
Unidad 3 – Parte 1

Hojas de estilo en cascada: CSS y CSS3

LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE INFORMACIÓN

CFGs DESARROLLO DE APLICACIONES MULTIPLATAFORMA

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

ÍNDICE DE CONTENIDOS

1. **Introducción**
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

1. Introducción

- ✓ CSS es un lenguaje de hojas de estilos que permite definir el aspecto o presentación de documentos HTML.
- ✓ Es la mejor forma de separar el contenido y su presentación.
 - Al crear una página web, se utiliza en primer lugar el lenguaje HTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.
 - Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.
- ✓ Demostración: https://www.w3schools.com/css/demo_default.htm

1. Introducción

✓ Un poco de historia:

- Las hojas de estilos aparecieron poco después del lenguaje de etiquetas SGML (alrededor del año 1970). Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.
- El gran impulso de los lenguajes de hojas de estilos se produjo con el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. El W3C propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron:
 - CHSS (*Cascading HTML Style Sheets*)
 - SSP (*Stream-based Style Sheet Proposal*).
- La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (*Cascading Style Sheets*).

1. Introducción

✓ Un poco de historia:

- En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1".
- La adopción de CSS por parte de los navegadores ha requerido un largo periodo de tiempo e incluso a día de hoy el soporte completo de todos los aspectos de CSS no es proporcionado por ningún navegador:
[https://en.wikipedia.org/wiki/Comparison_of_browser_engines_\(CSS_support\)](https://en.wikipedia.org/wiki/Comparison_of_browser_engines_(CSS_support))

1. Introducción

✓ Versiones de CSS:

- **1996 □ CSS 1:** Propiedades para el tipo de letra, Colores de texto y de fondo, Alineación de texto, imágenes y tablas, Margen, borde, padding y posicionamiento de los elementos, ...
- **1998 □ CSS 2:** Mejoras en el posicionamiento de elementos, Sombras, Dirección del texto, ...
- **2011 □ CSS 2.1:** Corrige errores de la versión anterior, Suprime elementos que no fueron tenido en cuenta por los navegadores, Añade las extensiones de los navegadores, ...
- **CSS 3:** Ha evolucionado desde 1999: Esquinas redondeadas, Gradientes, Transiciones y animaciones, Nuevos layouts (Flex y Grid), Media-Queries (diseño responsive)

ÍNDICE DE CONTENIDOS

1. Introducción
- 2. Reglas CSS**
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

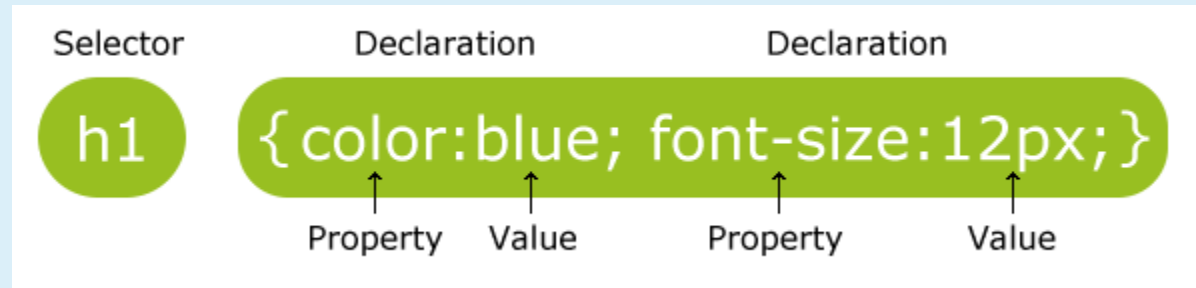
2. Reglas CSS

✓ Reglas CSS:

```
selector1 {  
    propiedad1: valor;  
    propiedad2: valor;  
}  
selector2 {  
    propiedad1: valor;  
}
```

2. Reglas CSS

✓ Reglas CSS:



- La declaración indica "*qué hay que hacer*".
 - El selector indica "*a quién hay que hacérselo*".
- ✓ A un mismo elemento HTML se le pueden aplicar varias reglas CSS y cada regla CSS puede aplicarse a un número ilimitado de elementos.

2. Reglas CSS

2.1. Reglas CSS: Comentarios

- ✓ Podemos añadir comentarios a un fichero CSS usando `/* */`

```
1  @charset "UTF-8";
2  /*Esto es una regla que afecta a un párrafo*/
3  p {
4      /* Esta propiedad da color a la letra del <p> */
5      color : white;
6      /* Esta propiedad da color al fondo del <p> */
7      background-color : green;
8      /* Esta propiedad define el tipo de letra al <p> */
9      font-family : arial;
10 }
```

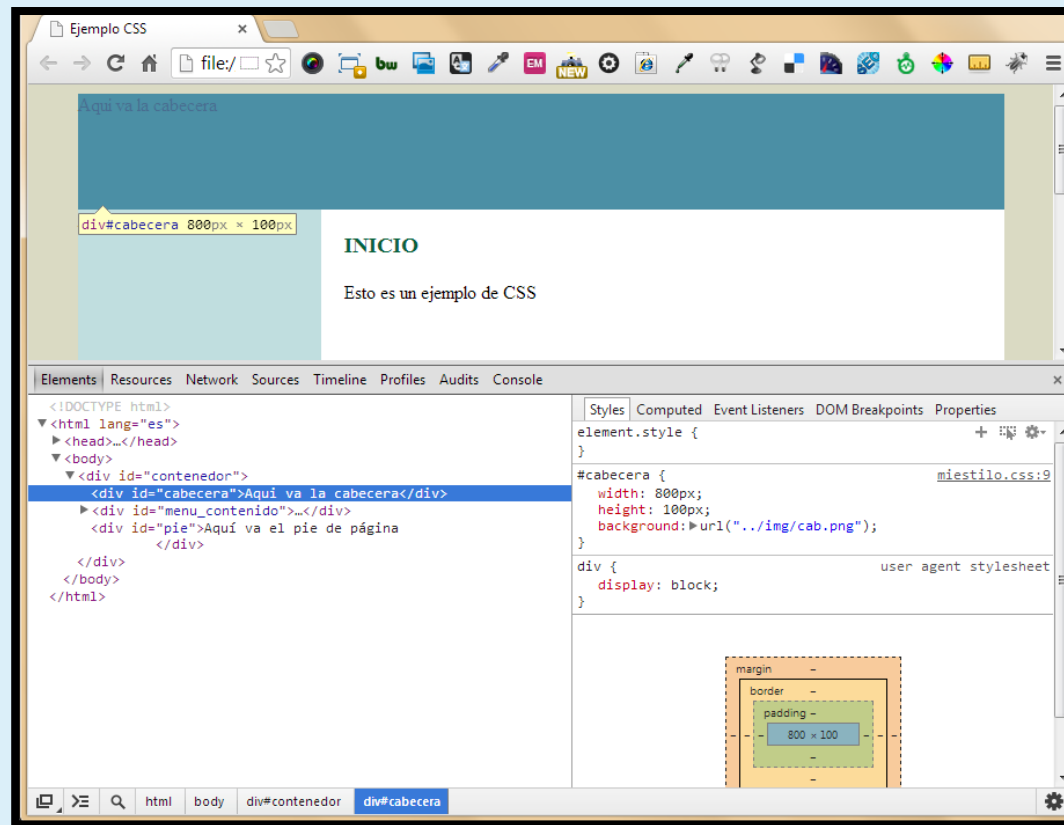
2. Reglas CSS

2.2. ¿Qué regla se está aplicando en cada momento?

- ✓ A un mismo elemento HTML se le pueden aplicar varias reglas CSS y cada regla CSS puede aplicarse a un número ilimitado de elementos. Las reglas CSS que se definen en un sitio web suele ser muy numerosas y en ocasiones pueden entrar en conflicto entre sí, estar duplicadas, aplicarse en cascada, etc.
- ✓ Para poder comprender bien qué regla está afectando a un determinado elemento HTML o por qué una regla no se aplica como tú quieres o simplemente para poder ajustar la posición de un elemento HTML tienes que familiarizarte con **la herramienta para desarrolladores que poseen ya todos los navegador web**.
- ✓ A partir de hoy esa herramienta junto con el editor de texto y el navegador en sí serán tus herramientas básicas de trabajo.

2. Reglas CSS

2.2. ¿Qué regla se está aplicando en cada momento?



ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
- 3. Incluir CSS en HTML**
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

3. Incluir CSS en HTML

- ✓ Existen varias formas de incluir CSS en HTML. Dependiendo de cómo se incluya habrá más acoplamiento entre nuestro estilo y nuestro contenido o menos. Las tres formas son:
 - Atributo style □ **Recuerda:** Atributo común a casi cualquier etiqueta HTML
 - Etiqueta <style>
 - Etiqueta <link>
- ✓ Ejemplo de un fichero sin ningún estilo:
 - Queremos que el color del texto de las cabeceras de nivel 3 sea azul:
 - **color:blue;**
 - Queremos que el tipo de letra de los párrafos sea Arial y el tamaño sea 1em:
 - **font-family:arial;**
 - **font-size:1em;**

3. Incluir CSS en HTML

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Ejemplo</title>
5   <meta charset="UTF-8">
6   <meta name="author" content="Alberto">
7 </head>
8 <body>
9   <h3>Producto 1:</h3>
10  <p>Descripción del producto 1.</p>
11  <h3>Producto 2:</h3>
12  <p>Descripción del producto 2.</p>
13  <h3>Producto 3:</h3>
14  <p>Descripción del producto 3.</p>
15
16 </body>
17 </html>
```

Producto 1:

Descripción del producto 1.

Producto 2:

Descripción del producto 2.

Producto 3:

Descripción del producto 3.

3. Incluir CSS en HTML

✓ Forma 1: Atributo style

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <title>Ejemplo</title>
5      <meta charset="UTF-8">
6      <meta name="author" content="Alberto">
7  </head>
8  <body>
9      <h3 style="color:blue;">Producto 1:</h3>
10     <p style="font-family:arial; font-size:1em">Descripción del producto 1.</p>
11
12     <h3 style="color:blue;">Producto 2:</h3>
13     <p style="font-family:arial; font-size:1em">Descripción del producto 2.</p>
14
15     <h3 style="color:blue;">Producto 3:</h3>
16     <p style="font-family:arial; font-size:1em">Descripción del producto 3.</p>
17
18 </body>
19 </html>
```

Producto 1:

Descripción del producto 1.

Producto 2:

Descripción del producto 2.

Producto 3:

Descripción del producto 3.

3. Incluir CSS en HTML

✓ Forma 2: Etiqueta <style>

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Ejemplo</title>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <style>
8     h3 {color:blue;}
9     p {font-family:arial;
10      font-size:1em;}
11  </style>
12 </head>
13 <body>
14   <h3>Producto 1:</h3>
15   <p>Descripción del producto 1.</p>
16
17   <h3>Producto 2:</h3>
18   <p>Descripción del producto 2.</p>
19
20   <h3>Producto 3:</h3>
21   <p>Descripción del producto 3.</p>
22 </body>
23 </html>
```

Producto 1:

Descripción del producto 1.

Producto 2:

Descripción del producto 2.

Producto 3:

Descripción del producto 3.

3. Incluir CSS en HTML

- ✓ **Forma 3: Etiqueta <link>.** Consiste en definir el estilo en un fichero aparte. Dicho fichero tendrá la extensión **".css"**

- Se debe especificar la codificación con **@charset="utf-8"**:

```
1 @charset "utf-8";
2 /* Mis estilos CSS */
3 h3 {color:blue;}
4 p {font-family:arial;
5     font-size:1em;}
6
```

- Desde el documento HTML se enlaza a dicho fichero usando la etiqueta <link> dentro del <head>:

```
<head>
  ...
  <link rel="stylesheet"
        type="text/css"
        href="nombreCarpetaCSS/nombreFicheroCSS.css">
</head>
```

3. Incluir CSS en HTML

✓ Forma 3: Etiqueta <link>.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Ejemplo</title>
5   <meta charset="UTF-8">
6   <meta name="author" content="Alberto">
7   <link rel="stylesheet" type="text/css" href="css/estilos.css">
8 </head>
9 <body>
10  <h3>Producto 1:</h3>
11  <p>Descripción del producto 1.</p>
12
13  <h3>Producto 2:</h3>
14  <p>Descripción del producto 2.</p>
15
16  <h3>Producto 3:</h3>
17  <p>Descripción del producto 3.</p>
18 </body>
19 </html>
```

```
1 @charset "utf-8";
2 /* Mis estilos CSS */
3 h3 {color:blue;}
4 p {font-family:arial;
5   font-size:1em;}
```

Producto 1:

Descripción del producto 1.

Producto 2:

Descripción del producto 2.

Producto 3:

Descripción del producto 3.

3. Incluir CSS en HTML

- ✓ ¿Qué ocurre si se incluyen estilos de las 3 formas posibles a la vez?

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Ejemplo</title>
5   <meta charset="UTF-8">
6   <meta name="author" content=" " >
7   <link rel="stylesheet" type="text/css" href="css/estilos.css">
8   <style>
9     p {font-family:arial;
10        font-size:0.8em;
11        color:red;}
12  </style>
13 </head>
14 <body>
15   <h3>Producto 1:</h3>
16   <p style="font-family:arial; font-size:2em;color:green;">
17     Descripción del producto 1.
18   </p>
19
20   <h3>Producto 2:</h3>
21   <p>Descripción del producto 2.</p>
22
23   <h3>Producto 3:</h3>
24   <p>Descripción del producto 3.</p>
25 </body>
26 </html>
```

```
1 @charset "utf-8";
2 /* Mis estilos CSS */
3 h3 {color:blue;}
4 p {font-family:arial;
5    font-size:1em;}
6
```

Producto 1:

Descripción del producto 1.

Producto 2:

Descripción del producto 2.

Producto 3:

Descripción del producto 3.

3. Incluir CSS en HTML

- ✓ ¿Qué ocurre si se incluyen estilos de las 3 formas posibles a la vez?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

1. Browser default
2. External style sheet
3. Internal style sheet (in the head section)
4. Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).



Note: If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
- 4. Selectores CSS**
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

4. Selectores CSS

- ✓ * □ Selector universal. Se utiliza para seleccionar todos los elementos de la página.
- ✓ #id □ Elemento con ese valor en el atributo id.
- ✓ .class □ Elemento con ese valor en el atributo clase.
- ✓ etiqueta □ Seleccionar esa etiqueta.
- ✓ selector1, selector2 □ Se cambian las propiedades a todos los elementos seleccionados por ambos.
- ✓ selector1 selector2 □ Se cambian las propiedades de los elementos seleccionados por selector2 que estén dentro de aquellos que se seleccionen con selector1.
- ✓ selector1>selector2 □ Se cambian las propiedades de los elementos seleccionados por selector2 que son hijos directos de aquellos que se seleccionen con selector1.
- ✓ selector1+selector2 □ De los elementos seleccionados por selector2 que están después de aquellos que se seleccionen por selector1.
- ✓ selector1~selector2 □ De los elementos seleccionados por selector2 que están justo antes de aquellos que se seleccionen por selector1.

4. Selectores CSS

- ✓ * Selector universal. Se utiliza para seleccionar todos los elementos de la página.
- ✓ El siguiente ejemplo establece el color de letra para todos los elementos de un sitio web.

```
* {  
    color: red;  
}
```

- ✓ Se debe tener mucho cuidado con este selector porque cambia el estilo de todos los elementos de un sitio web. Se suele emplear para "resetear" los estilos por defecto del navegador como márgenes, rellenos, fuentes, etc.
- ✓ El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML:

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

4. Selectores CSS

```
<body>
  <h1>Soy un h1</h1>
  <h3>Soy un h3</h3>
  <h6>Soy un h6</h6>
  <p>Soy un p</p>
  <pre>Soy un pre</pre>
  <a href="http://www.google.es" target="_blank">Soy un enlace</a>
  <ol>
    <li>Soy un elemento de una lista</li>
    <li>Soy un elemento de una lista</li>
  </ol>
</body>
```

```
* {
  color:red;
}
```



4. Selectores CSS

- ✓ Ejemplo: *Selector de etiqueta (o tipo o elemento)*

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <title>Ejemplo</title>
5     <meta charset="UTF-8">
6     <link rel="stylesheet" type="text/css" href="css/estilos.css">
7 </head>
8 <body>
9     <p>Párrafo 1</p>
10    <p>Párrafo 2</p>
11    <p class="azul">Párrafo 3</p>
12    <p class="azul">Párrafo 4</p>
13    <p class="verde">Párrafo 5</p>
14    <p class="verde">Párrafo 6</p>
15    <p id="p7">Párrafo 7</p>
16 </body>
17 </html>
```

```
1 @charset "utf-8";
2 /* Mis estilos CSS */
3 p {color:red;}
4
```

Parrafo 1
Parrafo 2
Parrafo 3
Parrafo 4
Parrafo 5
Parrafo 6
Parrafo 7

4. Selectores CSS

✓ Ejemplo: *Selector de clase (.nombreClase)*

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Ejemplo</title>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" type="text/css" href="css/estilos.css">
7 </head>
8 <body>
9   <p>Párrafo 1</p>
10  <p>Párrafo 2</p>
11  <p class="azul">Párrafo 3</p>
12  <p class="azul">Párrafo 4</p>
13  <p class="verde">Párrafo 5</p>
14  <p class="verde">Párrafo 6</p>
15  <p id="p7">Párrafo 7</p>
16 </body>
17 </html>
```

```
1 @charset "utf-8";
2 /* Mis estilos CSS */
3 p {color:red;}
4 .azul {color:blue;}
5 .verde {color:green;}
```

Párrafo 1
Párrafo 2
Párrafo 3
Párrafo 4
Párrafo 5
Párrafo 6
Párrafo 7

4. Selectores CSS

✓ Ejemplo: *Selector por identificador (#nombreidentificador)*

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>Ejemplo</title>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" type="text/css" href="css/estilos.css">
7 </head>
8 <body>
9   <p>Párrafo 1</p>
10  <p>Párrafo 2</p>
11  <p class="azul">Párrafo 3</p>
12  <p class="azul">Párrafo 4</p>
13  <p class="verde">Párrafo 5</p>
14  <p class="verde">Párrafo 6</p>
15  <p id="p7">Párrafo 7</p>
16 </body>
17 </html>
```

```
1 @charset "utf-8";
2 /* Mis estilos CSS */
3 p {color:red;}
4 .azul {color:blue;}
5 .verde {color:green;}
6 #p7 {color:orange;}
```

Párrafo 1
Párrafo 2
Párrafo 3
Párrafo 4
Párrafo 5
Párrafo 6
Párrafo 7

Recuerda: El identificador debe ser único en un fichero HTML.

4. Selectores CSS

- ✓ Ejemplo: *selector1*, *selector2*. Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores usando el operador coma (,). Es mejor agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos.

```
h1 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 2em;  
}  
h2 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 1em;  
}  
h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 0.5em;  
}
```

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}  
h1 {  
  font-size: 2em;  
}  
h2 {  
  font-size: 1em;  
}  
h3 {  
  font-size: 0.5em;  
}
```

4. Selectores CSS

- ✓ Ejemplo: *selector1 selector2* (Selector descendente). Siempre están formados por dos o más selectores separados entre sí por espacios en blanco. El último selector indica el elemento sobre el que se aplican los estilos y todos los selectores anteriores indican el lugar en el que se debe encontrar ese elemento. **La única condición es que un elemento debe estar dentro de otro elemento**, sin importar el nivel de profundidad en el que se encuentre.

```
<p>Esto es un párrafo con: <a href="http://www.as.com">Un enlace a as.com</a></p>  
<a href="http://www.marca.com">Un enlace a marca.com</a>
```

```
a {  
    color:red;  
}  
  
p a {  
    color:green;  
}
```

Esto es un párrafo con: Un enlace a as.com
Un enlace a marca.com

4. Selectores CSS

- ✓ Ejemplo: *selector1 selector2 (Selector descendente)*.
 - Utilizando el selector descendente es posible aplicar diferentes estilos a los elementos del mismo tipo.

```
p a {  
    color: orange;  
}  
h1 a {  
    color: blue;  
}
```

- **¡Cuidado!** No debe confundirse el selector descendente con la combinación de selectores:

```
/* El estilo se aplica a todos los elementos "p", "a", "span" y "em" */  
p, a, span, em { text-decoration: underline; }  
  
/* El estilo se aplica solo a los elementos "em" que se  
encuentran dentro de "p a span" */  
p a span em { text-decoration: underline; }
```


4. Selectores CSS

✓ Ejemplo: *Combinación de selectores*

```
<p class="negrita">Parrafo 1</p>  
<p class="rojo">Parrafo 2</p>  
<p class="negrita rojo">Parrafo 3</p>
```

```
.rojo {  
    color:red;  
}  
  
.negrita {  
    font-weight: bold;  
}
```

Parrafo 1

Parrafo 2

Parrafo 3

4. Selectores CSS

✓ Ejemplo: *Combinación de selectores*

```
<h1 class="rojo">Esto es un h1 en rojo</h1>  
<a class="rojo" href="http://www.goole.es">Esto es un a en rojo</a>  
<p class="rojo">Esto es un p en rojo (oscuro)</p>
```

```
.rojo {  
    color: red;  
}  
p.rojo {  
    color: darkred ;  
}
```

Esto es un h1 en rojo

Esto es un a en rojo

Esto es un p en rojo (oscuro)

4. Selectores CSS

✓ Recuerda:

```
/* Todos los elementos "p" con el atributo class="negrita" */  
p.negrita { ... }  
  
/* Todos los elementos de cualquier tipo con el atributo  
class="negrita" que estén dentro de un elemento "p" */  
p .negrita { ... }  
  
/* Todos los elementos de cualquier tipo con el atributo  
class="negrita" y todos los elementos "p" */  
p, .negrita { ... }
```

4. Selectores CSS

✓ Otros:

- **Selector de hijos:** Se utiliza para seleccionar un elemento que es hijo directo de otro elemento y se indica mediante el "signo de mayor que" (>).

https://www.w3schools.com/cssref/sel_element_gt.asp

- **Selector adyacente:** El selector adyacente se emplea para seleccionar elementos que en el código HTML de la página se encuentran justo a continuación de otros elementos. Su sintaxis emplea el signo + para separar los dos elementos.

https://www.w3schools.com/cssref/sel_element_pluss.asp

- **Selector de atributos:** Permiten seleccionar elementos HTML en función de sus atributos y/o valores de esos atributos.

https://www.w3schools.com/css/css_attribute_selectors.asp

- Referencia completa de todos los selectores CSS:

https://www.w3schools.com/cssref/css_selectors.asp

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
- 5. Propiedades y Valores**
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

5. Propiedades y Valores

- ✓ Las propiedades CSS han ido aumentando en los distintos niveles (o versiones) de CSS.
- ✓ Ejemplo:

```
body {  
    font: 100% Lucida Sans;  
    margin: 20px;  
    line-height: 26px;  
}  
div {  
    background-color: #32a4e7;  
    color: #ffffff;  
    padding: 10px;  
    width: 180px;  
    bottom: 0;  
    top: 0;  
    right: 0;  
    position: absolute;  
}
```

5. Propiedades y Valores

- ✓ Lista completa de propiedades:
 - <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
 - <https://www.w3schools.com/cssref/default.asp>
- ✓ Lista de propiedades esenciales:
 - https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Properties_Reference

5. Propiedades y Valores

- ✓ Los valores que pueden tomar una regla depende de la regla en sí y para saber qué valores puede tomar una determinada regla debemos consultar la documentación oficial.
 - Valores de posición: **left, right, top, bottom, etc.**
 - Valores reservados: **inline, block, flex, table, etc.**
 - Colores: **#F4A243, rgb(34,255,135), green, etc.**
 - Medidas: **45px, 2.5cm, 0.3em, etc.**

```
border: 1px solid #d4d4d4;  
background-color: #f1f1f1;  
text-align: center;  
padding: 10px;  
font-size: 70%;  
line-height: 14px;
```


5. Propiedades y Valores

✓ Colores:

- Métodos principales:

```
/* Color Names */  
p {background-color:red;} /*red*/  
  
/* Hexadecimal Colors */  
p {background-color:#ff0000;} /*red*/  
  
/* RGB Colors */  
p {background-color:rgb(255,0,0);} /*red*/  
  
/* RGBA Colors */  
p {background-color:rgba(255,0,0,0.3);} /*red with opacity*/
```

- https://www.w3schools.com/colors/colors_picker.asp

5. Propiedades y Valores

✓ Fondos:

- Propiedades para definir efectos de fondo:

Property	Description
<u>background</u>	Sets all the background properties in one declaration
<u>background-attachment</u>	Sets whether a background image is fixed or scrolls with the rest of the page
<u>background-color</u>	Sets the background color of an element
<u>background-image</u>	Sets the background image for an element
<u>background-position</u>	Sets the starting position of a background image
<u>background-repeat</u>	Sets how a background image will be repeated

- https://www.w3schools.com/css/css_background.asp

5. Propiedades y Valores

✓ Fondos:

- **Background - Shorthand property:** Nos permite definir en **una sola línea** todas las propiedades relacionadas con background. Se debe respetar el siguiente orden:
 - background-color
 - background-image
 - background-repeat
 - background-attachment
 - background-position

```
body {  
    background:#ffffff url('img_tree.png') no-repeat right top;  
}
```

5. Propiedades y Valores

✓ Dimensiones y unidades:

- Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos o para establecer el tamaño de letra del texto.
- Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco).
- CSS divide las unidades de medida en dos grupos: absolutas y relativas.
 - Las **medidas relativas** definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado.
 - Las **unidades absolutas** establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.
- https://www.w3schools.com/cssref/css_units.asp

5. Propiedades y Valores

✓ Dimensiones y unidades:

■ Unidades absolutas:

- **in:** pulgadas ("inches", en inglés). Una pulgada equivale a 2.54 centímetros.
- **cm:** centímetros.
- **mm:** milímetros.
- **pt:** puntos. Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros.
- **pc:** picas. Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros.

```
2  h1 {font-size: 1.5cm;
3    }
4  h2 {font-size: 0.5in;
5    }
6  p  {font-size: .5cm;
7      line-height: 10mm;
8    }
```

5. Propiedades y Valores

✓ Dimensiones y unidades:

- **Unidades relativas:** Las unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios.
 - **px:** (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.
 - **%:** porcentual: porcentaje de lo que ocupe el padre dentro del árbol.
 - **em:** en relación al tamaño por defecto de la letra del navegador (normalmente 16px).
 - **rem:** en relación al tamaño por defecto de la letra que tiene la etiqueta HTML.
 - **ex:** relativa respecto de la altura de la letra x ("*equis minúscula*") del tipo y tamaño de letra del elemento.
 - **ch:** relativa a la anchura "zero".

5. Propiedades y Valores

✓ Texto:

- Propiedades para definir efectos de texto:
https://www.w3schools.com/css/css_text.asp

Property	Description
<u>color</u>	Sets the color of text
<u>direction</u>	Specifies the text direction/writing direction
<u>letter-spacing</u>	Increases or decreases the space between characters in a text
<u>line-height</u>	Sets the line height
<u>text-align</u>	Specifies the horizontal alignment of text
<u>text-decoration</u>	Specifies the decoration added to text
<u>text-indent</u>	Specifies the indentation of the first line in a text-block
<u>text-shadow</u>	Specifies the shadow effect added to text
<u>text-transform</u>	Controls the capitalization of text
<u>unicode-bidi</u>	Used together with the <u>direction</u> property to set or return whether the text should be overridden to support multiple languages in the same document
<u>vertical-align</u>	Sets the vertical alignment of an element
<u>white-space</u>	Specifies how white-space inside an element is handled
<u>word-spacing</u>	Increases or decreases the space between words in a text

5. Propiedades y Valores

✓ Texto:

```
/* Text Color */
body { color:blue; }
h1 { color:#00ff00; }
h2 { color:rgb(255,0,0); }

/* Text Alignment */
h1 { text-align:center; }
p { text-align:justify; }
.derecha { text-align:right; }

/* Text Decoration */
h1 { text-decoration:overline; }
h2 { text-decoration:line-through; }
h3 { text-decoration:underline; }

/* Text Transform */
.mayuscula {text-transform:uppercase;}
.minuscula {text-transform:lowercase;}
.capital {text-transform:capitalize;}

/* Text Indent */
p {text-indent:50px;}
```


5. Propiedades y Valores

✓ Fuente:

- Propiedades para definir efectos de tipografía:

Property	Description
<u>font</u>	Sets all the font properties in one declaration
<u>font-family</u>	Specifies the font family for text
<u>font-size</u>	Specifies the font size of text
<u>font-style</u>	Specifies the font style for text
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font
<u>font-weight</u>	Specifies the weight of a font

- https://www.w3schools.com/css/css_font.asp

5. Propiedades y Valores

✓ Fuente:

```
/* Font Style */
p.normal {font-style:normal;}
p.italic {font-style:italic;}
p.oblique {font-style:oblique;}

/* Font Size */
h1 {font-size:40px;}
h2 {font-size:30px;}
h3 {font-size:2.5em;}
h4 {font-size:1.875em;}
p {font-size:0.875em;}

/* Font weight */
p.normal {font-weight:normal;}
p.light {font-weight:lighter;}
p.thick {font-weight:bold;}
p.thicker {font-weight:900;}

/* Font variant */
p.normal {font-variant:normal;}
p.small {font-variant:small-caps;}
```

5. Propiedades y Valores

✓ Fuente:

- **Font - Shorthand property:** Nos permite definir en **una sola línea** todas las propiedades relacionadas con font. Se debe respetar el siguiente orden:
 - font-style
 - font-variant
 - font-weight
 - font-size/line-height
 - font-family

5. Propiedades y Valores

✓ Listas:

- Propiedades para definir efectos en las listas:

Property	Description
<u>list-style</u>	Sets all the properties for a list in one declaration
<u>list-style-image</u>	Specifies an image as the list-item marker
<u>list-style-position</u>	Specifies if the list-item markers should appear inside or outside the content flow
<u>list-style-type</u>	Specifies the type of list-item marker

- https://www.w3schools.com/css/css_list.asp

5. Propiedades y Valores

✓ Listas:

■ list-style-type:

```
<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
```

```
ul.a {list-style-type:circle;}
ul.b {list-style-type:square;}
ol.c {list-style-type:upper-roman;}
ol.d {list-style-type:lower-alpha;}
```

Example of unordered lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola

- a. Coffee
- b. Tea
- c. Coca Cola

5. Propiedades y Valores

✓ Listas:

- list-style-image:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```
ul {
  list-style-image:url('sqpurple.gif');
}
```

- Coffee
- Tea
- Coca Cola

5. Propiedades y Valores

✓ Listas:

- list-style-position

Value	Description
inside	Indents the marker and the text. The bullets appear inside the content flow
outside	Keeps the marker to the left of the text. The bullets appears outside the content flow. This is default
inherit	Specifies that the value of the list-style-position property should be inherited from the parent element

```
ul {  
    list-style-position: inside;  
}
```

Outside:

- Coffee
- Tea
- Coca-cola

Inside:

- Coffee
- Tea
- Coca-cola

5. Propiedades y Valores

✓ Listas:

- **List-style - Shorthand property:** Nos permite definir en **una sola línea** todas las propiedades relacionadas con list. Se debe respetar el siguiente orden:
 - list-style-type
 - list-style-position
 - list-style-image

```
p{  
    list-style: disc outside url(image.gif);  
}
```


5. Propiedades y Valores

✓ Tablas:

✓ https://www.w3schools.com/css/css_table.asp

```
/* Table Color */
table, td, th { border:1px solid green; }
th { background-color:green;
    color:white;
}

/* Collapse Borders (Borde exterior) */
table { border-collapse:collapse; }

/* Table Width and Height */
table { width:100%; }
th { height:50px; }

/* Table Text Alignment */
td { text-align:right; }
td { vertical-align:bottom; }

/* Table Padding */
td { padding:15px; }

/* Caption Side */
caption {caption-side:bottom;}
```

5. Propiedades y Valores

✓ Enlaces:

- La etiqueta "a" permite definir lo que se conoce como selectores de estado (o pseudo-clases).
 - a:link □ Estilo de un enlace sin visitar.
 - a:visited □ Estilo de un enlace visitado.
 - a:hover □ Estilo de un enlace al poner el ratón sobre él.
 - a:active □ Estilo de un enlace al pulsarlo con el ratón.
- a:hover siempre debe aparecer después de a:link y a:visited
- a:active siempre debe aparecer después de a:hover

- https://www.w3schools.com/css/css_link.asp

```
a:link {color:#FF0000;  
        text-decoration:none;  
        background-color:#B2FF99;  
    }  
a:visited {color:#00FF00;  
           text-decoration:none;  
           background-color:#FFFF85;  
    }  
a:hover {color:#FF00FF;  
         text-decoration:underline;  
         background-color:#FF704D;  
    }  
a:active {color:#0000FF;  
         text-decoration:underline;  
         background-color:#FF704D;  
    }
```

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
- 6. Pseudo-clases y Pseudo-elementos**
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

6. Pseudo-clases y Pseudo-elementos

- ✓ Una **pseudoclase CSS** es una palabra clave que se añade a los selectores y que especifica un ESTADO especial del elemento seleccionado.
- ✓ **Sintaxis:**

```
selector:pseudo-clase {  
    propiedad1: valor;  
    propiedad2: valor;  
    ...  
    propiedadn: valor;  
}
```

6. Pseudo-clases y Pseudo-elementos

✓ Pseudo-clases de ESTADO:

- :link
- :visited
- :enabled
- :disabled
- :checked
- :required
- :optional
- :focus
- :hover
- :empty
- ...

✓ Pseudo-clases de POSICIÓN:

- :first
- :first-child
- :first-of-type
- :last
- :last-child
- :last-of-type
- :nth-child(n)
- :nth-last-child()
- :nth-last-of-type()
- ...

6. Pseudo-clases y Pseudo-elementos

✓ Ejemplos de Pseudo-clases:

- **:first-child** □ Selecciona el primer elemento hijo de un elemento.

```
p:first-child {  
    color: red;  
}
```

- **:hover** □ Se activa cuando el usuario pasa el ratón o cualquier otro elemento apuntador por encima de un elemento.

```
div:hover {  
    background-color: #F89B4D;  
}
```

- **:focus** □ Se activa cuando el elemento tiene el foco del navegador, es decir, cuando el elemento está seleccionado.

6. Pseudo-clases y Pseudo-elementos

- ✓ Al igual que las pseudo-clases, los **pseudo-elementos** se añaden a los selectores, pero en cambio, no describen un estado especial sino que, permiten añadir estilos a una parte concreta del documento. Por ejemplo, el pseudoelemento **::first-line** selecciona solo la primera línea del elemento especificado por el selector.

- ✓ **Sintaxis:**

```
selector::pseudo-elemento {  
    propiedad1 : valor ;  
    propiedad2 : valor ;  
    ...  
    propiedadn : valor ;  
}
```

6. Pseudo-clases y Pseudo-elementos

- ✓ Algunos Pseudo-elementos:
 - ::first-line
 - ::first-letter
 - ::after
 - ::before
 - ::selected
 - ...

6. Pseudo-clases y Pseudo-elementos

✓ Ejemplos de Pseudo-elementos:

- **::first-line** □ Permite seleccionar la primera línea de texto de un elemento.

```
p::first-line { text-transform: uppercase; }
```

- **::first-letter** □ Permite seleccionar la primera letra de la primera línea de texto de un elemento.

```
p::first-letter { text-transform: uppercase; }
```

ÍNDICE DE CONTENIDOS

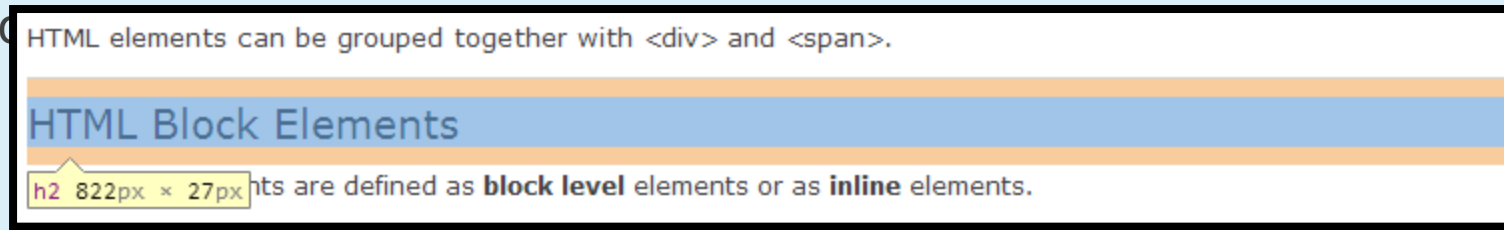
1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
- 7. Modelo de Caja**
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

7. Modelo de Caja

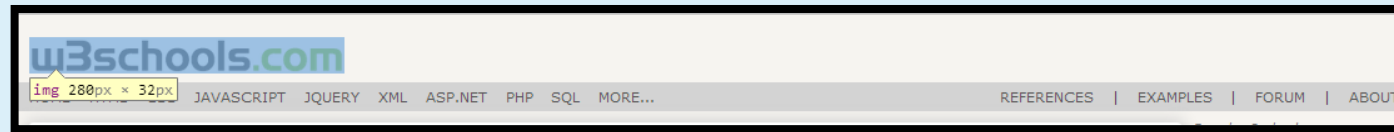
- ✓ En HTML se distingue entre etiquetas de nivel de bloque y etiquetas en línea.
 - Nivel de bloque: `<h1>`, `<p>`, ``, `<table>`, ...
 - En línea: ``, `<td>`, `<a>`, ``, ...

✓ ¿Qué significa esto?

- Ejemplo



- Ejemplo del comportamiento de una etiqueta `` (**en línea**):



7. Modelo de Caja

- ✓ Los **elementos en línea** definidos por HTML son:
 - *a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.*
- ✓ Los **elementos de bloque** definidos por HTML son:
 - *address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframe, noscript, ol, p, pre, table, ul.*
 - Los siguientes elementos también se considera que **son de bloque**: *dd, dt, frameset, li, tbody, td, tfoot, th, thead, tr.*
- ✓ **El comportamiento de un elemento (en línea o de bloque) se puede modificar en CSS.** Así que debes tener cuidado.

7. Modelo de Caja

✓ ¿Y para qué sirven las etiquetas `<div>` y ``?

- La etiqueta `<div>`:

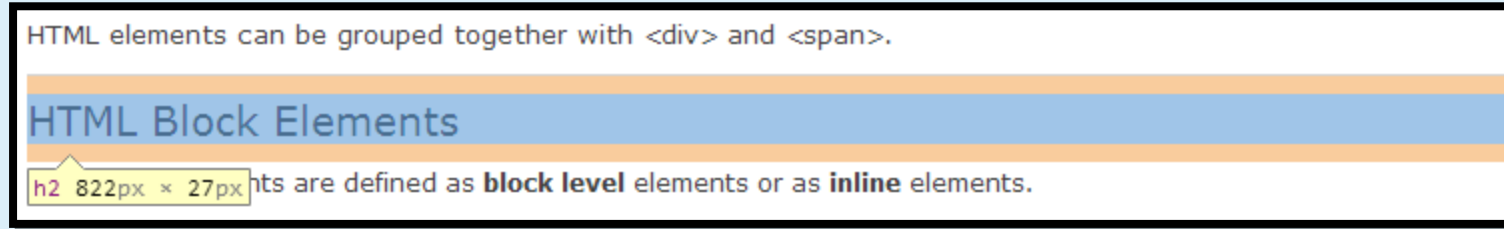
Es una etiqueta **de nivel de bloque** que se usa para contener otras etiquetas y organizar el contenido en una página html. Su uso está ligado a CSS.

- La etiqueta ``:

Es una etiqueta **de nivel en línea** que se usar para agrupar otras etiquetas, principalmente relacionadas con el texto. Su uso está ligado a CSS.

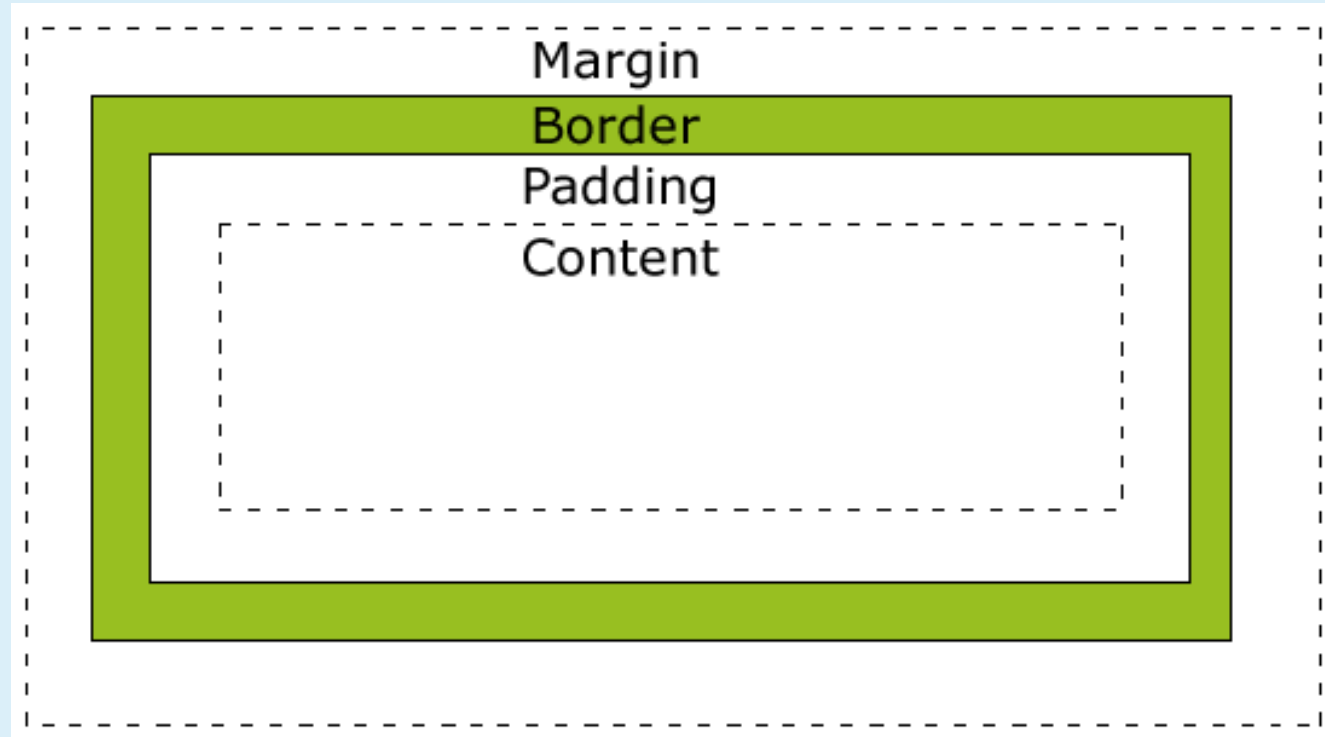
7. Modelo de Caja

- ✓ Nunca debes olvidar que **todos los elementos HTML** de una página web se tratan como si fueran "**cajas**". Algunas cajas serán en línea y otras en bloque:



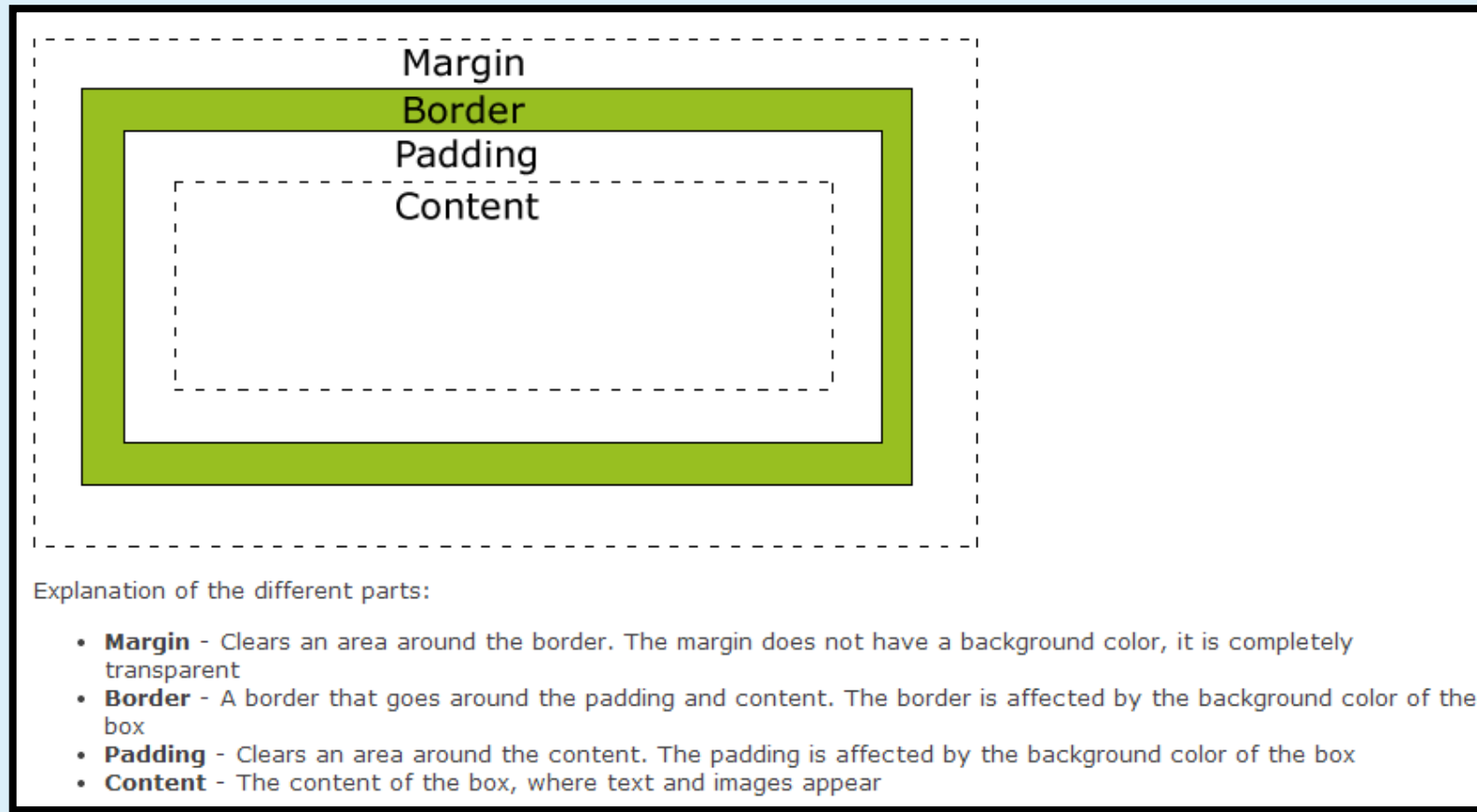
- ✓ Todas las cajas siguen lo que se conoce como **modelo de caja**.
- ✓ ¿Y en qué consiste dicho modelo?

7. Modelo de Caja



✓ https://www.w3schools.com/css/css_boxmodel.asp

7. Modelo de Caja



7. Modelo de Caja

The screenshot shows a web browser window displaying a CSS tutorial page. The page layout includes a sidebar with links to learn HTML, HTML5, CSS, CSS3, and JavaScript. The main content area features a green 'w3schools' logo and the word 'HTML', followed by a blue 'w3schools' logo and the word 'CSS'. Below these are buttons for 'HTML Tutorial' and 'CSS Tutorial'. The browser's developer tools are open, showing the 'Elements' panel with the following HTML structure:

```
<div id="maincol">
  <div class="box1">...</div>
  <div class="box1">
    <a class="box" href="css/default.asp" target="_top">
      <div class="image" style="background-color: #07B3D3;"></div>
      <h1>CSS</h1>
    </a>
    <div style="clear:both;"></div>
    <a class="btn" href="css/default.asp" target="_top">CSS Tutorial</a>
    <a class="btn" href="cssref/default.asp" target=
```

The 'Styles' panel on the right shows the box model for the selected element. The diagram illustrates the following dimensions:

- margin: 10
- border: 0.909
- padding: 8
- Content area: 170 x 20

7. Modelo de Caja

7.1. Border

- ✓ Sobre un borde ("border") podemos definir propiedades relacionadas con:
 - El estilo del borde: *border-style* https://www.w3schools.com/cssref/pr_border-style.asp
 - El grosor del borde: *border-width* https://www.w3schools.com/cssref/pr_border-width.asp
 - El color del borde: *border-color* https://www.w3schools.com/cssref/pr_border-color.asp
- ✓ **Border - Shorthand property:** Nos permite definir en una sola línea todas las propiedades relacionadas con border. Se debe respetar el siguiente orden:
 - border-width
 - border-style (required)
 - border-color

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

7. Modelo de Caja

7.1. Border

- ✓ **Propiedades individuales:** Podemos definir propiedades individuales: top, right, bottom y left para cada una de las propiedades anteriores.

```
p{  
    border-top-style:dotted;  
    border-right-style:solid;  
    border-bottom-style:dotted;  
    border-left-style:solid;  
}
```

7. Modelo de Caja

7.2. Outline (Borde exterior)

- ✓ Se considera una variante de la propiedad "border" y por eso no se presentó esta propiedad en el modelo de caja.
- ✓ Sobre un borde exterior (outline) podemos definir propiedades relacionadas con:
 - El estilo del borde: *outline-style* https://www.w3schools.com/cssref/pr_outline-style.asp
 - El grosor del borde: *outline-width* https://www.w3schools.com/cssref/pr_outline-width.asp
 - El color del borde: *outline-color* https://www.w3schools.com/cssref/pr_outline-color.asp
- ✓ Asimismo, podemos definir propiedades individuales: *top*, *right*, *bottom* y *left* para cada una de las propiedades anteriores.

7. Modelo de Caja

7.2. Outline (Borde exterior)

- ✓ **Outline - Shorthand property:** Nos permite definir en una sola línea todas las propiedades relacionadas con border exterior. Se debe respetar el siguiente orden:
- outline-color
 - outline-style (required)
 - outline-width

```
p {  
    outline: #00FF00 dotted thick;  
}
```

<u>outline</u>	Sets all the outline properties in one declaration	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i> inherit
----------------	--	---

7. Modelo de Caja

7.3. Margin

- ✓ **"margin"** indica la separación entre el borde ("border") de un elemento y el resto de elementos. Tiene la característica de que **esa separación es transparente**.
- ✓ Sobre un margen exterior ("margin") sólo se puede definir la **distancia de separación**. Dicha distancia se suele indicar en **px**.
- ✓ Igual que antes, podemos definir propiedades individuales: *top*, *right*, *bottom* y *left*
- ✓ https://www.w3schools.com/cssref/pr_margin.asp

7. Modelo de Caja

7.3. Margin

```
7  .caja1 {  
8      margin: 10px; /* Los cuatro márgenes son 10px */  
9  }  
10  
11  .caja2 {  
12      margin: 10px 5px; /* top - bottom ---- right - left */  
13  }  
14  
15  .caja3 {  
16      margin: 10px 15px 20px; /* top ---- right - left ---- bottom */  
17  }  
18  
19  .caja4 {  
20      margin: 10px 25px 15px 20px; /*top ---- right ---- bottom ---- left */  
21  }  
22
```

7. Modelo de Caja

7.4. Padding

- ✓ **"padding"** indica la separación (o relleno) entre el borde ("border") de un elemento y el contenido del propio elemento. Tiene la característica de que esa separación **NO es transparente**. Toma el color de fondo del contenido.
- ✓ Sobre un relleno interior ("padding") solo se puede definir **la distancia** de ese relleno. Dicha distancia se suele indicar en **px**.
- ✓ Igual que antes, podemos definir propiedades individuales: *top*, *right*, *bottom* y *left*.

7. Modelo de Caja

7.4. Padding

```
10  □ .caja1 {  
11      padding: 10px; /* Los cuatro márgenes son 10px */  
12  }  
13  
14  □ .caja2 {  
15      padding: 10px 5px; /* top - bottom ---- right - left */  
16  }  
17  
18  □ .caja3 {  
19      padding: 10px 15px 20px; /* top ---- right - left ---- bottom */  
20  }  
21  
22  □ .caja4 {  
23      padding: 10px 25px 15px 20px; /*top ---- right ---- bottom ---- left */  
24  }  
25
```

7. Modelo de Caja

7.4. Padding

- ✓ Resetear márgenes y rellenos que aplica el navegador por defecto:

```
1  @charset "utf-8";
2
3  * {
4      margin: 0px;
5      padding: 0px;
6  }
7
```

7. Modelo de Caja

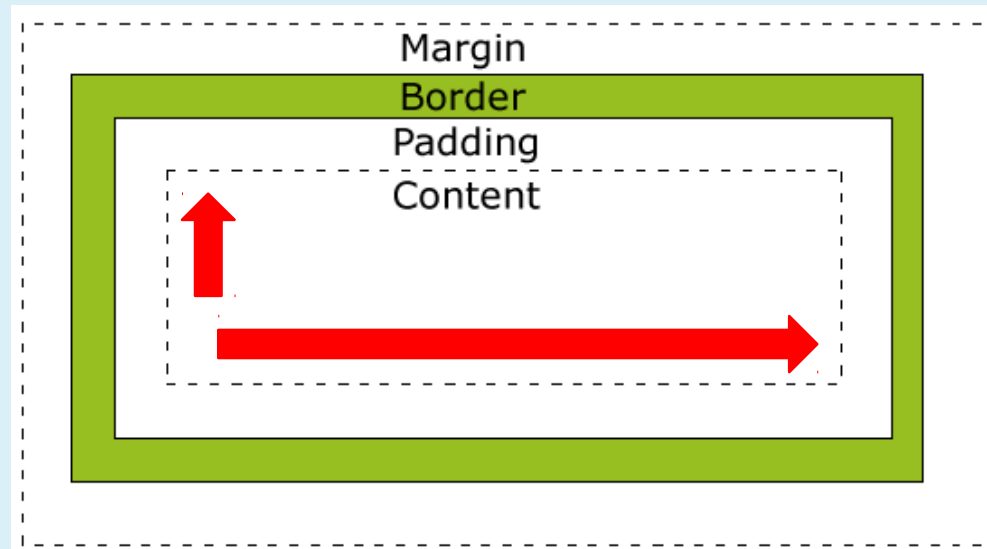
7.5. CSS Width and Height

- ✓ Cuando se diseña la estructura de una página web, se debe tomar la decisión de optar por un diseño de anchura fija o un diseño cuya anchura se adapta a la anchura de la ventana del navegador.
- ✓ Sin embargo, la mayoría de las veces sería conveniente una solución intermedia: que la anchura de la página sea variable y se adapte a la anchura de la ventana del navegador, pero respetando ciertos límites, es decir, que la anchura de la página no sea tan pequeña como para que no se puedan mostrar correctamente los contenidos y tampoco sea tan ancha como para que las líneas de texto no puedan leerse cómodamente.

7. Modelo de Caja

7.5. CSS Width and Height

- ✓ CSS proporciona una serie de reglas que permiten definir las dimensiones del **contenido** ("content") una caja.
- ✓ Estas dimensiones suelen darse en **px** o con valores porcentuales %.



7. Modelo de Caja

7.5. CSS Width and Height

- ✓ Existen reglas que permiten definir una anchura/altura fija (width y height) o una anchura/altura que se adapta al tamaño del navegador (max-width, min-width, max-height y min-height).

Property	Description	Values
<u>height</u>	Sets the height of an element	auto length % inherit
<u>width</u>	Sets the width of an element	auto length % inherit

Property	Description	Values
<u>max-height</u>	Sets the maximum height of an element	none length % inherit
<u>max-width</u>	Sets the maximum width of an element	none length % inherit
<u>min-height</u>	Sets the minimum height of an element	length % inherit
<u>min-width</u>	Sets the minimum width of an element	length % inherit

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
- 8. Posicionamiento en CSS**
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

8. Posicionamiento

✓CSS define 5 formas diferentes para posicionar una caja:

- **Posicionamiento normal o estático:** se trata del posicionamiento que utilizan los navegadores por defecto. Posiciona al elemento según el lugar donde fue definido en el HTML.
- **Posicionamiento relativo:** variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto a esa posición.
- **Posicionamiento absoluto:** la posición de una caja se establece de forma absoluta respecto de su elemento contenedor. El resto de elementos de la página ignoran la nueva posición del elemento.
- **Posicionamiento fijo:** su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- **Posicionamiento flotante:** se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

8. Posicionamiento

- ✓ Principales propiedades para posicionar:

<u>position</u>	Specifies the type of positioning for an element	absolute fixed relative static inherit
-----------------	--	--

```
h2 {  
    position: absolute;  
    ...  
}
```

- Además de indicar la posición que queremos, debemos **indicar a qué distancia queremos posicionar el elemento**. Esto se indicará generalmente en **px** y se hará indicando la distancia que queremos a la izquierda, a la derecha, arriba y abajo.

8. Posicionamiento

- ✓ Principales propiedades para posicionar:

<u>bottom</u>	Sets the bottom margin edge for a positioned box	auto <i>length</i> % inherit
<u>left</u>	Sets the left margin edge for a positioned box	auto <i>length</i> % inherit
<u>right</u>	Sets the right margin edge for a positioned box	auto <i>length</i> % inherit
<u>top</u>	Sets the top margin edge for a positioned box	auto <i>length</i> % inherit

```
h2 {  
    position: absolute;  
    left: 100px;  
    top: 150px;  
}
```

8. Posicionamiento

8.1. Posicionamiento normal o estático

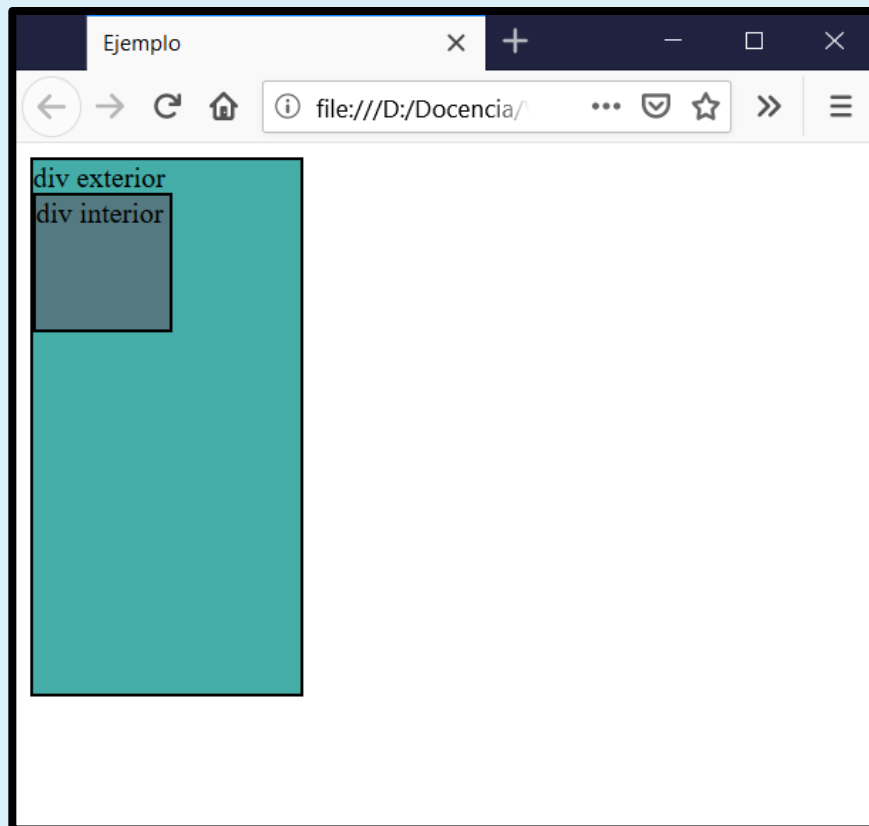
- ✓ Se trata del posicionamiento que utilizan los navegadores por defecto. Posiciona al elemento según el lugar donde fue definido en el HTML.

```
10 <body>
11   <div id="exterior">
12     div exterior
13     <div id="interior">
14       div interior
15     </div>
16   </div>
17 </body>
```

```
1  @charset "utf-8";
2  div {
3    border: 2px solid black;
4  }
5
6  #interior {
7    height: 75px;
8    width: 75px;
9    background-color: #547980;
10   position: static;
11 }
12
13 #exterior{
14   height: 300px;
15   width: 150px;
16   background-color: #45ADA8;
17   position: static;
18 }
```

8. Posicionamiento

8.1. Posicionamiento normal o estático



```
1 @charset "utf-8";
2 div {
3     border: 2px solid black;
4 }
5
6 #interior {
7     height: 75px;
8     width: 75px;
9     background-color: #547980;
10    position: static;
11 }
12
13 #exterior {
14     height: 300px;
15     width: 150px;
16     background-color: #45ADA8;
17     position: static;
18 }
```

8. Posicionamiento

8.2. Posicionamiento absoluto

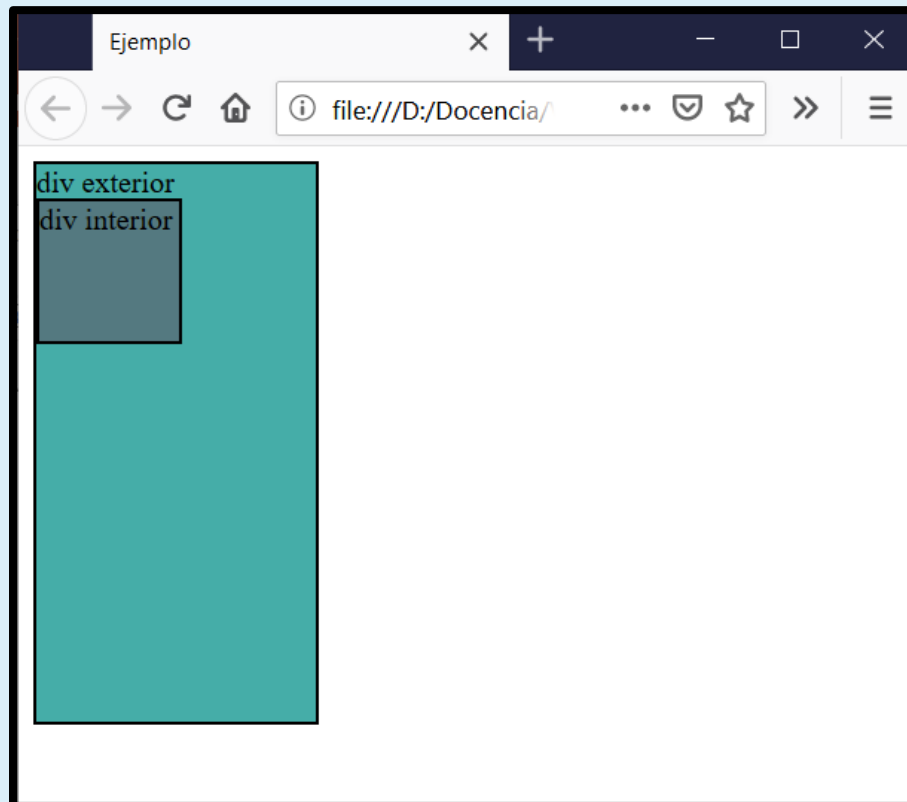
- ✓ La posición de una caja se establece de forma absoluta y el elemento es colocado exactamente donde nosotros queremos

```
10 <body>
11   <div id="exterior">
12     div exterior
13   <div id="interior">
14     div interior
15   </div>
16 </div>
17 </body>
```

```
1  @charset "utf-8";
2  div {
3    border: 2px solid black;
4  }
5
6  #interior {
7    height: 75px;
8    width: 75px;
9    background-color: #547980;
10   position: static;
11 }
12
13 #exterior {
14   height: 300px;
15   width: 150px;
16   background-color: #45ADA8;
17   position: absolute;
18   left: 100px;
19   top: 150px;
20 }
```

8. Posicionamiento

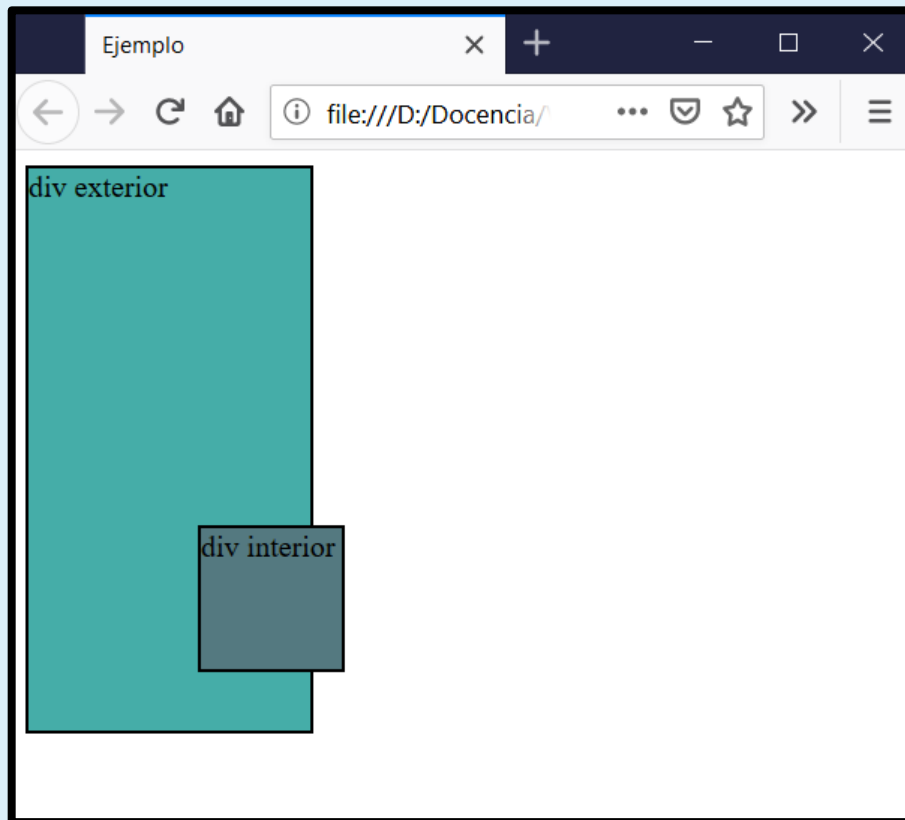
8.2. Posicionamiento absoluto



```
1  @charset "utf-8";
2  div {
3      border: 2px solid black;
4  }
5
6  #interior {
7      height: 75px;
8      width: 75px;
9      background-color: #547980;
10     position: static;
11 }
12
13 #exterior {
14     height: 300px;
15     width: 150px;
16     background-color: #45ADA8;
17     position: absolute;
18     left: 100px;
19     top: 150px;
20 }
```

8. Posicionamiento

8.2. Posicionamiento absoluto



```
1 @charset "utf-8";
2 div {
3     border: 2px solid black;
4 }
5
6 #interior {
7     height: 75px;
8     width: 75px;
9     background-color: #547980;
10    position: absolute;
11    left: 100px;
12    top: 200px;
13 }
14
15 #exterior{
16     height: 300px;
17     width: 150px;
18     background-color: #45ADA8;
19     position: absolute;
20     left: 100px;
21     top: 150px;
22 }
```

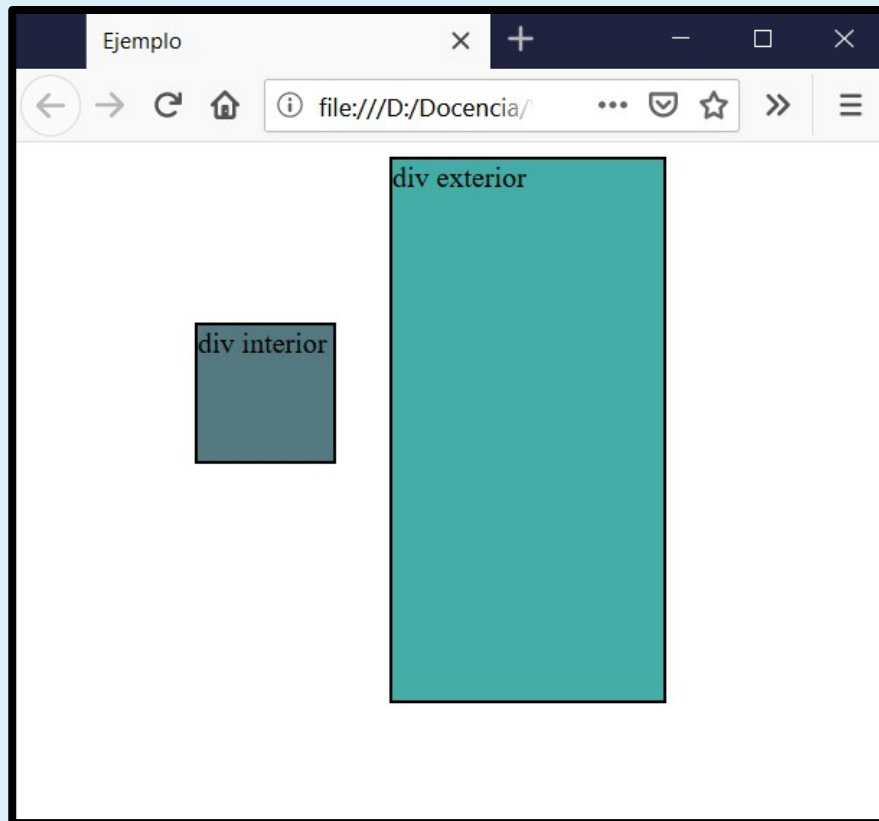
8. Posicionamiento

8.2. Posicionamiento absoluto

- ✓ La posición de una caja se establece de forma absoluta y el elemento es colocado exactamente donde nosotros queremos.
- ✓ **MUY IMPORTANTE:** Cuando se establece un elemento con `position: absolute`, se posiciona en relación con el primer elemento padre que tenga y que **no esté establecido con "position:static"**. Si no existe tal elemento, el elemento con `"position: absolute"` se posiciona en relación con `<html>`.
- ✓ En el ejemplo anterior el `<div>` con `id=interior` se posiciona con respecto al `<div>` con `id=exterior` porque dicho `<div>` tiene una posición no estática. Asimismo, el `<div>` con `id=exterior` se posiciona con respecto al `<html>` porque no tiene ningún padre con posición no estática.

8. Posicionamiento

8.2. Posicionamiento absoluto



```
1  @charset "utf-8";
2  div {
3      border: 2px solid black;
4  }
5
6  #interior {
7      height: 75px;
8      width: 75px;
9      background-color: #547980;
10     position: absolute;
11     left: 100px;
12     top: 200px;
13 }
14
15 #exterior{
16     height: 300px;
17     width: 150px;
18     background-color: #45ADA8;
19     position: static;
20     /* ¡Usa margin-left! No confundir con left */
21     margin-left: 300px;
22 }
```


8. Posicionamiento

8.3. Posicionamiento relativo

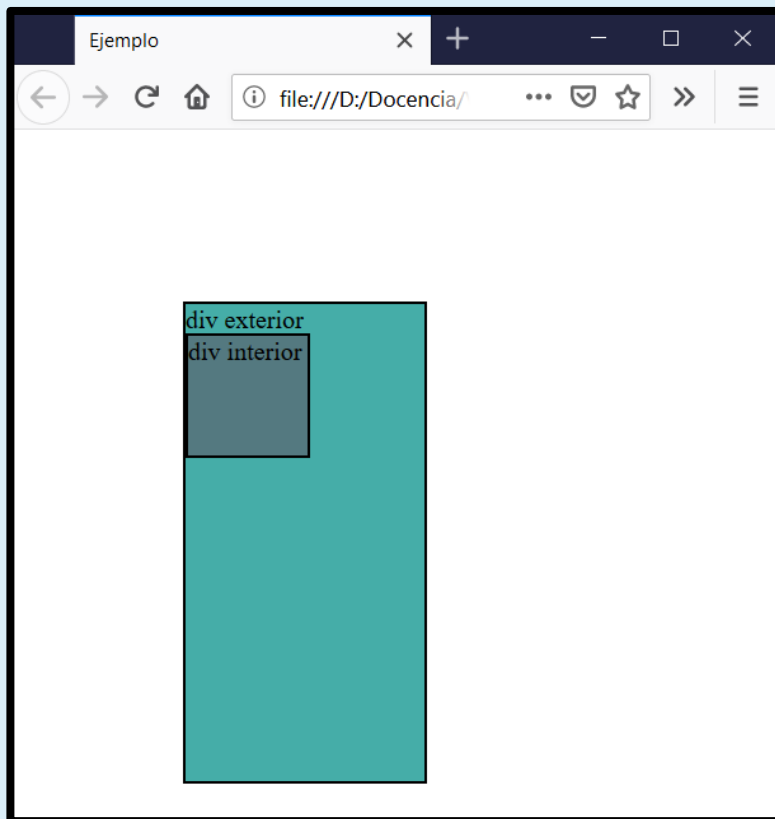
- ✓ Le indica al elemento que se desplace respecto del lugar donde hubiera estado posicionado de forma predeterminada como "static".

```
10 <body>
11   <div id="exterior">
12     div exterior
13     <div id="interior">
14       div interior
15     </div>
16   </div>
17 </body>
```

```
1  @charset "utf-8";
2  div {
3    border: 2px solid black;
4  }
5
6  #interior {
7    height: 75px;
8    width: 75px;
9    background-color: #547980;
10 }
11
12 #exterior {
13   height: 300px;
14   width: 150px;
15   background-color: #45ADA8;
16   position: relative;
17   left: 100px;
18   top: 100px;
19 }
```

8. Posicionamiento

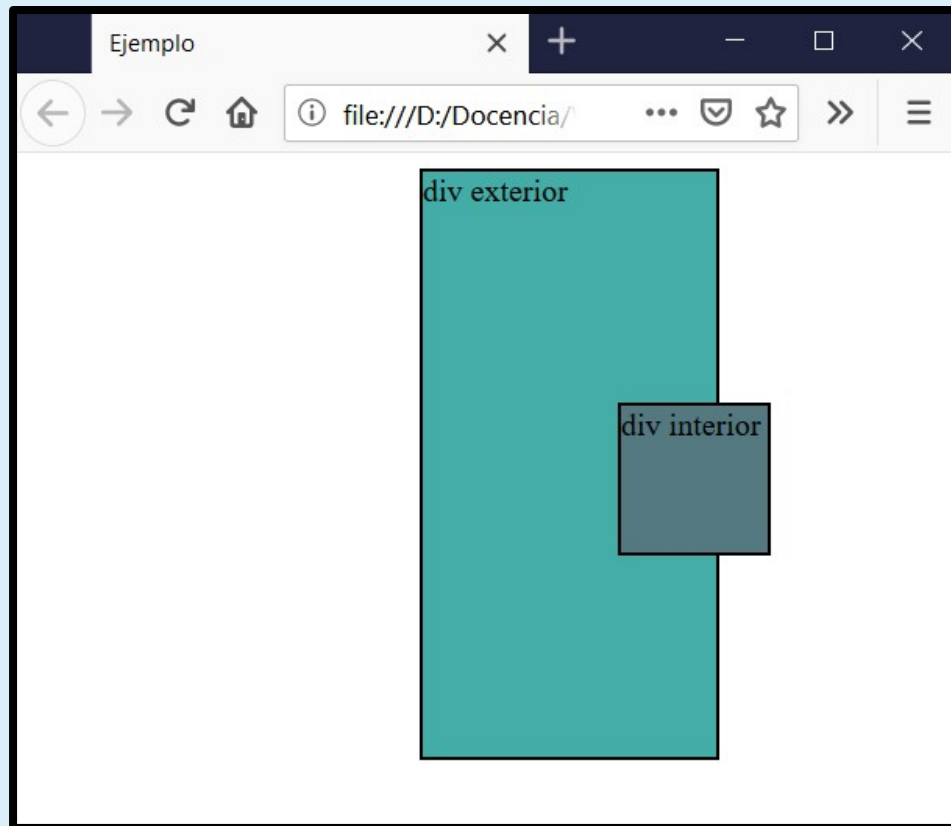
8.3. Posicionamiento relativo



```
1  @charset "utf-8";
2  div {
3      border: 2px solid black;
4  }
5
6  #interior {
7      height: 75px;
8      width: 75px;
9      background-color: #547980;
10 }
11
12 #exterior{
13     height: 300px;
14     width: 150px;
15     background-color: #45ADA8;
16     position: relative;
17     left: 100px;
18     top: 100px;
19 }
```

8. Posicionamiento

8.3. Posicionamiento relativo



```
1 @charset "utf-8";
2 div {
3     border: 2px solid black;
4 }
5
6 #interior {
7     height: 75px;
8     width: 75px;
9     background-color: #547980;
10    position: relative;
11    left: 100px;
12    top: 100px;
13 }
14
15 #exterior {
16     height: 300px;
17     width: 150px;
18     background-color: #45ADA8;
19     position: static;
20     /* Uso margin-left no confundir con left */
21     margin-left: 200px;
22 }
```

8. Posicionamiento

8.4. Posicionamiento fijo

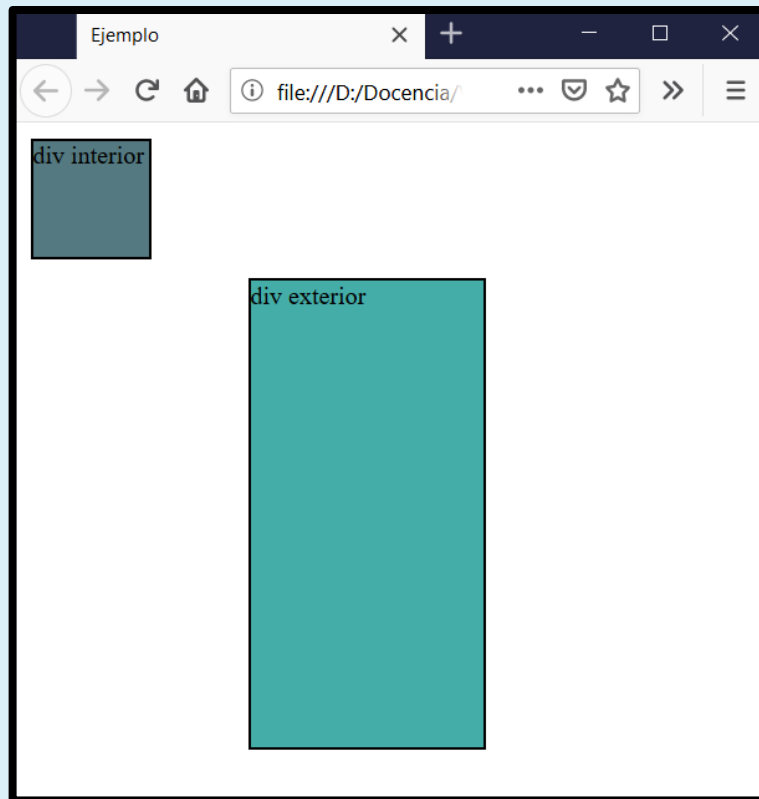
- ✓ Ancla un elemento a la ventana del navegador. Si desplazas la pantalla hacia arriba y hacia abajo, el elemento fijo se mantiene en su lugar. Su posición final será siempre fija.

```
10 <body>
11   <div id="exterior">
12     div exterior
13     <div id="interior">
14       div interior
15     </div>
16   </div>
17 </body>
```

```
1 @charset "utf-8";
2 div {
3   border: 2px solid black;
4 }
5
6 #interior {
7   height: 75px;
8   width: 75px;
9   background-color: #547980;
10  position: fixed;
11  left: 10px;
12  top: 10px;
13 }
14
15 #exterior {
16   height: 300px;
17   width: 150px;
18   background-color: #45ADA8;
19   position: absolute;
20   top: 100px;
21   left: 150px;
22 }
```

8. Posicionamiento

8.4. Posicionamiento fijo



```
1  @charset "utf-8";
2  div {
3      border: 2px solid black;
4  }
5
6  #interior {
7      height: 75px;
8      width: 75px;
9      background-color: #547980;
10     position: fixed;
11     left: 10px;
12     top: 10px;
13 }
14
15 #exterior{
16     height: 300px;
17     width: 150px;
18     background-color: #45ADA8;
19     position: absolute;
20     top: 100px;
21     left: 150px;
22 }
```

8. Posicionamiento

8.5. Posicionamiento flotante

- ✓ Se define usando las propiedades **float** y **clear**.
- ✓ Este tipo de posicionamiento se emplea principalmente a la hora de definir el esqueleto de un sitio web (cabecera, menú, contenido y pie).
- ✓ La **propiedad float** desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

<u>float</u>	Specifies whether or not a box should float	left right none inherit
--------------	---	----------------------------------

- ✓ Cuando usas float con un elemento en la página, le estás indicando a la página: "Estoy a punto de decirte donde ubicar este elemento, pero tienes que colocarlo dentro del flujo de los demás elementos." Esto quiere decir que si tienes varios elementos flotantes, todos saben que los demás están allí, y ninguno se superpone.

8. Posicionamiento

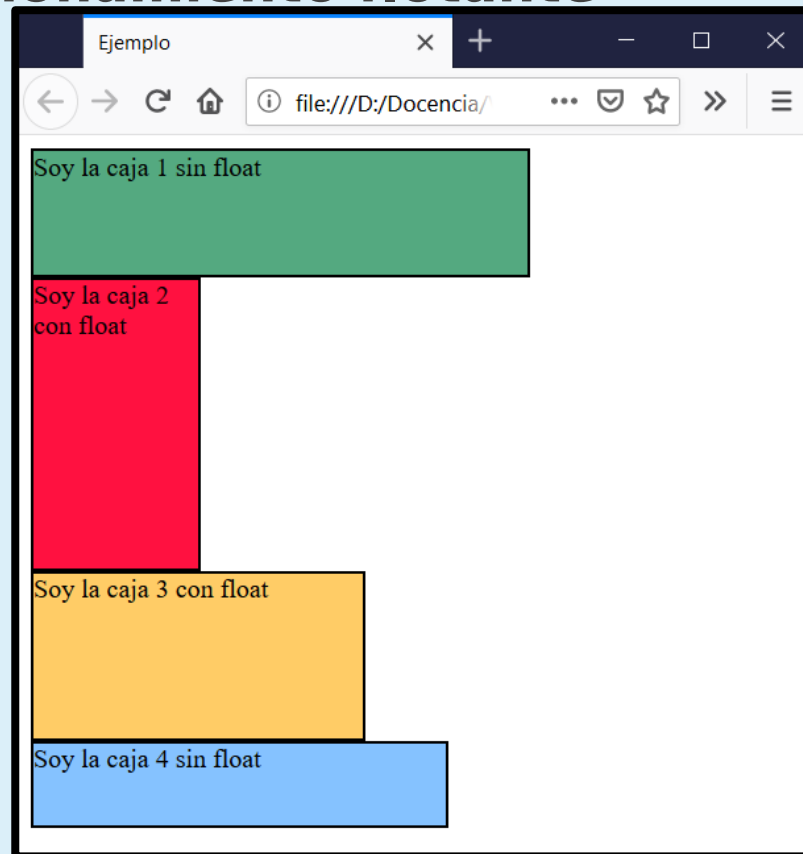
8.5. Posicionamiento flotante

```
10 <body>
11     <div id="caja1">Soy la caja 1 sin float</div>
12     <div id="caja2">Soy la caja 2 con float</div>
13     <div id="caja3">Soy la caja 3 con float</div>
14     <div id="caja4">Soy la caja 4 sin float</div>
15 </body>
```

```
1  @charset "utf-8";
2  div {
3      border: 2px solid black;
4  }
5  #caja1 {
6      height: 75px;
7      width: 300px;
8      background-color: #54A980;
9  }
10 #caja2 {
11     height: 175px;
12     width: 100px;
13     background-color: #FF1140;
14 }
15 #caja3 {
16     height: 100px;
17     width: 200px;
18     background-color: #FFCC66;
19 }
20 #caja4 {
21     height: 50px;
22     width: 250px;
23     background-color: #85C2FF;
24 }
```

8. Posicionamiento

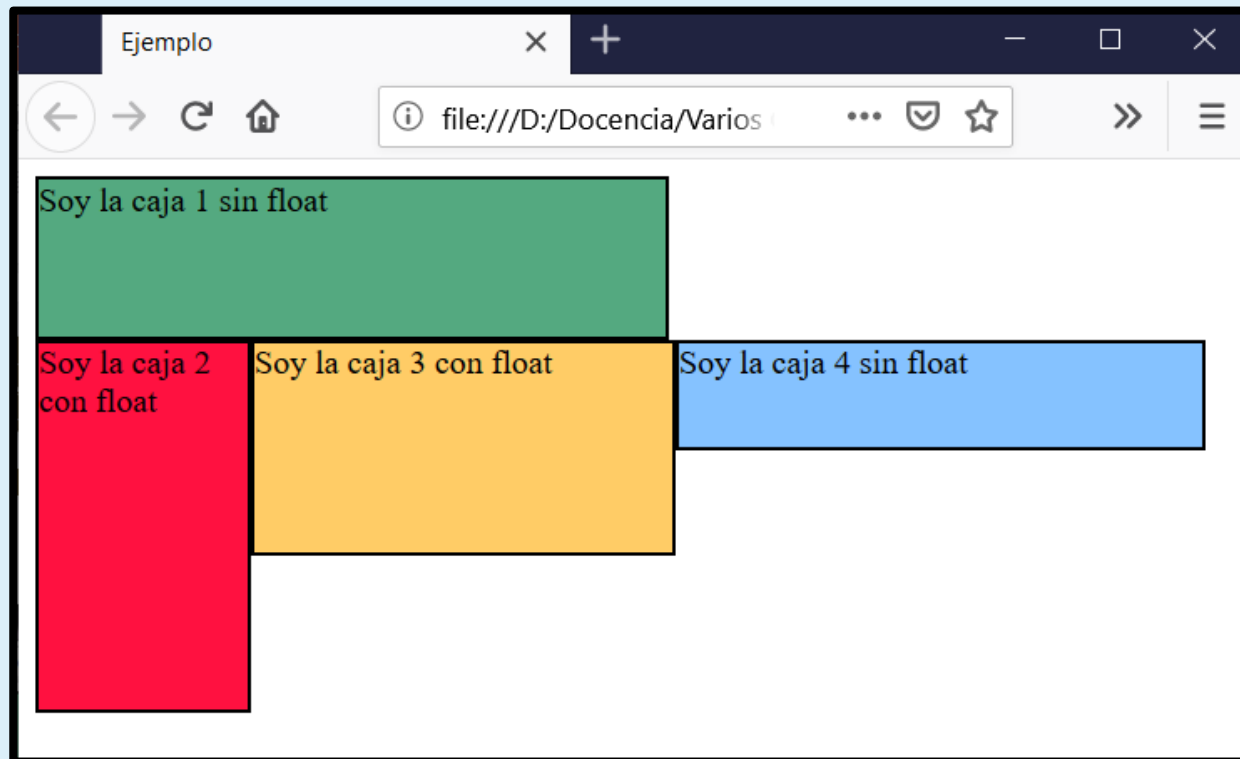
8.5. Posicionamiento flotante



```
1 @charset "utf-8";
2 div {
3     border: 2px solid black;
4 }
5 #caja1 {
6     height: 75px;
7     width: 300px;
8     background-color: #54A980;
9 }
10 #caja2 {
11     height: 175px;
12     width: 100px;
13     background-color: #FF1140;
14 }
15 #caja3 {
16     height: 100px;
17     width: 200px;
18     background-color: #FFCC66;
19 }
20 #caja4 {
21     height: 50px;
22     width: 250px;
23     background-color: #85C2FF;
24 }
```


8. Posicionamiento

8.5. Posicionamiento flotante

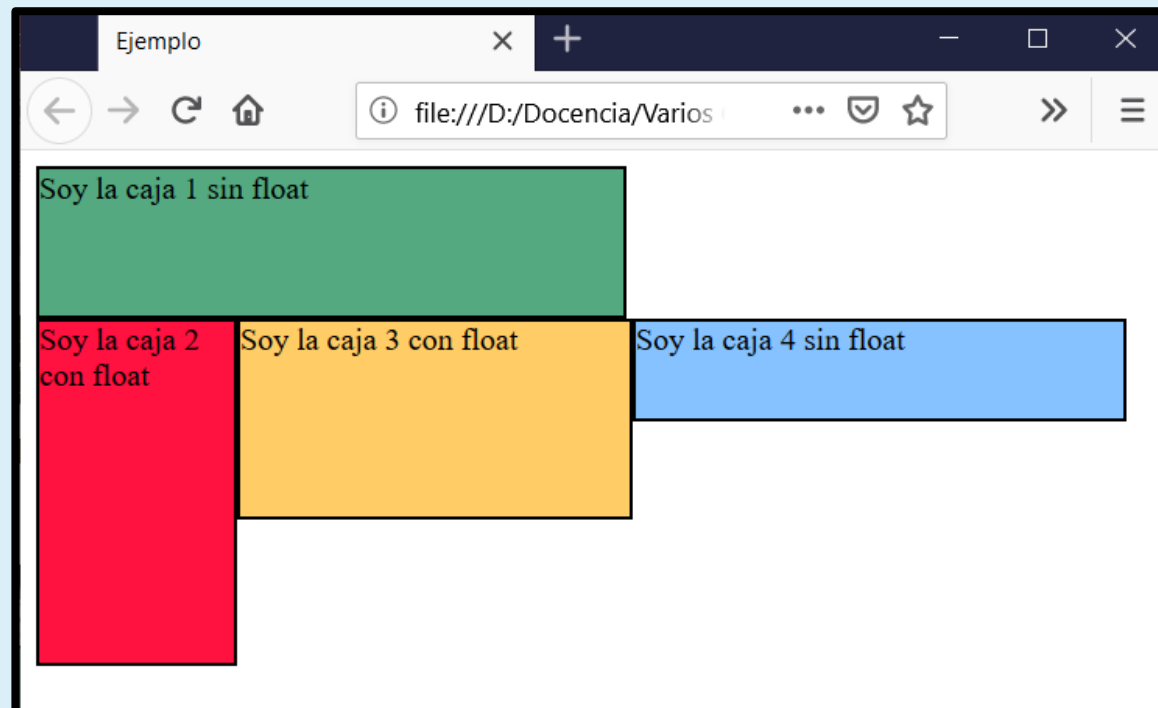


```
5 #caja1 {
6     height: 75px;
7     width: 300px;
8     background-color: #54A980;
9 }
10 #caja2 {
11     height: 175px;
12     width: 100px;
13     background-color: #FF1140;
14     float: left;
15 }
16 #caja3 {
17     height: 100px;
18     width: 200px;
19     background-color: #FFCC66;
20     float: left;
21 }
22 #caja4 {
23     height: 50px;
24     width: 250px;
25     background-color: #85C2FF;
26     float: left;
27 }
```

8. Posicionamiento

8.5. Posicionamiento flotante

✓ ¿Qué ocurre si dimensiono la ventana?



8. Posicionamiento

8.5. Posicionamiento flotante

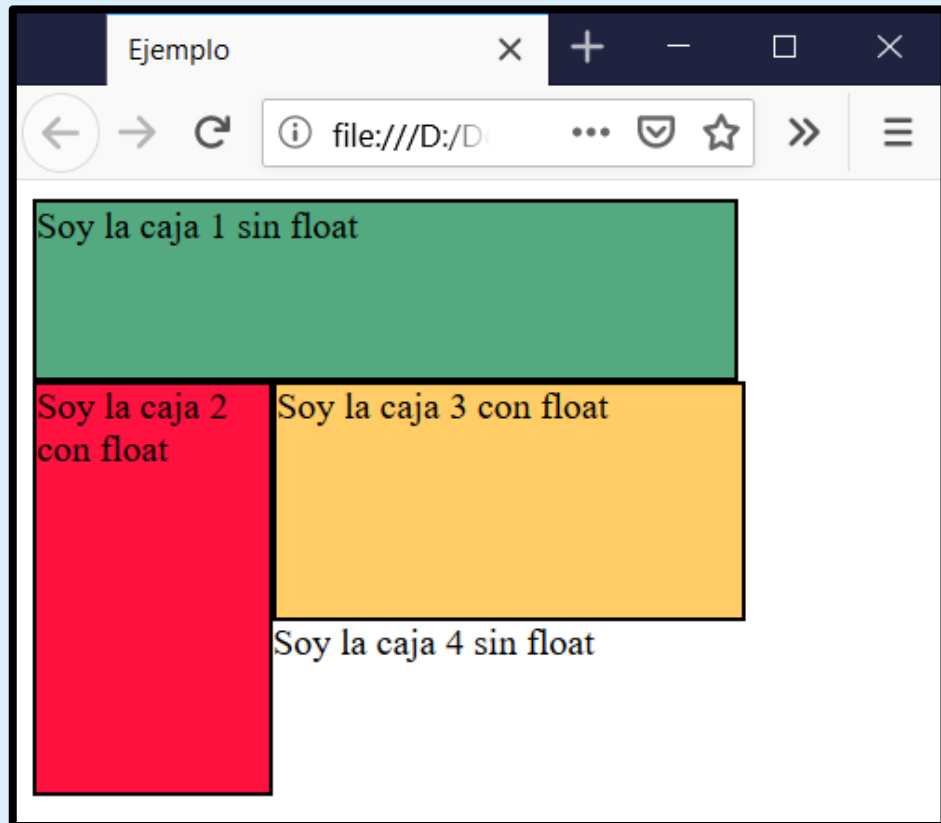
✓ La propiedad `clear`

- Por desgracia, a veces combinamos elementos grandes, flotantes y no flotantes, y los elementos terminan superponiéndose. Si usas la propiedad `clear: left` en un elemento, este inmediatamente se desplazará debajo de cualquier elemento flotante en el lado izquierdo de la página; también se puede usar `clear: right` para desplazarlo hacia el lado derecho. Y también puedes usar `clear: both` para que se aparte de los elementos que están en la izquierda y en la derecha.

Property	Description	Values
<u><code>clear</code></u>	Specifies which sides of an element where other floating elements are not allowed	left right both none inherit

8. Posicionamiento

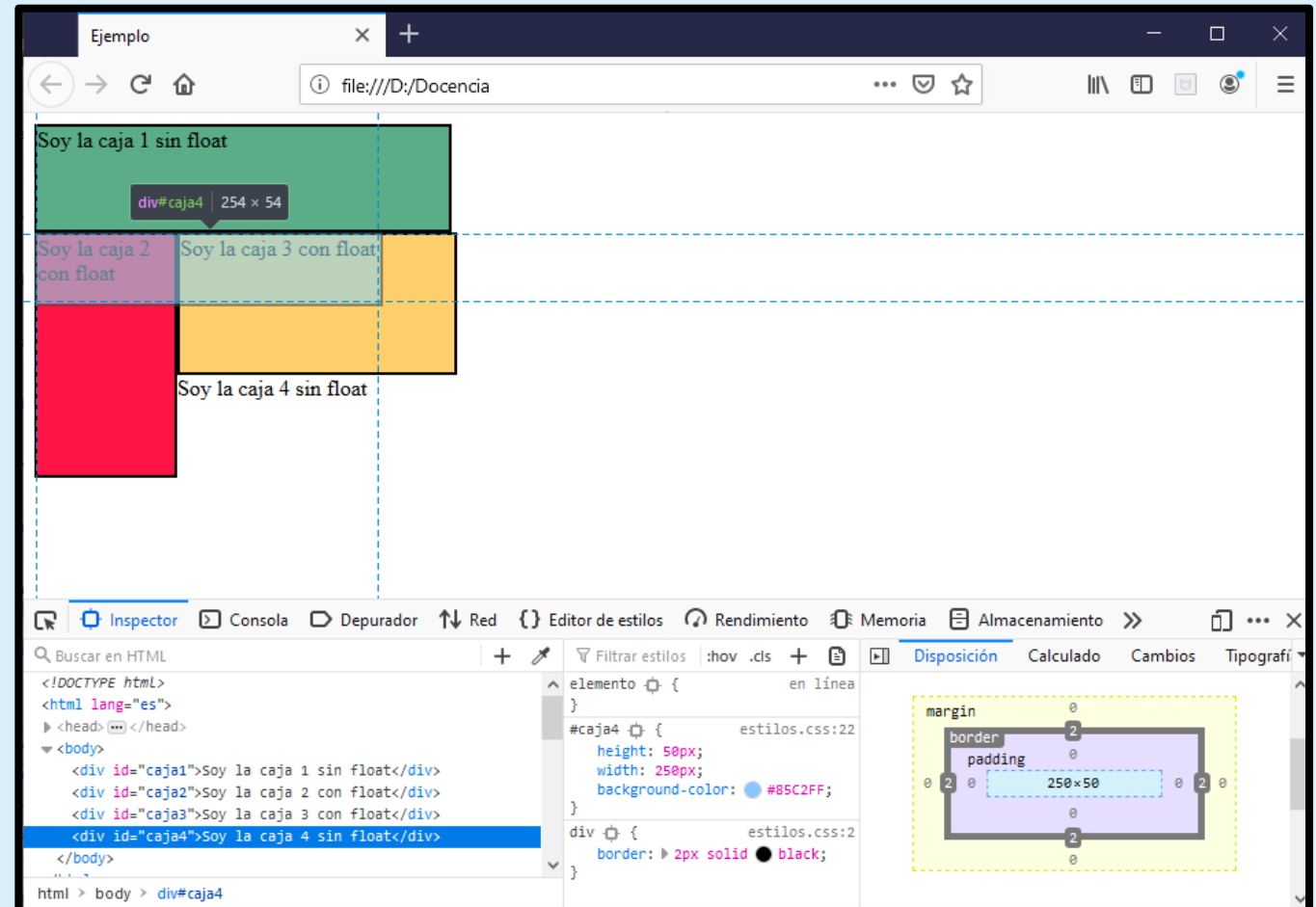
8.5. Posicionamiento flotante



```
5 #caja1 {  
6     height: 75px;  
7     width: 300px;  
8     background-color: #54A980;  
9 }  
10 #caja2 {  
11     height: 175px;  
12     width: 100px;  
13     background-color: #FF1140;  
14     float:left;  
15 }  
16 #caja3 {  
17     height: 100px;  
18     width: 200px;  
19     background-color: #FFCC66;  
20     float:left;  
21 }  
22 #caja4 {  
23     height: 50px;  
24     width: 250px;  
25     background-color: #85C2FF;  
26 }
```

8. Posicionamiento

8.5. Posicionamiento flotante



8. Posicionamiento

8.5. Posicionamiento flotante



```
5 #caja1 {  
6     height: 75px;  
7     width: 300px;  
8     background-color: #54A980;  
9 }  
10 #caja2 {  
11     height: 175px;  
12     width: 100px;  
13     background-color: #FF1140;  
14     float:left;  
15 }  
16 #caja3 {  
17     height: 100px;  
18     width: 200px;  
19     background-color: #FFCC66;  
20     float:left;  
21 }  
22 #caja4 {  
23     height: 50px;  
24     width: 250px;  
25     background-color: #85C2FF;  
26     clear: both;  
27 }
```

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
- 9. Visualización**
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

9. Visualización

- ✓ Además de las propiedades que controlan el posicionamiento de los elementos, CSS define otras cuatro propiedades para controlar su visualización:
 - **display:** permite ocultar elementos HTML y que no ocupen espacio en la página web (no existen para el navegador). Además permite cambiar el comportamiento de los elementos, para que un elemento en bloque se comporte como un elemento en línea y a la inversa.
 - **visibility:** permite ocultar elementos HTML, pero esos elementos siguen ocupando su espacio en la página web.
 - **z-index:** permite controlar la posición tridimensional de las cajas posicionadas. Es posible indicar las cajas que se muestran delante o detrás de otras cajas cuando se producen solapamientos.
 - **overflow:** permite controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos (scroll).

9. Visualización

- ✓ Los valores que puede tomar la propiedad **visibility** son:

Value	Description
visible	Default. The element is visible
hidden	The element is invisible (but still takes up space)
collapse	Only for table elements. collapse removes a row or column, but it does not affect the table layout. The space taken up by the row or column will be available for other content. If collapse is used on other elements, it renders as "hidden"
inherit	Specifies that the value of the visibility property should be inherited from the parent element

https://www.w3schools.com/cssref/pr_class_visibility.asp

9. Visualización

✓ Ejemplo de *visibility*:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {visibility:hidden;}
</style>
</head>

<body>
<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up
space.</p>
</body>
</html>
```

This is a visible heading

Notice that the hidden heading still takes up space.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {visibility:hidden;}
</style>
</head>

<body>
<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up
space.</p>
</body>
</html>
```

This is a visible heading

Notice that the hidden heading still takes up space.

9. Visualización

- ✓ Los valores que puede tomar la propiedad **display** son:

Value	Description
inline	Default. Displays an element as an inline element (like)
block	Displays an element as a block element (like <p>)
inline-block	Displays an element as an inline-level block container. The inside of this block is formatted as block-level box, and the element itself is formatted as an inline-level box
none	The element will not be displayed at all (has no effect on layout)
inherit	The value of the display property will be inherited from the parent element

Nota: Existen muchos más valores para la propiedad **display** en:
https://www.w3schools.com/cssref/pr_class_display.asp

9. Visualización

✓ Ejemplo de *display*:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {display:none;}
</style>
</head>

<body>
<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading does not take up
space.</p>
</body>
</html>
```

This is a visible heading

Notice that the hidden heading does not take up space.

9. Visualización

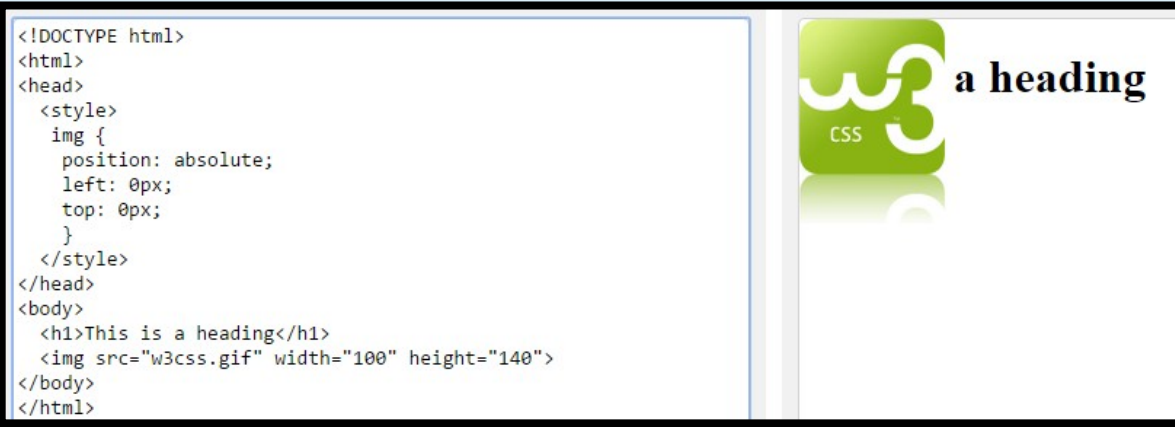
- ✓ Los valores que puede tomar la propiedad **z-index** son:

Value	Description
auto	Sets the stack order equal to its parents. This is default
<i>number</i>	Sets the stack order of the element. Negative numbers are allowed
initial	Sets this property to its default value. <u>Read about initial</u>
inherit	Inherits this property from its parent element. <u>Read about inherit</u>

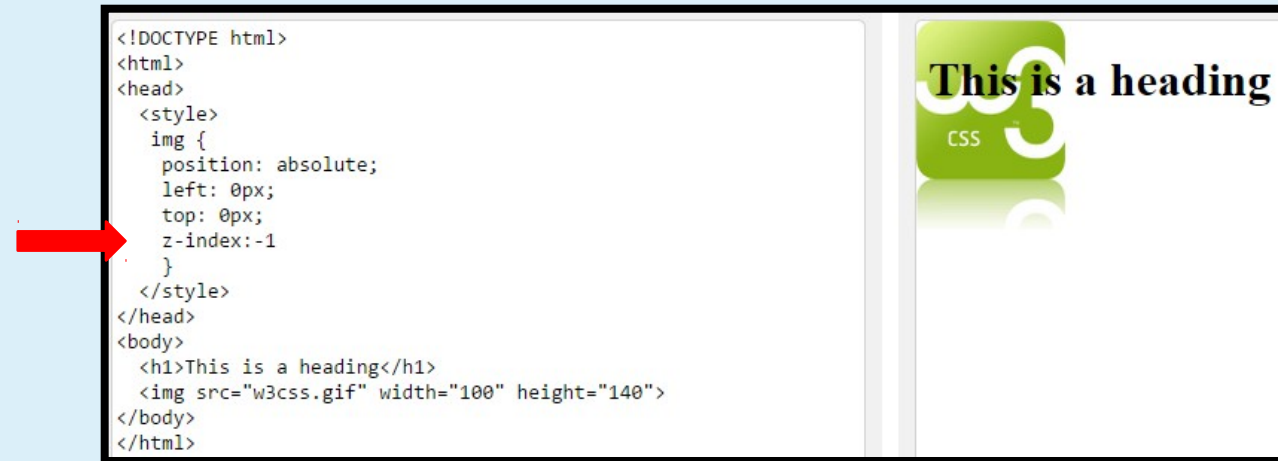
https://www.w3schools.com/cssref/pr_pos_z-index.asp

9. Visualización

✓ Ejemplo sin “z-index”:



✓ Ejemplo con “z-index”:



9. Visualización

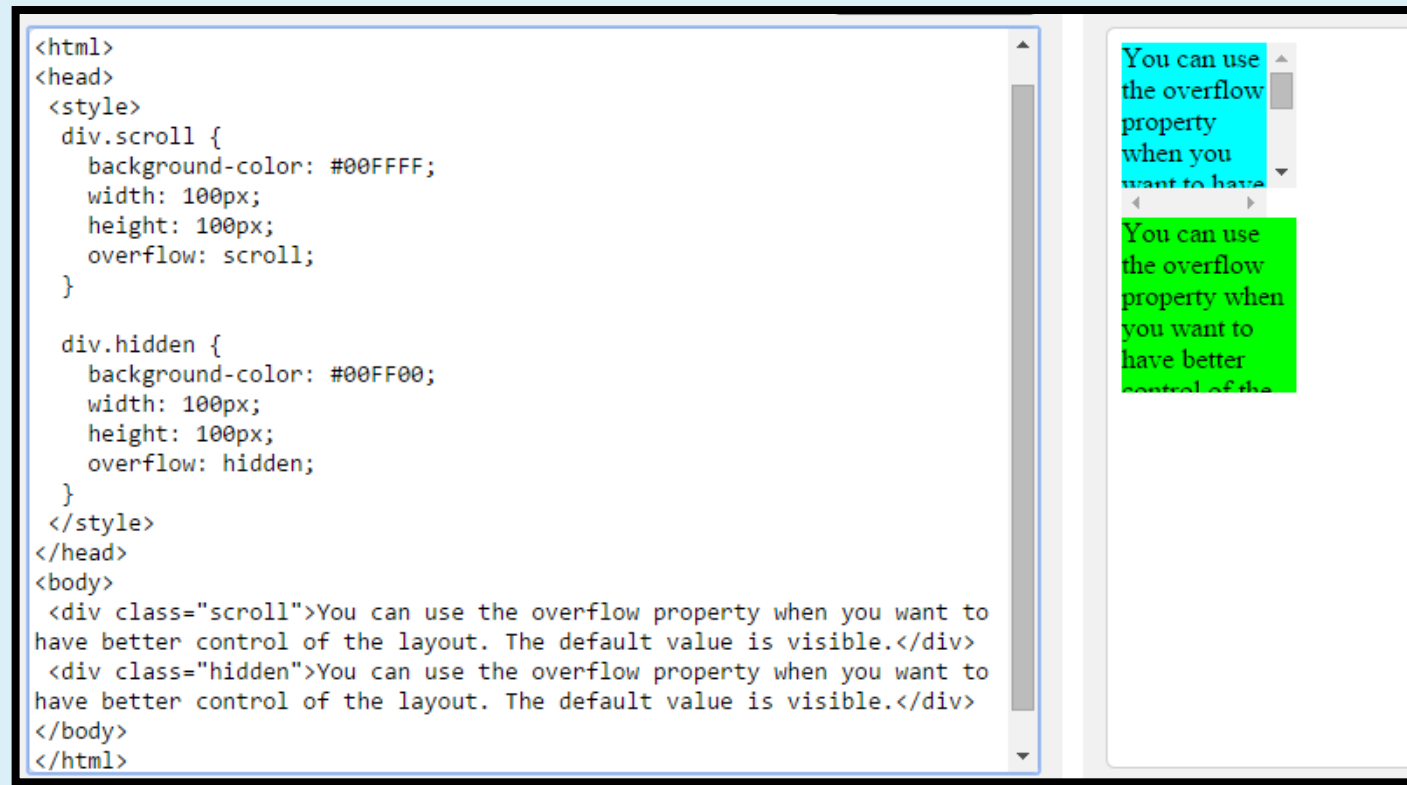
- ✓ Los valores que puede tomar la propiedad **overflow** son:

Value	Description
visible	The overflow is not clipped. It renders outside the element's box. This is default
hidden	The overflow is clipped, and the rest of the content will be invisible
scroll	The overflow is clipped, but a scroll-bar is added to see the rest of the content
auto	If overflow is clipped, a scroll-bar should be added to see the rest of the content
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

https://www.w3schools.com/cssref/pr_pos_overflow.asp

9. Visualización

- ✓ Ejemplo de *overflow*:

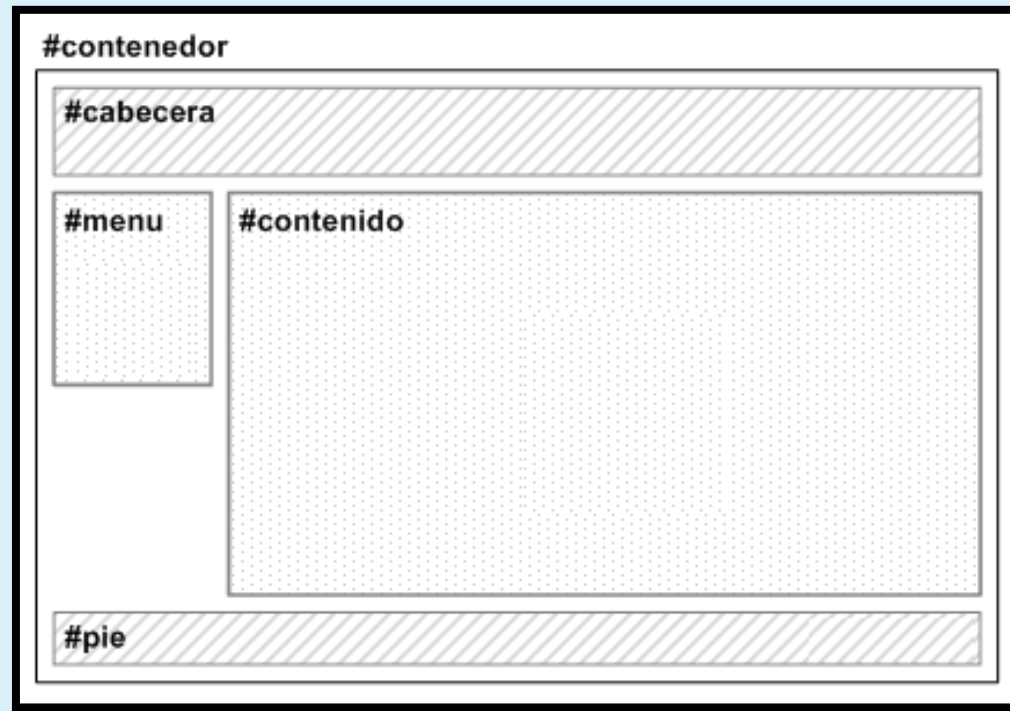


ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
- 10. Definición de las cajas contenedoras**
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

10. Definición de las cajas contenedoras

- ✓ Este es el diseño de una página web sencilla:



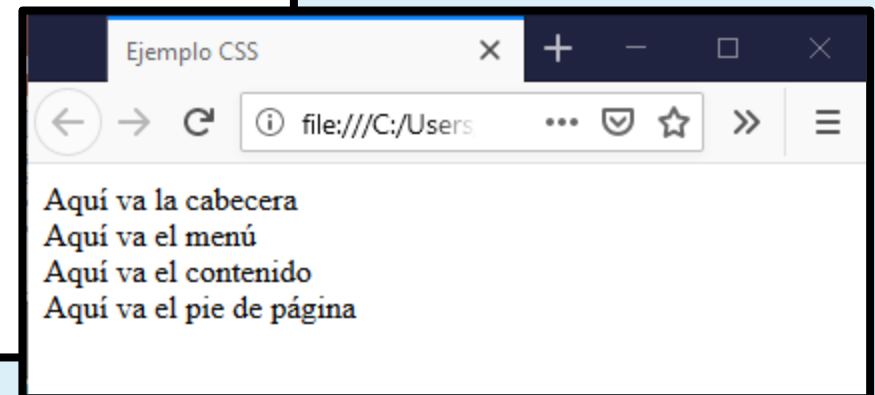
10. Definición de las cajas contenedoras

- ✓ El primer paso es definir en HTML las distintas cajas contenedoras que tendrá mi sitio web:

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <title>Ejemplo CSS</title>
5      <meta charset="UTF-8">
6      <link href="css/miestilo.css" rel="stylesheet" type="text/css" />
7  </head>
8  <body>
9      <div id="contenedor">
10         <div id="cabecera">Aquí va la cabecera
11         </div>
12         <div id="menu"> Aquí va el menú
13         </div>
14         <div id="contenido">Aquí va el contenido
15         </div>
16         <div id="pie">Aquí va el pie de página
17         </div>
18     </div>
19 </body>
20 </html>
```

10. Definición de las cajas contenedoras

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <title>Ejemplo CSS</title>
5      <meta charset="UTF-8">
6      <link href="css/miestilo.css" rel="stylesheet" type="text/css" />
7  </head>
8  <body>
9      <div id="contenedor">
10         <div id="cabecera">Aquí va la cabecera
11         </div>
12         <div id="menu"> Aquí va el menú
13         </div>
14         <div id="contenido">Aquí va el contenido
15         </div>
16         <div id="pie">Aquí va el pie de página
17         </div>
18     </div>
19 </body>
20 </html>
```

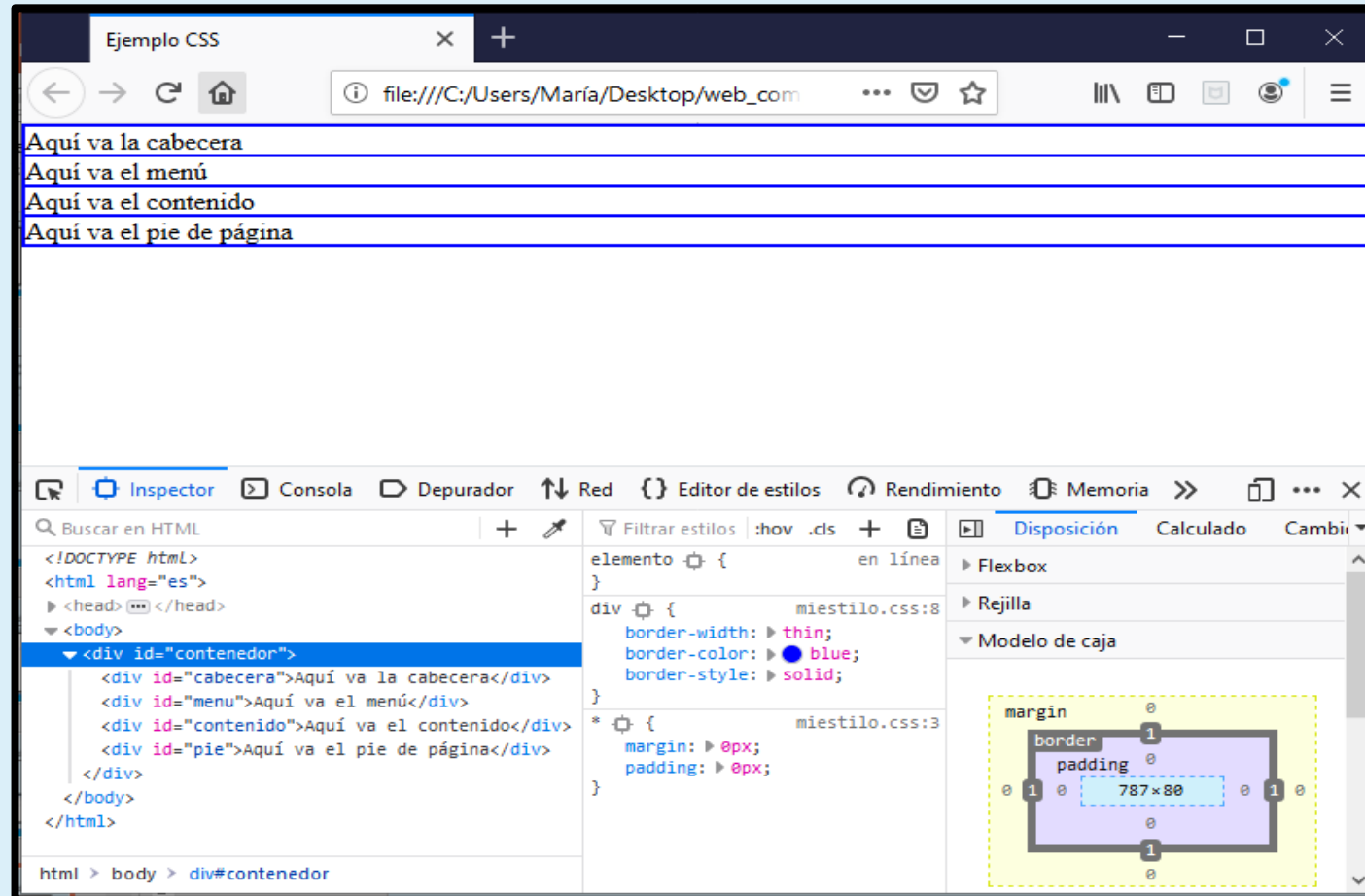


10. Definición de las cajas contenedoras

- ✓ El segundo paso es crear el fichero de estilos.
- ✓ En este fichero a cada caja le voy a dar un primer estilo de borde fino, solido y en azul:

```
1  @charset "utf-8";
2  /* Mis estilos CSS */
3  * {
4      margin: 0px;
5      padding: 0px;
6  }
7
8  div {
9      border-width: thin;
10     border-color: blue;
11     border-style: solid;
```

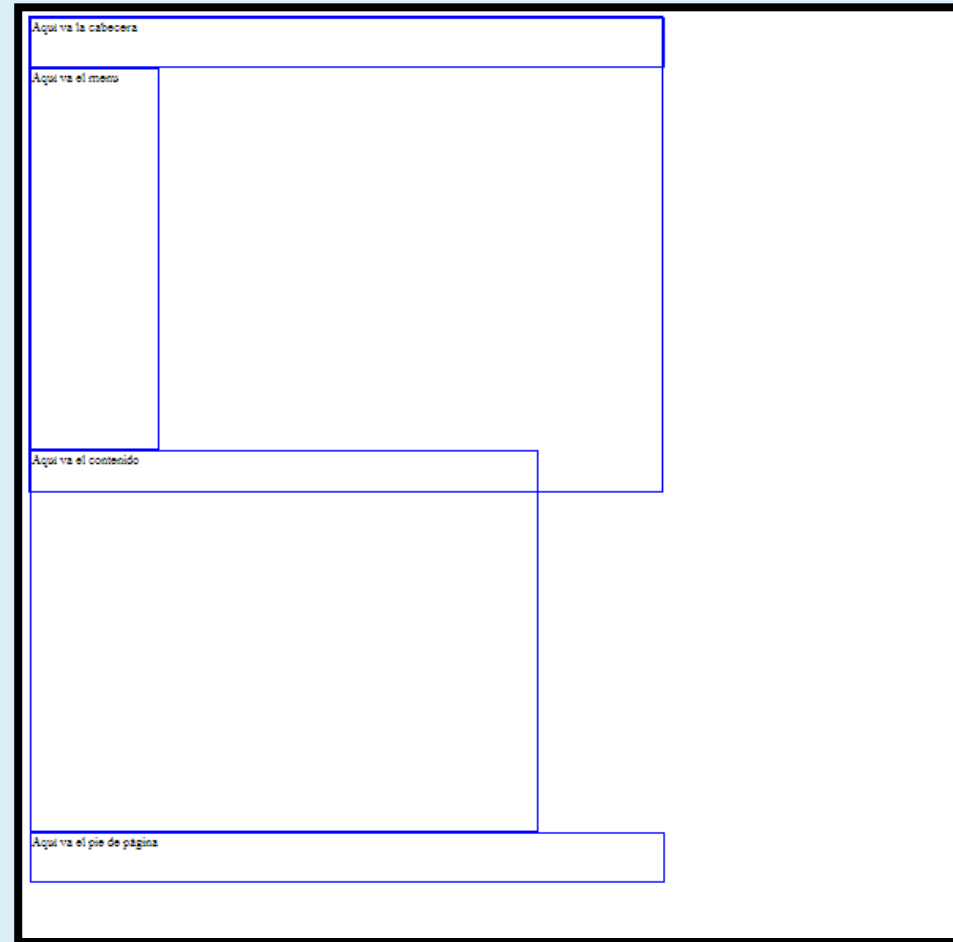
10. Definición de las cajas contenedoras



10. Definición de las cajas contenedoras

- ✓ La página tendrá un tamaño fijo de 800x600px (a modo de ejemplo).

```
12  #contenedor {  
13      width: 800px;  
14      height: 600px;  
15  }  
16  #cabecera{  
17      width: 100%;  
18      height: 10%;  
19  }  
20  #menu {  
21      width: 20%;  
22      height: 80%;  
23  }  
24  #contenido {  
25      width: 80%;  
26      height: 80%;  
27  }  
28  #pie{  
29      width: 100%;  
30      height: 10%;  
31  }
```



ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
- 11. Posicionamiento de las cajas contenedoras**
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. Bibliografía

11. Posicionamiento de las cajas contenedoras

- ✓ Para posicionar las cajas siguiendo el esquema planteado, la solución en CSS se basa en el uso de la propiedad **float** para los elementos posicionados como el menú y los contenidos y el uso de la propiedad **clear** en el pie de página **para evitar los solapamientos ocasionados por los elementos posicionados con float**. La cabecera se queda donde está.

Recuerda:

Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una caja flotante, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.

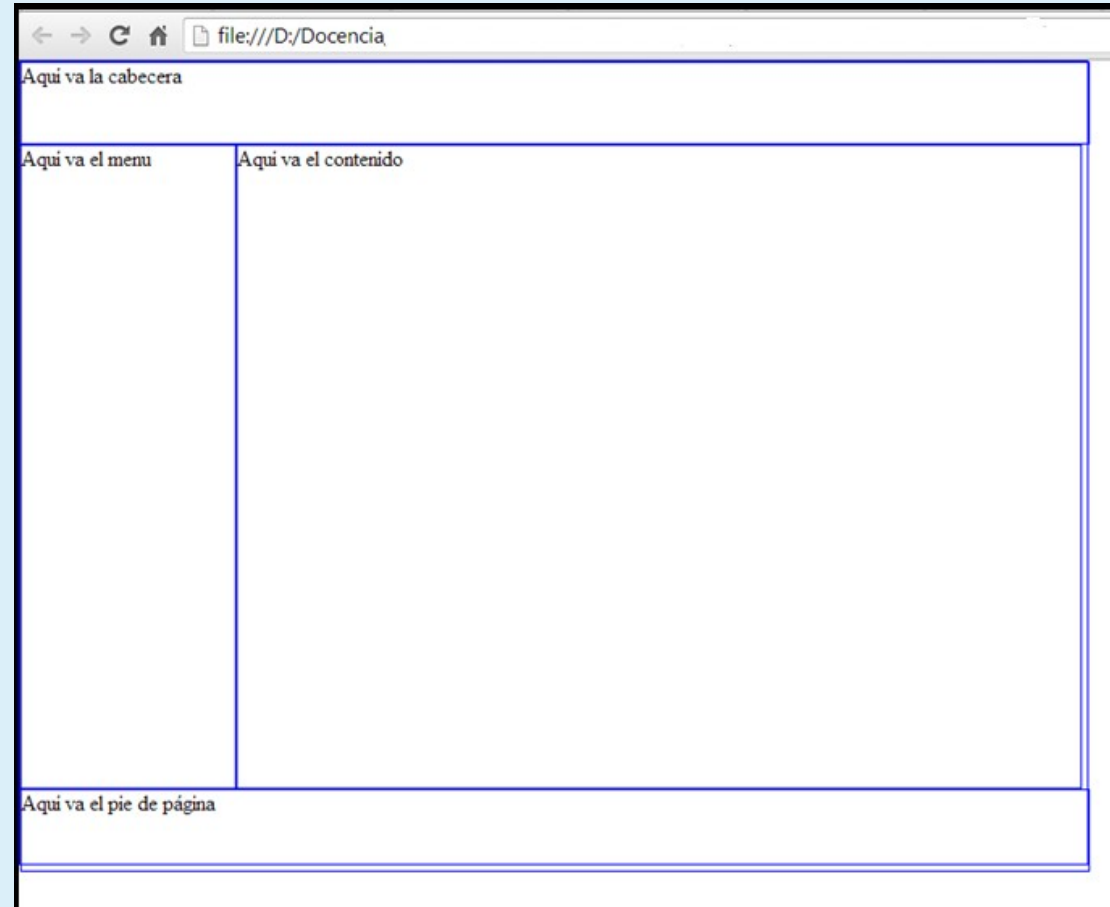
La propiedad clear permite modificar el comportamiento por defecto del posicionamiento flotante para **forzar a un elemento a mostrarse debajo de cualquier caja flotante**.

<https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-flotante>

11. Posicionamiento de las cajas contenedoras

```
16  #cabecera{
17      width: 100%;
18      height: 10%;
19  }
20  #menu {
21      width: 20%;
22      height: 80%;
23      float: left;
24  }
25  #contenido {
26      width: 79%;
27      /*Sería width: 79%;, pero recuerda
28      que estoy usando un borde azul
29      y eso ocupa un espacio.*/
30      height: 80%;
31      float: left;
32  }
33  #pie{
34      width: 100%;
35      height: 10%;
36      clear: both;
37  }
```

11. Posicionamiento de las cajas contenedoras



ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
- 12. Uso de medidas fijas**
13. Colores e imágenes de fondo
14. Bibliografía

12. Uso de medidas fijas

- ✓ Para dimensionar la caja "contenedor" hemos definido unas medidas fijas 800x600px. Eso presenta este problema:

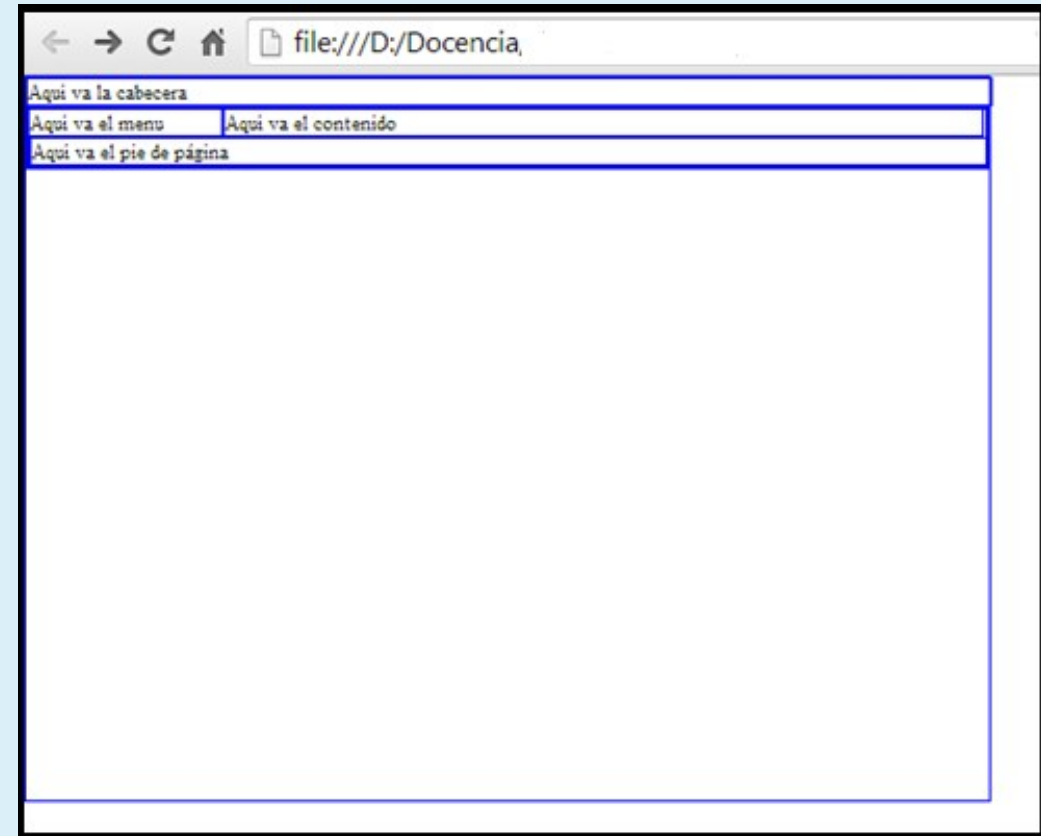
Aquí va la cabecera	
Aquí va el menu	Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo.
Aquí va el pie de página	Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo.

12. Uso de medidas fijas

✓ Solución 1:

```
12 #contenedor {  
13     width: 800px;  
14     min-height: 600px;  
15 }
```

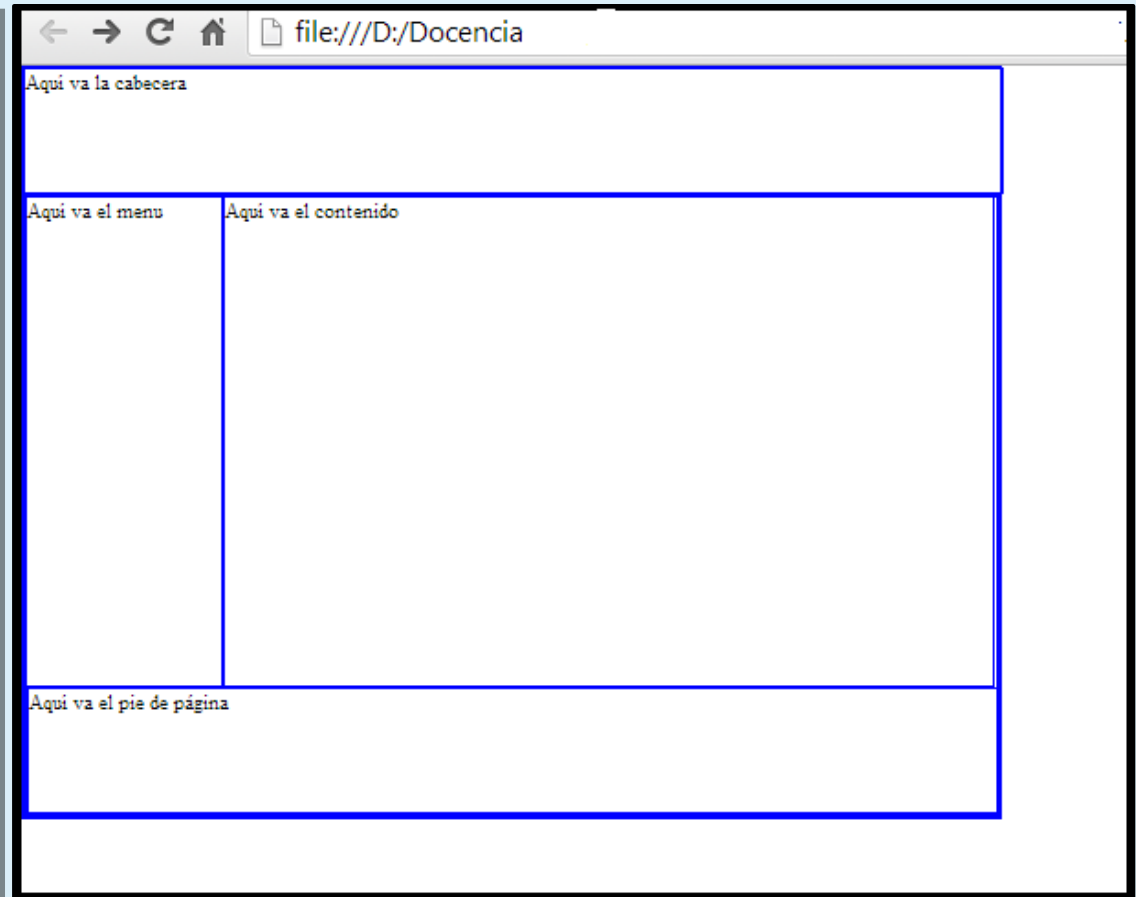
✓ Nuevo problema: →



12. Uso de medidas fijas

✓ Solución:

```
12 #contenedor {
13     width: 800px;
14     min-height: 600px;
15 }
16 #cabecera{
17     width: 100%;
18     height: 100px;
19 }
20 #menu {
21     width: 20%;
22     min-height: 400px;
23     float: left;
24 }
25 #contenido {
26     width: 79%;
27     min-height: 400px;
28     float: left;
29 }
30 #pie{
31     width: 100%;
32     height: 100px;
33     clear: both;
34 }
```



ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
- 13. Colores e imágenes de fondo**
14. Bibliografía

13. Colores e imágenes de fondo

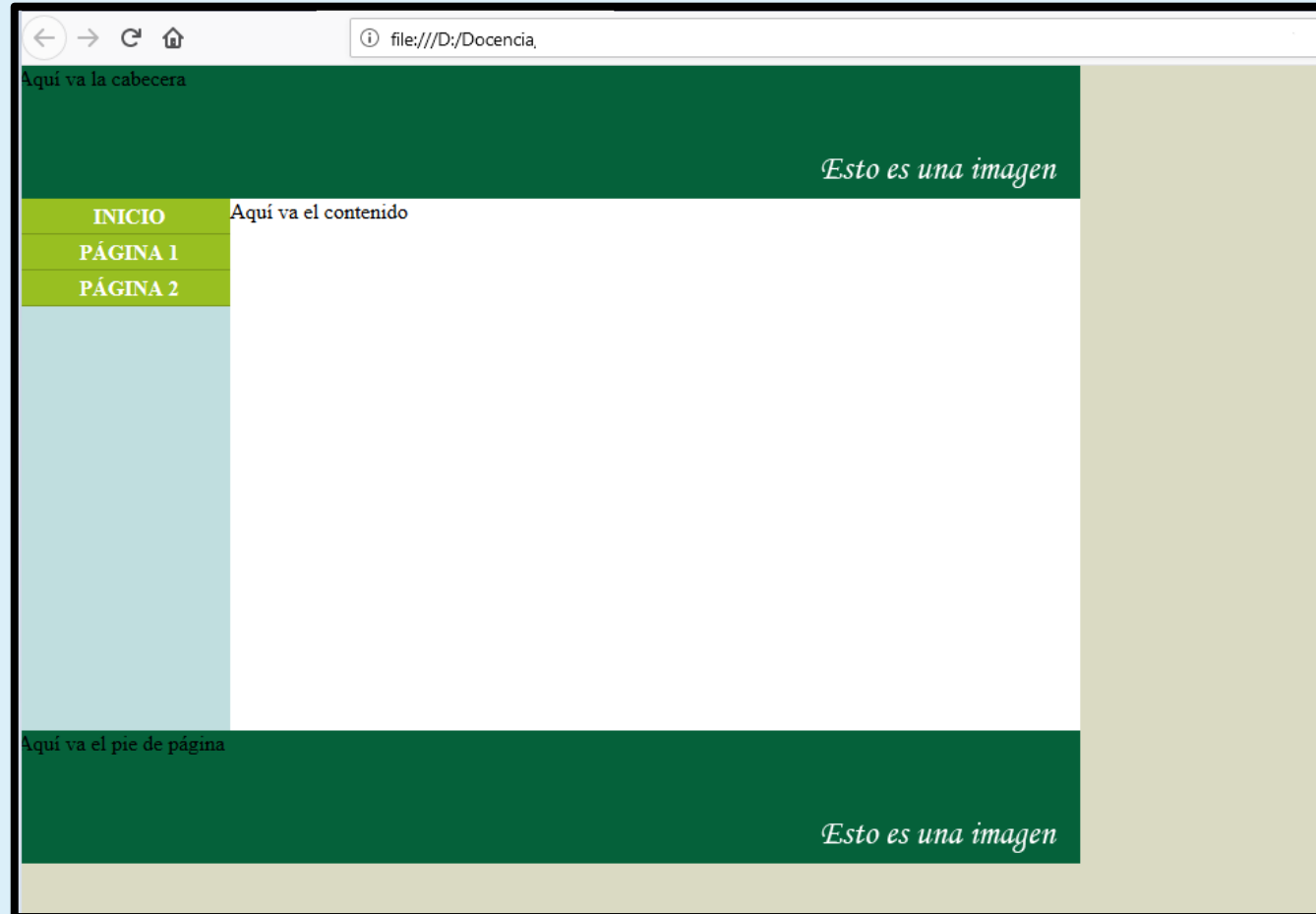
- ✓ En primer lugar voy a eliminar el borde azul de las cajas porque al darle colores e imágenes de fondo ya no es necesario remarcar las cajas.
- ✓ La cabecera y el pie de página serán dos imágenes ".png" de 800x100px.
- ✓ El menú y el contenido tendrá definido un color de fondo.
- ✓ Ya también aprovechamos para darle un color de fondo al <body>.

13. Colores e imágenes de fondo

```
7  body {  
8      background-color: #DADAC3;  
9  }  
10 #contenedor {  
11     width: 800px;  
12     min-height: 600px;  
13 }  
14 #cabecera {  
15     width: 100%;  
16     height: 100px;  
17     background: url("../img/cab.png");  
18 }  
19 #menu {  
20     width: 20%;  
21     min-height: 400px;  
22     float: left;  
23     background-color: #CODEDF;  
24 }  
25 #contenido {  
26     width: 80%;  
27     min-height: 400px;  
28     float: left;  
29     background-color: #FFFFFF;  
30 }
```

```
31 #pie {  
32     width: 100%;  
33     height: 100px;  
34     clear: both;  
35     background: url("../img/pie.png");  
36 }
```

13. Colores e imágenes de fondo



ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Propiedades y Valores
6. Pseudo-clases y Pseudo-elementos
7. Modelo de Caja
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
- 14. Bibliografía**

10. Bibliografía

- ✓ **Lenguajes de Marcas y Sistemas de Gestión de Información**
Sánchez, F. J. y otros. Editorial: RA-MA.
- ✓ **Uniwebsidad - Varios manuales**
<https://uniwebsidad.com/>
- ✓ **w3schools.com - HTML Tutorial & CSS Tutorial**
<https://www.w3schools.com/>
- ✓ **DesarrolloWeb.com - CSS a fondo**
<https://desarrolloweb.com/home/css>
- ✓ **w3c.org - HTML & CSS**
<https://www.w3.org/standards/webdesign/htmlcss>