



CSS3

M04 - Lenguaje de Marcas

Alicia Vázquez



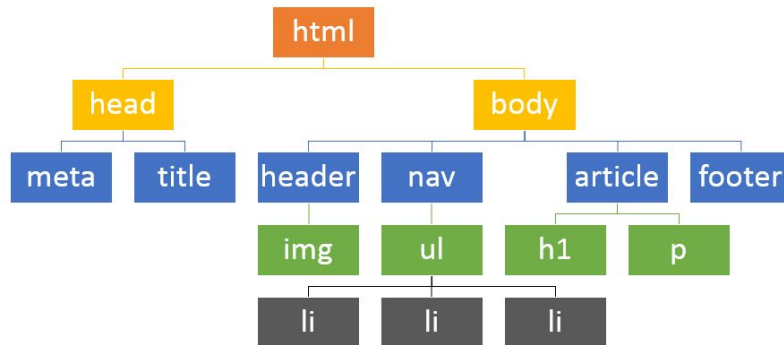
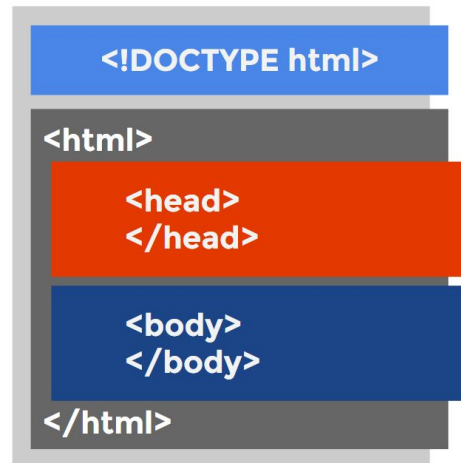


HTML5 - recap

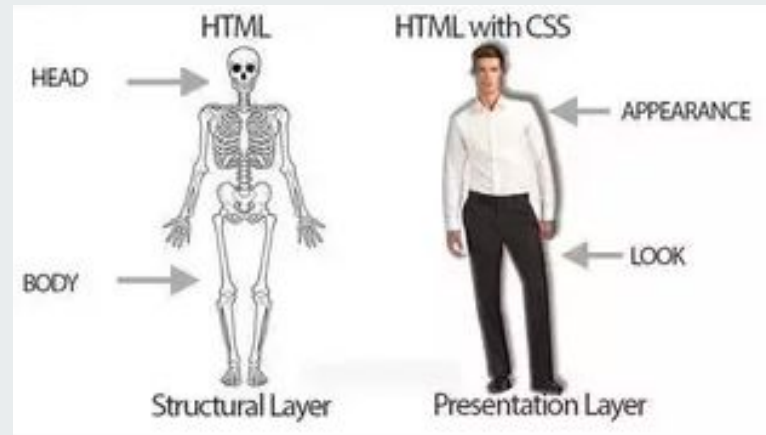
HTML5

Todo documento HTML tiene un mínimo de elementos y una **estructura** muy marcada siendo `<html>` la raíz del documento.

- En `<head>` pondremos los elementos necesarios para el navegador, no hay datos.
- En `<body>` es donde pondremos toda la información visible al usuario (texto, imágenes, videos...).
- HTML5 ofrece una **valor semántico** a todos sus tags, de esta manera podremos distribuir correctamente los datos y así facilitar su comprensión por parte de las “máquinas”.



CSS3



CSS

CSS (Cascading Style Sheets), en español "Hojas de estilo en cascada", es un **lenguaje de diseño gráfico** para definir y crear la **presentación** de un documento estructurado escrito en un lenguaje de marcado.

Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en un lenguaje de marcas.

CSS está diseñado principalmente para marcar la separación del **contenido** (html) del documento y la forma de **presentación** de este, características tales como las capas o layouts, los colores y las fuentes.



Cascading Style Sheets

Hojas de estilo en cascada: Usando un solo fichero css podemos marcar el estilo de todas las páginas html de nuestro proyecto.



Guardaremos en el fichero CSS principal el estilos general de nuestras páginas, de nuestra web y cambiaremos concretamente el estilo en un html si fuera necesario.



Definir un estilo en cada HTML, lo que dificulta el mantenimiento.



Sintaxis de CSS



```
etiqueta {  
    propiedad: valor;  
    propiedad: valor;  
}
```

Los elementos serán aquellos que encontramos en el **body**, no los del **head**, puesto que estos no se visualizan.

```
h1{  
    color: red;  
} /*Esto sería un comentario*/
```

Enlazar css con html

1. Añadir enlace al css en el **<head>** de la página html mediante **<link>**

```
<head>
  <link rel="stylesheet" href=" 01.css">
</head>
```

2. Definir directamente el estilo dentro del **<head>** de la página html usando el tag **<style>**

```
<head>
  <style>
    elemento{
      propiedad1: valor;
      propiedad2: valor;
    }
  </style>
</head>
```


Enlazar css con html

3. Directamente en el elemento en el que queremos aplicar o cambiar el estilo.

- a. Podemos usar la propiedad **style** que tendrá como valor todo el código del estilo separado por puntos y comas.

```
<elemento style="propiedad1: valor; propiedad2: valor;">  
</elemento>
```

- b. Podemos ir propiedad a propiedad e ir cambiando su valor.

```
<elemento propiedad1="valor" propiedad2="valor">  
</elemento>
```



Estas opciones no son recomendables puesto que son muy difíciles de seguir y de mantener.

Código ordenado



Nuestro código debe ser legible, entendible y mantenible por nosotros mismos a lo largo del tiempo y por cualquier otra persona de nuestro equipo de desarrollo. Para ello es muy importante seguir una serie de pautas que nos permiten mantener un código ordenado:

- **Indentar** siempre nuestro código.
- Elegir nombres (identificadores) que sean **descriptivos**, que explique qué hacen.
`<div class="content-video"> ... </div>`
- Seguimos el **mismo orden** en HTML que en CSS. Las características globales las pondremos arriba del todo. Añadir un índice en nuestro fichero CSS siempre ayuda a ubicarnos.



CSS3

Conceptos básicos

- ❏ Sintaxis
- ❏ Propiedades básicas

Sintaxis de CSS



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

- **selector:** En su forma más sencilla hace referencia a la misma etiqueta HTML a la que queremos asignar un estilo, un identificador, una clase o selectores complejos.
- **propiedad:** existen una lista de propiedades definidas.
- **valor:** según la propiedad tendrán una característica concreta.

El código de **CSS** irá en un archivo con extensión ***.css**

Estructura de CSS



```
selector {  
    propietat: valor;  
    propietat: valor;  
    (...)  
}
```

```
selector {  
    propietat: valor;  
}  
  
(...)
```

```
/* comentari css d'una linia*/  
/* 0 de més d'una  
Linia */
```

Propiedades básicas



- CSS está en continua evolución y los navegadores van añadiendo nuevas funcionalidades.
 - La web [Can I Use](#) no permite saber si está o no en uso una atributo o funcionalidad.
- Existen cientos de propiedades y es imposible conocerlas todas, pero trataremos de ver las más importantes y las que nos permitan hacer el máximo de cosas.
 - En este enlace encontraréis la [Referencia CSS del Mozilla Developer Network](#)

Propiedades - Text

- **font-family:** tipo de letra
 - `font-family: Helvetica`
 - `font-family: "Trebuchet MS", Verdana, sans-serif;`
- **font-size:** tamaño de letra.
 - `12px | 1.5em`
 - `1em` => tamaño actual. `Xem` => `X*tamaño actual`
- **font-weight:** grosor (`normal` | `bold` | `lighter` | `bolder`)
- **font-style:** estilo (`normal` | `italic` | `oblique`).
- **letter-spacing:** espacio entre letras.
- **line-height:** espacio entre líneas / altura de la línea.
- **text-decoration:** estilos en el texto (`underline` | `overline`)
- **text-align:** alineación del texto (`left` | `right` | `center` | `justify`)

Propiedades para colores

- **color:** color del elemento y los valores pueden definirse en diferentes formatos ([más ejemplos](#))
 - Nombre del color
 - Valor [hexadecimal](#)
 - `#00ff00`
 - `#RedGreenBlue`
 - Valor RGB
 - `rgb(214, 122, 127)`
 - Valor RGBA (alfa)
 - `rgba(214, 122, 126, 0.5)`

Propiedades para colores y fondos



- **background-color**: color del fondo del elemento.
- **background-image**: usar una imagen de fondo.
- **background-repeat**: usar una imagen de fondo como mosaico. Permite diferentes modos de organización de la imagen (ver detalles en material de referencia).
- **opacity**: opacidad del elemento. Va desde 0 (completamente transparente) hasta 1 (sólido). Un valor de 0.5 sería, por tanto, un nivel de transparencia del 50%.

Ejercicio 1 - Snatch

A partir del código HTML, crea un fichero CSS con los estilos necesarios para que la página tenga esta apariencia. No se puede modificar el fichero html, solo para añadir el enlace al css.

Snatch (cerdos y diamantes)

Año de producción

2000

Sinopsis

Franky (Benicio del Toro) es un ladrón de diamantes que tiene que entregar una enorme pieza a su jefe *Avi* (Dennis Lehane) para apostar en un combate ilegal de boxeo.

En realidad, es una trampa para robarle, por lo que, cuando *Avi* se entera, contrata a *Tony Dientes de Bala* (Vinnie Jones) y la triste destino de *Franky*, y la caza y captura de la gema desaparecida lleva a todo el mundo a un juego de locos que mezclan de forma sangrienta con gánsters, un tinker (Brad Pitt) que pelea para poder comprarle una caravana a su jefe de armas en una caótica situación... y en medio de todo ello un gran diamante.

Actores

- I. Jason Statham
- II. Brad Pitt
- III. Benicio Del Toro
- IV. Denis Farina

Urls

- I. [IMDB](#)
- II. [MovieDB](#)

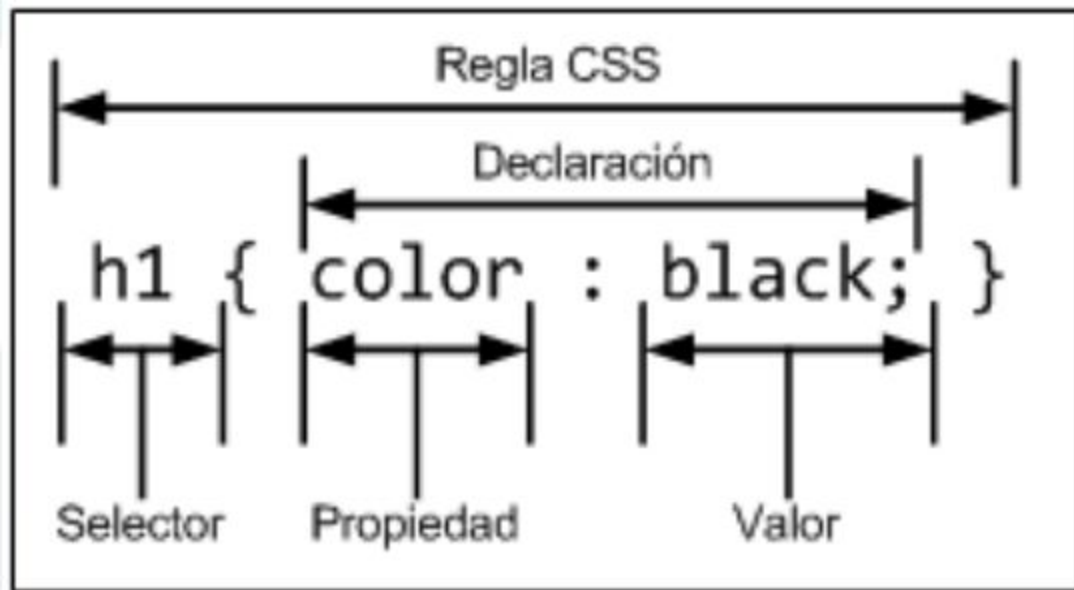


CSS3

Selectores

- ❏ Selector universal
- ❏ Selector por etiqueta
- ❏ Selector descendente
- ❏ Selector adyacente
- ❏ Selector de clase
- ❏ Selector de identificador

Selector



Selector universal o por etiqueta



¿A qué etiqueta debemos aplicar el estilo?

- Podemos querer aplicar un estilo a todos los elementos, a todas las etiquetas, hablamos de un selector universal *. Este diseño afectará a todos por igual.

```
*{  
    color: blue;  
    font-size: 2em;  
}
```

- Aplicaremos el estilo a la etiqueta HTML concreta a la que queremos definir el estilo.

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

Selector por etiqueta



Si varias etiquetas tienen el mismo estilo, podemos concatenarlas y así reducir código y hacerlo más mantenible. Agrupamos las propiedades comunes de varios elementos en una única regla CSS.

```
h2, h3{  
    color: #8A8E27;  
    font-weight: normal;  
    font-family: Arial, Helvetica, sans-serif;  
}
```

Esta es una práctica habitual para posteriormente definir las propiedades específicas de esos mismos elementos.

```
h2{font-size: 2em;}  
h3{font-size: 1.5em;}
```

Selector descendente

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.

```
p span{
    color: red;
}
h1 span{
    color: blue;
}
```

- Los elementos `` que se encuentran dentro de un elemento `<p>` se muestran de color rojo.
- Los elementos `` que se encuentran dentro de un elemento `<h1>` se muestran de color azul.
- El **resto** de elementos `` de la página, se muestran con el color por defecto aplicado por el navegador.

```
<body>
... <h1><span>ROJO</span></h1>
... <p><span>ROJO</span></p>
... <span>de cualquier color</span>
</body>
```

Selector descendente

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

- El estilo se aplica a todos los elementos "p", "a", "span" y "em"

```
p, a, span, em { text-decoration: underline; }
```

```
<body>
|   ... <p></p>
|   ... <a></a>
|   ... <span></span>
|   ... <em></em>
| </body>
```

- El estilo se aplica sólo a los elementos "em" que se encuentran dentro de "p a span"

```
p a span em { text-decoration: underline; }
```

```
<body>
|   ... <p><a><span><em></em></span></a></p>
| </body>
```


Selector descendente

```
<body>
... <p><a href="#">Enlace 1 </a></p>
... <p><span><a href="#">Enlace 2</a></span></p>
... <p><span><strong><a href="#">Enlace 3</a></strong></span></p>
</body>
```

- Si se emplea el selector descendente, hace referencia a cualquier punto de la descendencia (hijo, nieto, bisnieto...). Todos saldrán en rojo.

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

Enlace 1

Enlace 2

Enlace 3

- Sin embargo, al usar el selector universal, a partir del “nieto” (segundo nivel de descendencia) saldrá en rojo.

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

Enlace 1

Enlace 2

Enlace 3

Selector de descendente directo (hijo, nieto...)

- Siguiendo el ejemplo anterior, queremos que solo *Enlace 1* salga en rojo.
Etiquetas que están **directamente dentro (anidadas)** de otra etiqueta especificada.

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

Enlace 1

Enlace 2

Enlace 3

- Ahora queremos que solo el *Enlace 2* salga en rojo.
Etiquetamos usando el selector universal los anchor que estén dentro de algún tag que esté dentro de un párrafo.

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

Enlace 1

Enlace 2

Enlace 3

Selector descendente, hermanos

Hermanos: Etiquetas que están al mismo nivel, misma jerarquía que la indicada, aunque hayan otras etiqueta por medio.

```
<body>
  ... <p>Primer párrafo</p>
  ... <div>Otro hermano</div>
  ... <p>Segundo párrafo</p>
</body>
```

Primer párrafo

Otro hermano

Segundo párrafo

```
p ~ p { color: orange; }
```

Selector adyacente

Adyacentes: Etiquetas que están **justo después** de otra etiqueta concreta.

```
<body>
  ...<p>Primer párrafo</p>
  ...<div>Otro hermano</div>
  ...<p>Segundo párrafo</p>
  ...<p>Tecercer párrafo</p>
  ...<p>Cuarto párrafo</p>
</body>
```

`p + p {color: green;}`

Primer párrafo

Otro hermano

Segundo párrafo

Tecercer párrafo

Cuarto párrafo

Selector de clase (.)

Imaginemos que solo queremos aplicar un estilo a uno o varios párrafos concretos. Identificar, marcar el párrafo que nos interesa informando su atributo `class` con uno o varios valores.

```
<body>
  ...<p class="titular">Artículo sobre CSS</p>
  ...<p class="destacado titular">El CSS no es un lenguaje de programación. Se utiliza para...</p>
  ...<p>Clase que solo afecta<em class="destacado">solo que hay aquí dentro</em> fuera, no.</p>
</body>
```

Podemos definir en el CSS solo el nombre de la clase y así poder usarlo con las etiquetas que queramos o relacionarlo a una etiqueta acotando de esta manera su alcance.

```
.titular { color: green; }
.destacado { text-decoration: underline; }
em.destacado { color: blue; }
```

Artículo sobre CSS

El CSS no es un lenguaje de programación. Se utiliza para...

Clase que solo afecta solo que hay aquí dentro fuera, no.

Selector de identificador (#)

Usaremos un identificador id para definir un estilo muy concreto, para un momento determinado. Ese identificador debe ser único y no volveremos a usarlo en el resto de nuestro fichero HTML, en caso contrario habría que pensar si sería conveniente que fuera una clase y no un identificador.

```
<body>
  <p id="titular">Artículo sobre CSS</p>
  <p id="destacado">El CSS no es un lenguaje de programación. Se utiliza para...</p>
</body>
```

```
p#titular { color: green; }
p#destacado { text-decoration:
underline; }
```

Artículo sobre CSS

El CSS no es un lenguaje de programación. Se utiliza para...

Selector de atributo

Algunos elementos (etiquetas) de HTML pueden tener propiedades propias del elemento y según esta característica también vamos a poder definir un estilo.

```
<body>
...<h1 title="principal">Primer parrafo</h1>
...<h2 title="secundario">Otro hermano</h2>
...<h2 title="titulo">Otro hermano</h2>
...<h3 title="otro titulo">Otro hermano</h3>
...<h3 lang="es-ES">Español</h3>
</body>
```

Primer parrafo

Otro hermano

Otro hermano

Otro hermano

Español

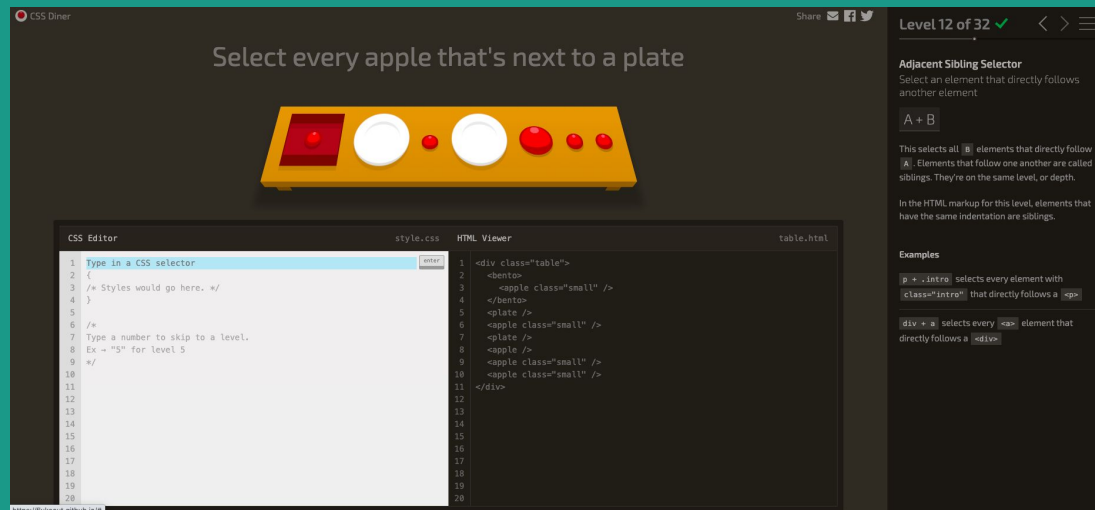
```
h1[title="secundario"] { color: orange; } /*que tenga ese valor*/
h2[title="titulo"] { color: green; }
h1[title] { color: red; } /*que tenga ese atributo*/
h3[title ~="titulo"] { color: green; } /*que contenga este
valor*/
h3[lang^="es"] { color: blue; } /*empieza por*/
```

Juega

Ya sé que todo esto puede parecer muy complicado, por eso te propongo un juego. Trata de hacer los niveles (1-14 y del 27 al 32), tienes una ayuda a tu derecha.

Puedes hacer el resto, son selectores que no hemos visto, pero con la ayuda y si te apetece seguro que lo consigues!

Flukeout



Pseudo clases y pseudo elementos

Las pseudo clases hacen referencia al estado de un elemento mientras que los pseudo-elementos hacen referencia a una parte del elemento.



Pseudo-clases

Links

Link
active
visited
hover

Interactive

focus
hover
active

Others

target
empty
root
not
lang(c)

Structural

first-child
first-of-type
nth-child(N)
nth-last-child(N)
nth-of-type(N)
nth-last-of-type(N)
last-child
last-of-type
only-child
only-of-type

Ejercicio 2 - Snatch

Modifica el fichero html del ejercicio anterior, añadiendo lo que sea necesario (clases) para que ahora tenga este aspecto. Sigue definiendo el CSS en un fichero a parte de CSS.

- h2 en degradado (cambia el tono del gris / cambio de estilo) sin usar *clases* ni *id*.
- asignar estilo según el actor
- modificar el estilo del anchor de la lista (ver y mayúsculas) sin modificar el texto en html.

Snatch (cerdos y diamantes)

Año de producción

2000

Sinopsis

Franky (Benicio del Toro) es un ladrón de diamantes que tiene que entregar una enorme pieza a su jefe **Avi** (Dennis Farina), pero en el camino es tentado por **Boris el Navaja** (Rade Serbedzija) para apostar en un combate ilegal de boxeo.

En realidad, es una trampa para robarle, por lo que, cuando **Avi** se entera, contrata a **Tony Dientes de Bala** (Vinnie Jones) para encontrar tanto a **Franky** como el diamante. Pronto se descubre el triste destino de **Franky**, y la caza y captura de la gema desaparecida lleva a todo el mundo a un juego de locos que corre el riesgo de descontrolarse, donde el engaño, el chantaje y el fraude se mezclan de forma sangrienta con gánsters, un tinker (Brad Pitt) que pelea para poder comprarle una caravana a su madre, combates de boxeo amañados, un perro hambriento y gran variedad de armas en una caótica situación... y en medio de todo ello un gran diamante.

Actores

- I. Jason Statham
- II. Brad Pitt
- III. Benicio Del Toro
- IV. Denis Farina
- V. Vinnie Jones
- VI. Rade Serbedzija

Urls

- I. IMDB
- II. MOVIEDB

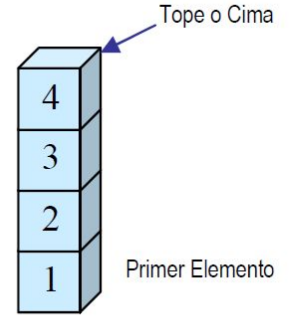


CSS3

Cascada

- Cascada
- Cascada vs Herencia

Cascada



En CSS es muy importante la **jerarquía**, programamos en cascada.

Orden en el que se aplican los estilos:

1. CSS del navegador que se esté usando.
2. Características que tenga configurado el usuario en su navegador.
3. Los archivos CSS enlazados en el orden en el que los han sido añadidos **<link>**, es decir que quedará cómo haya indicado el último archivo enlazado.
4. Las propiedades que se heredan, el estilo de mis padres...
5. Las propiedades propias del elemento, las específicas del elemento.
6. Clases asignadas, y en el mismo orden en el que estén definidas en el CSS (quedará como indica la última clase).
7. Los identificadores (**id**, debe estar escrito después del atributo **class**, si lo hay))
8. Escribir directamente el estilo en la propia etiqueta, en el código html.
9. **!important**, será lo último que tendremos en cuenta!

Stuff & Nonsense



a

1 x element selector

Sith power: 0,0,1



pa

2 x element selectors

Sith power: 0,0,2



.foo

1 x class selector *

Sith power: 0,1,0



a.foo

1 x element selector
1 x class selector

Sith power: 0,1,1



pa.foo

2 x element selectors
1 x class selector

Sith power: 0,1,2



.foo.bar

2 x class selectors

Sith power: 0,2,0



p.foo a.bar

2 x element selectors
2 x class selectors

Sith power: 0,2,2



#foo

1 x id selector

Sith power: 1,0,0



a#foo

1 x element selector
1 x id selector

Sith power: 1,0,1



.foo a#bar

1 x element selector
1 x class selector
1 x id selector

Sith power: 1,1,1



.foo .foo #foo

2 x class selectors
1 x id selector

Sith power: 1,2,0



style

1 x style attribute

Sith power: 1,0,0,0



* Same specificity
class selector =
attribute attribute =
pseudo-classes



!important

Cascada vs Herencia

La **cascada** en las hojas de estilo significa, tal y como hemos visto, que el orden de las reglas importa en CSS, cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el CSS.

El concepto de **herencia**, implica que algunas propiedades CSS heredan por defecto los valores establecidos en el elemento padre, el que los contiene, pero otras no. Cuidado, eso puede dar lugar a errores

LOS TRES PILARES DE CSS

Conocer estos conceptos te hará dominar este lenguaje



Prof. Alvaro Felipe
@alvarofelipe

HERENCIA

Los hijos heredan estilos de sus padres, así que no necesitas escribir estilos para todo.

```
body {  
  color: blue  
}  
  
h1 {  
  color: blue  
  /*No es necesario*/  
}
```

No es necesario definir un color de texto para los títulos pues son heredados de body



ESPECIFICIDAD

Cuando hay conflictos de estilos el navegador aplica solo el de mayor especificidad.

```
.parrafo {  
  color: blue  
}  
  
.article .parrafo {  
  color: red  
}
```

El párrafo es rojo por que el segundo selector es más específico



CASCADA

Todo estilo sobrescribe a uno anterior. La C de CSS significa Cascada

```
.parrafo {  
  color: blue  
}  
  
.parrafo {  
  color: red  
}
```

El párrafo es rojo por que sobrescribe a los anteriores estilos



Aprende CSS desde cero gratis en EDteam:

ed.team/cursos/css



a few CSS selectors

`div`

matches div elements

`<div>`

`#welcome`

matches the id

`<div id="welcome">`

`div .button`

match any `.button` that's a child of a div

`.button`

. matches the class

``

`div.button`

match divs with class "button"

`<div class="button">`

`div > .button`

match any `.button` that's a direct child of a div

`a:hover`

matches a elements that the cursor is hovering over

`ul li:first-child`

match the first item of a list. there's last-child too.

`a[href^="http"]`

match links where the href attribute starts with "http" (external links)

`:checked`

matches if a checkbox or radio button is checked

`tr:nth-child(odd)`

match every other row of a table (make stripes!)

`div:not(#header)`

match all divs except the one with id "header"



CSS3

Unidades de medida

- ❏ Absolutas
- ❏ Relativas
- ❏ Porcentuales

Relativas [\(converter\)](#)

Son las que debemos usar siempre en el desarrollo web.

PX: Es una medida relativa puesto que se irá adaptando a la pantalla y su resolución.

EM: Toma el tamaño de los que le viene por herencia y en base a eso coge la anchura de la **M**, que será la medida que tome como referencia. Escalamos.

REM: No funciona por herencia, se basa en la fuente definida en el root `<html>`, base a eso coge la anchura de la **M**, y escalamos.

EX: Toma el tamaño de los que le viene por herencia y en base a eso coge la altura de la **x**, que será la medida que tome como referencia. Escalamos.

REX: No funciona por herencia, se basa en la fuente definida en el root `<html>`, base a eso coge la anchura de la **x**, y escalamos.



Absolutas



Son medidas que pueden ser específicas de zonas concretas del mundo, y es bueno conocerlas.

Será el navegador el que haga los cálculos para ocupar en pixeles el tamaño especificado.

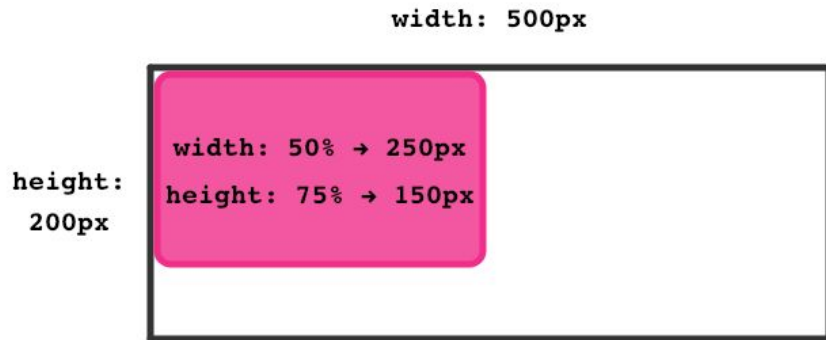
- in (inche o pulgadas): $1\text{in} = 2.54\text{cm}$
- cm: centímetros
- mm: milímetros
- pt (puntos): $1\text{pt} = 0.35\text{mm}$, suele usarse para impresión.
- pc (picas): $1\text{pc} = 12\text{pt} = 4.23\text{mm}$

Porcentuales

Son medidas relativas, que nos va bien para hacer nuestra web responsive.

En realidad, podemos decir que las medidas relativas, son fijas en el mismo dispositivo. Es decir, dentro de un mismo dispositivo 100px, serán siempre lo mismo. Lo que cambia es entre dispositivos.

Poner un **porcentaje %** querrá decir que ocupara ese porcentaje, pero del padre, del elemento directamente anterior.



Tamaño: width y height

- `width = 100 px` → Fijamos el tamaño a un determinado número de píxeles.
- `width = 50%` → indicamos que porcentaje es el que debe ocupar el elemento, respecto al padre, elemento que lo contiene directamente.
- `min_width = 250 px` → Fijamos un tamaño mínimo.
- `max_width = 400 px` → Fijamos un tamaño máximo.

Lo mismo para el `height`.

Si fijamos algún tamaño, solemos fijar uno de los dos tamaños y el otro indicamos un porcentaje.

CSS units

CSS has 2 kinds of units:
absolute & relative

absolute: px, pt, pc,
in, cm, mm

relative: em, rem, ch, ex,
vw, vh, %

rem

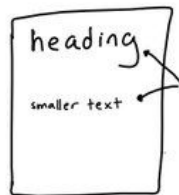
the root element's
font size

```
html {  
  font-size: 14px;  
}
```

this means 1rem = 14px
everywhere in the document

em

the current element's
font size



these 2 elements
have different
values of 1em

0 is the same
in all units

```
.btn {  
  margin: 0;  
}
```

you don't need to say 0px or
0em, 0 is always the same

in, cm, mm

Great for print stylesheets,
not so good for web

vw, vh

100vw is the viewport width
100vh is the viewport height

rem & em help with
accessibility

```
.modal {  
  width: 20rem;  
}
```

this scales nicely if the user
increases their browser's
default font size

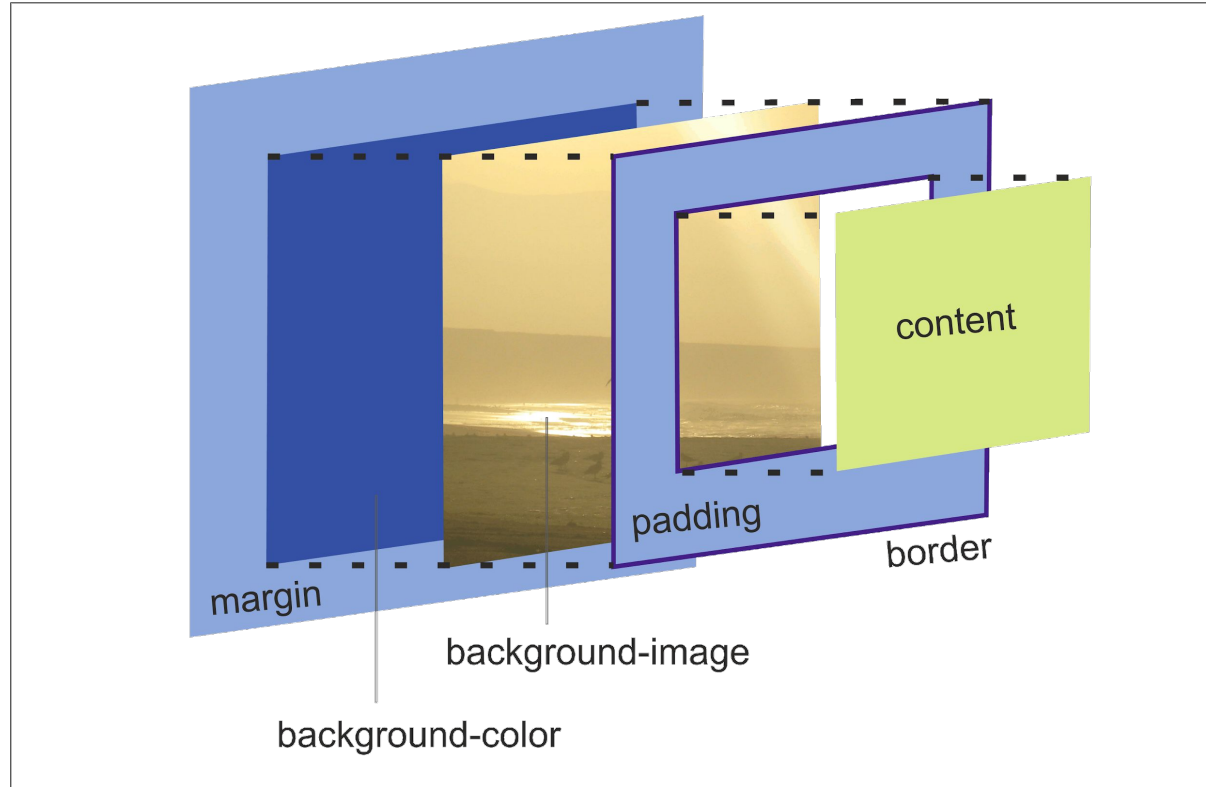


CSS3

Modelo de caja



Propiedades básicas: módulo de caja

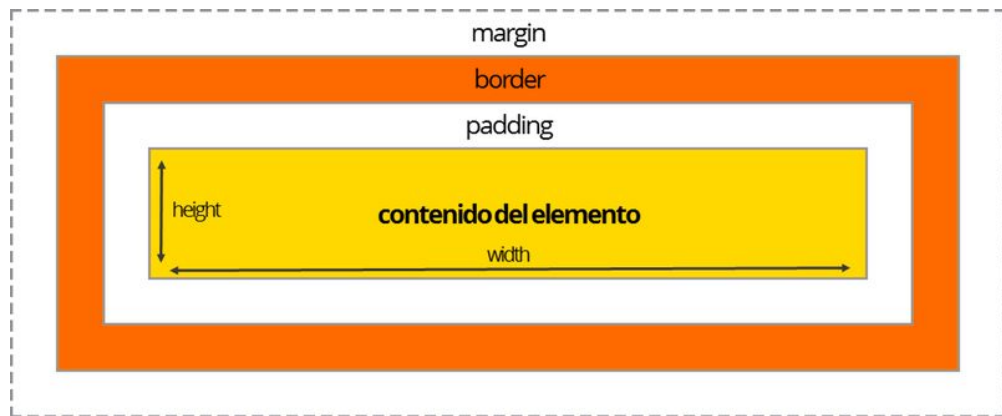


Propiedades básicas: módulo de caja

Los elementos se encuentran en “cajas” con propiedades que son como **las capas de una cebolla** y que dan mucho juego a nivel estético con espacios y bordes. Estas propiedades son **comunes a todos los elementos**.

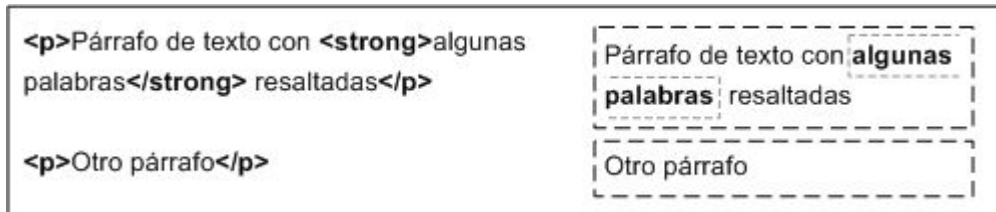
width: establece la anchura del elemento. Pueden ser valores absolutos o relativos al contenedor del elemento.

height: similar a width para la altura del elemento.



Propiedades básicas: módulo de caja

- **padding**: espacio entre el borde del elemento y su contenido. Es un borde interno (*-left, -right, -top o -bottom*).
- **border**: es el borde que rodea a nuestro elemento y podemos elegir entre (*-left, -right, -top o -bottom*).
- **margin**: espacio exterior al borde del elemento que mantiene con los elementos vecinos (*-left, -right, -top o -bottom*).
- **background-color**: color de fondo
- **background-image**: imagen de fondo
- **vertical-align**: alineamiento vertical
- **text-align**: alineamiento del texto



MAY 5, 2009

Soothe nervous clients with smart communication and planning tools. Tap the hidden intelligence of crowds. Feel better soon.

The Wisdom of Community

by [DEREK POWAZEK](#)

The Wisdom of Crowds (WOC) theory does not mean that people are smart in groups — they're not. Anyone who's seen an angry mob knows it. But crowds, presented with the right challenge and the right interface, can be wise. When it works, the crowd is wiser, in fact, than any single participant.

Taking the Guesswork Out of Design

by [DANIEL RITZENTHALER](#)

Clients, like other humans, often fear what they don't understand. Daniel Ritzenthaler explains how sound

AN EVENT APART
coming to a city near you

A BOOK APART
coming soon

EDITOR'S CHOICE

originally ran: May 6, 2008

Community: From Little Things, Big Things Grow

by [GEORGE DATES](#)

Q. What technology do you need to build the next Flickr?
A. Trick question. What you need to build the next Flickr is people. George Oates, a key member of the core team that shaped the Flickr community, shares lessons that can help you grow yours.

Search ALA

 ☐ include discussions

Topics

- [Code](#)
- [Content](#)
- [Culture](#)
- [Design](#)
- [Process](#)
- [User Science](#)

[Ad via The Deck](#)

JOB BOARD

[Kickball is looking for a Frontend engineer for Silicon Valley startup.](#)
[See more on the Job Board.](#)

HOSTED BY

(mt)

PUBLISHED BY

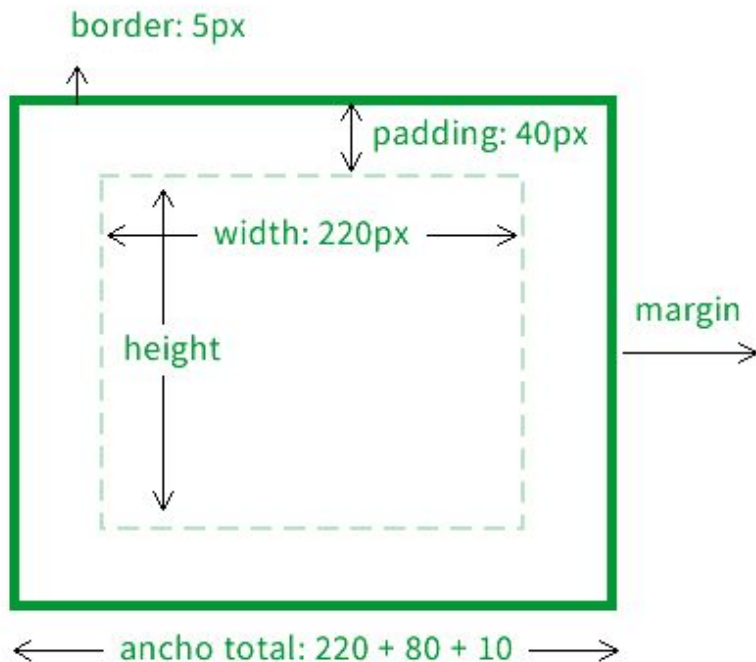
 **happy.co**

Margin y Padding

Margin: Hacemos referencia al borde externo

- `margin: 20px;` → todos iguales
- `margin: 20px 50px;` → TOP y BOTTOM
- `margin: 30px 50px 3% 15%;` →
Siguiendo las agujas del reloj vamos de TOP,
RIGHT, BOTTOM, LEFT
- También podemos especificar uno a uno:
`margin-bottom`, `margin-left`,
`margin-right`, `margin-top`

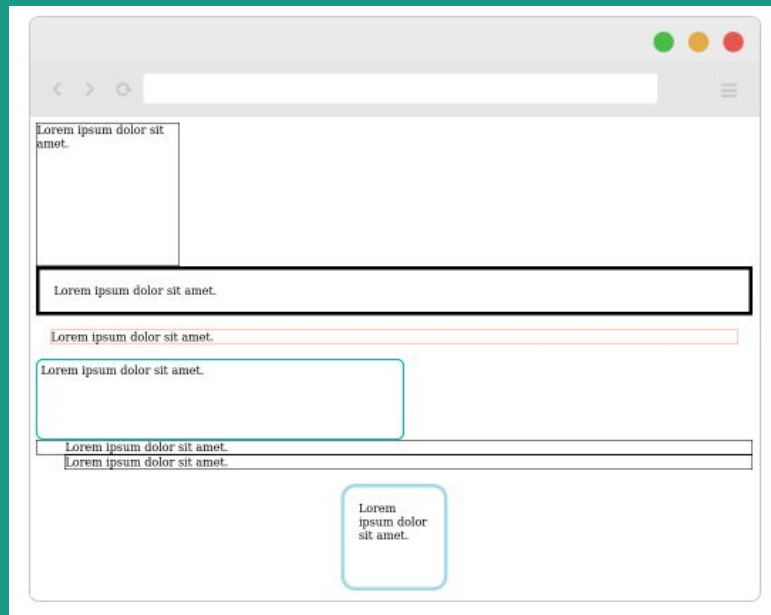
Padding: Hace referencia al borde interno



Ejercicio 3 - Modelo Caja

Teniendo este código HTML, crea un fichero CSS que consiga obtener el resultado de la imagen

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Box Properties</title>
  <link rel="stylesheet" href="css/box_properties.css">
</head>
<body>
<div class="figure1">Lorem ipsum dolor sit amet.</div>
<div class="figure2">Lorem ipsum dolor sit amet.</div>
<div class="figure3">Lorem ipsum dolor sit amet.</div>
<div class="figure4">Lorem ipsum dolor sit amet.</div>
<div class="figure5">Lorem ipsum dolor sit amet.</div>
<div class="figure6">Lorem ipsum dolor sit amet.</div>
<div class="figure7">Lorem ipsum dolor sit amet.</div>
</body>
</html>
```





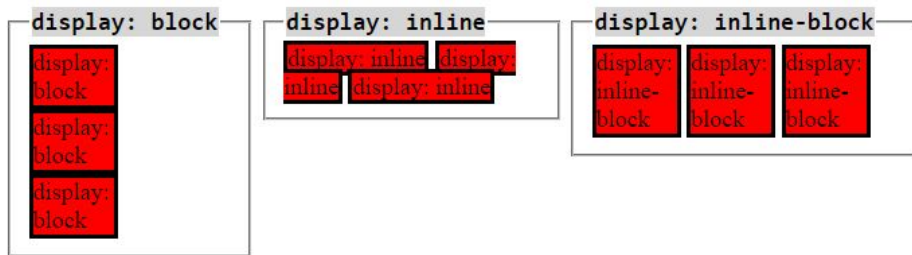
CSS3

Display

Display

La propiedad **display** permite especificar el **tipo de caja** que se aplica en un elemento. Los diferentes tipos de cajas corresponden a diferentes tipos de contenido (texto, listas, tablas)

- **none**: hace que no se genere caja, el elemento desaparece, deja de ser visible y no ocupa ningún espacio.
- **block**: caja rectangular que forma un bloque y hace un salto de línea
- **inline**: caja en línea que puede ocupar varias líneas.
- **inline-block**: caja en línea, pero que se formatea como una caja de tipo block.



Visibilidad de las capas



Visibility es la propiedad que permite no hacer visible un elemento pero ocupa su espacio, es decir que de cara al resto de elementos lo tendrán en cuenta a la hora de posicionarse. Puede adquirir los valores `hidden`, `visible`

Opacity es la propiedad que informándose con valores entre **0** y **1** podremos hacer que sea totalmente sólido o bien translúcido, incluso transparente.

La propiedad de **z-index** podemos decidir que objetos están delante y cuales detras. Representa la posición en el eje $z = -99, \dots, 0, 1, 2, 3, \dots, 99$; cuanto mayor sea más adelante estará.

Overflow

La propiedad de **overflow** sirve en el modelado de cajas para indicar al navegador qué es lo que debe hacer con el contenido que no cabe dentro de una capa, según las dimensiones que se le han asignado.

- **visible** → Muestra todo aunque no quepa.
- **hidden** → No mostrará lo que sobresale, recorta todo lo que no quepa en la ventana.
- **scroll** → Aparece un scroll para poder ver todo el contenido pero solo dentro del espacio de la ventana que recortará el contenido.
- **auto** → Solo si es necesario aparecerá un scroll.

Lorem ipsum dolor amet
next level banh mi actually
etsy craft beer. Portland
meh palo santo pitchfork
wayfarers raclette kinfolk
try-hard YOLO. Lo-fi cred
pork belly, cloud bread
artisan heirloom raw
denim kombucha. Godard
etsy ugh, letterpress roof.

visible

Lorem ipsum dolor amet
next level banh mi actually
etsy craft beer. Portland
meh palo santo pitchfork
wayfarers raclette kinfolk
try-hard YOLO. Lo-fi cred
pork belly, cloud bread
artisan heirloom raw
denim kombucha. Godard
etsy ugh, letterpress roof.

hidden

Lorem ipsum dolor amet
next level banh mi
actually etsy craft beer.
Portland meh palo santo
pitchfork wayfarers
raclette kinfolk try-hard
YOLO. Lo-fi cred pork
belly, cloud bread
artisan heirloom raw

scroll

Lorem ipsum dolor am
next level banh mi
actually etsy craft beer
Portland meh palo santo
pitchfork wayfarers
raclette kinfolk try-har
YOLO. Lo-fi cred pork
belly, cloud bread
artisan heirloom raw
denim kombucha.

overflow

Position (ejemplos)

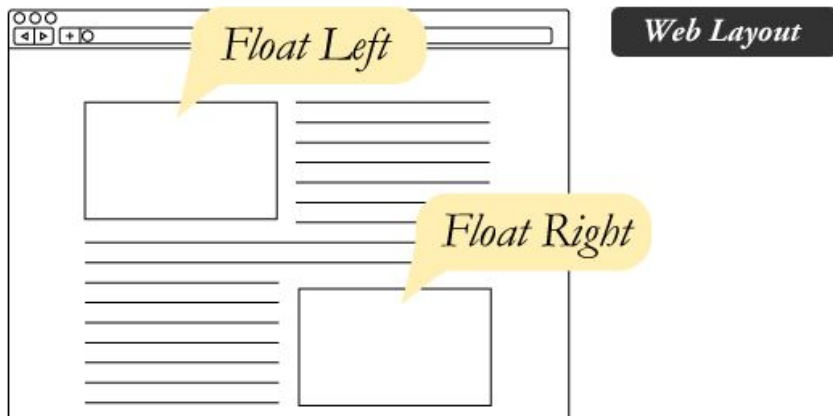
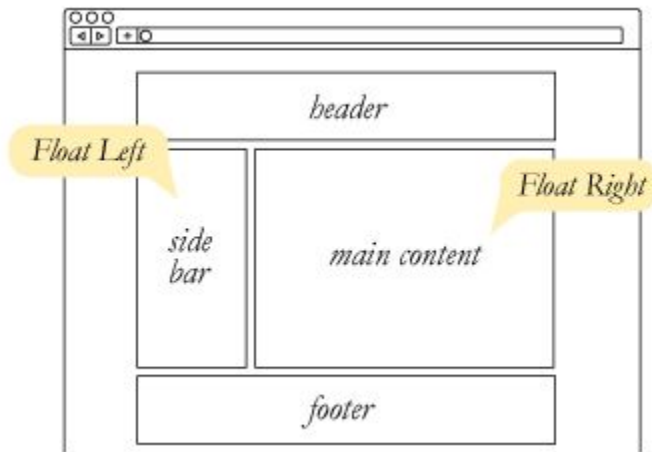
Position es la propiedad que indica en qué lugar de la pantalla queremos posicionar nuestro elemento.

- **static**: No hace caso de las coordenadas, los objetos se van posicionando uno tras otro según su **display**.
- **absolute**: Se colocara donde indican las coordenadas (`top` y `left`), posicionándose en ese punto concreto sin tener en cuenta el resto de elemento de la página .
- **relative**: Indicará que será respecto al padre, respecto al elemento que lo contenga. Indicaremos entonces las coordenadas en la que queremos que esté (puede ser negativo.) teniendo en cuenta cuál sería la posición original que le tocaría por defecto.
- **fixed**: **Siempre** posiciona sus coordenadas respecto al **body** y además se quedará siempre fijado, siempre visible, anclado en la página (ej: Menú)
- **sticky**: Es un **relative** y un **fixed** a la vez, es relativo mientras se visualiza al padre y si scrolamos pasará a ser **fixed** y será siempre visible.

Float, Clear

Float, es la propiedad que le damos al elemento cuando queremos que el resto lo vayan rodeando. Dependiendo de lo que queramos al resto de elemento es aconsejable ponerle `display = inline-block`.

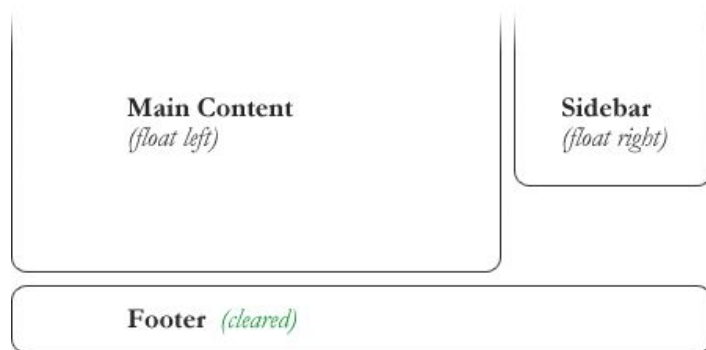
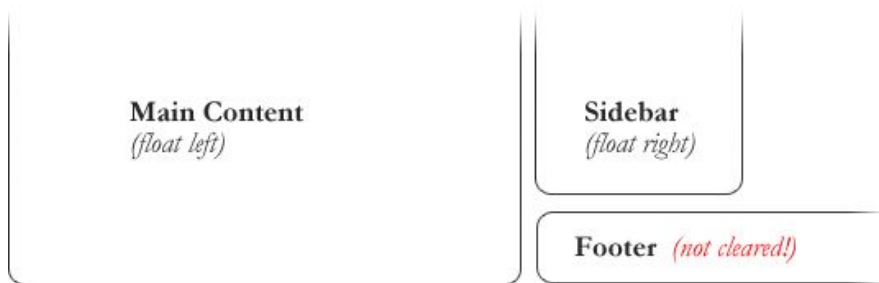
- `float = right/left` → Colocamos el elemento a la izquierda o a la derecha



Float, Clear

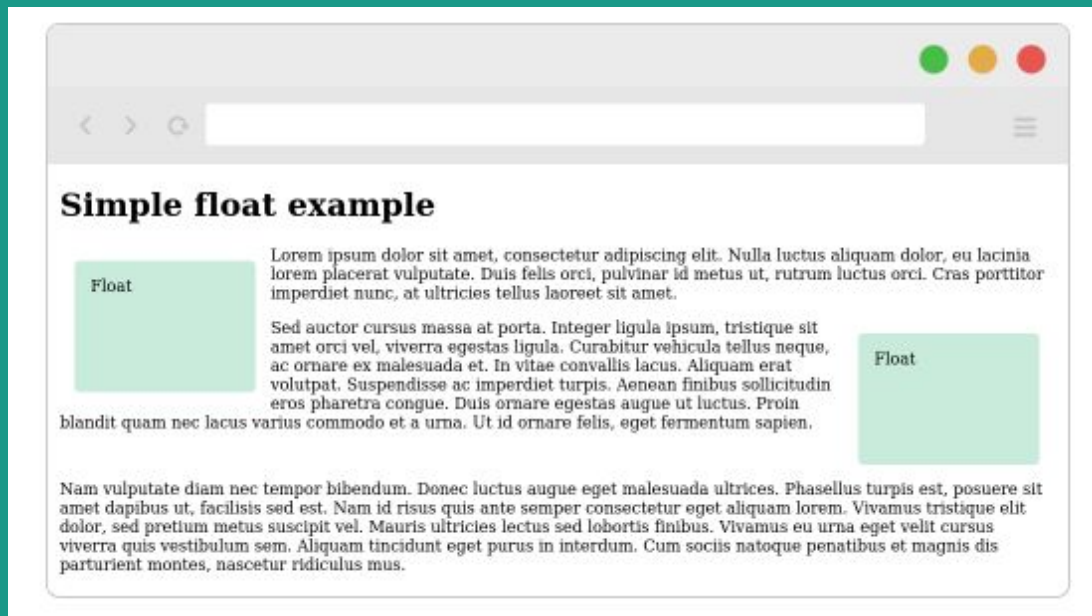
Clear, es la propiedad que resetea el float, deja de hacer caso a la propiedad de Float.

- **clear** = **left**, **both**, **right** → Limpia el float, para que deje de aplicarse.



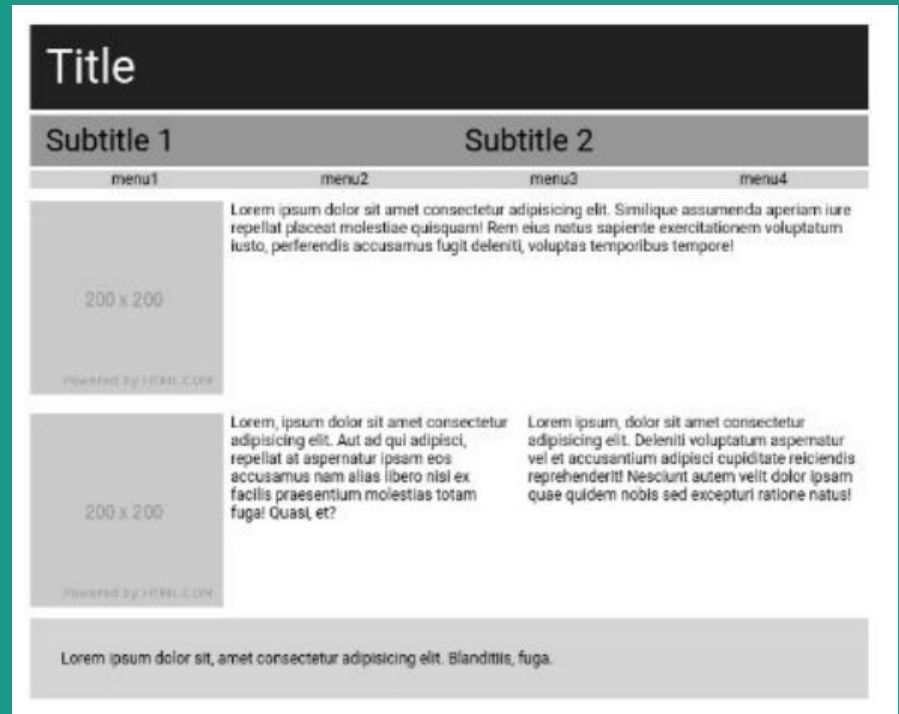
Ejercicio 4 - Float

Genera el fichero HTML y el CSS para acabar obteniendo este resultado:



Ejercicio 5

Con todo lo visto en clase, genera un fichero HTML bien estructurado y el fichero CSS que te permita conseguir este resultado.



Referencias

[Referencia de CSS](#): Web de manual de referencia de CSS

[Librosweb](#): un sitio gratuito fantástico para aprender todo sobre HTML y CSS, y mucho más.

[W3CSchools](#): un sitio de referencia que viene muy bien para consultar puntualmente dudas técnicas muy concretas.

[CSS Tricks](#): aquí encontrarás ejemplos de código CSS muy originales y muy avanzados en algunos casos. Una fuente fantástica para convertirte en un pro si lo deseas.

[Zengarden](#): un sitio clásico con muchos diseños para inspirarte. No está tan orientado a aprender a programar los diseños como CSS Tricks, sino mucho más a inspirarte con el resultado final.