

# UT 2. Lenguajes para la visualización de la información: HTML, XHTML y CSS

---

LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE INFORMACIÓN

I.E.S. Luis Vives - Desarrollo de Aplicaciones Web

Curso 2022/2023

# CSS

---

**Recursos Web.**

Guía referencia: <https://www.w3schools.com/cssref/>

Tutorial: <https://www.w3schools.com/css/default.asp>

Guías: [Web technology for developers | MDN \(mozilla.org\)](#)

Hojas de estilo en cascada (Cascading Style Sheets) o CSS, es un lenguaje creado por W3C que añade diferentes estilos a documentos HTML o XHTML, dotándolos de formato.

Se crearon para poder separar el contenido de un documento de su aspecto, lo que permite tener documentos mejor estructurados y limpios. Para separar el contenido de la presentación, se usa una serie de reglas sobre elementos que hay que declarar previamente, ya sea en la misma página o en un documento externo.

Las hojas de estilo pueden aplicarse a un documento entero, a una parte o, simplemente, a una etiqueta concreta, ya que pueden definirse varios estilos para una misma etiqueta.

Permiten modificar la presentación de los elementos de un documento HTML sin tocar su código.

Ahorran tiempo a la hora de definir estilos y ofrecen un abanico extenso de herramientas para el formato, más amplio que HTML.

Para ver en funcionamiento como una hoja de estilo modifica la visualización de una web, recomendando visitar CSS Garden.

**Recurso Web.** CSS garden es una web dónde puedes ver como el mismo código HTML es visualizado de diversas formas sólo usando CSS.

<http://www.csszengarden.com/>

Para poder comprender mejor el funcionamiento, puede observarse el ejemplo siguiente, donde se crean dos documentos independientes. Primero, se aprecia la página web con código HTML:

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  <h1>
    Principales cultivos subtropicales</h1>
  <ul>
    <li>Longan</li>
    <li>Aguacate</li>
    <li>Chirimollo</li>
    <li>Litchi</li>
  </ul>
</body>
</html>
```

## Principales cultivos subtropicales

- Longan
- Aguacate
- Chirimollo
- Litchi

Al incluir la siguiente hoja de estilos en la cabecera del fichero html.

```
<head>
  <title>LM</title>
  <link rel="stylesheet" type="text/css" href="hoja_estilo.css"/>
</head>
```

El fichero css hoja\_estilo.css con el siguiente contenido:

```
body{background-color:Blue;}
h1{font-family:Verdana; color:White;}
li{text-decoration:underline}
```

El resultado sería:



# Versiones

---

CSS



El W3C es el encargado de mejorar cada una de las versiones que van saliendo al mercado, dotándolas de más funcionalidad y corrigiendo algunos problemas que pudieran tener versiones antiguas.

- CSS 1: a finales de 1996, se publica la primera versión que permite describir formato, alineación y atributos de texto, así como las propiedades de las fuentes (tamaño, color, tipo, etc.). Actualmente, no se usa.
- CSS 2: 1998. incluye casi la totalidad de CSS 1. A mediados de 1998, se publica la segunda recomendación oficial de W3C. Añade como mejora para las capas <DIV> el tipo de posicionamiento y nivel.
- CSS 2.1: desde 2004. Elimina errores y funcionalidades en desuso de versiones anteriores.
- CSS 3: 2011. Usa módulos que añaden nueva funcionalidad a CSS 2, preservando de esa manera la compatibilidad con versiones anteriores. Gradientes, esquinas redondeadas, nuevos layouts (Flex y Grid), diseño responsivo, transiciones y animaciones.

# Formas de aplicar los estilos

---

CSS

De manera general podremos añadir CSS a nuestro HTML de cuatro formas:

- A. Etiqueta link
- B. Etiqueta style
- C. Estilos in-line
- D. Directiva @import

**A) Estilos aplicados en un documento externo.** Se creará un documento de hojas de estilo, cuya extensión será .css, donde se declararán las distintas reglas de la página web.

El fichero CSS que tiene que crearse será de texto plano, donde se añadirán las reglas con sintaxis CSS y se enlazará con el documento HTML mediante la **etiqueta <link>**, que se añade **dentro de la etiqueta <head>**. La etiqueta <link> corresponde al tipo de etiquetas vacía al no tener contenido. Se usa para relacionar un recurso externo con el documento HTML, aunque se emplea para incluir el fichero de hojas de estilo externo, puede tener más usos. Entre sus atributos, destacan:

- **type:** informa del tipo de contenido del fichero por enlazar. Para el uso de hojas de estilo, el valor que ha de añadirse es "text/css", que informa de que el archivo es de texto y su sintaxis es CSS.
- **href:** URL o ruta donde se localiza el nombre del documento CSS. Es la fuente del fichero que quiere enlazarse, que puede tener una ruta absoluta o relativa.
- **rel:** informa de la relación que existe entre el documento externo y el actual. Para el caso particular de hojas de estilo, su valor es "stylesheet", que indica que el fichero por enlazar es una hoja de estilos que aplicará el formato al documento HTML.
- Se puede añadir el [atributo media](#) para personalizar el tipo de dispositivo al que se aplicará (media="screen") aunque hay otras formas de [personalizar el medio](#)

**<link href="css/estilos.css" rel="stylesheet" type="text/css" />**

Etiqueta <link>:

```
<link rel="stylesheet" type="text/css" href="miestilo.css"/>
```

Documento HTML:

```
<html>
<head>
  <title>LM</title>
  <link rel="stylesheet" type="text/css" href="hoja_estilo.css" />
</head>
<body>
  <h1>Estilos aplicados a una tabla</h1>
  <table border="1">
    <tr>
      <td>floración</td>
    </tr>
    <tr>
      <td>fruto</td>
    </tr>
  </table>
</body>
</html>
```

Fichero de hojas de estilo: hoja\_estilo.css

```
td {
  background-color:Green;
  font-family:Arial Black;
}
```

## B) Estilos aplicados en el mismo documento.

Unificando todo en un único documento, distinguiendo claramente la zona de aplicación de estilos que serán efectivos para los elementos de la página web que se hayan declarado.

Se usa la etiqueta **<style>** dentro de la etiqueta **<head>**. El contenido de la etiqueta **<style>** estará compuesto por una serie de reglas CSS que afectarán a todo el documento. Cualquier cambio de formato que quiera hacerse sobre uno de los elementos de la página web tendrá que realizarse en dicho apartado. El documento HTML que se observa lleva insertados los estilos en la cabecera sin necesidad de crear un documento externo para enlazarlos. Cuando se trabaja con pocas reglas, el documento queda elegante y legible, pero si, por el contrario, el contenido de la etiqueta **<style>** es muy extenso, es recomendable ponerlo en un documento externo.

Si hubiera algún cambio en el estilo de mi sitio sólo tendría que hacerlo una vez por página. Sigue siendo mucho trabajo si mi sitio web está compuesto, por ejemplo, por miles de páginas.

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">

    td {
      background-color: Green;
      font-family: Arial Black;
    }

  </style>
</head>
<body>
  <h1>Estilos aplicados a una tabla</h1>
  <table border="1">
    <tr>
      <td>floración</td>
    </tr>
    <tr>
      <td>fruto</td>
    </tr>
  </table>
</body>
</html>
```

**C) Estilos aplicados a las etiquetas de un documento.** Se puede usar cuando pretenden tenerse dos elementos iguales, pero hay que aplicarles un formato distinto para cada uno, además es tan concreto que no se repetirá en el resto de la web. Un ejemplo podría ser tener dos etiquetas `<h3>` y que la primera se muestre con un color de texto gris y la segunda, azul.

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  <h3 style="color: Gray;">
    aplica estilo a la etiqueta h1 y pone el texto de color gris</h3>
  <h3 style="color: blue;">
    aplica estilo a la etiqueta h1 y pone el texto de color azul</h3>
  <table border="1">
    <tr>
      <td style="background-color:Yellow;">
        floración
      </td>
    </tr>
    <tr>
      <td style="background-color:Olive;">
        fruto
      </td>
    </tr>
  </table>
</body>
</html>
```

aplica estilo a la etiqueta h1 y pone el texto de color gris

aplica estilo a la etiqueta h1 y pone el texto de color azul

floración
fruto

## D) Estilos importados - [https://www.w3schools.com/cssref/pr\\_import\\_rule.php](https://www.w3schools.com/cssref/pr_import_rule.php)

Permite importar un fichero externo css directamente en nuestro HTML o en otro css.

Debe preceder a las demás reglas (excepto @charset) del head

Se utiliza dentro de la etiqueta style y con comillas simples, de esta manera:

```
<style>
```

```
  @import "css/import.css";
```

```
</style>
```

import.css sería el nombre de nuestro fichero de estilos pudiendo ser el que nosotros queramos.

Utilizar esta técnica me permite realizar cambios en el estilo de toda mi web únicamente tocando un fichero.

Sin embargo hay un inconveniente. El rendimiento, ya que es bloqueante en algunos navegadores, y se para la descarga del resto de la página hasta que no se ha descargado la hoja de estilos. Por lo tanto hay que usar esta opción con cuidado.



El orden creciente de prioridad (de **menor a mayor**) es:

- Prioridad 1. Propiedades por defecto del navegador.
- Prioridad 2. Hojas de estilo externas.
- Prioridad 3. Hojas de estilo internas.
- + Prioridad 4. Hojas de estilo en línea.

Los estilos declarados en la hoja de estilo en línea tienen, por tanto, la prioridad más alta, y por tanto, en el caso que haya estilos diferentes, el navegador web, interpretará que el estilo con mayor prioridad deberá ser mostrado por delante del resto.

a) Los estilos se van propagando hacia abajo o lo que es lo mismo si especificamos una propiedades en un elemento padre los hijos tienen el mismo valor para esas propiedades.

b) Si hay más de una regla que se puede aplicar al mismo elemento y hay conflicto, entonces se aplica la regla más específica (si son de igual nivel se aplica la última definida)

Es posible indicar que una regla tiene prioridad utilizando **!important**

```
h1 {  
  background_color: red !important;  
}
```

[https://developer.mozilla.org/es/docs/Learn/CSS/Building\\_blocks/Cascade\\_and\\_inheritance](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance)

# Sintaxis básica

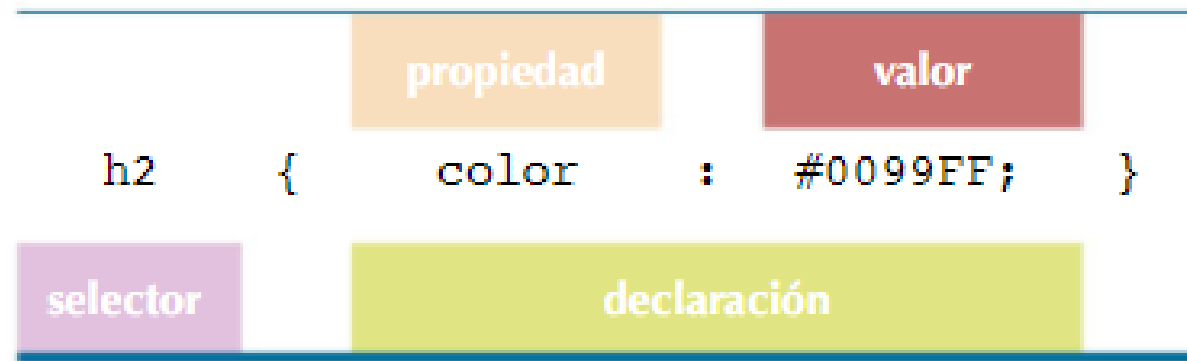
---

CSS

Para poder trabajar con hojas de estilo en cascada hay que tener en cuenta una serie de aspectos:

**Reglas CSS.** Las reglas son los pilares fundamentales a la hora de crear estilos a un documento. A continuación, se describirá cada uno de sus elementos tratando un ejemplo concreto.

- Regla. Es una declaración sobre un estilo de uno o varios elementos. Está compuesta por dos partes diferenciadas: un selector y una declaración.



- **Selector**. Pueden ser uno o varios los elementos a los que se aplica la regla CSS. Dependiendo de la declaración que se añada, afectará a uno o a varios elementos del documento creado. En este ejemplo, el selector es h2 y a todos los elementos h2 del documento HTML o XHTML les afectará la declaración realizada. Para definir el estilo de un selector, se escribe la etiqueta seguida de la lista de declaraciones encerradas entre llaves. Un selector puede tener varias declaraciones separadas por el carácter punto y coma.
- **Declaración**. Está compuesta por dos partes diferenciadas, propiedad y valor, separadas por el carácter dos puntos es donde se especifica el estilo del selector o elemento y puede contener varias propiedades en la misma declaración. Así, la declaración es color:#0099FF
- **Propiedad**. Permite modificar el aspecto de una característica de un selector. Ejemplos de propiedades pueden aplicarse sobre: color, fuente, márgenes, texto, etc.
- **Valor**. Indica la asignación que se hace a la propiedad. Por ejemplo, a la propiedad color, pueden aplicársele valores de la paleta de Windows (negro, blanco, azul, etc.).

# Selectores

---

[CSS Selectors Reference \(w3schools.com\)](https://www.w3schools.com/css/default.asp)

**Selectores.** Son reglas o patrones para seleccionar elementos de una página web y se utilizan para modificar sus propiedades en la página web

Formato:

selector {

prop1:valor

prop2:valor

...

propN: valor

}



Cualquier elemento HTML puede ser un selector. En la imagen anterior, se observan los más usados.

Algunos tipos de selectores

- **etiqueta** selecciona esas etiquetas concretas.
- **\*** Selecciona todo los elementos.
- **#id** selecciona el elemento que tenga ese valor (id en este ejemplo) en el atributo id.
- **.class** selecciona los elementos que tengan ese valor (class en este ejemplo) en el atributo class.
- **selector1,selector2** sirve para cambiar las propiedades de los elementos seleccionados por ambos selectores
- **selector1 selector2** sirve para cambiar las propiedades de los elementos seleccionados **por selector2 que se encuentran dentro** de aquellos seleccionados **por selector1**.
- **selector1>selector2** sirve para cambiar las propiedades de los elementos seleccionados por selector2 que son hijos directos (en el árbol DOM) de aquellos que se seleccionen con selector1.
- **selector1+selector2** sirve para cambiar las propiedades de los elementos seleccionados por selector2 que están justo después de aquellos que se seleccionan mediante selector1.
- **selector1~selector2** igual que el anterior pero todos del selector2 que están después del selector1 (no solo el primero sino varios)
- **[atribut expr valor]** siendo expr (=,~=,|=,\$=,...) para seleccionar elementos atendiendo al valor de sus atributos (igual, contiene, empieza por, termina por,...)

<https://code.tutsplus.com/es/tutorials/the-30-css-selectors-you-must-memorize--net-16048>

[https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_Selectors](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Selectors)

[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

También hay pseudoclasas como **:not(selector)** para seleccionar los que no cumplan el selector.

Selector de tipo. Se aplica a los elementos especificados de una página. Hay que añadir el nombre del elemento seguido de una declaración.

```
body{  
    background-color:Blue;  
}
```

Selector universal. Utiliza el carácter asterisco (\*) y se aplica la regla a todos los elementos de la página web. No suele usarse con frecuencia.

```
*{  
    declaración  
}
```



Selector descendiente. Se usa cuando un elemento es descendiente de otro, es decir, se encuentra dentro de otro (entre la etiqueta de apertura y cierre). **No hace falta que sea descendiente directo** y no afectará al resto de elementos de la página siempre y cuando no desciendan del especificado.

A continuación, se ha creado un selector descendiente (h3 i) que afecta a los elementos i que se encuentran dentro de un elemento de cabecera h3 aplicándoles el formato: color de texto rojo, tipo de fuente Verdana y texto subrayado.

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">
    h3 i { color: red;
          font-family: Verdana;
          text-decoration: underline;
        }
  </style>
</head>
<body>
  <h3> Cultivo del <b>Litchi</b> en la zona de la <i>Axarquía</i></h3>
  <h3> >Cultivo del <b><i>longan</i></b> en Canarias</h3>
  <h3> >Cultivo de <b>la carambola</b> en Málaga</h3>
  <h3> >Cultivo del <i>canisté</i> en Canarias</h3>
  <h4> >Cultivo de la <i>guanábana</i> en Canarias</h4>
</body>
</html>
```

Cultivo del Litchi en la zona de la **Axarquía**

Cultivo del **longan** en Canarias

Cultivo de la carambola en Málaga

Cultivo del **canisté** en Canarias

Cultivo de la *guanábana* en Canarias

Selector de clase. Se usa para aplicar estilos a un elemento en concreto del documento. El nombre que se asigna a un selector de clase puede ser el que se desee, aunque suele darse uno que identifique a la regla que quiere crearse y que **empiece por el carácter punto** (.). Existen dos maneras de definir un selector de clase:

- `NombreSelector{ declaración}.`
- `elemento.NombreSelector{ declaración}.`

Para poder usar el selector en un elemento, se usa el atributo *class* de las etiquetas HTML a las que quiere aplicarse el estilo. La nueva regla creada podrá aplicarse a todas las etiquetas HTML del documento que use el atributo *class* seguido del valor, que será el nombre del selector o clase **sin el punto**.

```
<etiqueta class="selector de clase">
```

Una etiqueta en html puede tener asociadas varias clases separadas por blanco Ejem `<span class="subr negrita"> hola</span>` hará que a la palabra hola se le aplique la regla de `.subr` y la de `.negrita` . También se le aplicaría la regla `span.subr.negrita {...}`

A continuación puedes ver un par de ejemplos

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">
    .colorRojo
    {
      color: red;
      font-family: Verdana;
    }
    .Color_fondo_azul
    {
      background-color: Blue;
    }
  </style>
</head>

<body>
  <h3 class="colorRojo">Cultivo del Litchi.</h3>
  <h3>Cultivo del longan en Canarias.</h3>
  <h3>Cultivo de la <b class="Color_fondo_azul">carambola.</b> </h3>
  <h3>Cultivo del <i class="colorRojo">canisté</i>.</h3>
  <h4 class="Color_fondo_azul"> Cultivo de la guanábana.</h4>
</body>
</html>
```

Cultivo del Litchi.

Cultivo del longan en Canarias.

Cultivo de la carambola.

Cultivo del *canisté*.

Cultivo de la guanábana.

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">
    b.colorRojo
    {
      color: red;
    }
  </style>
</head>
<body>
  <h3 class="colorRojo">Cultivo del Litchi.</h3>
  <h3>
    Cultivo del longan en Canarias.</h3>
  <h3>
    Cultivo de la <b class="colorRojo">carambola.</b>
  </h3>
  <h3>
    Cultivo del <i class="colorRojo">canisté</i>.</h3>
  <h4 class="colorRojo">
    Cultivo de la guanábana.</h4>
</body>
</html>
```

Cultivo del Litchi.

Cultivo del longan en Canarias.

Cultivo de la **carambola**.

Cultivo del *canisté*.

Cultivo de la guanábana.

Selector por ID. Similar a los selectores de clase, pero siendo la **almohadilla (#)** el carácter que precede al selector en vez del punto. Se usa para aplicar un estilo a un elemento concreto del documento mediante el valor del **atributo id** de los elementos, **que es único**.

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">
    #colorRojo
    {
      color: red;
    }
  </style>
</head>
<body>
  <h3 id="colorRojo">Cultivo del Litchi.</h3>
  <h3>Cultivo del longan en Canarias.</h3>
  <h3>Cultivo de la <b>carambola.</b></h3>
  <h3> Cultivo del canisté.</h3>
  <h4 >Cultivo de la guanábana.</h4>
</body>
</html>
```

Cultivo del Litchi.

Cultivo del longan en Canarias.

Cultivo de la carambola.

Cultivo del canisté.

Cultivo de la guanábana.

Selector hijo. Similar al descendiente, pero, esta vez, solo afecta a **hijos directos** de los elementos. Para ello, se usa el **signo mayor que (>)**.

En el siguiente ejemplo, se presenta un selector hijo (h3>i) y un selector descendiente (h3 i). El primero de ellos pondrá de color rojo el texto de las etiquetas que sean hijas directas de la cabecera h3, mientras que, en el segundo, al ser adyacente, no hace falta que sean hijas directas de h3, aplicando un estilo al texto (subrayado y aumentado de tamaño).

Cultivo del Litchi en la zona de la ***Axarquía***

Cultivo del ***longan*** en Canarias

Cultivo de la carambola en Málaga

Cultivo del ***canisté*** en Canarias

Cultivo de la *guanábana* en Canarias

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">
    h3 > i {
      color: red;
    }
    h3 i {
      text-decoration:underline;
      font-size:xx-large;
    }
  </style>
</head>
<body>
  <h3> Cultivo del <b>Litchi</b> en la zona de la <i>Axarquía</i></h3>
  <h3 >Cultivo del <b><i>longan</i></b> en Canarias</h3>
  <h3 >Cultivo de <b>la carambola</b> en Málaga</h3>
  <h3 >Cultivo del <i>canisté</i> en Canarias</h3>
  <h4 >Cultivo de la <i>guanábana</i> en Canarias</h4>
</body>
</html>
```

Selector adyacente. Se usa para elementos **que están seguidos** de otros y son **hermanos** (el elemento padre de ambos es el mismo). El **signo (+)** se emplea para la declaración entre los elementos. En el siguiente ejemplo, el selector adyacente es (h1 + h3), indica que se aplicará el estilo de color de texto rojo a los elementos de cabecera h3 que sean hermanos de h1 y aparecen **justo después** de él. Hay que destacar que h1 y h3 son hermanos, ya que el padre de ambos es el elemento <body>. El signo (~) se usa para cuando afecta a todos los que siguen y no solo al primero

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
```

```
  <style>
```

```
    h1~h3{ background-color: pink;}
```

```
    h1+h3{background-color: blue;}
```

```
  </style>
```

```
  <meta charset="UTF-8">
```

```
  <title>Document</title>
```

```
</head>
```

```
<body>
```

```
  <h1>nivel 1</h1>
```

```
  <h3>nivel 3</h3>
```

```
  <h3>nivel 3</h3>
```

```
  <h3>nivel 3</h3>
```

```
  <h1>nivel 1</h1>
```

```
  <h3>nivel 3</h3>
```

```
  <h2>nivel 2</h2>
```

```
  <h3>nivel 3</h3>
```

```
</body>
```

```
</html>
```

nivel 1

nivel 3

nivel 3

nivel 3

nivel 1

nivel 3

nivel 2

nivel 3

h1 ~ h3 { /\* cualquier h3 después de las cabeceras h1 \*/

background-color: pink;

h1 + h3 { /\* h3 justo después de cabeceras h1 \*/

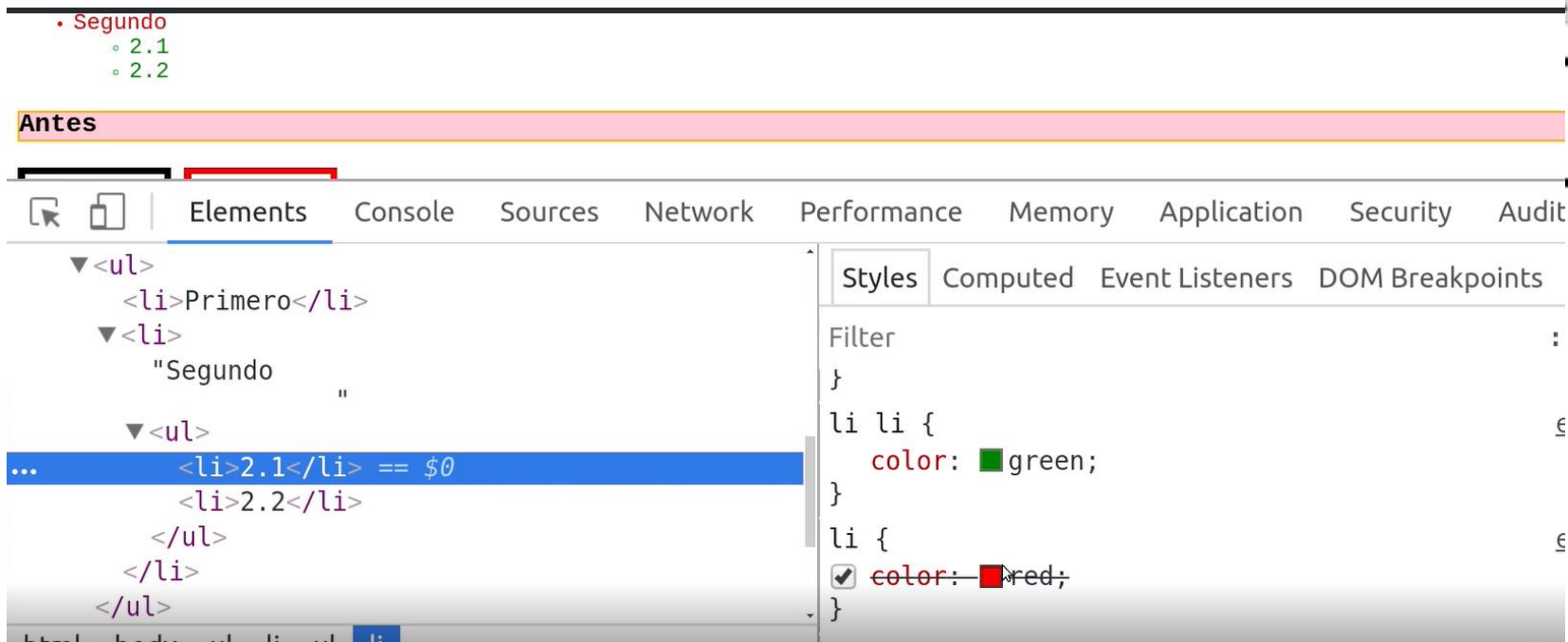
background-color: blue;

h1 ~h2 {background-color: yellow;}

/\*Si no estuviera la regla del h1 +h3 todos los h3 serían rosa\*/



Pulsando, con botón izquierdo de ratón, sobre la lista (li) en el navegador y seleccionado inspeccionar podemos ver lo que se está aplicando



Se pueden realizar modificaciones en los estilos o en el html, que serán visibles automáticamente en el navegador.

Vemos que una regla se está machacando y aparece tachada, la del li

Se puede hacer una integración de Edge y Visual Studio Code para facilitar el desarrollo:

<https://docs.microsoft.com/es-es/microsoft-edge/visual-studio-code/ide-integration>



# Unidades de medida

---

Gracias a CSS, se consigue mayor precisión a la hora de ubicar en una zona del documento un elemento concreto y definir distancias y visibilidad de los elementos, contando con un mayor número de unidades de medida absolutas y relativas que se establecen con un valor entero o real que ayudan a definir los márgenes, anchura, etc., de los distintos elementos.

## Unidades absolutas:

Unidad	Nombre	Descripción	Valor	Ejemplo
pt	punto	1 pt = 0.35mm	entero	48pt
mm	milímetro	1 mm = 0.1cm	entero	60mm
cm	centímetro	1 cm = 0.39in	entero	6cm
in	pulgada (inch)	1 in = 72pt	real	0.2in

[https://developer.mozilla.org/es/docs/Learn/CSS/Building\\_blocks/Values\\_and\\_units](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Values_and_units)

## Unidades relativas:

Unidad	Descripción	Valor	Ejemplo
em	Unidad relativa que se basa en el tamaño del tipo de letra por defecto de la página.	real	1.8em
ex	Unidad relativa a la altura de la letra minúscula en el elemento seleccionado.	real	1.3ex
px	Un píxel es la parte más pequeña de una imagen. Depende de la resolución de pantalla.	entero	220px
%	Porcentaje.	entero	80%

**No puede dejarse un espacio entre el valor y la unidad.** Si se deja algún espacio entre el valor y la unidad, la hoja de estilo no se acepta y, por lo tanto, no se muestra.

Ciertas propiedades aceptan valores negativos.

Por lo general, se **recomienda** utilizar la unidad **em** para describir el tamaño del tipo de letra y así lograr una estabilidad mayor entre los distintos sistemas operativos y navegadores.

Las hojas de estilo CSS proporcionan múltiples notaciones para declarar un color:

- **Hexadecimal:** el valor es "#rrggb", siendo ese grupo de letras 3 agrupaciones numéricas con valores hexadecimales (0 1 2 3 4 5 6 7 8 9 A B C D E F). Los valores de baja intensidad se representan con 00, donde FF son los de mayor intensidad. El color negro se representa con el código #000000 mientras que el blanco es #FFFFFF.
  - rr es la intensidad de rojo
  - gg es la intensidad de verde.
  - bb es la intensidad de azul.

- **Por nombre:** red, yellow, blue, brown, silver, gray, black, white, etc...
- **RGB:** son colores aditivos o primarios y están basados en la combinación de píxeles de colores rojo (RED), verde (GREEN) y azul (BLUE).
  - Los colores se codifican con un número.
  - Los valores para cada color van desde 0 a 255, por lo tanto, hay 256 valores posibles para cada valor. Con la unión de los tres colores en su máximo valor, se obtiene el color blanco (255,255,255) y, con la ausencia de todos, el negro (0,0,0).
  - Permite representar 256 x 256 x 256 combinaciones diferentes de colores, con lo que hay 16 millones de colores.
  - Debe tenerse en cuenta que puede que no todos los navegadores admitan esta codificación de colores.
  - La notación RGB también permite la notación por porcentaje, por ejemplo, color: rgb(25%, 50%, 0%). El valor 0% significa la ausencia de dicho componente y 100% su valor máximo.

# Propiedades y valores

---

**Propiedades y valores.** Un selector puede contener una o varias propiedades. A continuación se muestran las más importantes. Para obtener mayor detalle de las propiedades que veremos aquí y buscar más propiedades que se le pueden aplicar a HTML usando CSS podéis visitar el siguiente recurso web:

## Recursos Web.

Guía referencia: <https://www.w3schools.com/cssref/>

Algunas propiedades y valores de texto :

selector {

font-size: Tamaño

font-weight: anchura

font-family: tipo de letra

text-align: alinear, justificar

text-decoration: subrayar, tachar,

text-indent: tabular

text-transform: convertir mayúsculas o minúsculas

}



## Propiedades de texto.

- **text-decoration.** Decora el texto. Sus valores son: **overline**, **line-through**, **underline**, **none**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Lenguaje de Marcas</title>
    <style>
      #tOverline {
        text-decoration: overline;
      }
      #tLineThrough {
        text-decoration: line-through;
      }
      #tUnderline {
        text-decoration: underline;
      }
      #tNone {
        text-decoration: none;
      }
    </style>
  </head>
  <body>
    <p id="tOverline">Esto es una prueba de texto</p>
    <p id="tLineThrough">Esto es una prueba de texto</p>
    <p id="tUnderline">Esto es una prueba de texto</p>
    <p id="tNone">Esto es una prueba de texto</p>
  </body>
</html>
```

Esto es una prueba de texto

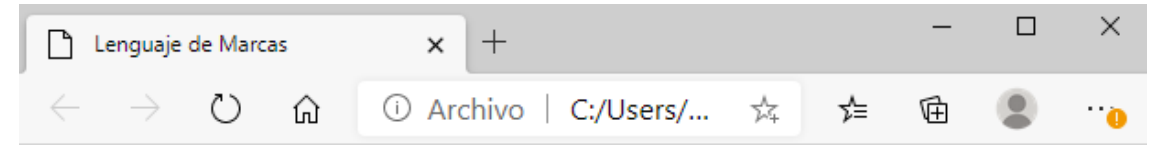
Esto es una prueba de texto

Esto es una prueba de texto

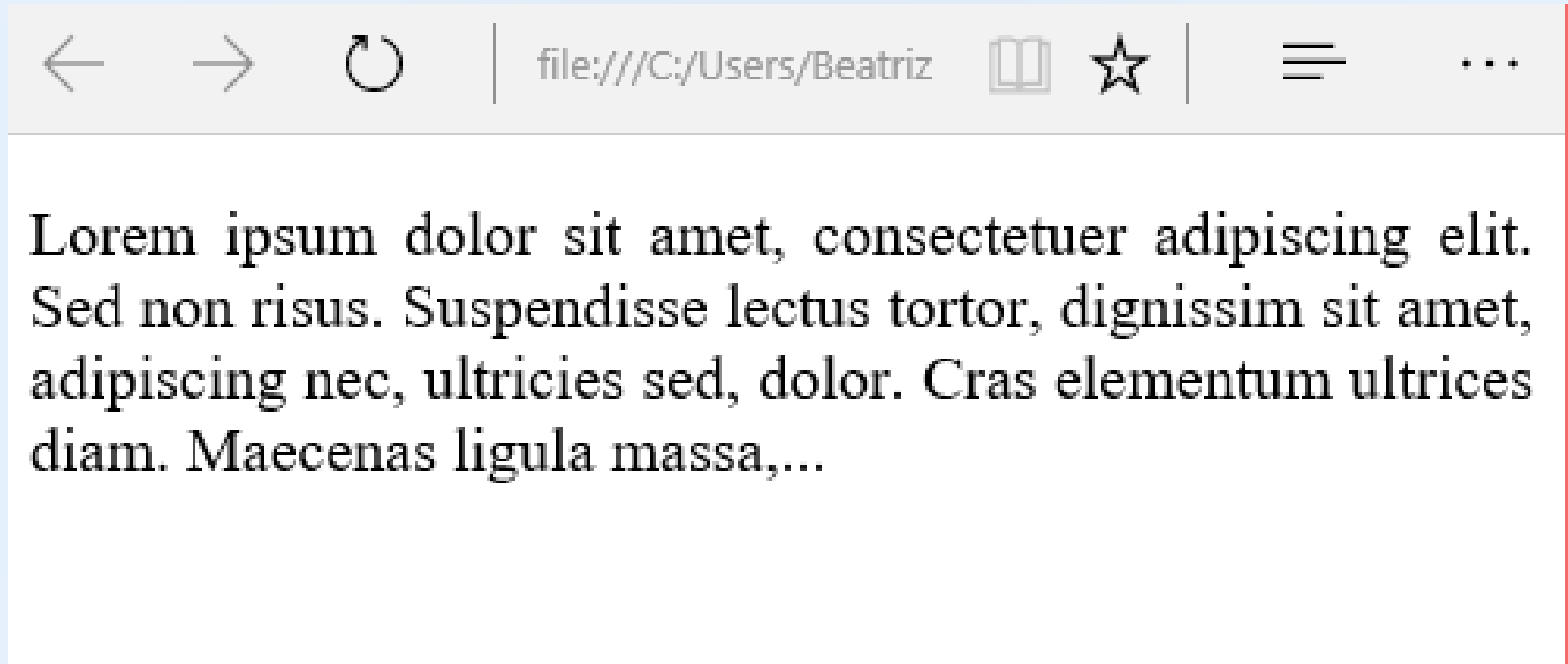
Esto es una prueba de texto

- **color**: Define el color del texto, aunque esta propiedad color puede ser usada en otros muchos elementos como, por ejemplo, líneas horizontales, elementos de un formulario o fondos.
- **text-align**: Alinea el texto, posibles valores (**center, justify, left, right**)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Lenguaje de Marcas</title>
    <style>
      #rojoDcha {
        color: red;
        text-align: right;
      }
    </style>
  </head>
  <body>
    <p id="rojoDcha">Esto es una prueba de texto</p>
  </body>
</html>
```

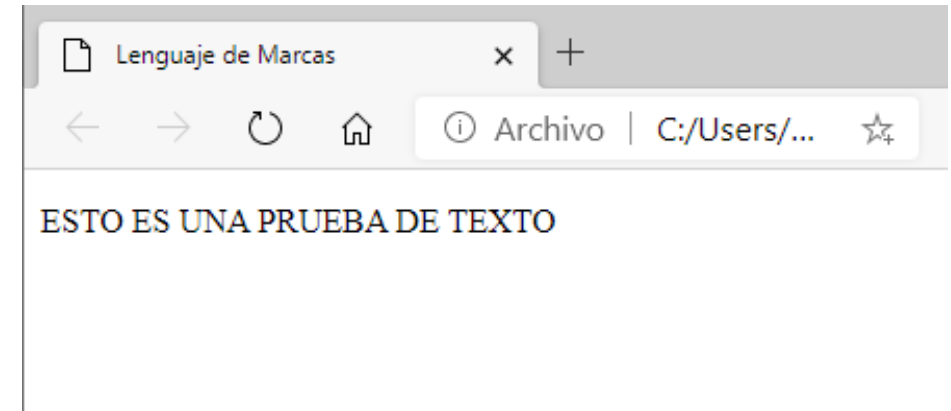


Esto es una prueba de texto

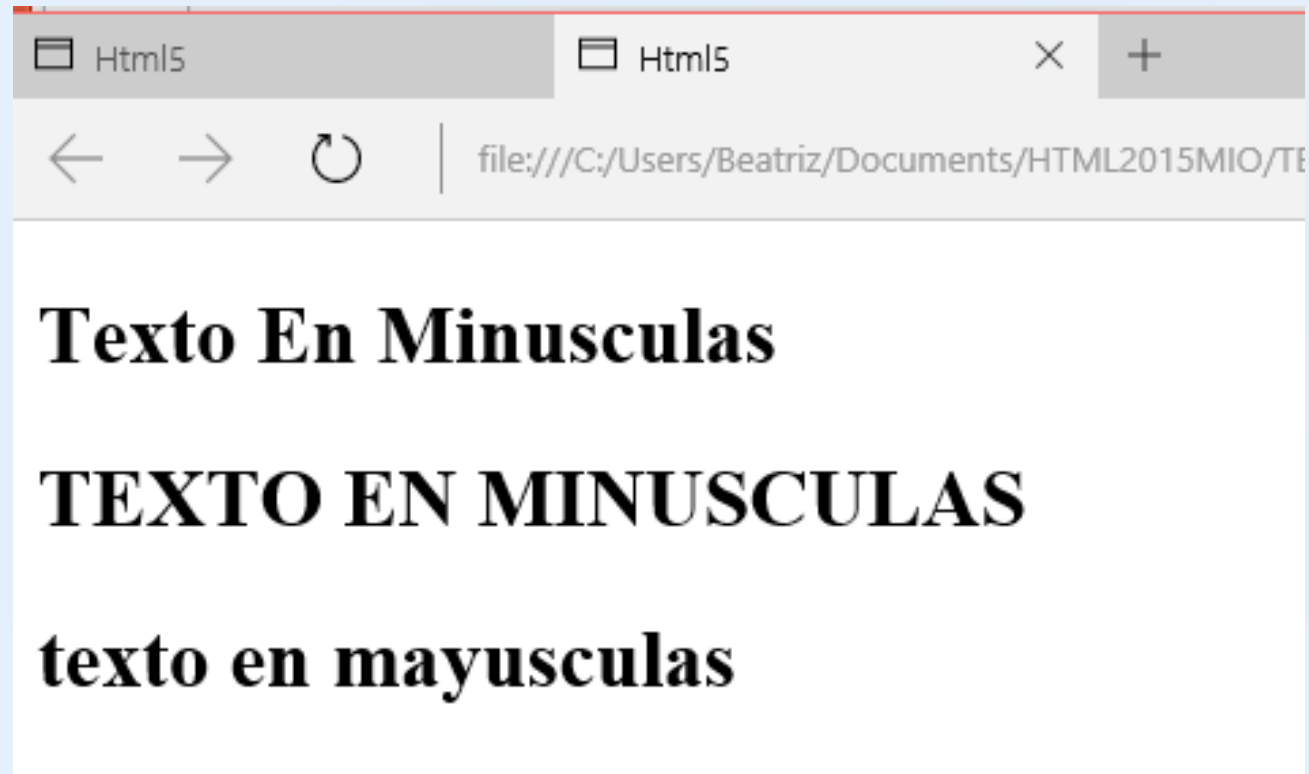
**Actividad 1.** Crea la siguiente web usando CSS

- **text-transform:** Convierte todo el texto a mayúsculas, minúsculas, etc. Los posibles valores son (**capitalize, none, lowercase, uppercase**)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Lenguaje de Marcas</title>
    <style>
      #test {
        text-transform: uppercase;
      }
    </style>
  </head>
  <body>
    <p id="test">Esto es una prueba de texto</p>
  </body>
</html>
```



## Actividad 2. Crea la siguiente web usando CSS



## Propiedades de tipo de letra.

- **font-size.** Tamaño del tipo de letra. Sus posibles valores se pueden especificar:
  - Con texto: **xx-small, x-small, small, medium, large, x-large, xx-large.**
  - Con longitud: Establece el tamaño de la fuente basado en unidades (**px, cm, etc**).
  - Con porcentaje %: Establece el tamaño de la fuente basado en un porcentaje con referencia al tamaño de la fuente del elemento padre.
  - Referenciado: Establece el tamaño en texto referenciado al padre, **initial** (pone el valor por defecto para el campo), **inherit** (hereda la propiedad de su padre), **smaller** (pone el tamaño de la fuente más pequeño que el del padre), **larger** (pone el tamaño de la fuente más grande que el del padre)
- **font-family.** Tipo o fuente de letra.
- **font-style.** Estilo de la fuente, sus posibles valores son (normal, italic, oblique)
- **font-weight.** Esta propiedad establece cómo se deben mostrar los caracteres de delgados o gruesos en el texto. (**normal, bold, bolder, lighter**)

- **font.** Si bien las propiedades de la fuente pueden definirse de forma independiente, la fuente abreviada permite agrupar, en una declaración única, las diferentes propiedades **font-style, font-variant, font-weight, font-size y font-family**. Esto representa, a efectos prácticos de escritura, una ganancia de tiempo apreciable. Un ejemplo sería:

*font: italic bold small-caps 24pt Arial, sans-serif;*

Esta declaración única se corresponde con:

*font-style: italic;*

*font-weight: bold;*

*font-variant: small-caps;*

*font-size: 24pt;*

*font-family: Arial, sans-serif;*

Los distintos atributos deben informarse seguidos y separados por un espacio. **No importa el orden** a la hora de declararlos. No es obligatorio definir cada una de las propiedades de la fuente. Para aquellas propiedades no definidas, el navegador devolverá los valores por defecto. La fuente abreviada permite a su vez definir el **interlineado**. La notación es algo particular. Se indica el **valor** del tamaño de letra (propiedad font-size), seguido de una **barra oblicua**, seguido del valor del **interlineado** (propiedad line-height). Así, por ejemplo, para **font: italic bold small-caps 24pt/1.5 Arial, sans-serif;** el tamaño de la fuente sería de 24 puntos con un interlineado de 1½.

**Actividad 3.** Crea una web que tenga el siguiente texto con el siguiente formato, únicamente usando CSS en una hoja de estilos independiente. Dentro de la etiqueta <body> de HTML solo se puede usar la etiqueta <p> y la etiqueta <span>:

Q. La ventura va guiando nuestras cosas mejor de lo que acertáramos a desear; porque ves allí, amigo *Sancho Panza*, donde se descubren treinta o ~~pocos más desaforados gigantes~~, con quien pienso hacer batalla y ~~quitarles a todos las vidas~~, con cuyos despojos comenzaremos a enriquecer, que esta es buena guerra, y es gran servicio de **Dios** quitar tan mala simiente de **sobre la faz de la tierra**.

S. ¿Qué gigantes?

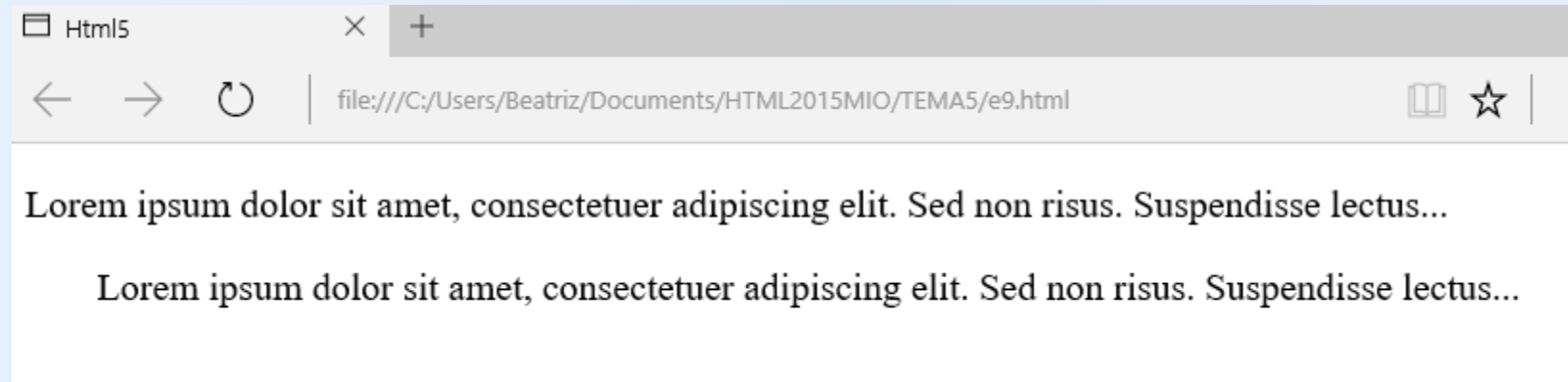
Q. ¡**Aquellos que allí ves**!, de los brazos largos, que los suelen tener algunos de casi dos **leguas**.

S. Mire vuestra merced, que aquellos que allí se parecen no son *gigantes*, sino **molinos de viento**, y lo que en ellos parecen brazos son las aspas, que, *volteadas del viento*, hacen andar la piedra del **molino**.



- **text-indent:** Indentación, puede definirse con un valor preciso (por ejemplo 20px) o bien un valor relativo en porcentaje respecto al ancho del párrafo. El valor de indentación puede ser negativo. Esto permite obtener efectos no esperados, aunque el resultado no siempre está garantizado.

## Actividad 4. Crea la siguiente web usando CSS.



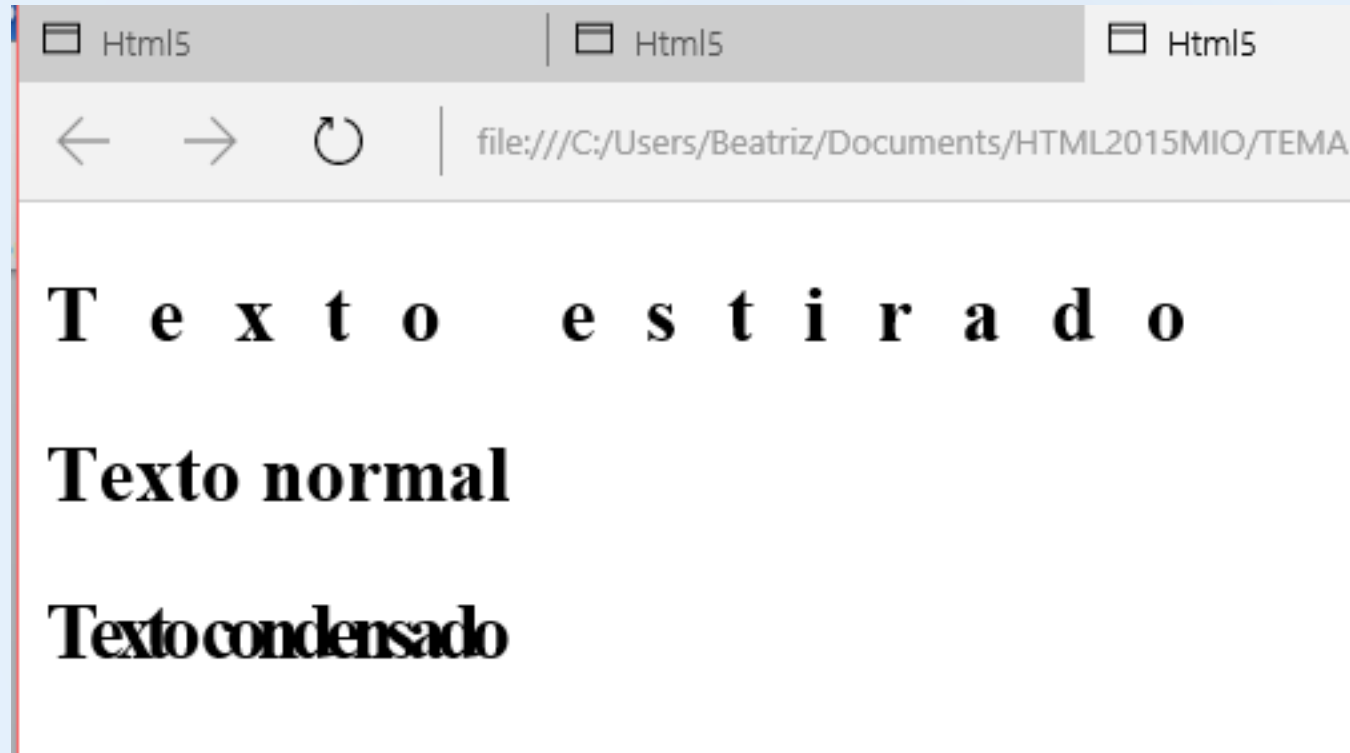
**letter-spacing:** permite variar el espacio entre los caracteres. Esta propiedad, usada sin exageración, permite mejorar la legibilidad de un texto.

Los valores de longitud deben determinarse **en valores absolutos**. Los valores relativos, por ejemplo en porcentaje, no están admitidos en esta propiedad.

Un valor positivo agrega espacio entre las letras. Un valor negativo disminuye el espacio entre las letras.

La justificación del texto exige al navegador modificar el espacio entre las letras. Un valor de letter-spacing definido a 0 impedirá la justificación del texto.

Actividad 5. Crea la siguiente web usando CSS.

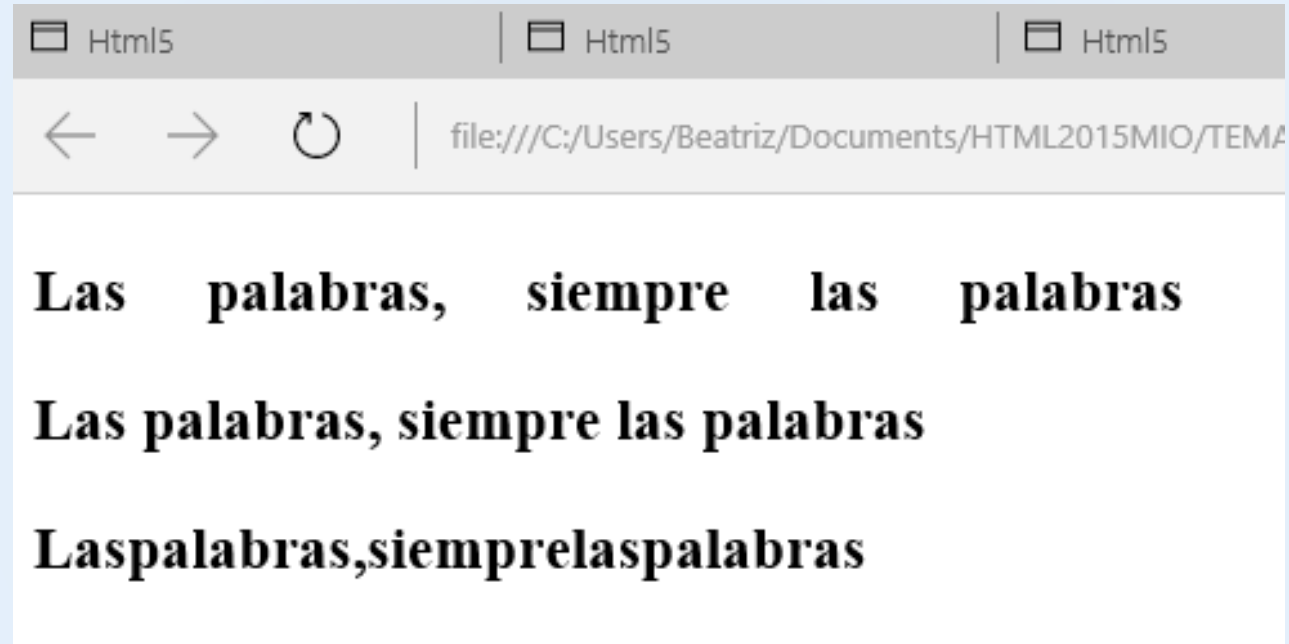


**word-spacing:** Esta propiedad, muy próxima a la anterior respecto al espacio entre letras, permite aquí variar el espacio entre palabras. Esta propiedad, usada con moderación, permite mejorar la legibilidad del texto. Los valores de longitud deben determinarse en valores absolutos. Los valores relativos, por ejemplo en porcentaje, no están admitidos en esta propiedad.

Un valor positivo agrega espacio entre las palabras. Un valor negativo disminuye el espacio entre las palabras.

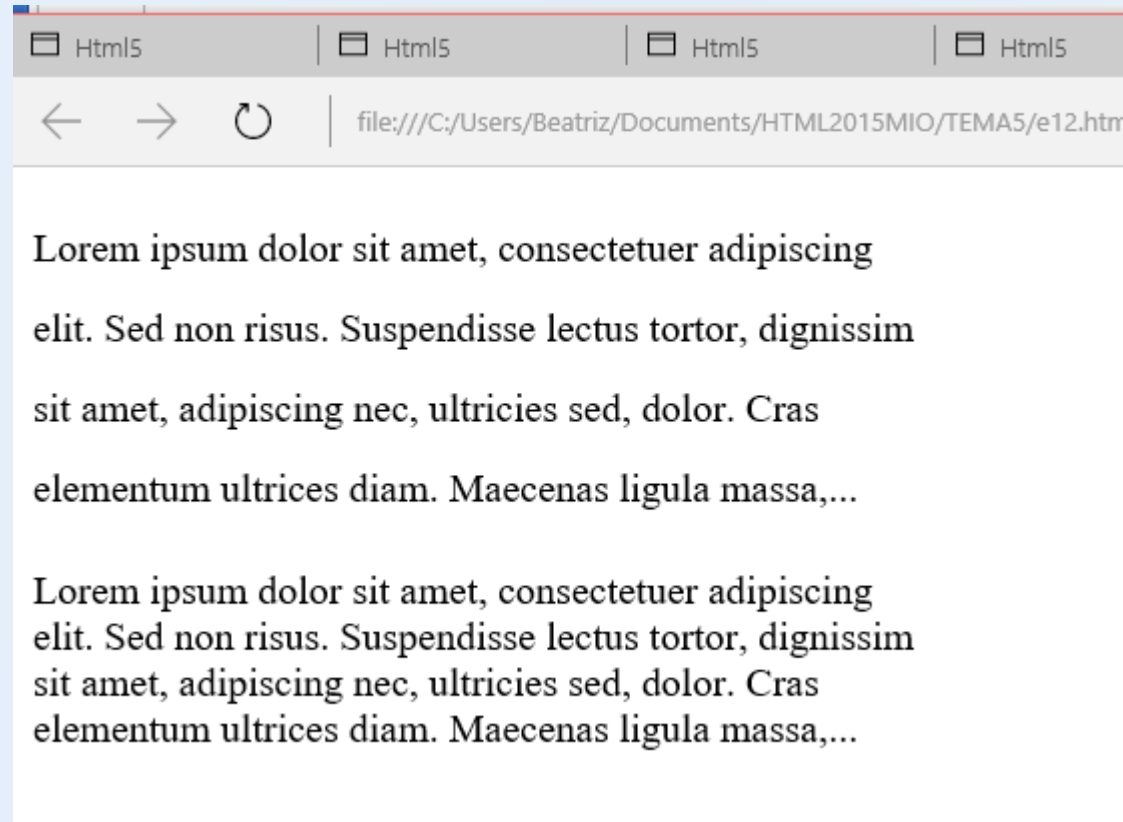
La justificación del texto permite al navegador modificar el espacio entre las palabras. Un valor de word-spacing definido a 0 impedirá la justificación del texto.

## Actividad 6. Crea la siguiente web usando CSS.



**line-height:** Esta propiedad, ofrece la posibilidad de modificar el interlineado.

- Un **número** en line-height multiplica el interlineado por defecto de la fuente. Por ejemplo el valor 2 sirve para obtener un interlineado doble.
- Un **valor de longitud**. Esto determina el interlineado sin tener en cuenta el tipo de fuente. Por ejemplo 18px.
- Un **porcentaje**. El interlineado se determina en porcentaje **respecto al interlineado normal** de la fuente. De este modo un interlineado doble se determina mediante un valor de 200%.
- El valor normal corresponde al interlineado por defecto.

**Actividad 7.** Crea la siguiente web usando CSS.



**white-space:** Esta propiedad, permite controlar **los espacios vacíos**. Sus posibles valores son:

- **pre.** Preserva los espacios múltiples y los saltos de línea.
- **nowrap.** impide los saltos de línea automáticos y muestra el texto de forma horizontal, sea cual sea el espacio disponible en el navegador. En ciertos casos el usuario tendrá que usar la barra de desplazamiento horizontal para leer la totalidad del texto.
- **normal.** Deja que el navegador gestione a su criterio los saltos de línea y los espacios múltiples.
- etc...

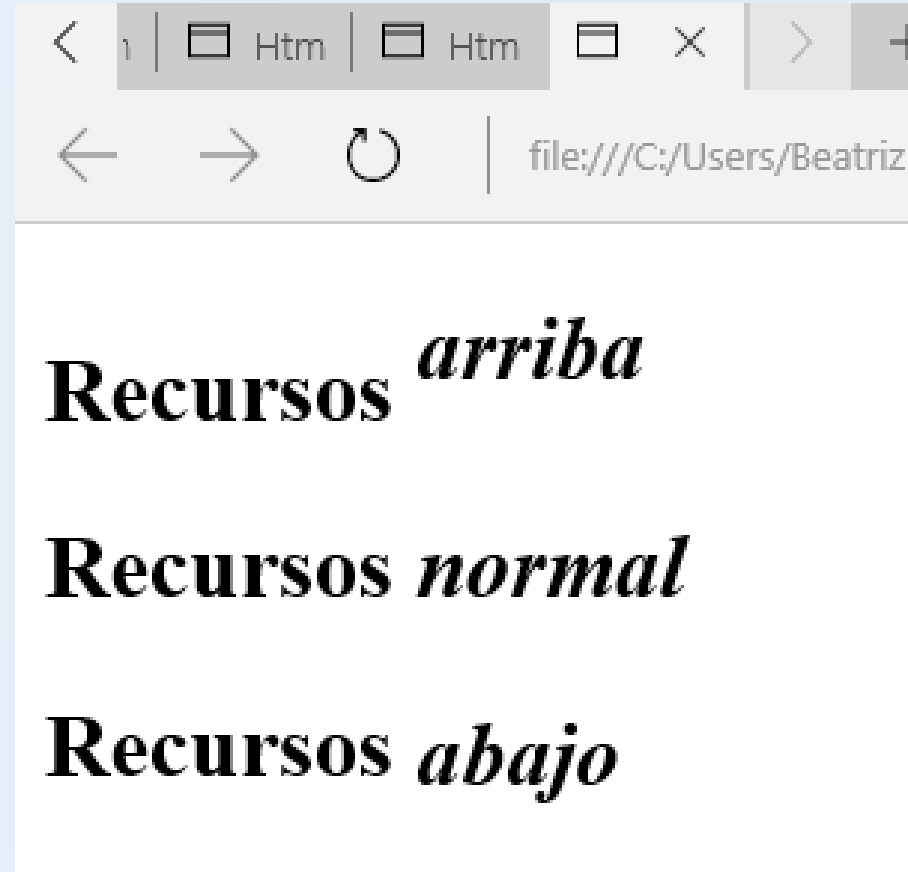
Desde sus orígenes, el lenguaje Html ignora los espacios vacíos, que agrupa en un único espacio, igual que con los saltos de línea. Los espacios vacíos múltiples y los saltos de línea se retoman en Html5 mediante la etiqueta **<pre> ... </pre>**.

**Actividad 8.** Crea la siguiente web usando CSS.

**vertical-align:** Esta propiedad determina la alineación vertical del texto respecto a otros elementos del código:

- **baseline, sub, super.** se usan respecto a una línea de texto, siendo baseline el valor por defecto para esta propiedad. El valor baseline alinea el texto respecto a la parte inferior de la línea de texto (línea de base). El valor sub escribe el texto con formato de subíndice, normalmente por debajo de la línea de base. El valor super escribe el texto con formato de superíndice o exponente, normalmente por encima de la línea de base.
- La **alineación** sin definir, **con valor de longitud** o con porcentaje deja que el navegador gestione la visualización. Un valor de longitud positivo alinea el texto por encima de la línea de base. Un valor negativo lo hace por debajo. Por ejemplo, **vertical-align: 1.7em** escribirá el texto en superíndice situado a 1.7 em. Un porcentaje positivo alinea el texto por encima de la línea de base. Un valor negativo lo hace por debajo. Por ejemplo, **vertical-align: -20%** escribirá un subíndice situado al 20%.
- Los valores **top, middle, bottom** alinean el texto respecto al elemento padre, que puede ser un texto o incluso las celdas de una tabla. El valor top realiza la alineación respecto a la parte superior del elemento padre. El valor middle realiza la alineación respecto a la parte intermedia del elemento padre. El valor bottom realiza la alineación respecto a la parte inferior del elemento padre.

**Actividad 9.** Crea la siguiente web usando CSS.

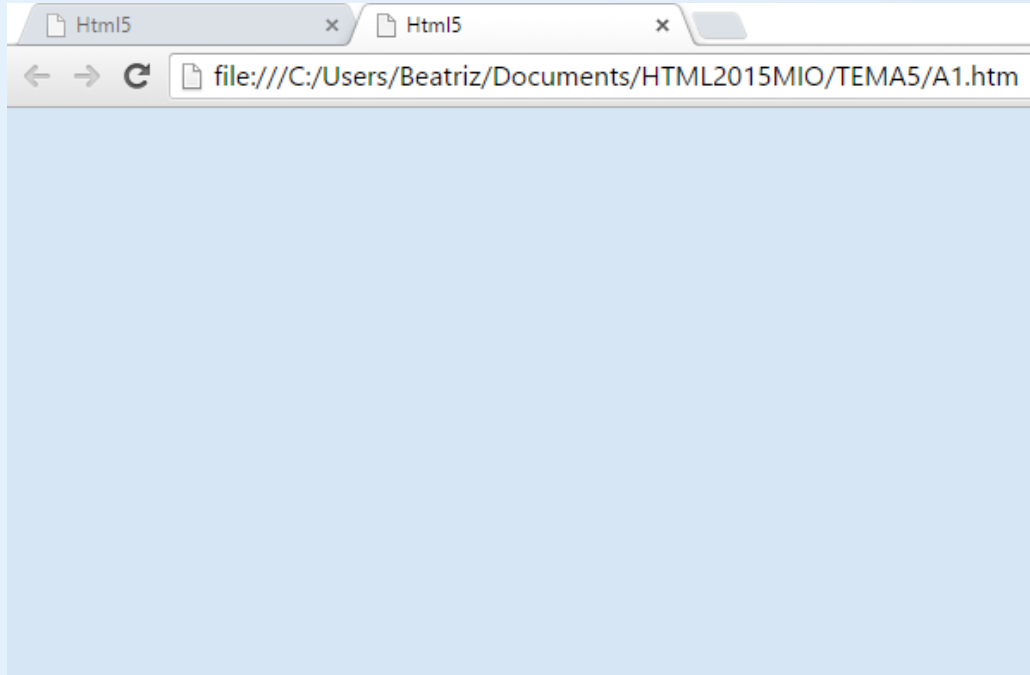


## Propiedades de fondos:

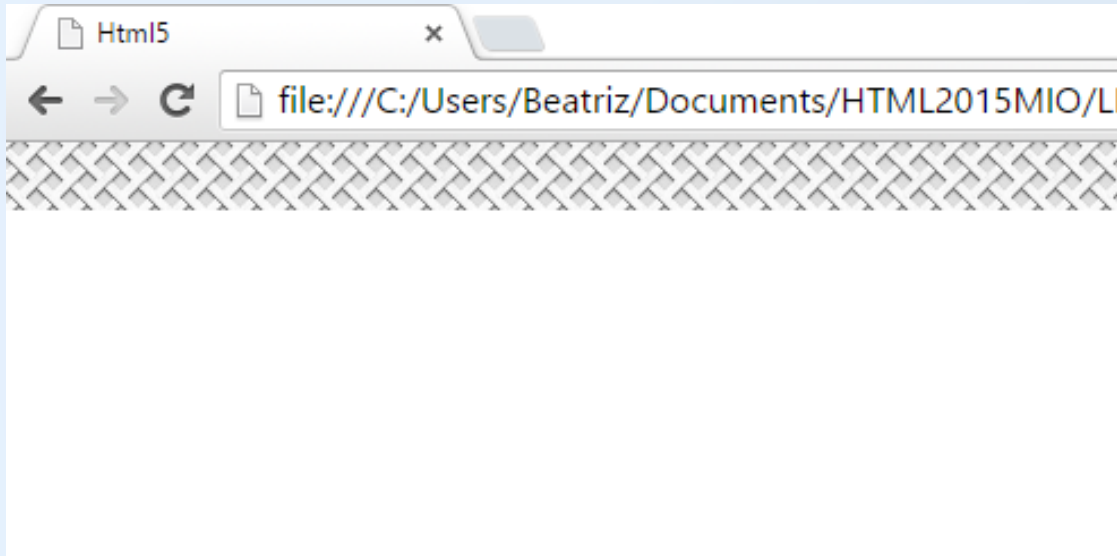
- **background-color:** Añade un color de fondo, puedes establecer como valores un color o transparent si quieres que este sea transparente.
- **background-image:** Añade una imagen de fondo, los valores posibles son none o una uri (en este caso habrá que referenciarlo mediante url y entre paréntesis el path a la imagen).
- **background-repeat:** Esta propiedad le da el control al diseñador en cuanto a la repetición de la imagen, para no tener que disponerla necesariamente en mosaico. Esta declaración de estilo sólo tiene efecto si la imagen de fondo se ha definido mediante la propiedad background-image: url(archivo\_imagen).
  - **repeat** repite la imagen horizontal y verticalmente. El resultado es el clásico, e idéntico al de la declaración background-image: url(archivo\_imagen).
  - El valor **repeat-x** repite la imagen solamente de forma horizontal (según el eje x). La imagen forma así una línea horizontal en la parte superior del elemento.
  - El valor **repeat-y** repite la imagen solamente de forma vertical (según el eje y). La imagen forma así una línea vertical a lo largo de la parte izquierda del elemento.
  - El valor **no-repeat** muestra la imagen una única vez, sin repetirla.
- **background-origin** donde empieza la repetición de imagen de fondo, dando coordenadas desde zona superior izquierda

Nada nos impide definir (mediante la propiedad background-color) un color de fondo y a la vez una imagen de fondo. Esto nos permite lograr algunos efectos muy creativos. Por otro lado, el fondo de color se mostrará en cualquier caso, incluso si no se ha podido encontrar la imagen de fondo en la ruta indicada.

### Actividad 10. Crea las siguientes webs usando CSS.



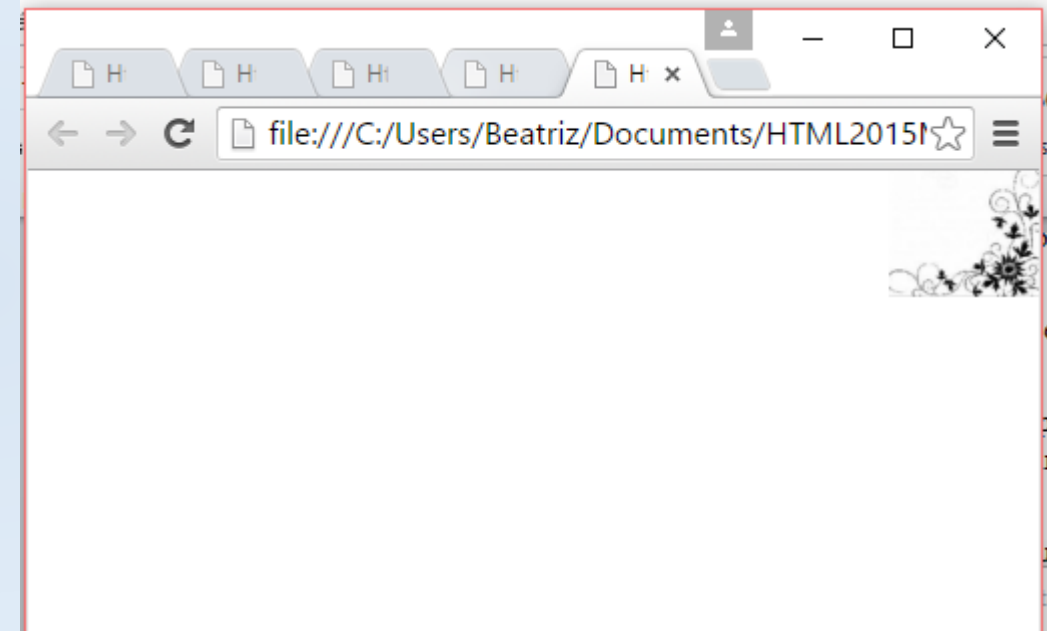
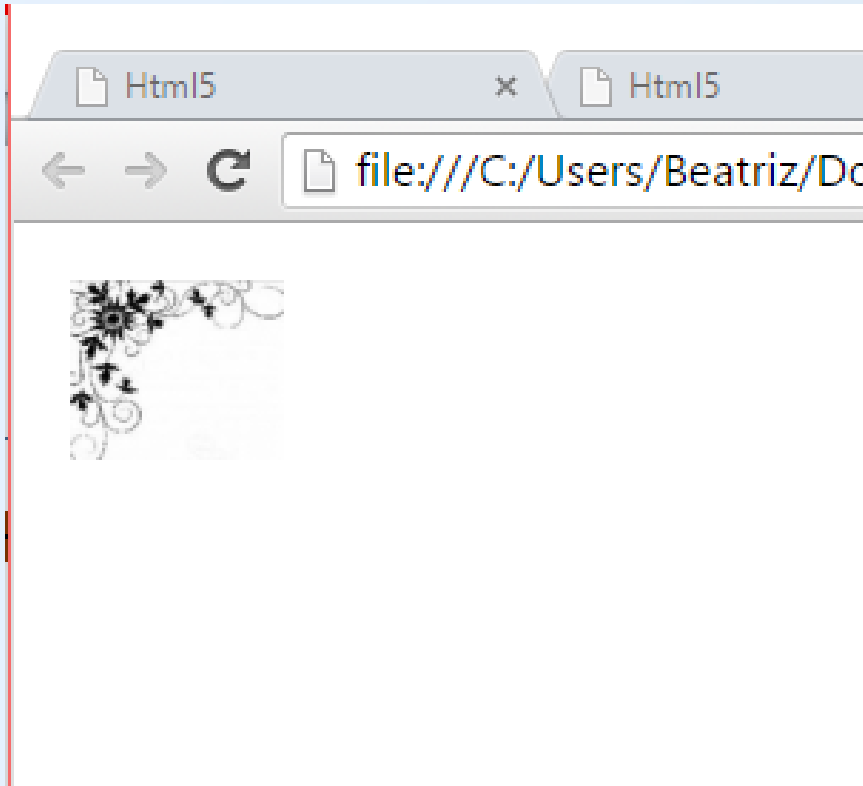
### Actividad 11. Crea las siguientes webs usando CSS.



- **background-position**: Esta propiedad nos permite posicionar con precisión la imagen de fondo de un elemento, a nivel de píxel. Esta declaración de estilo sólo tiene sentido si la imagen de fondo se ha definido mediante la declaración `background-image: url(archivo_imagen)`. Generalmente sólo se utiliza esta declaración cuando la imagen no está repetida (`background-no-repeat`).
  - El **primer** valor indica la **posición respecto al borde izquierdo** (eje horizontal) y el **segundo** valor respecto al **borde superior** (eje vertical). Ambos valores tienen que estar separados mediante un espacio.
  - Los porcentajes respetan el mismo principio aunque, evidentemente, en porcentaje respecto al tamaño del elemento padre.
  - Es posible asignar un único valor. En este caso, será interpretado como el valor horizontal (el primer valor). El valor vertical o segundo valor se definirá de forma automática al valor por defecto `center` o `50%`.
  - Es posible **combinar** los valores en **porcentaje** y en **longitud**. Por ejemplo, la declaración `background-position: 50px 30%`; es correcta.
  - Los valores **left** (izquierda), **center** (centro) o **right** (derecha) determinan la posición horizontal.
  - Los valores **top** (arriba), **center** (centro), **bottom** (abajo) determinan la posición vertical.
  - Los **valores negativos** también están aceptados.

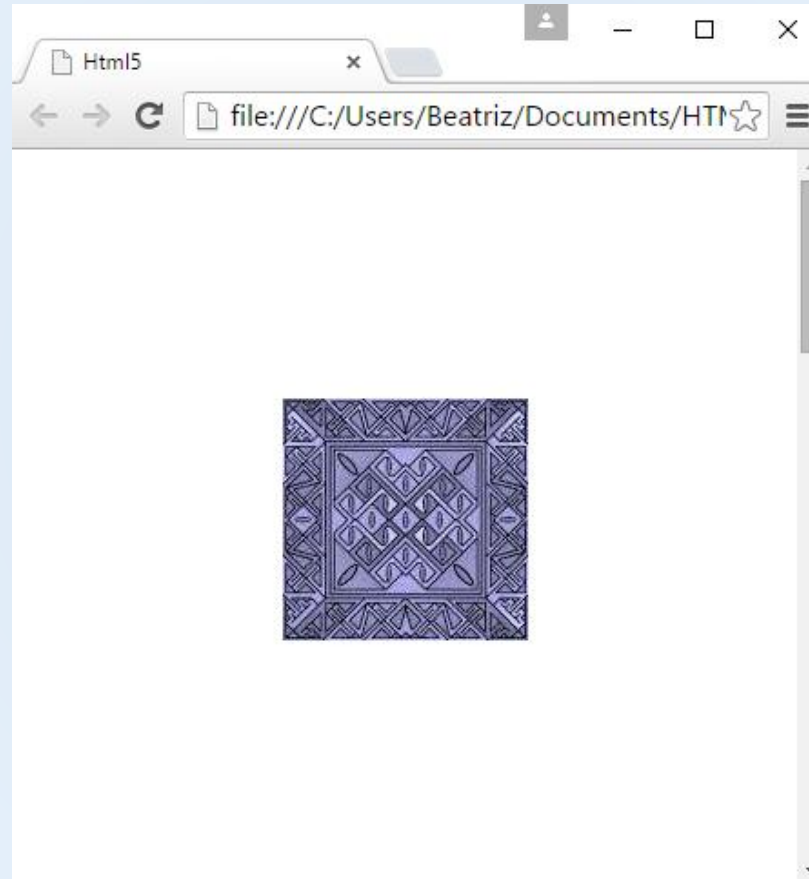


## Actividad 12. Crea las siguientes webs usando CSS.



- **background-attachment:** Esta propiedad permite fijar una imagen de fondo que permanece fija y no se mueve con el resto de la página.
  - El valor **scroll** permite a la imagen de fondo deslizarse con el contenido de la página.
  - El valor **fixed** deja la imagen de fondo fija mientras que el contenido de la página puede deslizarse.

## Actividad 13. Crea la siguiente web usando CSS.



- **background:** Esta propiedad background permite **agrupar** las distintas **propiedades** en una sola línea, incluyendo en ella todos los valores. Como las distintas propiedades y valores para definir los fondos son bastante largos de escribir de forma manual, esta sintaxis abreviada permite ganar algo de tiempo, aunque también reduce la legibilidad del código.  
**background: url(xxx.gif) white no-repeat right top fixed;** Esta declaración se corresponde con:
  - background-image: url(xxx.gif); background-color: white; background-repeat: no-repeat; background-position: right top; background-attachment: fixed;
  - Los distintos atributos deben indicarse uno tras otro separados por un espacio.
  - El orden de los atributos no tiene importancia.
  - No es obligatorio definir cada propiedad del fondo; para las propiedades no definidas el navegador mantendrá los valores por defecto.

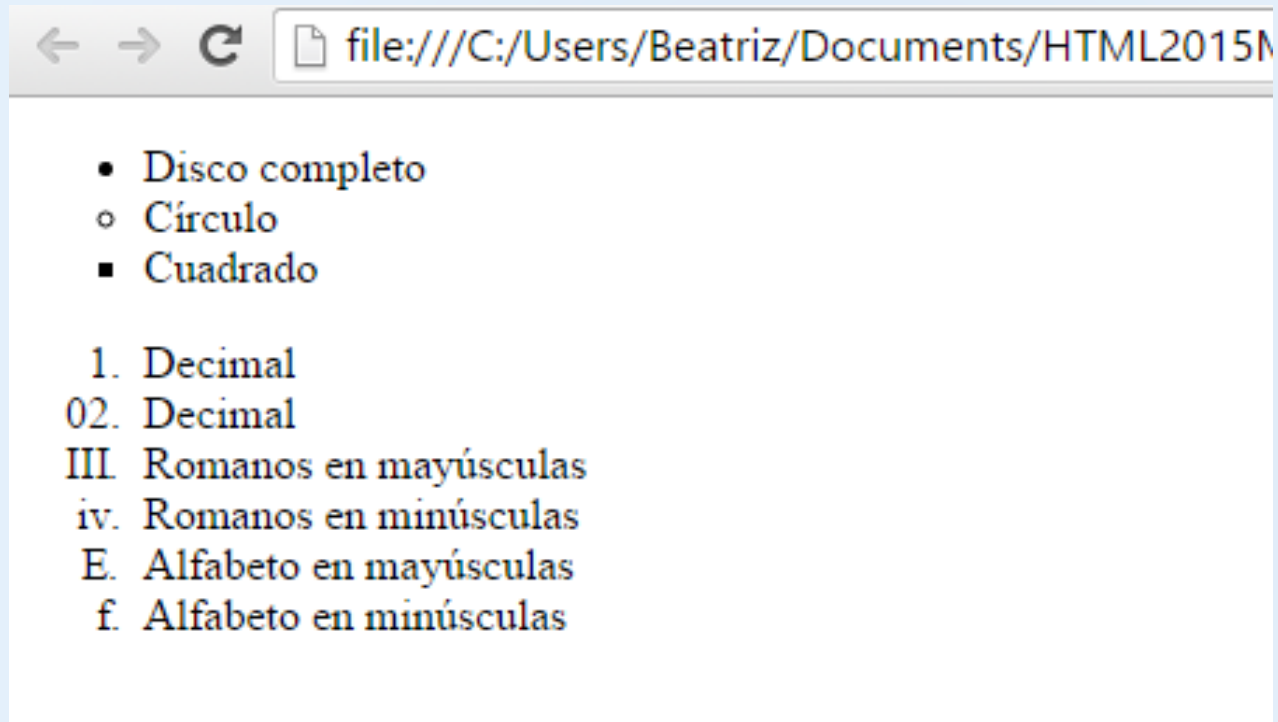
**Busca información.** Busca y estudia las diferentes posibilidades que ofrece CSS para cambiar la imagen del cursor.

[https://www.w3schools.com/cssref/pr\\_class\\_cursor.asp](https://www.w3schools.com/cssref/pr_class_cursor.asp)

## Tipo de marcadores.

- **list-style-type:** Esta propiedad permite determinar la apariencia del marcador o el estilo de numeración de la lista.
  - disc: Disco completo.
  - circle: Circulo vacío.
  - square: Cuadrado
  - decimal: Decimal, ej; 1, 2, 3, etc...
  - decimal-leading-zero: Decimal precedido de 0, ej; 01, 02, 03, etc...
  - upper-roman: Números romanos en mayúsculas ej; I, II, III, etc...
  - lower-roman: Números romanos en minúsculas ej; i, ii, iii, etc...
  - upper-alpha: Letras en mayúsculas ej; A, B, C, etc...
  - lower-alpha: Letras en minúsculas ej; a, b, c, etc...

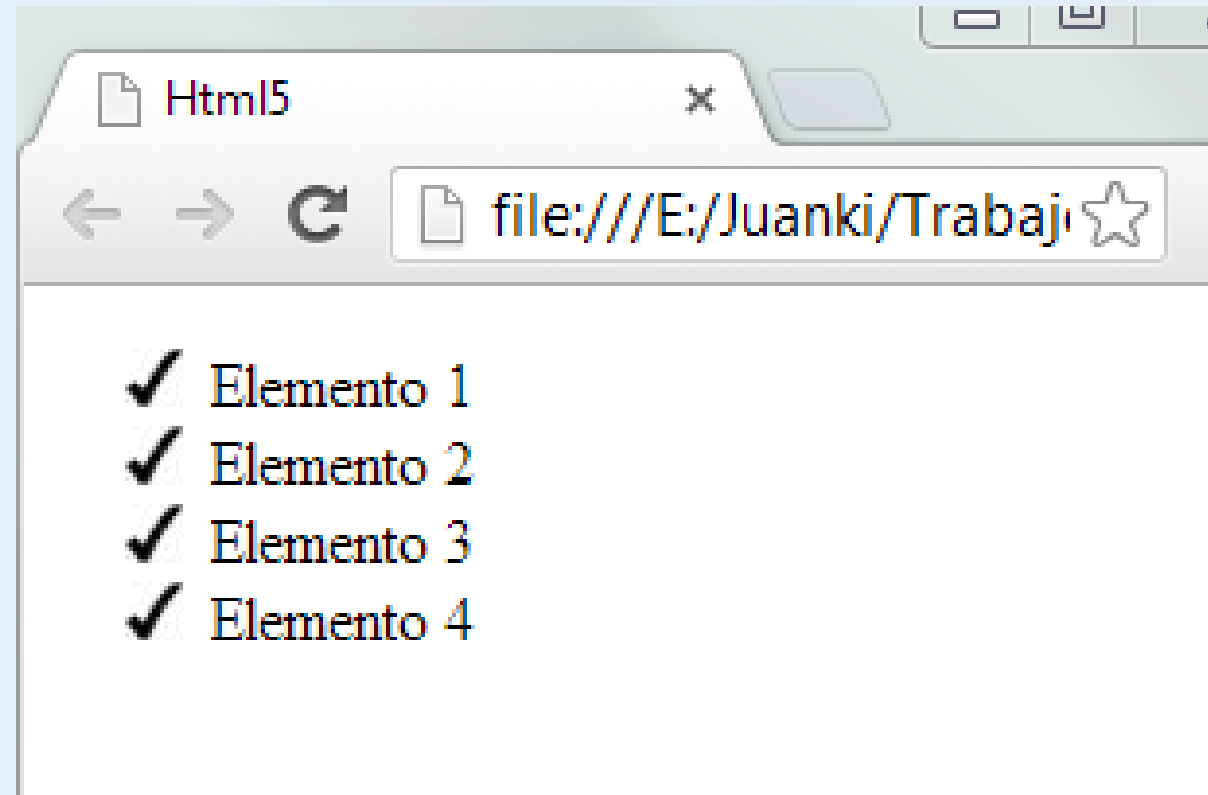
## Actividad 14. Crea la siguiente web usando CSS.



- **list-style-image:** Esta propiedad permite reemplazar los marcadores por una imagen. Esto aporta un poco de variedad respecto a los marcadores tradicionales de Html. El valor es `url(marcador_personalizado.gif)` indica la ruta (URL) de la imagen que debe utilizarse como marcador. Es posible emplear una dirección relativa hacia una imagen alojada en la misma carpeta. Las imágenes pueden tener los formatos GIF, JPEG o PNG.

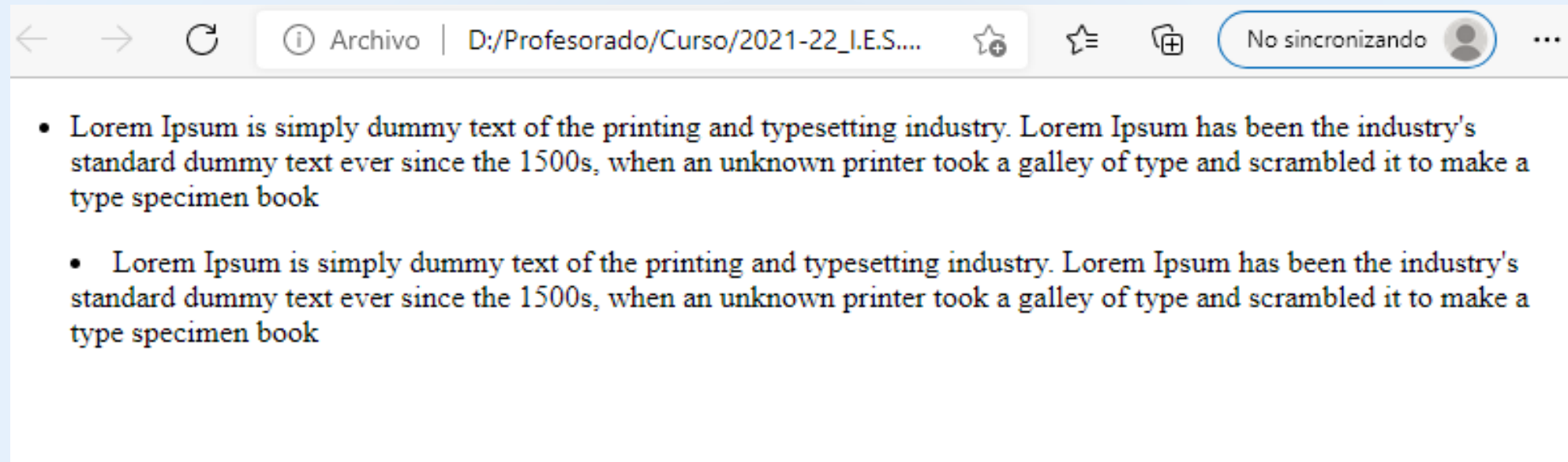


**Actividad 15.** Crea la siguiente web usando CSS.



- **list-style-position:** Esta propiedad determina la posición de la segunda línea (y las siguientes) de un elemento de lista respecto al marcador o a la numeración. Posibles valores:
  - El valor **outside** coincide con la presentación clásica de las listas con todas las líneas del elemento con un pequeño desplazamiento respecto al marcador o al número.
  - El valor **inside** es una presentación algo particular que presenta un desplazamiento solamente para la primera línea del elemento de la lista. Las demás líneas del elemento se alinean con el marcador.

## Actividad 16. Crea la siguiente web usando CSS.



- Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book
- Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book

- **list-style:** Declaración abreviada de una lista. permite retomar en una declaración de estilo única todos los valores anteriores (list-style-type, list-style-image, list-style-position). Por ejemplo list-style: url(check.gif) circle inside. Los valores se agrupan y separan por espacios.

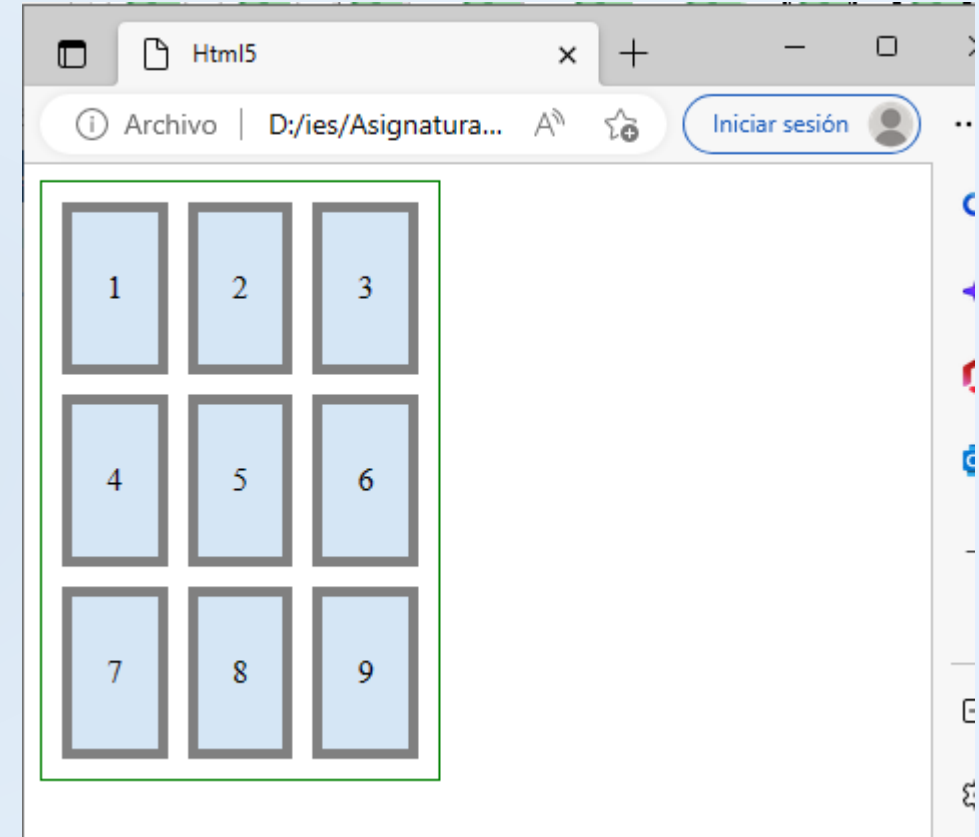
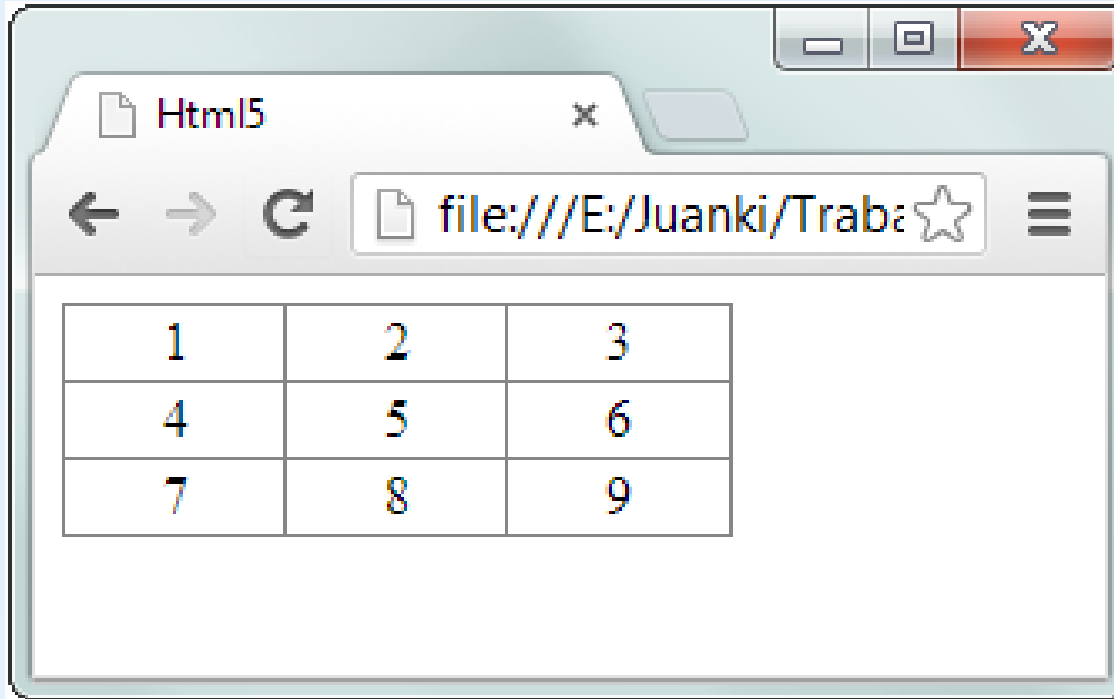
Puede parecer extraño determinar un marcador mediante una imagen y un marcador de tipo círculo. En CSS esto significa que si el navegador no encuentra el recurso correspondiente a la imagen en la ruta especificada, mostrará un marcador con forma de círculo.

Ningún valor es obligatorio, y es posible definir un solo valor.

## Tablas.

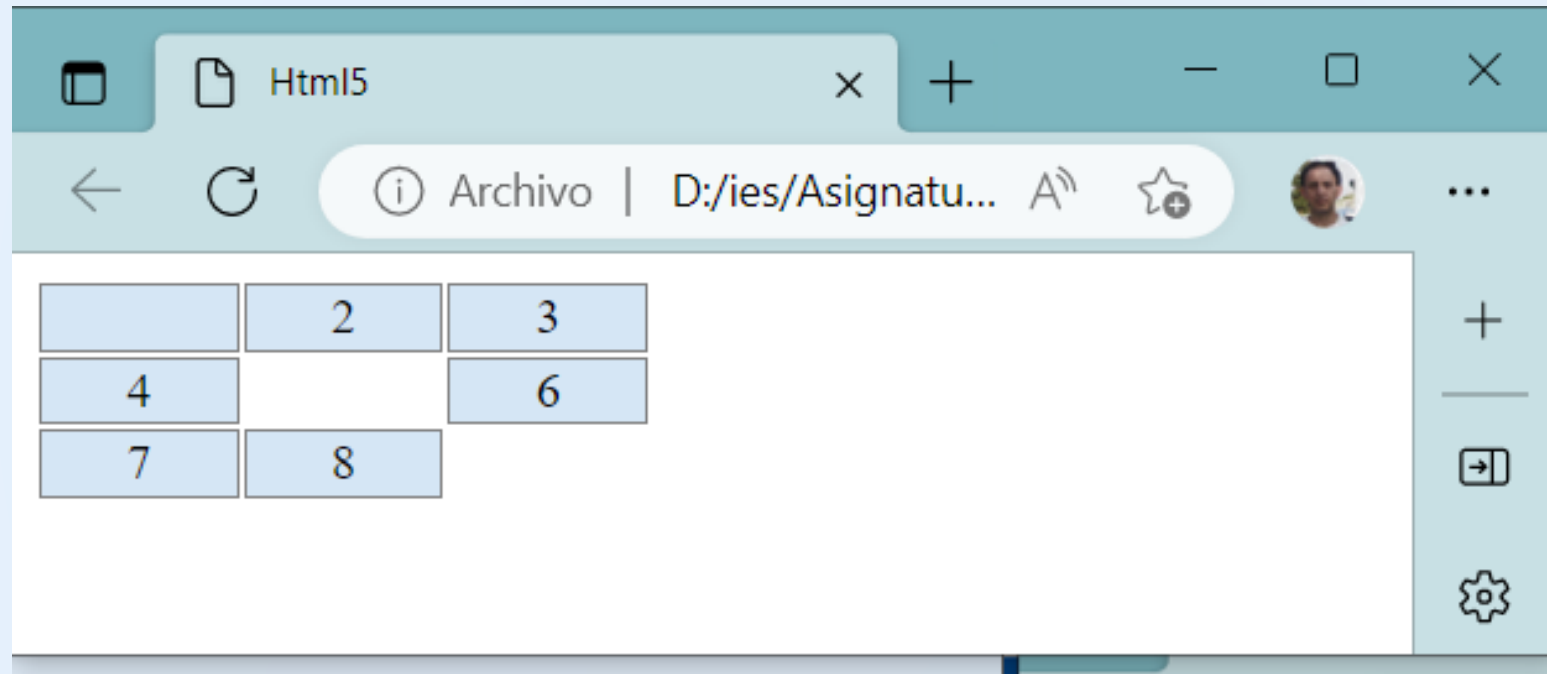
- **border-spacing:** Esta propiedad permite especificar el espaciamiento entre cada celda. Es equivalente al atributo cellpadding de Html 4.0. Es posible tener un solo borde para la tabla con border-spacing: 0px.
- **border-collapse:** Hemos visto que, por defecto, el navegador muestra dos bordes en las tablas: el borde que compone el contorno de la tabla y los bordes que componen el contorno de las celdas. La propiedad de estilo border-collapse permite gestionar esta situación
  - El valor **collapse** fusiona los dos bordes adyacentes, otorgando el aspecto de un único borde.
  - El valor **separate** muestra los dos bordes de forma separada, situación por **defecto**.

## Actividad 17. Crea las siguientes webs usando CSS.



- **empty-cells**: Esta propiedad permite determinar el comportamiento **del navegador cuando se encuentra una celda vacía** en una tabla.
  - El valor **show** muestra la celda vacía. Los bordes y el fondo también son visibles.
  - El valor **hide** no muestra la celda vacía. Los bordes y el fondo también son invisibles.

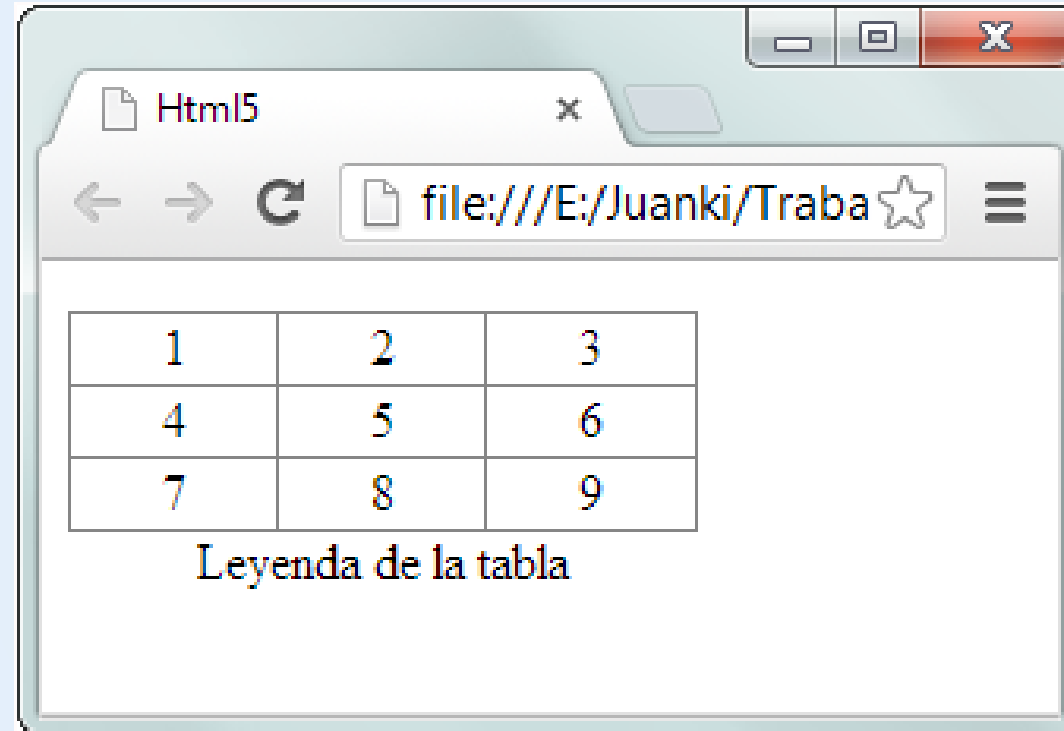
## Actividad 18. Crea las siguientes webs usando CSS.





- **caption-side**: Esta propiedad permite posicionar la leyenda en la tabla y sus dos valores son
  - El valor **top**, leyenda situada por encima de la tabla, valor por defecto.
  - El valor **bottom**, leyenda situada por debajo de la tabla.

## Actividad 19. Crea las siguientes webs usando CSS.

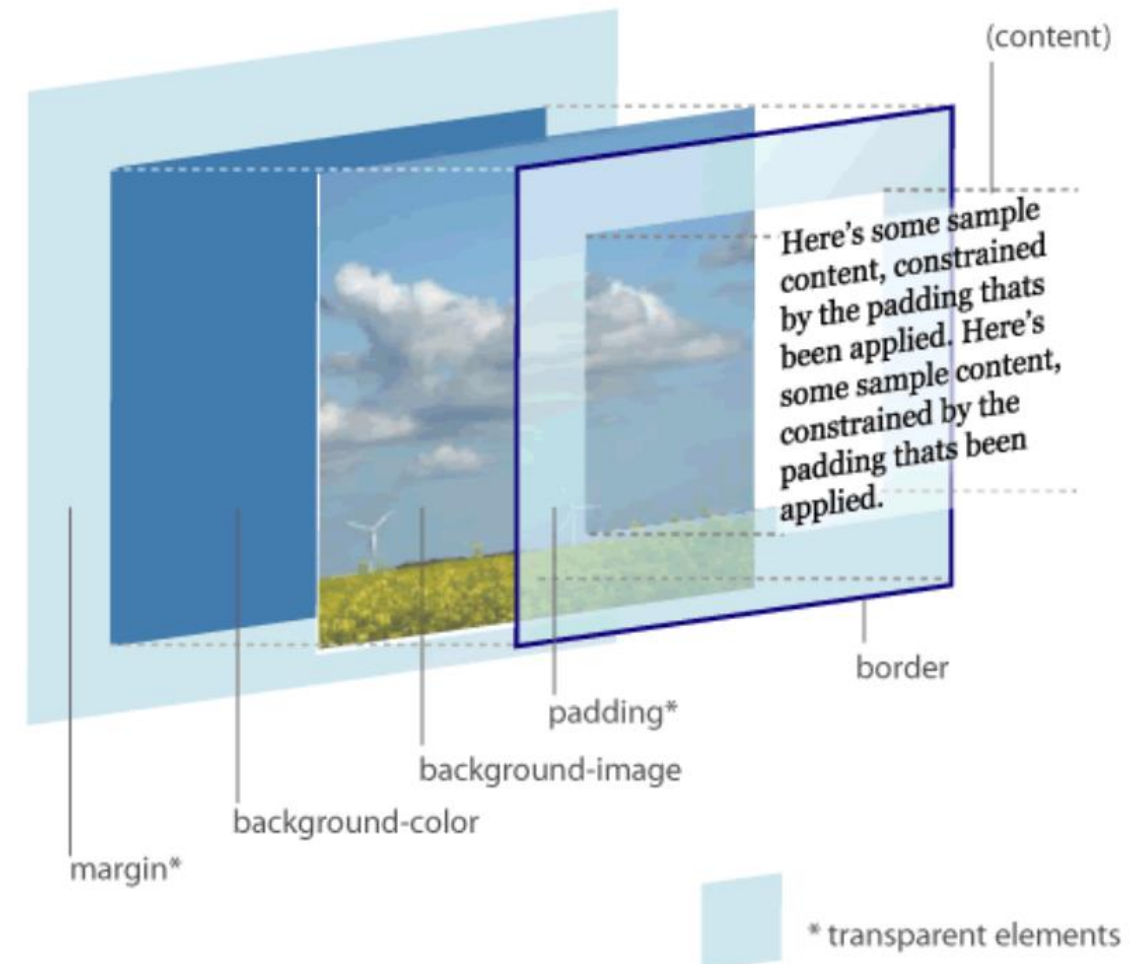
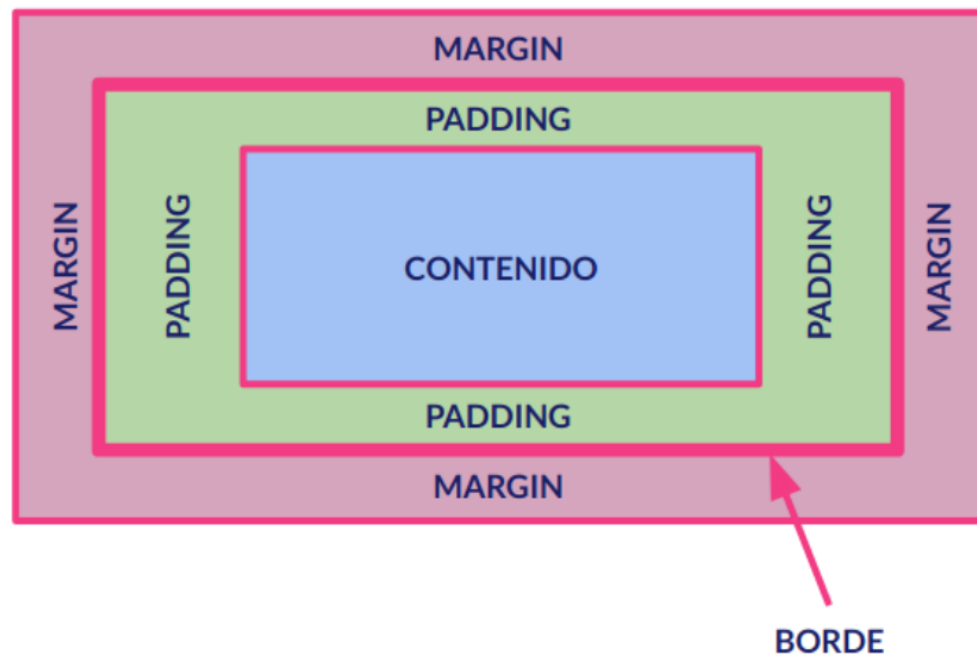


**Actividad 20.** Crea la siguiente web usando CSS.

# Modelo de cajas

---

En CSS todos y cada uno de los elementos de mi página web son cajas. Todos los elementos que representan algo en HTML tiene la siguiente estructura visual:



El W3C define un elemento caja como una zona rectangular en el que tenemos las siguientes zonas:

- El **contenido** que es lo “importante”, el texto en las etiquetas de texto, la imagen en las etiquetas de imagen etc..
- El **padding** (margen interior) que es la distancia que existe entre el contenido y el borde de la caja.
- El **borde** que es el elemento que marca la división entre el elemento y el resto de los elementos de la página.
- El **margen** (margen exterior) que es la distancia entre el elemento y el resto de los elementos de la página.

Podemos comprobar estas propiedades, y el hecho de que por defecto los navegadores les dan valor, utilizando las herramientas para desarrolladores de los mismos.

- El contenido puede ser el texto de un párrafo o el contenido de un div.
- Está rodeado por un espacio en torno al texto hasta los bordes del cuadro, que llamaremos margen interior o interno.
- A continuación están los bordes del cuadro (opcionales).
- Por último existe un espacio que envuelve todo el conjunto, llamado margen exterior o externo.
- Las hojas de estilo extienden esta noción de caja y proporcionan al diseñador un control total sobre los márgenes interior y exterior, así como del borde.

Las cajas se pueden comportar como bloques o como líneas.

- Los **elementos en bloque** son elementos que, independientemente de la anchura que tengan, se separan verticalmente de los elementos anteriores y posteriores. Es como si “provocaran” un salto de línea antes y uno después (figuradamente hablando). Ejemplos de elementos de bloque `<h1>`, `<p>`
- Los **elementos en línea** se colocan a la derecha del anterior, hasta que se llena la línea y, en ese caso, se coloca debajo de la línea actual. Ejemplo de elementos en línea las `img`

Puedes probar el comportamiento disminuyendo el tamaño de la ventana del navegador.

En las herramientas para desarrolladores puedes ver las diferentes cajas.



Con las hojas de estilo CSS podremos, por ejemplo, regular de forma independiente:

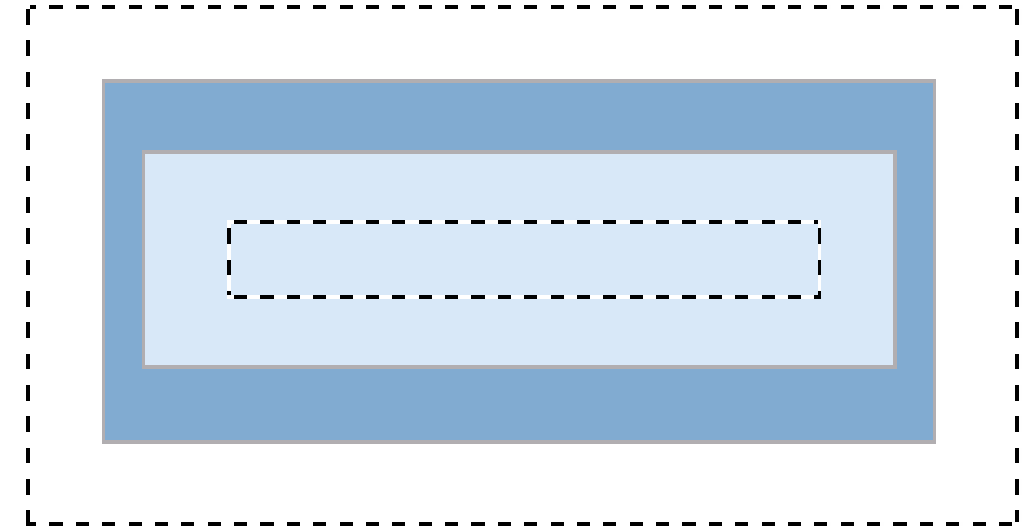
- Los cuatro márgenes exteriores en cualquier sentido.
- Los cuatro bordes (dimensión, estilo, color).
- Las cuatro márgenes interiores (padding) en cualquier sentido.
- Las dimensiones (anchura y altura) del contenido.

La dimensión total del elemento caja se calcula del siguiente modo:

contenido + márgenes interiores + bordes + márgenes exteriores.

Tomemos una caja con las propiedades siguientes:

- Anchura del contenido igual a 200 píxeles
- Margen interior de 20 píxeles
- Borde con una anchura de 10 píxeles
- Margen exterior de 20 píxeles.



La anchura del elemento caja es, en este caso, de 300 píxeles que se descomponen de la forma siguiente (de izquierda a derecha): 20 píxeles de margen exterior + 10 píxeles para el borde + 20 píxeles de margen interior + 200 píxeles de contenido + 20 píxeles de margen interior + 10 píxeles de borde + 20 píxeles de margen exterior. El total es, en efecto, 300 píxeles.

**float:** La propiedad CSS float ubica un elemento **al lado izquierdo o derecho de su contenedor, permitiendo a los elementos de texto y en línea aparecer a su costado**. El elemento es removido del normal flujo de la página, aunque aún sigue siendo parte del flujo (a diferencia del posicionamiento absoluto).

La propiedad float puede tener los siguientes valores:

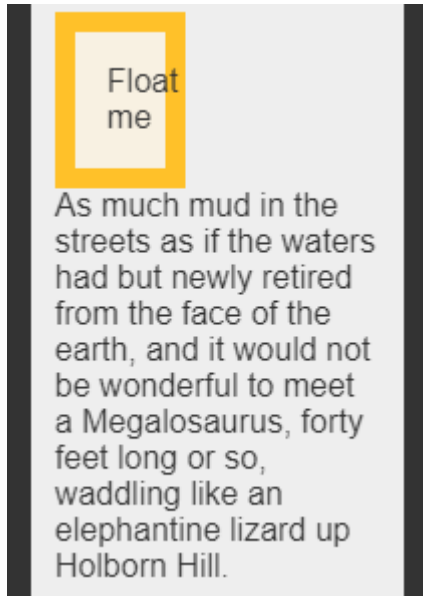
- **left** – El elemento flota a la izquierda de su contenedor
- **right** - El elemento flota a la derecha de su contenedor
- **none** – El elemento no flota (valor por defecto)
- **inherit** – Se hereda el valor de float de su padre

[https://www.w3schools.com/css/css\\_float.asp](https://www.w3schools.com/css/css_float.asp)

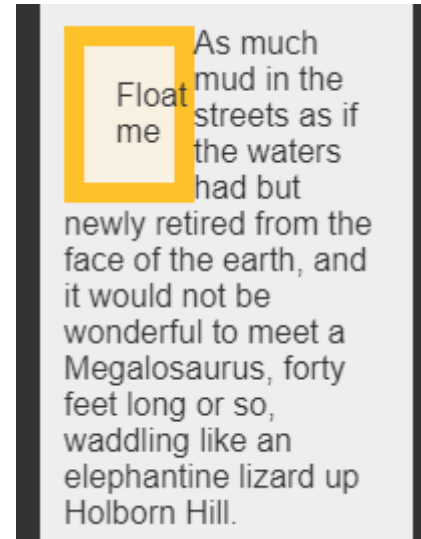
[https://www.w3schools.com/css/css\\_float\\_clear.asp](https://www.w3schools.com/css/css_float_clear.asp)

## Ejemplos

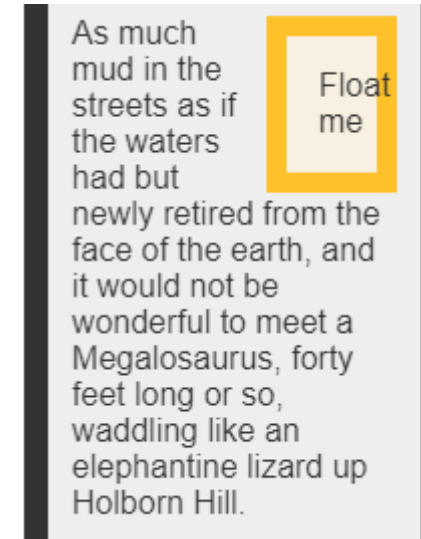
float:none



float:left



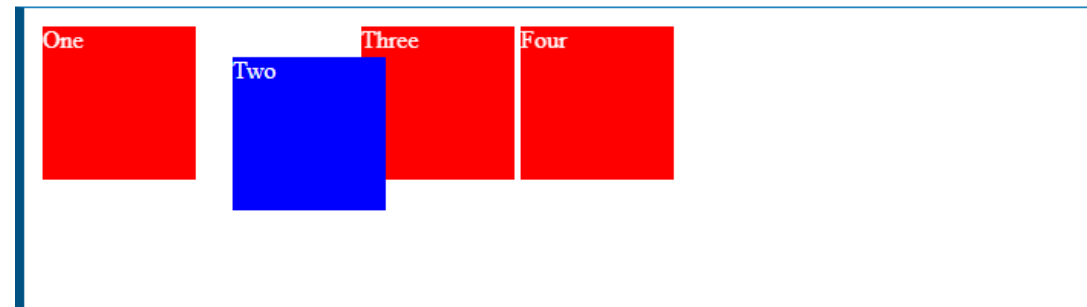
float:right



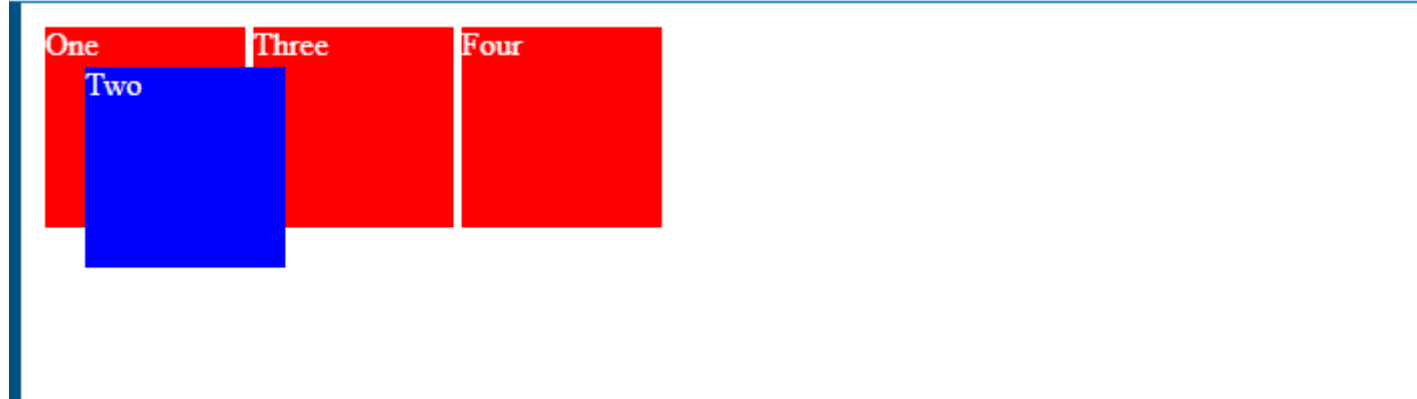
**Position.** La propiedad position de CSS especifica cómo un elemento es posicionado en el documento. La propiedad position es especificada como una palabra única elegida de la siguiente lista de valores. [https://www.w3schools.com/cssref/pr\\_class\\_position.php](https://www.w3schools.com/cssref/pr_class_position.php)

- **static.** El elemento es posicionado de acuerdo al flujo normal del documento. Las propiedades **top, right, bottom, left, and z-index no tienen efecto.** Este es el valor **por defecto.**

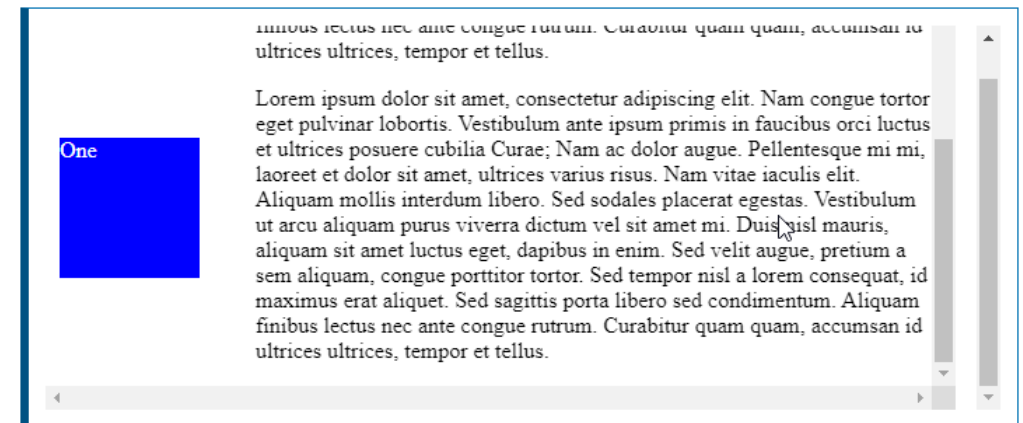
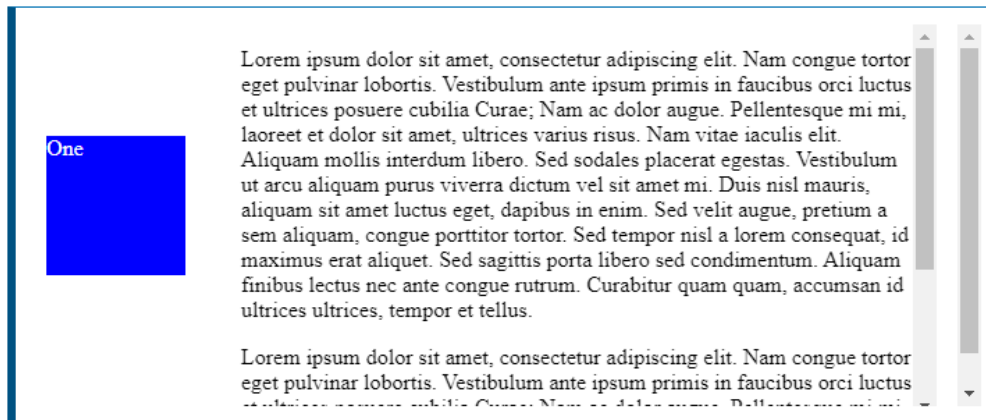
- **relative.** El elemento es **posicionado de acuerdo al flujo normal del documento**, y **luego es desplazado con relación a sí mismo**, con base en los valores de top, right, bottom, and left. El desplazamiento no afecta la posición de ningún otro elemento; por lo que, el espacio que se le da al elemento en el esquema de la página es el mismo como si la posición fuera static. Este valor crea un nuevo contexto de apilamiento, donde el valor de z-index no es auto. Elementos posicionados relativamente son desplazados una cantidad dada de su posición normal en el documento, pero sin que su desplazamiento afecte a otros elementos. En el ejemplo siguiente, nótese cómo **los demás elementos se ubican como si "Two" estuviera ocupando el lugar de su ubicación normal**.



- **absolute.** El elemento es removido del flujo normal del documento, sin crearse espacio alguno para el elemento en el esquema de la página. Es **posicionado relativo a su ancestro posicionado más cercano**, si lo hay; de lo contrario, se ubica relativo al bloque contenedor inicial. Su posición final está determinada por los valores de top, right, bottom, y left. Este valor crea un nuevo contexto de apilamiento cuando el valor de z-index no es auto. Elementos absolutamente posicionados pueden tener margen, y no colapsan con ningún otro margen.

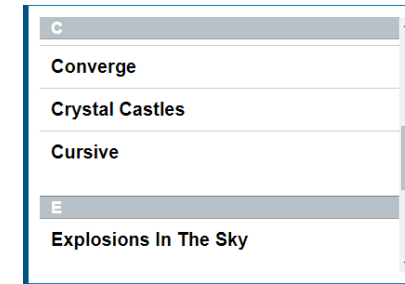
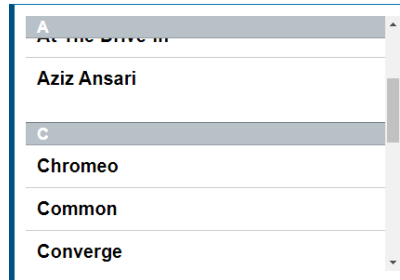
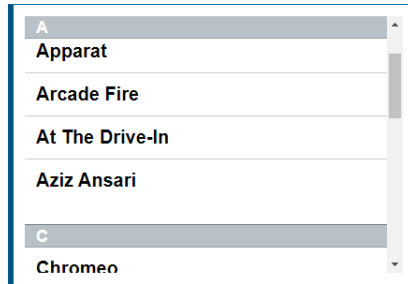


- **fixed.** El elemento es removido del flujo normal del documento, sin crearse espacio alguno para el elemento en el esquema de la página. Es posicionado con relación al bloque contenedor inicial. Su posición final es determinada por los valores de top, right, bottom, y left. Estos valores siempre crean un nuevo contexto de apilamiento. En documentos impresos, el elemento se coloca en la misma posición en cada página. El posicionamiento fijo es **similar al posicionamiento absoluto, con la excepción de que el bloque contenedor del elemento es el viewport** (área visible de la página web). Esto puede usarse para crear un **elemento flotante** que se mantiene en la misma posición independientemente del desplazamiento.





- **sticky.** (pegajoso) El elemento se posiciona según la posición de desplazamiento del usuario. El posicionamiento sticky puede considerarse un **híbrido** de los posicionamientos **relativo y fijo**. Un elemento con posicionamiento sticky es **tratado como un elemento posicionado relativamente hasta que cruza un umbral** especificado, **en cuyo punto se trata como fijo** hasta que alcanza el límite de su padre. Por ejemplo...



**left:** esta propiedad afecta la posición horizontal de un elemento posicionado. Esta propiedad no tiene ningún efecto sobre los elementos no posicionados.

- Si posición: absoluta; o posición: fija; - la propiedad establece el borde izquierdo de un elemento en una unidad a la izquierda del borde izquierdo de su antepasado posicionado más cercano.
- Si posición: relativa; - la propiedad establece el borde izquierdo de un elemento en una unidad a la izquierda de su posición normal.
- Si posición: pegajosa; - la propiedad se comporta como si su posición fuera relativa cuando el elemento está dentro de la ventana gráfica, y como si su posición fuera fija cuando está fuera.

**right:** esta propiedad afecta la posición horizontal de un elemento posicionado. Esta propiedad no tiene ningún efecto sobre los elementos no posicionados.

- Si posición: absoluta; o posición: fija; la propiedad establece el borde derecho de un elemento en una unidad a la derecha del borde derecho de su antepasado posicionado más cercano.
- Si posición: relativa; la propiedad establece el borde derecho de un elemento en una unidad a la derecha de su posición normal.
- Si posición: pegajosa; - la propiedad se comporta como si su posición fuera relativa cuando el elemento está dentro de la ventana gráfica, y como si su posición fuera fija cuando está fuera.

[CSS left property \(w3schools.com\)](http://w3schools.com)

**top:** esta propiedad afecta la posición vertical de un elemento posicionado. Esta propiedad no tiene ningún efecto sobre los elementos no posicionados.

- Si posición: absoluta; o posición: fija; - determina la distancia entre el borde superior del elemento y el borde superior del bloque que lo contiene.
- Si posición: relativa; - determina el desplazamiento hacia abajo que tendrá el elemento con respecto a su posición normal.
- Si posición: pegajosa; - la propiedad se comporta como si su posición fuera relativa cuando el elemento está dentro de la ventana gráfica, y como si su posición fuera fija cuando está fuera.

**bottom:** esta propiedad afecta la posición horizontal de un elemento posicionado. Esta propiedad no tiene ningún efecto sobre los elementos no posicionados.

- Si posición: absoluta; o posición: fija; la propiedad bottom determina la distancia entre el borde inferior del elemento y el borde inferior de su bloque contenedor.
- Si posición: relativa; la propiedad bottom determina la distancia que el elemento se mueve hacia arriba desde su posición normal.
- Si posición: pegajosa; - la propiedad se comporta como si su posición fuera relativa cuando el elemento está dentro de la ventana gráfica, y como si su posición fuera fija cuando está fuera.

El valor de las propiedades left, right, top y bottom se mide:

- Con **longitud**: Establece la posición del borde izquierdo/derecho en unidades (px, cm, etc). Se permiten valores negativos
- Con **porcentaje** %: Establece la posición del borde izquierdo/derecho en % del elemento contenedor. Se permiten valores negativos.
- **Referenciado**: Establece el tamaño en texto referenciado al padre, puede ser initial (pone el valor por defecto para el campo) o inherit (hereda la propiedad de su padre).

**display.** La propiedad de estilo display permite redefinir un elemento en línea como un elemento en bloque, y a la inversa. Los posibles valores son:

- **inline:** muestra los elementos en línea (como <div>). Acepta **margin y padding solo en horizontal**, pero las propiedades de altura y ancho no tendrán efecto. Las etiquetas span, img, b , etc. son etiquetas que por defecto son inline
- **block:** muestra los elementos en bloque, coloca un salto **de línea antes y después**. Comienza en una nueva línea y ocupa todo el ancho. Las etiquetas div, h1,h2,... p, section, nav , ul, ol tienen block por defecto.
- **none:** hace que desaparezca la caja pero no deja un espacio vacío, como si lo hace visibility: hidden
- **inline-block:** permite a los elementos comportarse como inline, pero se le pueden especificar sus parámetros de **height y width**.

Por ejemplo:

Los items de las listas son elementos de bloque. Aparecen unos debajo de otros. Mediante la propiedad display: inline, es posible mostrarlos en pantalla en una única línea.

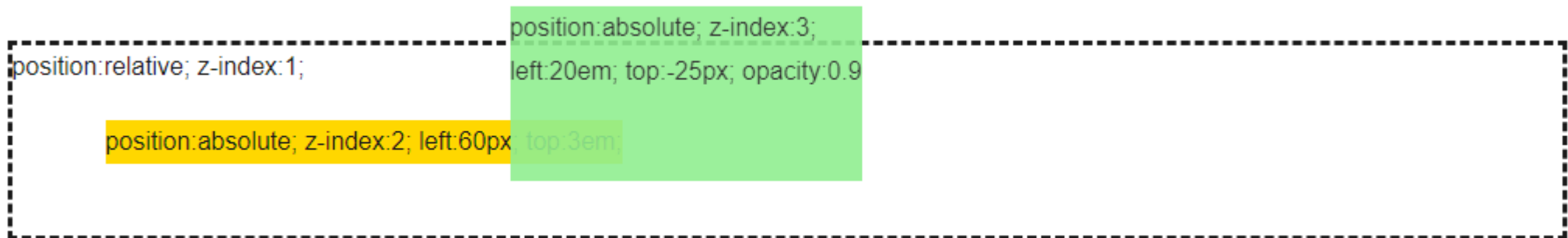
[https://www.w3schools.com/cssref/pr\\_class\\_display.php](https://www.w3schools.com/cssref/pr_class_display.php)

**z-index:** establece el apilamiento de las capas, puede ser auto o bien establecer un numero entero.

La propiedad CSS z-index indica el orden de un elemento posicionado y sus descendientes. Cuando varios elementos se superponen, los elementos con mayor valor z-index cubren aquellos con menor valor.

Para una caja posicionada (es decir, una con **cualquier position que no sea static**), la propiedad z-index especifica:

- El nivel de apilamiento en el actual contexto de apilado.
- Si la caja establece un contexto de apilamiento local.



Las propiedades de estilo `width` y `height` van a fijar, respectivamente, la **anchura y la altura del elemento caja**. Especifican la anchura y la altura del contenido (sin márgenes interiores, sin bordes y sin márgenes exteriores).

**width:** sus posibles valores son:

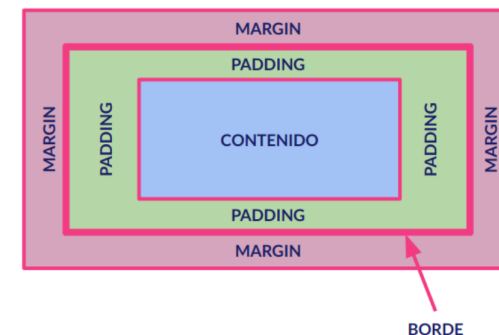
- `auto`: Ajusta automáticamente la dimensión del contenido.
- Valor de longitud: Especifica un valor fijo.
- Porcentaje: Especifica un valor relativo.

**height:** sus posibles valores son:

- `auto`: Ajusta automáticamente la dimensión del contenido.
- Valor de longitud: Especifica un valor fijo.
- Porcentaje: Especifica un valor relativo.

**margin.** Esta propiedad permite definir el margen exterior del elemento caja. Posibles valores:

- auto: Deja al navegador la gestión de los márgenes exteriores.
- Valor de longitud: Define con precisión los márgenes exteriores.
- Un porcentaje: Define la longitud respecto a un elemento padre.



El margen exterior de cada lado del elemento caja puede definirse también, con `margin-top`, `margin-right`, `margin-bottom` y `margin-left`. Las direcciones `top`, `right`, `bottom`, `left` para los bordes superior, derecho, inferior e izquierdo. Si se ponen los 4 valores se toma en **sentido horario**: arriba izquierda, arriba derecha, abajo derecha, abajo izquierda.

```
selector { margin: 10px 20px 30px 40px }
```

Si se ponen 2 se toma el primero para arriba y abajo y el 2º para derecha e izquierda

```
margin: 10px 20px 30px 40px
```



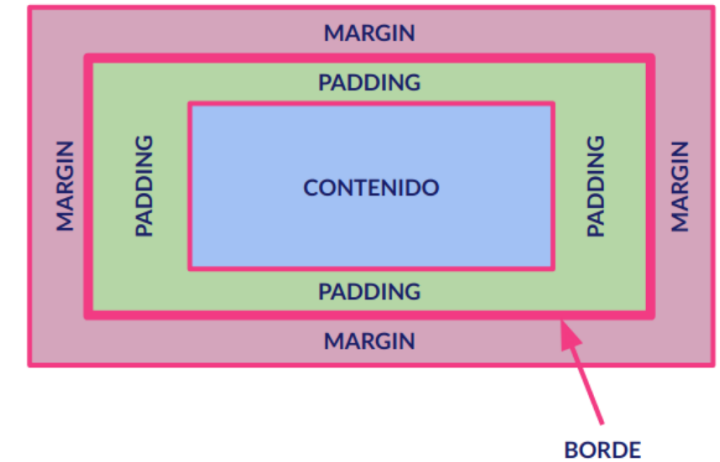
Es posible abreviar la escritura de las propiedades del margen externo **margin-top**, **margin-right**, **margin-bottom** y **margin-left**, informado uno, dos, tres o cuatro valores mediante la propiedad **margin**.

- Si se informa un único margen, se aplica a los cuatro lados.
- Con dos márgenes, el primero se aplica a los lados superior e inferior. El segundo se aplica a los laterales (márgenes derecho e izquierdo).
- Con tres valores, el primero se aplica al lado superior, el segundo a los márgenes laterales y el tercero al lado inferior.
- Con cuatro márgenes, éstos se aplican en el sentido de las agujas del reloj partiendo del lado superior, es decir superior, derecho, inferior e izquierdo. `margin: 10px, 20px, 30px, 40px`

Los márgenes pueden definirse con valores negativos, lo que puede generar efectos visuales interesantes, aunque también existe el riesgo de que se interpreten de forma distinta según cada navegador.

**padding.** Esta propiedad permite definir el margen interior del elemento caja. Los posibles valores son:

- **auto:** Deja al navegador la gestión de los márgenes exteriores.
- **Valor de longitud:** Define con precisión los márgenes exteriores.
- Un **porcentaje:** Define la longitud respecto a un elemento padre.



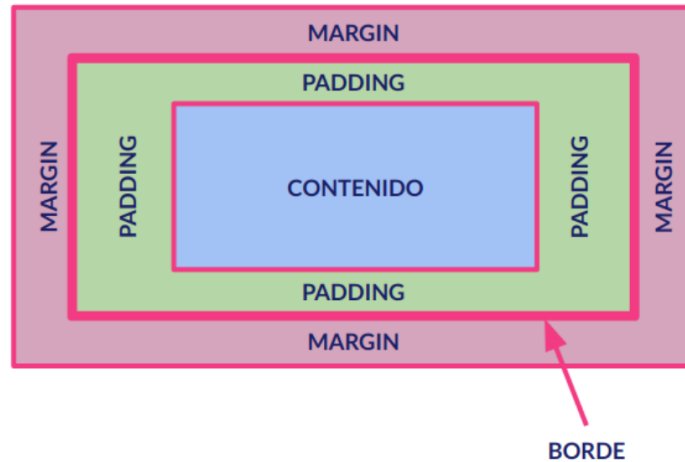
El margen exterior de cada lado del elemento caja puede definirse, igualmente, con ayuda de las propiedades **padding-top**, **padding-right**, **padding-bottom** y **padding-left**. Las direcciones top, right, bottom, left diseñan, respectivamente, los bordes superior, derecho, inferior e izquierdo.

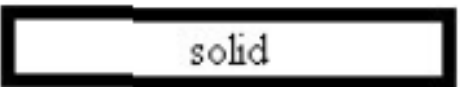
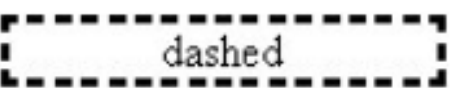
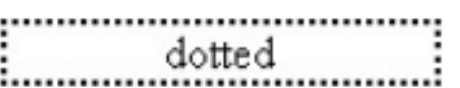
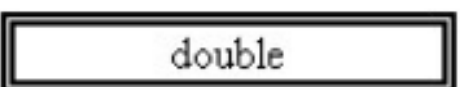


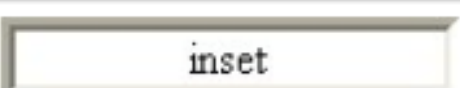
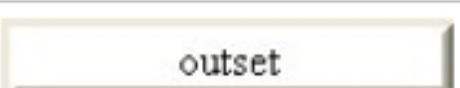
Es posible abreviar la escritura de las propiedades del margen interno padding-top, padding-right, padding-bottom y padding-left, informado uno, dos, tres o cuatro valores mediante la propiedad **margin**.

- Si se informa un **único padding**, se aplica a los **cuatro lados**.
- Con **dos padding**, el primero se aplica a los lados **superior e inferior**. El segundo se aplica a los **laterales** (márgenes derecho e izquierdo).
- Con **tres valores**, el primero se aplica al lado **superior**, el segundo a los márgenes **laterales** y el tercero al lado **inferior**.
- Con **cuatro padding**, éstos se aplican en el **sentido de las agujas del reloj** partiendo del lado superior, es decir superior, derecho, inferior e izquierdo.

**border-style**. Esta propiedad CSS aporta cierta variedad a los bordes respecto al estilo único de los bordes de Html. Sus posibles valores son:

- **solid**: un trazo liso.
- **dashed**: guiones.
- **dotted**: pequeños puntos.
- **double**: trazos lisos dobles.
- **groove**: efecto 3D de relieve incrustado en la página (opuesto a ridge).
- **ridge**: efecto 3D de relieve saliendo de la página (opuesto a groove).
- **inset**: bordes entrantes, incrustados en la página (inverso de outset).
- **outset**: bordes salientes, biselados de la página (inverso de inset).
- **hidden**: sin borde (influye sobre el borde adyacente).
- **none**: equivalente a border-width: 0px o, lo que es lo mismo, sin borde.



solid	
dashed	
dotted	
double	
groove	
ridge	
inset	
outset	

El estilo de **borde no se mostrará** si no se ha definido el grosor del borde (**border-width**).

El **estilo de cada lado** puede igualmente **definirse de forma individual** gracias a las propiedades **border-top-style, border-right-style, border-bottom-style y border-left-style**. Las direcciones top, right, bottom, left informan, respectivamente, los bordes superior, derecho, inferior e izquierdo.

**border-width:** Esta propiedad permite definir el grosor de los cuatro bordes de forma simultánea o de cada borde de forma individual. Los valores posibles son:

- Valor de **longitud**
- El valor **thin** se traduce en un borde fino,
- El valor **medium** en un borde medio
- El valor **thick** en un borde grueso.

La interpretación del grosor, a partir de las palabras clave, puede variar de un navegador a otro.

Es posible definir el **grosor** de cada lado de forma **individual** gracias a las propiedades **border-top-width**, **border-right-width**, **border-bottom-width** y **border-left-width**. Las direcciones top, right, bottom, left informan, respectivamente, los bordes superior, derecho, inferior e izquierdo.

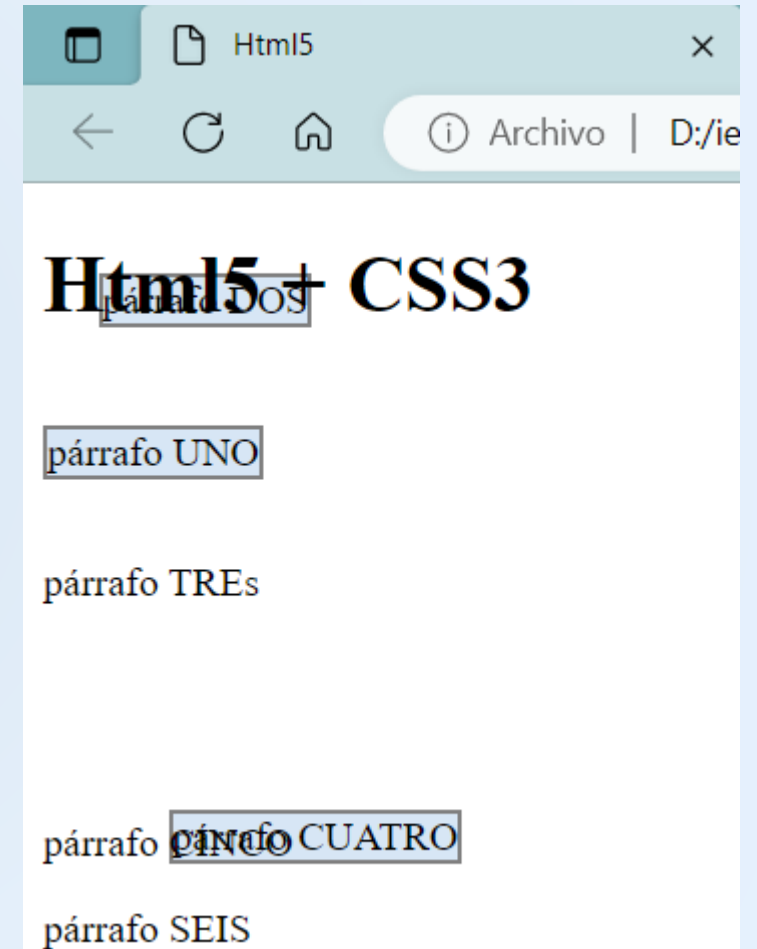
**border-color:** Esta propiedad de estilo permiten definir el color de los cuatro bordes del elemento caja o de cada borde tomado individualmente. El **color no puede aplicarse si no se han definido** el tipo de borde (**border-style**) ni el grosor del mismo (**border-width**), sin los cuales el borde no existe. Los posibles valores son:

- Nombre del color.
- Notación hexadecimal.
- Notación en formato RGB con valores o porcentajes.
- Notación en formato RGBa
- Notación en formato HSL y HSLa
- transparent

Los colores de cada lado del elemento caja pueden **definirse individualmente** gracias a las propiedades **border-top-color, border-right-color, border-bottom-color y border-left-color**. Las direcciones top, right, bottom, left informan respectivamente los bordes superior, derecho, inferior e izquierdo.

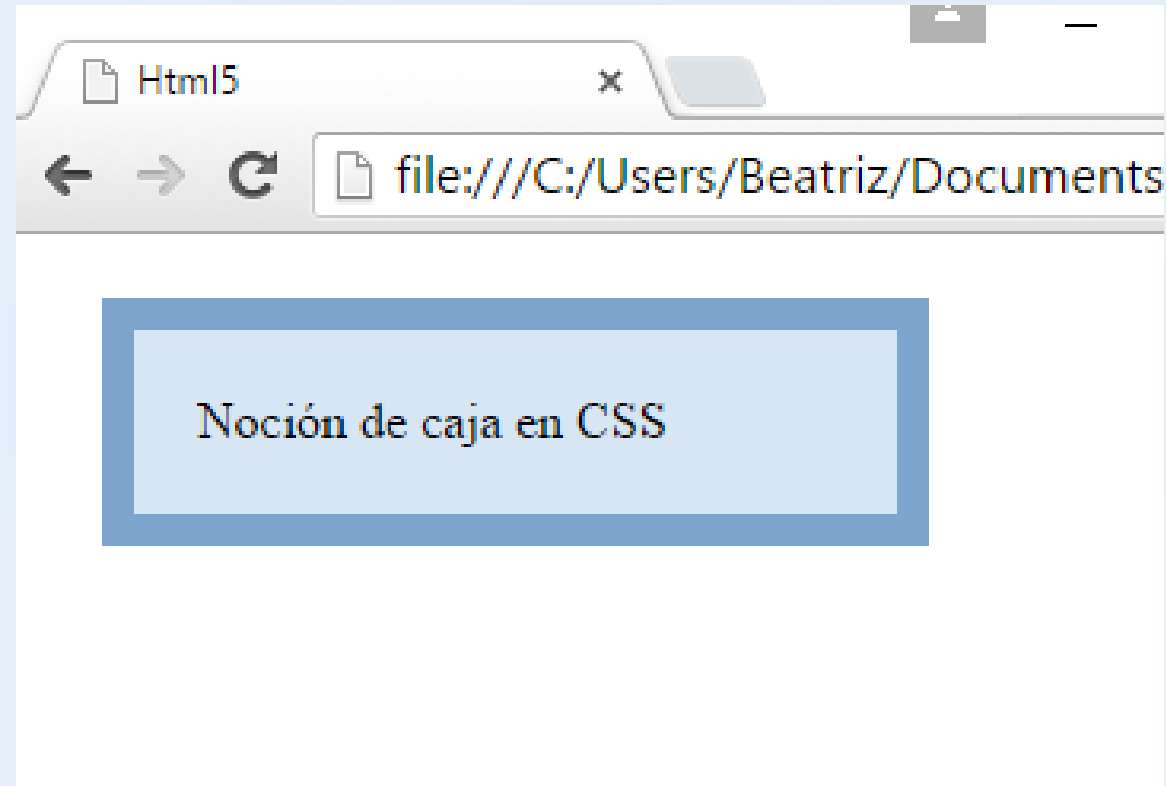
**Actividad 21A.** Crea la siguiente web usando CSS con el siguiente contenido de html:

```
<body>  
  <h1>Html5 + CSS3</h1>  
  <p class="uno">párrafo UNO </p>  
  <p class="dos">párrafo DOS </p>  
  <p> párrafo TRES </p>  
  <p class="cuatro">párrafo CUATRO </p>  
  <p>párrafo CINCO </p>  
  <p>párrafo SEIS </p>  
</body>
```





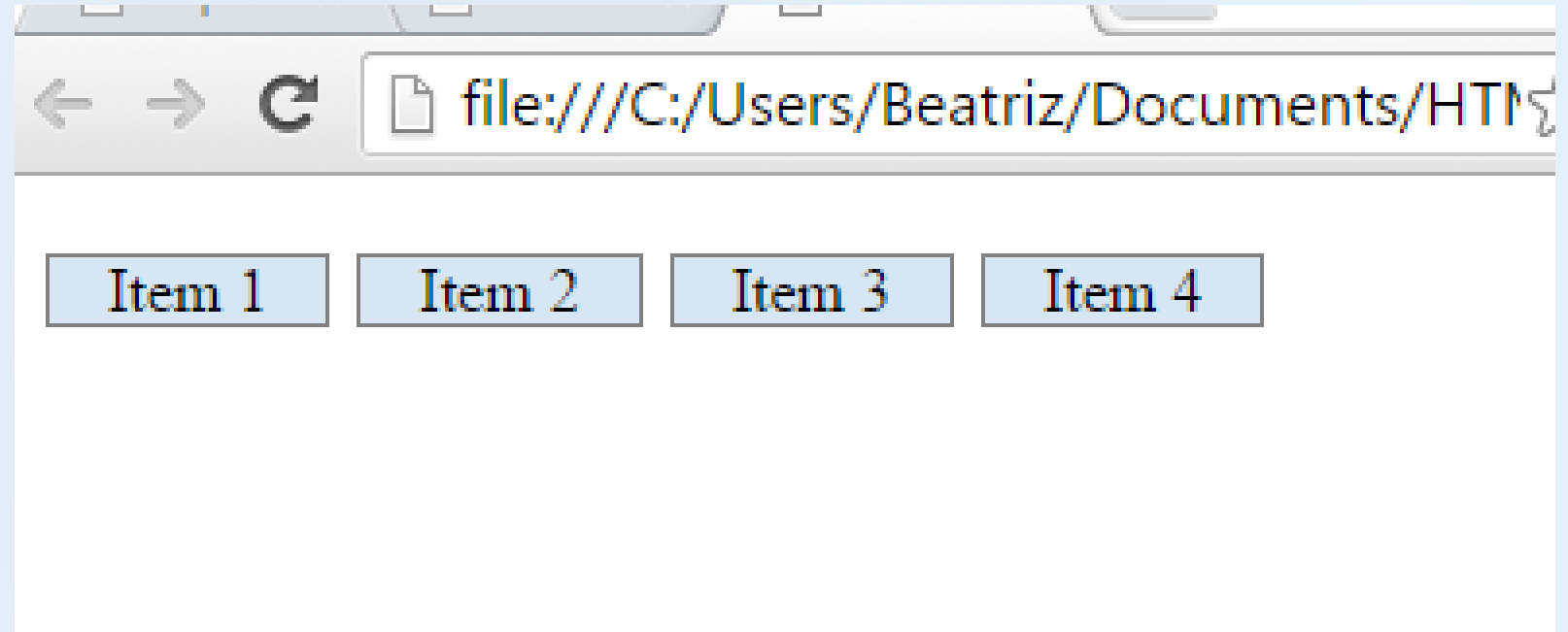
## Actividad 21B. Crea la siguiente web usando CSS



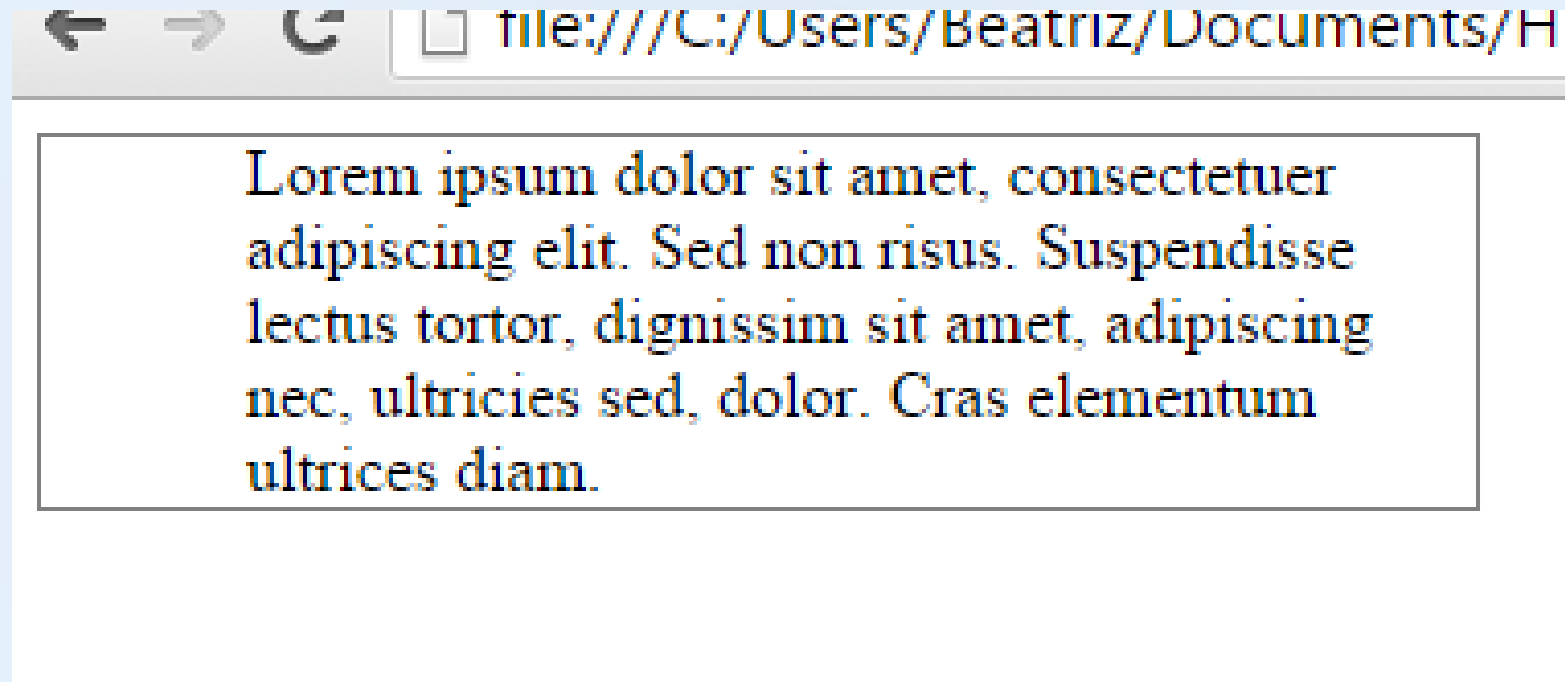
**Actividad 22.** Crea la siguiente web usando CSS. Los Item son elementos de una lista no ordenada.

Código html:

```
<body>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ul>
</body>
```



## Actividad 23. Crea la siguiente web usando CSS



## Actividad 24. Crea la siguiente web usando CSS



**Html5 + CSS3**

La propiedad CSS [border-radius](#) permite a los desarrolladores Web definir qué tan **redondeadas** serán las **esquinas**. La redondez de cada esquina está definida usando uno o dos valores para el radio que define su forma dependiendo si es un círculo o una elipse.

Esta propiedad es un atajo para establecer las cuatro propiedades **border-top-left-radius**, **border-top-right-radius**, **border-bottom-right-radius** y **border-bottom-left-radius**.

Los valores se pueden definir en longitud y porcentaje.

Suelen usarse para definir botones con las esquinas curvas, dándole un aspecto más suavizado a las esquinas

Ver:

Columnas:

[https://www.html6.es/t17\\_columnas.html](https://www.html6.es/t17_columnas.html)

[CSS Multiple Columns \(w3schools.com\)](https://www.w3schools.com/css/css_multiple_columns.asp)

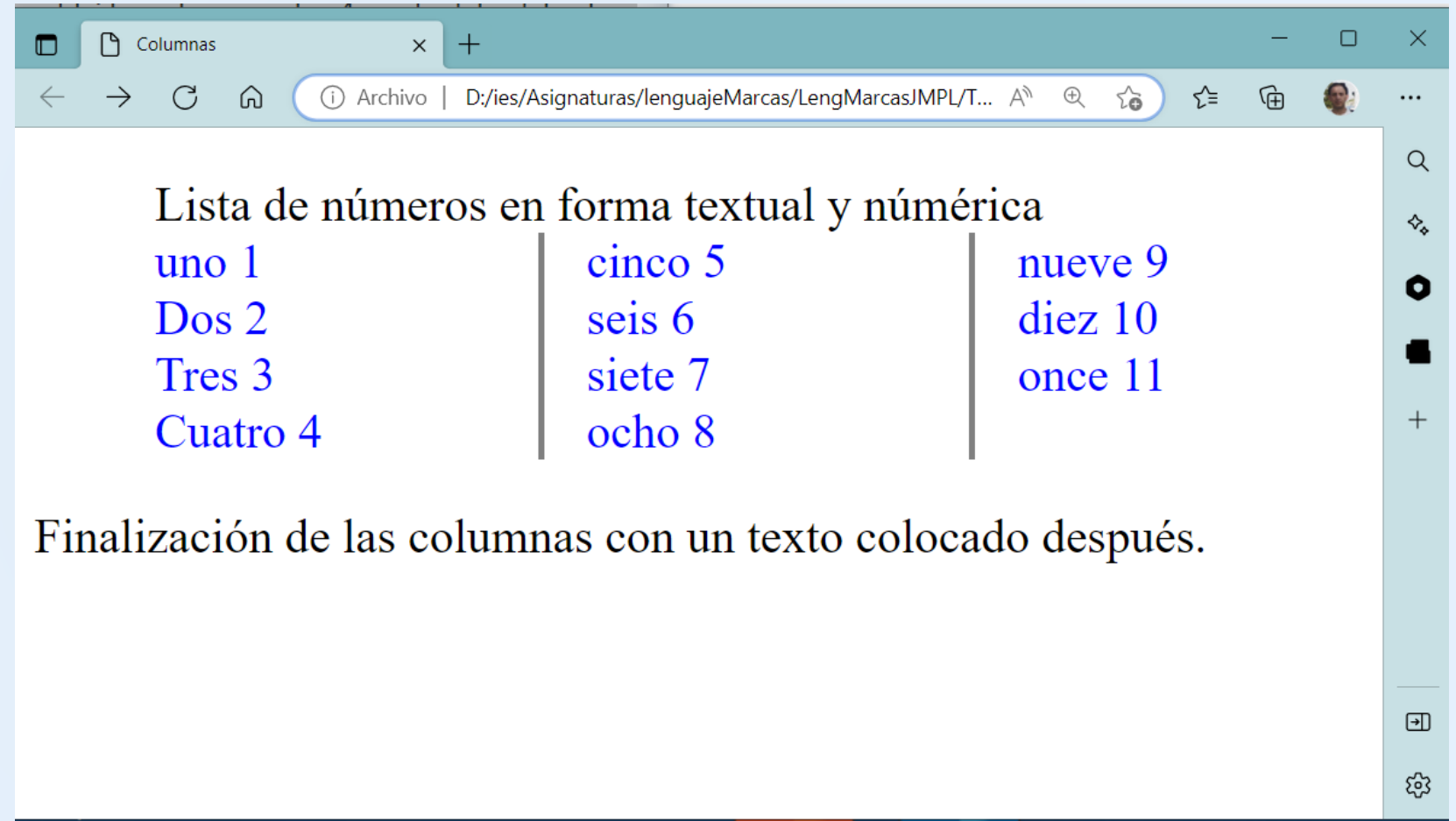
Layouts:

[HTML Layout Elements and Techniques \(w3schools.com\)](https://www.w3schools.com/html/html_layout_elements.asp)

[https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_layout\\_float](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_layout_float)

<https://www.positioniseverything.net/css-layout>

**Actividad 25.** Crea la siguiente web usando CSS . Sigue las indicaciones de las webs recomendadas



## Actividad 26. Crea la siguiente web usando CSS . Sigue las indicaciones de las webs recomendadas para utilizar layouts (FlexBoxLayout)

Este párrafo se colocará en la parte superior

### Productos Huerta



[Patatas](#)  
[Zanahorias](#)  
[Remolacha](#)

#### Patata



Tubérculos comestibles de la patatera (*Solanum tuberosum* L), una planta de la familia de las Solanáceas, emparentada con el tomate, la berenjena y los pimientos. Constituyen un alimento muy antiguo, utilizado por los pueblos de América antes de que Cristóbal Colón llegase a ella.

Tubérculos comestibles de la patatera (*Solanum tuberosum* L), una planta de la familia de las Solanáceas, emparentada con el tomate, la berenjena y los pimientos. Constituyen un alimento muy antiguo, utilizado por los pueblos de América antes de que Cristóbal Colón llegase a ella.

Tubérculos comestibles de la patatera (*Solanum tuberosum* L), una planta de la familia de las Solanáceas, emparentada con el tomate, la berenjena y los pimientos. Constituyen un alimento muy antiguo, utilizado por los pueblos de América antes de que Cristóbal Colón llegase a ella.

Tubérculos comestibles de la patatera (*Solanum tuberosum* L), una planta de la familia de las Solanáceas, emparentada con el tomate, la berenjena y los pimientos. Constituyen un alimento muy antiguo, utilizado por los pueblos de América antes de que Cristóbal Colón llegase a ella.

Tubérculos comestibles de la patatera (*Solanum tuberosum* L), una planta de la familia de las Solanáceas, emparentada con el tomate, la berenjena y los pimientos. Constituyen un alimento muy antiguo, utilizado por los pueblos de América antes de que Cristóbal Colón llegase a ella.

Tubérculos comestibles de la patatera (*Solanum tuberosum* L), una planta de la familia de las Solanáceas, emparentada con el tomate, la berenjena y los pimientos. Constituyen un alimento muy antiguo, utilizado por los pueblos de América antes de que Cristóbal Colón llegase a ella.

#### Zanahoria



llamada popularmente zanahoria, es la forma domesticada de la zanahoria silvestre (*Daucus carota*), especie de la familia de las umbelíferas, también denominadas apiáceas, y considerada la más importante y de mayor consumo dentro de esta familia. Es oriunda de Europa y Asia sudoccidental. Se cultiva por su raíz mucho más grande, sabrosa y de textura menos fibrosa que la especie silvestre

#### Remolacha

Beta vulgaris es una especie herbácea perteneciente a la subfamilia Betoideae de la familia Amaranthaceae. Económicamente, es el cultivo más importante del gran orden de los Caryophyllales. Existen numerosas variedades cultivadas, algunas para su consumo como verdura, y otras como materia prima industrial



Come alimentos sanos y naturales y mejorará tu salud



Todos los campos de texto tienen soporte completo para las propiedades relacionadas con el modelo de cajas de CSS (width, height, padding, margin y border).

Para modificar un estilo concreto de un tipo de formulario se puede usar la directiva para seleccionar atributos **input[type=XXX]** siendo XXX el tipo de formulario (text, search, password...)

[https://www.w3schools.com/css/css\\_form.asp](https://www.w3schools.com/css/css_form.asp)

La directiva selector de atributos también se puede aplicar en

h1[class=nomClase]{color: red;} aunque en este caso es más sencillo utilizar el selector h1.nomClase  
para el html : <h1 class="kkk">ejemplo selector atributos</h1>

## Bordes

Quizás la manera más rápida y sencilla de modificar radicalmente la apariencia de un formulario es cambiando los bordes de los campos que en él se despliegan.

Para ello tenemos que hacer uso de la propiedad CSS "**border**", que nos permitirá cambiar el tipo de borde, su color y tamaño. Si quisiéramos por ejemplo tener un borde de tipo sólido, color gris y con 1 píxel de anchura, basta con utilizar una sentencia como esta:

```
input, textarea { border: 1px solid #333; }
```

Esto se aplicará a todos los elementos de tipo "input" y "textarea" que tengamos en el formulario.

En cambio, si quisiéramos ceñirnos más al estilo original pero cambiando los colores, podemos utilizar la propiedad "**border-color**" con un valor para cada sección, como se muestra aquí:

```
input, textarea {
```

```
    border: 1px solid;
```

```
    border-color: #333 #666 #666 #333;
```

```
}
```

En cuanto a lo que bordes para **selects** se refiere, estos se ven más limitadas por parte de los navegadores, aunque en la actualidad estos problemas han tendido a reducirse.

Lo mismo puede ocurrir con los checkbox y los radio button, su estilo puede permanecer diferente al resto de los bordes que manejamos.

## Fuentes

Podemos especificar el tipo de fuente con la que se va a desplegar el texto dentro de cada campo del formulario, y para ello basta con recurrir a la propiedad "font" de CSS. Al igual que con el resto de los elementos donde podemos aplicar esta propiedad, el color y el tamaño son aspectos elegibles.

```
input, textarea { font: 10px arial }
```

## Fondo

El último aspecto relevante que se puede modificar para resaltar o dar un estilo distinto a los elementos de un formulario, es el fondo.

Por lo general esta propiedad, representada por "background", es tratada con mucho cuidado y sólo se utiliza en casos donde la interfaz lo necesite.

Un uso muy práctico del cambio de fondo se puede hacer en conjunto con el evento "focus" mediante CSS, así podemos cambiar el fondo del campo únicamente cuando el usuario se encuentre posicionado sobre él:

```
input:focus, textarea:focus { background: #ccc }
```

## Actividad 27. Crea el siguiente formulario usando CSS .

### Formulario petición datos

Nombre

Last Name

Pais



# Comentarios

---

**Comentarios.** Los comentarios son fundamentales en todo lenguaje de programación que proporcionan información de utilidad a la hora de estructurar o indicar cualquier apreciación sobre el contenido. Pueden **ocupar una o varias líneas**, dependiendo de lo que se quiera añadir.

En el caso de CSS, al igual que los comentarios en HTML, no son visibles por el navegador. Todo comentario va entre los caracteres de inicio y fin:

- Los caracteres de inicio del comentario son la barra y el asterisco (/\*).
- Los caracteres de fin de comentario son el asterisco y la barra (\*/).

```
<style type="text/css">
  h1 + h3 {
    color: red;
    /*pone de color rojo el texto*/
  }
</style>
</head>
```



# Agrupamientos y herencia

---

**Agrupamientos.** Cuando pretende aplicarse la misma declaración a distintos selectores, la manera de hacerlo es agrupando los selectores separados por el carácter coma (,).

```
<style type="text/css">
  h1{color: red;}
  h2{color: red;}
  h3{color: red;}
</style>
```

```
h1,h2,h3{
  color: red;
}
```

**Herencia.** Al tener un documento HTML una estructura de árbol, los sectores anidados heredan las declaraciones de los padres. Esto se ve perfectamente con la etiqueta <body>, ya que, si se le aplica un estilo, afectará a todos los elementos del documento, debido a que es el padre del resto de etiquetas que se añadan. En el caso de que el usuario modifique un elemento concreto mediante un selector de clase, ID u otros, prevalecerá dicho estilo en ese elemento.

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">
    h1{color: red;}
    body{color:Blue;}
  </style>
</head>
<body>
  <h1> Cultivo del <b>Litchi</b> en la zona de la <i>Axarquía</i></h1>
  <h3> >Cultivo del <b><i>longan</i></b> en Málaga</h3>
  <h3> >Cultivo del <b>platano</b> en Canarias</h3>
  <h1> >Cultivo de <b>la carambola</b> en Málaga</h1>
  <h2> >Cultivo del <i>canisté</i> en Canarias</h2>
  <h3> >Cultivo de la <i>guanábana</i> en Canarias</h3>
</body>
</html>
```

**Cultivo del Litchi en la zona de la Axarquía**

Cultivo del *longan* en Málaga

Cultivo del platano en Canarias

**Cultivo de la carambola en Málaga**

Cultivo del *canisté* en Canarias

Cultivo de la *guanábana* en Canarias

# CSS3

---

En este apartado se detallan las nuevas características y propiedades de la versión 3 de CSS, que es un lenguaje que dota de nuevos estilos y apariencia a los elementos de un documento como HTML XML, etc. Permite, entre otros aspectos, añadir sombras de cajas, transiciones, poner esquinas redondeadas, usar gradientes lineales y radiales, insertar imágenes en los bordes, opacidad y transparencia, etc., ahorrando bastante tiempo de trabajo y mejorando el rendimiento.

CSS3 presenta nuevos **selectores de atributos**, como son:

- **[atributo^="valor"]** → selecciona los elementos que tengan un **atributo** que **empiece por** el valor introducido.
- **[atributo\$="valor"]** → selecciona los elementos que tengan un atributo que **termine por** el valor introducido.
- **[atributo\*="valor"]** → selecciona los elementos que tengan un **atributo** que **contengan** el valor introducido.

# Nuevas propiedades

---

CSS3

Se ha añadido un nuevo canal llamado Alpha al modo RGB: **RGBA**, que toma como valores posibles los del rango [0...1], donde el valor 0 se interpreta como completamente transparente y el 1 corresponde a completamente opaco. En el siguiente ejemplo se muestran distintas etiquetas de cabecera aplicando al canal Alpha distintos valores para ver cómo a menor valor, aumenta la transparencia del elemento.

```
<!DOCTYPEhtml>
<html>
<head>
<title>Editorial síntesis</title>
<styletype="text/css">
h1 { background-color:rgba(20, 90, 50 ,1);    }
h2 { background-color:rgba(20, 90, 50 ,0.75);  }
h3 { background-color:rgba(20, 90, 50 ,0.50);  }
h4 { background-color:rgba(20, 90, 50 ,0.25);  }
h5 { background-color:rgba(20, 90, 50 ,0);
</style>
</head>
<body>
<center>
<h1>Introducción CSS3</h1>
<h2>Editorial síntesis</h2>
<h3>Aplicanco transparencias colores RGBA</h3>
<h4>Dando nuevos valores al parámetro alpha</h4>
<h5>En este caso se verá la aplicación de los valores alpha del
100%, 75%, 50%, 25% y 0% a las distintas etiquetas</h5>

</center>
</body>
</html>
```

## Introducción CSS3

### Editorial síntesis

#### Aplicanco transparencias colores RGBA

#### Dando nuevos valores al parámetro alpha

En este caso se verá la aplicación de los valores alpha del 100%, 75%, 50%, 25% y 0% a las distintas etiquetas

También se puede conseguir  
transparencia en la imagen con  
[opacity:0.7;](#) con valor entre 0.0 y 1.0

Los valores HSL y HSLA son otra manera de definir los colores mediante tono, saturación y luminosidad o brillo. La aplicación es similar al RGBA con los siguientes rangos de valores: Tono: [0..360]. Saturación: valor porcentual. Luminosidad: valor porcentual. La sintaxis es la siguiente:

hsl(tono, saturación y luminosidad)

[https://www.w3schools.com/colors/colors\\_hsl.asp](https://www.w3schools.com/colors/colors_hsl.asp)

hsla(tono, saturación y luminosidad, canal Alpha)

El siguiente código de ejemplo se aplica para el color hsl para las dos primeras cabeceras y hsla para las dos restantes con un valor Alpha de 0,75 y 0,25, respectivamente.

```
<!DOCTYPEhtml>
<html>
<head>
<title>Editorial síntesis</title>
<styletype="text/css">

h1 { background-color:hsl(200, 40%, 70%); }
h2 { background-color:hsl(200, 75%, 75%); }
h3 { background-color:hsla(200, 75%, 75%, 0.75); }
h4 { background-color:hsla(200, 75%, 75%, 0.25); }

</style>
</head>
<body>
<center>
<h1>Introducción CSS3</h1>
<h2>Editorial síntesis</h2>
<h3>Aplicanco colores HSL</h3>
<h4>Dando nuevos valores a los parámetros</h4>
</center>
</body>
</html>
```

**Introducción CSS3**

**Editorial síntesis**

**Aplicanco colores HSL**

**Dando nuevos valores a los parámetros**

# Pseudo-classes

---

CSS



Una pseudoclase/pseudoselector CSS es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado. Por ejemplo, **:hover** aplicará un estilo cuando el usuario **pase el ratón sobre el elemento** especificado por el selector.

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

Las **pseudoclases**, junto con los **pseudoelementos**, permiten **aplicar un estilo** a un elemento no sólo en relación con el contenido del árbol de documento, sino también en **relación a factores externos** como el historial del navegador (**:visited**, por ejemplo), el estado de su contenido (como **:checked** en algunos elementos de formulario), o la posición del ratón (como **:hover** que permite saber si el ratón está encima de un elemento o no).

## Sintaxis

```
selector:pseudoclase { propiedad: valor; }
```

Hay muchas pseudo-clases distintas, pero veremos solo las más usadas:

**:active**. La pseudo-clase :active de CSS representa un elemento (como un botón) que el usuario está activando. Cuando se usa un mouse, la "activación" generalmente comienza cuando el usuario **presiona el botón primario del mouse y termina cuando se suelta**. La pseudo-clase :active se usa comúnmente en los elementos <a> y <button>, pero también se puede usar en otros elementos.

/\* Selecciona cualquier <a> que esté siendo activado \*/

```
a:active {  
  color: red;  
}
```

Los estilos definidos por la pseudoclase :active serán anulados por cualquier pseudoclase posterior relacionada con el enlace (:link, :hover o :visited) que tenga al menos la misma especificidad. Para darle un estilo apropiado a los enlaces, coloque la regla :active después de todas las demás reglas relacionadas con el enlace, tal como lo define el orden LVHA: :link — :visited — :hover — :active.

**:checked**. La pseudo-clase :checked de CSS representa cualquier **radio** (<input type="radio">), **checkbox** (<input type="checkbox">) u **option** (<option> en un elemento <select>) que está marcado o **conmutado** a un estado **on**. El usuario puede activar este estado marcando/seleccionando un elemento, o desactivándolo desmarcando/deseleccionando el elemento.

/\* Coincide con cualquier checked/selected radio, checkbox, u option \*/

```
:checked {
    margin-left: 25px;
    border: 1px solid blue;
}
input:checked {
    height: 50px;
    width: 50px;
}
```

**:required**. La pseudo-clase :required de CSS representa cualquier elemento <**input**>, <**select**>, o <**textarea**> que tenga el atributo **required** establecido en él. Esta pseudo-clase es útil para **resaltar campos** que deben tener **datos válidos** antes de que se pueda enviar un formulario.

/\* Selecciona cualquier <input> requerido \*/

```
input:required {
    border: 1px dashed red;
}
```

**:disabled**. La pseudo-clase :disabled de CSS representa a cualquier **elemento deshabilitado**. Un elemento se encuentra deshabilitado si no puede ser activado (por ejemplo ser seleccionado, hacer click en él o aceptar texto de entrada) ni aceptar el foco.

```
/* Selecciona cualquier <input> deshabilitado */  
input:disabled {  
    background: #ccc;  
}
```

**:enabled**. La pseudo-clase :enabled CSS representa cualquier **elemento habilitado**. Un elemento está habilitado si puede ser activado (es decir seleccionado, se puede hacer click en él, acepta que se le introduzca texto, etc.) o si acepta el foco.

```
/* Selecciona cualquier <input> habilitado */  
input:enabled {  
    color: blue;  
}
```

**:focus**. La pseudo-clase :focus CSS representa un elemento (como una entrada de formulario) que ha recibido el foco. Generalmente se activa cuando el **usuario hace clic**, toca un elemento o lo selecciona con la tecla "Tab" del teclado.

```
/* Selecciona cualquier <input> cuando se enfoca */
```

```
input:focus {  
  color: red;  
}
```

**:hover**. La pseudo-clase :hover de CSS coincide cuando **el usuario interactúa** con un elemento con un dispositivo señalador, **pero no necesariamente lo activa**. Generalmente se activa cuando el usuario se desplaza sobre un elemento con el cursor (puntero del mouse).

```
/* Selecciona cualquier elemento <a> cuando está "mantenido (hovered)" */
```

```
a:hover {  
  color: orange;  
}
```

**:invalid**. La pseudo-clase :invalid de CSS representa cualquier elemento <input> u otro elemento <form> cuyos **contenidos se validen insatisfactoriamente**. Esta pseudo-clase es útil para resaltar errores de campo para el usuario.

```
/* Selecciona cualquier <input> no válido */  
input:invalid {  
    background-color: pink;  
}
```

**:valid**. La pseudo-clase :valid de CSS representa cualquier elemento <input> u otro elemento <form> cuyo contenido se valide satisfactoriamente. Esto permite que los campos válidos adopten fácilmente una apariencia que ayuda al usuario a confirmar que sus datos están formateados correctamente. Esta pseudo-clase es útil para resaltar los campos correctos para el usuario.

```
/* Selecciona cualquier <input> válido */  
input:valid {  
    background-color: powderblue;  
}
```

[:link](#). La pseudo-clase :link CSS representa un elemento que **aún no se ha visitado**. Coincide con cada elemento no visitado <a>, <area>, o <link> que tiene un atributo href.

```
/* Selecciona cualquier <a> que aún no se haya visitado */
```

```
a:link { color: red; }
```

[:visited](#). La pseudo-clase :visited de CSS representa enlaces que el usuario **ya ha visitado**. Por motivos de privacidad, los estilos que se pueden modificar con este selector son muy limitados.

```
/* Selecciona cualquier <a> que ha sido visitado */
```

```
a:visited { color: green; }
```

[:empty](#) seleccionar elementos vacíos. Por ejemplo <p></p>

Puedes ver más selectores en :

[https://www.w3schools.com/cssref/css\\_selectors.php](https://www.w3schools.com/cssref/css_selectors.php)

# Pseudo-elementos

---

CSS



Al igual que las pseudo-classes, 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 { propiedad: valor; }
```

Hay muchos pseudo-elementos distintos, pero veremos solo los más usados:

**::after**. En CSS, ::after **Inserta algo después del contenido de cada elemento seleccionado**. Este elemento se muestra en línea con el texto de forma predeterminada.

*/\* Añade una flecha después de los enlaces \*/*

```
a::after {  
  content: "→";  
}
```

**::before**. En CSS, ::before **Inserta algo antes del contenido de cada elemento seleccionado**. Este elemento se muestra en línea con el texto de forma predeterminada.

*/\* Añade un corazón antes de los enlaces \*/*

```
a::before {  
  content: "♥";  
}
```

**::first-letter**. El pseudo-elemento ::first-letter aplica **estilos a la primera letra** de la primera línea un elemento de bloque, sólo cuando no es precedido de otro contenido (como imágenes o tablas).

```
/* Selecciona la primera letra de <p> */
```

```
p::first-letter {  
  font-size: 130%;  
}
```

**::first-line**. El Pseudoelemento ::first-line aplica estilos **a la primera línea** de un elemento de bloque. Nótese que la longitud de la primera línea depende de muchos factores, incluyendo el ancho del elemento, el ancho del documento y el tamaño de fuente del texto.

```
/* Selecciona la primera línea de un <p> */
```

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

- ▶ :first primera ocurrencia del selector
- ▶ :first-child primer hijo dentro del árbol
- ▶ :first-of-type primera ocurrencia de ese tipo seleccionada por el selector
- ▶ :last última ocurrencia del selector
- ▶ :last-child ultimo hijo dentro del árbol
- ▶ :last-of-type ultimo de su tipo
- ▶ :nth-child(n) enésimo hijo
- ▶ :nth-last-child() enésimo ultimo hijo
- ▶ :nth-last-of-type() enésimo de su tipo

```
p:nth-last-of-type(2) {  
    background: red;  
}
```

```
li:first-child { /* primer hijo de la lista ponerlo en gris */  
  background-color: grey;  
}
```

```
ul:first-of-type li:last-child { /* último hijo dentro de la primera lista ponerlo azul */  
  background-color: blue;  
  color: white;  
}
```

```
ul:last-of-type { /* ultima lista ponerla con borde negro*/  
  border: 1px solid black;  
}
```

```
ul:last-of-type li:nth-child(2) { /*2º hijo de la última lista ponerla naranja*/  
  color: orange;  
}
```

## Selector de atributo

El selector [atributo] se utiliza para seleccionar elementos que tengan un determinado atributo. Por ejemplo para seleccionar las etiquetas <a> que tengan un atributo target:

```
a[target] {  
  background-color: yellow;  
}
```

## Selector de valor de atributo

El selector [atributo="valor"] se utiliza para seleccionar elementos con un valor concreto en un atributo. En el siguiente ejemplo se seleccionan las etiquetas <a> con el valor de target="\_blank" :

```
a[target="_blank"] {  
  background-color: yellow;  
}
```

[https://www.w3schools.com/css/css\\_attribute\\_selectors.asp](https://www.w3schools.com/css/css_attribute_selectors.asp)

Los navegadores tienen estilos por defecto que se aplican cuando no utilizamos nuestros propios estilos (<https://datobyte.blogspot.com/2017/02/cual-es-el-css-por-defecto-que-usan-los.html> ) y el usuario del navegador también puede definir sus propios estilos (<https://desarrolloweb.com/articulos/reglas-estilo-css-usuario-autor.html> )..

También es posible adaptar nuestra página a navegador que se vaya a usar utilizando javascript, pero esto se sale del contenido de este curso: <https://www.elmaquetadorweb.com/2015/04/css-especificos-para-cada-navegador.html>

## Hojas de reseteo

Para conseguir uniformidad debemos utilizar hojas que eliminen los estilos por defecto de los navegadores. Entre ellas están las proporcionadas en:

- <https://meyerweb.com/eric/tools/css/reset/>
- <https://cssdeck.com/blog/scripts/html5-doctor-css-reset-stylesheet/>

Al utilizarlas se modificas sustancialmente la página y se requiere más trabajo de customización.

- **Minimizar** el CSS para subirlo a producción (quitar cualquier espacio en blanco).
- **Borrar reglas** innecesarias y duplicadas.
- **Ordenar** las propiedades de nuestras reglas o selectores por orden **alfabético** para que todo sea más legible y fácil de mantener.
- Poner el **CSS** en la **cabecera** (aunque es eso algo sobre lo que ya hemos insistido).
- Si el **CSS** es muy grande lo **partiremos** en varios (**varias etiquetas link**).
- **Organizar** las reglas poniendo las que tengan **relación juntas**.
- **Documentar** tus ficheros CSS.

Para ayudaros con estas tareas existen muchas herramientas pero yo os recomiendo estas:

- [CSS Minifier](#). Minimiza espacios. Copia el código en input CSS y pulsa Minify
- [CSS Optimizer](#). Elimina reglas redundantes. Copia el css , elige propiedades y process CSS
- [Unused CSS](#). Indica reglas que no se aplican. (solo para urls válidas, no para fichs en disco)
- Validación Online: <https://jigsaw.w3.org/css-validator/>



[Preprocesador CSS](#) es un programa que genera CSS con una estructura más legible y más fácil de mantener.

Añaden al CSS tradicional características como:


- Las variables
- Los selectores anidados
- Los bucles
- Los mixins –parametrizar la inclusión de ciertas partes.
- Etc.

Ejemplo de sass donde se usa la variable \$font-stack

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;
```

```
body {  
  font: 100% $font-stack;  
  color: $primary-color;  
}
```

Herramientas:

- [less](#). [Aprende LESS en 15 Minutos](#)  - YouTube
- [sass](#)
- [postcss](#).

- Maquetación web: transformar el diseño de una web en un conjunto de archivos (html, css y js) para que los navegadores web puedan interpretarlos y reproducirlos correctamente en diferentes dispositivos. La maquetación es una fase avanzada del proceso de desarrollo web.
- Variables en css: [https://developer.mozilla.org/es/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/es/docs/Web/CSS/Using_CSS_custom_properties)
- Flexbox: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- Grid\_layout: [https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout)
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- Foro html, css y otros: <http://www.forosdelweb.com/f53/>
- Diseño responsivo : Se adapta a cualquier dispositivo
- Animaciones y efectos css, transition-property,...
- CSS **Grid** Layout y CSS Flexbox Layout

- [Web Creme](#): Ejemplos diarios de páginas diseñadas con CSS. Búsquedas a partir del color utilizado en la página.
- [CSS Remix](#): Páginas diseñadas exclusivamente con CSS y con la posibilidad de puntuar su diseño.
- [CSS Zen Garden](#): Diseños complejos realizados mediante CSS.
- [Open Source Web Design](#): Plantillas gratuitas con posibilidad de utilizarlas libremente en aplicaciones personales y comerciales.
- Trucos: <https://uniwebsidad.com/tutoriales-sobre-css>

**Recurso Web.** Bootstrap es un framework de código abierto para el diseño de interfaces web mediante hojas de estilo y JavaScript. Es una colección de clases CSS y funciones JavaScript y se utiliza para diseño responsivo (webs que **se adaptan a los diferentes tamaños de pantalla** reorganizando el contenido, escalando las imágenes y/o simplificando el menú).

<https://getbootstrap.com/>

[https://fgarciajulia.github.io/mi\\_primera\\_pagina/](https://fgarciajulia.github.io/mi_primera_pagina/)