

Unidad 3

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. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

ÍNDICE DE CONTENIDOS

1. **Introducción**
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. 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. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

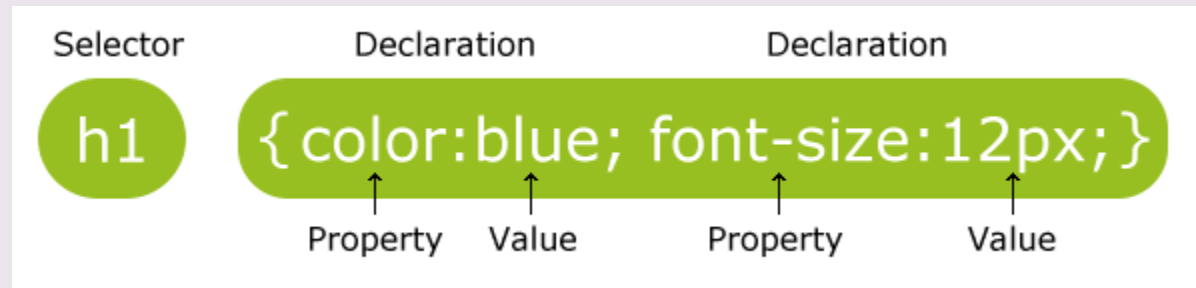
2. Reglas CSS

✓ 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 }
```

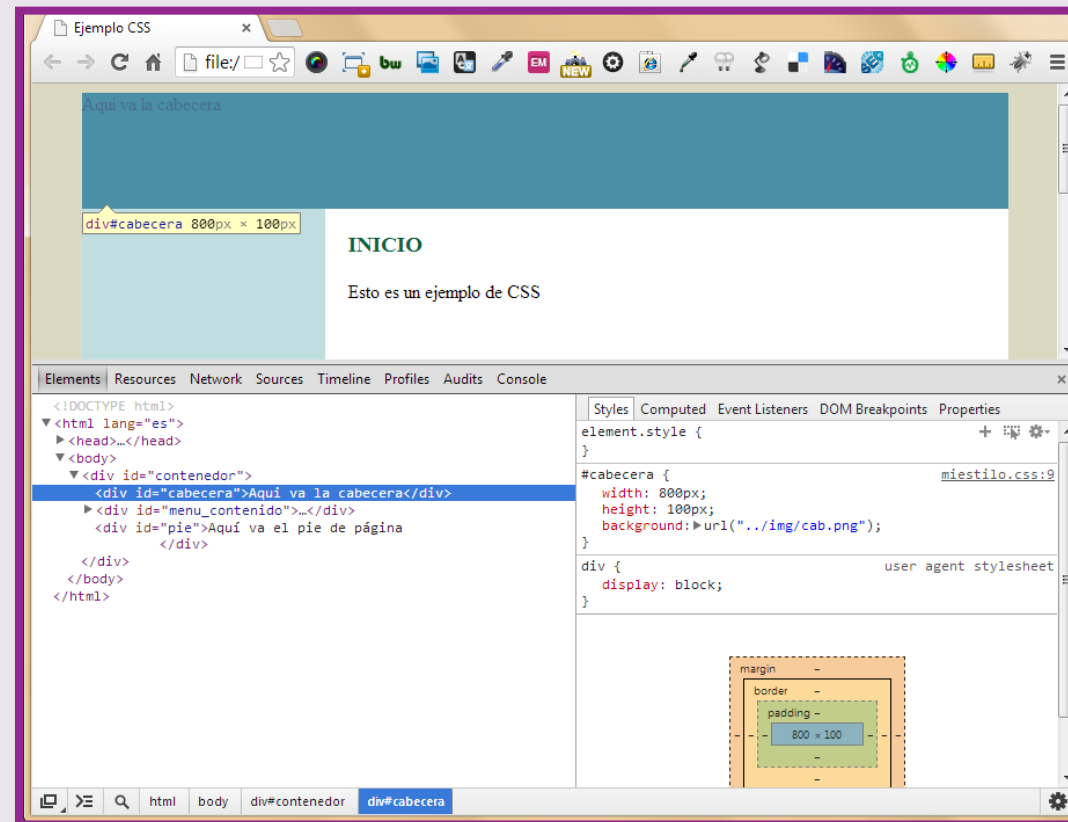
2. Reglas CSS

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 navegadores 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. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. 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="description" content="Ejemplo de CSS">
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?

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

➤ https://www.w3schools.com/css/css_howto.asp

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. **Selectores CSS**
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

4. Selectores CSS

- ✓ ***** → **Selector universal**. Se utiliza para seleccionar todos los elementos de la página.
- ✓ **etiqueta** → Seleccionar esa etiqueta.
- ✓ **.class** → Elemento con ese valor en el atributo clase.
- ✓ **#id** → Elemento con ese valor en el atributo id.
- ✓ **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

✓ 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

✓ 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

✓ 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

✓ 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

✓ 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](http://www.as.com)
[Un enlace a marca.com](http://www.marca.com)

4. Selectores CSS

✓ 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

✓ 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

✓ 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. **Pseudo-clases y Pseudo-elementos**
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

5. 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;  
}
```

5. 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()
- ...

5. 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.

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

5. 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 ;  
}
```

5. Pseudo-clases y Pseudo-elementos

✓ Pseudo-elementos:

- ::first-line
- ::first-letter
- ::after
- ::before
- ::selection
- ::marker

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

5. 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 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. Pseudo-clases y Pseudo-elementos
6. **Propiedades y Valores**
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

6. 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;  
}
```

6. 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
- ✓ 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.

6. 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

6. 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-clip</u>	Specifies the painting area of the background
<u>background-color</u>	Sets the background color of an element
<u>background-image</u>	Sets the background image for an element
<u>background-origin</u>	Specifies where the background image(s) is/are positioned
<u>background-position</u>	Sets the starting position of a background image
<u>background-repeat</u>	Sets how a background image will be repeated
<u>background-size</u>	Specifies the size of the background image(s)

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

6. Propiedades y Valores

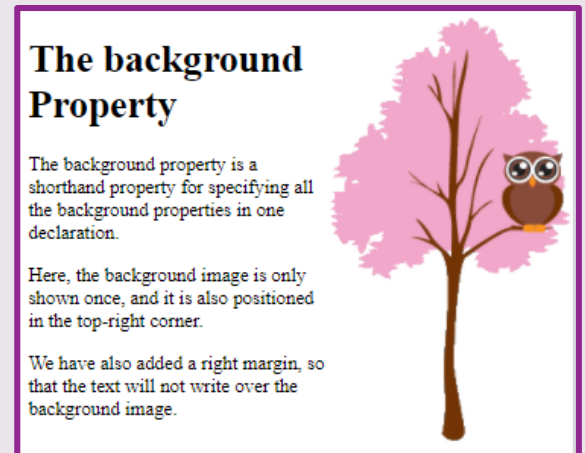
✓ Fondos:

- **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-color: #ffffff;  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
}
```



```
body {  
  background: #ffffff url("img_tree.png") no-repeat right top;  
  margin-right: 200px;  
}
```



6. 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

6. 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   }
```

6. 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 de fuente del elemento actual.
 - **rem:** en relación al tamaño de fuente que tiene la etiqueta HTML ("root em").
 - **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" (carácter "0")

6. 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>text-overflow</u>	Specifies how overflowed content that is not displayed should be signaled to the user
<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

6. 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;}
```

6. 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

6. 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;}
```


6. 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

```
p.a {  
    font: 20px Arial, sans-serif;  
}  
  
p.b {  
    font: italic bold 12px/30px Georgia, serif;  
}
```

6. 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 the position of the list-item markers (bullet points)
<u>list-style-type</u>	Specifies the type of list-item marker

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

6. 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

6. 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

6. Propiedades y Valores

✓ Listas:

- list-style-position:

Value	Description
inside	The bullet points will be inside the list item
outside	The bullet points will be outside the list item. This is default
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

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

Outside:

- Coffee
- Tea
- Coca-cola

Inside:

- Coffee
- Tea
- Coca-cola

6. 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);  
}
```

6. 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;}
```

6. 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

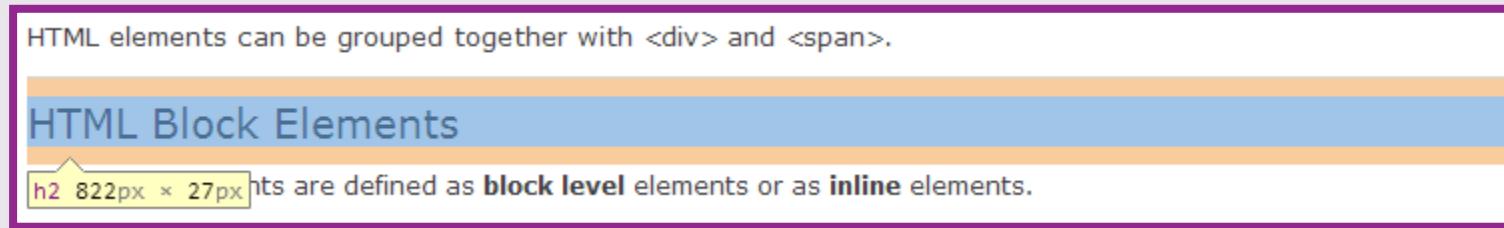
```
a:link {color:#FF0000;
        text-decoration:none;
        background-color:#B2FF99;
      }
a:visited {color:#00FF00;
            text-decoration:none;
            background-color:#FFFF85;
          }
a:hover {color:#FF00FF;
          text-decoration:underline;
          background-color:#FF704D;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


ÍNDICE DE CONTENIDOS

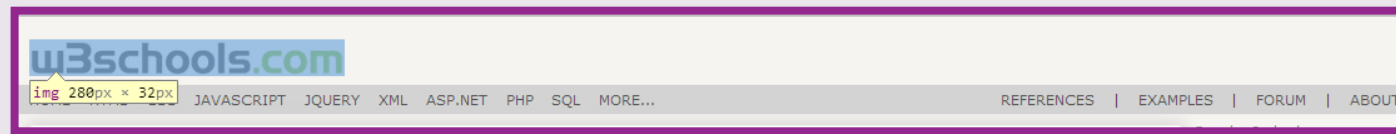
1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. **Modelo de Cajas**
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
11. Posicionamiento de las cajas contenedoras
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

7. Modelo de Cajas

- ✓ 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 del comportamiento de una etiqueta `<h1>` (de **bloque**):



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



7. Modelo de Cajas

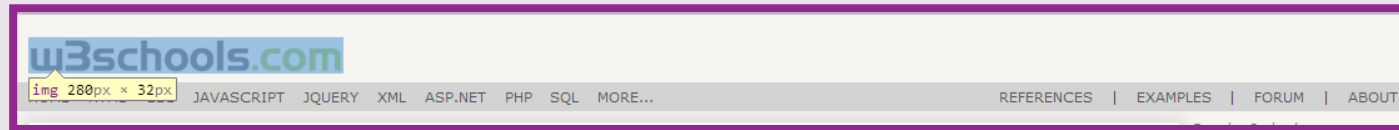
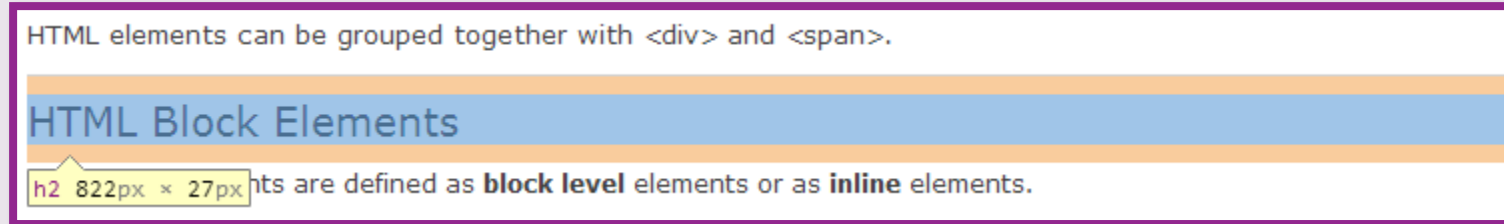
- ✓ Los **elementos en línea** definidos por HTML son:
 - a, abbr, b, bdo, br, cite, code, dfn, em, i, img, input, label, q, s, samp, select, small, span, strong, sub, sup, textarea, time, etc.
 - https://developer.mozilla.org/es/docs/Web/HTML/Inline_elements
- ✓ Los **elementos de bloque** definidos por HTML son:
 - address, blockquote, dialog, div, dl, dd, dt, form, h1, h2, h3, h4, h5, h6, header, hr, li, ol, ul, main, nav, p, pre, table, section, etc.
 - https://developer.mozilla.org/es/docs/Web/HTML/Block-level_elements
- ✓ **El comportamiento de un elemento (en línea o de bloque) se puede modificar en CSS.** Así que debes tener cuidado.

7. Modelo de Cajas

- ✓ ¿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 Cajas

- ✓ 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:



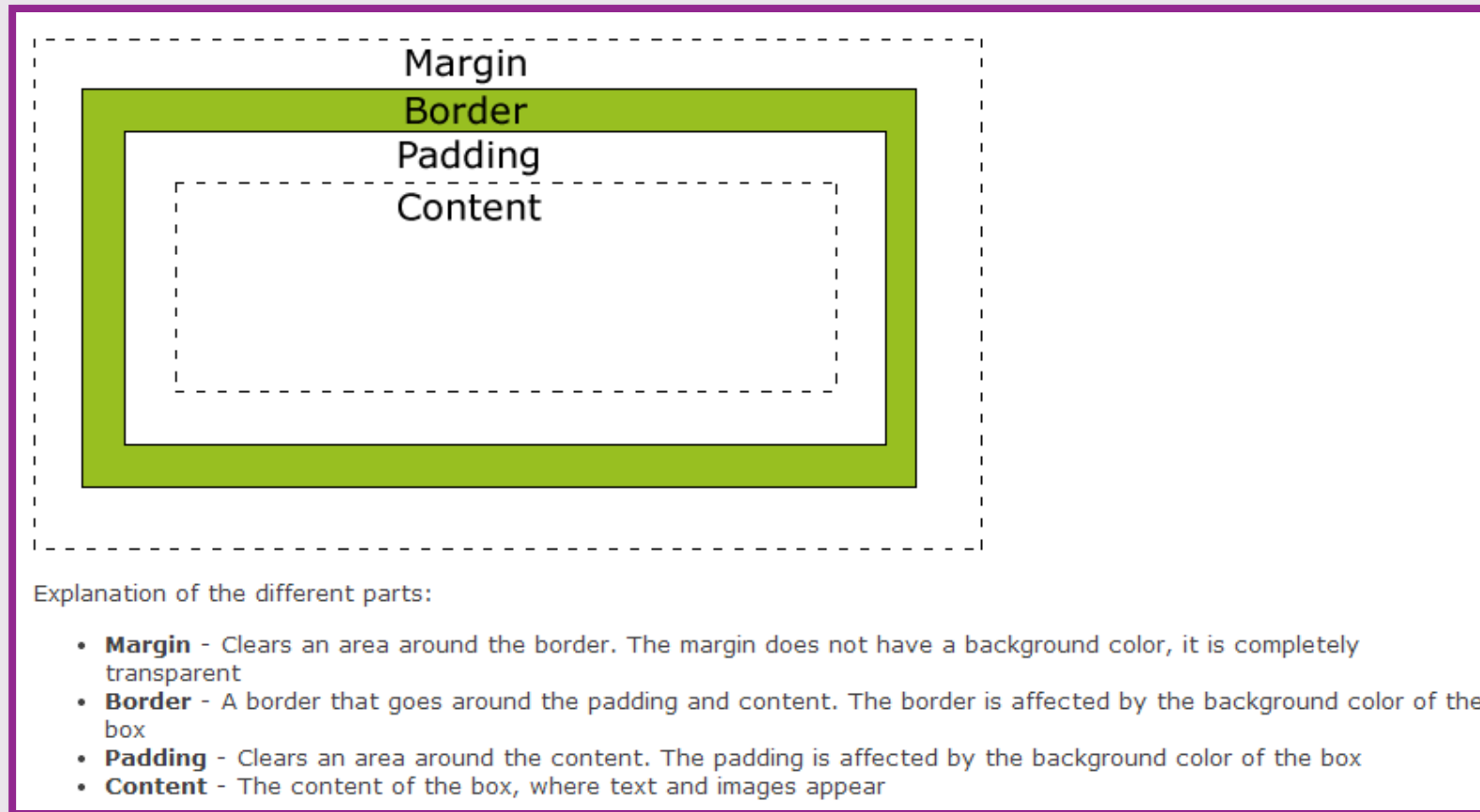
- ✓ Todas las cajas siguen lo que se conoce como **modelo de cajas** (*box model*).
- ✓ ¿Y en qué consiste dicho modelo?

7. Modelo de Cajas



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

7. Modelo de Cajas



7. Modelo de Cajas

The screenshot shows a web browser window displaying a CSS tutorial page. The page has a purple header with the text "HTML/CSS" and "JavaScript". Below the header, there are links to "Learn HTML", "Learn HTML5", "Learn CSS", and "Learn CSS3". The main content area features a green "HTML Tutorial" button and a blue "CSS Tutorial" button. The browser's developer tools are open, showing the "Elements" panel on the left and the "Styles" panel on the right. The "Elements" panel displays the DOM tree, with the following 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 shows the box model for the selected element. The diagram illustrates the box model with the following dimensions:

- margin: 10
- border: 0.909
- padding: 8
- content: 170 x 20

7. Modelo de Cajas

✓ 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.
- Igual que antes, podemos definir propiedades individuales: *top, right, bottom y left*
- https://www.w3schools.com/cssref/pr_margin.asp

7. Modelo de Cajas

✓ 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 Cajas

✓ 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 Cajas

✓ 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 Cajas

✓ Outline (Contorno)

- Se considera una variante de la propiedad "border" y por eso no se presentó esta propiedad en el modelo de cajas.
- 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 Cajas

✓ Outline (Contorno)

- **Outline - Shorthand property:** Nos permite definir en una sola línea todas las propiedades relacionadas con border exterior.
 - 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> <i>inherit</i>
----------------	--	--

7. Modelo de Cajas

✓ 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.
- Igual que antes, podemos definir propiedades individuales: *top*, *right*, *bottom* y *left*.
- https://www.w3schools.com/cssref/pr_padding.asp

7. Modelo de Cajas

✓ 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 Cajas

✓ 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 Cajas

✓ 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 Cajas

✓ CSS Width and Height

- CSS proporciona una serie de reglas que permiten definir las dimensiones del **contenido** ("content") una caja.



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

7. Modelo de Cajas

✓ 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
height	Sets the height of an element
max-height	Sets the maximum height of an element
max-width	Sets the maximum width of an element
min-height	Sets the minimum height of an element
min-width	Sets the minimum width of an element
width	Sets the width of an element

Value	Description
auto	The browser calculates the height. This is default
<i>length</i>	Defines the height in px, cm, etc. Read about length units
%	Defines the height in percent of the containing block
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. 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 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 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 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

✓ Propiedad *position*

Value	Description
static	Default value. Elements render in order, as they appear in the document flow
absolute	The element is positioned relative to its first positioned (not static) ancestor element
fixed	The element is positioned relative to the browser window
relative	The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position
sticky	<p>The element is positioned based on the user's scroll position</p> <p>A sticky element toggles between <i>relative</i> and <i>fixed</i>, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).</p> <p>Note: Not supported in IE/Edge 15 or earlier. Supported in Safari from version 6.1 with a -webkit- prefix.</p>
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>

8. Posicionamiento

✓ Propiedad position

- Además de indicar la posición que queremos, debemos **indiciar a qué distancia queremos posicionar el elemento**.

Property	Description
<u>bottom</u>	Sets the bottom margin edge for a positioned box
<u>clip</u>	Clips an absolutely positioned element
<u>left</u>	Sets the left margin edge for a positioned box
<u>position</u>	Specifies the type of positioning for an element
<u>right</u>	Sets the right margin edge for a positioned box
<u>top</u>	Sets the top margin edge for a positioned box

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


8. Posicionamiento

✓ Posicionamiento normal o estático

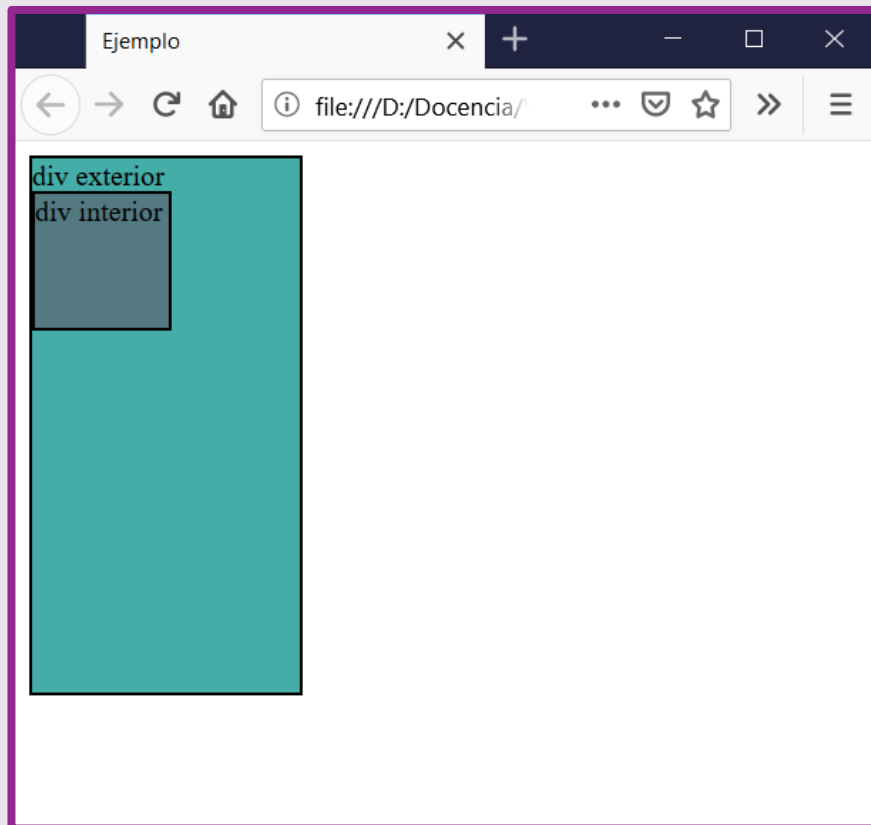
- Se trata del posicionamiento que utilizan los navegadores por defecto. Posiciona al elemento según el lugar donde fue definido en el HTML.

```
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

✓ 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

✓ Posicionamiento relativo

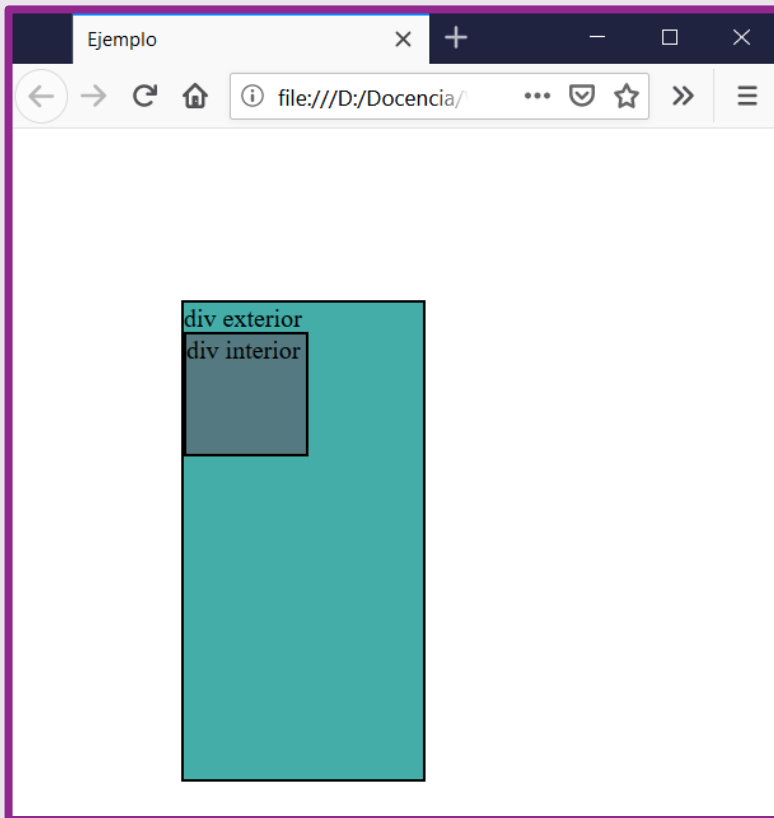
- Le indica al elemento que se desplace respecto del lugar donde hubiera estado posicionado de forma predeterminada como "static".

```
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

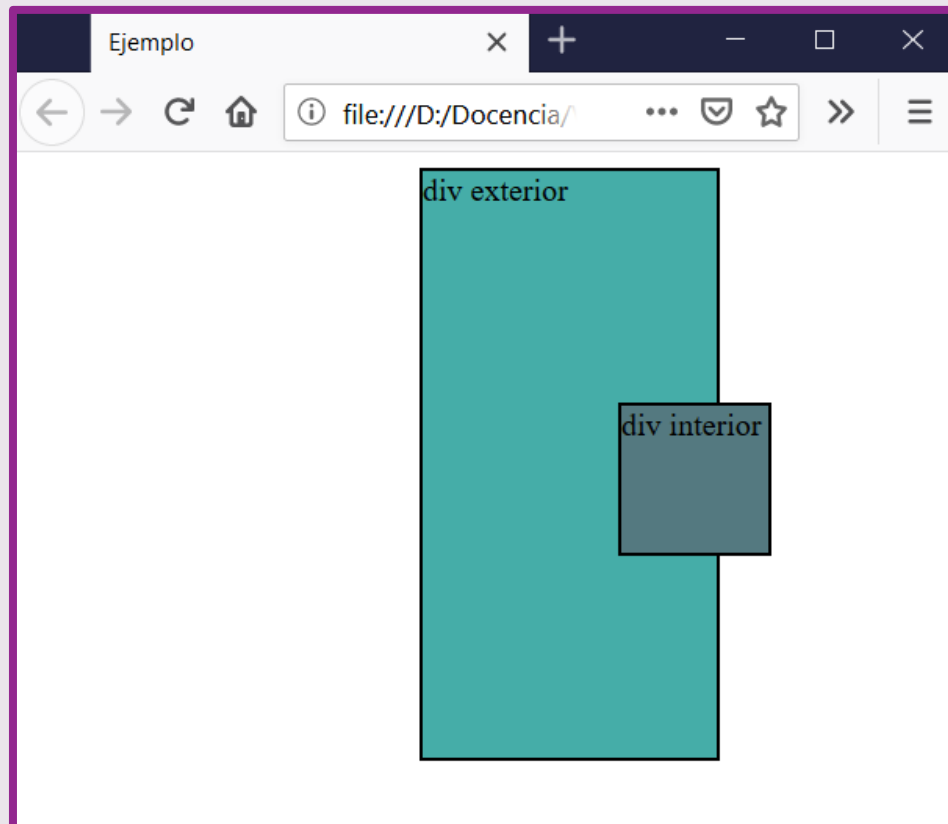
✓ Posicionamiento relativo



```
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

✓ 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

✓ Posicionamiento fijo

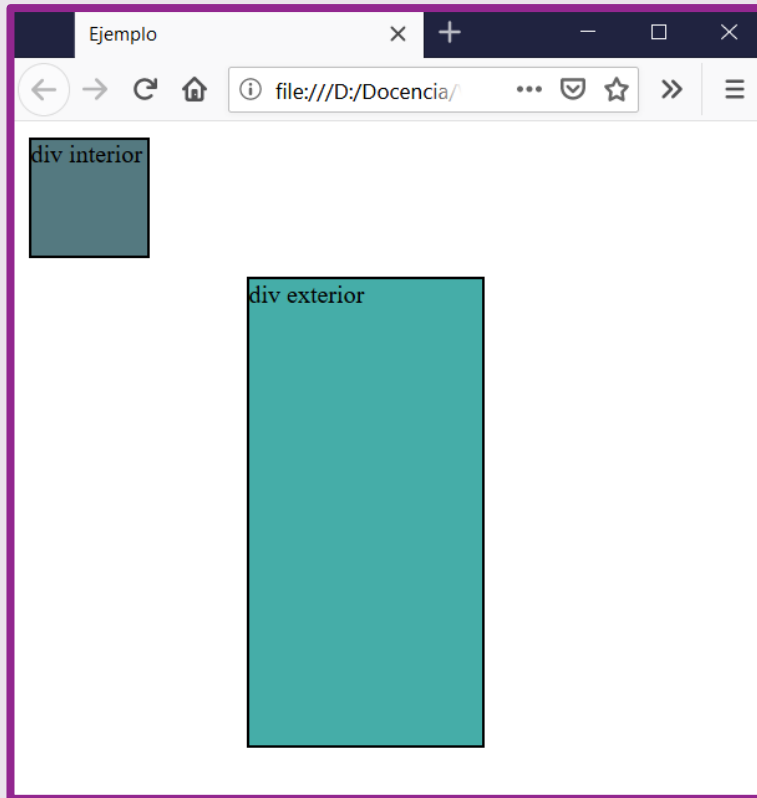
- Ancla un elemento a la ventana del navegador. Si desplazas la pantalla hacia arriba y hacia abajo, el elemento fijo se mantiene en su lugar. Su posición final será siempre fija.

```
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

✓ 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

✓ Posicionamiento absoluto

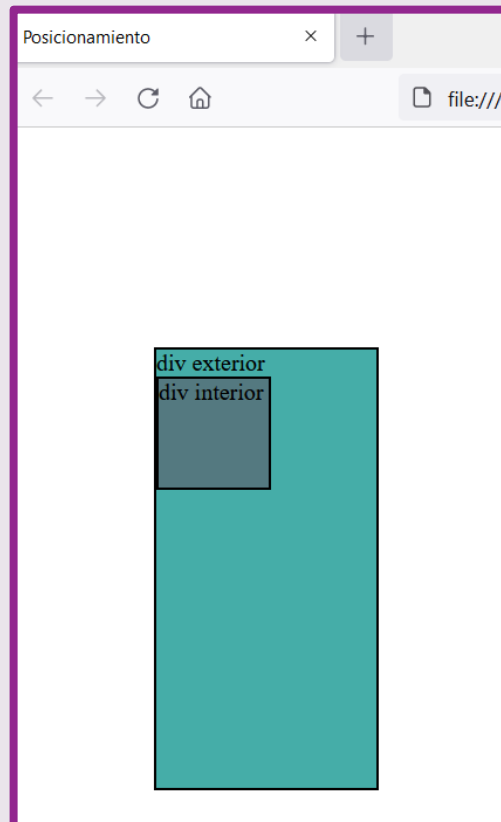
- La posición de una caja se establece de forma absoluta y el elemento es colocado exactamente donde nosotros queremos.

```
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

✓ 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

✓ 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 }
```

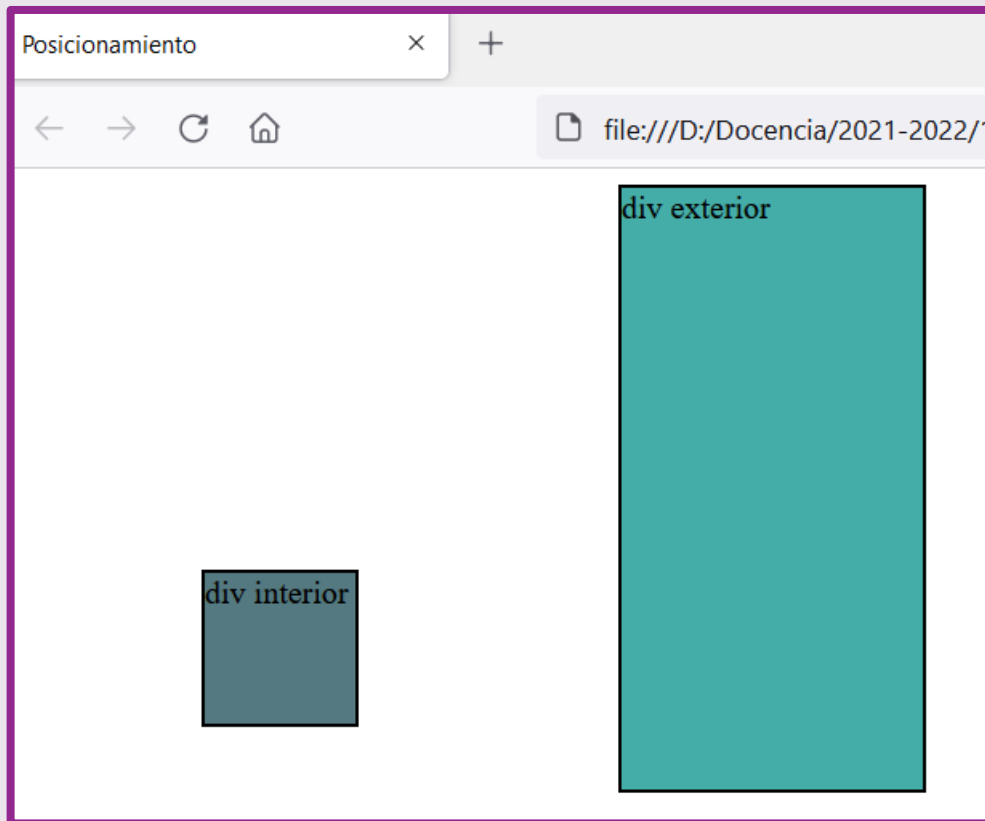
8. Posicionamiento

✓ 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

✓ 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     /* ¡Uso margin-left! No confundir con left */
21     margin-left: 300px;
22 }
```

8. Posicionamiento

✓ Posicionamiento flotante: float y clear

- El 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.

Value	Description
none	The element does not float, (will be displayed just where it occurs in the text). This is default
left	The element floats to the left of its container
right	The element floats the right of its container
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

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

8. Posicionamiento

✓ Posicionamiento flotante: float y clear

- 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

✓ Posicionamiento flotante: float y clear

```
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

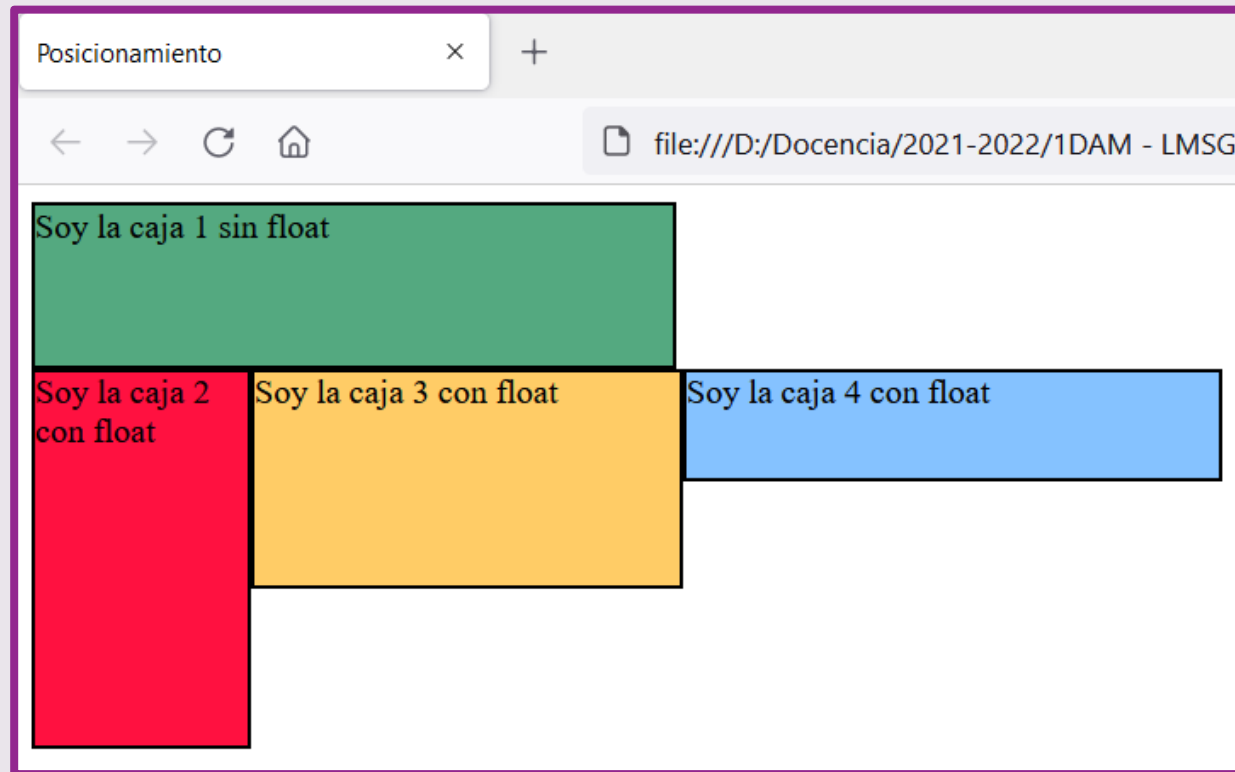
✓ Posicionamiento flotante: float y clear



```
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

✓ Posicionamiento flotante: float y clear

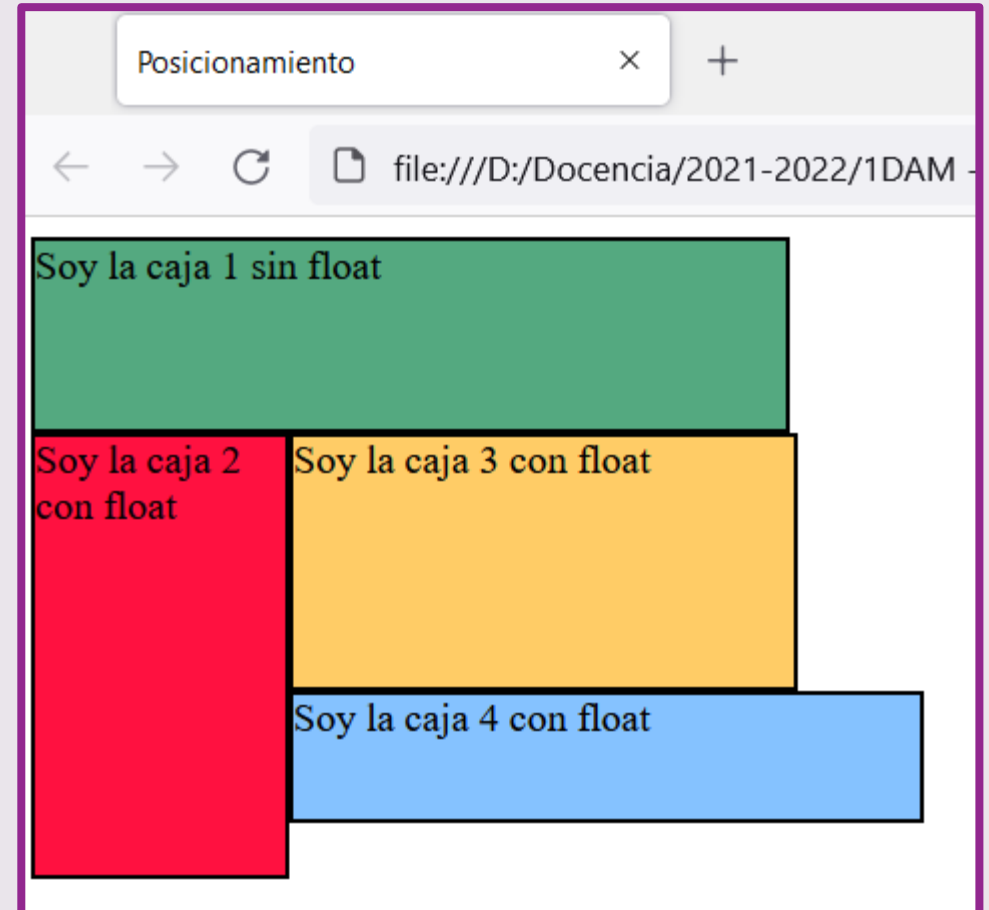
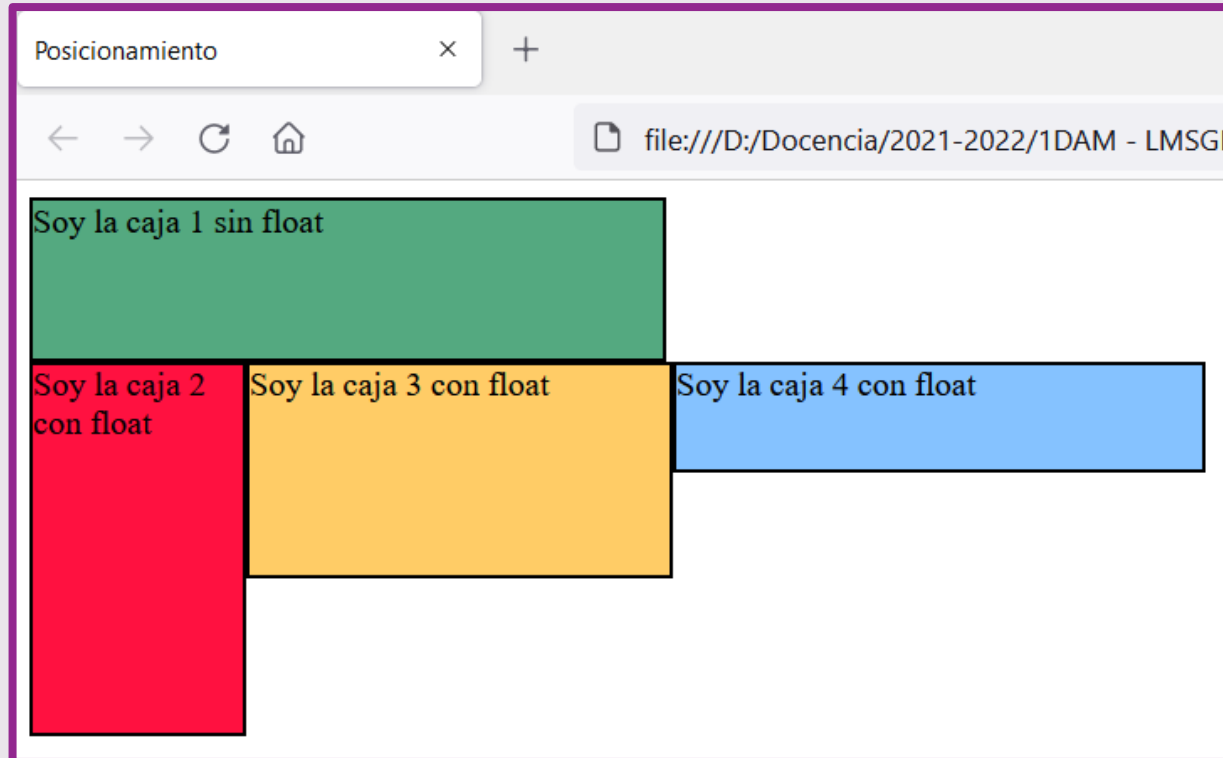


```
5 #caja1 {
6     height: 75px;
7     width: 300px;
8     background-color: #54A980;
9 }
10 #caja2 {
11     height: 175px;
12     width: 100px;
13     background-color: #FF1140;
14     float: left;
15 }
16 #caja3 {
17     height: 100px;
18     width: 200px;
19     background-color: #FFCC66;
20   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

8. Posicionamiento

✓ Posicionamiento flotante: float y clear

- ¿Qué ocurre si redimensiono la ventana?



8. Posicionamiento

✓ Posicionamiento flotante: float y 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.

Value	Description
none	Default. The element is not pushed below left or right floated elements
left	The element is pushed below left floated elements
right	The element is pushed below right floated elements
both	The element is pushed below both left and right floated elements
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

8. Posicionamiento

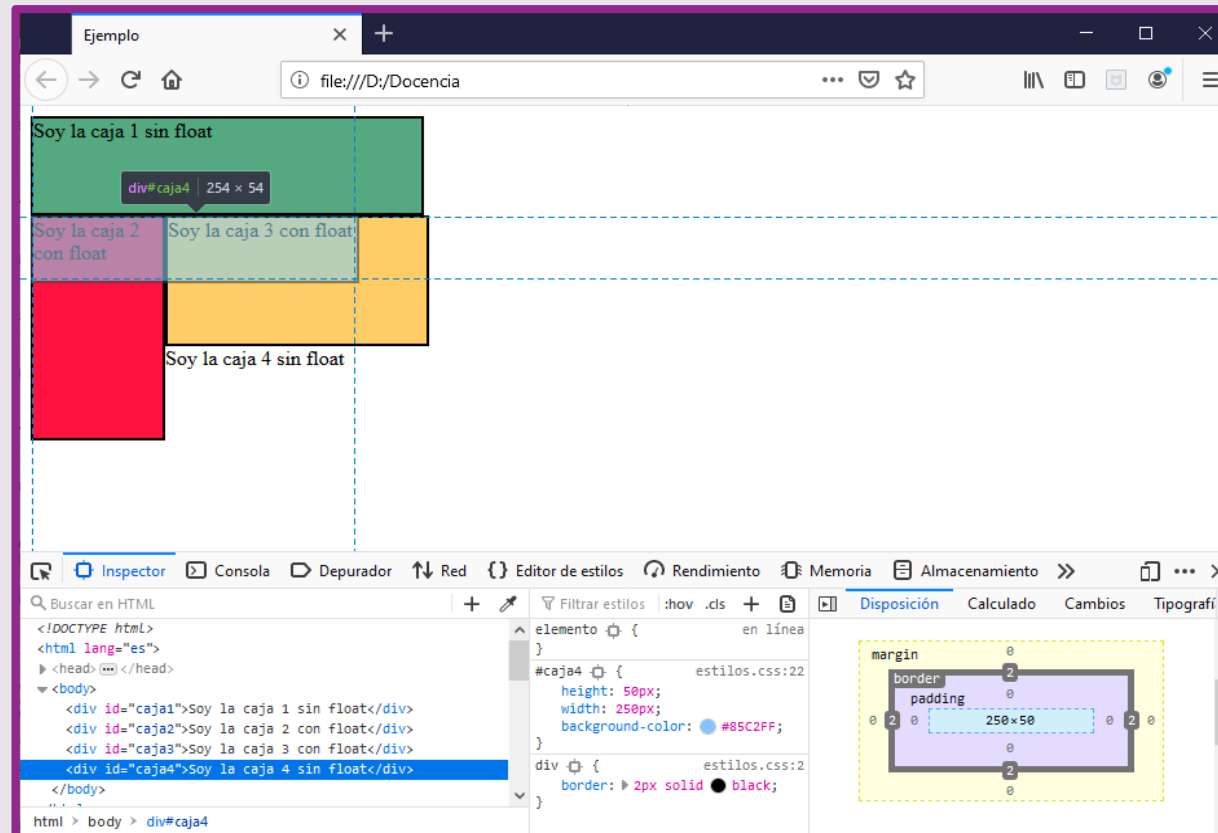
✓ Posicionamiento flotante: float y clear



```
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

✓ Posicionamiento flotante: float y clear



8. Posicionamiento

✓ Posicionamiento flotante: float y clear



```
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. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. 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.
 - **overflow:** permite controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos (scroll).
 - **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.

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 **visibility** son:

Value	Description
visible	Default value. The element is visible
hidden	The element is hidden (but still takes up space)
collapse	Only for table rows (<tr>), row groups (<tbody>), columns (<col>), column groups (<colgroup>). This value 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"
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

- ✓ 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

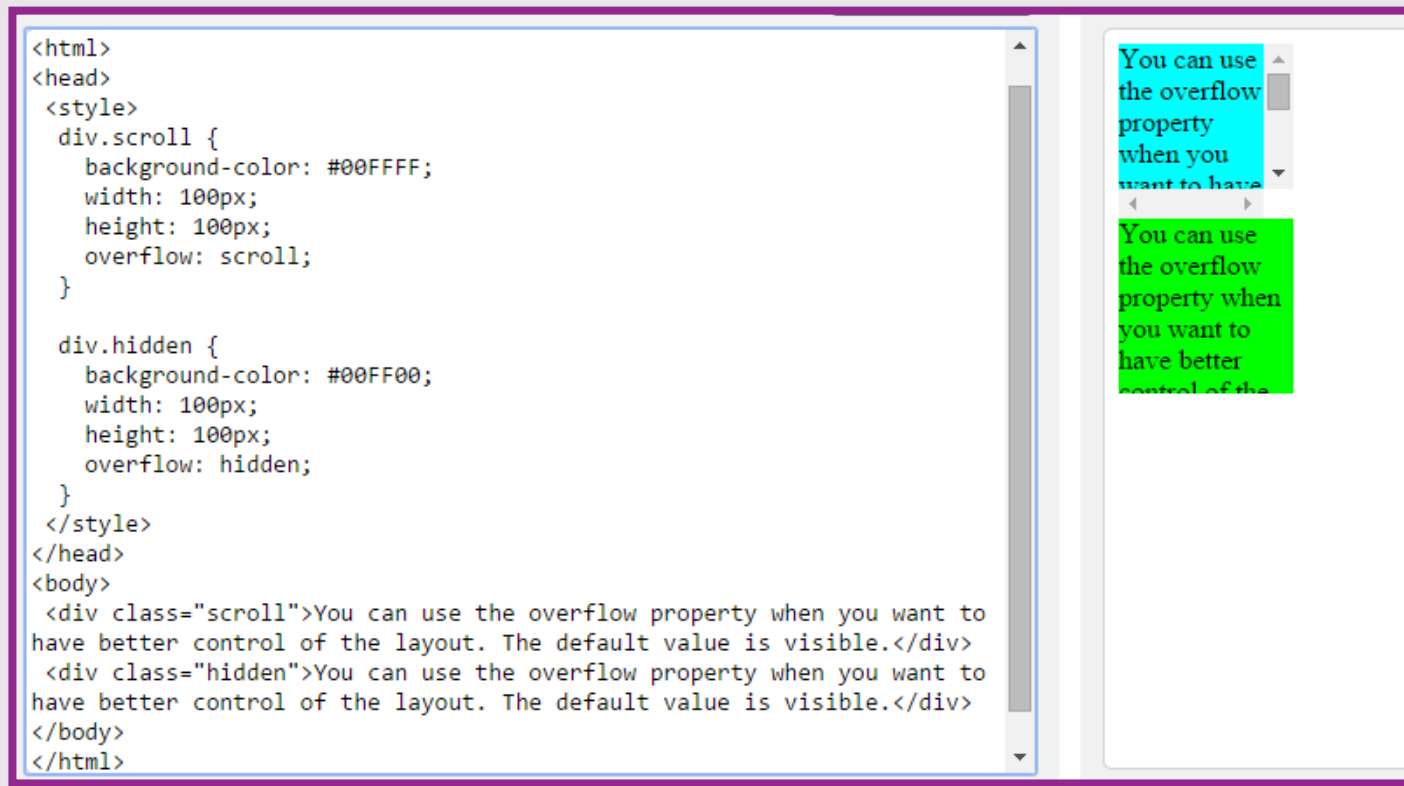
- ✓ Los valores que puede tomar la propiedad **overflow** son:

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. <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_overflow.asp

9. Visualización

- ✓ Ejemplo de *overflow*:



9. Visualización

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

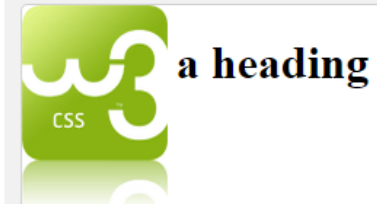
Value	Description
auto	Sets the stack order equal to its parents. This is default
<i>number</i>	Sets the stack order of the element. Negative numbers are allowed
initial	Sets this property to its default value. <u>Read about <i>initial</i></u>
inherit	Inherits this property from its parent element. 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

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

9. Visualización

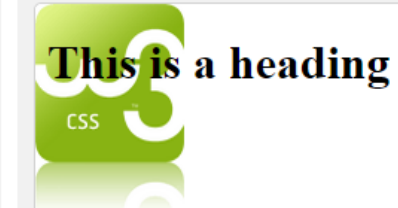
✓ Ejemplo sin “z-index”:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img {
      position: absolute;
      left: 0px;
      top: 0px;
    }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
  
</body>
</html>
```



✓ Ejemplo con “z-index”:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img {
      position: absolute;
      left: 0px;
      top: 0px;
      z-index: -1;
    }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
  
</body>
</html>
```

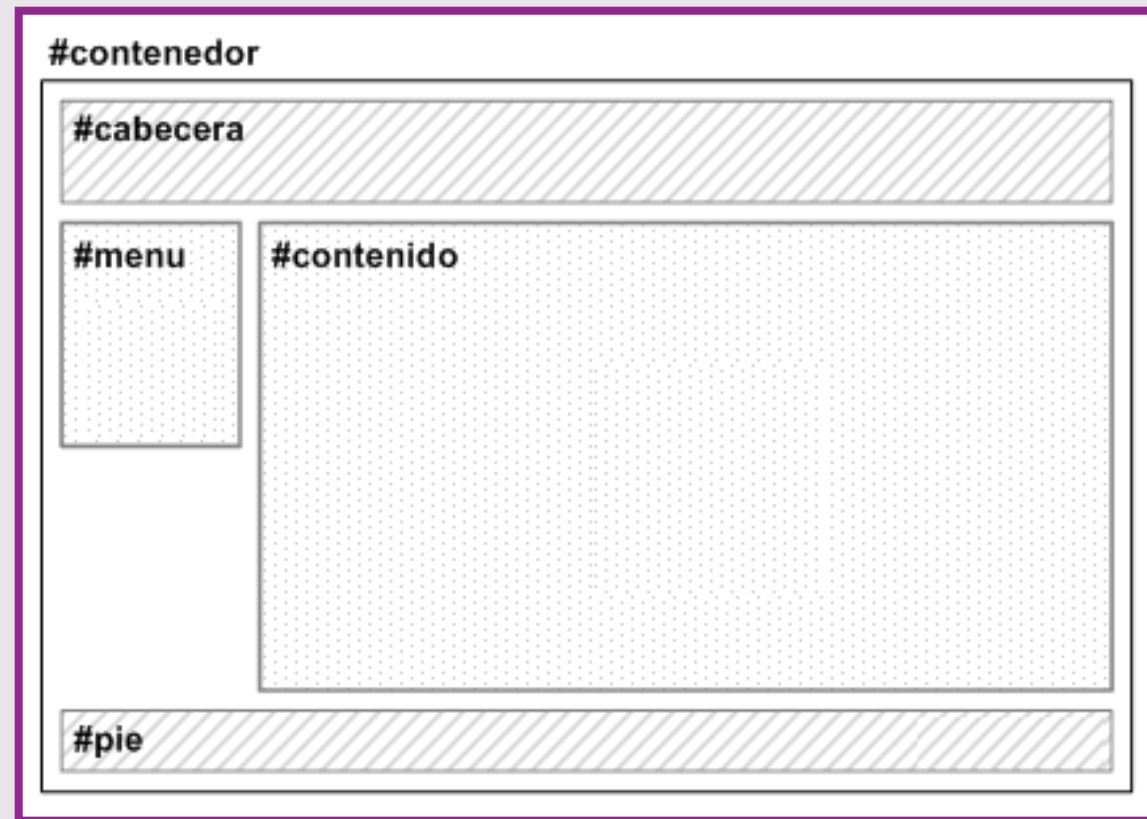


ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

10. Definición de las cajas contenedoras

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



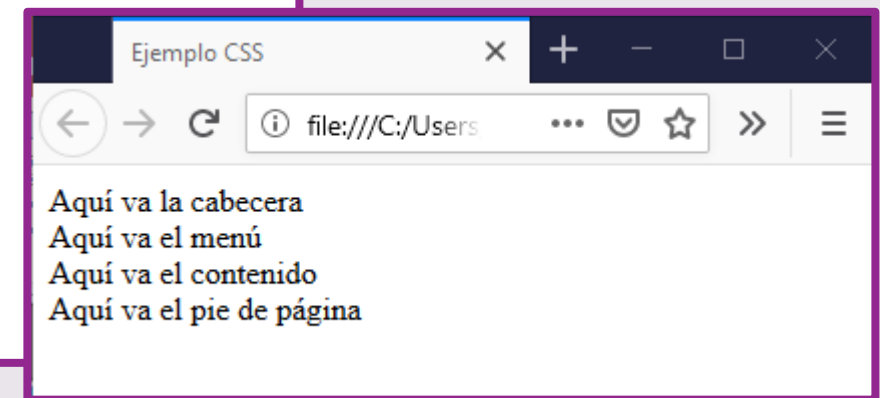
10. Definición de las cajas contenedoras

- ✓ 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>
```

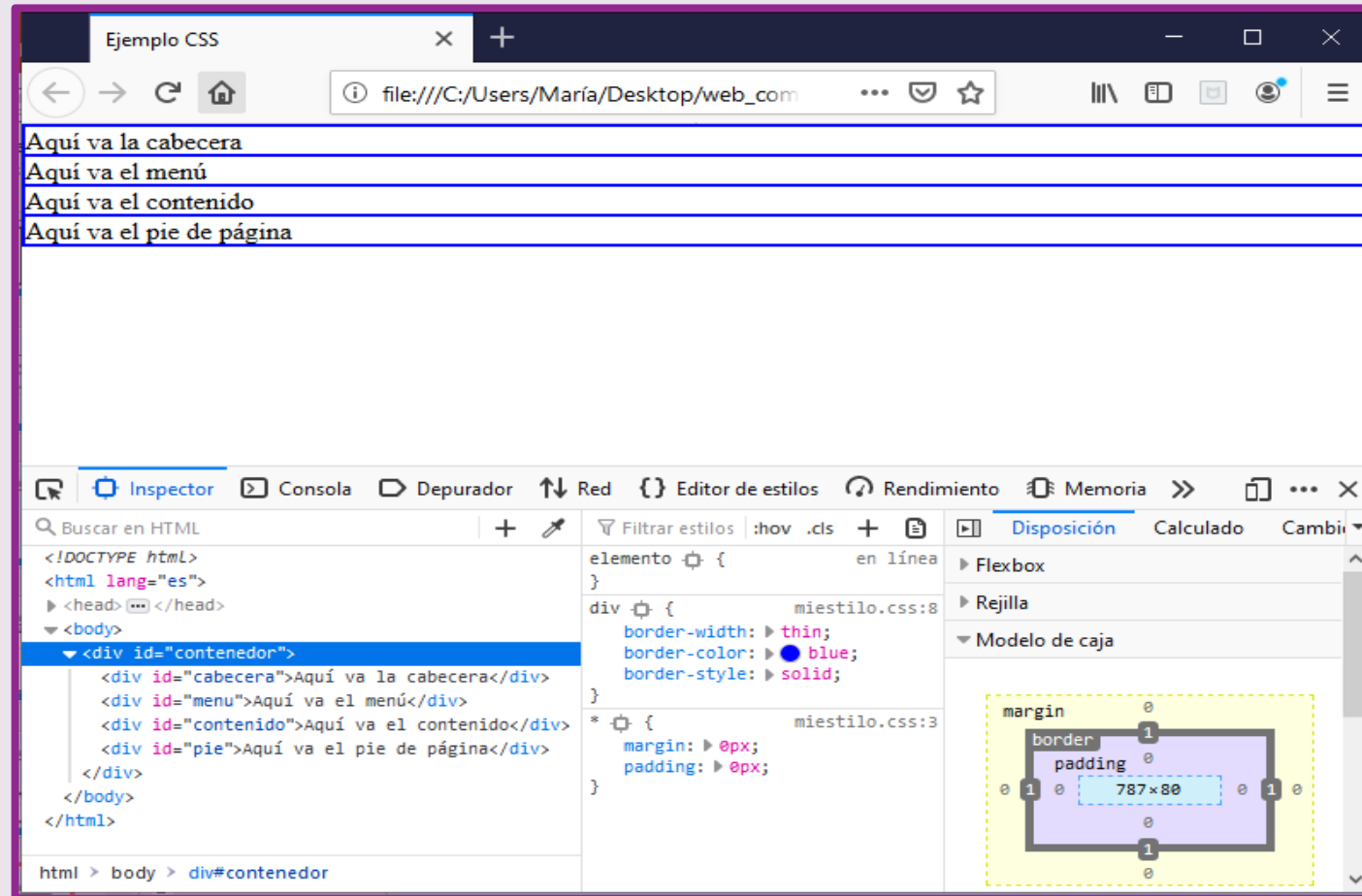


10. Definición de las cajas contenedoras

- ✓ 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;
```

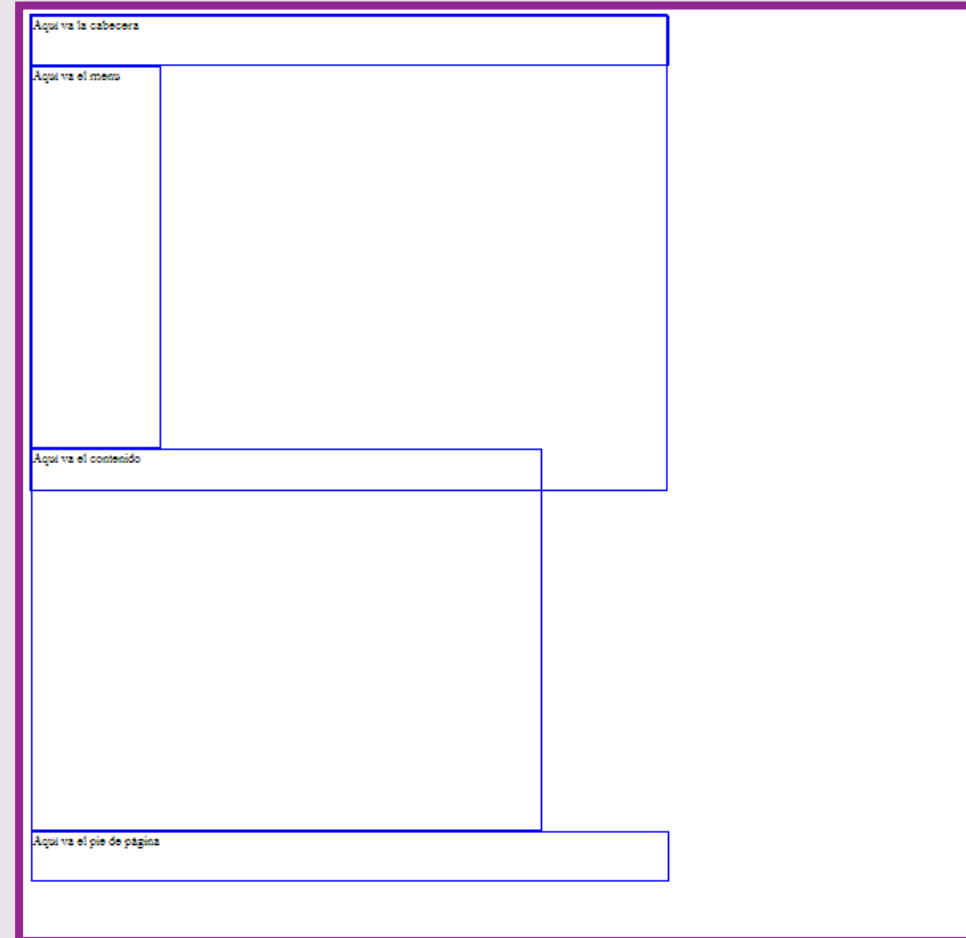
10. Definición de las cajas contenedoras



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

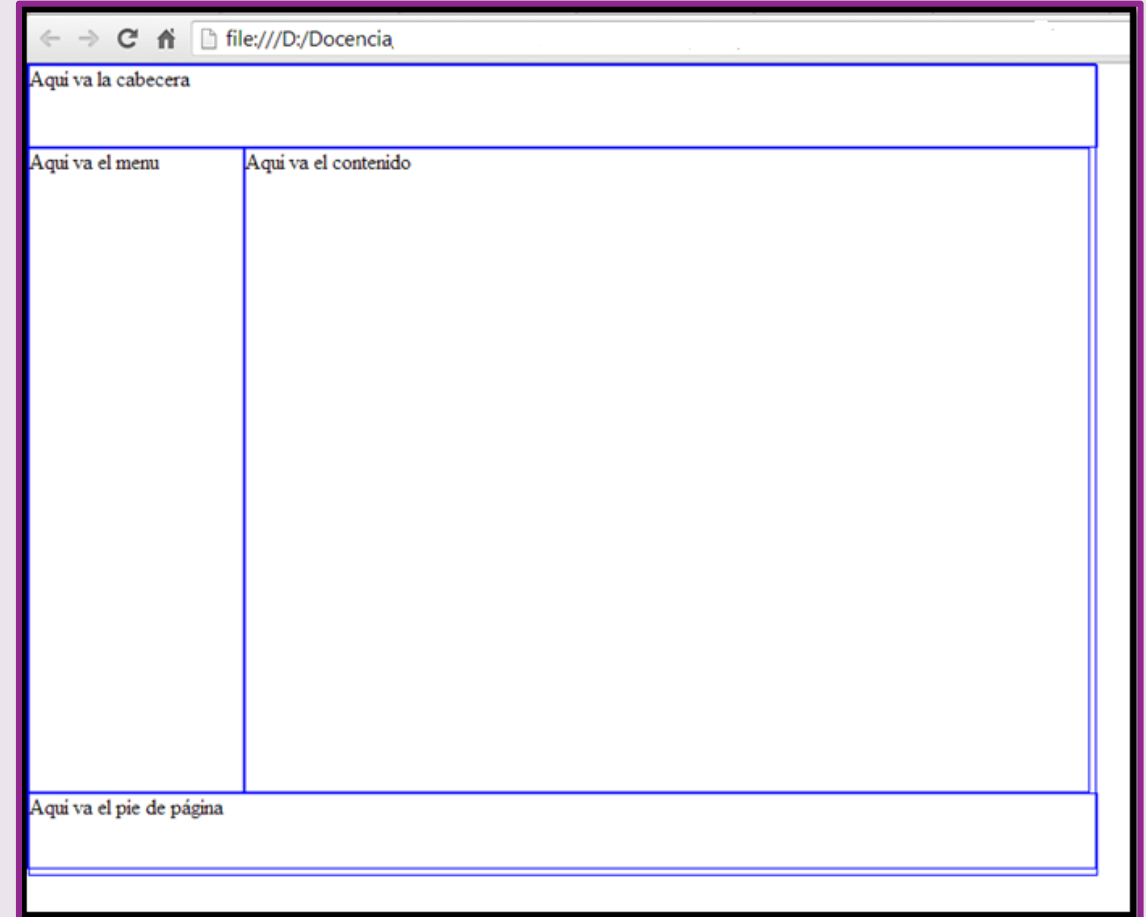
1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
8. Posicionamiento en CSS
9. Visualización
10. Definición de las cajas contenedoras
- 11. Posicionamiento de las cajas contenedoras**
12. Uso de medidas fijas
13. Colores e imágenes de fondo
14. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. 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 }
```



ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

12. Uso de medidas fijas

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

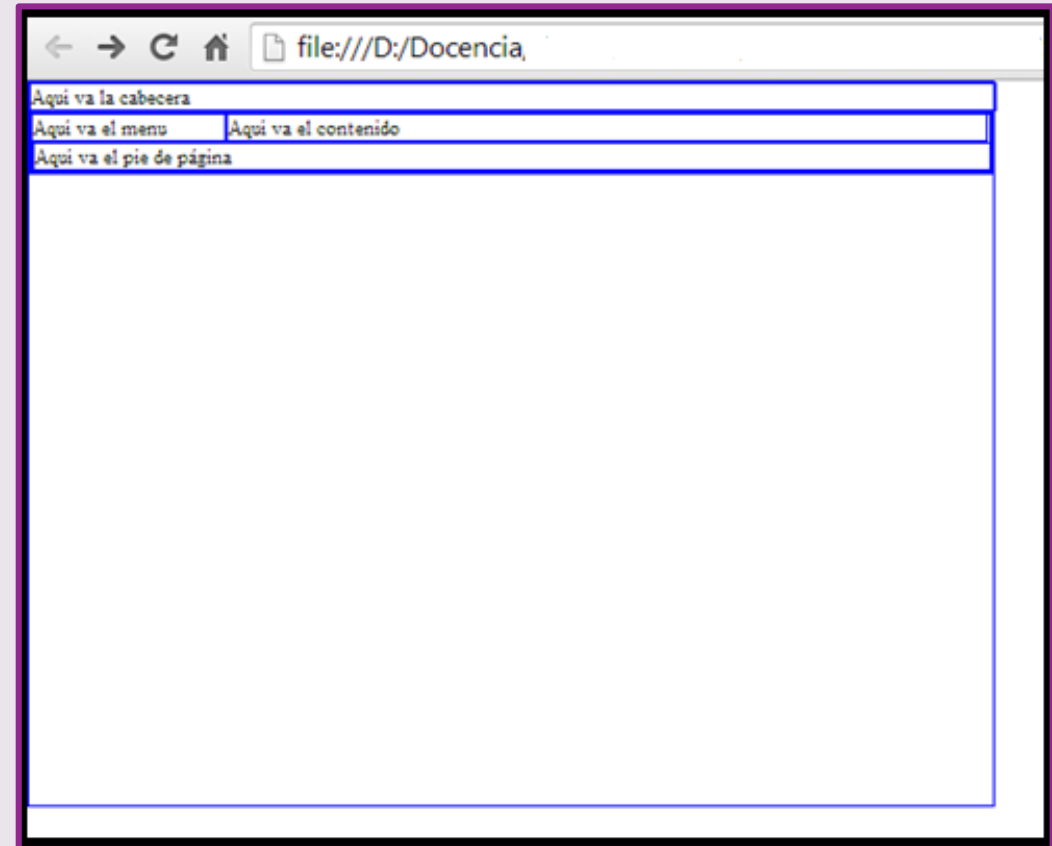
Aquí va la cabecera	
Aquí va el menu	Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo. Soy un párrafo.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

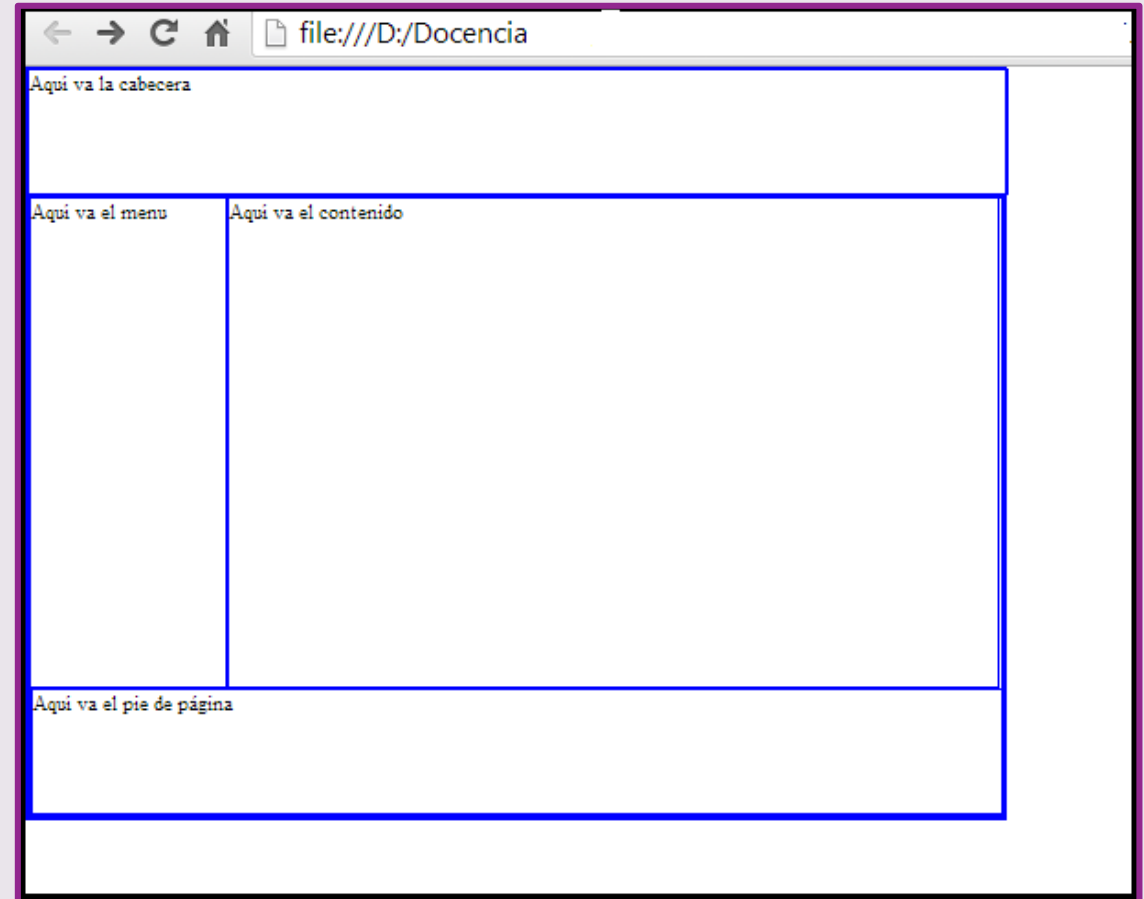
✓ Nuevo problema: →



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. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

13. Colores e imágenes de fondo

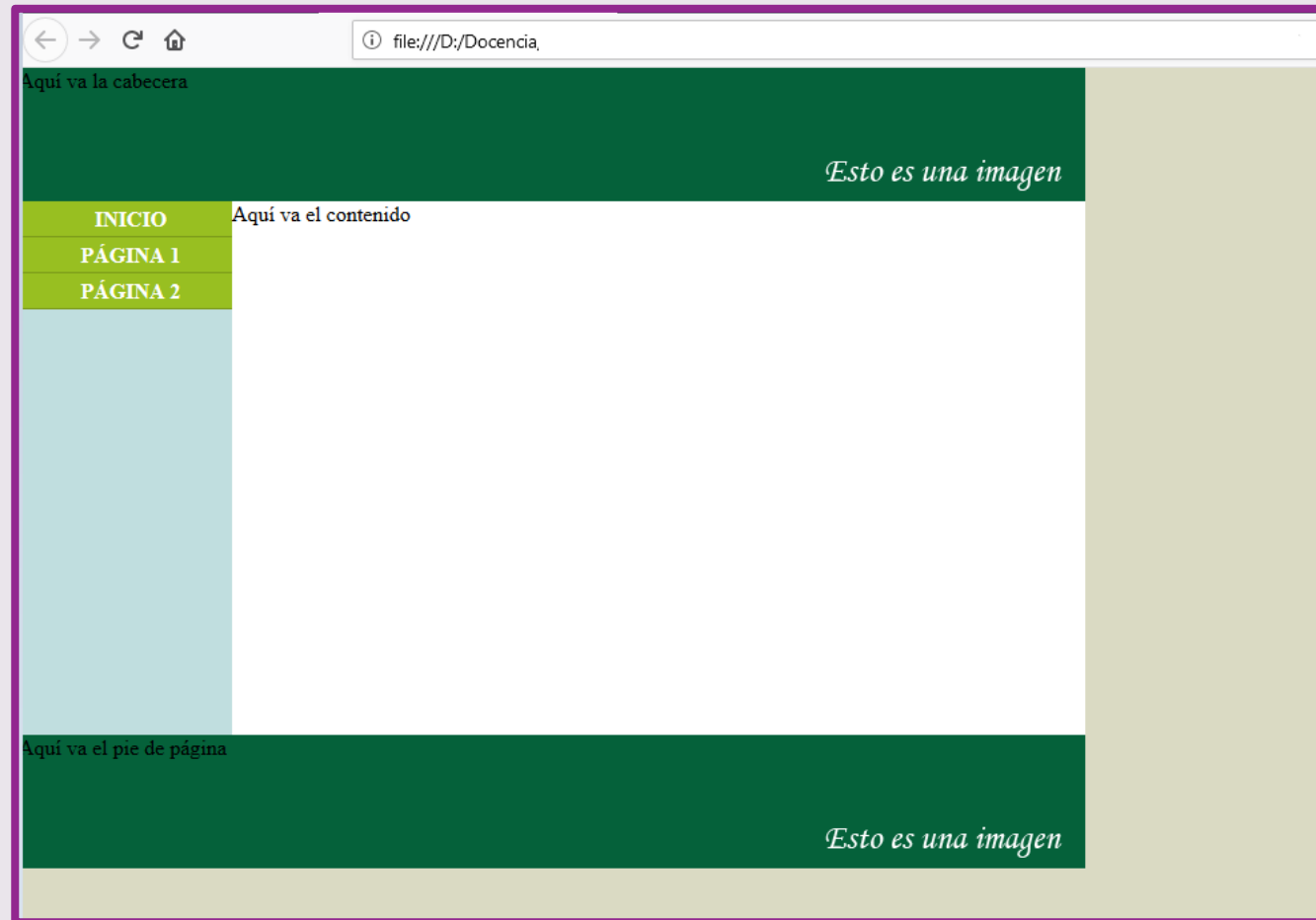
- ✓ En primer lugar voy a eliminar el borde azul de las cajas porque al darle colores e imágenes de fondo ya no es necesario remarcar las cajas.
- ✓ La cabecera y el pie de página serán dos imágenes ".png" de 800x100px.
- ✓ El menú y el contenido tendrá definido un color de fondo.
- ✓ Ya también aprovechamos 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. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. **CSS3**
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

14. CSS3

- ✓ **Objetivo inicial de CSS:** separar el contenido de la forma.
 - ✓ **CSS 3:** Incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas y mayor número de efectos visuales, sin tener que recurrir a trucos, que a menudo complicaban el código de las web.
 - ✓ Código más simple para muchas tareas.
 - ✓ Mayores opciones de gráficas.
- Evolución de CSS**



The diagram illustrates the evolution of CSS through three stages, each represented by a car. The first stage, CSS 1, is represented by a green Volkswagen Beetle. The second stage, CSS 2.1, is represented by a teal classic convertible. The third stage, CSS 3, is represented by a red sports car. The title 'Evolución de CSS' is written in red above the cars.
- ✓ **Desventaja:** No tiene compatibilidad 100% con ningún navegador. Para determinadas propiedades debemos usar los prefijos -moz- y -webkit- para que funcionen en navegadores basados en motores Gecko y WebKit, como Firefox, Safari y Google Chrome.

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
- 15. CSS Flexbox**
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

15. CSS Flexbox

- ✓ El Módulo de Caja Flexible, comúnmente llamado *flexbox*, fue diseñado como un modelo unidimensional de layout, y como un método que pueda ayudar a distribuir el espacio entre los ítems de una interfaz y mejorar las capacidades de alineación.
- ✓ Flexbox provee herramientas para crear layouts complejos y flexibles que, históricamente, han sido difíciles de implementar en CSS.
- ✓ Permite colocar los elementos de una página para que se comporten de forma predecible cuando el diseño de la página debe acomodarse a diferentes tamaños de pantalla y diferentes dispositivos. Para muchas aplicaciones, el modelo "caja flexible" produce una mejora sobre el modelo "bloque" porque no utiliza la propiedad float, ni hace que los márgenes del contenedor flexible interfieran con los márgenes de sus contenidos.
- ✓ https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox

15. CSS Flexbox

- ✓ La principal idea que hay detrás de la maquetación **FLEX** es que vamos a tener un elemento, es decir, una etiqueta que va a poder *controlar* las propiedades de los elementos que contiene.
- ✓ Por lo tanto, en esta situación, vamos a poder distinguir dos tipos de elementos:
 - El **contenedor flex** que tendrá asignada la propiedad CSS ***display:flex*** y que va a controlar ciertas propiedades de los elementos que contiene. Añadiendo esta propiedad podremos ver que de manera automática los elementos ajustan su anchura a su contenido y flotan a la izquierda.
 - Los **elementos flexibles** que son los elementos que están dentro del contenedor y cuyas propiedades modificaremos.
- ✓ Podremos modificar propiedades CSS como:
 - La altura y anchura de los elementos flexibles.
 - El orden en el que se nos van a presentar.
 - La alineación vertical y horizontal.
 - La distribución de los elementos flexibles a lo largo del contenedor.
- ✓ Es decir, vamos a poder controlar propiedades que usamos para maquetar y además, vamos a poder maquetar de manera mucho más ágil a lo que lo hacemos con las técnicas tradicionales de maquetado.

15. CSS Flexbox

✓ Propiedades en el contenedor Flex

■ Dirección de los elementos flexibles:

- Propiedad **flex-direction**: Puede tomar los siguientes valores:
 - **row**: es la opción por defecto y ajustará los elementos flexibles de izquierda a derecha.
 - **row-reverse**: igual que la anterior pero de derecha a izquierda.
 - **column**: ajustará los elementos flexibles en columna, de arriba a abajo.
 - **column-reverse**: igual que la de arriba pero de abajo a arriba.

15. CSS Flexbox

✓ Propiedades en el contenedor Flex

■ Ajuste de los elementos flexibles:

- Por defecto y sin indicar ninguna anchura, los elementos flexibles adecuan su tamaño a su contenido y se colocan todos a la izquierda permaneciendo siempre así aunque el ancho de la pantalla sea muy pequeño. Esto puede provocar desajustes en pantallas muy pequeñas.
- Propiedad **flex-wrap**: Puede tomar los siguientes valores:
 - **no-wrap**: es el valor por defecto y fuerza para que siempre los elementos estén en la misma línea aunque esto suponga que se salgan del contenedor (les haya dado o no les haya dado anchura).
 - **wrap**: provoca un salto de línea si la anchura de los elementos (fijada por nosotros o por el contenedor) es superior a la del contenedor.
 - **wrap-reverse**: lo mismo que arriba pero de abajo a arriba.
- Las dos propiedades, **flex-direction** y **flex-wrap** podemos juntarlas en la propiedad **flex-flow** con dos partes como por ejemplo:

```
flex-flow: row wrap;
```

15. CSS Flexbox

✓ Propiedades en el contenedor Flex

■ Alineación horizontal de los elementos flexibles:

- Podemos alinear horizontalmente los elementos flexibles, tengan o no tengan establecida una anchura.
- Propiedad **justify-content**: Puede tomar los siguientes valores:
 - **flex-start**: Los elementos flexibles se sitúan al principio.
 - **flex-end**: Los elementos flexibles se sitúan al final.
 - **center**: Los elementos se centran horizontalmente.
 - **space-between**: Distribuye el espacio restante entre los elementos pero el primero y el último están en los bordes.
 - **space-around**: Distribuye el espacio restante entre los elementos pero no tiene en cuenta la distancia a los bordes.
 - **space-evenly**: Distribuye el espacio restante entre los elementos y tiene en cuenta la distancia a los bordes.

15. CSS Flexbox

✓ Propiedades en el contenedor Flex

■ Alineación vertical de los elementos flexibles:

- Propiedad **align-items**: Puede tomar los siguientes valores:
 - **flex-start**: Los elementos se ponen junto al borde superior.
 - **flex-end**: Los elementos se ponen junto al borde inferior.
 - **center**: Los elementos flexibles se centran verticalmente.
 - **stretch**: Los elementos crecen en altura para ocupar toda la altura del contenedor flexible. No deben tener altura fija establecida.
 - **baseline**: Los elementos se alinean en relación con la primera línea de texto que posean los elementos flexibles.

15. CSS Flexbox

✓ Propiedades en el contenedor Flex

- **Alineación vertical – Wrap (cuando tengo varias líneas):**
 - Si hemos usado la propiedad ***flex-wrap:wrap*** y resulta que tenemos varias líneas de elementos flexibles también se pueden alinear usando la propiedad CSS **align-content**. Los valores que puede tomar son análogos a los vistos anteriormente:
 - **flex-start**
 - **flex-end**
 - **center**
 - **stretch**
 - **space-between**
 - **space-around**

15. CSS Flexbox

✓ Elementos flexibles:

- Además de modificando las propiedades del contenedor FLEX, podemos actuar sobre los elementos flexibles modificando otra serie de propiedades relacionadas.
- **Orden de los elementos flexibles:**
 - Los elementos flexibles se muestran dentro del contenedor flex en el mismo orden en que están escritos en nuestro código HTML.
 - Si queremos modificar esto debemos añadir la propiedad CSS ***order*** a los **elementos** cuyo orden queremos modificar.
 - Por defecto este valor es 0 y se mostrarán primeros aquellos elementos que tenga un mayor orden. En caso de empate se muestra antes el que primero estuviera en el código.

15. CSS Flexbox

✓ Elementos flexibles:

■ Tamaño de los elementos flexibles:

- **flex-grow:** que sirve para indicar, mediante un número, el factor de crecimiento de un elemento flexible cuando se distribuye entre los elementos flexibles el espacio restante. Por defecto es 1 pero si queremos que un elemento participe en el reparto es necesario añadir esta propiedad.
- **flex-shrink:** que sirve para indicar, mediante un número el factor de contracción de un elemento flexible cuando el tamaño de todos sobrepasa el tamaño del contenedor. Por defecto es 1 pero si quiero que un elemento participe en la contracción es necesario añadir esta propiedad.
- **flex-basis:** que sirve para indicar el tamaño de un elemento antes de que el espacio restante (negativo positivo) se distribuya. Por defecto el valor de esta propiedad es *auto* y hace que la anchura del elemento flexible se ajusta a su contenido.

15. CSS Flexbox

✓ Elementos flexibles:

- En ocasiones podemos necesitar que un elemento flexible tenga una **alineación vertical diferente al resto**. En este caso, en el elemento para el que quiero una alineación diferente, debo añadir la propiedad CSS ***align-self*** que puede tomar los mismo valores (y con el mismo significado) que la propiedad ***align-items***.
 - **flex-start**
 - **flex-end**
 - **center**
 - **stretch** (no debe tener altura establecida)
 - **Baseline**
- **NOTA:** Los elementos flex no hace caso a la propiedad *vertical-align*.

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. **CSS Grid Layout**
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. Bibliografía

16. CSS Grid Layout

- ✓ La idea principal que hay detrás de la maquetación **GRID** es que vamos a tener un elemento, es decir, un etiqueta que nos va a permitir *controlar* propiedades de los elementos que contiene y establecer **estructuras complejas** para distribuirlos.
- ✓ https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Grids
- ✓ Por lo tanto, en esta situación, vamos a poder distinguir dos tipos de elementos:
 - El **contenedor GRID** que tendrá asignada la propiedad CSS *display:grid* (si queremos que nuestra rejilla sea un elemento de bloque) o *display:inline-grid* (si queremos que nuestra rejilla sea un elemento en línea). Todos los elementos que contiene pasan a convertirse de manera automática en elementos del GRID cuya colocación y propiedades podremos empezar a modificar desde el contenedor.
 - Los **elementos GRID** que son los elementos que están dentro del contenedor, elementos que distribuiremos y cuyas propiedades modificaremos.
- ✓ Desde el contenedor podemos modificar:
 - La estructura en filas y columnas y la separación entre ellas.
 - Definir áreas del GRID con nombre.
 - La alineación horizontal y vertical de los elementos del GRID y del propio GRID dentro del elemento que lo contiene.

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

- **Definición de la estructura del GRID:**
 - **grid-template-columns:** Número de columnas del grid y tamaño de las mismas.
 - **grid-template-rows:** Número de filas del grid y tamaño de las mismas.
 - **grid-row-gap:** Para establecer la separación entre las diferentes filas.
 - **grid-column-gap:** Para establecer la separación entre las diferentes columnas.

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

- Definición de la estructura del GRID:

- Ejemplo con columnas:

```
/* Tres columnas que se reparten el 100% del contenedor */  
grid-template-columns: 30% 50% 20%;  
  
/* Cuatro columnas: tres de tamaño fijo de 150px y la otra ocupa  
el espacio libre restante */  
grid-template-columns: 150px auto 150px 150px;  
  
/* Cuatro columnas: todas con un tamaño igual */  
grid-template-columns: auto auto auto auto auto;  
  
/* Tres columnas cada una con un nombre (entre []).  
   Dos con tamaño fijo y la otra ocupando el espacio restante */  
grid-template-columns : [id] 100px [nombre] 300px [apellidos] auto;
```

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

- Definición de la estructura del GRID:

- Ejemplo con filas:

```
/* Tres filas que se reparten toda la altura del contenedor (la que sea) */  
grid-template-rows: 30% 50% 20%;  
  
/* Cuatro filas: tres de altura fijo de 150px y la otra ocupará  
el espacio libre restante hasta llenar el contenedor en altura*/  
grid-template-rows: 150px auto 150px 150px;  
  
/* Cuatro filas que se reparten de manera equitativa el alto del contenedor */  
grid-template-rows: auto auto auto auto auto;  
  
/* Tres filas: todas con nombre, entre [].  
   Dos con tamaño fijo y la otra ocupará todo el alto libre */  
grid-template-rows: [uno] 100px [dos] 300px [tres] auto;
```

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

■ Definición de la estructura del GRID:

- Se pueden repetir valores y usar la unidad *fr* para establecer ratios para que los elementos se repartan el espacio restante.

```
/* Cuatro columnas: Tres de 20% con el nombre col-start; y la última  
ocupará el resto del espacio libre */  
grid-template-column: repeat(3, 20% [col-start]) auto;  
  
/* Cuatro columnas: Una de tamaño fijo, y las demás se reparten el  
espacio libre en 5 partes: 2+1+2 */  
grid-template-column: 2fr 200px 1fr 2fr;
```

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

■ Definición de la estructura del GRID:

○ Ejemplo:

```
<div class="container">
  <div>Primero</div>
  <div>Segundo</div>
  <div>Tercero</div>
  <div>Cuarto</div>
  <div>Quinto</div>
  <div>Sexto</div>
  <div>Séptimo</div>
</div>
```

```
@charset "UTF-8";

.container {
  background-color: #0F0;
  display: grid;
  grid-column-gap: 10px;
  grid-row-gap: 20px;
  grid-template-columns: 20% auto 20%;
  grid-template-rows: repeat(3, 100px);
  margin: 20px auto;
  padding: 1em;
  width: 80%;
}

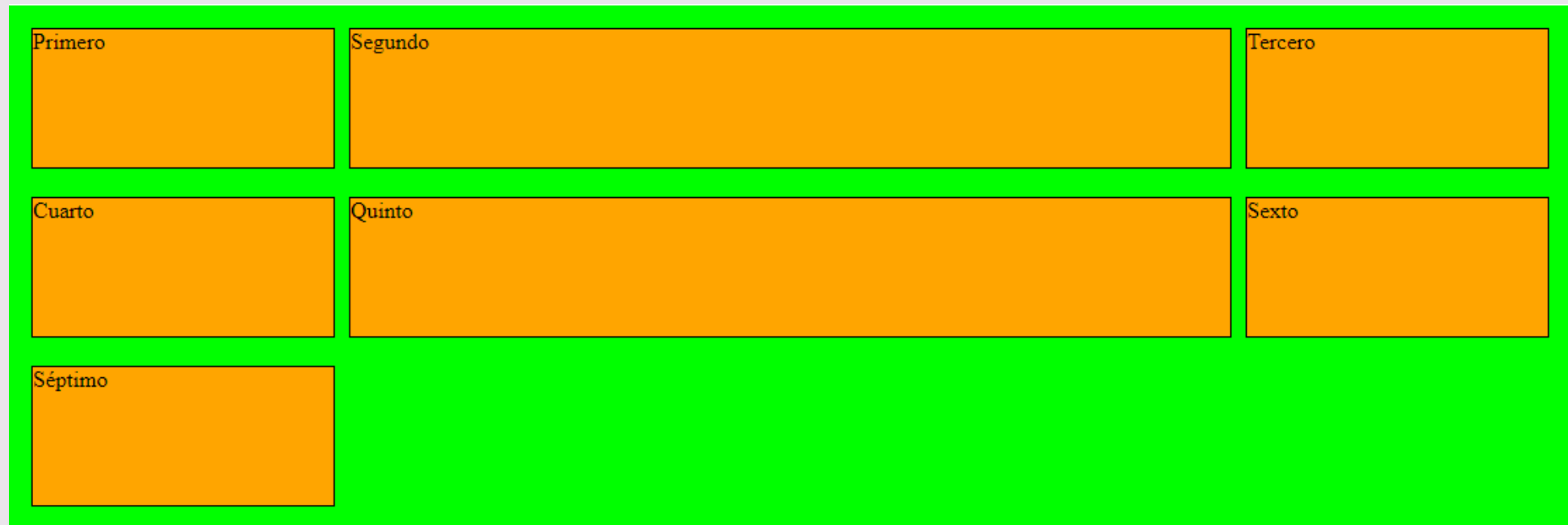
.container > div {
  background-color: orange;
  border: 1px solid black;
}
```

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

- Definición de la estructura del GRID:

- Ejemplo:



16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

■ Alineación Horizontal:

- Por defecto los elementos del GRID ocupan todo el ancho de la celda que le corresponde pero podemos optar por otro tipo de alineaciones horizontales dando valores a la propiedad ***justify-items***. Los diferentes valores que puede tomar son los siguientes:

- **start**
- **end**
- **center**
- **stretch** (opción por defecto)

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

■ Alineación Vertical:

- Por defecto los elementos del GRID ocupan todo el alto de la celda que le corresponde pero podemos optar por otro tipo de alineaciones verticales dando valores a la propiedad ***align-items***. Los diferentes valores que puede tomar son los siguientes:

- **start**
- **end**
- **center**
- **stretch** (opción por defecto)

- Se puede juntar las dos alineaciones (horizontal y vertical)usando la propiedad ***place-items*** indicando primero el valor para ***align-ítems*** (vertical) y después el valor para ***justify-ítems*** (horizontal): `place-items: start end;`

16. CSS Grid Layout

✓ Propiedades en el contenedor Grid

■ Distribución dentro del contenedor:

- En determinados casos puede suceder que los elementos del GRID no ocupen todo el ancho o todo el alto del contenedor GRID. En estas ocasiones puedo distribuir las columnas y las filas usando las propiedades **justify-content** (horizontal) y **align-content** (vertical).
- Se puede juntar las dos alineaciones (horizontal y vertical)usando la propiedad **place-content** indicando primero el valor para **align-content** (vertical) y después el valor para **justify-content** (horizontal): `place-content: start end;`

16. CSS Grid Layout

✓ Elementos del Grid

- Los **Elementos GRID** son los hijos directos, dentro del árbol DOM de nuestra página HTML, del elemento con la propiedad CSS ***display:grid***.
- De manera individual podemos modificar las propiedades de estos elementos para conseguir lo siguiente:
 - Definir el área o zona del GRID (rejilla) que va a ocupar.
 - Especificar la alineación horizontal del elemento.
 - Especificar la alineación vertical del elemento.

16. CSS Grid Layout

✓ Elementos del Grid

■ Área del elemento GRID:

- **grid-column-start:** especifica cuál es la columna de inicio.
- **grid-column-end:** especifica cuál es la columna de fin.
- **grid-row-start:** especifica cuál es la fila de inicio.
- **grid-row-end:** especifica cuál es la fila de fin.

■ Podemos juntar estas propiedades:

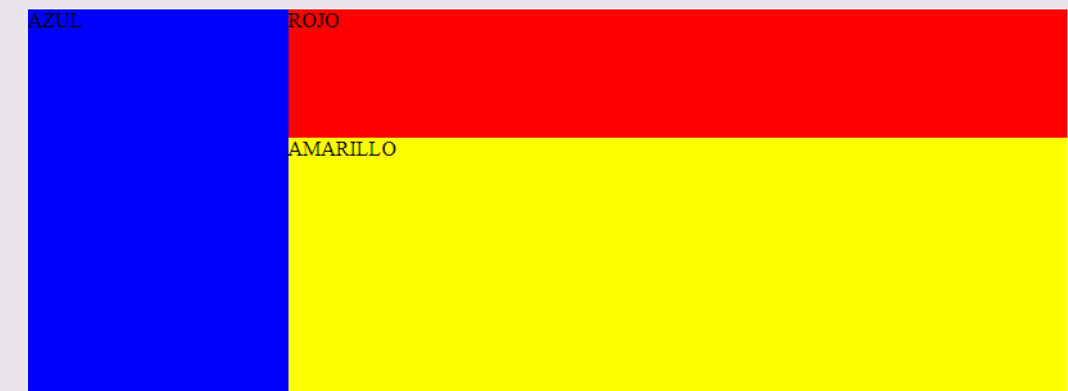
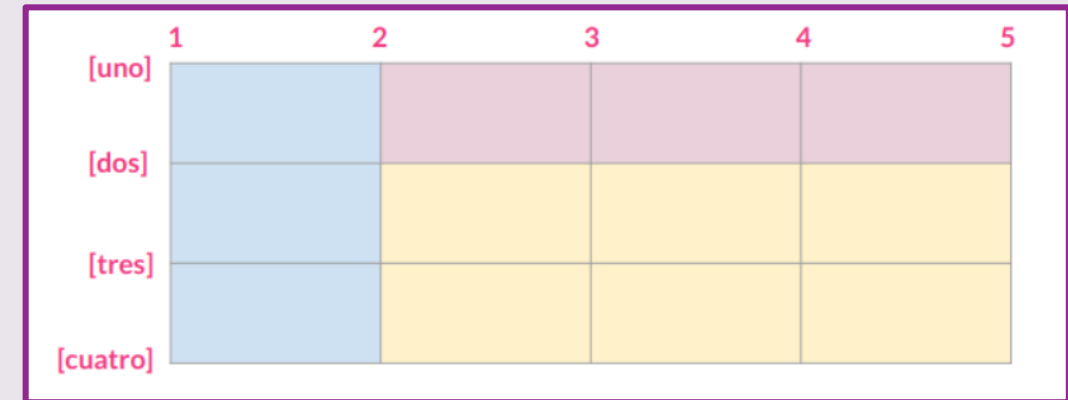
```
/* Para juntar las dos propiedades referentes a las columnas */  
grid-column: start / end;  
  
/* Para juntar las dos propiedades referentes a las filas */  
grid-row: start / end;  
  
/* Para juntar las cuatro propiedades que nos permiten definir el área */  
grid: row-start column-start row-end column-end;
```

16. CSS Grid Layout

✓ Elementos del Grid

- Área del elemento GRID:
 - Ejemplo:

```
#azul {  
  background-color: ■blue;  
  grid-column-start: 1;  
  grid-column-end: 2;  
  grid-row-start: uno;  
  grid-row-end: cuatro;  
}  
  
#rojo {  
  background-color: ■red;  
  grid-column-start: 2;  
  grid-column-end: 5;  
  grid-row-start: uno;  
  grid-row-end: dos;  
}  
  
#amarillo {  
  background-color: ■yellow;  
  grid-column-start: 2;  
  grid-column-end: 5;  
  grid-row-start: dos;  
  grid-row-end: span 2;  
}
```



16. CSS Grid Layout

✓ Elementos del Grid

■ Alineación horizontal:

- La alineación horizontal de un elemento de manera individual se consigue usando la propiedad ***justify-self*** que puede tomar los mismos valores y funciona igual que la propiedad *justify-items* que dábamos al contenedor GRID. Estos valores son:
 - **start**
 - **end**
 - **center**
 - **stretch** (por defecto)

16. CSS Grid Layout

✓ Elementos del Grid

■ Alineación vertical:

- La alineación vertical de un elemento de manera individual se consigue usando la propiedad ***align-self*** que puede tomar los mismos valores y funciona igual que la propiedad *align-items* que dábamos al contenedor GRID. Estos valores son:
 - **start**
 - **end**
 - **center**
 - **stretch** (por defecto)

16. CSS Grid Layout

✓ Áreas Grid

- Podemos maquetar únicamente nombrando áreas. Para ello usaremos las siguiente propiedades:
 - **grid-area** en los elementos GRID
 - **grid-template-area** en el contenedor GRID.

```
.container {  
  display: grid;  
  grid-template-columns: repeat(5, 20%);  
  grid-template-rows: repeat(3, 100px);  
}
```

```
#cab {  
  background-color: orange;  
  grid-area: cab;  
}  
  
#pie {  
  background-color: green;  
  grid-area: pie;  
}  
  
#menu {  
  background-color: lightblue;  
  grid-area: menu;  
}  
  
#principal {  
  background-color: purple;  
  grid-area: main;  
}
```


16. CSS Grid Layout

✓ Áreas Grid

- Añadimos al contenedor la propiedad **grid-template-area** que define la estructura sin dar ninguna propiedad adicional a los elementos del Grid.
- Cada nombre representa el elemento que va a ocupar una celda. De esta manera, el contenido de cada fila va entre "" y:
 - La **primera fila** será ocupada totalmente por el elemento con *grid-area: cab*;
 - En la **segunda fila**, la primera celda para el elemento con *grid-area: menu*, luego un hueco que se indica con . y posteriormente tres celdas para el elemento con *grid-area: main*.
 - En la **tercera fila** la primera celda es para el elemento con *grid-area: menu* y el resto para el que tenga *grid-area: pie*

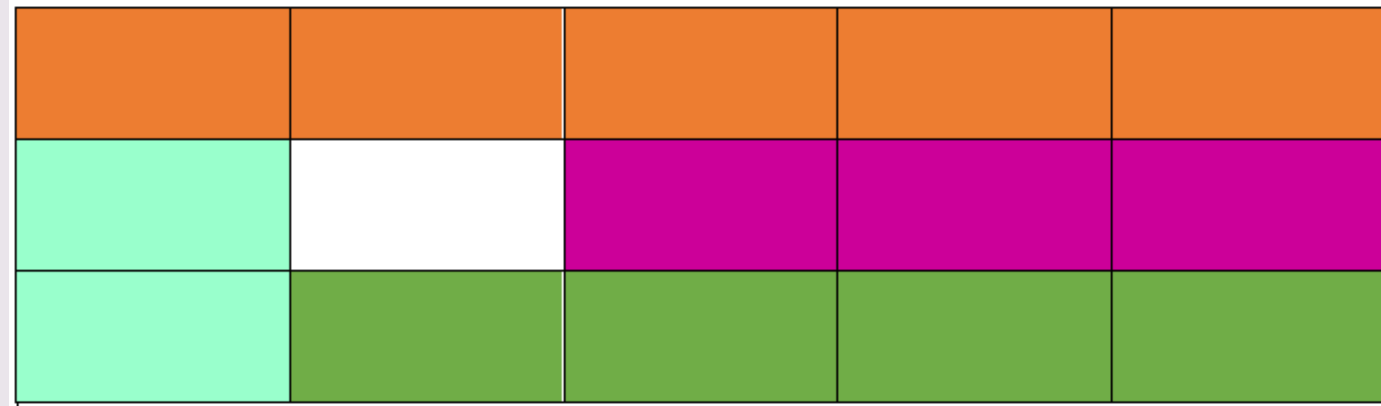
```
.container {  
  display: grid;  
  grid-template-columns: repeat(5, 20%);  
  grid-template-rows: repeat(3, 100px);  
  grid-template-areas:  
    "cab cab cab cab cab"  
    "menu . main main main"  
    "menu pie pie pie pie";  
  margin: 20px auto;  
  width: 80%;  
}
```

16. CSS Grid Layout

✓ Áreas Grid

```
#cab {  
  background-color: orange;  
  grid-area: cab;  
}  
  
#pie {  
  background-color: green;  
  grid-area: pie;  
}  
  
#menu {  
  background-color: lightblue;  
  grid-area: menu;  
}  
  
#principal {  
  background-color: purple;  
  grid-area: main;  
}
```

```
.container {  
  display: grid;  
  grid-template-columns: repeat(5, 20%);  
  grid-template-rows: repeat(3, 100px);  
  grid-template-areas:  
    "cab cab cab cab cab"  
    "menu . main main main"  
    "menu pie pie pie pie";  
  margin: 20px auto;  
  width: 80%;  
}
```



ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. **Propiedades avanzadas de CSS**
18. Responsive Web Design
19. Bibliografía

17. Propiedades de CSS avanzadas

- ✓ CSS Custom Properties
- ✓ Box-sizing
- ✓ Cursor del ratón
- ✓ Efectos de texto
- ✓ Sombra
- ✓ Fuentes
- ✓ Bordes
- ✓ Degradados
- ✓ Transformaciones
- ✓ Transiciones
- ✓ Animaciones
- ✓ Columnas
- ✓ Interfaz de usuario

17. Propiedades de CSS avanzadas

✓ CSS Custom Properties

- **CSS Custom Properties** (muchas veces conocidas por *variables CSS*) son un mecanismo de CSS que permite dar **un valor personalizado** a las propiedades. El objetivo principal suele ser evitar escribir múltiples veces ese valor, y en su lugar, ponerle un nombre más lógico y fácil de recordar, que hará referencia al valor real.
- De esta forma, si necesitamos cambiar el valor en algún momento, podemos hacerlo en esa **propiedad personalizada** y no en múltiples partes del documento, donde nos podríamos equivocar al añadir el mismo valor o incluso hacer mucho más difícil de mantener el código.
- Se suelen agrupar en las primeras líneas de un bloque CSS.

```
:root {  
  --color_h1: ■ #FA003F;  
}
```

```
h1 {  
  color: var(--color_h1);  
}
```

17. Propiedades de CSS avanzadas

✓ Propiedad *box-sizing*:

- Por defecto, el ancho y el alto de un elemento se calcula así:
 - ancho + relleno + borde = ancho real de un elemento.
 - alto + relleno + borde = altura real de un elemento.
 - Cuando se establece el ancho / alto de un elemento, el elemento a menudo parece más grande de lo que se ha establecido (porque el borde y el relleno del elemento se agregan al ancho / alto especificado del elemento).
- **box-sizing**: permite incluir el relleno y el borde en el ancho y alto total de un elemento.
- https://www.w3schools.com/css/css3_box-sizing.asp

17. Propiedades de CSS avanzadas

✓ Propiedad para cambiar el tipo de cursor del ratón

- Especifica el cursor del ratón que se mostrará al apuntar sobre un elemento.
- Sintaxis: *cursor: value;*
- https://www.w3schools.com/cssref/pr_class_cursor.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para dar efectos al texto

- **word-wrap:** sirve para especificar que las palabras que sean demasiado largas se deben cortar, de manera que quepan en el ancho disponible de la caja.
- **text-overflow:** especifica cómo se debe señalar al usuario el contenido desbordado que no se muestra. Se puede recortar, mostrar puntos suspensivos (...) o mostrar una cadena personalizada. Las dos propiedades siguientes son necesarias: *white-space: nowrap;* *overflow: hidden.*

```
p.test {  
  width: 11em;  
  border: 1px solid #000000;  
  word-wrap: break-word;  
}
```

This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

```
p.test1 {  
  white-space: nowrap;  
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: clip;  
}  
  
p.test2 {  
  white-space: nowrap;  
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

text-overflow: clip:

This is some long text that will

text-overflow: ellipsis:

This is some long text that ...

https://www.w3schools.com/css/css3_text_effects.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para dar efectos al texto

- **word-break:** especifica cómo deben romperse las palabras al llegar al final de una línea.
- **writing-mode:** especifica si las líneas de texto se disponen horizontal o verticalmente.

```
p.test1 {  
  width: 140px;  
  border: 1px solid #000000;  
  word-break: keep-all;  
}  
  
p.test2 {  
  width: 140px;  
  border: 1px solid #000000;  
  word-break: break-all;  
}
```

This paragraph
contains some text.
This line will break-
at-hyphens.

This paragraph contain
s some text. The lin
es will break at any c
haracter.

```
p.test1 {  
  writing-mode: horizontal-tb;  
}  
  
span.test2 {  
  writing-mode: vertical-rl;  
}  
  
p.test2 {  
  writing-mode: vertical-rl;  
}
```

Some text with default writing-mode.

Some text with a span element with a vertical-rl writing-mode.

Some text with writing-mode: vertical-rl.

https://www.w3schools.com/css/css3_text_effects.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para dar efectos de sombra

- **text-shadow:** Aplica sombra al texto. Se especifica la sombra horizontal, la sombra vertical y el color de la sombra.

```
h1 {  
  text-shadow: 2px 2px red;  
}
```

Text-shadow effect!

- **box-shadow:** añadir sombras a las cajas.

```
div {  
  width: 300px;  
  height: 100px;  
  padding: 15px;  
  background-color: yellow;  
  box-shadow: 10px 10px grey;  
}
```

This is a div element with a box-shadow

https://www.w3schools.com/css/css3_shadows.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para fuentes

- A la hora de escoger una fuente para usar en una página web, tradicionalmente se encontraba la limitación de que el usuario tuviera ese tipo de letra instalada en su ordenador, porque de no ser así, los textos se mostrarían con otras tipografías, distintas a las que habíamos elegido.
- **@font-face:** permite definir en nuestra hoja de estilos cualquier tipo de tipografía, independientemente de si el usuario dispone de ella o no y para ello lo único que debe preocuparnos es que este instalada en nuestro servidor. Admite formatos tipográficos: .eot (Embedded OpenType); .ttf (True Type Fonts); .otf (OpenType Fonts).
- https://www.w3schools.com/css/css3_fonts.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para fuentes

■ OBLIGATORIOS:

- *font-family*: Nombre de la fuente
- *src*: Ubicación en el servidor.

■ OPCIONALES:

- *font-stretch*: Define cómo se debe estirar la fuente. El valor predeterminado es "normal"
- *font-style*: Define cómo se debe diseñar la fuente. El valor predeterminado es "normal"
- *font-weight*: Define el ancho la fuente. El valor predeterminado es "normal"
- *unicode-range*: Define el rango de caracteres Unicode que admite la fuente.

```
@font-face{  
    font-family:<nombre_fuente>;  
    src: <source>[,<source>]*;  
    [font-weight:<weight>;  
    [font-style:<style>;  
}
```

```
@font-face {  
    font-family: Vivaldi;  
    font-style: normal;  
    font-weight: normal;  
    src: url(VIVALDI0.eot);  
}
```

```
H2{  
    font-family: "Vivaldi";  
}
```

17. Propiedades de CSS avanzadas

✓ Propiedades para fuentes

- Google Fonts: <https://fonts.google.com/>
 - Permite descargar las fuentes en formato TTF.
 - Pueden utilizarse en nuestras páginas web sin necesidad de alojarlas en nuestro propio servidor, con el inconveniente de que en cualquier momento pueden no estar disponibles.
 - Incluir al principio de la hoja de estilo una regla-arroba **@import**. La fuente se puede utilizar en la hoja de estilo como una familia más.

```
@import url('https://fonts.googleapis.com/css?family=Montserrat|Passion+One:900');
```

17. Propiedades de CSS avanzadas

✓ Propiedades de bordes

- **border-radius:** crea las esquinas redondeadas, especificando las medidas del radio que deben darse a la curva de las esquinas.
 - https://www.w3schools.com/cssref/css3_pr_border-radius.asp
- **border-imagen:** utilizar imágenes como bordes. Sintaxis: `border-image: source slice width outset repeat|initial|inherit;`
 - *source*: Para indicar la URL de la imagen que vamos a utilizar como borde.
 - *slice*: Indica el espacio de la imagen que será visible como borde, en los cuatro lados del elemento, es decir, top, right, bottom y left.
 - *width*: Para indicar la anchura del borde.
 - *outset*: Nos sirve para indicar el área en la que la imagen de borde se extiende más allá del área del elemento, en los 4 lados del mismo.
 - *repeat*: Permite marcar si se desea o no que se repita la imagen del borde haciendo un mosaico o si se desea que se estire, etc.
 - https://www.w3schools.com/cssref/css3_pr_border-image.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

- Hasta la llegada de CSS3, necesitábamos usar imágenes como fondo de los elementos para implementarlos.
- Implementan un gradiente de color, que pasa de un estado a otro a lo largo del fondo de los elementos HTML, ya sea capas, elementos de listas, botones, etc.
- Se obtendrán por medio de la especificación de una serie de características, como la posición inicial, la dirección hacia donde se realizará, si es circular o linear, y los colores que se incorporarán en cada uno de los pasos del gradiente.
- El navegador es el encargado de renderizar el gradiente en función de las características escritas para definirlo. Podemos asignar gradientes como fondo en cualquier elemento HTML donde se podía implementar una imagen de fondo.
- Tipos:
 - Gradientes lineales (va abajo / arriba / izquierda / derecha / diagonal)
 - Gradientes radiales (definido por su centro)
- https://www.w3schools.com/css/css3_gradients.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

■ Gradientes lineales:

- Se crea un gradiente que va de un color a otro de manera lineal. Puede ser de arriba a abajo, de izquierda a derecha y viceversa. Incluso se puede conseguir un degradado en un gradiente de una línea con cualquier ángulo.
- Sintaxis:

```
background-image: linear-gradient(direction, color-stop1, color-stop2, ...);
```


17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

■ Gradientes lineales:

- Ejemplo: Gradiente lineal de izquierda a derecha

```
#grad1 {  
  height: 200px;  
  background-color: red; /* For browsers that do not support gradients */  
  background-image: linear-gradient(to right, red , yellow); /* Standard syntax (must be last) */  
}
```



17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

■ Gradientes lineales:

- Ejemplo: Gradiente lineal que comienza en la parte superior izquierda (y va hacia la parte inferior derecha). Empieza en rojo, pasando a amarillo

```
#grad1 {  
  height: 200px;  
  background-color: red; /* For browsers that do not support gradients */  
  background-image: linear-gradient(to bottom right, red, yellow); /* Standard syntax (must be last) */  
}
```



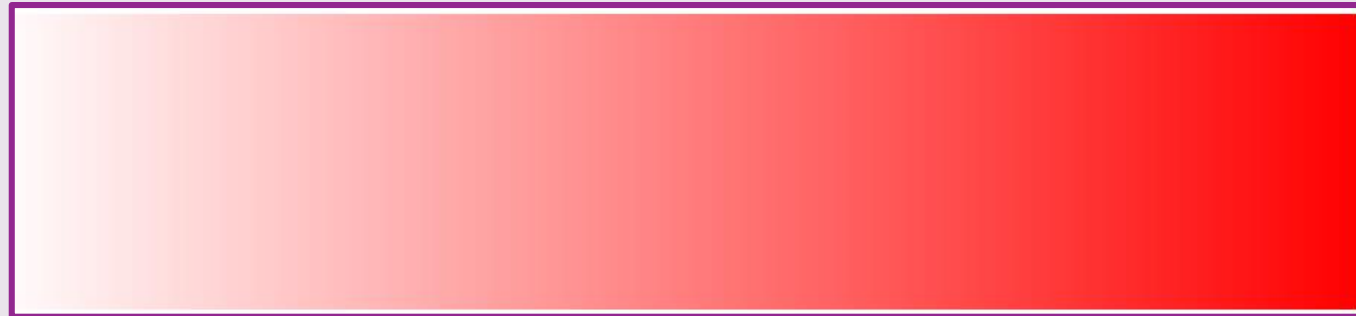
17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

■ Gradientes lineales:

- Ejemplo: Usando transparencia: e puede usar para crear efectos de desvanecimiento. Se utiliza la función rgba (). El último parámetro de la función rgba () puede ser un valor de 0 a 1, y se define la transparencia del color: 0 indica una transparencia total, 1 indica todo color (sin transparencia).

```
#grad1 {  
  height: 200px;  
  background-image: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); /* Standard syntax (must be last) */  
}
```



17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

■ Gradientes radiales:

- Se implementa un gradiente que se distribuye radialmente, desde un punto del elemento hacia fuera, de manera circular, que puede tener el mismo valor de radio (para hacer degradados en círculos perfectos) o con valores de radio variables (lo que generaría elipses).
- Sintaxis:

```
background-image: radial-gradient(shape size at position, start-color, ..., last-color);
```

- Posición inicial del gradiente circular: se especifica con una o dos coordenadas, que pueden tener distintas unidades CSS. Si se omite, se entiende que el degradado tiene que comenzar en el punto central del fondo del elemento
- La forma puede ser circular o elipse, para lo cual especificamos las palabras "*circle*" o "*ellipse*". El tamaño lo expresamos con otra serie de palabras clave, que indican hasta donde debe crecer el círculo o elipse: *closest-side* | *closest-corner* | *farthest-side* | *farthest-corner*. Por ejemplo, *closest-side* indica que el círculo o elipse debe crecer hasta el lado más cercano. La palabra *farthest-corner* indicaría que debe crecer hasta la esquina más lejana.

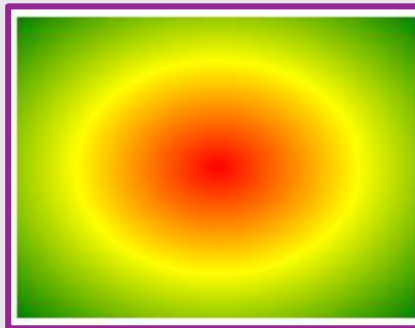
17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

■ Gradientes radiales:

- Ejemplo: Gradiente radial con forma de elipse (por defecto)

```
#grad1 {  
  height: 150px;  
  width: 200px;  
  background-color: red; /* For browsers that do not support gradients */  
  background-image: radial-gradient(red, yellow, green); /* Standard syntax (must be last) */  
}
```



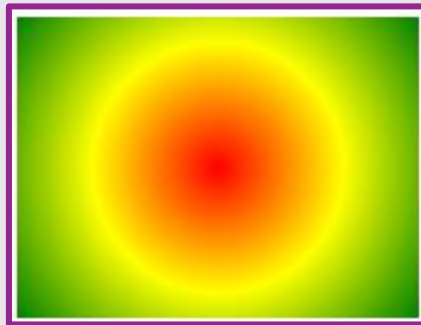
17. Propiedades de CSS avanzadas

✓ Propiedades para definir un gradiente

■ Gradientes radiales:

- Ejemplo: Gradiente radial con forma de círculo

```
#grad2 {  
  height: 150px;  
  width: 200px;  
  background-color: red; /* For browsers that do not support gradients */  
  background-image: radial-gradient(circle, red, yellow, green); /* Standard syntax (must be last) */  
}
```



17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 2D:

- Una transformación es un efecto que permite un cambio de elemento de forma, tamaño y posición.
- La propiedad ***transform*** puede operar cuatro transformaciones básicas:
 - *rotate* (rotar)
 - *scale* (escalar)
 - *skew* (inclinarse)
 - *translate* (trasladar o mover)
 - *matrix* (permiten rotar, escalar, mover e inclinar elementos)
- https://www.w3schools.com/css/css3_2dtransforms.asp

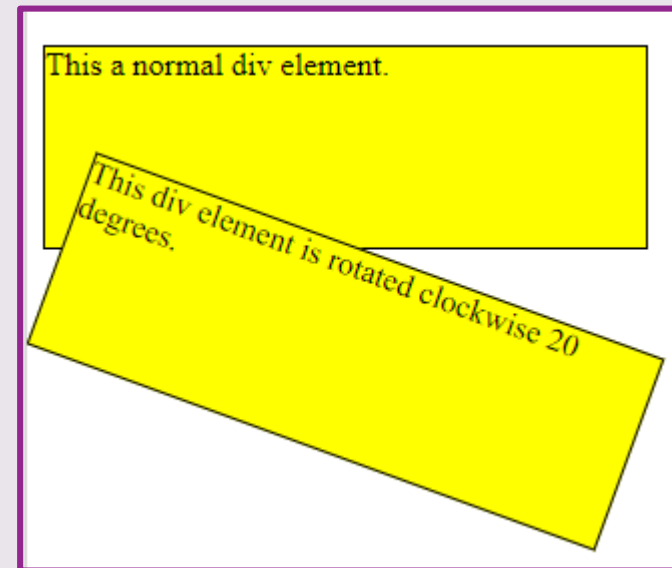
17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 2D:

- Método *rotate()*: Gira un elemento en sentido horario o antihorario según un grado dado. El uso de valores negativos girará el elemento en sentido contrario a las agujas del reloj.

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
div#myDiv {  
  -ms-transform: rotate(20deg); /* IE 9 */  
  -webkit-transform: rotate(20deg); /* Safari prior 9.0 */  
  transform: rotate(20deg); /* Standard syntax */  
}
```



17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 2D:

- Método *scale()*: Aumenta o disminuye el tamaño de un elemento (de acuerdo con los parámetros dados para el ancho y la altura).

```
div {  
  margin: 150px;  
  width: 200px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
  -ms-transform: scale(2,3); /* IE 9 */  
  -webkit-transform: scale(2,3); /* Safari prior 9.0 */  
  transform: scale(2,3); /* Standard syntax */  
}
```

This div element is two times
of its original width, and three
times of its original height.

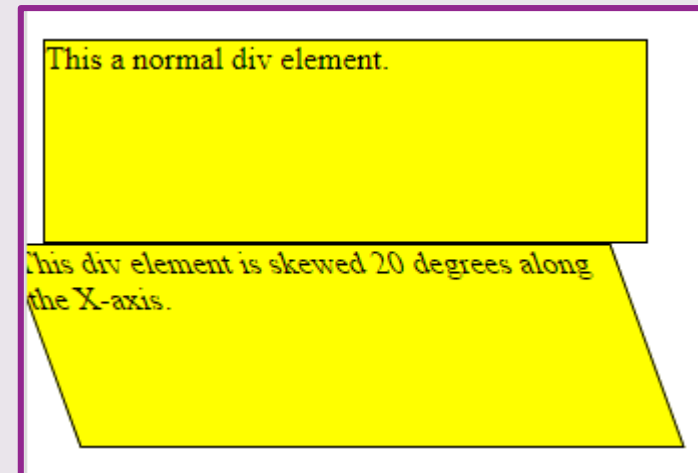
17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 2D:

- Método *skew()*: Sesga un elemento a lo largo de los ejes X e Y según los ángulos dados. Si no se especifica el segundo parámetro, tiene un valor cero.

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
div#myDiv {  
  -ms-transform: skewX(20deg); /* IE 9 */  
  -webkit-transform: skewX(20deg); /* Safari prior 9.0 */  
  transform: skewX(20deg); /* Standard syntax */  
}
```



17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 2D:

- Método *translate()*: Mueve un elemento desde su posición actual (de acuerdo con los parámetros dados para el eje X y el eje Y).

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
  -ms-transform: translate(50px,100px); /* IE 9 */  
  -webkit-transform: translate(50px,100px); /* Safari prior 9.0 */  
  transform: translate(50px,100px); /* Standard syntax */  
}
```

This div element is moved 50 pixels to the right, and 100 pixels down from its current position.

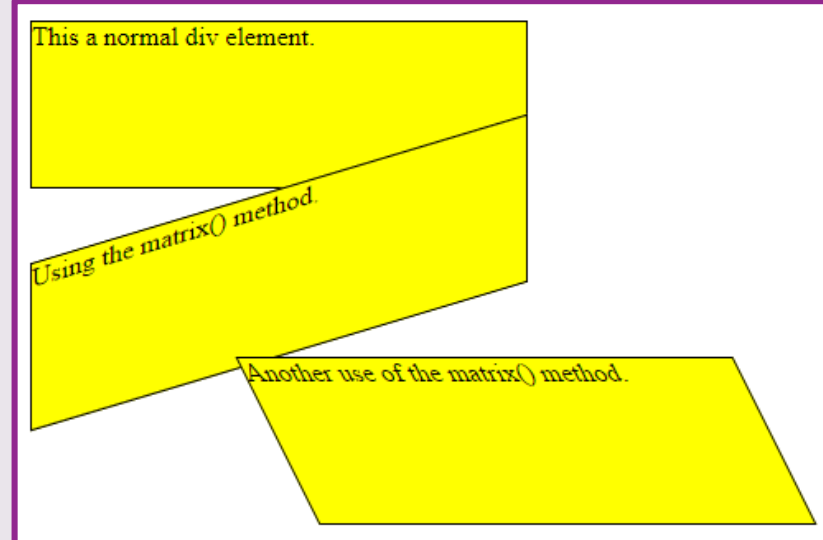
17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 2D:

- Método *matrix()*: Combina todos los métodos de transformación 2D en uno. Toma seis parámetros, lo que le permite rotar, escalar, mover e inclinar elementos.
- *matrix (scaleX (), skewY (), skewX (), scaleY (), translateX (), translateY ())*

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
div#myDiv1 {  
  -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */  
  -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari prior 9.0 */  
  transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */  
}  
  
div#myDiv2 {  
  -ms-transform: matrix(1, 0, 0.5, 1, 150, 0); /* IE 9 */  
  -webkit-transform: matrix(1, 0, 0.5, 1, 150, 0); /* Safari prior 9.0 */  
  transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */  
}
```



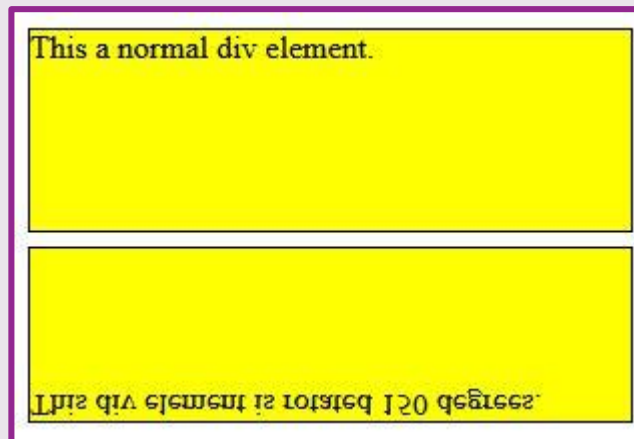
17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 3D:

- https://www.w3schools.com/css/css3_3dtransforms.asp
- Método *rotateX()*: Gira un elemento alrededor de su eje X en un grado dado.

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
div#myDiv1 {  
  -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */  
  -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari prior 9.0 */  
  transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */  
}  
  
div#myDiv2 {  
  -ms-transform: matrix(1, 0, 0.5, 1, 150, 0); /* IE 9 */  
  -webkit-transform: matrix(1, 0, 0.5, 1, 150, 0); /* Safari prior 9.0 */  
  transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */  
}
```



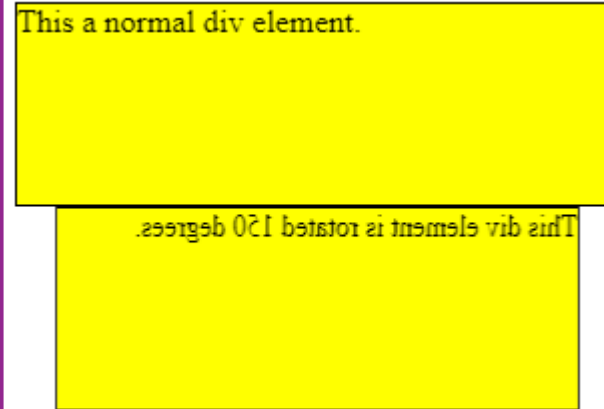
17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 3D:

- https://www.w3schools.com/css/css3_3dtransforms.asp
- Método *rotateY()*: Gira un elemento alrededor de su eje Y en un grado dado.

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
#myDiv {  
  -webkit-transform: rotateY(150deg); /* Safari prior 9.0 */  
  transform: rotateY(150deg); /* Standard syntax */  
}
```



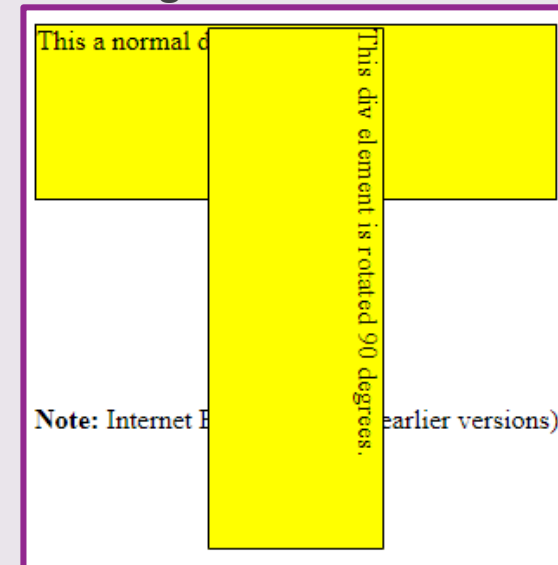
17. Propiedades de CSS avanzadas

✓ Propiedades para transformaciones:

■ Transformaciones 3D:

- https://www.w3schools.com/css/css3_3dtransforms.asp
- Método *rotateZ()*: Gira un elemento alrededor de su eje Z en un grado dado.

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
#myDiv {  
  -webkit-transform: rotateZ(90deg); /* Safari prior 9.0 */  
  transform: rotateZ(90deg); /* Standard syntax */  
}
```



17. Propiedades de CSS avanzadas

✓ Propiedades para transiciones

- Con CSS3, podemos agregar un efecto al cambiar de un estilo a otro, sin el uso de animaciones Flash o archivos JavaScript.
- **Transiciones:** Efectos que permiten un elemento cambiar gradualmente de un estilo a otro.
 - Para crear un efecto de transición, debe especificar dos cosas:
 - la propiedad CSS a la que desea agregar un efecto
 - la duración del efecto
 - **Nota:** Si la parte de la duración no se especifica, la transición no tendrá efecto, porque el valor predeterminado es 0.
- https://www.w3schools.com/css/css3_transitions.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para transiciones

- **transition:** Una propiedad abreviada para configurar las cuatro propiedades de transición en una sola propiedad.
- **transition-delay:** Especifica un retraso (en segundos) para el efecto de transición.
- **transition-duration:** Especifica cuántos segundos o milisegundos tarda en completarse un efecto de transición.
- **transition-property:** Especifica el nombre de la propiedad CSS para la cual se aplica el efecto de transición.
- **transition-timing-function:** Especifica la curva de velocidad del efecto de transición.

17. Propiedades de CSS avanzadas

✓ Propiedades para animaciones

- Las animaciones CSS permiten la animación de la mayoría de los elementos HTML sin usar JavaScript o Flash.
- Una animación permite que un elemento cambie gradualmente de un estilo a otro.
- Puedes cambiar tantas propiedades CSS como quieras, tantas veces como quieras.
- Para usar la animación CSS, primero debe especificar algunos fotogramas clave para la animación.
- Los fotogramas clave contienen qué estilos tendrá el elemento en ciertos momentos.
- https://www.w3schools.com/css/css3_animations.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para animaciones

■ La regla @keyframes

- La regla @ keyframes es donde se crea la animación. Es necesario especificar un estilo CSS dentro de la regla @ keyframes y la animación cambiará gradualmente del estilo actual al nuevo estilo.
- Para que una animación funcione, se debe vincular la animación a un elemento.
- Ejemplo: Vincula la animación "ejemplo" al elemento <div>. La animación durará 4 segundos y cambiará gradualmente el color de fondo del elemento <div> de "rojo" a "amarillo".
- https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation1

17. Propiedades de CSS avanzadas

✓ Propiedades para animaciones

- **@keyframes:** Especifica el código de animación.
- **animation:** Una propiedad abreviada para establecer todas las propiedades de animación.
- **animation-delay:** Especifica un retraso para el inicio de una animación.
- **animation-direction:** Especifica si una animación debe reproducirse hacia adelante, hacia atrás o en ciclos alternos.
- **animation:duration:** Especifica cuánto tiempo debe tomar una animación para completar un ciclo
- **animation-fill-mode:** Especifica un estilo para el elemento cuando la animación no se está reproduciendo (antes de que comience, después de que termine, o ambos)
- **animation-iteration-count:** Especifica el número de veces que se debe reproducir una animación.
- **animation-name:** Especifica el nombre de la animación @keyframes.
- **animation-play-state:** Especifica si la animación se está ejecutando o está en pausa.
- **animation-timing-function:** Especifica la curva de velocidad de la animación.

17. Propiedades de CSS avanzadas

✓ Propiedades para animaciones

- Ejemplo: Ejecuta una animación con todas las propiedades establecidas:
 - https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation4
 - https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation5 (Propiedad abreviada)

17. Propiedades de CSS avanzadas

✓ Propiedades para animaciones

■ Usando un CDN

- Incluir en el documento html un enlace como si fuera un enlace a una hoja de estilo:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">
```

- Agregar al elemento en cuestión la clase **animate__animated** junto con el nombre de la animación:

```
<h1 class="animate__animated animate__bounce">Un elemento animado!</h1>
```

- <https://animate.style/>

17. Propiedades de CSS avanzadas

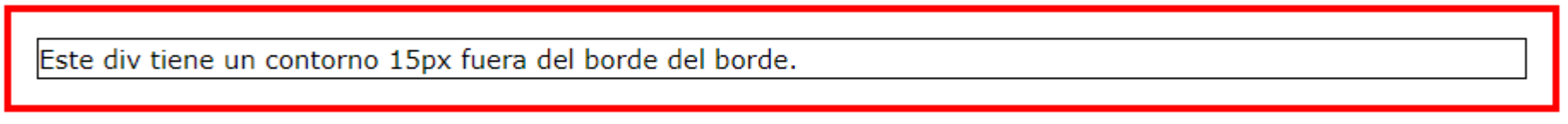
✓ Propiedades para diseñar varias columnas

- El diseño de varias columnas de CSS permite una fácil definición de múltiples columnas de texto, como en los periódicos:
 - *column-count*: Especifica el número de columnas en que se debe dividir un elemento
 - *column-fill*: Especifica cómo rellenar columnas.
 - *column-gap*: Especifica el espacio entre las columnas.
 - *column-span*: Especifica cuántas columnas debe abarcar un elemento.
 - *column-width*: Especifica un ancho óptimo sugerido para las columnas.
 - *columns*: Una propiedad abreviada para establecer el ancho de columna y el conteo de columnas
- https://www.w3schools.com/css/css3_multiple_columns.asp

17. Propiedades de CSS avanzadas

✓ Propiedades para la interfaz de usuario:

- En CSS3, algunas de las nuevas características de la interfaz de usuario está cambiando el tamaño los elementos, caja de tamaño, y la esquematización.
- https://www.w3schools.com/css/css3_user_interface.asp
- La propiedad **resize** especifica si (y cómo) un elemento debe ser redimensionable por el usuario. https://www.w3schools.com/css/tryit.asp?filename=trycss3_resize_width
- La propiedad **outline-offset** agrega espacio entre un contorno y el borde o borde de un elemento. https://www.w3schools.com/css/tryit.asp?filename=trycss3_outline-offset



Este div tiene un contorno 15px fuera del borde del borde.

ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. **Responsive Web Design**
19. Bibliografía

18. Responsive Web Design


- ✓ “Enfoque del diseño web que hace que las páginas web se muestren bien en una variedad de dispositivos y tamaños de pantalla o ventana, desde el tamaño mínimo al máximo. El trabajo reciente también considera la proximidad del espectador como parte del contexto de visualización como una extensión de RWD. El contenido, el diseño y el rendimiento son necesarios en todos los dispositivos para garantizar la usabilidad y la satisfacción. Un sitio diseñado con RWD adapta el diseño al entorno de visualización.”
- ✓ https://en.wikipedia.org/wiki/Responsive_web_design
- ✓ https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design

18. Responsive Web Design

✓ Viewport

- Área de la pantalla en la que el navegador puede renderizar contenido, es decir, el espacio disponible para mostrar mi página web.
- Añadiendo la siguiente línea en la cabecera de nuestra página web podremos empezar a maquetar diseños responsivos:

```
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8" />
  <meta name="author" content="María" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>
```



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

18. Responsive Web Design

✓ Breakpoints

- Anchura de la pantalla en la que se produce el cambio.
- **Breakpoints** elegidos por [Twitter Bootstrap](#):
 - <576px (pantallas pequeñas)
 - 576px-768px (móviles apaisados)
 - 768px-992px (tablets)
 - 992px-1200px (desktop)
 - >1200px (pantallas grandes)

18. Responsive Web Design

✓ Media Queries

- Una de las herramientas esenciales para poder realizar un diseño responsivo de calidad son las **media-queries** que son: “... módulo de CSS3 que nos permite adaptar al representación del contenido a las características del dispositivo...” (Definición de Wikipedia)
- Las *media queries* son expresiones en las que indicamos un tipo de medio y una consulta en relación a las características del dispositivo como alto, ancho e incluso color:

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

- https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries
- https://www.w3schools.com/css/css3_mediaqueries.asp
- https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

18. Responsive Web Design

✓ Media Queries

- Tipos de valores para *mediatype*:

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

18. Responsive Web Design

✓ Media Queries

- Condiciones que se pueden consultar:
 - width | min-width | max-width
 - height | min-height | max-height
 - orientation (landscape / portrait)
 - aspect-ratio | min-aspect-ratio | max-aspect-ratio
 - color | min-color | max-color
- Ejemplo: https://www.w3schools.com/css/tryit.asp?filename=trycss3_media_queries1

18. Responsive Web Design

✓ Media Queries

- Ejemplo:

```
/* Estilos para todo tipo de pantallas con una anchura máxima de 576px */
@media all and (max-width: 576px) {
  h1 {
    font-size: 2em !important;
  }
}

/* Estilos para pantallas con al menos 992px de anchura y que estén apaisadas (más ancho que alto) */
@media screen and (min-width: 992px) and (orientation: landscape) {
  p {
    margin: 0 0 30px;
  }
}

/* Estilos sólo para pantallas que tengan al menos 768px de anchura */
@media only screen and (min-width: 768px) {
  h1 {
    font-size: 3.5em !important;
  }
}
```


ÍNDICE DE CONTENIDOS

1. Introducción
2. Reglas CSS
3. Incluir CSS en HTML
4. Selectores CSS
5. Pseudo-clases y Pseudo-elementos
6. Propiedades y Valores
7. Modelo de Cajas
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. CSS3
15. CSS Flexbox
16. CSS Grid Layout
17. Propiedades avanzadas de CSS
18. Responsive Web Design
19. **Bibliografía**

18. Bibliografía

- ✓ Lenguajes de Marcas y Sistemas de Gestión de Información. Sánchez, F. J. y otros. Editorial: RA-MA.

- ✓ Recursos web:
 - <https://www.w3.org/standards/webdesign/htmlcss>
 - <https://developer.mozilla.org/en-US/docs/Web/CSS>
 - <https://lenguajecss.com/css/>
 - <https://www.w3schools.com/>
 - <https://desarrolloweb.com/home/css>
 - <https://uniwebsidad.com/>