lightcyan;  
}
```

```
<div id="contenedor">  
    <p id="item1">ITEM 1. Lorem ipsum ... </p>  
    <p id="item2">ITEM 2. Proin mauris ...</p>  
</div>  
<p id="item3">ITEM 3. Nullam in ...</p>
```

ITEM 1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec in lectus quis dolor malesuada lobortis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur condimentum ultrices massa at congue. Ut commodo non odio in efficitur. Proin nec feugiat leo. Praesent vulputate turpis nec porttitor suscipit. Phasellus quis molestie nisi.

ITEM 2. Proin mauris leo, congue dictum ullamcorper ac, elementum a tellus. Donec pellentesque tortor a tortor fringilla, vitae convallis eros tristique.

ITEM 3. Nullam in nisi sit amet enim mollis sagittis. Mauris orci augue, egestas sed lacus in, sollicitudin tempus ex.

CSS: Posicionamiento
Posicionamiento flotante

```
<div id="contenedor">
  <p id="item1">ITEM 1. Lorem ipsum ... </p>
  <p id="item2">ITEM 2. Proin mauris ...</p>
  <p>uno dos tres cuatro cinco seis siete</p>
</div>
<p id="item3">ITEM 3. Nullam in ...</p>
```

ITEM 1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec in lectus quis dolor malesuada lobortis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur condimentum ultrices massa at congue. Ut commodo non odio in efficitur. Proin nec feugiat leo. Praesent vulputate turpis nec porttitor suscipit. Phasellus quis molestie nisi.	uno dos tres	ITEM 2. Proin mauris leo, congue dictum ullamcorper ac, elementum a tellus. Donec pellentesque tortor a tortor fringilla, vitae convallis eros tristique.	ITEM 3. Nullam in nisi sit amet enim mollis sagittis. Mauris orci augue, egestas sed lacus in, sollicitudin tempus ex.
cuatro cinco seis siete			

CSS: Posicionamiento
Posicionamiento flotante

```
<div id="contenedor">
  <p id="item1">ITEM 1. Lorem ipsum ... </p>
  <p id="item2">ITEM 2. Proin mauris ...</p>
  <p>uno dos tres cuatro cinco seis siete</p>
</div>
<div class="limpiar"></div>
<p id="item3">ITEM 3. Nullam in ...</p>
```

```
.limpiar {
  clear: both;
```

Con overflow se puede conseguir el mismo efecto (depende de contenido)

```
#contenedor {
  background-color: lightgreen;
  overflow: auto; /* overflow: hidden */
}
```

CSS: Posicionamiento
Tamaño de la caja relativo a ...

Box-sizing: cálculo del tamaño de la caja

El tamaño de la caja (width, height, max-width, max-height) se calcula relativo al contenido (por defecto no incluye margin, border ni padding)

box-sizing: content-box box-sizing: border-box	Por defecto El tamaño incluye border y padding pero no margin
---	--

```
<body>
<div class="contenedor">
    <div class="caja">Lorem ipsum dolor sit amet</div>
    <div class="caja">Lorem ipsum dolor sit amet</div>
</div>
</body>
```

```
.caja {
    box-sizing: border-box;
    width: 50%;
    border: 10px solid orange;
    float: left;
}
```

```
.contenedor {
    width: 250px;
    border: 10px solid blue;
    overflow: auto;
}

.caja {
    box-sizing: content-box;
    width: 50%;
    border: 10px solid orange;
    float: left;
}
```

Tecnologías Web
3º Grado en Ingeniería Informática
Diseño y desarrollo de páginas web

UNIVERSIDAD DE GRANADA

DECSAI

- 1. Introducción
- 2. Estructura y contenido de la página: HTML
- 3. Diseño de la página: CSS
 - 1. Fundamentos
 - 2. Distribución de los elementos (layout)
 - 1. Modelo de cajas (box model)
 - 2. Posicionamiento de los elementos
 - 3. El módulo Flexible box
 - 4. El módulo Grid
 - 5. Políticas de diseño de la página
 - 6. Diseño adaptable
 - 7. Bibliografía



El módulo Flexible Box

Ejemplo

Ejemplo: página multi columna

... con float → complicado

```
<body id="principal">
  <p id="p1">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <p id="p2"> Nunc vitae sem luctus, sollicitudin ante eget, luctus felis. Praesent in
erat vel felis faucibus vulputate. In augue nisl, placerat et arcu sed, molestie cursus
diam.</p>
</body>

#p1 {
  width: 40%;
  float: left;
}
#p2 {
  width: 60%;
  float: left;
  background-color: #ccccee;
}
#p1 {
  width: 40%;
  float: left;
  background-color: #cceecc;
}
#p2 {
  width: 60%;
  float: left;
  background-color: #ccccee;
}

#principal {
  background-color: #cceecc;
  overflow: hidden;
}
```

Lorem ipsum dolor sit amet, consectetur sollicitudin ante eget, luctus felis. Praesent in erat vel felis faucibus vulputate. In augue nisl, placerat et arcu sed, molestie cursus diam.

Nunc vitae sem luctus, sit amet, consectetur sollicitudin ante eget, luctus felis. Praesent in erat vel felis faucibus vulputate. In augue nisl, placerat et arcu sed, molestie cursus diam.



El módulo Flexible Box

Ejemplo

Ejemplo: página multi columna

... con float → complicado

```
<body id="principal">
  <p id="p1">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <p id="p2"> Nunc vitae sem luctus, sollicitudin ante eget, luctus felis. Praesent in
erat vel felis faucibus vulputate. In augue nisl, placerat et arcu sed, molestie cursus
diam.</p>
</body>

#p1 {
  width: 100px;
  float: left;
  margin-right: -100px;
  position: relative;
}
#p2 {
  float: left;
  width: 200px;
  border-left: 100px solid #cceecc;
  background-color: #ccccee;
}
#principal {
  width: 300px;
  margin: 0 auto;
}

#p1 {
  width: 100px;
  float: left;
  margin-right: -100px;
  position: relative;
}
#p2 {
  float: left;
  width: 200px;
  border-left: 100px solid #cceecc;
  background-color: #ccccee;
}
#principal {
  width: 300px;
  margin: 0 auto;
}
```

Lorem ipsum dolor sit amet, consectetur sollicitudin ante eget, luctus felis. Praesent in erat vel felis faucibus vulputate. In augue nisl, placerat et arcu sed, molestie cursus diam.

Nunc vitae sem luctus, sit amet, consectetur sollicitudin ante eget, luctus felis. Praesent in erat vel felis faucibus vulputate. In augue nisl, placerat et arcu sed, molestie cursus diam.

En general:

- Complicado
- Muy a medida
- “Tricky”

El módulo Flexible Box

Ejemplo

Ejemplo: página multi columna

Solución actual → usar herramientas más avanzadas de diseño de CSS
Cuidado → son recientes y pueden no estar soportadas por el navegador

```
#p1 {
    display: table-cell;
    background-color: #cceccc;
    width: 40%;
}
#p2 {
    display: table-cell;
    background-color: #cccccc;
    width: 60%;
```

Table layout

```
#principal {
    display: flex;
}
#p1 {
    background-color: #cceccc;
    width: 40%;
}
#p2 {
    background-color: #cccccc;
    width: 60%;
```

Flexbox layout

Flexbox layout

Grid layout

Grid layout

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

39

El módulo Flexible Box

Introducción

El módulo FlexBox

Disponible en CSS3
Mejora el posicionamiento frente a técnicas que usan float

- Working Draft (WD).
- **Candidate Recommendation (CR)**. Document that W3C believes has been widely reviewed and satisfies the Working Group's technical requirements. W3C publishes a Candidate Recommendation to gather implementation experience.
- Proposed Recommendation (PR). Mature technical report that W3C has sent to the W3C Advisory Committee for final endorsement.
- W3C Recommendation (REC). Similar to the standards published by other organizations.

<https://www.w3.org/TR/css-flexbox-1/>

Febrero 2018

IE	Edge	Firefox	Chrome	Safari	iOS Safari	Opera Mini	Chrome for Android
11	16	58	49 62 63	11	10.2 10.3	all	64
14	17	59	64 65	11.1	11.3	http://caniuse.com	

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

40

El módulo Flexible Box

Introducción

El módulo FlexBox

Pensado para distribuir elementos en 1D (en vertical o en horizontal)

La distribución con elementos float tiene inconvenientes:

- Alineación vertical de cajas:
 - No permite que varios elementos en una caja tengan un mismo alto
- Hay que cancelar su efecto para evitar que se propague
- Pueden dejar vacíos los contenedores

...

Flexbox facilita:

- Alineación vertical y horizontal de contenidos
- Control de la separación y espaciado entre elementos
- Cambiar el orden de los elementos

El módulo Flexible Box

Contenedores y elementos Flex

El módulo FlexBox

- **Contenedor Flex.** Cualquier elemento HTML (normalmente div o estructural) dentro del cual se aplica el modelo de FlexBox
- **Elemento Flex.** Cada uno de los elementos dentro de un contenedor Flex

```
<body>
  <div id="contenedor">
    <div id="caja1">Caja número 1</div>
    <div id="caja2">Caja número 2</div>
    <div id="caja3">Caja número 3</div>
  </div>
</body>
```

Caja número 1

Caja número 2

Caja número 3

```
#contenedor {
  display: flex;
}
```

Caja número 1
Caja número 2
Caja número 3

Contenedor Flex
Elementos Flex

El módulo Flexible Box

Ejes

Conceptos

- Eje principal (main axis)
- Eje secundario (cross axis)

Los elementos se distribuyen a lo largo del eje principal

- `flex-direction: row` Eje principal = horizontal (por defecto)
- `flex-direction: column` Eje principal = vertical

```
#contenedor {
  display: flex;
  flex-direction: column;
}
```

Caja número 1

Caja número 2

Caja número 3

<https://www.w3.org/TR/css-flexbox-1/>

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

43

El módulo Flexible Box

Ejemplo

Un ejemplo con más elementos

```
<div id="contenedor">
  <div id="caja1">Caja primera</div>
  <div id="caja2">Caja 2</div>
  <div id="caja3">Caja número 3</div>
  <div id="caja4">La 4</div>
  <div id="caja5">Quinta caja</div>
  <div id="caja6">Y otra más (6)</div>
  <div id="caja7">La última es la 7</div>
</div>
<p>Continua ...</p>
```

Caja primera

Caja 2

Caja número 3

La 4

Quinta caja

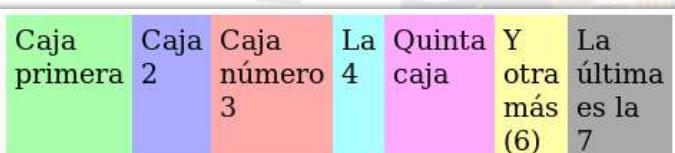
Y otra más (6)

La última es la 7

Continua ...

El tamaño de cada caja se ajusta automáticamente según su contenido

```
#contenedor {
  display: flex;
}
```



Continua ...



El módulo Flexible Box

El orden de los elementos

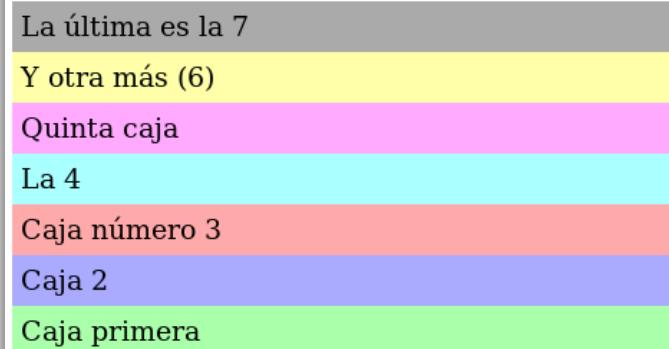
Cambiando el orden de los elementos

- Invertir el orden

```
#contenedor {
  display: flex;
  flex-direction: row-reverse;
}
```



```
#contenedor {
  display: flex;
  flex-direction: column-reverse;
}
```



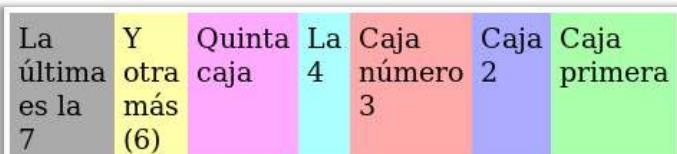
El módulo Flexible Box

El orden de los elementos

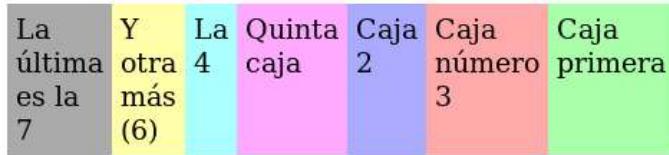
Cambiando el orden de los elementos

- Especificar el orden

```
#contenedor {
  display: flex;
}
```



```
#caja1 { order: 4; }
#caja2 { order: 1; }
#caja3 { order: 3; }
#caja6 { order: -1; }
#caja7 { order: -2; }
```



- Si no se especifica: en orden de aparición (los primeros)
- Positivo: orden a continuación de los no numerados
- Negativo: orden antes de los no numerados

El módulo Flexible Box

Alineamiento

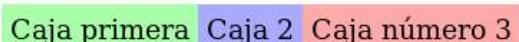
Alineamiento dentro del contenedor

- Respecto al eje principal

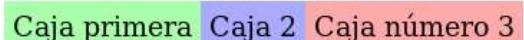
`justify-content: flex-start / flex-end / center / space-between / space-around`

```
<div id="contenedor">
  <div id="caja1">Caja primera</div>
  <div id="caja2">Caja 2</div>
  <div id="caja3">Caja número 3</div>
</div>
```

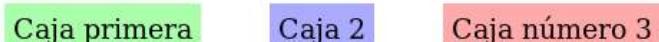
```
#contenedor {
  display: flex;
}
```



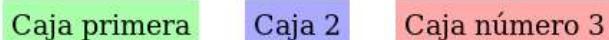
```
#contenedor {
  display: flex;
  justify-content: flex-end;
}
```



```
#contenedor {
  display: flex;
  justify-content: space-between;
}
```



```
#contenedor {
  display: flex;
  justify-content: space-around;
}
```



El módulo Flexible Box

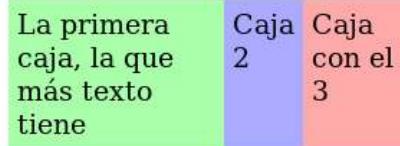
Alineamiento

Alineamiento dentro del contenedor

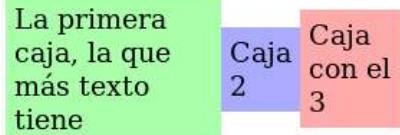
- Respecto al eje secundario

`align-items: stretch / flex-start / flex-end / center`

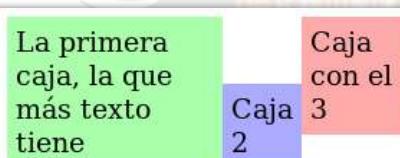
```
#contenedor {
  display: flex;
  align-items: stretch; /* Por defecto */
}
```



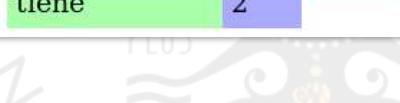
```
#contenedor {
  display: flex;
  align-items: center;
}
```



```
#contenedor {
  display: flex;
  align-items: flex-end;
}
```



```
#caja3 {
  align-self: flex-start;
}
```





El módulo Flexible Box

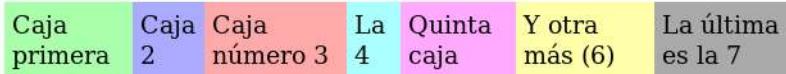
División en líneas

División en varias líneas

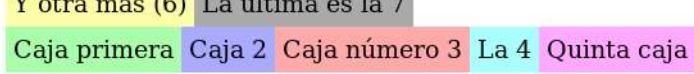
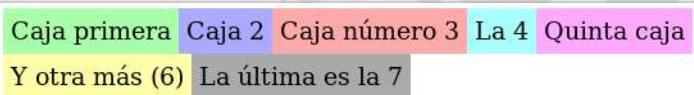
Si las cajas no caben bien en una línea → continuar en la siguiente

```
<div id="contenedor">
  <div id="caja1">Caja primera</div>
  <div id="caja2">Caja 2</div>
  <div id="caja3">Caja número 3</div>
  <div id="caja4">La 4</div>
  <div id="caja5">Quinta caja</div>
  <div id="caja6">Y otra más (6)</div>
  <div id="caja7">La última es la 7</div>
</div>
<p>Continua ...</p>
```

```
#contenedor {
  display: flex;
  flex-wrap: wrap;
}
```



```
#contenedor {
  display: flex;
  flex-wrap: wrap-reverse;
}
```



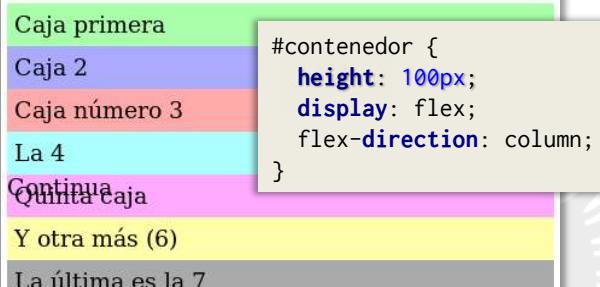
El módulo Flexible Box

División en columnas

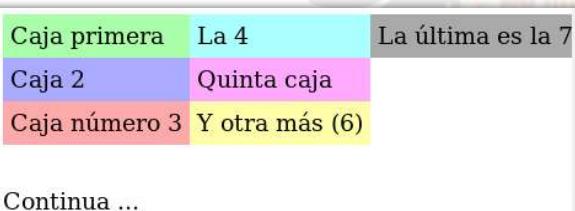
División en varias columnas

```
Caja primera
Caja 2
Caja número 3
La 4
Quinta caja
Y otra más (6)
La última es la 7
Continua ...
```

```
#contenedor {
  display: flex;
  flex-direction: column;
}
```



```
#contenedor {
  height: 100px;
  display: flex;
  flex-wrap: wrap;
  flex-direction: column;
}
```



Continua ...

El módulo Flexible Box
Cajas flexibles

Cajas flexibles

flex-basis Tamaño recomendado (width) si el contenedor lo admite.

```
#contenedor {
    width: 600px;
    display: flex;
}

#caja1 { flex-basis: 100px; }
#caja2 { flex-basis: 100px; }
#caja3 { flex-basis: 200px; }
```

100	100	200
-----	-----	-----


```
#contenedor { width: 300px; }
```

75	75	150
----	----	-----

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 51

El módulo Flexible Box
Cajas flexibles

Cajas flexibles

flex-grow Si sobra espacio en el contenedor: redistribuir el espacio

```
#contenedor {
    width: 600px;
    display: flex;
}

#caja1 { flex-basis: 100px; flex-grow: 2; }
#caja2 { flex-basis: 100px; flex-grow: 1; }
#caja3 { flex-basis: 200px; flex-grow: 1; }
```

200	150	250
-----	-----	-----

Sobran 200px:

- 2 partes para #caja1 (100)
- 1 parte para #caja2 (50)
- 1 parte para #caja3 (50)

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 52

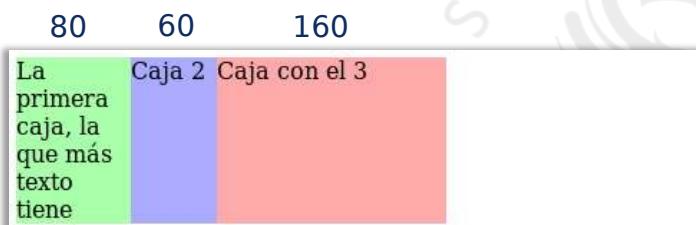
El módulo Flexible Box**Cajas flexibles****Cajas flexibles**

flex-shrink

Si falta espacio en el contenedor: redistribuir las pérdidas

```
#contenedor {
    width: 300px;
    display: flex;
}

#caja1 { flex-basis: 100px; flex-shrink: 1; }
#caja2 { flex-basis: 100px; flex-shrink: 2; }
#caja3 { flex-basis: 200px; flex-shrink: 1; }
```



Ancho del contenedor: 300

Suma de ítems: 400

Hay que reducir: $400 - 300 = 100$

#caja1: 1x100 / 500 → -20px

#caja2: 2x100 / 500 → -40px

#caja3: 1x200 / 500 → -40px

El módulo Flexible Box**Ejemplo****Encabezado de la página**

Página 1
Página 2
Página 3
Página 4
Página 5
Página 6

¿Qué es Lorem Ipsum?

Lorem Ipsum es simplemente el texto de relleno de las imprentas y archivos de texto. Lorem Ipsum ha sido el texto de relleno estándar de las industrias desde el año 1500, cuando un impresor (N. del T. persona que se dedica a la imprenta) desconocido usó una galería de textos y los mezcló de tal manera que logró hacer un libro de textos especímen. No sólo sobrevivió 500 años, sino que también ingresó como texto de relleno en documentos electrónicos, quedando esencialmente igual al original. Fue popularizado en los 60s con la creación de las hojas "Letraset", las cuales contenían pasajes de Lorem Ipsum, y más recientemente con software de autoedición, como por ejemplo Aldus PageMaker, el cual incluye versiones de Lorem Ipsum.

De donde viene

Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. Tiene sus raíces en una pieza clásica de la literatura del Latín, que data del año 45 antes de Cristo, haciendo que este adquiera mas de 2000 años de antigüedad. Richard McClintock, un profesor de Latin de la Universidad de Hampden-Sydney en Virginia, encontró una de las palabras más oscuras de la lengua del latín, "consectetur", en un pasaje de Lorem Ipsum, y al seguir leyendo distintos textos del latín, descubrió la fuente inindudable. Lorem Ipsum viene de las secciones 1.10.32 y 1.10.33 de "de Finibus Bonorum et Malorum" (Los Extremos del Bien y El Mal) por Cicero, escrito en el año 45 antes de Cristo. Este libro es un tratado de teoría de éticas, muy popular durante el Renacimiento. La primera linea del Lorem Ipsum, "Lorem ipsum dolor sit amet..", viene de una linea en la sección 1.10.32

¿Dónde puedo conseguirlo?

Hay muchas variaciones de los pasajes de Lorem Ipsum disponibles, pero la mayoría sufrió alteraciones en alguna manera, ya sea porque se le agregó humor, o palabras aleatorias que no parecen ni un poco creíbles. Si vas a utilizar un pasaje de Lorem Ipsum, necesitas estar seguro de que no hay nada avergonzante escondido en el medio del texto. Todos los generadores de Lorem Ipsum que se encuentran en Internet tienden a repetir trozos predefinidos cuando sea necesario, haciendo a este el único generador verdadero (válido) en la Internet. Usa un diccionario de mas de 200 palabras provenientes del latín, combinadas con estructuras muy útiles de sentencias, para generar texto de Lorem Ipsum que parezca razonable. Este Lorem Ipsum generado siempre estará libre de repeticiones, humor agregado o palabras no características del lenguaje, etc.

Motocicletas "La veloz"



Autos "de papá"



Televisores "Qué bien se ve"



El módulo Flexible Box

Ejemplo

```

<body>
<div id="pagina">
  <header id="encabezado"><h1>Encabezado de la página</h1></header>
  <nav id="menu"> <ul> <li>Página 1</li> <li>Página 2</li> <li>Página 3</li>
    <li>Página 4</li> <li>Página 5</li> <li>Página 6</li> </ul> </nav>
  <div id="contenidos">
    <article class="articulo"> <h1>¿Qué es Lorem Ipsum?</h1>
      <p>Lorem Ipsum es ...</p> </article>
    <article class="articulo"> <h1>De donde viene</h1>
      <p>Al contrario del pensamiento popular ...</p> </article>
    <article class="articulo"> <h1>¿Dónde puedo conseguirlo?</h1>
      <p>Hay muchas variaciones ... </p> </article>
  </div>
  <aside id="anuncios">
    <article class="anuncio"> <p>Motocicletas "La veloz" </p>
       </article>
    <article class="anuncio"> <p>Autos "de papá" </p>
       </article>
    <article class="anuncio"> <p>Televisores "Qué bien se ve" </p>
       </article> </aside>
  <footer id="pie">Pie de página</footer>
</div> </body>

```

El módulo Flexible Box

Ejemplo

```

/****** Disposición global de la página */
#pagina {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}

#pagina #encabezado,
#pagina #pie {
  width: 100%;
}

#pagina #anuncios {
  width: 160px;
}

#pagina #menu {
  width: 140px;
}

#pagina #contenidos {
  flex-basis: 10%;
  flex-grow: 1;
}

/****** Disposición de contenidos */
#pagina #contenidos {
  display: flex;
}

#pagina #contenidos #articulo {
  flex-grow: 1;
}

```



El módulo Flexible Box

Ejemplo

Si disminuye el tamaño de la página ...

Encabezado de la página

Página 1	¿Qué es Lorem Ipsum?	De donde viene	¿Dónde puedo conseguirlo?
Página 2	<p>Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. Tiene sus raíces en una pieza clásica de la literatura del Latín, que data del año 45 antes de Cristo, haciendo que este</p>	<p>esas imágenes. La primera linea del Lorem Ipsum, "Lorem ipsum dolor sit amet, ", viene de una linea en la sección 1.10.32</p>	<p>Hay muchas variaciones de los pasajes de Lorem Ipsum disponibles, pero la mayoría sufrió alteraciones en alguna manera, ya sea porque se le agregó humor, o palabras aleatorias que no sirven para nada.</p> <p>http://es.lipsum.com</p>
Página 3			
Página 4			
Página 5			
Página 6			

Motocicletas "La veloz"

Autos "de papá"

Televisores "Qué bien se ve"

Pie de página

Encabezado de la página

Página 1	¿Qué es Lorem Ipsum?	De donde viene	¿Dónde puedo conseguirlo?
Página 2	<p>Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. Tiene sus raíces en una pieza clásica de la literatura del Latín, que data del año 45 antes de Cristo, haciendo que este</p>	<p>esas imágenes. La primera linea de una linea en la sección 1.10.32</p>	<p>Hay muchas variaciones de los pasajes de Lorem Ipsum disponibles, pero la mayoría sufrió alteraciones en alguna manera, ya sea porque se le agregó humor, o palabras aleatorias que no sirven para nada.</p>
Página 3			
Página 4			
Página 5			
Página 6			

Motocicletas "La veloz"

Autos "de papá"

Televisores "Qué bien se ve"

Pie de página



El módulo Flexible Box

Ejemplo

Si quitamos las reglas que controlan el módulo Flexbox la página es apta para visualización en pantallas pequeñas

Las insertamos en un media query:

```
@media (min-width: 800px) {
    /* Reglas Flexbox
    ...
}
}
```

¿media query? ... lo vemos más adelante

Encabezado de la página

Página 1	¿Qué es Lorem Ipsum?	De donde viene	¿Dónde puedo conseguirlo?
Página 2	<p>Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. Tiene sus raíces en una pieza clásica de la literatura del Latín, que data del año 45 antes de Cristo, haciendo que este</p>	<p>esas imágenes. La primera linea de una linea en la sección 1.10.32</p>	<p>Hay muchas variaciones de los pasajes de Lorem Ipsum disponibles, pero la mayoría sufrió alteraciones en alguna manera, ya sea porque se le agregó humor, o palabras aleatorias que no sirven para nada.</p>
Página 3			
Página 4			
Página 5			
Página 6			

Motocicletas "La veloz"

Autos "de papá"

Televisores "Qué bien se ve"

Pie de página

El módulo Flexible Box

Resumen

Contenedor flex

```
display: flex;
/* Sentido de la distribución */
flex-direction: row
            column
            row-reverse
            column-reverse

/* Alineación en principal */
justify-content: flex-start
                flex-end
                center
                space-between
                space-around

/* Alineación en secundario */
align-items: stretch
                flex-start
                flex-end
                center

/* Usar varias filas/columnas */
flex-wrap: wrap
            wrap-reverse
```

Elemento flex

```
/* Orden de presentación */
order: <int>;
```



```
/* Alineación en secundario */
align-self: stretch
            flex-start
            flex-end
            center
```



```
/* Tamaño recomendado */
flex-basis: cant;
```



```
/* Distribuir espacio sobrante */
flex-grow: prop;
```



```
/* Distribuir pérdidas*/
flex-shrink: prop;
```

Tecnologías Web

3º Grado en Ingeniería Informática

Diseño y desarrollo de páginas web



UNIVERSIDAD
DE GRANADA

- 1. Introducción
- 2. Estructura y contenido de la página: HTML
- 3. Diseño de la página: CSS
- 1. Fundamentos
- 2. Distribución de los elementos (layout)
- 3. El módulo Flexible box
- 4. El módulo Grid
- 5. Políticas de diseño de la página
- 6. Diseño adaptable
- 7. Bibliografía

»»

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

DECSAI

CSS: Grid

Elementos del grid

El módulo Grid

Disponible en CSS3
Distribución 2D de elementos

Conceptos:

- **Contenedor** grid. Elemento que contiene el grid completo
- **Elemento** grid. Descendientes directos del contenedor
- **Línea** grid. Líneas divisorias de filas y columnas
- “**Track**” grid. Espacio entre dos líneas adyacentes
- **Celda** grid. Espacio entre dos líneas horizontales y dos líneas verticales
- **Área** grid. Espacio entre dos líneas horizontales y dos líneas verticales

Febrero 2018

IE	Edge	*	Firefox	Chrome	Safari
				49	
				62	
11	16		57	63	
17			58	64	11
				65	11.1

Soporte reciente y limitado

- Firefox V54, junio 2017
- Chrome V58, abril 2017
- IE: no tiene soporte completo

<http://caniuse.com>

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 61

DECSAI

CSS: Grid

El elemento contenedor

El contenedor

Define la forma del grid

display: grid | grid-inline Define un elemento como un contenedor grid

grid-template-rows
grid-template-columns
grid-template-areas Definen la forma del grid

```
.contenedor {
  grid-template-columns: 40px 50px auto 50px 40px;
  grid-template-rows: 25% 100px auto;
}
```

Las líneas horizontales y verticales se autonumeran 1, 2, ...

Los ejemplos de esta sección obtenidos de <https://css-tricks.com/snippets/css/complete-guide-grid/>

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 62



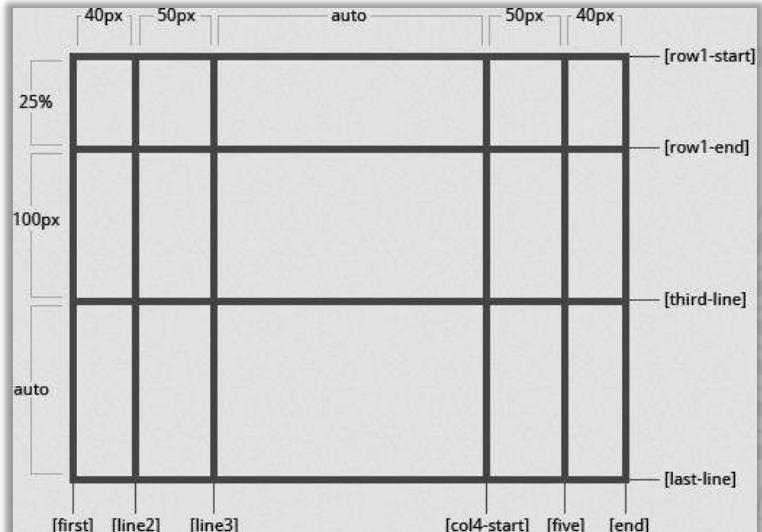
El contenedor

Dando nombre a las líneas separadoras de celdas

```
.contenedor {
  grid-template-columns: [first] 40px [line2] 50px [line3] auto [col4-start] 50px [five] 40px [end];
  grid-template-rows: [row1-start] 25% [row1-end] 100px [third-line] auto [last-line];
}
```

[nom1 nom2] se permiten varios nombres para una línea

Se permite repetir nombres en distintas líneas

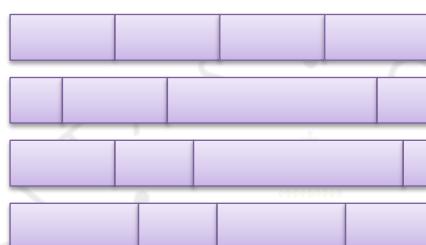


Fracciones de tamaño

El tamaño de las filas/columnas puede ser proporcional al tamaño del grid

fr=fracción

```
grid-template-columns: 1fr 1fr 1fr 1fr;
grid-template-columns: 1fr 2fr 4fr 1fr;
grid-template-columns: 1fr 150px 2fr 50px;
grid-template-columns: 1fr 150px 1fr 20%;
```





Definición de áreas

Define áreas sobre el grid y les asigna nombre
Permite asociar elementos HTML a áreas del grid

```
.contenedor {
    grid-template-columns: 50px 50px 50px 50px;
    grid-template-rows: auto;
    grid-template-areas:
        "header header header header"
        "main main . sidebar"
        "footer footer footer footer";
}
```

. → celdas sin asignar a área
(puede ser más de un punto)

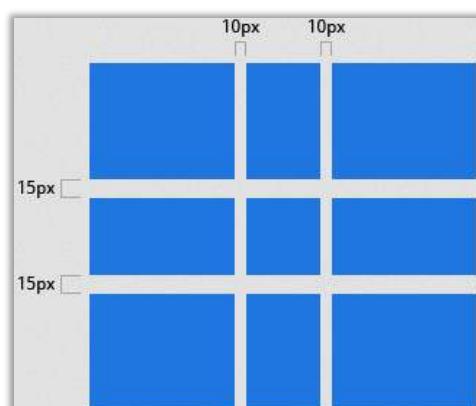
```
#elem {
    grid-area: header;
}
```

Posiciona un elemento #elem en el área header



Separación entre filas y columnas

```
.contenedor {
    grid-template-columns: 100px 50px 100px;
    grid-template-rows: 80px auto 80px;
    grid-column-gap: 10px;
    grid-row-gap: 15px;
}
```



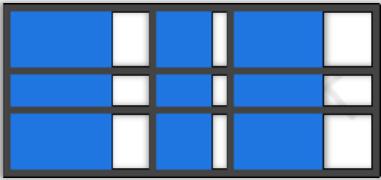
CSS: Grid
Alineación de los elementos del grid

Alineación y justificación de los elementos del grid

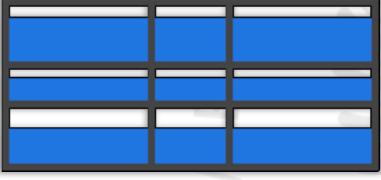
justify-items: justificación del contenido de los elementos (en horizontal)
align-items: justificación del contenido de los elementos (en vertical)

start | end | center | stretch

```
.contenedor {
  justify-items: start;
}
```



```
.contenedor {
  align-items: end;
}
```



Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 67

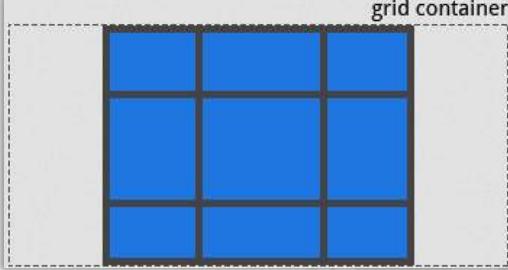
CSS: Grid
Alineación del grid

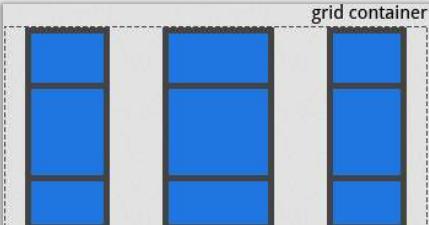
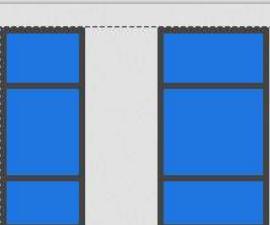
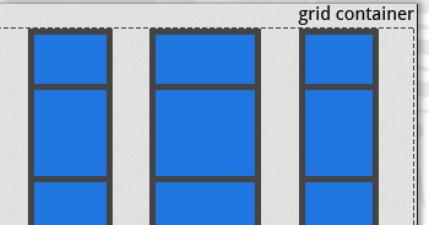
Alineación y justificación del grid dentro del contenedor

justify-content: justificación del grid (en horizontal)
align-content: justificación del grid (en vertical)

start | end | center | stretch | space-around | space-between | space-evenly

```
.contenedor {
  justify-content: center;
}
```



space-around	space-between	space-evenly
		

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 68



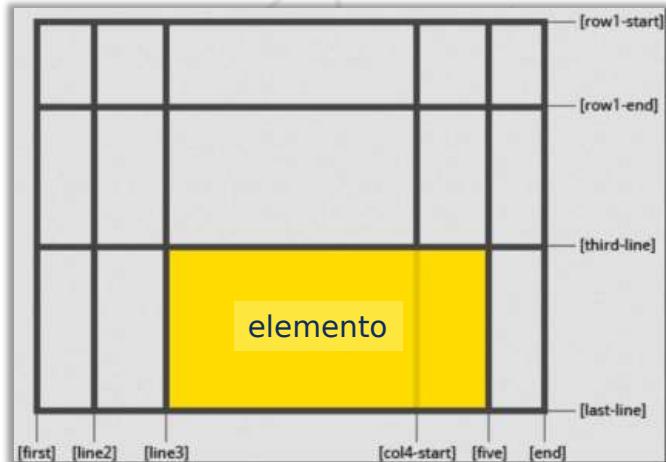
Los elementos del grid

Una vez definido el grid → posicionamos los elementos en él a partir de las líneas definidas

```
grid-column: línea_de_inicio / (línea_de_fin | span número_de_líneas)
grid-row: línea_de_inicio / (línea_de_fin | span número_de_líneas)
```

Las líneas pueden referenciarse con números y con nombres

```
#elemento {
    grid-column: 3 / span 2;
    grid-row: third-line / 4;
}
```



Los elementos del grid

Una vez definido el grid → posicionamos los elementos en él a partir de las áreas definidas (o definiendo nuevas áreas)

```
grid-area: nombre_del área
grid-area: lin_fila_ini / lin_col_ini / lin_fila_fin / lin_col_fin
```

```
.contenedor {
    grid-template-columns: 50px 50px 50px 50px;
    grid-template-rows: auto;
    grid-template-areas:
        "header header header header"
        "main main . sidebar"
        "footer footer footer footer";
}

#elemento1 {
    grid-area: header;
}

#elemento2 {
    grid-area: 2 / 1 / 4 / 3;
}
```



CSS: Grid
Alineación de elementos individuales

Alineación o justificación de elementos individuales del grid

justify-self: justificación del contenido de un elemento (en horizontal)
align-self: justificación del contenido de un elemento (en vertical)

start | end | center | stretch

```
.elem1 {
  align-self: end;
}
```

```
.el2 {
  justify-self: center;
}
```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 71

CSS: Grid
... y más

Posicionamiento automático y elementos implícitos

- Si algún elemento no se posiciona de forma explícita → algoritmo automático
- Si se añaden elementos fuera del grid definido → se extiende el grid

```
.container {
  grid-template-columns: 60px 60px;
  grid-template-rows: 90px 90px
}
```

```
.item-a {
  grid-column: 1 / 2;
  grid-row: 2 / 3;
}

.item-b {
  grid-column: 5 / 6;
  grid-row: 2 / 3;
}
```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 72



UNIVERSIDAD
DE GRANADA

Tecnologías Web

3º Grado en Ingeniería Informática

Diseño y desarrollo de páginas web

1. Introducción

2. Estructura y contenido de la página: HTML

3. Diseño de la página: CSS

1. Fundamentos

2. Distribución de los elementos (layout)

1. Modelo de cajas (box model)

2. Posicionamiento de los elementos

3. El módulo Flexible box

4. El módulo Grid

5. Políticas de diseño de la página

6. Diseño adaptable

7. Bibliografía

»»»

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena



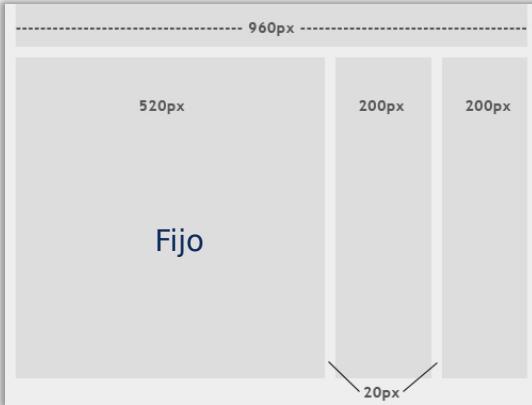
CSS: Posicionamiento

Layouts: distribución de contenidos

Estrategias de distribución de contenidos

Hay distintos enfoques para el diseño de webs dependiendo de la estrategia para distribuir sus contenidos (layouts):

- Fijo. Se impone un ancho fijo en el contenedor de la página (en px). Otros componentes pueden dimensionarse en px, en %, ...
- Fluido/Líquido. Los anchos se dan en porcentajes. Se ajustan al tamaño de la pantalla.
- Elástico. Los anchos se expresan en función del tamaño de letra (em)
- Híbrido

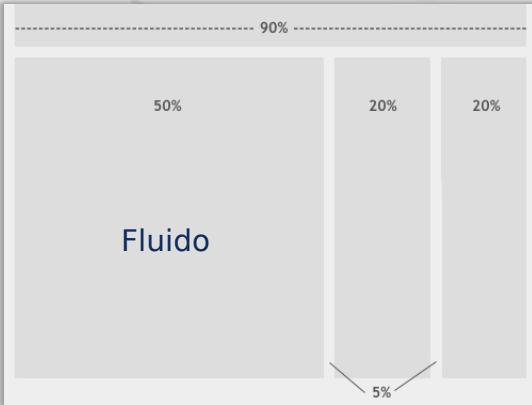


960px

520px 200px 200px

Fijo

20px



90%

50% 20% 20%

Fluido

5%

<https://www.smashingmagazine.com/2009/06/fixed-vs-fluid-vs-elastic-layout-whats-the-right-one-for-you/>

CSS: Posicionamiento
Layouts: distribución de contenidos

	Ventajas	Inconvenientes
Fijo	<ul style="list-style-type: none"> Diseño fácil Mismo tamaño en cualquier navegador En algunos diseños es necesario (p. ej. con muchos gráficos) 	<ul style="list-style-type: none"> Pantallas anchas: más margen vacío Pantallas pequeñas: barra de scroll horizontal
Fluido	<ul style="list-style-type: none"> Más agradable para usuario La proporción de márgenes vacíos se mantiene Sin barras de scroll horizontal Aprovecha mejor la pantalla 	<ul style="list-style-type: none"> Al diseñar hay menos control sobre el aspecto final Las imágenes y videos deben acomodarse a distintos tamaños Pantallas muy anchas: poco estético, menos usabilidad (mucha información)
Elástico	<ul style="list-style-type: none"> Más agradable para el usuario: mantiene proporciones El tamaño varía según las preferencias del usuario 	<ul style="list-style-type: none"> Al diseñar: probar todas las posibilidades es tedioso Si se aumenta mucho el tamaño ...

CSS: Posicionamiento
Layouts: distribución de contenidos

Ejemplo de layout fijo

Ejemplo de layout fijo

The screenshot shows a fixed layout website. At the top is a header with the 'DECSAI' logo and the text 'CSS: Posicionamiento' and 'Layouts: distribución de contenidos'. Below the header is a main content area with a grid of news cards. Each card has a small thumbnail image, a title, and a brief description. To the right of the main content area is a sidebar with more news cards.

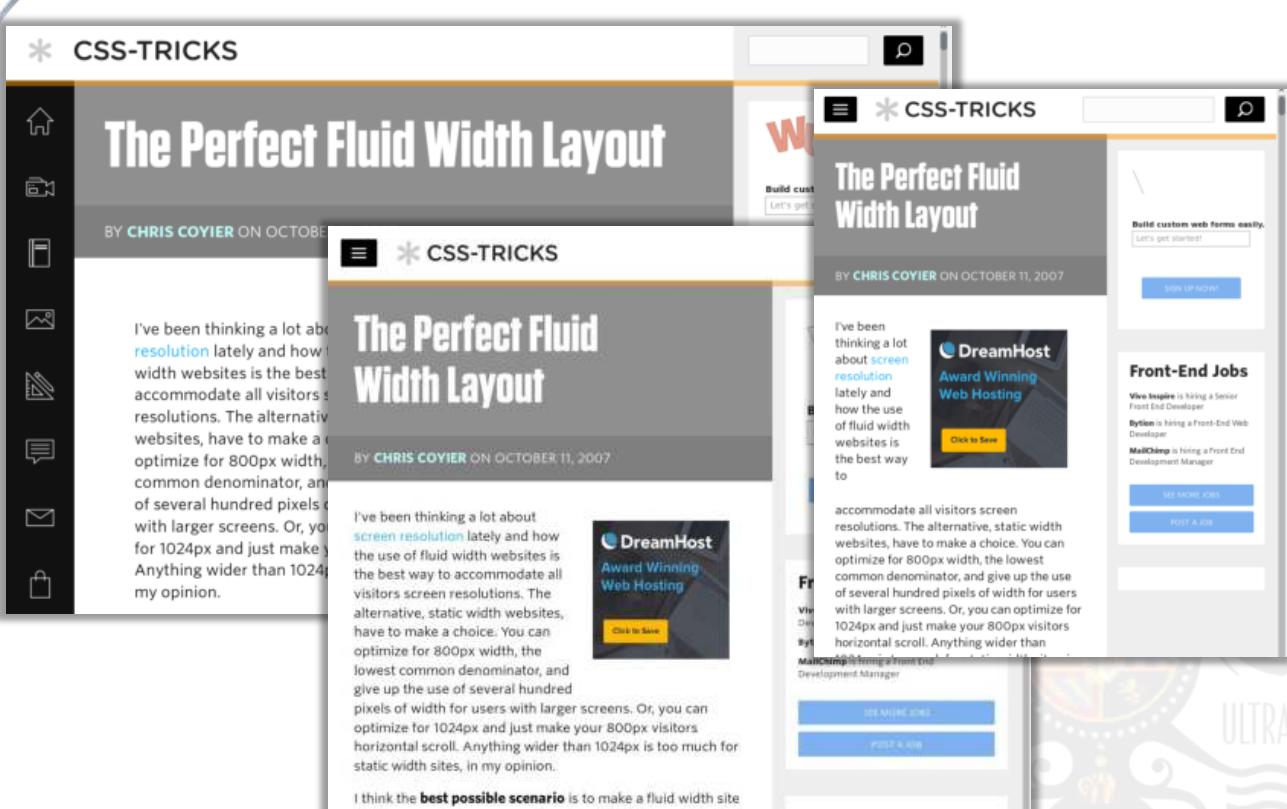
Ejemplo de layout fluido

The screenshot shows a fluid layout website. It compares two versions of a blog post. The left version is for 'OCTOBER 25, 2016' and the right version is for 'OCTOBER 25, 2016'. Both posts are titled 'So I switched to Android'. The content of the posts is identical, discussing the decision to switch from iOS to Android. To the right of each post is a sidebar with the heading 'ABOUT ME' and a bio for Andrew, mentioning his work at 'Evolve' and his role in the 'Genova WordPress Meetup'.

 DECSAI

CSS: Posicionamiento Layouts: distribución de contenidos

Ejemplo de layout elástico



The Perfect Fluid Width Layout

BY CHRIS COYIER ON OCTOBER 11, 2007

I've been thinking a lot about [screen resolution](#) lately and how fluid width websites is the best way to accommodate all visitors screen resolutions. The alternative, static width websites, have to make a choice. You can optimize for 800px width, the lowest common denominator, and give up the use of several hundred pixels of width for users with larger screens. Or, you can optimize for 1024px and just make your 800px visitors horizontal scroll. Anything wider than 1024px is too much for static width sites, in my opinion.

The Perfect Fluid Width Layout

BY CHRIS COYIER ON OCTOBER 11, 2007

I've been thinking a lot about [screen resolution](#) lately and how the use of fluid width websites is the best way to accommodate all visitors screen resolutions. The alternative, static width websites, have to make a choice. You can optimize for 800px width, the lowest common denominator, and give up the use of several hundred pixels of width for users with larger screens. Or, you can optimize for 1024px and just make your 800px visitors horizontal scroll. Anything wider than 1024px is too much for static width sites, in my opinion.

I think the **best possible scenario** is to make a fluid width site

DreamHost Award Winning Web Hosting Click to Learn More

Front-End Jobs

Vivo Inspire is hiring a Senior Front End Developer

Byton is hiring a Front-End Web Developer

MailChimp is hiring a Front-End Development Manager

SEE MORE JOBS POST A JOB

ULTRA

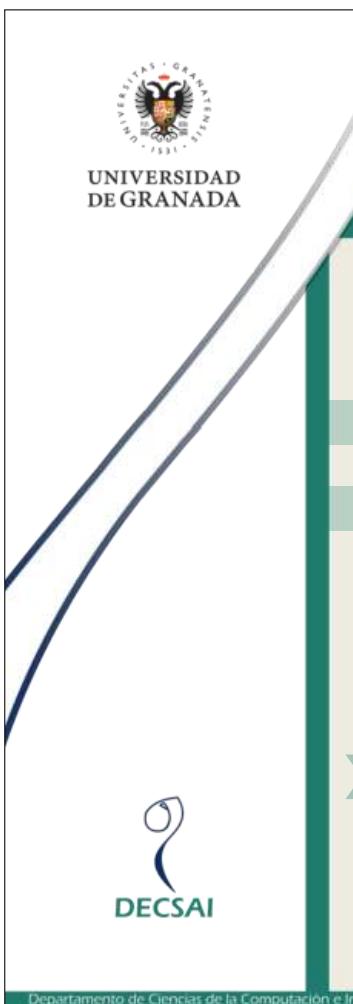
Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 79

 UNIVERSIDAD DE GRANADA

Tecnologías Web

3º Grado en Ingeniería Informática

Diseño y desarrollo de páginas web



1. Introducción

2. Estructura y contenido de la página: HTML

3. Diseño de la página: CSS

- 1. Fundamentos**
- 2. Distribución de los elementos (layout)**
- 3. El módulo Flexible box**
- 4. El módulo Grid**
- 5. Políticas de diseño de la página**
- 6. Diseño adaptable**
- 7. Bibliografía**

»»»

 DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena

CSS: diseño adaptable Objetivo

Diseño adaptable/flexible (*responsive design*)

Permite adaptar una página a la pantalla del dispositivo en que se visualiza



Se basa en

- Esquema de distribución fluida (Fluid layout)
- CSS media queries
- meta viewport (introducido en HTML5 y bien soportado)
@viewport (definido en CSS2.1 pero poco soportado)

Problemas:

- Tamaños de pantalla
- Resoluciones de dispositivos

CSS: diseño adaptable Comportamiento por defecto

decsai.ugr.es
Ancho de 1024 px



El browser móvil suele escalar por defecto

320px / 5cm
162 dpi



320px / 5cm
162 dpi



1080px / 6.2cm
445 dpi

CSS: diseño adaptable**Viewport****Definición de viewport**

Es la zona de la pantalla donde se renderiza la imagen del navegador (lienzo de dibujo)

**Layout viewport**

- La página se renderiza al tamaño del layout viewport
- El navegador escala el resultado para que se vea en la pantalla del móvil
- Si no se escala hay que hacer scrolling

CSS: diseño adaptable**Viewport****Control del tamaño del viewport**

```
<meta name="viewport" content="width=TAMAÑO_EN_PIXELS">
```

```
<meta name="viewport" content="width=320">
```



<https://dev.opera.com/articles/an-introduction-to-meta-viewport-and-viewport/>

CSS: diseño adaptable
Viewport

Lo habitual ... ajustar el ancho al tamaño físico de la pantalla

```
<meta name="viewport" content="width=device-width">
```

```
<meta name="viewport" content="width=device-width">
```

<https://dev.opera.com/articles/an-introduction-to-meta-viewport-and-viewport/>

CSS: diseño adaptable
Viewport

```
<meta name="viewport" content="width=device-width">
```

```
<body>
<div id="wrapper">
<h1>Alan Turing</h1>
<p>Alan Mathison Turing, OBE (1912-1954), fue un matemático, lógico, científico de la computación, criptógrafo, filósofo, maratoniano y corredor de ultra distancia británico.</p>
<p id="par">Es considerado uno de los padres de la ciencia de la computación</p> y precursor de la informática moderna. Proporcionó una influyente formalización de los conceptos de algoritmo y computación: la <em>máquina de Turing</em>. Formuló su propia versión de la hoy ampliamente aceptada tesis de Church-Turing (1936).</p>
...
#wrapper {
  width: 100%;
}
#par {
  margin: auto; /* centrado */
  width: 300px;
  border: 1px solid black;
}
```

1080px / 6.2cm
445 dpi

CSS: diseño adaptable

Píxeles CSS vs píxeles físicos

CSS píxeles

El lienzo sobre el que dibuja el navegador está formado por CSS-píxeles (unidad abstracta), que no coinciden con los píxeles reales del dispositivo.

- Dependiendo del dispositivo varía la relación CSS/device pixel

Dispositivo	Pulgadas	Ancho físico	Ancho CSS	Densidad
iPhone 3GS	3.5	320	320	1
iPhone 4	3.5	640	320	2
iPhone X	5.8	1125	375	3
Nexus 4	4.7	768	384	2
Nexus 5	5.0	1080	360	3
S. Galaxy S6	5.1	1440	360	4
iPad	9.7	768	768	1
iPad Retina	9.7	1536	768	2

<https://material.io/devices/>

CSS: diseño adaptable
Media types/queries**Control del aspecto en función del dispositivo**

- Media types: permiten definir estilos dependiendo del tipo de dispositivo en el que se visualiza la página (pantalla, impresora, etc)
- Media queries: (media type + condiciones) para definir estilos en función de las capacidades de la pantalla

Media types: screen, print, braille, tv, ...

```
@media print {
    /* Código CSS con estilos en caso de renderizar en una impresora */
    h1 {
        color: red;
    }
}

@media screen {
    /* Código CSS con estilos en caso de renderizar en una pantalla */
    h1 {
        color: green;
    }
}
```

CSS: diseño adaptable

Media queries

Media queries

Consultas sobre características del dispositivo:

- width, max-width, min-width
- height, max-height, min-height
- orientation, resolution, aspect-ratio
- ...

```
<!-- Inclusión de CSS condicionada en HTML -->
<link rel="stylesheet" media="screen and (min-width:500px)" href="estilo.css" />

@media (min-width:500px) {
  ...
}

@media (orientation: portrait) {
  ...
}

@media screen and (device-aspect-ratio: 16/9) {
  ...
}
```

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

CSS: diseño adaptable

Media queries

Media queries

```
@media screen and (min-width:600px) and (max-width:1000px) {
  ...
}

@media screen and (width:600px), screen and (width:800px) {
  ...
}
```

Uso habitual: distinguir tipo de dispositivo

```
/* Móviles (portrait) */
@media screen and (max-device-width: 480px) and (orientation: portrait) { ... }

/* Móviles (landscape) */
@media screen and (max-device-width: 640px) and (orientation: landscape) { ... }

/* Móviles (portrait/landscape) */
@media screen and (max-device-width: 640px) { ... }

/* Tabletas */
@media screen and (min-device-width: 768px) and (max-device-width: 1024px) { ... }

/* Desktop */
@media screen and (min-width: 1024px) { ... }
```

CSS: diseño adaptable**Ejemplo**

```

body {
    text-align: center;
}

.container {
    background: lightgray;
    overflow: auto;
    padding: 10px;
    margin: auto;
    max-width: 600px;
}

```

```

section {
    background: bisque;
    margin: 1%;
    padding: 1%;
}

aside {
    background: aquamarine;
    margin: 1%;
    padding: 1%;
}

```

```

@media all and (min-width: 400px) {
    section {
        float: left;
        width: 55%;
    }

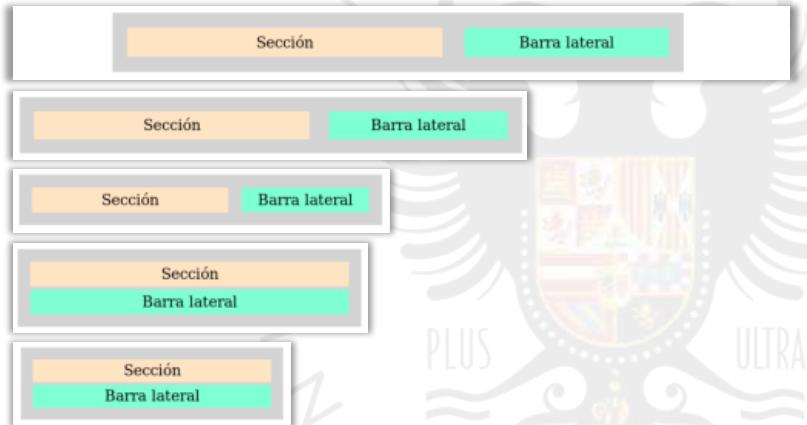
    aside {
        float: right;
        width: 35%;
    }
}

```

```

<body>
    <div class="container">
        <section>Sección</section>
        <aside>Barra lateral</aside>
    </div>
</body>

```

**CSS: diseño adaptable****Estrategias****Estrategias de diseño**

- Desktop first. Diseño general pensando en tamaño desktop y luego adaptar a tamaños menores
- Mobile first. Diseño general pensando en pantallas pequeñas y luego adaptar a tamaños mayores.

Mobile first:

- Diseño general para tamaños pequeños
- Añadir media queries con min-width

Desktop first:

- Diseño para tamaño grande
- Añadir media queries con max-width

Ventajas de mobile first:

- Es más fácil añadir contenidos que reducir contenidos
- Las páginas cargan más rápido en móviles
 - Hay menos estilos que cargar
- El flujo normal de HTML favorece diseños first mobile
 - Por defecto los bloques ocupan el 100%
 - No hay que "deshacer" estilos

Estrategias de diseño

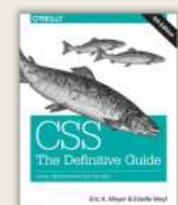
- El número de media queries no debe ser muy elevado
 - Móvil, tableta, desktop
- Puede ocurrir que el navegador cargue todo el contenido CSS de una media query aunque no la aplique.
- Determinar los puntos de cambio de hoja de estilo (breakpoints)
- El orden de las media queries es importante en caso de que las condiciones no sean completamente excluyentes

Diseño adaptable

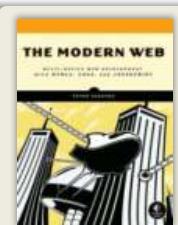
- Esquema de distribución fluida (Fluid layout)
- CSS media queries
- meta viewport

Diseño y desarrollo de páginas web - CSS

Bibliografía

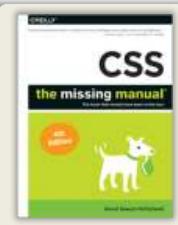


Eric Meyer, Estelle Weyl
CSS: The definitive guide (4th ed)
 O'Reilly. 2017



Peter Gasston
The Modern Web.
Multi-device web development with HTML5, CSS3, and JavaScript
 No Starch Press. 2013

<https://www.nostarch.com/modernweb>



David Sawyer McFarland
CSS: The missing manual (4 ed)
 O'Reilly. 2015

- <https://www.w3.org/Style/CSS/>
- <http://www.w3schools.com/css/>
- <http://www.alistapart.com/topics/code/css>
- <http://css-tricks.com>