



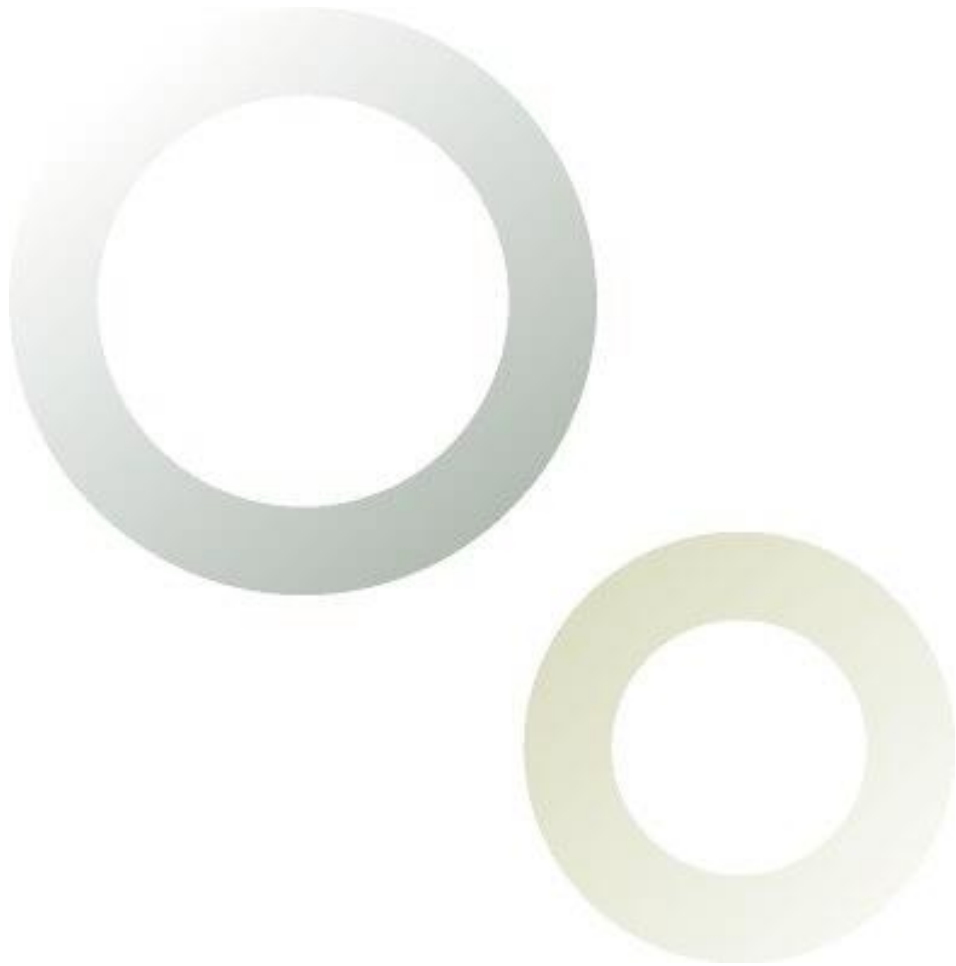
# LENGUAJES DE MARCAS

## UD4. Introducción a CSS

Desarrollo de Aplicaciones Web  
Curso 2021-2022

Profesores: Diana Bautista / Óscar Villar

# Índice de contenido

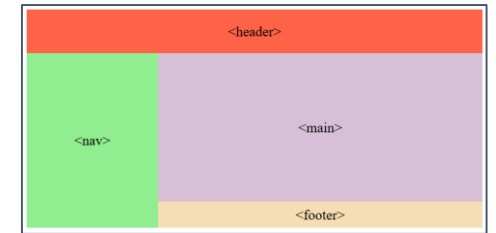
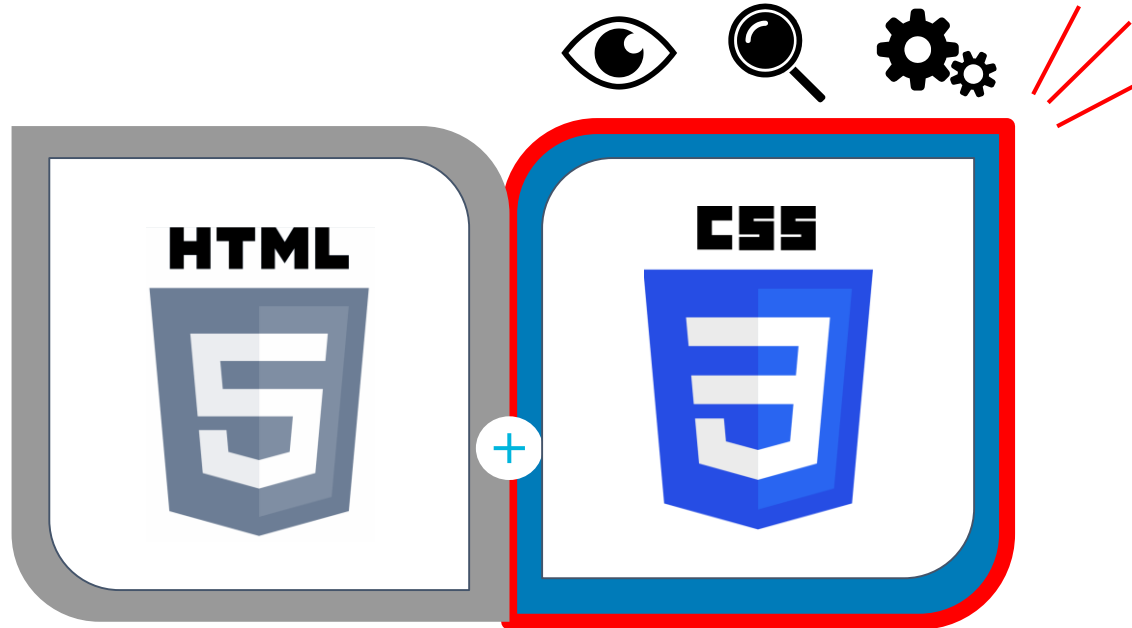


1. Introducción
2. Declarar reglas CSS
3. Selectores básicos
4. Modelo de caja
5. Propiedades de texto
6. Colores
7. Unidades de medida
8. Posicionamiento FLOAT/CLEAR

# 1.Introducción

# Dónde nos encontramos?

Ya tenemos una base en HTML que nos permite avanzar hacia un estudio de estilos para nuestros desarrollos web.



Layout

**HTML:** define el contenido de la web. Es decir, los elementos que la componen.

**CSS:** define los estilos de la web. Aspectos visuales, así como la disposición de elementos o Layout.

# ¿Recordamos todos que..

El navegador (user-agent) aplica unos estilos CSS “computed” por defecto para cada elemento HTML declarado?



EJEMPLO: Los encabezados (por defecto) son de color negro.. y de un tamaño..

Podemos variar estos estilos en cualquier momento aplicando un conjunto de reglas CSS

# Reglas CSS

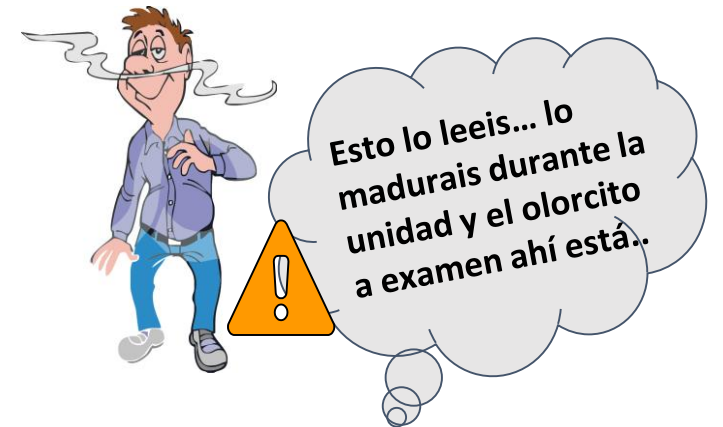
## LOS 3 PILARES DE CSS(\*)

**Cascada:** El orden de las reglas importa. Cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el CSS.

**Herencia:** Las reglas aplicadas a un elemento se propagan a sus elementos hijos, salvo que estos declaren una regla concreta que únicamente les concierne a ellos. Algunas propiedades (como el ancho ) no se propagan.

**Especificidad:** Podemos aplicar reglas tan específicas para un elemento como deseemos. Un ejemplo de regla altamente específica es aquella cuyo selector se basa en un id. Una menos específica sería la que haga uso de una clase. De las menos específicas son aquellas que hagan uso de selectores de etiqueta o selector universal.

```
h1 { color: red; }  
h1 { color: blue; } /* Esta gana */
```



(\*) Podemos construir edificios si, incluso sin saber de arquitectura ni obra... levanten paredes dijeron algunos..

## 2. Declarar reglas CSS

# Estilos en línea

Reglas CSS declaradas mediante el atributo **style** contenido en la etiqueta de comienzo de un elemento HTML.

```
<h1 style="color:red"> Encabezado Principal</h1>
```

PROBLEMA: ¿Y si declaro varios encabezados que quiero que sean morados?

*Moraleja: solo usad estilos en línea de forma puntual si tenéis una muy buena razón!*

Ostras esto me suena..



Rápida declaración puntual sobre los elementos.



Decrementa legibilidad.

Código no mantenible.

Y alguno más..



# <style>

Etiqueta HTML declarada en el apartado <head> de una página donde podemos declarar reglas CSS con la sintaxis propia de las hojas de estilos.

```
...  
<head>  
  <style>  
    h1{ color: red; }  
  </style>  
</head>  
... index.html
```



Regla CSS

Ostras esto  
me suena..



Rápida declaración de reglas CSS.

Posibilidad de navegar en el mismo documento y trabajar por separado HTML y CSS.

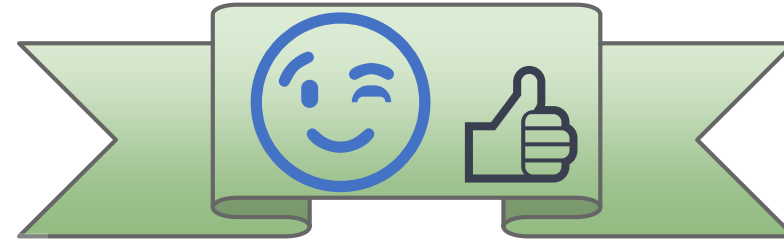
Puede ser interesante desde el punto de vista didáctico o para pequeños desarrollos.



La extensión de fichero HTML crece rápido.

No es la opción más mantenible de todas.

# Hojas de estilo CSS



Conjunto de reglas CSS declaradas en ficheros de extensión .css los cuales deben estar enlazados con la página HTML que recibe los estilos.

```
...  
<head>  
    <link rel="stylesheet" href="estilos.css">  
</head>  
...
```

index.html

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

estilos.css

Referencia  
a la hoja de estilos:  
"estilos.css"



Separación total: estilos CSS / elementos HTML

Más mantenible / legible

**Forma recomendada de trabajo con HTML5/CSS3**

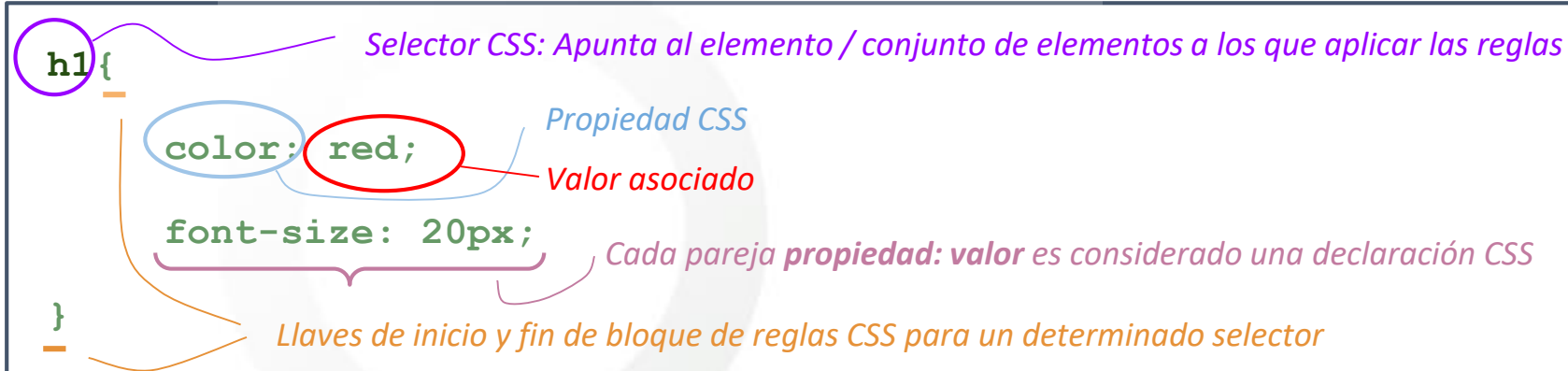
(y la asignatura de Lenguajes de Marcas)



La posible dificultad de trabajar con dos ficheros de código separados.

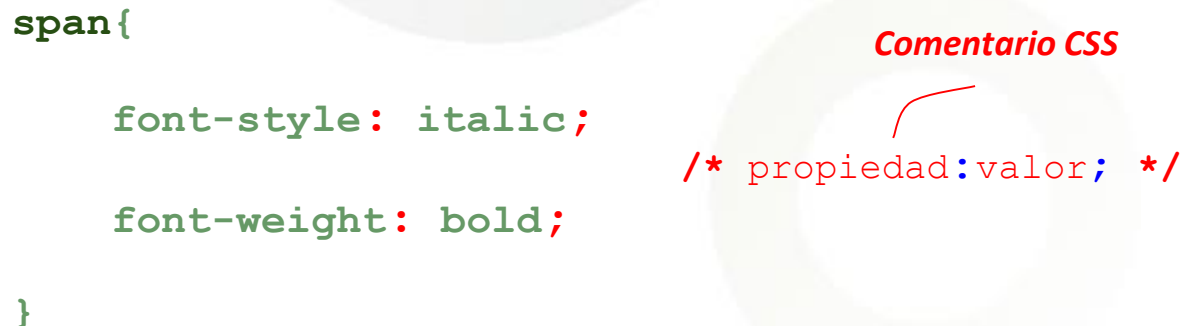
## 2. Selectores básicos

# Estructura de una regla CSS



The diagram illustrates the structure of a CSS rule for the `h1` selector. The selector `h1` is circled in purple, with a line pointing to the text "Selector CSS: Apunta al elemento / conjunto de elementos a los que aplicar las reglas". The opening curly brace `{` is underlined in orange, with a line pointing to "Llaves de inicio y fin de bloque de reglas CSS para un determinado selector". Inside the rule, the property `color` is circled in blue, labeled "Propiedad CSS", and its value `red` is circled in red, labeled "Valor asociado". The property-value pair `color: red;` is bracketed in purple, with a line pointing to "Cada pareja **propiedad: valor** es considerado una declaración CSS". The closing curly brace `}` is also underlined in orange.

```
h1{  
  color: red;  
  font-size: 20px;  
}
```



The diagram shows a CSS rule for the `span` selector. The selector `span` is followed by an opening curly brace `{`. The first property is `font-style: italic;`. The second property is `font-weight: bold;`. A comment `/* propiedad: valor; */` is shown in red, with a line pointing to it from the label "Comentario CSS". The rule ends with a closing curly brace `}`.

```
span{  
  font-style: italic;  
  font-weight: bold;  
  /* propiedad: valor; */  
}
```

CSS FILE / CSS RULES into <STYLE> TAG

# Selectores básicos



Selector de elemento HTML **h1**{ color: red; font family: "Courier"; } /\* Todos los h1..

\*/

Selector de id **#BotonEnviar**{ border-radius: 10px; } /\* Solo

aquel elemento con id.. \*/

Selector de clase **.encabezados**{ text-align: center; }

/\*Elementos cuya clase sea encabezado\*/

Selector de clase **p.encabezados**{ text-align: center; } /\*elementos p cuya

clase sea encabezado\*/

Selector agrupado

**h2, h3**{ color: steelblue; }

# Ejercicio

## CONTRATO DE COMPRA-VENTA

Vamos a completar nuestro primer ejemplo de hoja de estilos. Conforme vayamos avanzando, aplicaremos estilos más complejos con diferentes selectores.

### Comprador principal

Nombre y Apellidos: Luis López Santagueda  
Dirección: C./ Giner de los ríos, 15-6  
CP: 46023

### Educación y formación

Nombre y Apellidos: Sara Moreno Jimeno  
Dirección: Avd. Del Puerto, 15-9  
CP: 45021

### Relación de compra venta

#### *Propiedades*

Propiedad 1: C/Lorca, 15-26 *Madrid*.  
Propiedad 2: C/ San Lucas, 15-6 *Toledo*.

**Empresa: Transiciones S.L**  
***info@transiciones.org***

# Comentarios

Lo primero de todo, pensad que aunque los estilos no queden exactamente iguales no pasa nada. Lo importante es que los elementos que tengan estilos diferentes, los sigan teniendo. Dicho esto:

El texto se compone de elementos con las siguientes etiquetas:

<h1>, <h2>, <h3>, <p> <em> <strong> <address>

Recordad: EM aportaba un aspecto en cursiva, pero tenía contenido semántico, al igual que STRONG, que aportaba una apariencia en negrita.

La hoja de estilos tiene las siguientes propiedades:

Body: Color de fondo lavenderbush, fuente sans-serif, alineación justificada.

H1: Color salmon, alineación centrada, mayúsculas (text-transform: uppercase).

H2: Color de fondo skyblue, color de letra: teal, subrayado ( text-decoration: underline).

H3: Color de fondo: thistle, color de texto: steelblue, tamaño de fuente 130%, estilo de fuente: cursiva.

Párrafo: color de fondo: rosa

EM: color de texto gris, negrita y cursiva (font-style: italic, font-weight: bold)

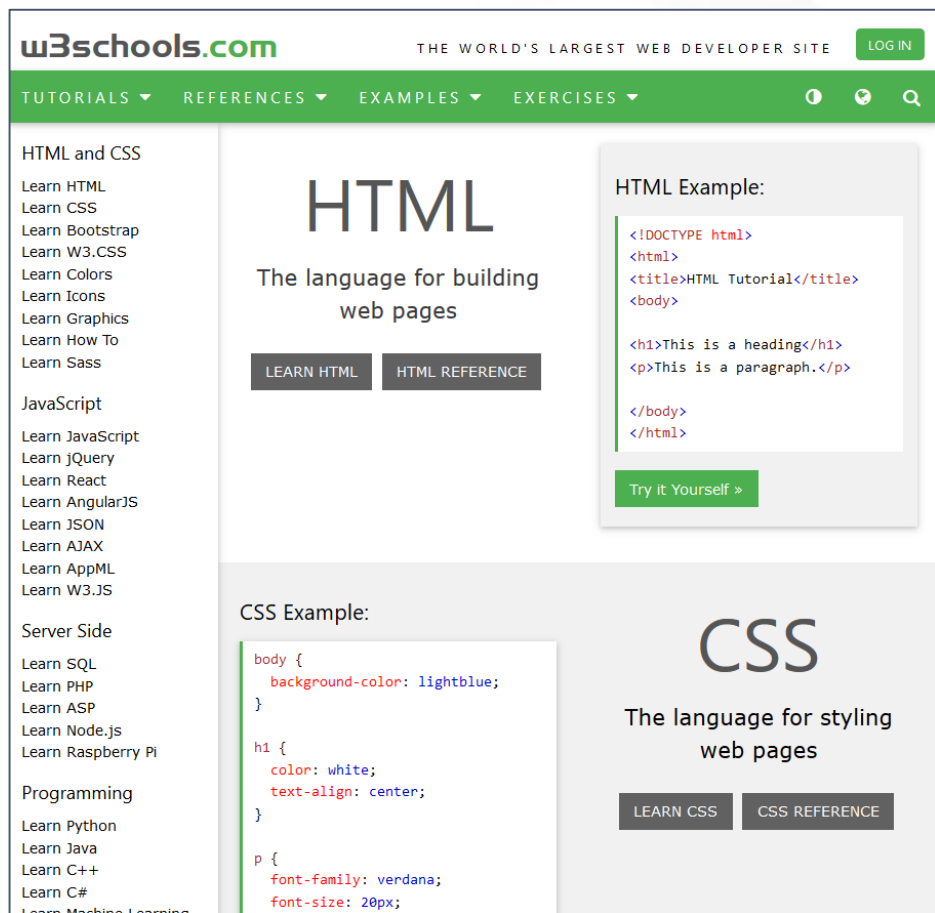
STRONG: color tomato, y SIN negrita

Address: Color de fondo blanco, negrita y alineación centrada.

### 3. Modelo de caja



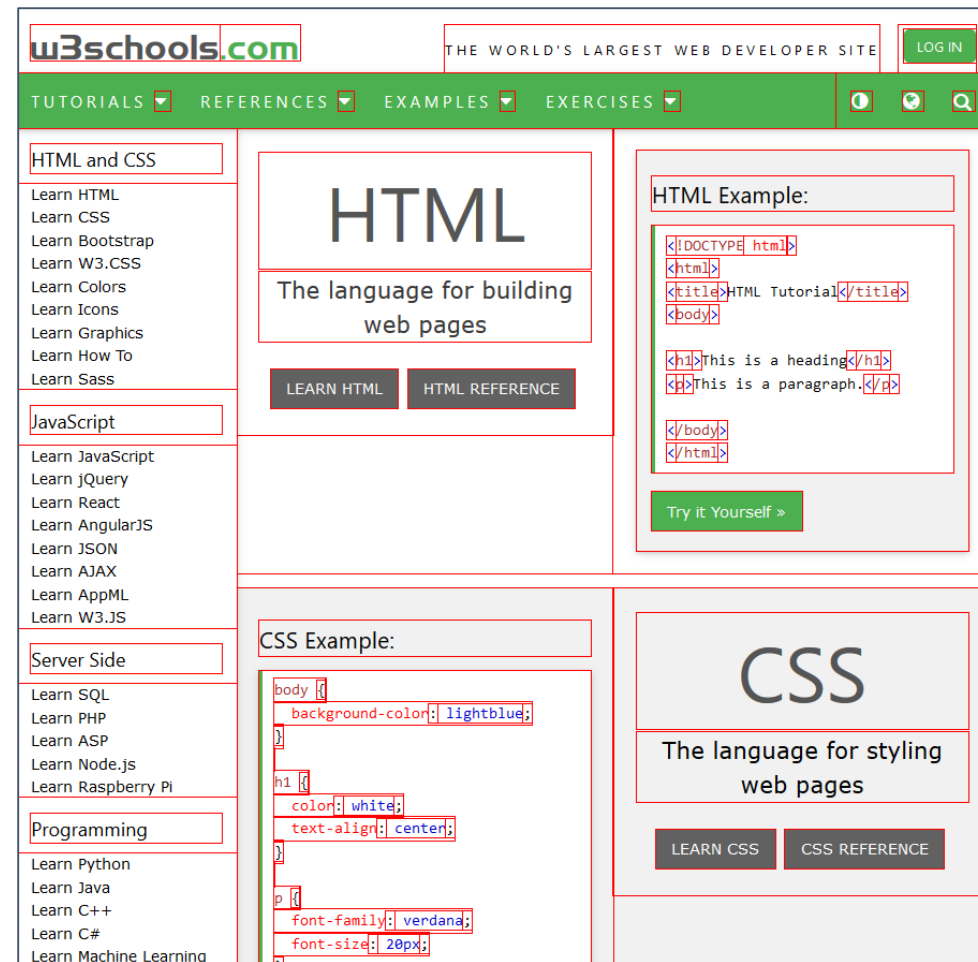
# Todos los elementos HTML son.. CAJAS!!



The screenshot shows the w3schools.com homepage. The layout is clean with a green header. The main content area is divided into three columns. The left column contains a list of topics like HTML and CSS, JavaScript, and Server Side. The middle column features a large 'HTML' heading and a description 'The language for building web pages'. The right column shows an 'HTML Example' with a code block containing basic HTML tags. Below this, there is a 'CSS Example' with a code block showing a simple CSS rule for the body background color.

Aplica, aplica esta regla CSS!

\*{ outline: 1px solid red; }



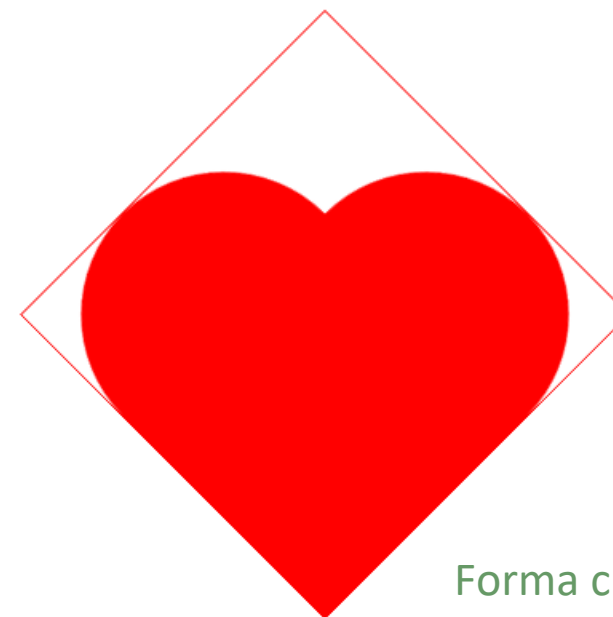
The screenshot shows the same w3schools.com homepage, but with the CSS rule applied. A red outline is visible around every element on the page, including the header, navigation menu, sidebar, main content area, and footer. This visualizes the concept that all HTML elements are boxes that can be styled with CSS.

# Todos los elementos HTML son..

## CAJAS!!



Este (y el resto) también, pese a ser una figura hecha con CSS !!

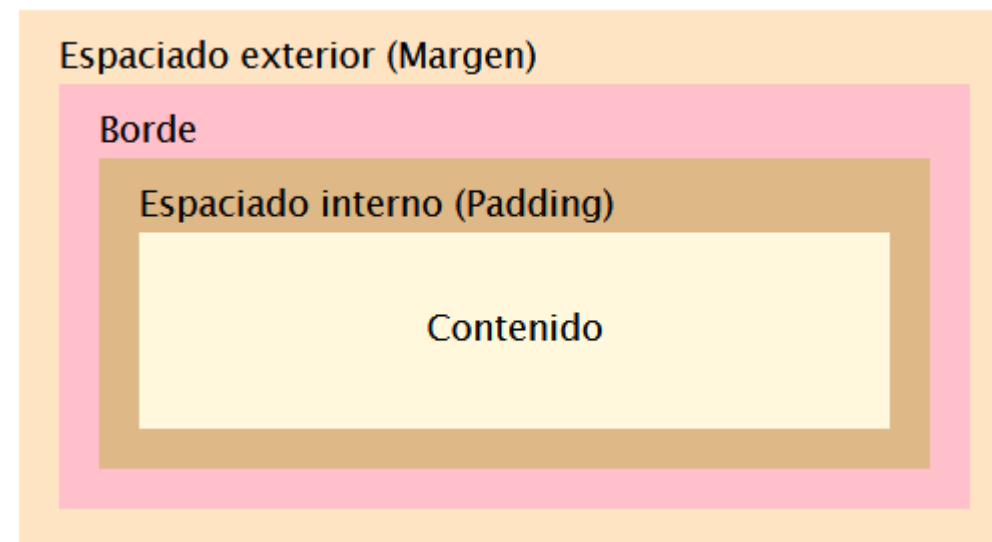


Forma creada con  
pseudelementos y  
rotación de la caja.

# Modelo de caja

Todos los elementos HTML tienen un contenido, un espaciado interno, un borde y un espaciado exterior (algunas de estas propiedades pueden tener valor 0).

El modelo de caja considera únicamente el contenido, espaciado interno y el borde. El espaciado exterior no es considerado.



**PROFE, ENSEÑAME...**

Evitar el empuje accidental de elementos con unas dimensiones declaradas, aplicando:

**`*{ box-sizing: border-box; }` && inspector de código**

# Alto y ancho (height/width)

Propiedades referidas a las dimensiones de un elemento.

Debemos recordar que cada elemento tiene su propia naturaleza:

- Elemento de bloque: Puede definir alto/ancho
- Elemento de línea: El alto y ancho lo define su contenido (texto?, dimensiones del pixelado de una imagen?)
- Elemento de línea-bloque: Permite definir alto/ancho

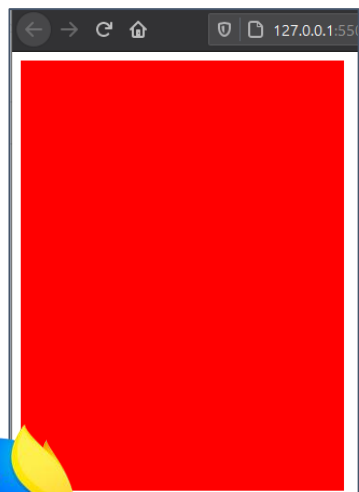
Debe declararse un alto y ancho para aquellos elementos que deban mostrarse en el visor, especialmente si estos no tienen contenido interno.

Para las imágenes puede ser interesante declarar una de estas dos propiedades y la restante dejar que la calcule el navegador automáticamente (respetar el aspect-ratio original).

Pueden ser declaradas con las unidades de medida disponibles: px, %, em, rem, vh, vw (las estudiaremos con calma en un apartado de esta unidad)

# Inspeccionemos un <div>

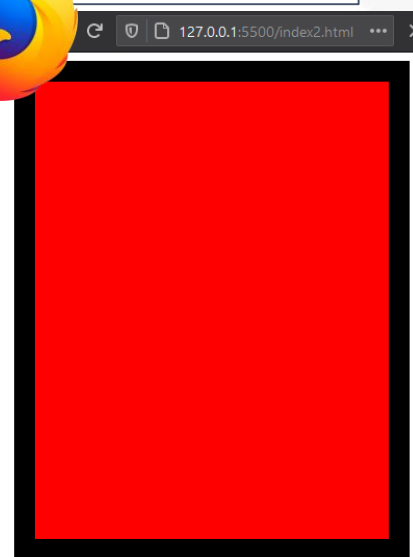
OJO un div sin dimensiones ni contenido no se dibuja!!!



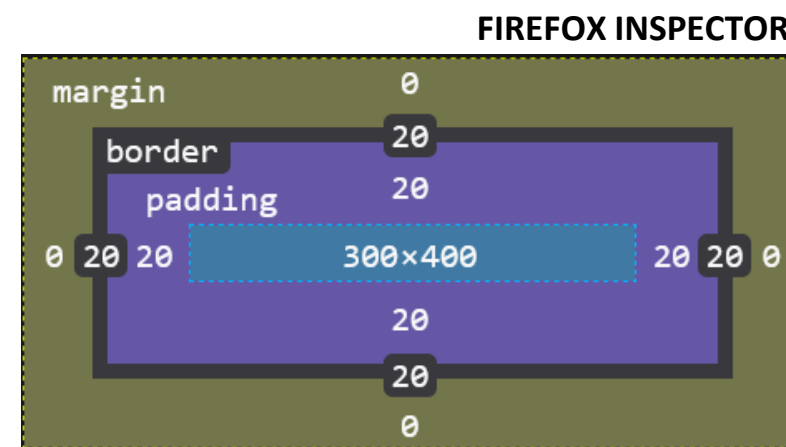
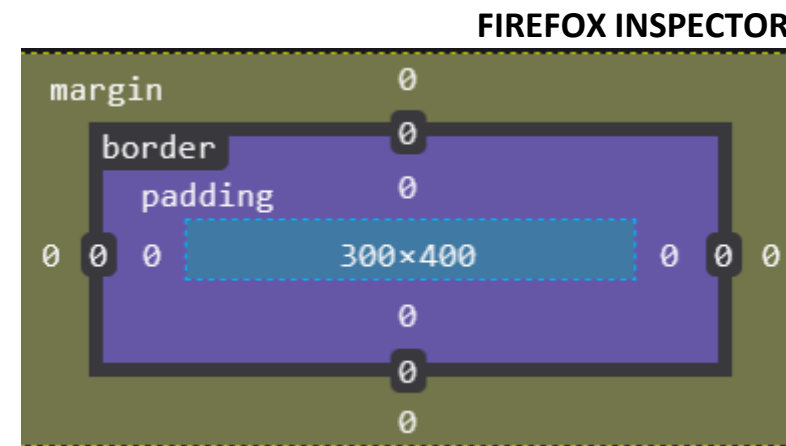
```
<style>
  div{
    background-color: red;
    width: 300px;
    height: 400px;
  }
</style>
```



F12 o inspeccionar elemento..



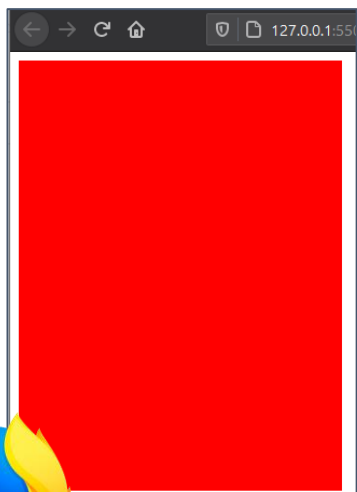
```
<style>
  div{
    background-color: red;
    width: 300px;
    height: 400px;
    border: 20px solid black;
    padding: 20px;
  }
</style>
```



El problema del empuje..

El tamaño del elemento aumenta: borde y espaciados se suman a las dimensiones del contenido.

# Box-sizing: border-box

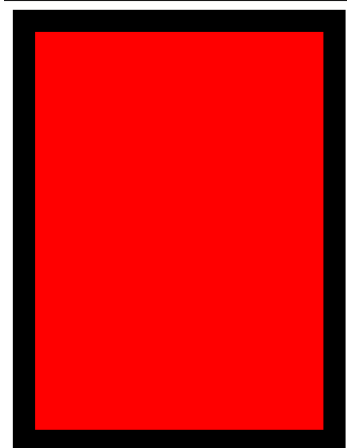


```
<style>
  div{
    background-color: red;
    width: 300px;
    height: 400px;
  }
</style>
```



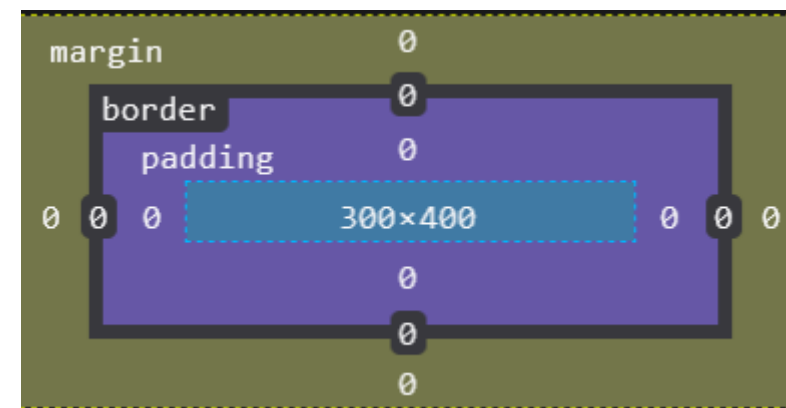
F12 o  
inspeccionar  
elemento..

El problema del  
empuje "resuelto"

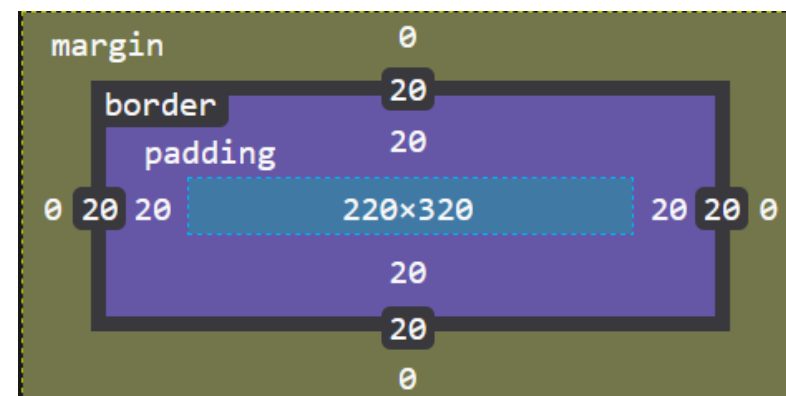


```
<style>
  div{
    background-color: red;
    width: 300px;
    height: 400px;
    border: 20px solid black;
    padding: 20px;
    box-sizing: border-box;
  }
</style>
```

FIREFOX INSPECTOR



FIREFOX INSPECTOR



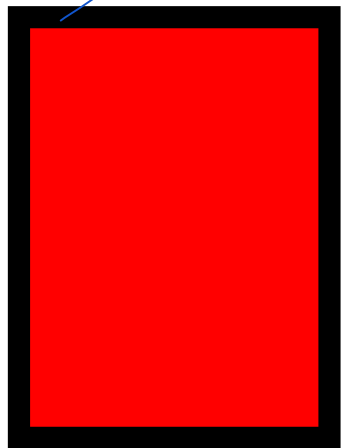
SIN aumento del tamaño del elemento: el tamaño del contenido se ajusta automáticamente para alcanzar las medidas width/height declaradas del elemento.

# Bordes

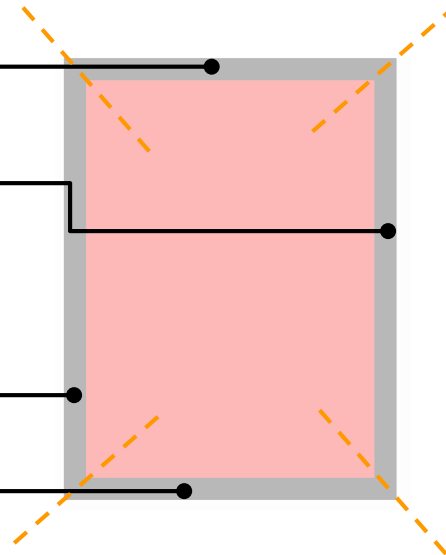
La propiedad CSS **border** es un “abreviado o shorthand” que aúna: border-top, border-right, border-bottom, border-left.

**border:** 1px solid black;

border-top: 1px solid black;  
border-right: 1px solid black;  
border-bottom: 1px solid black;  
border-left: 1px solid black;



```
<style>  
  div{  
    background-color: red;  
    width: 300px;  
    height: 400px;  
    border: 20px solid black;  
    padding: 20px;  
    box-sizing: border-box;  
  }  
</style>
```



Los bordes de cada lado acaban juntándose en 90 grados

dashed



ridge



dotted



inset



# Y un borde que jamás empuje para señalar el área de una caja?

La propiedad CSS **outline** tiene la apariencia de un borde externo dibujado el cual no produce un empuje de elementos, dado que no forma parte del modelo de caja.

Sirve como guía visual para el desarrollador. Indica el alcance del área de una caja para un determinado elemento.

También puede servir para indicar que un elemento está seleccionado.. aunque quizás no es lo más recomendable.

**outline: 1px solid red;**





# Bordes redondeados

La propiedad CSS **border-radius** permite efectuar un redondeado de bordes.  
Es un “shorthand o abreviado” que incluye: border-top-radius, border-right-radius...

**border-radius: 20px;**

border-top-left-radius: 20px;  
border-top-right-radius: 20px;  
border-bottom-left-radius: 20px;  
border-bottom-right-radius: 20px;

**border-radius: 50% 0 0 0;**

border-top-left-radius: 20px;

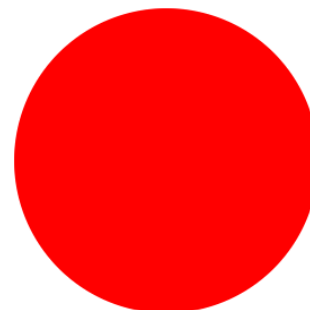
sin border-radius



border-radius: 20 px;



border-radius: 50%;



# Espaciados internos (padding)

La propiedad CSS **padding** es un “abreviado o shorthand” que aúna: padding-top, padding-right, padding-bottom, padding-left.

Se refiere al espaciado interno existente desde el borde de un elemento hasta su contenido. Si un elemento no tiene contenido, pero padding declarado se genera un espaciado interno en función de la medida declarada y la naturaleza del elemento (block|inline).

Generalmente sirve para acomodar o empujar un elemento contenido en otro.



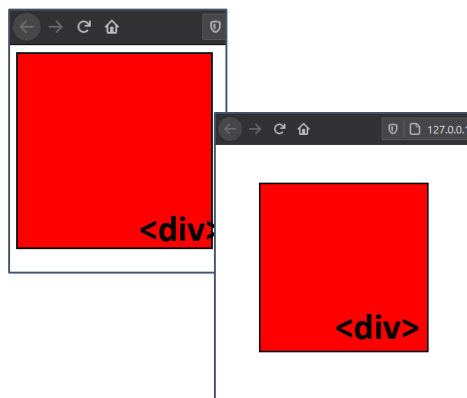
# Espaciados externos (margin)

La propiedad CSS **margin** es un “abreviado o shorthand” que aúna: margin-top, margin-right, margin-bottom, margin-left.

Se refiere al espaciado externo existente desde el borde de un elemento hasta su contenedor más próximo o elementos cercanos. Si un elemento no tiene contenido, pero margin declarado se genera un espaciado externo.

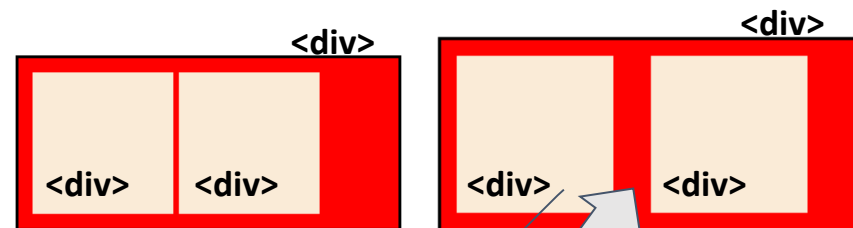
Generalmente sirve para conseguir espaciar un elemento con respecto a otros o conseguir una separación con el contenedor padre.

No podemos dibujar un borde que abrace el margin!



**margin: 50px;**

margin-top: 50px;  
margin-right: 50px;  
margin-bottom: 50px;  
margin-left: 50px;



**IMPORTANTE:** declaramos margin a un elemento el cual queremos separar de una zona concreta o elementos cercanos.

# Común para todos los shorthands..

Bordes, redondeados de bordes, espaciados internos y espaciados externos parten de su propiedad CSS abreviada o shorthand que en cada caso es: border, border-radius, padding y margin.

Estas propiedades incluyen implícitamente, si solo se declara un valor, la réplica de este para los 4 lados de la caja.

```
margin: 50px;
```

Si declaramos 2 valores, top y bottom se verán afectados por el primer valor y right y left por el segundo

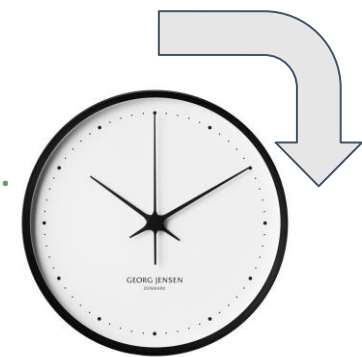
```
margin: 2px 3px;
```

La declaración de 3 valores implica top y bottom declarados explícitamente y la réplica del segundo valor para el lado derecho e izquierdo.

```
margin: 2px 3px 4px;
```

Si declaramos 4 valores es obvio que estamos haciendo una asignación explícita a cada uno de los lados. El orden que se sigue es el de las agujas del reloj: arriba | derecha | abajo | izquierda.

```
margin: 2px 3px 0 4px;
```



# Box shadow

La propiedad CSS **box-shadow** permite dibujar sombras externas a nivel de caja.

```
div{
```

```
...
```

```
box-shadow: 4px 4px grey;
```

```
...
```

```
}
```

desplazamiento  
horizontal

desplazamiento  
vertical

color



difuminado de borde (opcional)



sombra interna (opcional)



```
box-shadow: 4px 4px 4px grey;
```

```
box-shadow: 0 0 28px grey inset;
```

No es necesario declarar la  
unidad si el valor es 0

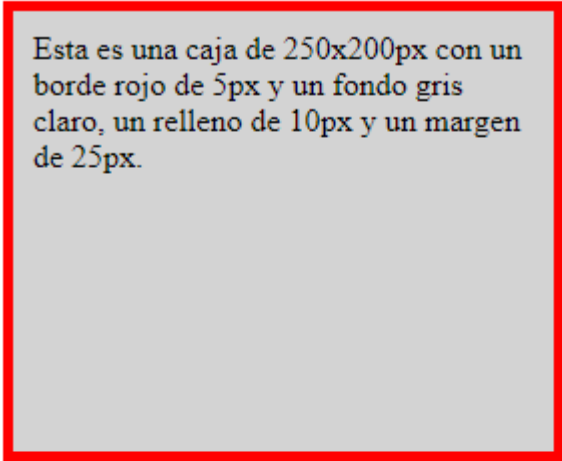
explotando box-shadow...

```
box-shadow:  
0 0 0 2px #000,  
0 0 0 3px #999,  
0 0 0 9px #fa0,  
0 0 0 10px #666,  
0 0 0 16px #fd0,  
0 0 0 18px #000;
```

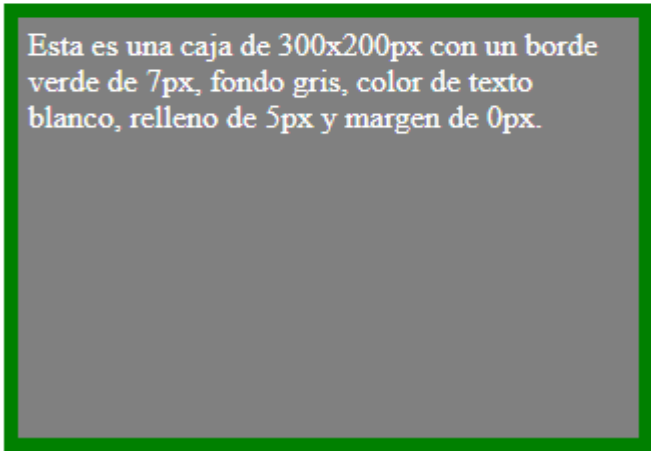
generación de varios bordes usando sombras  
**OJO:** estas no empujan



**Actividad.** Ambas cajas tienen una clase definida en CSS que establece los valores de las propiedades width, height, margin, border y padding. A partir de este ejemplo crea una página web con cinco cajas diferentes, cada una con diferentes estilos. Añade algunas sombras al texto usando diferentes estilos y colores.



Esta es una caja de 250x200px con un borde rojo de 5px y un fondo gris claro, un relleno de 10px y un margen de 25px.



Esta es una caja de 300x200px con un borde verde de 7px, fondo gris, color de texto blanco, relleno de 5px y margen de 0px.

## 4. Propiedades de texto

# font-family

Variamos la tipografía de fuente con la propiedad CSS font-family.  
Podemos usar una que tengamos instalada o importar una externa.

Es recomendable declarar más de una fuente por si el navegador del usuario que visualiza la página tiene problemas para utilizar la fuente declarada (como única o primera opción).

```
font-family: 'Courier New', Courier, monospace;
```

Glifo



Opciones de importación  
de fuentes de Google Fonts

```
<meta>
...
<link href="https://fonts.googleapis... rel="stylesheet">
...
</meta>
```

HTML File

```
font-family: 'Syne Mono', monospace;
```

CSS Rule

To embed a font, copy the code into the  
<head> of your html

☒ <link> ☐ @import

```
<link href="https://fonts.googleapi
s.com/css2?family=Syne+Mono&display=
swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Syne Mono', monospace;
```

To embed a font, copy the code into the  
<head> of your html

☐ <link> ☒ @import

```
<style>
@import url('https://fonts.googleapi
s.com/css2?family=Syne+Mono&display=
swap');
</style>
```

CSS rules to specify families

```
font-family: 'Syne Mono', monospace;
```



# font-weight

Propiedad CSS asociada al grosor de los glifos.

Podemos usar los **valores**:

- bolder | bold | normal | light | lighter
- numéricos en el dominio de las centenas: 100..200..**400** (valor por defecto)..600..800
- valores de herencia: inherit | unset | initial

`font-weight: bold;` Equivale a... `font-weight: 700;`

`font-weight: normal;` Equivale a... `font-weight: 400;`

Hola alumnos de LM!  
**Hola alumnos de LM!**

# font-style

Propiedad CSS asociada al estilo itálica.

**Valores:** normal | italic | oblique | initial | inherit

```
font-style: italic;
```

# vertical-align

Alineación vertical de los elementos en base a una referencia.

**Valores:** baseline | length | sub | super | top | text-top | middle | bottom | text-bottom | initial | inherit

```
<p>Hola alumnos de.. <span id="superindice">Lenguajes de Marcas!</span></p>  
<p>Hola alumnos de.. <span id="subindice">Lenguajes de Marcas!</span></p>
```

```
#superindice { vertical-align: super; }  
#subindice { vertical-align: sub; }
```

# font-variant

Propiedad que consigue un texto en mayúsculas con resalte en la primera letra de cada frase.

**Valores:** normal | small-caps | initial | inherit;

HOLA ALUMNOS DE LENGUAJES DE MARCAS!

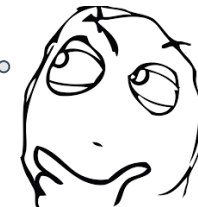
Hola alumnos de.. Lenguajes de Marcas!

Hola alumnos de.. Lenguajes de Marcas!

¿Esto lo puedo hacer  
con **<sup>** y **<sub>**



Así es crack!



# line-height

Propiedad CSS asociada al interlineado.

**Valores:** normal | *<valor&unidad>* | length | initial | inherit

```
line-height: 30px;
```

- |          |          |
|----------|----------|
| • Que    | • Que    |
| • pasada | • pasada |
| • esto   | • esto   |
| • del    | • del    |
| • CSS3!  | • CSS3!  |

# word-spacing

Propiedad que consigue un espaciado entre palabras (positivo | negativo)

**Hola alumnos!**

**Holaalumnos!**

# letter-spacing

Propiedad que consigue un espaciado entre caracteres (positivo | negativo)

```
letter-spacing: 2px;
```

**Hola alumnos de LM**

**Hola alumnos de LM**

**HolaalumnosdeLM**

# text-align

Propiedad CSS asociada al **centrado de texto**.

Podemos usar los valores:

➤ left | right | **center** | justify | initial | inherit;

**IMPORTANTE:** para justificar  
texto la W3Schools recomienda  
añadir una segunda regla css:  
`text-justify: inter-word;`

**IMPORTANTE:** no centramos  
texto con etiquetas HTML4  
<center>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Culpa, est!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Culpa, est!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Culpa, est!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatibus maxime distinctio fugit esse quam dolores, ipsum molestiae ducimus temporibus architecto commodi iusto eos, eveniet voluptates officia. Ipsam inventore corrupti atque.

# text-decoration

Propiedad CSS asociada al subrayado / tachado de texto.

Podemos usar los valores:

➤ overline | line-through | underline | none | *<multiple values>*

Aprende CSS

~~Aprende CSS~~

Aprende CSS

Aprende CSS

# color

Propiedad CSS asociada al color del texto de un elemento.

Podemos usar nombres de color, hexadecimal, rgb, rgba, hsl..

*(\*) Estudiaremos los colores en esta unidad.*

`color: violet;`

`color: #fff;`

`color: rgba(255, 0, 0, .5);`

# text-transform

Propiedad CSS que permite variar el texto entre mayúsculas / minúsculas y efecto capitalize:

Valores: uppercase | lowercase | normal | capitalize

# text-shadow

Propiedad CSS que permite definir sombras en el texto.

```
text-shadow: 2px 2px;  
text-shadow: 2px 2px blue;  
text-shadow: 2px 2px 5px blue;
```

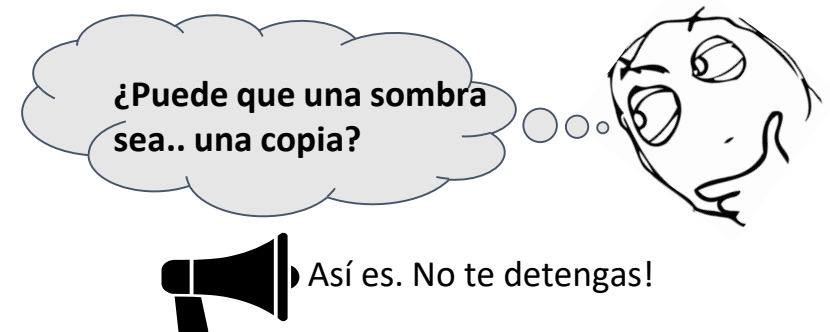
Parámetros opcionales:  
difuminado y color

Aprende CSS3 con nosotros!

aprende css3 con nosotros!

APRENDE CSS3 CON NOSOTROS!

Aprende CSS3 Con Nosotros!



# 5. Coiores

# La panorámica de colores en CSS

Podemos definir colores con los siguientes formatos:

➤ Nombres de colores (alrededor de 140 nombres distintos)



➤ RGB (Red, Green, Blue) ó RGBA (4º valor: opacidad)

color: `rgb(255, 0, 0);`      background-color: `rgba(255, 0, 0, .5);`



➤ Hexadecimal: réplica del RGB con valores declarados en HEX (podemos omitir un carácter hexadecimal para un valor r,g o b si hay carácter duplicado)

border: 1px dashed `#ff0000` (ó `#f00`);



➤ HSL (Hue, Saturation, Lightness) ó HSLA (4º valor: opacidad)

background-image: linear-gradient( `hsl(0, 100%, 50%)` , `hsla(0, 100%, 50%, .5)` );





# Aplicaciones

**IMPORTANTE:** debemos pensar en el conjunto de colores con el que “vestimos” una página web (Paleta de colores).

**Color de texto:** propiedad color

**Color de fondos:** background-color

**Gradientes:** lineales / radiales

**Bordes:** border / outline

**Sombras:** texto / caja

Réplica de color con la herramienta recoge color de M.Firefox



# RGB / RGBA

Combinación de colores primarios Red Green Blue.

Valores r, g y b comprendidos entre el 0 y 255  
(posibilidad de declarar  $255^3$ ) colores distintos.

RGBA permite declarar un 4º valor (alpha)  
opacidad en el rango [0-1]

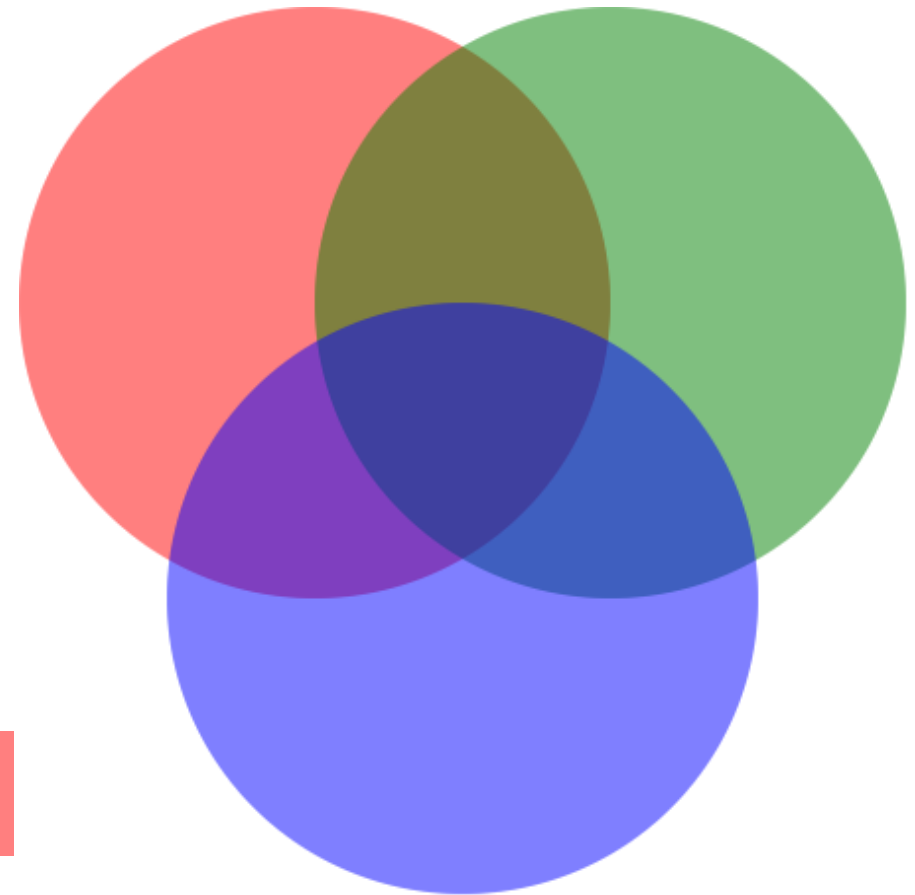
```
color: rgb(255, 0, 0);
```



```
background-color: rgba(255, 0, 0, .5);
```



Opacidad



# HEX

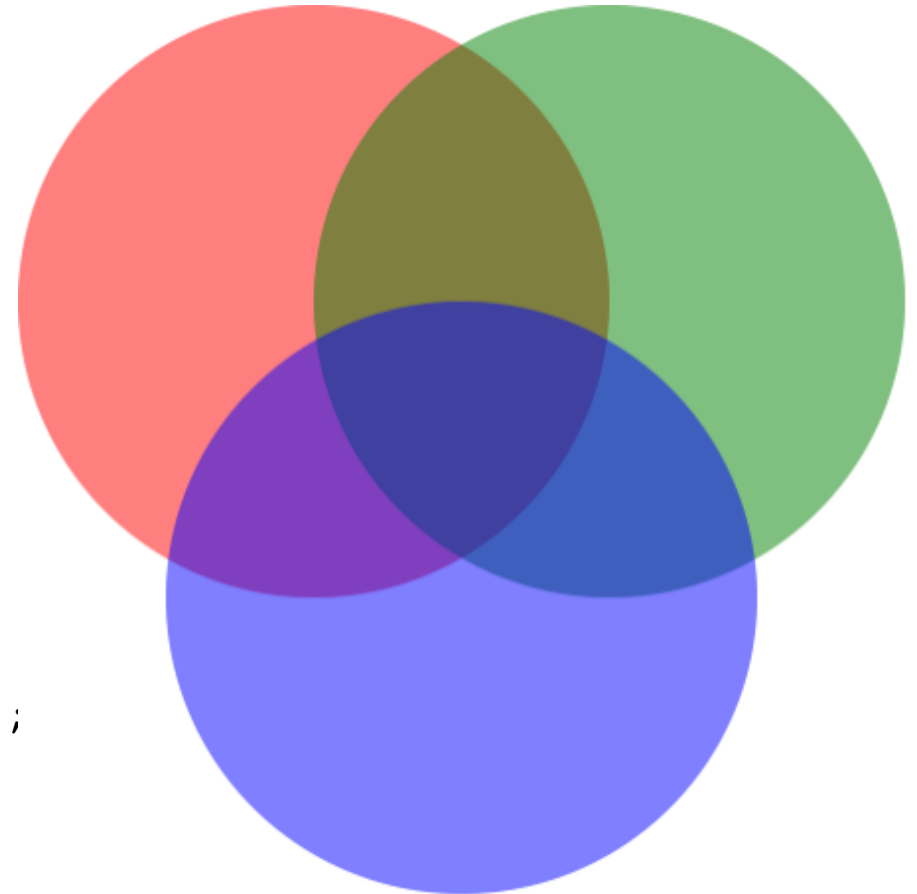
Fundamento idéntico a RGB basado en colores primarios.

Formato de valores para cada componente basado en el sistema numérico hexadecimal.

Posibilidad de omitir caracteres si estos están duplicados para un valor r, g o b.

```
border: 1px dashed #ff0000;
```

```
border: 1px dashed #f00;
```



# HSL (Hue Saturation Lightness) / HSLA

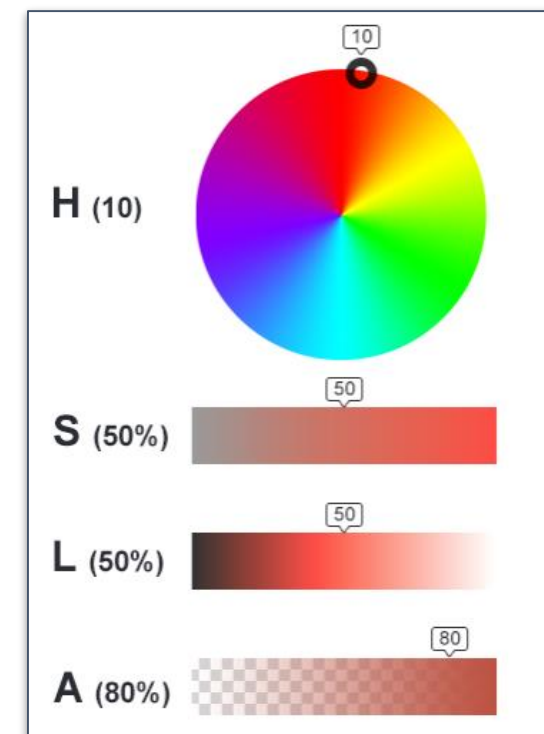
Matiz, Saturación y Luminosidad

**Matiz:** grados del círculo cromático. Valor de inicio 0: rojo.

**Saturación:** el valor 0 será un gris y 100 su máximo esplendor.

**Luminosidad:** el valor 0 será la oscuridad total y 100 la luminosidad total

**Alpha (opacidad):** Valor contenido en el rango [0-1]



```
background-image: linear-gradient( hsl(0, 100%, 50%) , hsla(0, 100%, 50%, .5) );
```

```
background-image: radial-gradient( hsl(0, 100%, 50%) , hsla(0, 100%, 50%, .5) );
```

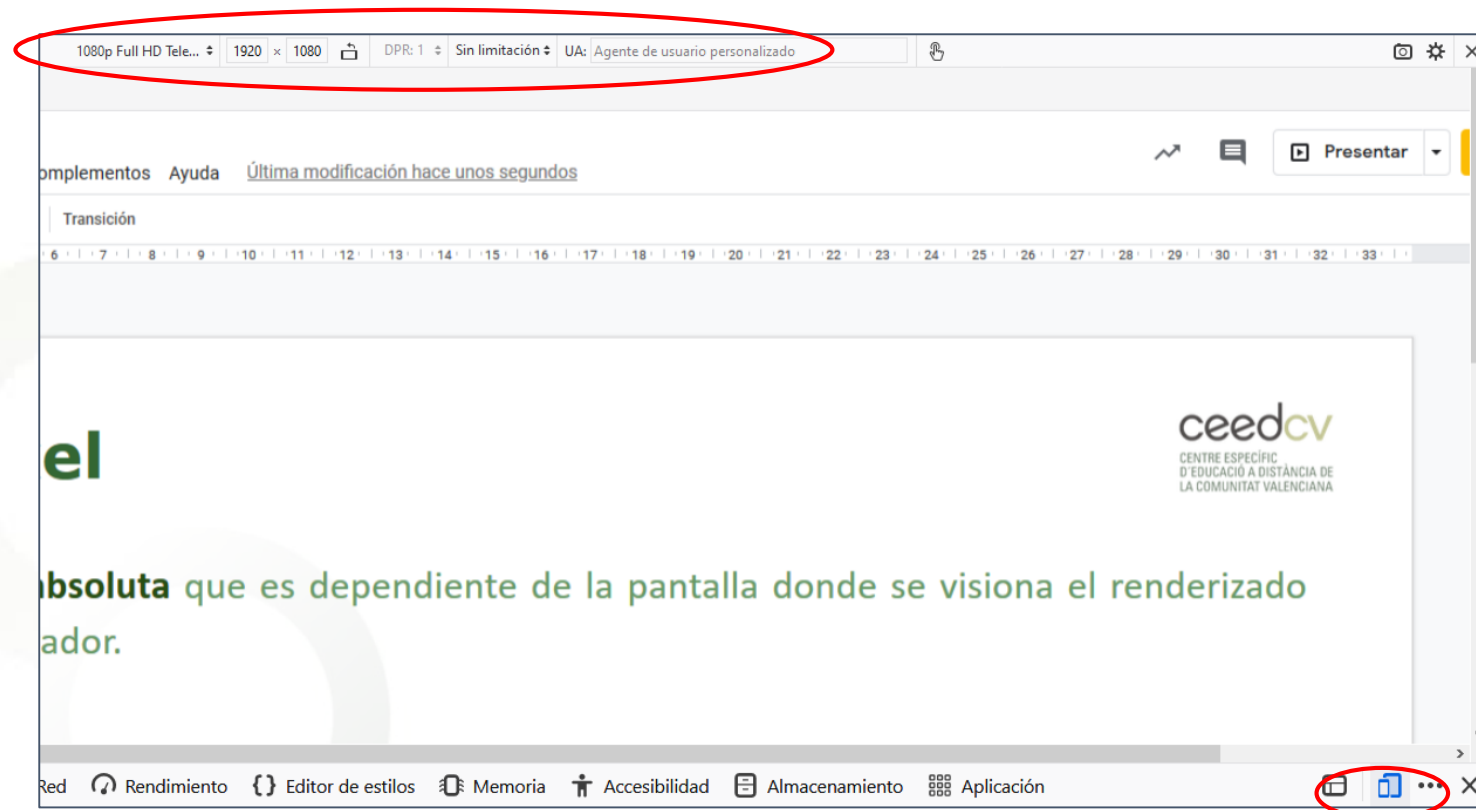
## 6. Unidades de medida

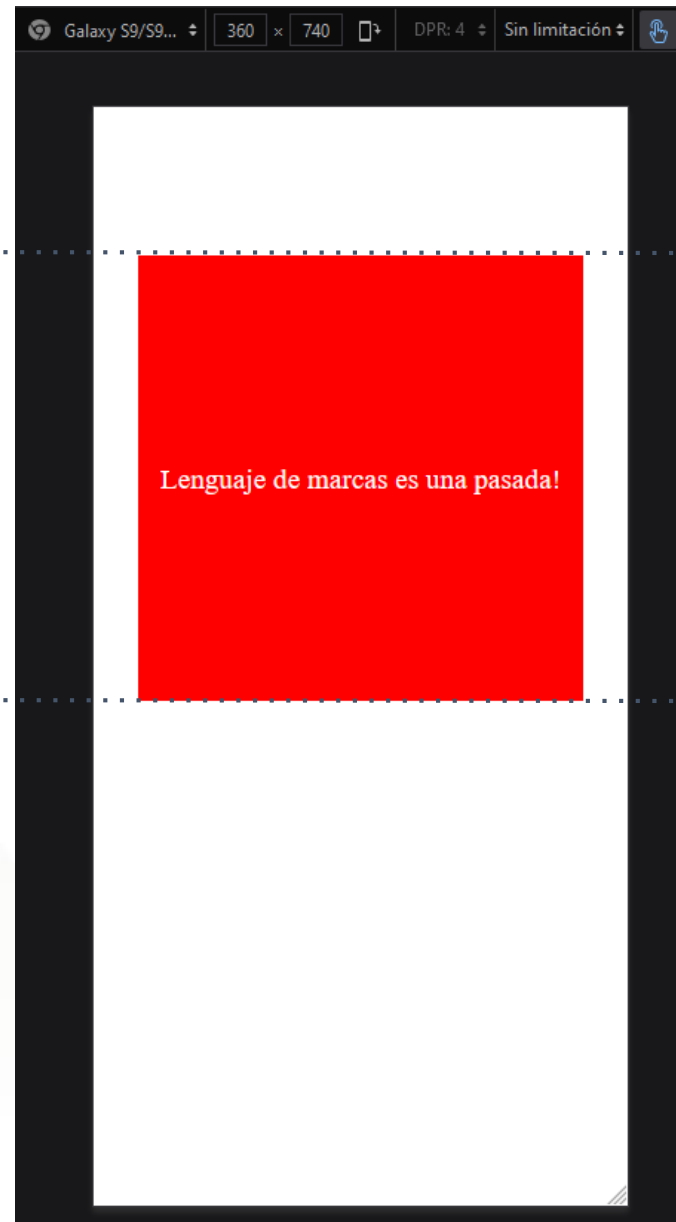
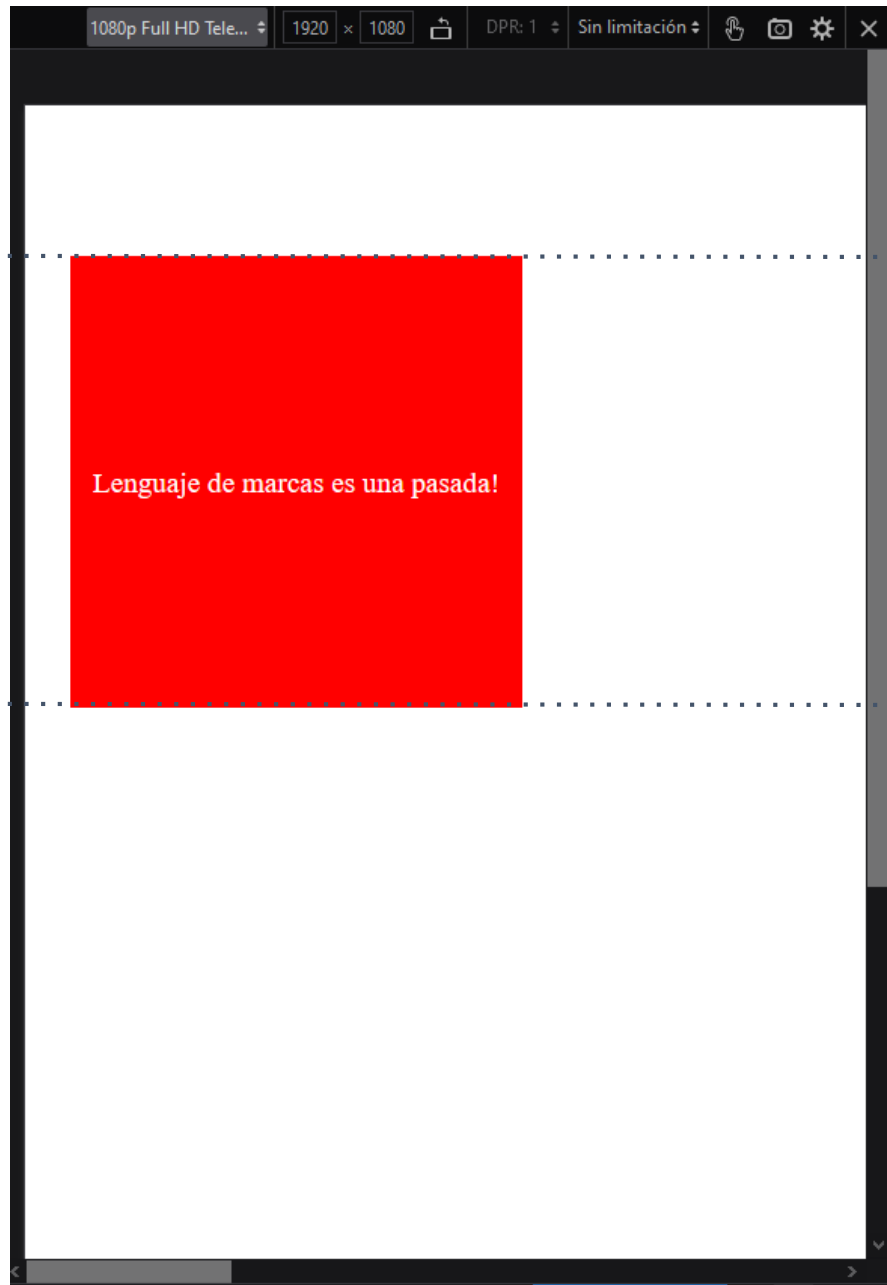
# El píxel

**Medida absoluta** dependiente de la pantalla donde se visiona el renderizado del navegador.

Podemos saber el pixelaje de nuestro visor con el inspector de código (F12).

Podemos probar con diferentes configuraciones de pantallas de dispositivos y emular el aspecto visual sobre los mismos.





```
div{  
  height: 300px;  
  width: 300px;  
  background-color: ...  
}
```

# %

**Medida relativa** que permite definir las dimensiones de un elemento **con respecto a las de su contenedor padre**.

```
.externo{  
  width: 300px;  
  height: 300px;  
  background-color: antiquewhite;  
}  
  
.interno{  
  background-color: black;  
  height: 50%;  
  width: 50%;  
}
```





# REM

**Medida relativa** que permite establecer dimensiones de texto y elementos en base al **tamaño de fuente declarado en el elemento raíz (<html>)**.

`<html>`  
`1rem = 16px` →  
`¿font-size: 16px?`  
`32px`

`2rem =`

**TIP:** (Si declaramos 16px en root..)  
Usar fracciones de 16, ayudará a  
armonizar el aspecto visual final.

`.5rem = 8px`

CSS

HTML

```
<span>Píxeles</span>
<span>REM</span>
<span>REM text x 2</span>
```

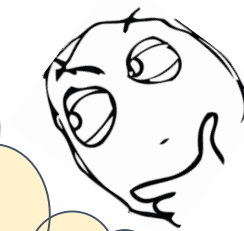
```
html{ font-size: 16px; } /*medida por defecto*/
span:nth-child(1){ font-size: 16px; }
span:nth-child(2){ font-size: 1rem; }
span:nth-child(3){ font-size: 2rem; }
```

Píxeles  
REM

REM text x 2



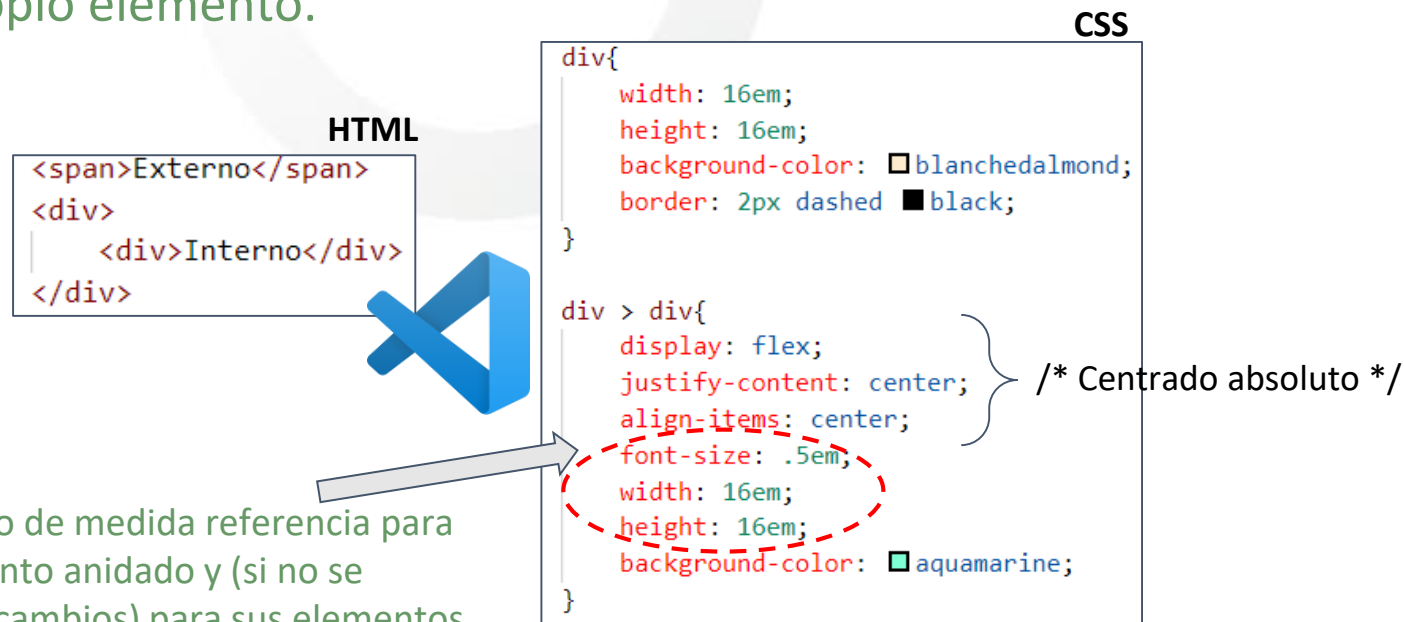
¿Si declaro un font-size en la raíz más grande o pequeño y el resto de elementos de mi web dependen de la unidad rem, será más grande o pequeño el site verdad?



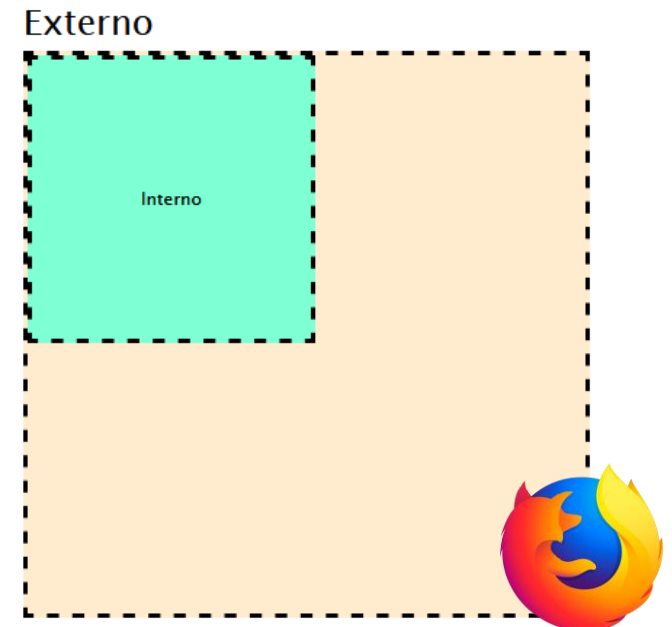
Matrícula de  
honor vitalicia

**Medida relativa** que permite establecer dimensiones de texto y elementos en base al **tamaño de fuente que hereda el propio elemento**.

Recordad que la relación de herencia viene siempre de la mano de los contenedores “padre” del propio elemento.



Recálculo de medida referencia para el elemento anidado y (si no se definen cambios) para sus elementos internos.



# VH (viewportheight) / VW(viewportwidth)

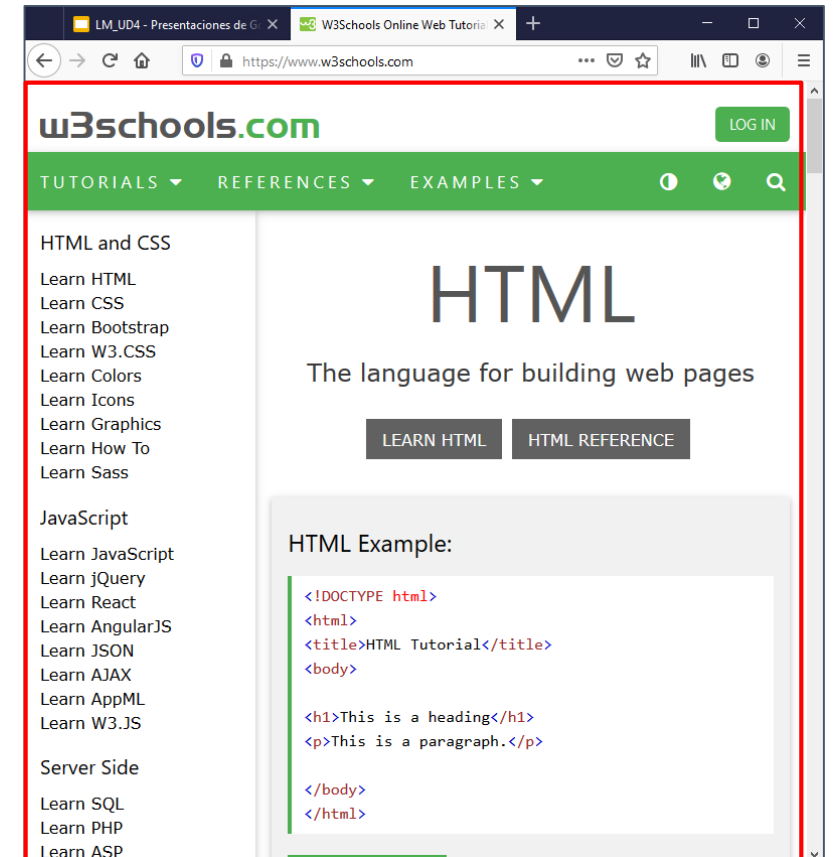
**Viewport:** Visor del navegador ( no incluye la barra de scroll ni marcos/bordes del mismo)

El viewport puede variar de tamaño en cualquier momento.. si agrandamos o encogemos la ventana del navegador.

1vh es un 1% del alto del visor.

1vw es un 1% del ancho del visor.

Son **medidas relativas** al visor y contribuyen a realizar escalados dinámicos de elementos propios del diseño responsivo web.



Área del viewport

## 7. Posicionamiento float/clear

# Estamos en 2021/2022.. float se usa?

## Se usa y se usará



CSS Float



CSS Position



CSS Flexbox

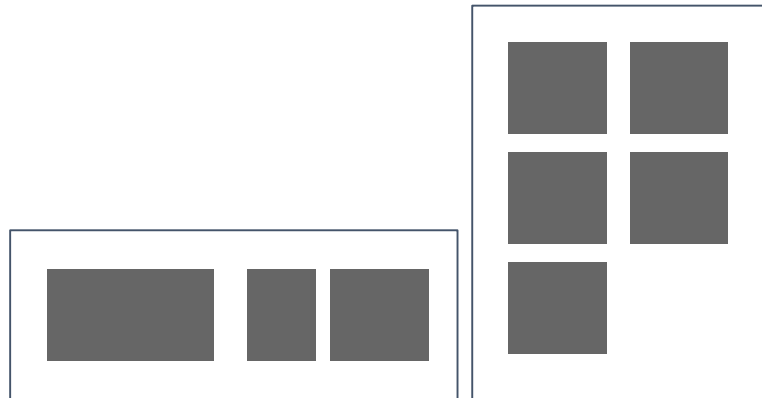
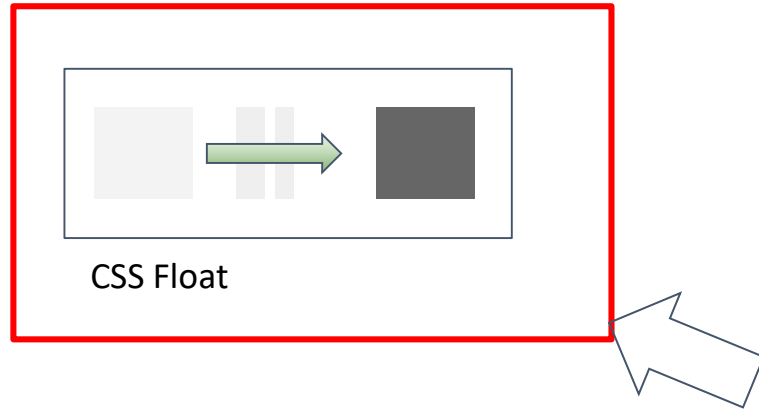


CSS Grid

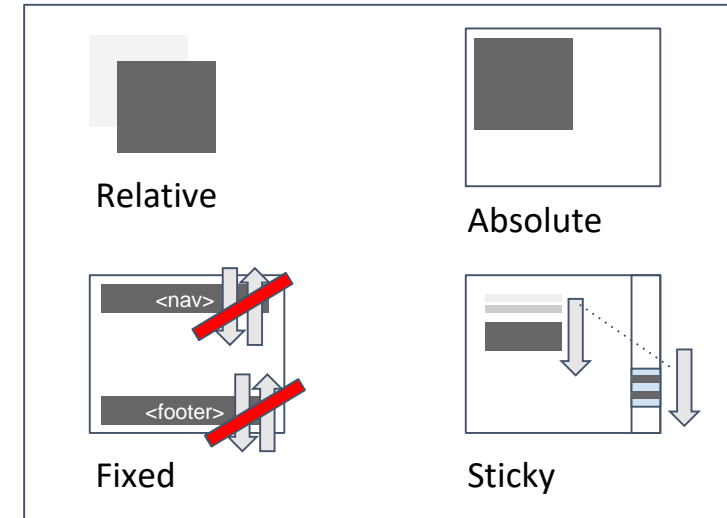


Un símil de herramientas más o menos avanzadas para la construcción de un edificio pero todas ellas útiles en un determinado contexto.

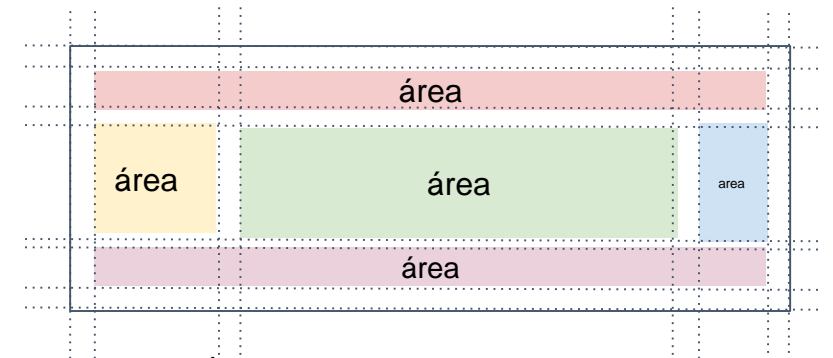
# Posicionamiento / Layout



CSS Flexbox



CSS Position



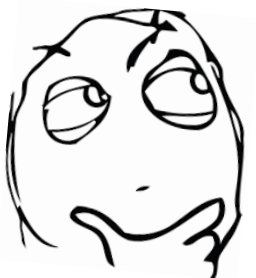
CSS Grid

# Antes de ver float.. recordemos lo básico

## Elementos de bloque

- Ocupan todo el espacio horizontal disponible.
- Crean una nueva línea (El siguiente elemento será posicionado debajo).
- Puede definirse un ancho (width) y alto (height).

<div>, <h1..h6>, <p>, <pre>, <header>, <footer>..



## Elementos de línea

- Ocupan sólo el espacio necesario para su contenido.
- Se disponen los elementos de forma contigua hasta que no quepan en la misma línea (entonces se produce el salto de línea).
- No tienen dimensiones (A excepción de las imágenes).

Generalmente son aquellos elementos referidos a texto..

<span>, enlaces <a>, <input>, <img>..

## Elementos inline-block

- Ocupan sólo el espacio necesario para su contenido ó puede definirse un ancho (width) y alto (height).
- Se disponen los elementos de forma contigua hasta que no quepan en la misma línea (entonces se produce el salto de línea).
- Espaciados internos/externos son tenido en cuenta.

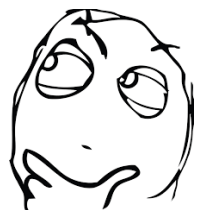
A este estado se alcanza variando la naturaleza del elemento con la propiedad/valor: **display: inline-block;**

# Float

Herramienta básica para posicionar (con relativo poco esfuerzo) elementos a la izquierda o derecha de un contenedor o área.

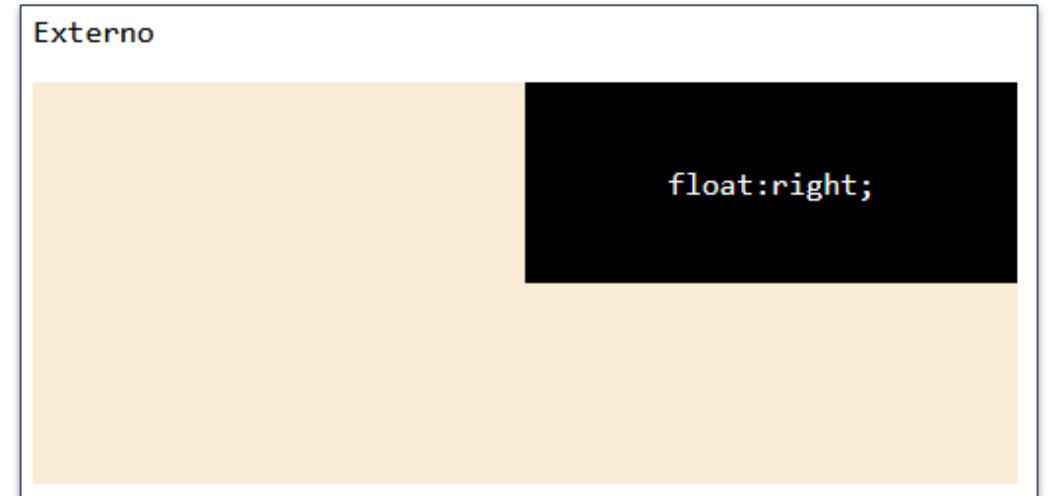
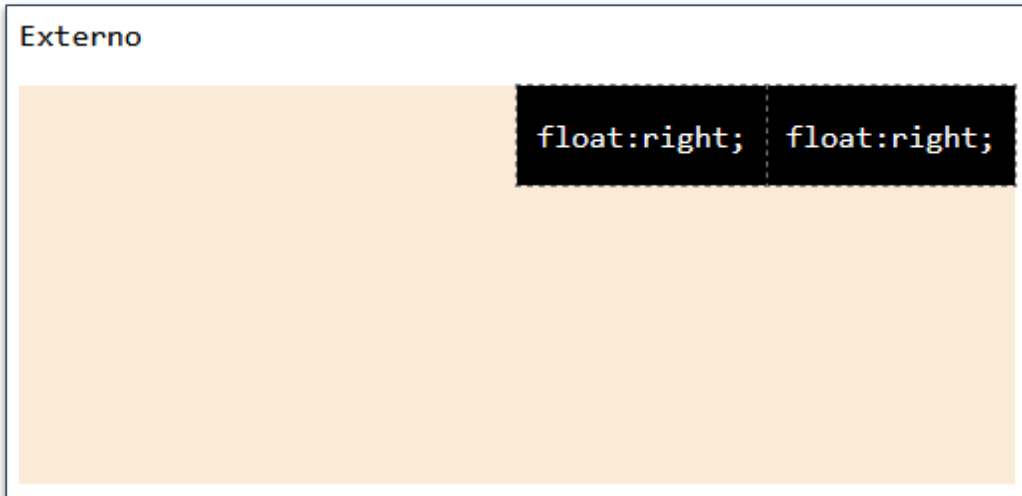
Un elemento “flotado” libera su espacio reservado en el flujo HTML. Podemos imaginar que los elementos “flotados” viven en una capa aparte en el eje Z.

Si un elemento “flotado” es interno a un contenedor padre y no hay más elementos que conserven su posición original o el padre no tiene dimensiones declaradas, el contenedor perderá la noción del espacio reservado del elemento en cuestión (y menguará su tamaño).

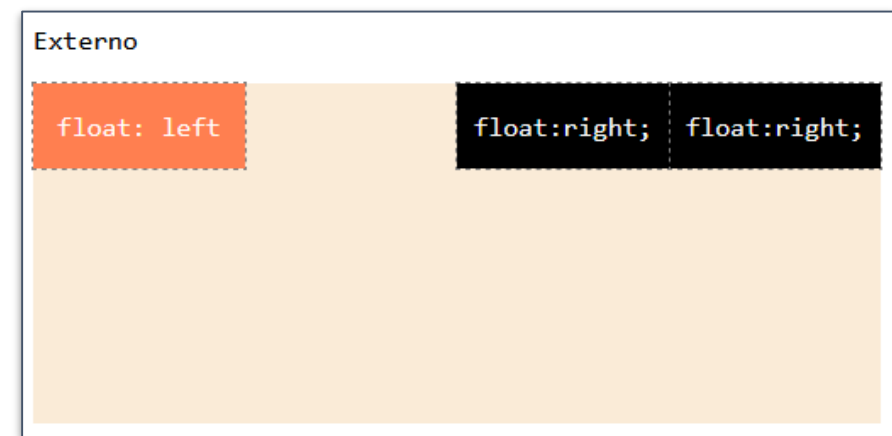
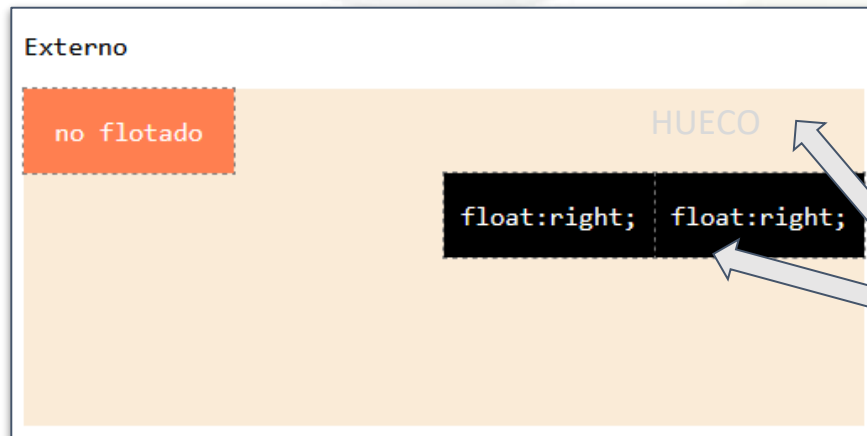




# Float

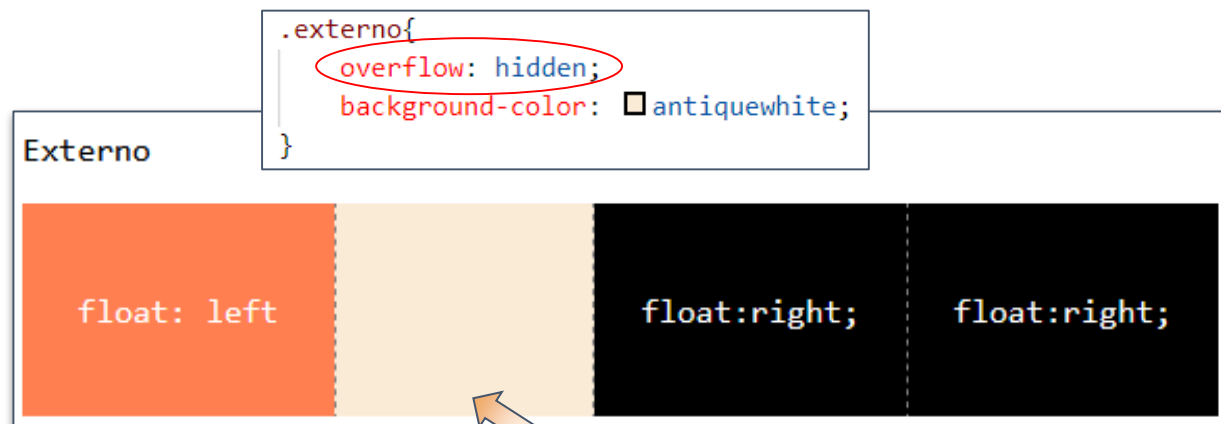
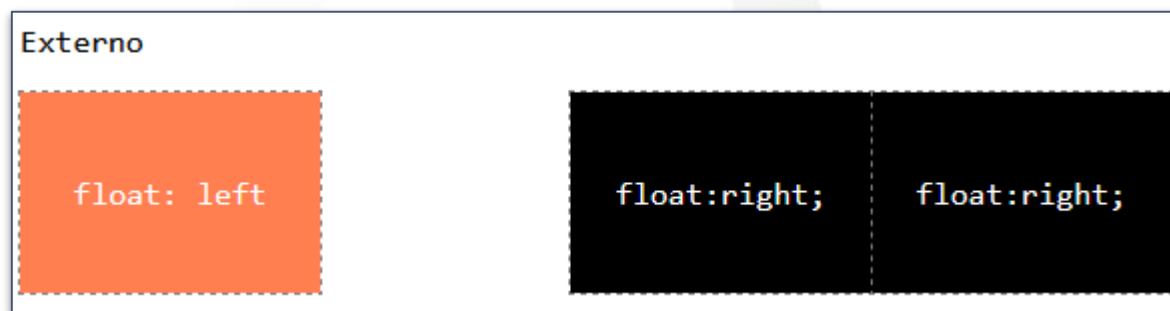


El contenedor **padre** tiene unas **dimensiones fijas**: width: 500px height: 200px



# Float & Overflow: hidden

Overflow:hidden hace que el contenedor padre sin dimensiones “abraze” la totalidad de elementos flotados internos.



El contenedor “abraza”  
sus elementos “hijos” flotados

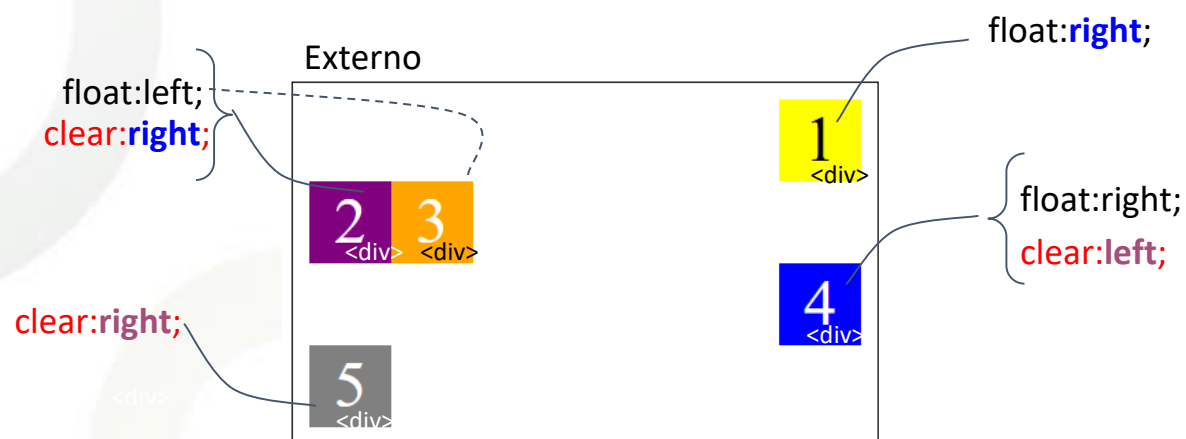
Ahora el contenedor **padre** no tiene **dimensiones ..**

# Float & Clear

La propiedad clear especifica qué elementos pueden flotar junto al elemento eliminado y en qué lado. La forma más común de usar la propiedad clear es después de haber usado una propiedad flotante en un elemento.

## VALORES CLEAR:

- none:** permite elementos flotantes en ambos lados. Esto es predeterminado
- left:** no se permiten elementos flotantes en el lado izquierdo
- right:** No se permiten elementos flotantes en el lado derecho
- ambos:** no se permiten elementos flotantes ni en el lado izquierdo ni en el derecho
- heredar:** el elemento hereda el valor clear de su padre



**IMPORTANTE:** Al usar **clear** se debe hacer coincidir con **float**: si un elemento flota a la izquierda, entonces debe despejar a la izquierda. Su elemento flotante seguirá flotando, pero el elemento borrado aparecerá debajo de él en la página web.

Un contenedor con todos sus elementos "flotados" pero que estos definan la propiedad clear recupera la noción de espacio de sus elementos internos.

La teoría es imprescindible dado que el **estudio formal** de CSS3 es complejo.



Solo con la **práctica**, alcanzareis la subida de nivel que se espera de vosotros!!! Buscad cualquier excusa para crear vuestros desarrollos personales (aficiones, asociaciones, recrear websites..)



**15 years** of musician



A new **master** has born



# Bibliografía / Webgrafía

W3Schools: <https://www.w3schools.com/>

Documentación Mozilla Firefox: <https://developer.mozilla.org/es/>