

### ÍNDICE

<b>1. INTRODUCCIÓN CSS.....</b>	<b>4</b>
1.1 DEFINICIÓN .....	4
1.2 VENTAJAS.....	5
1.3 SOPORTE DE LOS NAVEGADORES .....	6
<b>2. REGLAS CSS.....</b>	<b>7</b>
▪ FORMATO Y ELEMENTOS .....	7
▪ COMO INTERPRETAR LA DOCUMENTACIÓN OFICIAL DE LAS REGLAS CSS .....	8
▪ SHORTHAND.....	9
▪ VALIDADORES CSS .....	10
ACTIVIDAD 1: CONSULTAR DOCUMENTACIÓN: VALORES DE LAS PROPIEDADES CSS .....	10
<b>3. INCORPORAR CSS A UN DOCUMENTO HTML.....</b>	<b>10</b>
3.1 ATRIBUTO HTML STYLE .....	10
ACTIVIDAD 2: INTEGRAR CSS CON EL ATRIBUTO HTML STYLE .....	10
3.2 ETIQUETA HTML <STYLE> .....	11
ACTIVIDAD 3: INTEGRAR CSS CON LA ETIQUETA HTML <STYLE> .....	11
3.3 EN UN FICHERO CSS INDEPENDIENTE .....	11
ACTIVIDAD 4: INTEGRAR CSS EN UN FICHERO INDEPENDIENTE .....	12
<b>4. SELECTORES .....</b>	<b>12</b>
4.1 DEFINICIÓN DE SELECTOR.....	12
4.2 TIPOS DE SELECTORES.....	13
▪ SELECTOR UNIVERSAL.....	13
▪ SELECTOR DE ETIQUETA .....	13
▪ SELECTOR DE AGRUPACIÓN .....	13
▪ SELECTOR DESCENDENTE .....	14
▪ SELECTOR DE CLASE .....	15
▪ SELECTOR DE ID .....	16
▪ SELECTOR DE HIJOS.....	17
▪ SELECTOR DE HERMANOS.....	18
▪ SELECTOR ADYACENTE .....	18
▪ SELECTOR DE ATRIBUTOS .....	19
▪ SELECTOR DE PSEUDO-CLASE .....	21
▪ SELECTOR DE PSEUDO-ELEMENTO.....	25
4.3 COMBINACIÓN DE SELECTORES .....	27
4.4 RESUMEN DE SELECTORES .....	28
4.5 ELEMENTOS HTML <DIV> Y <SPAN> PARA FORMATO .....	29

ACTIVIDAD 5: SELECTORES .....	31
<b>5. HERENCIA Y CASCADA .....</b>	<b>32</b>
5.3 HERENCIA.....	32
▪ INHERITED .....	33
▪ INHERIT E INITIAL.....	33
5.4 CASCADA.....	34
▪ CONFLICTO .....	34
▪ REGLAS DE LA CASCADA .....	35
ACTIVIDAD 6. CASCADA.....	36
<b>6. MODELO DE CAJAS .....</b>	<b>36</b>
6.1 DEFINICIÓN DEL MODELO .....	36
6.2 ELEMENTOS DE LA CAJA .....	37
6.3 ALGUNAS PROPIEDADES CSS PARA CAJAS .....	38
▪ PADDING .....	38
▪ MARGIN.....	39
▪ BORDER-WIDTH .....	40
▪ BORDER-STYLE .....	40
▪ BORDER-COLOR.....	40
▪ BORDER.....	40
▪ BORDER-RADIUS .....	41
ACTIVIDAD 7. BORDES Y ESPACIOS.....	41
▪ BOX-SIZING .....	42
▪ WIDTH Y HEIGHT.....	42
▪ BOX-SHADOW .....	43
▪ DISPLAY .....	44
ACTIVIDAD 8: CAMBIO DEL TIPO DE ELEMENTO UTILIZANDO DISPLAY .....	44
▪ OVERFLOW .....	45
▪ BACKGROUND-CLIP .....	45
▪ BACKGROUND-COLOR .....	46
▪ BACKGROUND-IMAGE.....	46
▪ BACKGROUND-REPEAT .....	47
▪ BACKGROUND-ATTACHMENT.....	47
▪ BACKGROUND-SIZE .....	47
▪ BACKGROUND-POSITION .....	47
▪ BACKGROUND.....	47
ACTIVIDAD 9. REPETICIÓN DE IMAGEN DE FONDO .....	48
ACTIVIDAD 10. SCROLL DE IMAGEN DE FONDO .....	48
▪ COLAPSO DE MARGEN .....	48

ACTIVIDAD 11. COLAPSO DE MARGEN .....	49
6.4 RESUMEN DE PROPIEDADES CSS PARA CAJAS .....	50

#### Enlaces de interés

[CSS Tutorial \(w3schools.com\)](https://www.w3schools.com/css/)

[CSS: Cascading Style Sheets | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/CSS)

[CSS \(w3.org\)](https://www.w3.org/Style/CSS/)

# 1. Introducción CSS

## 1.1 Definición

**CSS:** Hojas de estilo en cascada (Cascading Style Sheets)

Es un lenguaje estandarizado por [W3C \(World Wide Web Consortium\)](#) para aplicar formato a documentos web. Puede ser aplicado a documentos HTML, XHTML, XML, SVG, etc.

Antes de emplear CSS, para diseñar una página web con un formato determinado era necesario introducir etiquetas HTML de formato.

Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Ejemplo de estilos sin CSS</title>
</head>
<body>
  <h2> <font color="red" face="Arial" size="5"> Titular </font> </h2>
  <p> <font color="gray" face="Verdana" size="2"> Un párrafo de texto. </font></p>
</body>
</html>
```

**Esto presenta muchos problemas.** Por ejemplo, si existiesen 50 títulos y 50 párrafos en la página, habría que repetir las etiquetas <font> para cada uno de ellos.

Si el sitio web además tuviese varias páginas, habría que repetir <font> en cada página. Esto dificultaría además las actualizaciones y cambios.

Con CSS este ejemplo podría resolverse de la forma:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Ejemplo de estilos con CSS</title>
  <style>
    h2 { color: red;
         font-family: Arial;
         font-size: large;}
    p { color: gray;
        font-family: Verdana;
        font-size: medium; }
  </style>
</head>
<body>
  <h2>Titular de la página</h2>
  <p>Un párrafo de texto.</p>
</body>
</html >
```

Existe una zona especial en la que se incluye toda la información relacionada con los estilos de la página. Esos estilos se aplicarán a todos los elementos <h2> y <p> que se definan en el documento, de forma automática.

CSS permite **separar los contenidos de la página y la configuración de su aspecto**.

El formato CSS es el siguiente (se profundizará más adelante):

```
h2 { color: red; font-family: Arial; font-size: large; }
```

## 1.2 Ventajas

Al utilizar HTML y CSS, se está separando el contenido del documento del formato de presentación del mismo. Las ventajas de esta separación son:

- ➔ **Mejora de la accesibilidad.** Incorpora funcionalidades para crear hojas de estilos auditivas, especialmente diseñadas para invidentes.
- ➔ Mayor **flexibilidad** en las características de presentación de la información.
- ➔ Posibilidad de presentar el mismo documento en distintos formatos, en función del tipo de dispositivo, tamaño de pantalla, opciones de navegador...
- ➔ Posibilidad de tener la misma hoja de estilos para distintos documentos HTML.
- ➔ Evita la duplicidad de código de presentación.
- ➔ Disminución de la complejidad del documento al agrupar directivas de formato.
- ➔ El lector puede especificar su propia hoja de estilos para sobrescribir la del programador.
- ➔ Los resultados de presentación son predecibles aplicando las reglas en cascada para calcular prioridades.

### 1.3 Soporte de los navegadores

La adopción de CSS por parte de los navegadores ha requerido un largo periodo de tiempo. Puede que alguna funcionalidad CSS no sea compatible con todos los navegadores.






El mismo año que se publicó la versión CSS 1, Microsoft lanzaba su navegador Internet Explorer 3.0, que disponía de un soporte bastante reducido de CSS. El primer navegador con soporte completo de CSS 1 fue la versión para Mac de Internet Explorer 5, que se publicó en el año 2000.

Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina **motor**. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador.

En el enlace [https://en.wikipedia.org/wiki/Comparison\\_of\\_browser\\_engines\\_\(CSS\\_support\)](https://en.wikipedia.org/wiki/Comparison_of_browser_engines_(CSS_support)) puede encontrarse información detallada sobre la compatibilidad de los motores con las versiones CSS.

Para comprobar la compatibilidad de una propiedad CSS debe consultarse su documentación. Por ejemplo:

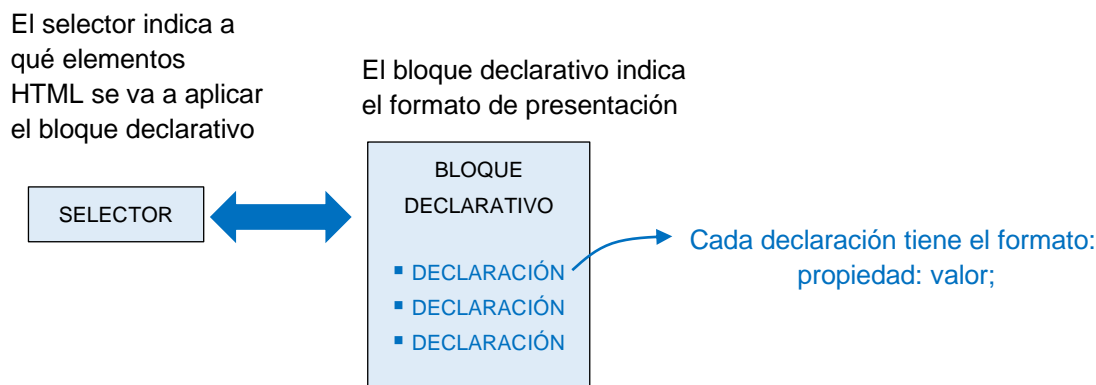
- La propiedad CSS font-stretch configura cómo van a visualizarse las letras respecto al su estiramiento. Accediendo a su documentación en [w3cschools](https://www.w3schools.com/css/css_font-stretch.asp) se puede comprobar qué navegadores son compatibles, es decir, la reconocen, y desde qué versión.

Browser Support					
The numbers in the table specify the first browser version that fully supports the property.					
Property					
font-stretch	48.0	9.0	9.0	11.0	35.0

## 2. Reglas CSS

### ▪ Formato y elementos

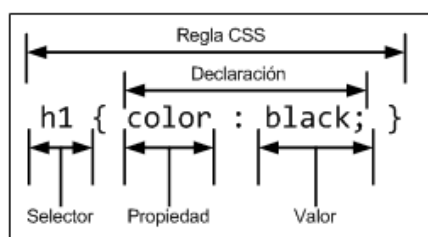
Una regla CSS asocia un bloque declarativo a uno o varios selectores.



- ▶ **Regla CSS:** cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de “selectores”, un símbolo de apertura “{”, otra parte denominada “bloque declarativo” y por último, un símbolo de cierre “}”.
- ▶ **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
- ▶ **Bloque declarativo:** formato o estilo que se aplicará al selector. Está compuesto por todas las declaraciones de la regla CSS.
- ▶ **Declaración:** especifica un estilo o formato de presentación. Asocia a una propiedad un valor concreto. Formato: **propiedad:valor;**
- ▶ **Propiedad:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
- ▶ **Valor:** establece el nuevo valor de la propiedad.

Tanto las propiedades como los valores de las declaraciones están predefinidos en CSS (es decir, no se pueden inventar).

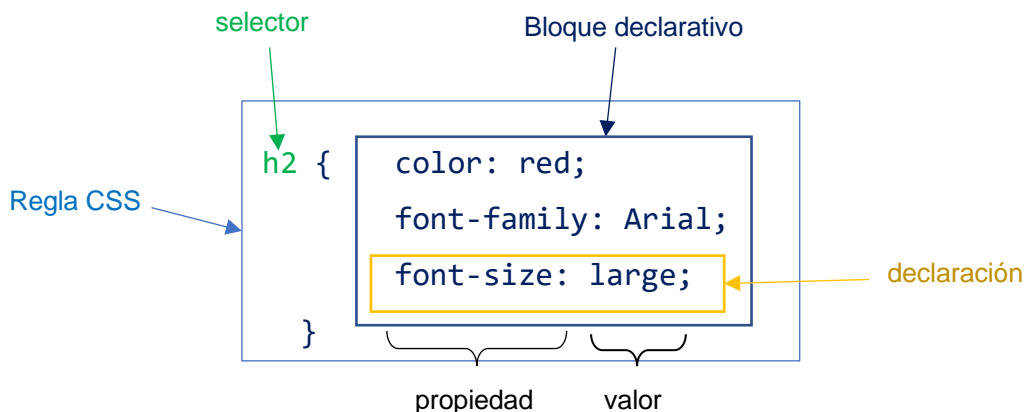
Ejemplo de regla CSS con una única declaración:



Vamos a estudiar el siguiente ejemplo:

```
h2 { color: red;
      font-family: Arial;
      font-size: large;
    }
```

Se trata de una regla CSS en la que se asocia, a todos los elementos <h2> de la página html, un bloque declarativo compuesto por tres declaraciones distintas. En concreto: se les aplicará un color de texto rojo y un tipo de letra Arial de tamaño grande.



#### ▪ Como interpretar la documentación oficial de las reglas CSS

Existen muchas propiedades CSS y cada una ellas puede tener una serie de valores distintos, con distintos formatos.

Para escribir código CSS es muy frecuente tener que recurrir a documentación.

Es por ello que se debe conocer cómo se describe el formato de las propiedades CSS formalmente, así se podrá interpretar la documentación.

Algo fundamental es la lista de posibles valores que admite la propiedad. Para definir la lista de valores permitidos se sigue un formato estándar basado en expresiones regulares que aporta mucha información de un vistazo.

Si el valor permitido se indica como una sucesión de palabras sin ningún carácter que las separe (paréntesis, comas, barras, etc.) el valor de la propiedad se debe indicar tal y como se muestra y con esas palabras en el mismo orden.

Vamos a analizar los posibles valores que puede adoptar la propiedad CSS “color”, que especifica el color de la letra.

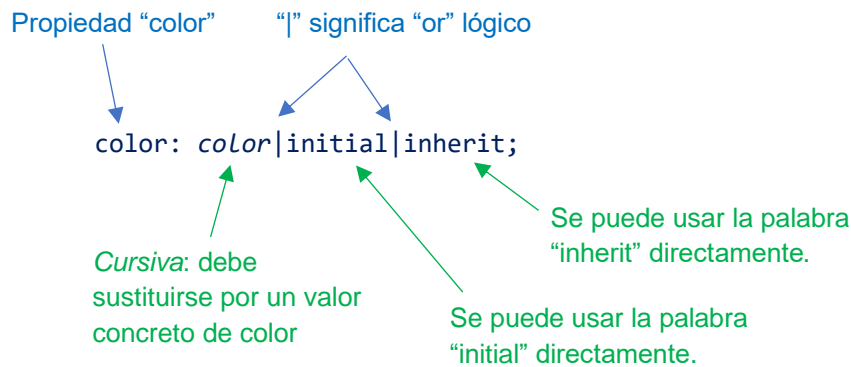
Accede a la documentación de la propiedad: [https://www.w3schools.com/csSref/pr\\_text\\_color.asp](https://www.w3schools.com/csSref/pr_text_color.asp)

#### CSS Syntax

Valores que admite

```
color: color|initial|inherit;
```





Posibles declaraciones correctas de la propiedad color:

```
color: black;
color: initial;
color: inherit;
color: red;
```

En la documentación aparece qué significa cada uno de los valores:

Property Values

Value	Description	Demo
<i>color</i>	Specifies the text color. Look at <a href="#">CSS Color Values</a> for a complete list of possible color values.	<a href="#">Demo &gt;</a>
initial	Sets this property to its default value. <a href="#">Read about initial</a>	
inherit	Inherits this property from its parent element. <a href="#">Read about inherit</a>	

Annotations from the image:

- Distintos formatos de *color***: Points to the *color* value in the table.
- Pruebas de funcionamiento**: Points to the [Demo >](#) button.

## ▪ Shorthand

Normalmente una propiedad admite un único valor, por ejemplo:

```
color: red;
```

Pero a veces existen propiedades compuestas, esto es, propiedades que agrupan varias propiedades en una, y por lo tanto admiten varios valores. Por ejemplo:

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

La propiedad border realmente está compuesta por tres propiedades:

```
border width: 1px;
border-style: solid;
border-color: black;
```

### ▪ Validadores CSS

Para comprobar si las reglas CSS están bien escritas, existen múltiples validadores CSS online. Por ejemplo <http://www.css-validator.org/validator.html.es>

Abre el validador y en “entrada directa” escribe:

```
h2 {      color: red;
          font-family: Arial;
          font-size: large;
        }
```

Prueba a modificar la sintaxis para comprobar los posibles errores.

## Actividad 1: Consultar documentación: valores de las propiedades CSS

1.1 Propiedad CSS simple

1.2 Propiedad CSS compuesta (shorthand)

## 3. Incorporar CSS a un documento HTML

Existen tres métodos para incorporar estilo CSS a una página web HTML:

### 3.1 Atributo HTML style

Cualquier etiqueta HTML admite el atributo style. El valor que se da a ese atributo es la propiedad o propiedades CSS que se desean añadir.

Ejemplo:

```
<p style="color:blue;">Esto es un párrafo</p>
```

Este método **no está recomendado**, ya que mezcla HTML con CSS, de forma que pierde muchas de las ventajas CSS:

- No separa la información del formato de presentación
- Mayor dificultad a la hora de interpretar y modificar el código
- No evita duplicidad de código CSS

Se emplea sólo en casos muy concretos, por ejemplo cuando no se tiene acceso al archivo .css.

## Actividad 2: integrar CSS con el atributo HTML style

### 3.2 Etiqueta HTML <style>

Se puede incorporar el elemento HTML <style> para dar formato a todo el documento HTML. Este elemento se coloca dentro de la cabecera HTML, es decir, entre las etiquetas <head> y </head>.

Es necesario indicar el selector al que se aplicará la regla CSS.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p{ color:blue; }
</style>
</head>
<body>
  <p>Esto es un párrafo</p>
</body>
</html>
```

#### Actividad 3: integrar CSS con la etiqueta HTML <style>

### 3.3 En un fichero CSS independiente

El código CSS se define en un archivo con extensión .css.

En HTML es necesario asociar ese fichero .css al documento mediante la siguiente etiqueta, que se incluirá en la cabecera:

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

El código CSS que contiene el fichero .css es idéntico al que se pondría dentro de las etiquetas <style>

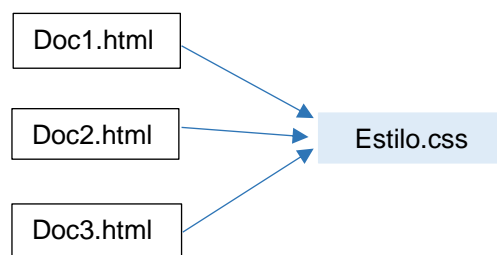
Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Pruebas CSS</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href= "miestilo.css">
</head>
<body>
  <p>Esto es un párrafo</p>
</body>
</html>
```

Fichero  
miestilo.css

p{color:blue;}

Es el método más aconsejado en las webs profesionales, ya que independiza completamente el código CSS del HTML, y permite utilizar el mismo fichero CSS para distintos documentos.



## Actividad 4: integrar CSS en un fichero independiente

4.1 Crear el fichero CSS

4.2 Añadir declaraciones a una regla css

4.3 Añadir reglas css nuevas

## 4. Selectores

### 4.1 Definición de selector

El selector es la parte de la regla css que **indica a qué elementos html se van a aplicar las declaraciones de formato**.

Recuerda dónde aparece el selector en el [formato de la reglas CSS](#).

A un mismo elemento HTML se le pueden aplicar varias reglas CSS y cada regla CSS puede aplicarse a un número ilimitado de elementos. En otras palabras, una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

Existen distintos tipos de selector, y en algunos casos pueden existir problemas de compatibilidad en algunos navegadores.

Información completa sobre selectores de nivel 3 con ejemplos:

<https://www.w3.org/TR/selectors-3/>

- Para trabajar este apartado en clase, vamos a trabajar con el fichero **selectores.html**, que puedes descargar del Aula Virtual.

DIBUJA EL ÁRBOL DE ELEMENTOS HTML DE ESE FICHERO

## 4.2 Tipos de selectores

### ▪ Selector universal

Un \* indica que las declaraciones se van a aplicar a todos los elementos html. Hay que utilizar este selector con precaución.

Ejemplo:

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

No se suele utilizar este selector de forma aislada, sino en combinación con otros.

- ▶ Aplica a todo el documento selectores.html el color rojo para la letra

### ▪ Selector de etiqueta

Es el que hemos visto hasta ahora. Se indica la etiqueta html sobre la que se quiere aplicar la regla.

Ejemplo:

```
p {  
    color: red;  
    font-family: arial;  
}
```

- ▶ Aplica al documento selectores.html el color rojo para la letra sólo en los párrafos

### ▪ Selector de agrupación

Cuando se indican varios selectores separados por comas, significa que la regla se aplica a todos ellos de igual forma.

Por ejemplo:

```
p { color: red;  
    font-family: arial; }  
h2{ color: red;  
    font-family: arial; }
```

Equivale a



```
p, h2 { color: red;  
        font-family: arial; }
```

- ▶ Aplica al documento selectores.html el color rojo para la letra en los párrafos y en la tabla

### ▪ Selector descendente

Se representa mediante dos elementos separados por un espacio.

Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

Ejemplo. Se tiene el código html:

```
<p>Esto es un <em>párrafo</em></p>
<ul>
  <li>Primer <em>elemento</em></li>
  <li>Segundo elemento</li>
</ul>
```

Regla CSS:

“em” dentro  
de “p”



```
p em { color: red;
        font-family: arial; }
```

Resultado:

Esto es un **párrafo**

- Primer *elemento*
- Segundo elemento

- Puede usarse una concatenación de selectores descendentes. Por ejemplo:

```
p a span em { color: red;}
```

El estilo de la regla anterior se aplica a los elementos de tipo `<em>` que se encuentren dentro de elementos de tipo `<span>`, que a su vez se encuentren dentro de elementos de tipo `<a>` que se encuentren dentro de elementos de tipo `<p>`.

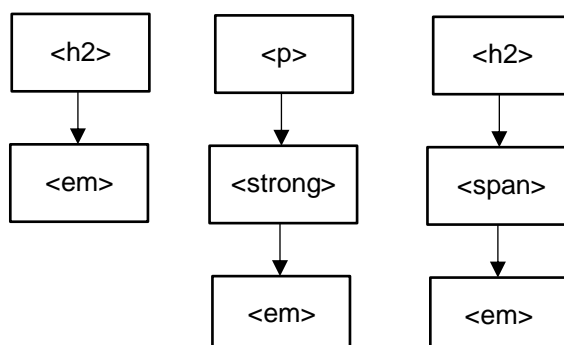
- No debe confundirse el selector descendente con la combinación de selectores:

```
p, a, span, em { color: red;}
```

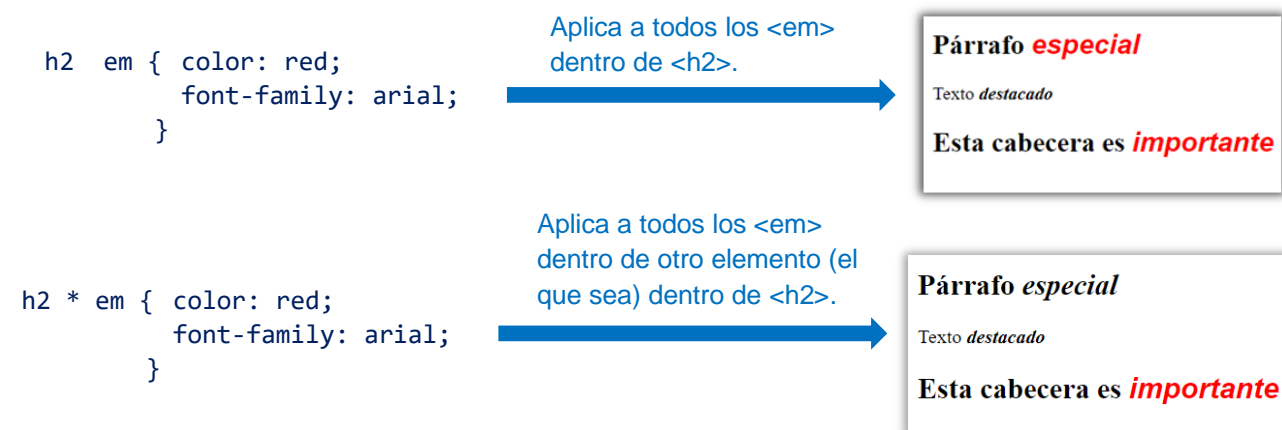
El estilo de la regla anterior se aplica a todos los elementos de tipo `<em>`, `<span>`, `<a>` y `<p>`.

- A veces se utiliza el selector descendente en combinación con el selector universal. Por ejemplo, en el código html:

```
<h2>Párrafo <em>especial</em></h2>
<p>Texto <strong><em>destacado</em></strong></p>
<h2>Esta <span class="myclass">cabecera es <em>importante </em></span></h2>
```



Observa las diferencias:



- ▶ Aplica al documento selectores.html el color rojo para la letra solo en los elementos `<u>` que estén dentro de un párrafo.
- ▶ Aplica al documento selectores.html el color rojo para la letra solo en los elementos `<u>` que estén dentro de otro (`<p>` o `<h3>` en este caso). "Fin" debe quedar fuera de la regla. Emplea una única regla y un único selector en ella.

#### ▪ Selector de clase

Se representa mediante un punto (.) inicial.

Una forma para aplicar estilos a un solo elemento de la página, o a un conjunto de ellos, consiste en utilizar el [atributo HTML class](#).

El nombre de la clase no está prefijado (puede inventarse).

Esto es recomendable solo en caso puntuales, ya que obliga a modificar el código HTML, y no se aplicará a futuros nuevos elementos.

Ejemplo:

```
<p>Esto es un <em>párrafo</em></p>
<p>
<ul>
  <li>Primer <em>elemento</em></li>
  <li class="destacado">Segundo <em>elemento</em></li>
  <li class="destacado">Tercer <em>elemento</em></li>
</ul>
</p>
```

Se define la regla CSS:



- Se puede definir un atributo class con más de un valor, de la forma:

```
<p class="especial destacado">Párrafo 1</p>
<p class="destacado">Párrafo 2</p>
```

- Es posible combinar el selector de clase con otros tipos de selectores. Ejemplos:

<code>p.destacado</code>	todos los párrafos con <code>class="destacado"</code>
<code>p .destacado</code>	todos los elementos con <code>class="destacado"</code> que además estén dentro de un párrafo
<code>p, .destacado</code>	todos los párrafos y todos los elementos con <code>class="destacado"</code>

- Aplica al documento `selectores.html` el color rojo para los elementos con la clase "coloreado".

## Selector de id

Se representa mediante un #.

Una forma para aplicar estilos a un solo elemento de la página, o a un conjunto de ellos, consiste en utilizar el [atributo HTML id](#). En una página **sólo debe asignarse un id concreto a un único elemento HTML**.

El nombre del id no está prefijado (puede inventarse).

Esto es recomendable solo en caso puntuales, ya que obliga a modificar el código HTML.

Ejemplo:

```
<p>Esto es un <em>párrafo</em></p>
<p>
<ul>
  <li>Primer <em>elemento</em></li>
  <li id="destacado">Segundo <em>elemento</em></li>
  <li>Tercer <em>elemento</em></li>
</ul>
</p>
```

Se define la regla CSS:

```
#destacado { color: red; }
```

resultado

Esto es un párrafo

- Primer *elemento*
- **Segundo *elemento***
- Tercer *elemento*

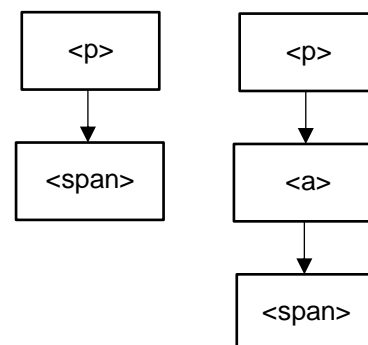


### ▪ Selector de hijos

Se utiliza para seleccionar un elemento que es **hijo directo** de otro elemento y se indica mediante el signo ">"

Ejemplo:

```
<p><span>Texto1</span></p>
<p><a href="#"><span>Texto2</span></a></p>
```



Se define la regla CSS:

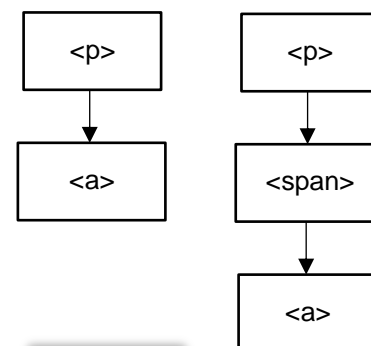
```
p > span { color: red; }
```

resultado

- No hay que confundir un selector descendente con un selector de hijos. En el primer caso, se aplica a los elementos que descienden de otro, sin importar el nivel de descendencia (hijos, nietos, etc), en cambio en el segundo caso aplica sólo a los elementos que descienden directamente, es decir a los hijos.

Ejemplo:

```
<p><a href="#">Enlace1</a></p>
<p><span><a href="#">Enlace2</a></span></p>
```



Observa las diferencias:

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

Aplica a todos los <a> dentro de <p> (al nivel que sea).

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

Aplica a todos los <a> que sean hijos directos de <p>.

- Aplica al documento selectores.html el color rojo para los elementos <i> que son hijos de un párrafo. (Solo deberían colorearse los números 1,2,3 y 4).

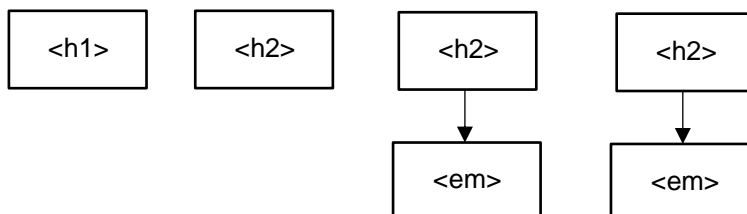
## ▪ Selector de hermanos

Se emplea el signo ~ (ASCII: alt 126).

Se utiliza para seleccionar un elemento que **está al mismo nivel que otro y está después de él** (a continuación o no).

Ejemplo:

```
<h1>Título principal</h1>
<h2>Título 1</h2>
<h2>Título <em>2</em></h2>
<h2>Título <em>3</em></h2>
```



Se define la regla CSS:

```
h1 ~ h2 { color: red; }
```

resultado

**Título principal**

Título 1

Título 2

Título 3

- Aplica al documento selectores.html el color rojo para los párrafos hermanos del párrafo con clase "coloreado".

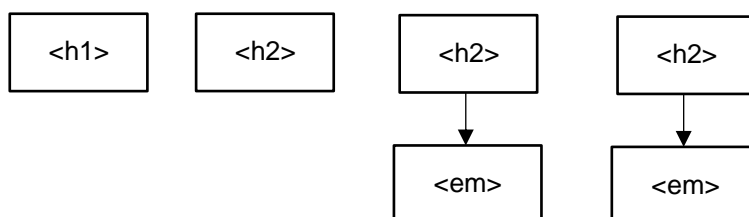
## ▪ Selector adyacente

Se emplea el signo más (+).

Se utiliza para seleccionar un elemento que **está al mismo nivel que otro y además va justo a continuación** ("siguiente hermano").

Ejemplo:

```
<h1>Título principal</h1>
<h2>Título 1</h2>
<h2>Título <em>2</em></h2>
<h2>Título <em>3</em></h2>
```



Se define la regla CSS:

```
h1 + h2 { color: red; }
```

resultado

**Título principal**

Título 1

Título 2

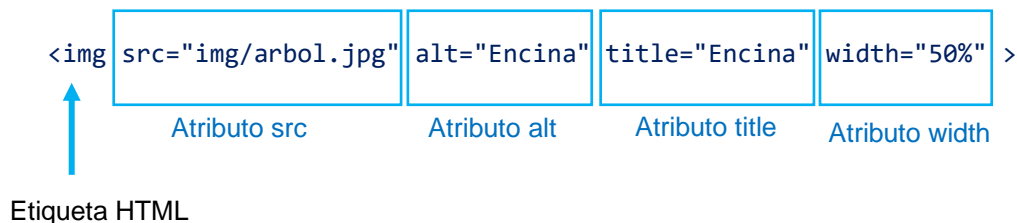
Título 3

- Aplica al documento selectores.html el color rojo para el párrafo adyacente al párrafo con clase "coloreado".

## ▪ Selector de atributos

Permiten seleccionar elementos HTML en función de sus atributos HTML y/o valores de esos atributos. El nombre del atributo en el selector se pone entre corchetes [ ].

Recordamos que un atributo HTML es aquel que se incluye en la etiqueta y se le da un valor, por ejemplo:

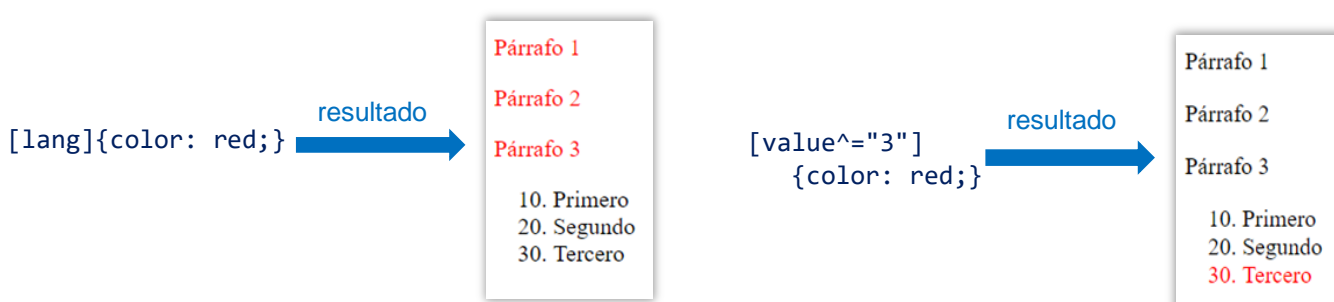


Hay varias formas de utilizar selectores de atributo:

<code>[atrib]</code>	Elementos que tienen ese atributo (sea cual sea su valor)
<code>[atrib = "valor"]</code>	Elementos que tienen ese atributo con ese valor exacto
<code>[atrib  = "valor"]</code>	Elementos que tienen ese atributo y el valor es una serie de palabras separadas por guiones (-), siendo la primera el valor indicado. Este formato se utiliza para el lenguaje (atributo lang).
<code>[atrib *= "valor"]</code>	Elementos que tienen ese atributo y el valor contiene el valor indicado.
<code>[atrib ^= "valor"]</code>	Elementos que tienen ese atributo y el valor comienza por el valor indicado.
<code>[atrib \$= "valor"]</code>	Elementos que tienen ese atributo y el valor termina por el valor indicado.

Ejemplos. Se tiene el código HTML:

```
<p lang="es-ES">Párrafo 1</p>
<p lang="es-AR">Párrafo 2</p>
<p lang="en-EN">Párrafo 3</p>
<ol>
  <li value="10">Primero</li>
  <li value="20">Segundo</li>
  <li value="30">Tercero</li>
</ol>
```



```
[lang="es-AR"]  
{ color: red; }
```

resultado

Párrafo 1  
Párrafo 2  
Párrafo 3  
  
10. Primero  
20. Segundo  
30. Tercero

```
[value$="0"]  
{color: red;}
```

resultado

Párrafo 1  
Párrafo 2  
Párrafo 3  
  
10. Primero  
20. Segundo  
30. Tercero

```
[lang|="es"]  
{color: red;}
```

resultado

Párrafo 1  
Párrafo 2  
Párrafo 3  
  
10. Primero  
20. Segundo  
30. Tercero

```
[value*="2"]  
{color: red;}
```

resultado

Párrafo 1  
Párrafo 2  
Párrafo 3  
  
10. Primero  
20. Segundo  
30. Tercero

- ▶ Aplica al documento selectores.html el color rojo para los elementos que tengan el atributo "colspan".
- ▶ Aplica al documento selectores.html el color rojo para los elementos que tengan el atributo "colspan" con valor "2".

### ▪ Selector de pseudo-clase

Una pseudoclase aplicada a un elemento HTML delimita la selección sobre ese elemento: **se aplica sólo a elementos que cumplan una condición**. Se representa mediante dos puntos (:) siendo:

```
p:first-child { bloque declarativo css }
```

Los nombres de pseudoclasses están predefinidos (no pueden inventarse).

Por ejemplo, la pseudoclase `:first-child` selecciona el primer hijo de una serie de elementos.

Algunas pseudoclasses aplicadas:

<code>p:first-child</code>	elemento <code>&lt;p&gt;</code> que sea el primer hijo de otro elemento
<code>p:last-child</code>	elemento <code>&lt;p&gt;</code> que sea el último hijo de otro elemento
<code>p:nth-child(n)</code>	elemento <code>&lt;p&gt;</code> que sea el hijo número <code>n</code> de otro elemento
<code>p:nth-child(odd)</code>	elemento <code>&lt;p&gt;</code> que sea hijo impar de otro elemento
<code>p:nth-child(even)</code>	elemento <code>&lt;p&gt;</code> que sea hijo par de otro elemento
<code>p:first-of-type</code>	elemento <code>&lt;p&gt;</code> que sea el primero del tipo
<code>p:last-of-type</code>	elemento <code>&lt;p&gt;</code> que sea el último del tipo
<code>p:nth-of-type</code>	elemento <code>&lt;p&gt;</code> que sea el número <code>n</code> de ese tipo
<code>p:nth-of-type(odd)</code>	elemento <code>&lt;p&gt;</code> que sea número impar
<code>p:nth-of-type(even)</code>	elemento <code>&lt;p&gt;</code> que sea número par
<code>p:empty</code>	elemento <code>&lt;p&gt;</code> vacío (sin elementos dentro)
<code>a:hover</code>	elemento <code>&lt;a&gt;</code> sobre el que se ha puesto el ratón
<code>a:visited</code>	elemento <code>&lt;a&gt;</code> sobre que ya ha sido visitado (es enlace)
<code>a:link</code>	elemento <code>&lt;a&gt;</code> que aún no ha sido visitado (es enlace)
<code>input:focus</code>	elemento <code>&lt;input&gt;</code> con el foco puesto (cursor encima)
<code>input:invalid</code>	elemento <code>&lt;input&gt;</code> que es inválido (no cumple validación)
<code>input:valid</code>	elemento <code>&lt;input&gt;</code> que es válido (cumple validación)

La pseudoclase negación `:not()` selecciona en caso contrario. Por ejemplo:

`p:not(.destacado)`      elemento `<p>` que no tenga la clase “destacado”

## Ejemplo 1:

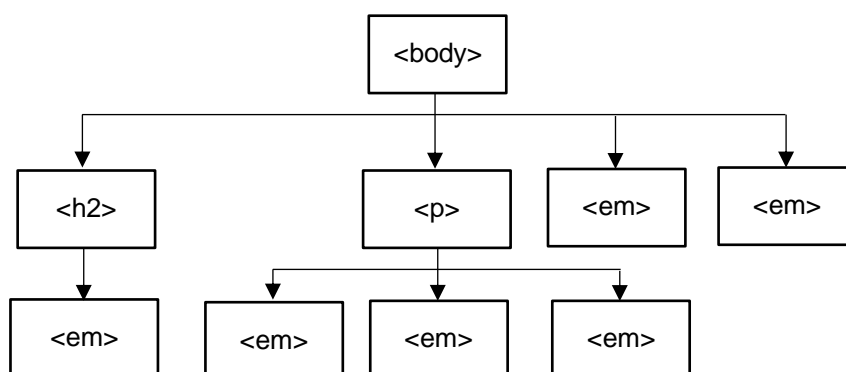
```
<h2>Las <em>pseudoclases</em></h2>
<p>En este <em>párrafo</em> hay varias <em>palabras</em> con distinto <em>formato</em>
</p>
<em>Palabra1</em>
<em>Palabra2</em>
```

Se visualiza:

**Las *pseudoclases***

En este *párrafo* hay varias *palabras* con distinto *formato*

*Palabra1 Palabra2*



A continuación, se muestran una serie de selectores y el resultado:

`em:first-child {color:red;}` → resultado

**Las *pseudoclases***

En este *párrafo* hay varias *palabras* con distinto *formato*  
*Palabra1 Palabra2*

`em:last-child {color:red;}` → resultado

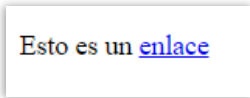
**Las *pseudoclases***

En este *párrafo* hay varias *palabras* con distinto *formato*  
*Palabra1 Palabra2*

## Ejemplo 2:

```
<p>Esto es un <a href="#">enlace</a></p>
```

Se visualiza:



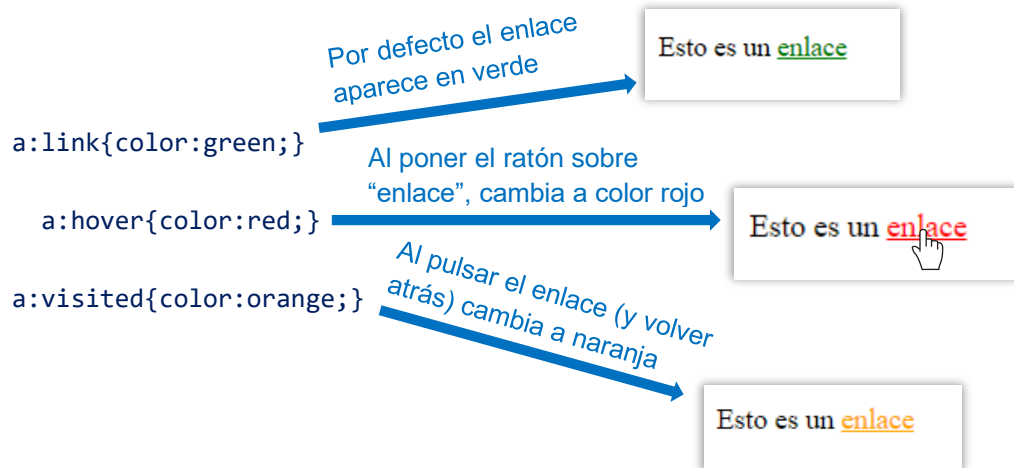
```
em:nth-child(2) {color:red;}
```

resultado

**Las pseudoclases**

En este párrafo hay varias *palabras* con distinto formato  
Palabra1 Palabra2

A continuación se muestran una serie de selectores y el resultado:



## Ejemplo 3:

```
<form method="get" action="">
  Nombre:<input type="text" pattern="[A-Z]\w+"><br><br>
  Apellidos:<input type="text"><br><br>
  <input type="submit">
</form>
```

Se visualiza:

Nombre:

Apellidos:

A continuación se muestran una serie de selectores y el resultado:

`input:focus{background-color:yellow;}`

Al situar el cursor sobre un campo, se pone amarillo

`input:invalid{color:red;}`

Al escribir un valor correcto en el campo la letra se ve verde (según su pattern)

`input:valid{color:green;}`

Al escribir un valor incorrecto en el campo la letra se ve roja (según su pattern)

Nombre:

Apellidos:

Nombre:

Apellidos:

Nombre:

Apellidos:

- ▶ Aplica al documento selectores.html el color rojo para la primera fila de la tabla.
- ▶ Aplica al documento selectores.html el color rojo para el segundo párrafo.
- ▶ Aplica al documento selectores.html el color rojo para la primera columna de la tabla.



## ▪ Selector de pseudo-elemento

Un pseudoelemento aplicado a un elemento delimita la selección sobre ese elemento: [se aplica sólo a una parte del elemento](#). Se representa mediante cuatro puntos (::).

Los nombres de pseudoelementos están predefinidos (no pueden inventarse).

Por ejemplo, el pseudoelemento :first-letter selecciona la primera letra de un elemento.

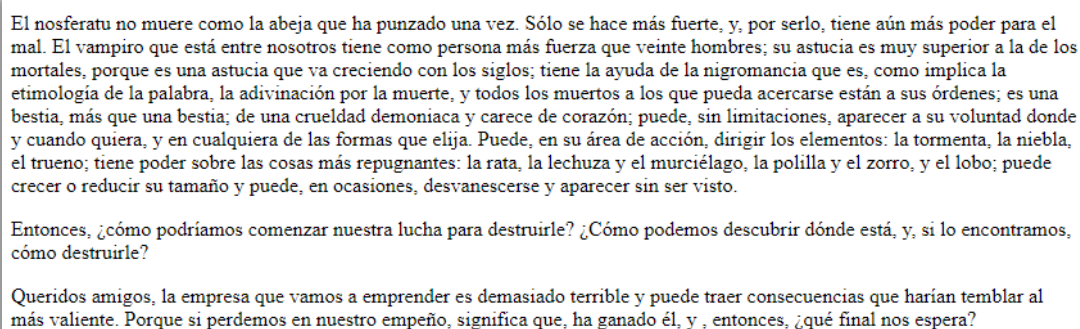
Algunos pseudoatributos aplicados:

p::first-letter	primera letra del elemento <p>
p::first-line	primera línea del elemento <p>
p::before	antes del elemento <p>
p::after	después del elemento <p>

### Ejemplo 1:

```
<p>El nosferatu no muere como la abeja que ha punzado una vez. Sólo se hace más fuerte, y, por serlo, tiene aún más poder para el mal. El vampiro que está entre nosotros tiene como persona más fuerza que veinte hombres; su astucia es muy superior a la de los mortales, porque es una astucia que va creciendo con los siglos; tiene la ayuda de la nigromancia que es, como implica la etimología de la palabra, la adivinación por la muerte, y todos los muertos a los que pueda acercarse están a sus órdenes; es una bestia, más que una bestia; de una crueldad demoniaca y carece de corazón; puede, sin limitaciones, aparecer a su voluntad donde y cuando quiera, y en cualquiera de las formas que elija. Puede, en su área de acción, dirigir los elementos: la tormenta, la niebla, el trueno; tiene poder sobre las cosas más repugnantes: la rata, la lechuza y el murciélago, la polilla y el zorro, y el lobo; puede crecer o reducir su tamaño y puede, en ocasiones, desvanecerse y aparecer sin ser visto.</p>
<p>Entonces, ¿cómo podríamos comenzar nuestra lucha para destruirle? ¿Cómo podemos descubrir dónde está, y, si lo encontramos, cómo destruirle?</p>
<p>Queridos amigos, la empresa que vamos a emprender es demasiado terrible y puede traer consecuencias que harían temblar al más valiente. Porque si perdemos en nuestro empeño, significa que, ha ganado él, y , entonces, ¿qué final nos espera?</p>
```

Se visualiza:



El nosferatu no muere como la abeja que ha punzado una vez. Sólo se hace más fuerte, y, por serlo, tiene aún más poder para el mal. El vampiro que está entre nosotros tiene como persona más fuerza que veinte hombres; su astucia es muy superior a la de los mortales, porque es una astucia que va creciendo con los siglos; tiene la ayuda de la nigromancia que es, como implica la etimología de la palabra, la adivinación por la muerte, y todos los muertos a los que pueda acercarse están a sus órdenes; es una bestia, más que una bestia; de una crueldad demoniaca y carece de corazón; puede, sin limitaciones, aparecer a su voluntad donde y cuando quiera, y en cualquiera de las formas que elija. Puede, en su área de acción, dirigir los elementos: la tormenta, la niebla, el trueno; tiene poder sobre las cosas más repugnantes: la rata, la lechuza y el murciélago, la polilla y el zorro, y el lobo; puede crecer o reducir su tamaño y puede, en ocasiones, desvanecerse y aparecer sin ser visto.

Entonces, ¿cómo podríamos comenzar nuestra lucha para destruirle? ¿Cómo podemos descubrir dónde está, y, si lo encontramos, cómo destruirle?

Queridos amigos, la empresa que vamos a emprender es demasiado terrible y puede traer consecuencias que harían temblar al más valiente. Porque si perdemos en nuestro empeño, significa que, ha ganado él, y , entonces, ¿qué final nos espera?

A continuación se aplican dos selectores y se muestra el resultado:

```
p::first-line{color:red;}
p::first-letter{font-size:xx-large;}
```

**E**l nosferatu no muere como la abeja que ha punzado una vez. Sólo se hace más fuerte, y, por serlo, tiene aún más poder para el mal. El vampiro que está entre nosotros tiene como persona más fuerza que veinte hombres; su astucia es muy superior a la de los mortales, porque es una astucia que va creciendo con los siglos; tiene la ayuda de la nigromancia que es, como implica la etimología de la palabra, la adivinación por la muerte, y todos los muertos a los que pueda acercarse están a sus órdenes; es una bestia, más que una bestia; de una crueldad demoníaca y carece de corazón; puede, sin limitaciones, aparecer a su voluntad donde y cuando quiera, y en cualquiera de las formas que elija. Puede, en su área de acción, dirigir los elementos: la tormenta, la niebla, el trueno; tiene poder sobre las cosas más repugnantes: la rata, la lechuza y el murciélago, la polilla y el zorro, y el lobo; puede crecer o reducir su tamaño y puede, en ocasiones, desvanecerse y aparecer sin ser visto.

**E**ntonces, ¿cómo podríamos comenzar nuestra lucha para destruirle? ¿Cómo podemos descubrir dónde está, y, si lo encontramos, cómo destruirle?

**Q**ueridos amigos, la empresa que vamos a emprender es demasiado terrible y puede traer consecuencias que harían temblar al más valiente. Porque si perdemos en nuestro empeño, significa que, ha ganado él, y , entonces, ¿qué final nos espera?

### Ejemplo 2:

```
<p>Párrafo 1</p>
<p>Párrafo 2</p>
<p>Párrafo 3</p>
```

Se visualiza:

Párrafo 1  
Párrafo 2  
Párrafo 3

A continuación se aplica un selector y se muestra el resultado:

```
p::before{content:"->";}
```

->Párrafo 1  
->Párrafo 2  
->Párrafo 3

- Aplica al documento selectores.html el color rojo para la primera letra de cada párrafo.

## 4.3 Combinación de selectores

CSS permite la combinación de uno o más tipos de selectores. Ejemplos:

```
div.aviso span.especial { ... }
```

Selecciona los elementos de tipo `<span>` que tengan una clase “especial” y que estén dentro de un elemento `<div>` que tiene una clase “aviso”

```
ul#menuPrincipal + li.destacado a.inicio { ... }
```

Selecciona los elementos `<a>` que tengan la clase “inicio” que estén dentro de un elemento `<li>` con clase “destacado” que esté al mismo nivel y a continuación de un elemento `<ul>` con id “menuPrincipal”

```
table td, table th { . . . }
```

Selecciona los elementos `<th>` y `<td>` que estén dentro de `<table>`

```
table thead th { . . . }
```

Selecciona los elementos `<th>` que estén dentro de `<thead>` que estén dentro de `<table>`

```
table tbody td + td { . . . }
```

Selecciona los `<td>` que estén precedidos de otro `<td>` (hermanos adyacentes), los cuales estén dentro de `<tbody>` que a su vez está dentro de `<table>`

```
table tbody td:last-child { . . . }
```

Selecciona los `<td>` que son último hijo y además están dentro de `<tbody>` que está dentro de `<table>`

```
.with-currency[lang="en-US"] td:last-child::before {  
  content: '$';  
}
```

Selecciona la parte anterior de los `<td>` que son último hijo y además están dentro de un elemento que tiene la clase “with-currency” y el atributo lang con valor “en-US”.

Lo que hace esta regla es añadir \$ al principio del elemento.

```
.with-currency[lang="fr"] td:last-child::after {  
  content: ' €';  
}
```

Selecciona la parte posterior de los `<td>` que son último hijo y además están dentro de un elemento que tiene la clase “with-currency” y el atributo lang con valor “fr”.

Lo que hace esta regla es añadir € al final del elemento.

## 4.4 Resumen de selectores

TIPO DE SELECTOR	REPRESENTACIÓN	ELEMENTO/S SELECCIONADO/S
Selector universal	*	Aplica a todos los elementos
Selector de etiqueta	elem	Aplica a un elemento HTML. Ejemplo: p
Selector de agrupación	elem1, elem2	Aplica a varios elementos. Ejemplo: p, h1
Selector descendente	elem1 elem2	Aplica a un elemento que está dentro de otro (a cualquier nivel). Ejemplo: p em
Selector de clase	.clase	Aplica a los elementos que tienen esa clase asignada (class="miclase"). Ej: p.miclase
Selector de id	#id	Aplica al elemento que tiene ese id asignado (id="miid"). Debe ser único. Ej: #miid
Selector de hijos	elem1 > elem2	Aplica a todos los elementos elem2 que son hijos directos de elem1 (no nietos). Ej: p > em
Selector de hermanos	elem1 ~ elem2	Aplica a todos los elementos elem2 que son hermanos de elem1 (están al mismo nivel y van después). Ej: p ~ div
Selector adyacente	elem1 + elem2	Aplica al elemento elem2 adyacente a elem1 (está al mismo nivel y va justo a continuación). Ejemplo: p + div
Selector de atributos	[attr] tiene el atributo [attr=valor] tiene ese valor [attr =valor] empieza por valor- [attr*=valor] contiene el valor [attr^=valor] empieza por valor [attr\$=valor] termina por valor	Se seleccionan elementos según sus atributos
Selector de pseudo-clase	:first-child      :last-child :nth-child(n) :nth-child(odd)   :nth-child(even) :first-of-type    :last-of-type :nth-of-type(n) :nth-of-type(odd) :nth-of-type(even) :empty :hover :link                :visited :focus :invalid            :valid :not()	Se selecciona el elemento sólo si cumple una condición. Ejemplo:    p:nth-of-type(4) p:not(.miclase)
Selector de pseudo-elemento	::first-letter ::first-line ::before ::after	Se selecciona sólo una zona del elemento Ejemplo: p::first-letter

- En selectores.html intenta averiguar el elemento seleccionado con las siguientes reglas antes de hacer la prueba.

```

h3,u           {color:red;}
h3 u           {color:red;}
tr td          {color:red;}
.coloreado+p   {color:red;}
.coloreado+p   {color:red;}
td:first-child {color:red;}
td:last-child  {color:red;}
td:nth-child(odd) {color:red;}
p:first-child  {color:red;}
td~td         {color:red;}
h3+p          {color:red;}
h3+*          {color:red;}
p:hover       {color:red;}

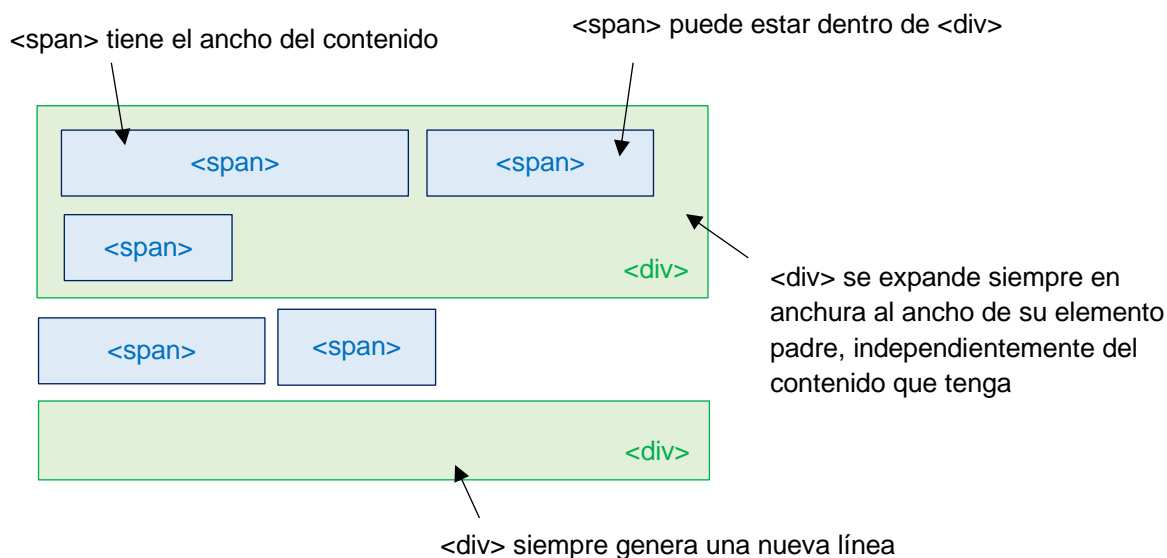
```

## 4.5 Elementos HTML <div> y <span> para formato

En ocasiones se quiere aplicar formato a una sección de la página web que no está delimitada por una etiqueta HTML. En estos casos se pueden usar las etiquetas HTML <div> y <span>, que no dan información sobre la semántica ni la visualización del texto, sino que son configurables por el programador.

Nota: es recomendable no abusar de estas etiquetas.

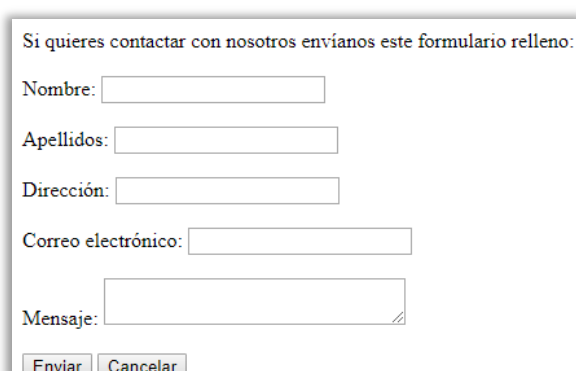
La diferencia principal entre ambas es que <div> es una etiqueta en bloque y <span> en línea.



Ejemplo. Se tiene el siguiente código HTML:

```
<form method="get" action="">
  Si quieres contactar con nosotros envíanos este formulario relleno: <br><br>
  Nombre: <input type="text" name="nombre"><br>
  Apellidos: <input type="text" name="apellidos"><br>
  Dirección: <input type="text" name="direccion"><br>
  Correo electrónico: <input type="text" name="correo"><br>
  Mensaje: <textarea name="mensaje" cols=30 rows=2></textarea>
  <br><br>
  <input type="submit" value="Enviar">
  <input type="reset" value="Cancelar">
</form>
```

Que tiene como resultado:



Se quiere poner con fondo naranja ese texto utilizando la clase "naranja":

```
.naranja{background-color:orange;}
```

¿A qué elemento se le puede asignar esa clase?

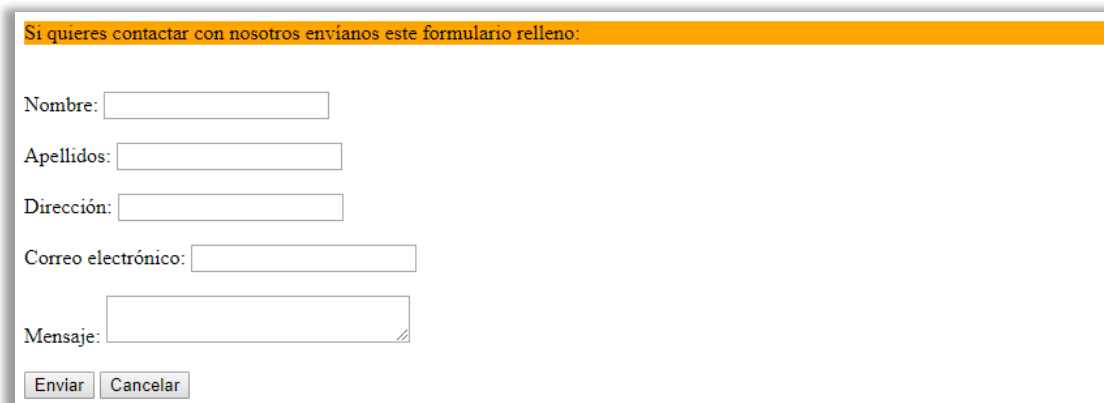
Una posible solución sería por ejemplo definir la frase como un párrafo con <p>, pero eso puede conllevar que aplique el estilo de párrafo del navegador, y puede que no interese.

Otra solución sería definir un <div> o <span> específico para esa frase.

Si se define un <div>:

```
<div class="naranja">Si quieres contactar con nosotros envíanos este formulario relleno: </div>
```

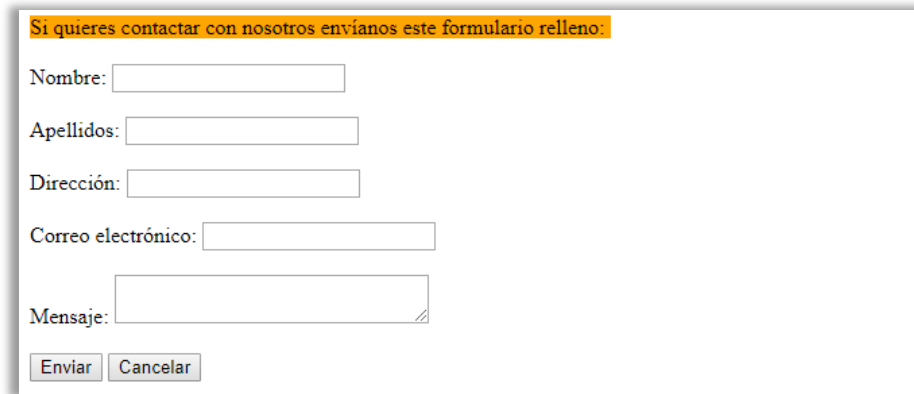
El resultado sería:



Si se define un `<span>`:

```
<span class="naranja">Si quieres contactar con nosotros envíanos este formulario  
relleno: </span>
```

El resultado sería:



Si quieres contactar con nosotros envíanos este formulario relleno:

Nombre:

Apellidos:

Dirección:

Correo electrónico:

Mensaje:

- ▶ La diferencia entre `<div>` y `<span>` es que el primero rellena la línea todo el ancho visible, mientras que el segundo sólo la rellena en el texto. Elegir uno u otro dependerá del diseño de la página web.

## Actividad 5: selectores

### 5.1 Añadir selectores básicos

### 5.2 Añadir reglas CSS

Diner

## 5. Herencia y cascada

### 5.3 Herencia

Una de las características principales de CSS es la herencia de los estilos definidos para los elementos. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Herencia</title>
  <meta charset='utf-8'>
</head>
<body>
  <p>Párrafo 1 <span>palabra1</span>
    <span>palabra2</span> </p>
  <p>Párrafo 2</p>
</body>
</html>
```

*body{color:blue;}  
<span> hereda de su  
padre, que es <p>*

Párrafo 1 palabra1 palabra2  
Párrafo 2

*body{color:blue;}  
p{color:orange;}  
<p> y <span> heredan el  
estilo del padre, que es  
<body>*

Párrafo 1 palabra1 palabra2  
Párrafo 2

En un documento HTML todo tiene ascendiente excepto el elemento <html>. Así, en el ejemplo, <body> hereda de <html>, <p> hereda de <body> y <span> hereda de <p>.



Sin embargo, **no todas las propiedades se heredan**.

Los creadores de CSS pensaron que sería útil que algunas propiedades se heredaran porque suelen mantenerse constantes dentro de una caja. Por ejemplo, lo más normal es usar un tipo de letra en un contenedor y no mezclar distintos tipos de letra. En cambio, para otras propiedades era poco lógico que se heredaran, como los márgenes, ya que por ejemplo las fotografías dentro de un artículo normalmente tenían un margen distinto al que tenía el propio artículo respecto a su contenedor o los párrafos.




## ▪ Inherited

La característica “**inherited**” de una propiedad CSS indica si una propiedad es heredada o no por defecto.

## Definition and Usage

The **color** property specifies the color of text.

**Tip:** Use a background color combined with a text color that makes the text easy to read.

<b>Default value:</b>	<i>not specified</i>
<b>Inherited:</b>	yes 
<b>Animatable:</b>	yes. <a href="#">Read about animatable</a>
<b>Version:</b>	CSS1
<b>JavaScript syntax:</b>	<i>object.style.color="#0000FF"</i>

Por ejemplo la propiedad “font-family” hereda de su padre, pero los márgenes (por ejemplo la propiedad “margin-top”) no se heredan excepto si se indica explícitamente que deben heredarse.

- Busca en la documentación w3c el valor de la característica “inherited” para las propiedades Font-family y margin-top.

## ▪ Inherit e initial

Para cambiar el comportamiento de una propiedad respecto a la herencia, existen 2 valores de la propiedad que pueden usarse:

- ➡ **inherit** La propiedad se heredará del padre
- ➡ **initial** La propiedad no se heredará, sino que se pondrá su valor por defecto (si no tiene valor por defecto, se heredará, en ese caso initial=inherit)

## CSS Syntax

```
color: color|initial|inherit;
```

Por ejemplo:

```
p {color:inherit;}
```

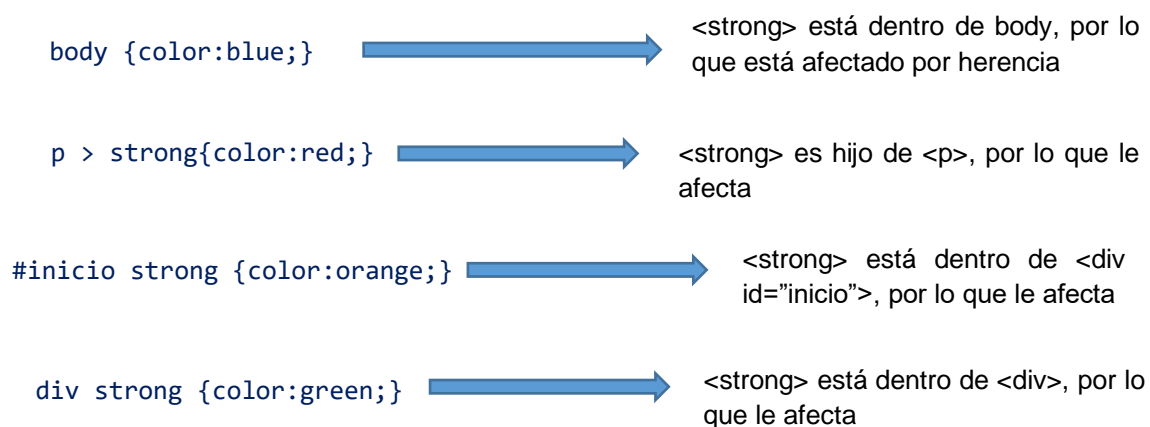
## 5.4 Cascada

### ▪ Conflicto

Vamos a analizar el siguiente ejemplo. ¿De qué color se mostrará la palabra “relevante”? **NO** hagas la prueba en un navegador, solo intenta predecirlo.

```
<!DOCTYPE html>
<html>
<head>
  <title>Práctica CSS 1</title>
  <meta charset='utf-8'>
  <style>
    body {color:blue;}
    p > strong { color:white;
                  background-color:black;}
    #inicio strong{color:orange;}
    div strong   {color:green;}
  </style>
</head>
<body>
  <div id="inicio">
    <p>Aquí hay información <strong>relevante</strong></p>
  </div>
</body></html>
```

Podemos ver que **hay conflicto** con respecto al color:



Todas estas situaciones se denominan **situaciones de conflicto o colisión** y CSS define una forma de resolución denominado “cascada”.

La **cascada** se basa en determinar todas las reglas que son de aplicación a un elemento y que entran en conflicto, en ordenar esas reglas en base a unos criterios y mostrar como resultado en el navegador la regla que haya quedado en primer lugar, a la que denominamos “**regla ganadora**”.

La cascada define qué reglas se aplican sobre un elemento cuando hay conflicto de selectores.

Hay que tener en cuenta que **este cálculo de prioridad entra en juego sólo cuando hay conflicto en una propiedad**, es decir, hay más de una regla CSS que aplicaría a un elemento y en esas reglas se da valor a una misma propiedad (valores contradictorios). Si las propiedades son distintas no hay problema, se aplican todas.

En el ejemplo anterior vemos que no hay conflicto con la propiedad background-color.

## ■ Reglas de la cascada

Los criterios para elegir la “regla CSS ganadora” son los siguientes, aplicados en orden:

### 1) Importancia

Se añade “!important” a una declaración dentro de un bloque declarativo, y esa declaración siempre prevalece por encima de todo. Justificado sólo para casos excepcionales.

```
.clase_1 {
  background-color: gray;
  border: none !important;
}
```

### 2) Especificidad

Hay unos selectores más específicos que otros. Cuanto más específico, más prioridad.

**Especificidad** = medida de lo específico que es un selector.

Existe una fórmula para calcular especificidad. Se define una tabla con 4 posiciones que equivalen al valor posicional en un número: millares, centenas, decenas y unidades. En cada posición hay elementos que suman puntos:

1000	100	10	1
Style (atributo HTML)	Selector ID (#id)	Selector clase (.clase)	Selector de elemento
		Selector de atributo ([attr])	Selector pseudo-elemento (::pseudoelemento)
		Selector pseudo-clase (:pseudoclase)	

Nota: No afectan a especificidad: \* (selector universal), combinaciones de selectores ((espacio), >, ~, +) ni la negación (:not).

Ejemplos:

	1000	100	10	1	Especificidad
h1 {color:red;}	0	0	0	1	0001
#destacado {color:red;}	0	1	0	0	0100
h1 + p::first-letter {color:red;}	0	0	0	3	0003
li > a[href="en-US"] > .aviso {color:red;}	0	0	2	2	0022
<p style="div > div > a:hover { color:red;}">	1	0	1	3	1013

Hay que tener en cuenta que **este cálculo de prioridad entra en juego sólo cuando hay conflicto en una propiedad**, es decir, hay más de una regla CSS que aplicaría a un elemento y en esas reglas se da valor a una misma propiedad (valores contradictorios). Si las propiedades son distintas no hay problema, se aplican todas.

### 3) Orden del código

Cuando hay igualdad en importancia y especificidad, tiene más prioridad la regla que esté más abajo (se ha escrito más recientemente).

```
p { color: blue; }  
p { color: red; } /*se aplicaría esta*/
```

- Aplica las reglas de la cascada al [ejemplo con conflicto de arriba](#), intentando predecir qué regla aplicará finalmente el navegador. Después comprueba la respuesta.

## Actividad 6. Cascada

- 6.1 Calcular la especificidad
- 6.2 Calcular la especificidad
- 6.3 Cascada
- 6.4 Cascada

## 6. Modelo de cajas

### 6.1 Definición del modelo

El diseño de las páginas web está basado en un [modelo de cajas](#): todos los elementos de las páginas se representen mediante cajas rectangulares.

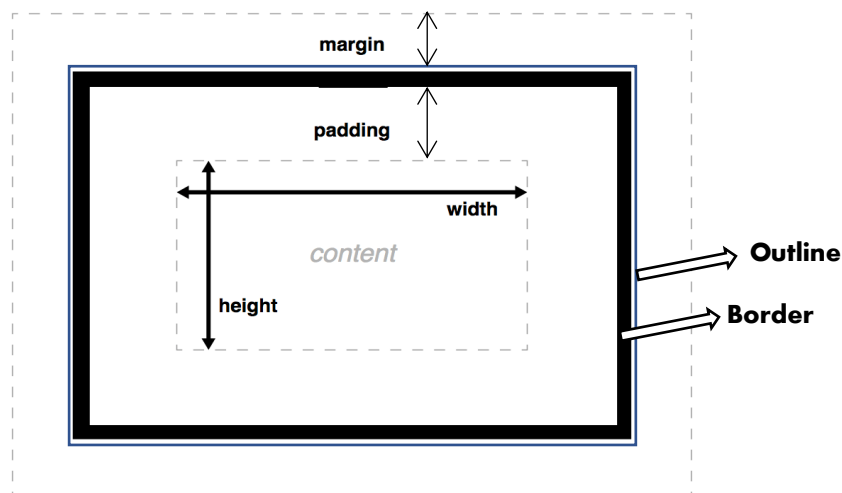
Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.

Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde.

- Abre el documento `prueba_cajas.html`. Como puedes ver, este documento incluye párrafos, encabezados y listas. Añade una regla CSS para visualizar los bordes de las cajas que existen, pero no son visibles: `h1, h2, p, ul, ol {border: 1px solid black;}`

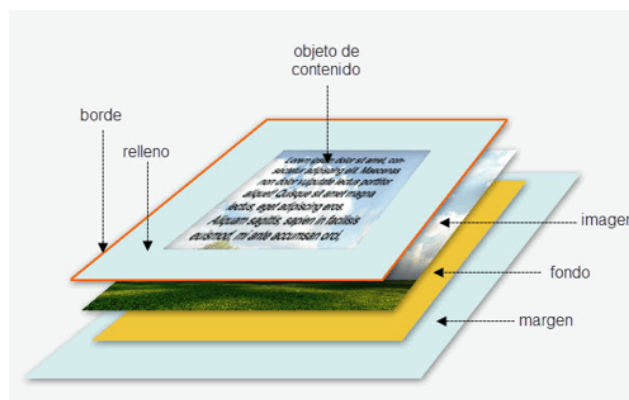
## 6.2 Elementos de la caja

Cada caja tiene una serie de elementos configurables, representados en el siguiente esquema:



- ➔ **Contenido (content):** se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.). El contenido de la caja es el espacio central en el que se representa el texto de la caja.
- ➔ **Relleno (padding):** espacio libre opcional existente entre el contenido y el borde. Transparente.
- ➔ **Borde (border):** línea que encierra completamente el contenido y su relleno.
- ➔ **Línea externa (outline):** línea que encierra el borde (exterior del borde).
- ➔ **Margen (margin):** separación opcional existente entre la caja y el resto de cajas adyacentes. Elemento transparente.

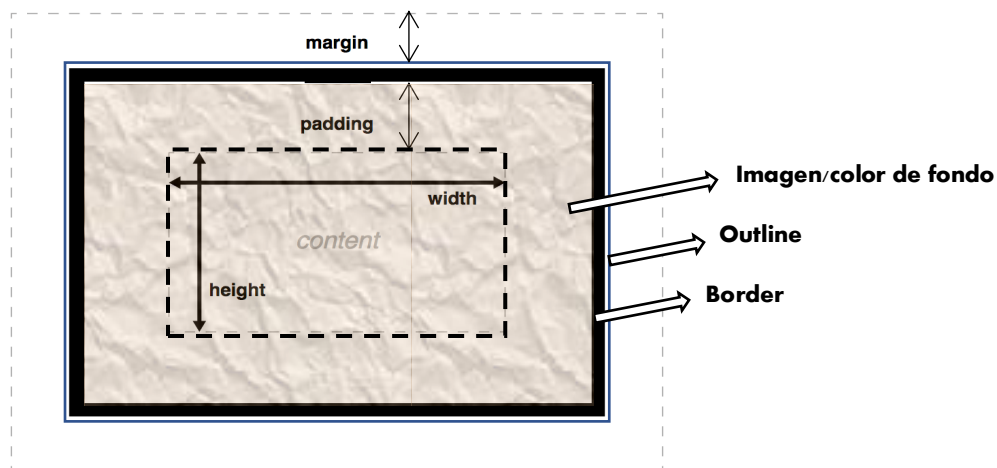
Con respecto a la superposición de esos elementos (cómo se superponen entre ellos):



- ➔ **Color de fondo (background-color):** color que se muestra en el contenido (detrás del texto) y el padding, que es transparente.
- ➔ **Imagen de fondo (background-image):** imagen que se muestra en el contenido (detrás del texto) y el padding, que es transparente.

La imagen está por encima del color de fondo, por ello si en una caja hay definidos tanto un color como una imagen de fondo, tiene más prioridad la segunda. Si la imagen no cubre todo el espacio, también se visualizará el color de fondo.

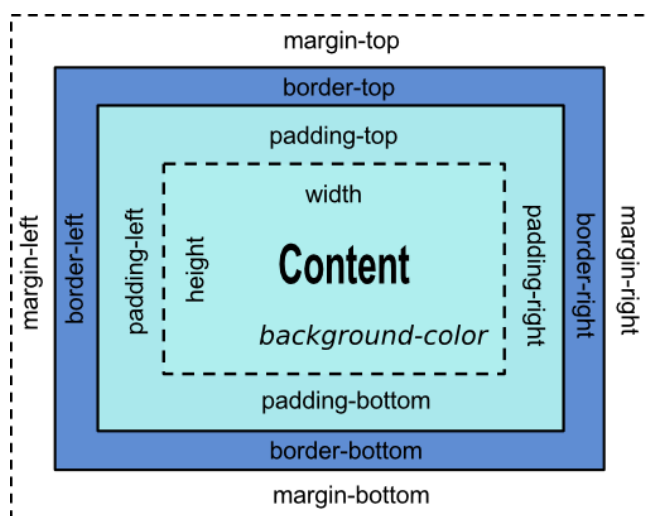
El margen no se rellena ni con el color ni con la imagen de fondo definidos para la caja. Se rellenarán con el color o imagen de fondo de su elemento padre, o los definidos para todo el documento.



### 6.3 Algunas propiedades css para cajas

- **padding**

Propiedad compuesta (shorthand), que configura el relleno de la caja. El relleno es asimétrico: puede tener valores distintos en cada uno de sus componentes.

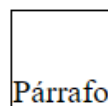


`padding: padding-top padding-right padding-bottom padding-left`

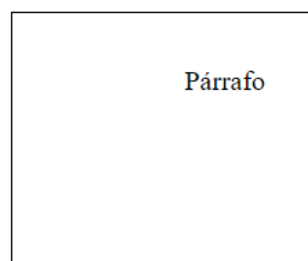


Ejemplos:

```
p{
  border: 1px solid black;
  padding-top:30px;
  display:inline-block;
}
```



```
p{
  border: 1px solid black;
  padding:30px 30px 100px 100px;
  display:inline-block;
}
```



Existen otras posibilidades para el shorthand:

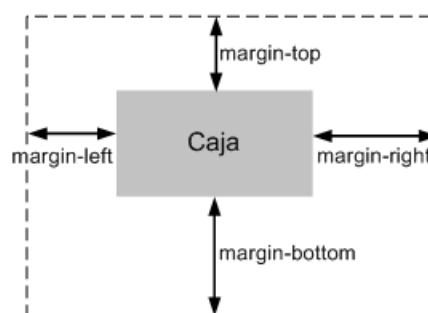
[https://www.w3schools.com/cssref/pr\\_padding.asp](https://www.w3schools.com/cssref/pr_padding.asp)

## ▪ margin

Espacio alrededor de la caja.

Ejemplos:

```
margin-bottom: 20px;
margin-left: 5%;
margin: 10px;
```



El margen es asimétrico: puede tener valores distintos en cada uno de sus componentes. En este caso “margin” es una propiedad compuesta (shorthand):

*margin: margin-top margin-right margin-bottom margin-left*



Ejemplos:

```
margin: 10px 20px 5px 15px;
p {
  margin: 50px;
  margin-left: 10px;
}
```

Existen otras posibilidades para el shorthand:

[https://www.w3schools.com/cssref/pr\\_margin.asp](https://www.w3schools.com/cssref/pr_margin.asp)

Los márgenes verticales (margin-top y margin-bottom) sólo se pueden aplicar a los elementos de bloque y las imágenes, mientras que los márgenes laterales (margin-left y margin-right) se pueden aplicar a cualquier elemento.

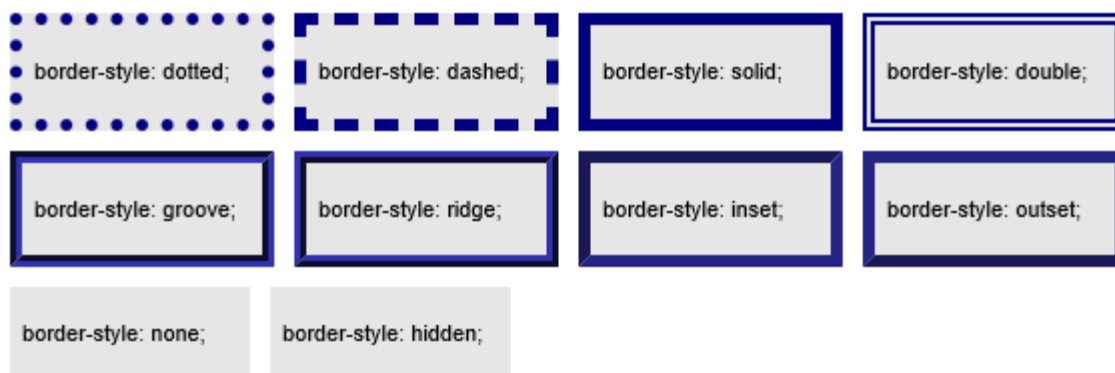
- **border-width**

Grosor del borde.

```
border-width: 1px;
```

- **border-style**

Estilo o formato del borde.



Los estilos none y hidden son idénticos visualmente, pero se diferencian en la forma que los navegadores resuelven los conflictos entre los bordes de las celdas adyacentes en las tablas.

- **border-color**

Color del borde.

```
border-color: blue;
```

- **border**

Propiedad compuesta (shorthand) que a su vez está formada por:

*border-width border-style border-color*

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

El borde es asimétrico: puede tener valores distintos en cada uno de sus componentes:

**border-top:** (shorthand) a su vez está formado por las propiedades:

*border-bottom-width border-bottom-style y border-bottom-color*

```
border-bottom: 1px solid black;
```

**border-right:** (shorthand) a su vez está formado por las propiedades:

*border-right-width border-right-style y border-right-color*

```
border-right: 10px dotted blue;
```

**border-bottom:** (shorthand) a su vez está formado por las propiedades:

*border-bottom-width border-bottom-style y border-bottom-color*

```
border-right: 5px dashed red;
```

**border-left:** (shorthand) a su vez está formado por las propiedades:

*border-left-width border-left-style y border-left-color*

```
border-right: 5px double red;
```



Ejemplo:

```
p{
  border-top: 10px solid black;
  border-right: 10px dashed blue;
  border-bottom: 10px dotted red;
  border-left: 10px double green;
  padding:30px;
  display:inline-block;
}
```



#### ▪ border-radius

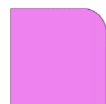
Redondea esquinas. Está formada por cuatro propiedades (shorthand):



*border-top-left-radius border-top-right-radius border-bottom-right-radius  
border-bottom-left-radius*



`border-radius: 50px 0px 0px 0px;`



`border-radius: 0px 50px 0px 0px;`



`border-radius: 0px 0px 50px 0px;`



`border-radius: 0px 0px 0px 50px;`

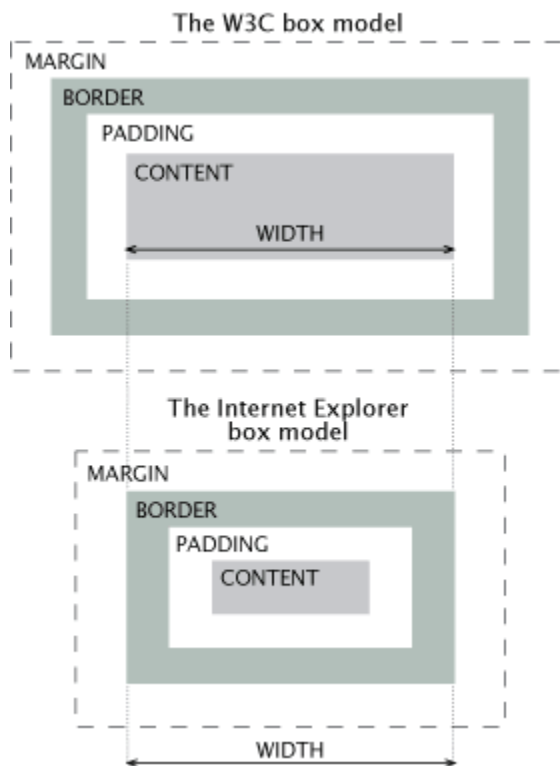
## Actividad 7. Bordes y espacios

7.1 Aplicar bordes

7.2 Aplicar márgenes y rellenos

- **box-sizing**

Indica hasta donde se miden el ancho y el alto en las propiedades width y height.



`box-sizing: content-box;`

Valor por defecto. El ancho y alto sólo se aplican al content de la caja.

`box-sizing: border-box;`

El ancho y alto se aplican al content, padding y border de la caja.

- **width y height**

width indica el ancho del content o del content+padding+border si box-sizing=border-box.

height indica el alto del content o del content+padding+border si box-sizing=border-box.

Ejemplos:

```
width: 20%; /*porcentaje del bloque que lo contiene*/
height: 30px;
width: auto; /*se calcula automáticamente*/
```

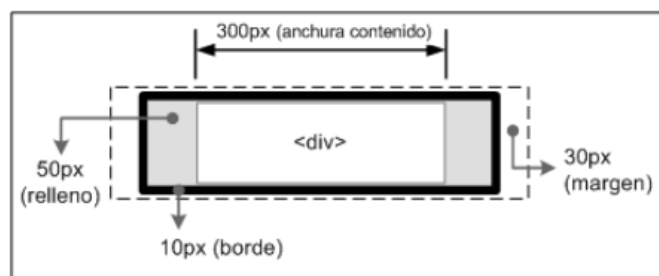
Estas propiedades no admiten valores negativos y los valores en porcentaje se calculan a partir de la anchura o altura de su elemento padre.

Podríamos pensar que al definir un ancho de 300px, la caja ocuparía un ancho de 300px:

```
div {
  width: 300px;
  padding-left: 50px;
  padding-right: 50px;
  margin-left: 30px;
  margin-right: 30px;
  border: 10px solid black; }
```

Pero la realidad es que la anchura real de la caja será 480px:

$$30\text{px} + 10\text{px} + 50\text{px} + 300\text{px} + 50\text{px} + 10\text{px} + 30\text{px} = 480 \text{ píxel}$$



### ▪ box-shadow

Establece la sombra de la caja. Es una propiedad compuesta (shorthand) por los valores:

Desplazamiento horizontal de la sombra      Desplazamiento vertical de la sombra

box-shadow: h-offset v-offset blur color inset

Difuminado de la sombra      Color de la sombra      Sombra interna en lugar de externa



Ejemplos:

Sombra en la caja      `box-shadow: none;`

Sombra en la caja      `box-shadow: 5px 5px 5px grey;`

Sombra en la caja      `box-shadow: 15px 5px 10px blue;`

Sombra en la caja      `box-shadow: 5px 5px 5px red inset;`



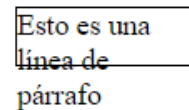
### ▪ **overflow**

Indica cómo actuar si el contenido de la caja sobrepasa el tamaño de la caja dimensionada. Posibles valores:

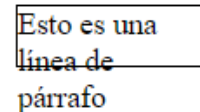
```
overflow: auto;      /*aparece scroll*/
overflow: hidden;    /*desaparece el contenido que no cabe*/
overflow: visible;    /*el contenido desborda la caja. Por defecto*/
```

Por ejemplo, se tienen el siguiente párrafo: `<p>Esto es una línea de párrafo</p>`

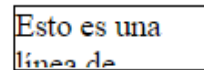
```
p{
  width:100px;
  height:30px;
  border: 1px solid black;
}
```



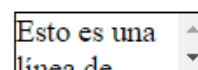
```
p{
  width:100px;
  height:30px;
  border: 1px solid black;
  overflow:visible;
}
```



```
p{
  width:100px;
  height:30px;
  border: 1px solid black;
  overflow: hidden;
}
```



```
p{
  width:100px;
  height:30px;
  border: 1px solid black;
  overflow:auto;
}
```



### ▪ **background-clip**

Indica hasta dónde se aplica el fondo (background), cuando se pone una imagen o color. Valores:

```
background-clip: padding-box;
background-clip: content-box;
```

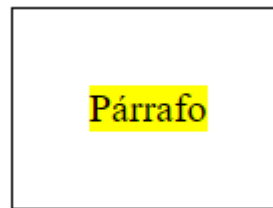
- **background-color**

Color del fondo de la caja (por defecto hasta límite exterior del borde)

```
p{
  border: 1px solid black;
  padding:30px;
  background-color: yellow;
  display:inline-block;
}
```



```
p{
  border: 1px solid black;
  padding:30px;
  background-clip:content-box;
  background-color: yellow;
  display:inline-block;
}
```



- **background-image**

Imagen del fondo de la caja (por defecto hasta límite exterior del borde)

```
body{
  background-image: url(imgCSS/fondo1.png);
}
```



Si se define una imagen y un color de fondo, la primera se muestra sobre el segundo. Suele ser habitual indicar un color de fondo siempre que se muestra una imagen de fondo. En caso de que la imagen no se pueda mostrar o contenga errores, el navegador mostrará el color indicado (que debería ser, en lo posible, similar a la imagen) y la página no parecerá que contiene errores.

Si la imagen que es demasiado grande para el fondo del elemento, solamente se muestra la parte de imagen comprendida en el tamaño del elemento.

Si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento. Esto es útil para dar por ejemplo texturas, ya que la imagen que se tiene que descargar tiene muy poco peso, pero de forma repetida rellena toda la página.

### ▪ **background-repeat**

Repetición de la imagen de fondo

```
background-repeat: repeat;           /*se repite horizontal y verticalmente*/
background-repeat: no-repeat;        /*no se repite*/
background-repeat: repeat-x;         /*se repite horizontalmente*/
background-repeat: repeat-y;         /*se repite verticalmente*/
```

### ▪ **background-attachment**

Scroll de la imagen de fondo

```
background-attachment: scroll;        /*scroll con la página*/
background-attachment: fixed;        /*no scroll*/
background-attachment: local;        /*scroll con el contenido*/
```

### ▪ **background-size**

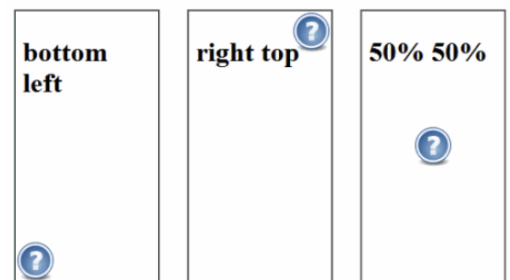
Tamaño de la imagen del fondo

```
background-size: 140px;
```

### ▪ **background-position**

Posición de la imagen de fondo

```
background-position: center;
background-position: left top;
background-position: center bottom;
background-position: 10% 20%; /*horizontal y vertical*/
background-position: 100px 400px; /*horizontal y vertical*/
```



### ▪ **background**

Propiedad compuesta (shorthand) que está formada por:

```
background: background-color background-image background-attachment background-
            position background-repeat
```

Ejemplo:

```
body{
  background-image: url(imgCSS/fondo1.png);
  background-repeat: no-repeat;
  background-position: top center;
}
```

→ `body{background: url(imgCSS/fondo1.png) no-repeat top center;`

## Actividad 9. Repetición de imagen de fondo

9.1 Imagen de fondo de caja con repetición

9.2 Imagen de fondo de caja sin repetición

## Actividad 10. Scroll de imagen de fondo

10.1 Imagen con scroll

10.2 Imagen fija

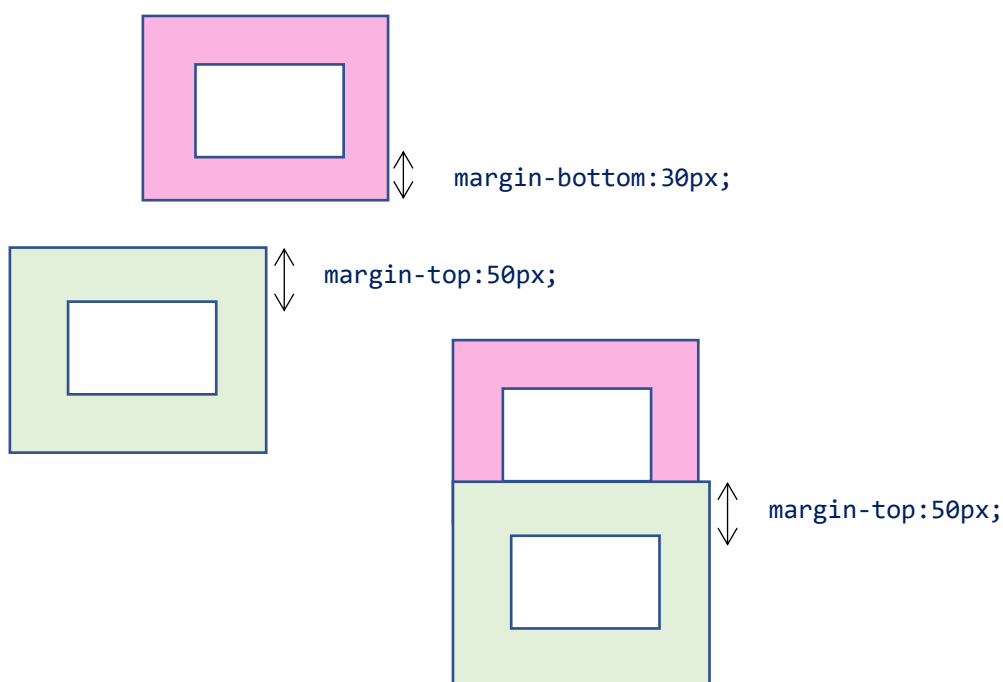
Ejercicios: propiedades de cajas. Ejercicio 1.

### ▪ Colapso de margen

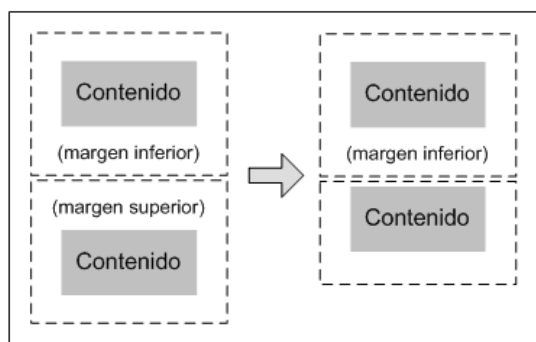
Cuando dos bloques que tienen márgenes inferior y superior se sitúan uno encima de otro, el margen resultante entre ambos no es la suma del margen inferior del elemento de arriba con el margen superior del elemento de abajo, sino que se produce un colapso: ambos márgenes se combinan en uno con el tamaño mayor entre ambos.

Esto sólo ocurre para los [márgenes superior e inferior](#), nunca para los laterales, ni para el padding.

Ejemplo. Se tienen dos elementos:

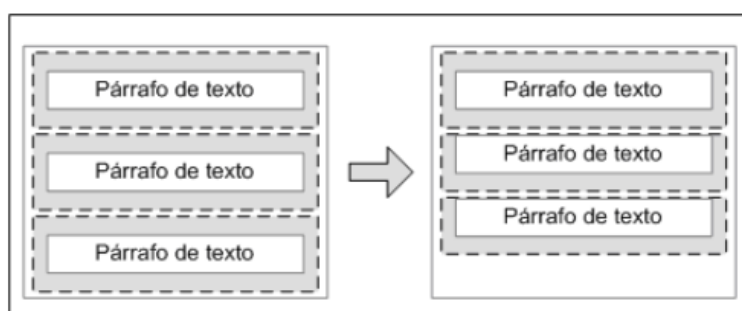






¿Por qué se produce el colapso de margen?

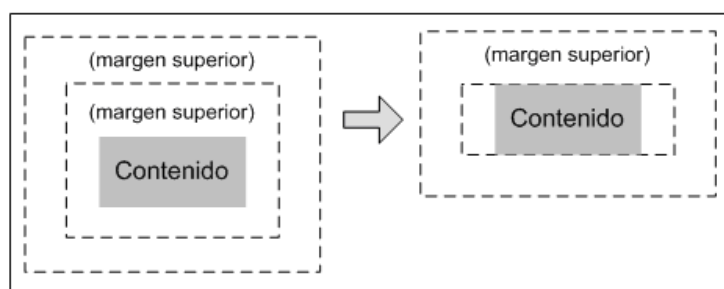
Se trata de normalizar el formato de presentación con respecto al primer elemento. Se utiliza en párrafos.



Sin colapso de margen el primer párrafo aparece con una distancia menor al extremo superior que las distancias entre párrafos

Con colapso de margen todos los párrafos se ven con la misma separación, incluido el primero

De la misma forma, si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado:



## Actividad 11. Colapso de margen

11.1 Propiedades de cajas. Distancia igualitaria entre párrafos

11.2 Distancia distinta entre párrafos

11.3 Propiedad background-repeat

## 6.4 Resumen de propiedades css para cajas

propiedad	descripción	Algunos valores
width	ancho	20 px 10% auto
height	alto	
box-sizing	indica hasta donde se miden el ancho y el alto en las propiedades width y height	content-box border-box
box-shadow	establece la sombra de la caja. Compuesta.	15px 5px 10px blue; 15px 5px 10px blue inset;
display	establece el comportamiento de la caja	none inline block inline-block flex
overflow	indica cómo actuar si el contenido de la caja sobrepasa el tamaño de la caja dimensionada	auto hidden visible
padding	Relleno de la caja. Compuesta: padding-top padding-right padding-bottom padding-left	20px 10%
background-clip	indica hasta dónde se aplica el fondo (background), cuando se pone una imagen o color	padding-box content-box
background-color	color del fondo de la caja	orange
background-image	imagen del fondo de la caja	url("img/favorite-2.jpg")
background-repeat	repetición de la imagen de fondo	repeat no-repeat repeat-x repeat-y
background-attachment	scroll de la imagen de fondo	scroll fixed local
background-size	tamaño de la imagen del fondo	300px
background-position	posición de la imagen de fondo	center left top 10% 5% 100px 300px

background	propiedad compuesta	blue url(/img/ctrip.gif) repeat-x 0 0
border-width	grosor del borde	10px
border-style	estilo del borde	none solid dotted dashed inset
border-color	color del borde	red
border	Borde. Compuesta	1px solid black
border-top border-right border-bottom border-left	Bordes. Propiedades compuestas	
border-radius	redondea esquinas. Compuesta	
margin	Margen. Compuesta	
margin-top margin-right margin-bottom margin-left	Margen. Propiedades compuestas	10px 10%