



UNIVERSIDAD DE CÓRDOBA

## PROGRAMACIÓN WEB – BLOQUE III.2

# CSS\_ Modelo de caja y posicionamiento

Dr. José Raúl Romero Salguero  
jrromero@uco.es

{CSS}



# Contenidos

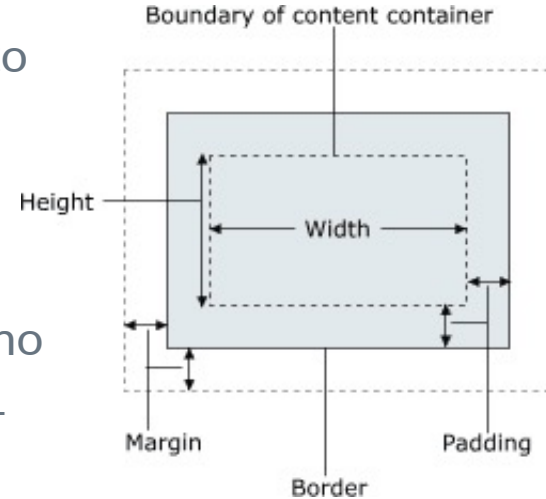
1. Aspectos básicos
2. Mecanismos de selección
3. Guía de diseño
4. Texto y colores
5. Fondos
6. El modelo de caja
7. Posicionamiento

6.

# El modelo de caja

# El modelo de caja de CSS

- Para propósitos de diseño, cada elemento está compuesto de:
  - ❑ El contenido del elemento
  - ❑ Un **border** alrededor del elemento
  - ❑ **padding** entre el contenido y el **border** (*inside*)
  - ❑ Un **margin** entre el **border** y otro contenido externo
- **width** = content-width + L/R-padding + L/R-border + L/R-margin
- **height** = content-height + T/B-padding + T/B-border + T/B-margin



# Dimensiones en CSS

```
p { width: 350px; background-color: yellow; }  
h2 { width: 50%; background-color: aqua; }
```

Este párrafo utiliza el estilo definido arriba

**Cabecera h2**

Propiedad	Descripción
<u>width</u> , <u>height</u>	ancho o alto (solo elementos de bloque)
<u>max-width</u> , <u>max-height</u> , <u>min-width</u> , <u>min-height</u>	tamaño máx/mín del elemento en una dimension dada

# Bordes, márgenes y relleno

Referencia modelo de caja - [http://www.w3schools.com/css/css\\_boxmodel.asp](http://www.w3schools.com/css/css_boxmodel.asp)

## ➤ **border:**

- ❑ Cada caja tiene un **border**, aunque sea invisible o especificado para ser 0 **pixels** (px) de ancho
- ❑ Separa el borde de una caja de otra

## ➤ **margin:**

- ❑ Se localiza fuera de la frontera del borde
- ❑ Puede establecer el **margin** width para crear un espacio entre el **border** de dos cajas adyacentes

## ➤ **padding:**

- ❑ El espacio entre el **border** de una caja y su contenido
- ❑ Agregar **padding** aumenta la legibilidad del contenido

# Bordes en CSS

```
h2 { border: 5px solid red; }
```

## Cabecera de tipo H2

Propiedad	Descripción
<u><a href="#">border</a></u>	grosor / estilo / color del borde en los 4 lados

- **thickness** (especificado en px, pt, em, o thin, medium, thick).
- **style** (none, hidden, dotted, dashed, double, groove, inset, outset, ridge, solid).
- **color** (especificado como se vio anteriormente para el texto y los colores de fondo).

# Bordes en CSS

Propiedad	Descripción
<u><a href="#">border-color</a></u> , <u><a href="#">border-width</a></u> , <u><a href="#">border-style</a></u>	propiedades específicas del borde en los 4 lados
<u><a href="#">border-bottom</a></u> , <u><a href="#">border-left</a></u> , <u><a href="#">border-right</a></u> , <u><a href="#">border-top</a></u>	todas las propiedades del borde en un lado particular
<u><a href="#">border-bottom-color</a></u> , <u><a href="#">border-bottom-style</a></u> , <u><a href="#">border-bottom-width</a></u> , <u><a href="#">border-left-color</a></u> , <u><a href="#">border-left-style</a></u> , <u><a href="#">border-left-width</a></u> , <u><a href="#">border-right-color</a></u> , <u><a href="#">border-right-style</a></u> , <u><a href="#">border-right-width</a></u> , <u><a href="#">border-top-color</a></u> , <u><a href="#">border-top-style</a></u> , <u><a href="#">border-top-width</a></u>	propiedades de borde en un lado particular



# Bordes en CSS\_\_ ejemplo

```
h2 {  
  border-left: thick dotted #CC0088;  
  border-bottom-color: rgb(0, 128, 128);  
  border-bottom-style: double;  
}
```

**This is a heading.**

- Las propiedades del ***border*** de cada lado se pueden configurar individualmente.
- Si se omite alguna propiedad, recibe los valores predeterminados (por ej., ***border-bottom-width*** arriba)

# Bordes en CSS\_\_ border-radius

```
p {  
  border: 3px solid blue;  
  border-radius: 12px;  
  padding: 0.5em;  
}
```

Esto es un párrafo

Esto es otro párrafo.  
Ocupa multiples líneas.

- El **radio del borde** de cada lado se puede establecer individualmente, separando por espacios.

# Márgenes en CSS

Propiedad	Descripción
<u><a href="#">margin</a></u>	margen en los 4 lados
<u><a href="#">margin-bottom</a></u>	margen solo en la parte inferior
<u><a href="#">margin-left</a></u>	margen solo en la parte izquierda
<u><a href="#">margin-right</a></u>	margen solo en la parte derecho
<u><a href="#">margin-top</a></u>	margen en solo en la parte superior
<u><a href="https://www.w3schools.com/cssref/pr_margin.asp">https://www.w3schools.com/cssref/pr_margin.asp</a></u>	

# Márgenes en CSS\_\_ ejemplo

```
p {  
  margin: 50px;  
  background-color: fuchsia;  
}
```

Este es el primer párrafo

Este es el segundo párrafo

Hay que considerar que **los márgenes son siempre transparentes** (no contienen el color de fondo del elemento, etc.)

# Márgenes en CSS\_\_ ejemplo

```
p {  
  margin-left: 8em;  
  background-color: fuchsia;  
}
```

Este es el primer párrafo

Este es el segundo párrafo

El margen de cada lado se puede configurar individualmente

# Relleno en CSS

propiedad	descripción
<a href="#"><u>padding</u></a>	padding en los 4 lados
<a href="#"><u>padding-bottom</u></a>	padding solo en la parte inferior
<a href="#"><u>padding-left</u></a>	padding solo en la parte izquierda
<a href="#"><u>padding-right</u></a>	padding solo en la parte derecha
<a href="#"><u>padding-top</u></a>	padding solo en la parte superior
<a href="https://www.w3schools.com/cssref/pr_padding.asp"><u>https://www.w3schools.com/cssref/pr_padding.asp</u></a>	

# Relleno en CSS\_\_ ejemplo

```
p { padding: 20px; border: 3px solid black; }  
h2 { padding: 0px; background-color: yellow; }
```

Este es el primer párrafo

Este es el segundo párrafo

**Esto es la cabecera**

# Relleno en CSS\_\_ ejemplo

```
p {  
  padding-left: 200px; padding-top: 30px;  
  background-color: fuchsia;  
}
```

Este es el primer párrafo

Este es el segundo párrafo

- El ***padding*** de cada lado se puede configurar individualmente
- Obsérvese que el ***padding*** comparte el color de fondo del elemento



# Centrado de elemento de bloque: márgenes automáticos

```
p {  
  margin-left: auto;  
  margin-right: auto;  
  width: 750px;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna aliqua.

- Para centrar elementos en línea dentro de un elemento de bloque:  
**text-align: center;**
- **NOTA:** Funciona mejor si se establece el ancho, de lo contrario puede ocupar todo el ancho de la página

# Cambio de elementos de bloque e *inline*

- **display**: permite mostrar un elemento en línea como si fuera un elemento de nivel de bloque y viceversa. También se puede usar para ocultar un elemento.

[http://www.w3schools.com/css/css\\_display\\_visibility.asp](http://www.w3schools.com/css/css_display_visibility.asp)

```
li {  
    display: inline;  
    margin-right: 10px;}  
li.future {  
    display: none;}
```

- Valores **display**:

- ☐ **inline**: hace que un elemento a nivel de bloque actúe como un elemento en línea
- ☐ **block**: hace que un elemento en línea actúe como un elemento de nivel de bloque
- ☐ **inline-block**: hace que un elemento a nivel de bloque fluya como un elemento en línea mientras conserva otras características de un elemento a nivel de bloque (por ejemplo, dimensiones)
- ☐ **none**: oculta el elemento de la página (frecuente con Javascript para ocultar elementos)

# Ocultación de cajas

- **visibility**: le permite ocultar cuadros a los usuarios pero deja un espacio donde habría estado el elemento

```
li {  
    display: inline;  
    margin-right: 10px;}  
li.future {  
    visibility: hidden;}
```

- Valores **visibility**:
  - ❑ **hidden**: oculta el elemento (pero deja un espacio en blanco en su lugar)
  - ❑ **visible**: muestra el elemento
- Si no desea mostrar el espacio en blanco, use **display** en lugar de **visibility**.

7.

# Posicionamiento

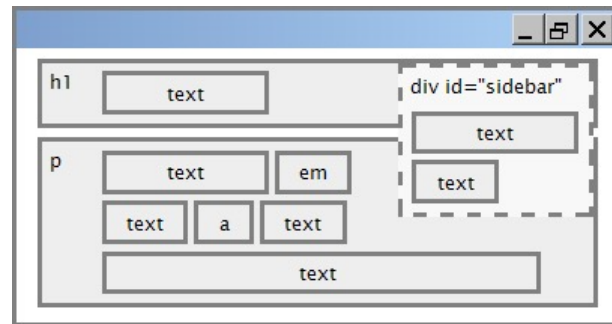


# Posicionamiento básico en CSS

# Posicionamiento en CSS\_ float

propiedad	descripción
float	lado sobre el que posicionarse; puede ser izquierda, derecha, o ninguno (por defecto)

- Un elemento flotante **se elimina del flujo** normal del documento
- El texto subyacente lo envolverá según sea necesario



# Posicionamiento en CSS\_ float

```
  
Lorem ipsum dolor sit amet, consectetur adipiscing elit....
```

```
img.imagen {  
    float: left;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam scelerisque purus ut dui mollis, sed malesuada leo pretium. Morbi bibendum mi at lacus rutrum convallis. Duis id eros dolor. In id eros blandit lectus viverra facilisis at commodo velit. Cras pretium nunc id nisl elementum, at interdum odio blandit. Donec luctus rutrum iaculis. Praesent luctus ante et cursus suscipit. Nullam congue egestas lorem nec luctus. Donec tincidunt tortor mi, nec ultricies orci bibendum a. Aliquam viverra metus nec ligula varius feugiat. In lacinia ligula accumsan tortor porttitor ornare. Donec interdum mattis purus sit amet ultrices.

# Posicionamiento en CSS\_ flotante con `width`

## Elemento flotante, sin ancho

flotante a derecha, sin ancho indicado

Esto es un flotante a la derecha, sin ancho indicado, pero es tan largo que ocupa el ancho completo y realmente no parece flotante

flotante, 45% width

flotante a la derecha, 45% width

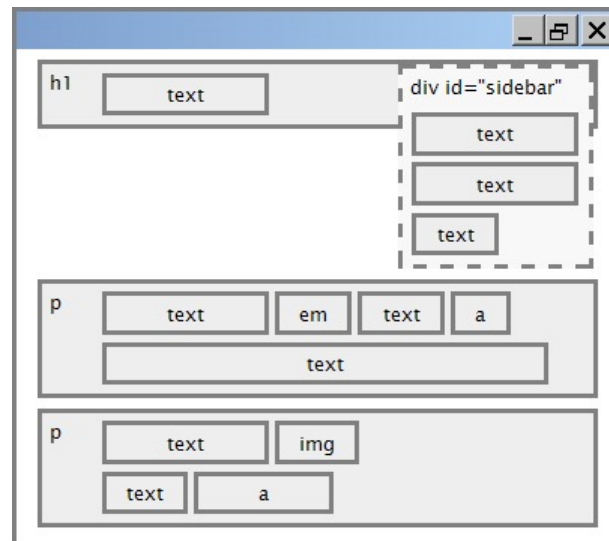
- A menudo los elementos flotantes **deben especificar un valor para `width`**
  - ❑ Si no se especifica el ancho, es posible que otro contenido no pueda ajustarse alrededor del elemento flotante



# Posicionamiento en CSS\_ `clear`

Propiedad	Descripción
<code>clear</code>	No permite que los elementos flotantes se superpongan a este elemento; puede ser izquierda, derecha, ambas o ninguna (por defecto)

```
div#sidebar { float: right; }  
p { clear: right; }
```



# Posicionamiento en CSS\_ desbordamiento

```
<p>  
Lorem ipsum dolor [...] diam velit.</p>
```

```
img { float: right; }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.



- Queremos que el elemento `p` conteniendo la imagen se extienda hacia abajo para que su borde encierre toda la imagen

# Posicionamiento en CSS\_ `overflow`

propiedad	descripción
<code>overflow</code>	especifica qué hacer si el contenido de un elemento es demasiado grande; puede ser automático, visible, oculto o desplazarse

# Posicionamiento en CSS\_ multicolumna

```
<div>  
  <p>Primer párrafo aquí</p>  
  <p>Segundo párrafo aquí</p>  
  <p>Tercer párrafo aquí</p>  
  Algo de texto que se quiera poner  
</div>
```

```
p { float: right; width: 20%; margin: 0.5em;  
    border: 2px solid black; }  
div { border: 3px dotted green; overflow: hidden; }
```

Algo de texto que se quiera poner

Tercer párrafo aquí

Segundo párrafo aquí

Primer párrafo aquí

# Posicionamiento en CSS\_ position

```
div#ad {  
  position: fixed;  
  right: 10%;  
  top: 45%;  
}
```

¡Aquí estoy!

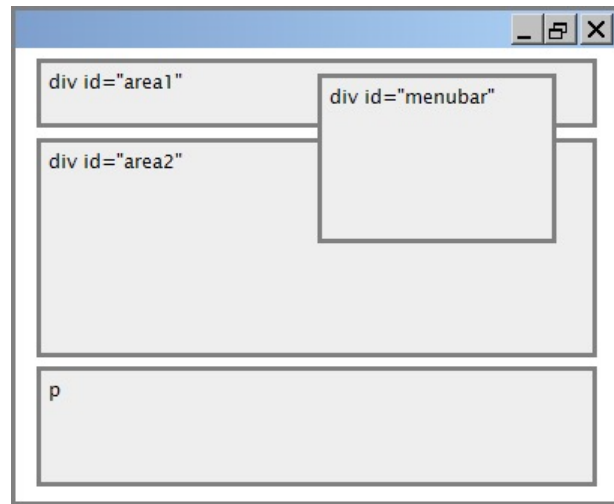
Propiedad	Valor	Descripción
position	static	Posición por defecto
	relative	Desplazamiento de su posición estática normal
	absolute	Una posición fija dentro de su elemento contenedor
	fixed	Una posición fija dentro de la ventana del navegador
<u>top</u> , <u>bottom</u> , <u>left</u> , <u>right</u>	Posiciones de las esquinas de la caja	



# Posicionamiento en CSS\_ *position* (*absolute*)

```
#menubar {  
    position: absolute;  
    left: 400px;  
    top: 50px;  
}
```

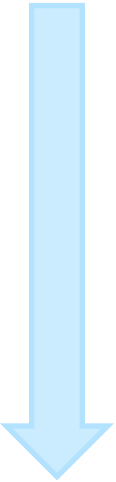
- Eliminado del flujo normal (como los flotantes)
- Posicionado respecto al elemento de bloque que los contiene (suponiendo que el bloque también use posicionamiento *absolute* o *relative*).
- Posición real determinada por los valores *top*, *bottom*, *left*, *right*
- A menudo también debe especificar una propiedad *width*







# Alineación Vs. Flotación Vs. Posicionamiento

- 
- Si es posible, diseñar un elemento **alineando** su contenido:
    - ❑ **Alineación horizontal:** `text-align`
      - Si se establece esto en un elemento de bloque; se alinea el contenido dentro de él (no el elemento de bloque en sí)
    - ❑ **Alineación vertical:** `vertical-align`
      - Si se establece esto en un elemento en línea, lo alinea verticalmente dentro de su elemento contenedor
  - Si la alineación no funciona, intentar hacer **flotante** el elemento
  - Si la flotación no funciona, intentar **posicionar** el elemento
    - ❑ El posicionamiento absoluto/fijo es el último recurso y no debe usarse en exceso

# Cajas *inline*\_ Consideraciones

- Las propiedades de tamaño (`width`, `height`, `min-width`, etc.) se ignoran para los cajas *inline*
  - ❑ `margin-top` y `margin-bottom` se ignoran, pero `margin-left` y `margin-right` no
- La propiedad `text-align` del cuadro de bloque contenedor controla la posición horizontal de los cuadros *inline* dentro de él
  - ❑ `text-align` no alinea elementos de bloque dentro de la página
- La propiedad `vertical-align` de cada cuadro *inline* lo alinea verticalmente dentro de su cuadro de bloque



# Programación Web

Presentación de la asignatura\_\_ Curso 2021/22