



CSS

Cascada, Modelado de caja y Posicionamiento

Cordero Hernández,
Marco R.



Siguiente >>



En sesiones pasadas...

- CSS básico
 - Selectores básicos
 - Selectores por atributos
 - Pseudo-clases
 - Pseudo-elementos
 - Selectores combinados





En esta sesión

01

Orígenes de
CSS

02

Orden de
Cascada y
colisiones

03

Especificidad

04

Herencia

05

Unidades de
medida

06

Modelo de
caja

07

Propiedad
display

08

Posicionamiento

Siguiente >>



01

ORÍGENES DE CSS





Orígenes de CSS

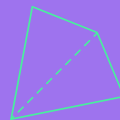
El estilo puede venir de múltiples fuentes predeterminadas o personalizadas.

- Hoja de estilo de *agente de usuario* (**User-agent stylesheets**):
Estilo dado por el navegador por defecto.
- Hoja de estilo de *autor* (**Author stylesheets**):
Aquellas en donde son creadas como parte del diseño del sitio web; El autor define estas hojas usando uno o varios archivos que definen el estilo del sitio web.
- Hoja de estilo de *usuario* (**User stylesheets**):
El usuario/lector del sitio web puede sobrescribir los estilos creando sus propias hojas de estilo diseñadas para mejorar la experiencia de navegación de forma que se adecue a sus necesidades o deseos.



02

ORDEN DE CASCADA Y COLISIONES





Orden de cascada

Recordatorio: CSS = Cascading Style Sheets.

Así como existen las jerarquías de operadores en matemáticas y en lógica, CSS también cuenta con el siguiente orden de aplicación de reglas (presentado de mayor a menor peso):

1. Agente de usuario + !important
2. Usuario + !important
3. Autor + !important
4. Autor
5. Usuario
6. Agente de usuario

... es decir, las reglas predefinidas por el navegador, en combinación de alguna propiedad con valor !important, se aplicará por encima de reglas definidas por el usuario con la misma combinación, etcétera.



Colisiones de estilos

Cuando dos o más reglas apuntan a un mismo elemento, se dice que existe una **colisión de estilo** y es importante conocer cuál de todas las definiciones reemplazará al resto en la renderización final de la página.

Las colisiones pueden ser evitadas cuando se tiene un diseño bien definido desde el inicio y se modularizan las reglas para todos los elementos. Esto puede ser logrado (en conjunto de múltiples archivos de estilo) a través de:

- Origen e importancia
- *Especificidad*
- Orden del código



Colisiones de estilos

Reglas básicas

- Un selector tiene más prioridad entre más específico sea
- Si ambos selectores tiene la misma importancia, se considera la última regla (definida después en orden de arriba hacia abajo)

Prioridad

- Unidades: Selectores de tipo o etiqueta y pseudo-elementos
- Decenas: Selectores de clase, selectores de atributos, y pseudo-classes
- Centenas: Selectores de ID (#)
- Millares: Estilos inline (atributo **style** dentro de etiquetas)
- !important: Sobrescribe cualquier otra (no recomendado)



03

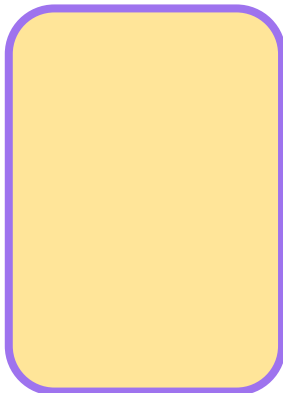
ESPECIFICIDAD



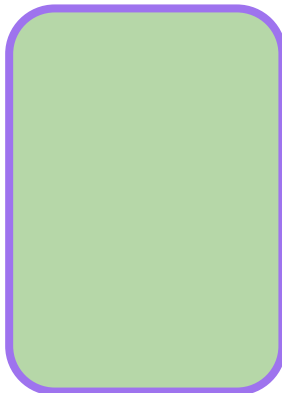


Especificidad

Atributo ***style***



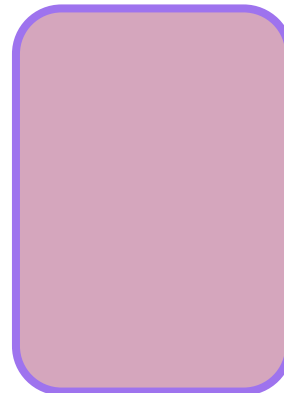
ID



Clase,
Pseudo-clase,
atributo



Elementos



Valor más
específico

Valor menos
específico

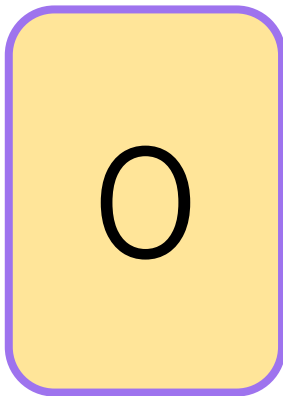
Siguiente >>



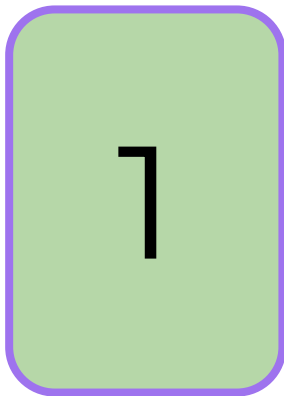
Ejemplos

`ul#nav li.active a`

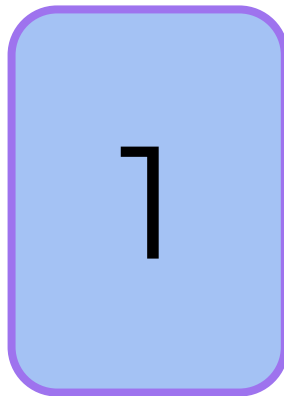
Atributo ***style***



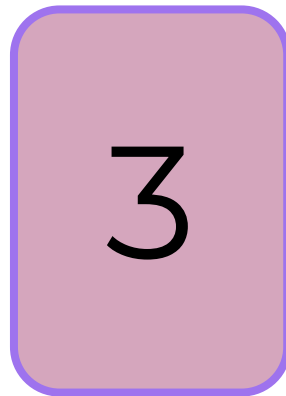
ID



Clase,
Pseudo-clase,
atributo



Elementos

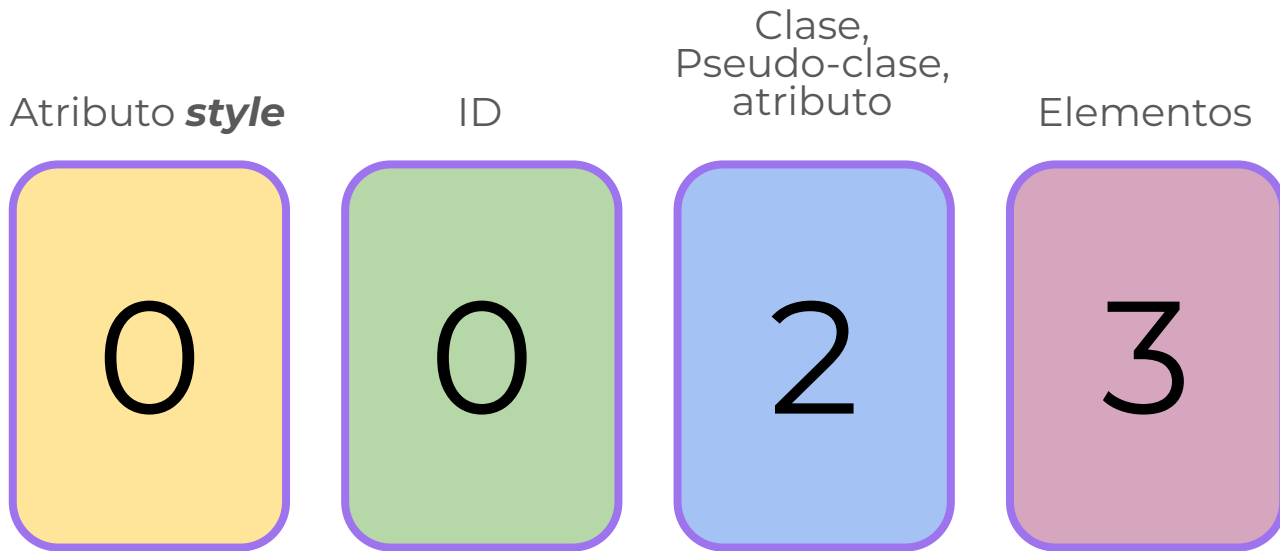


Siguiente >>



Ejemplos

`body.ie7 .col_3 h2 ~ h2`

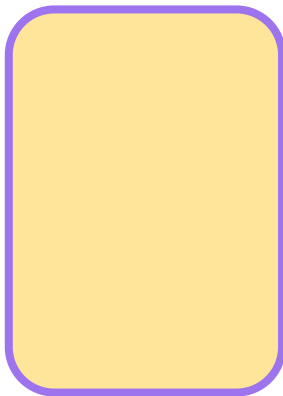




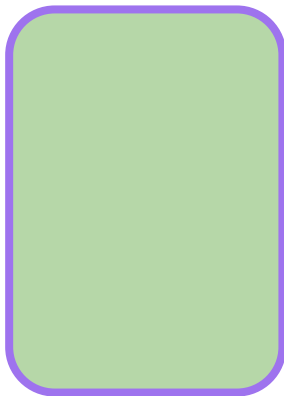
Ejemplos

```
<li style="color: red;">
```

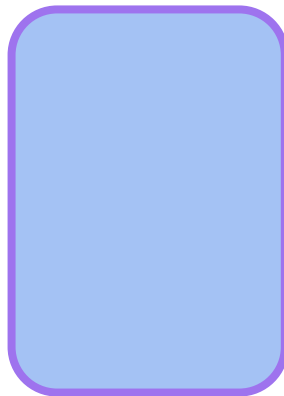
Atributo ***style***



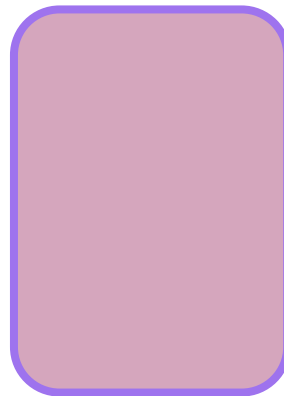
ID



Clase,
Pseudo-clase,
atributo



Elementos

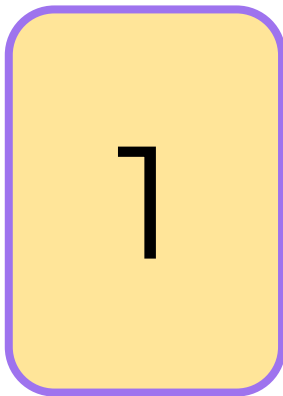




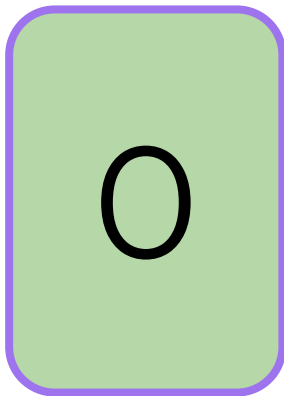
Ejemplos

```
<li style="color: red;">
```

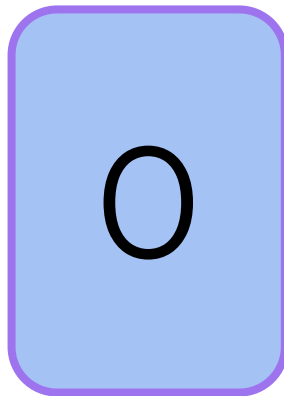
Atributo ***style***



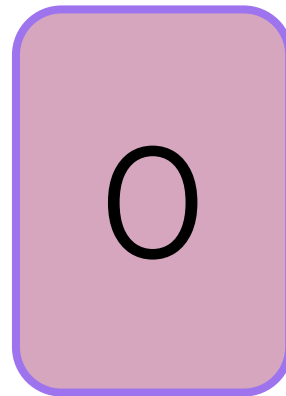
ID



Clase,
Pseudo-clase,
atributo



Elementos



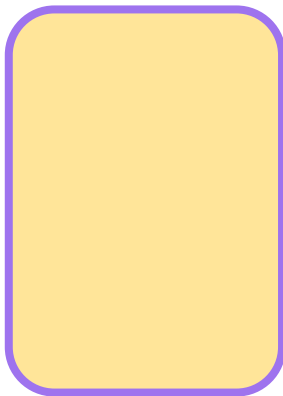
Siguiente >>



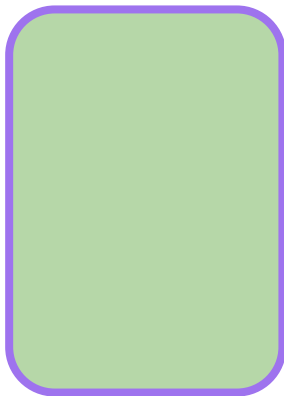
Ejemplos

```
#footer *:not(nav) li
```

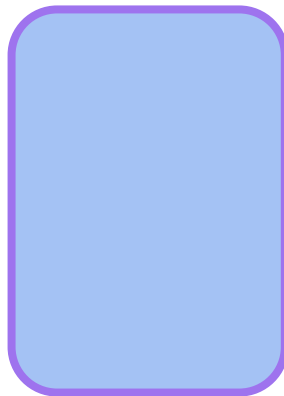
Atributo ***style***



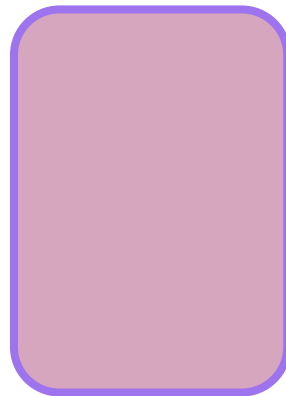
ID



Clase,
Pseudo-clase,
atributo



Elementos

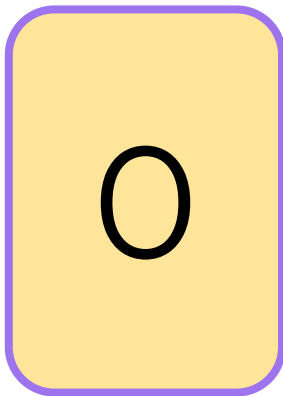




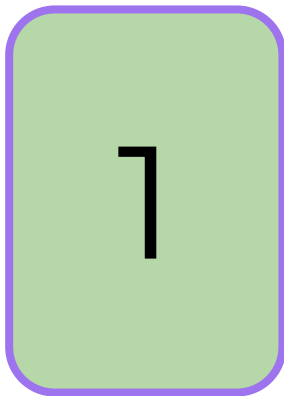
Ejemplos

```
#footer *:not(nav) li
```

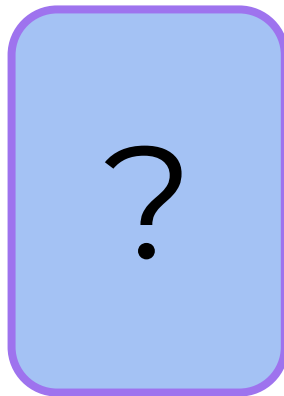
Atributo ***style***



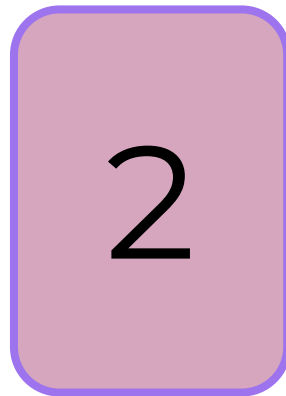
ID



Clase,
Pseudo-clase,
atributo



Elementos



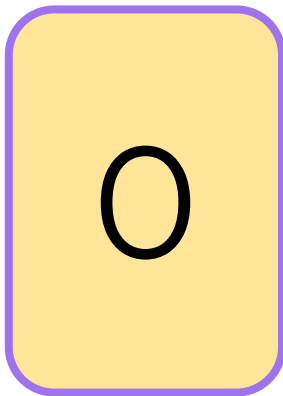
Siguiente >>



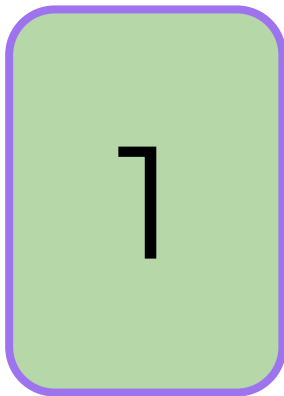
Ejemplos

```
#footer *:not(nav) li
```

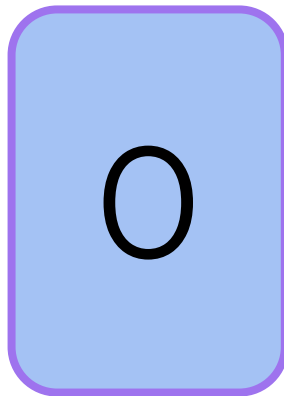
Atributo ***style***



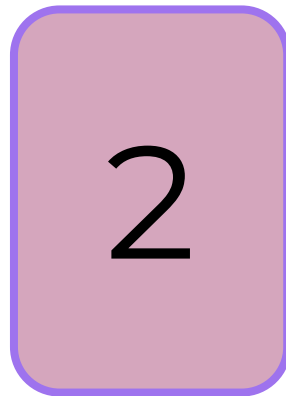
ID



Clase,
Pseudo-clase,
atributo



Elementos



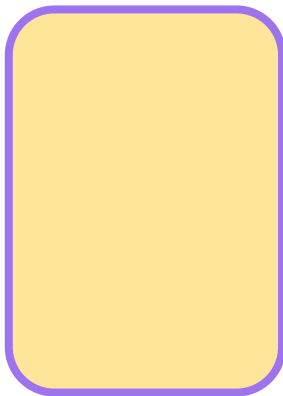
Siguiente >>



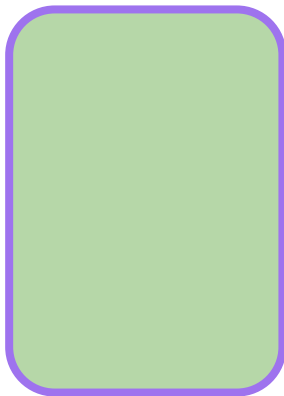
Ejemplos

`ul > li ul li ol li:first-letter`

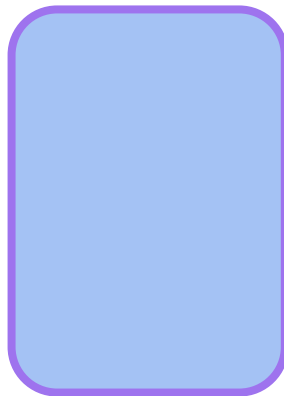
Atributo ***style***



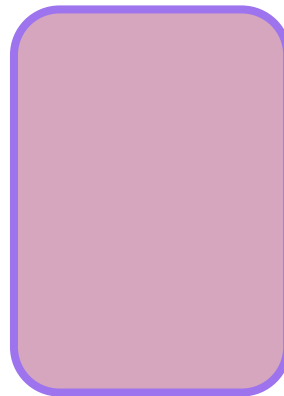
ID



Clase,
Pseudo-clase,
atributo



Elementos

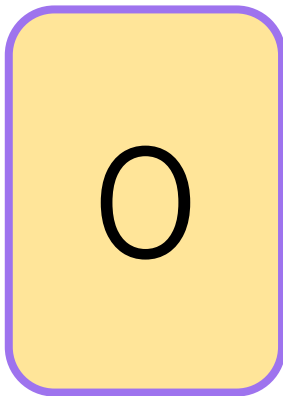




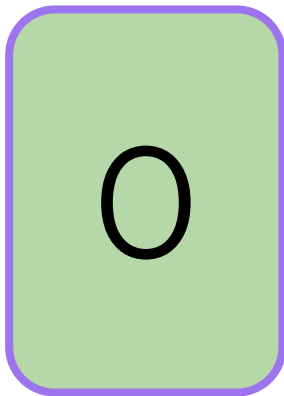
Ejemplos

`ul > li ul li ol li:first-letter`

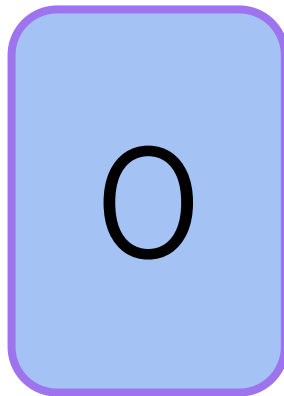
Atributo ***style***



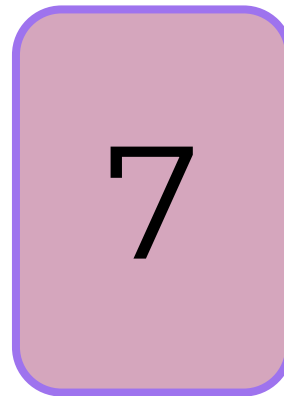
ID



Clase,
Pseudo-clase,
atributo



Elementos



Siguiente >>

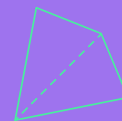


Consideraciones

- El selector universal `*` no tiene especificidad (vale 0)
- Los pseudo-elementos tienen un valor de 1 para elementos
- La pseudo-clase `:not()` no añade especificidad en el tercer campo, solo a lo que está adentro
- *!important* toma precedencia sobre todo lo demás (incluso propiedad *style*)

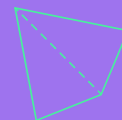
```
<ul id="nav">
  ...
  <li class="active">
    ...
    <a>

Selector Specificity: (1, 1, 3)
ul#nav li.active a {
  display: initial;
}
```



04

HERENCIA

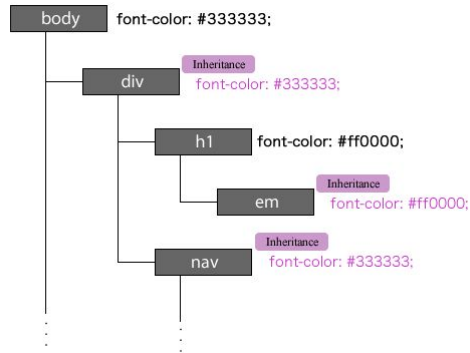




Herencia

Existen algunos atributos que se heredan de los elementos que los contienen (su padre) para evitar la repetición de reglas idénticas, por ejemplo, la fuente o el color del texto.

También, existen otros atributos que no se heredan, por ejemplo: margin, padding, border, background-image, etc. (Revisar referencia para ver todos los atributos heredables)





Herencia

CSS tiene algunos valores especiales para especificar la herencia:

- **inherit:** Asigna el mismo valor en la propiedad que el de su elemento padre
- **initial:** Asigna el del valor por defecto de la propiedad *según cada implementación hecha por el navegador*. En caso de no existir una propiedad predefinida, automáticamente se hereda del padre contenedor. **OJO:** No todas las propiedades están definidas por el navegador.
- **unset:** Toma el valor heredado en caso de que la propiedad lo herede, si no, toma el valor inicial



Ejemplos de herencia

```
<div>  
  A continuación pondremos un comentario:  
  <h1>Comentario: </h1>  
  <textarea name="comment" cols="30" rows="10">Comentario</textarea> <br>  
  <button> botón </button>  
</div>
```

A continuación pondremos un comentario:

Comentario:

A screenshot of a web form. It features a text area with the placeholder text "Comentario" in a small, italicized font. Below the text area is a button labeled "botón". The text area has a small icon in the bottom right corner, likely for clearing the text.

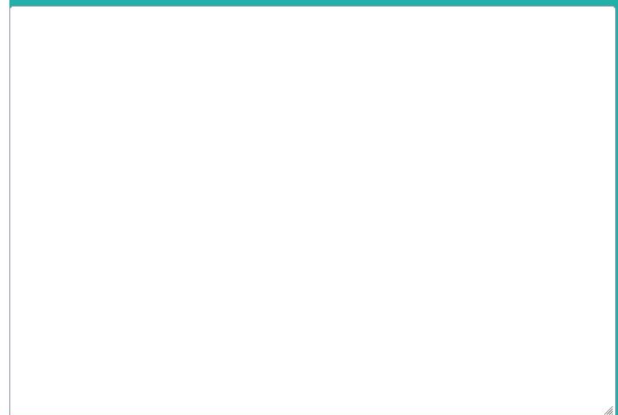


Ejemplos de herencia

```
* {  
    font-weight: normal;  
    font-size: 20px;  
    color: white;  
}  
  
div {  
    background-color: lightseagreen;  
}  
  
/* button {  
    color: initial;  
    background-color: inherit;  
}  
  
textarea {  
    background-color: inherit;  
} */
```

A continuación pondremos un comentario:

Comentario:



botón



Ejemplos de herencia

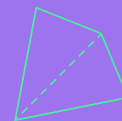
```
* {  
    font-weight: normal;  
    font-size: 20px;  
    color: white;  
}  
  
div {  
    background-color: lightseagreen;  
}  
  
button {  
    color: initial;  
    background-color: inherit;  
}  
  
textarea {  
    background-color: inherit;  
}
```

A continuación pondremos un comentario:

Comentario:

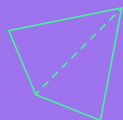
Comentario

botón



05

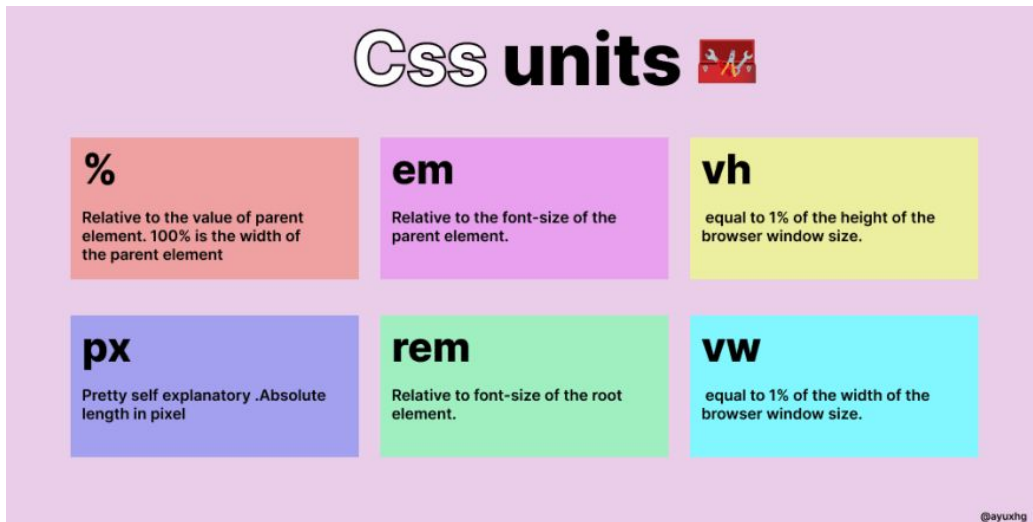
UNIDADES DE MEDIDA





Unidades de medida

CSS también cuenta con múltiples unidades de medida para expresar una longitud, las cuales pueden ser divididas en absolutas y relativas.





Unidades absolutas

Unidad	Descripción
cm	Centímetros
mm	Milímetros
in	Pulgadas (1in = 96px = 2.54cm)
px	Pixeles, unidad por defecto (1px = 1/96 de 1in)
pt	Puntos: Comúnmente usada para altura de fuente (1pt = 1/72 de 1in)
pc	Picas; Usada para longitudes largas (1pc = 12pt)



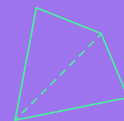
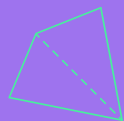
Unidades relativas

Unidad	Descripción
em	Relativo al tamaño de la fuente del elemento ($2em = \text{Tamaño actual de fuente} * 2$)
ex	Relativo a la altura de la x de la fuente actual (raramente usado)
ch	Relativo al ancho del 0 (cero)
rem	Relativo al tamaño de la fuente del elemento raíz
vw	Relativo al 1% del ancho de la ventana del navegador
vh	Relativo al 1% del ancho de la altura del navegador
vmin	Relativo al 1% del menor entre vw o vh
vmax	Relativo al 1% del mayor entre vw o vh
%	Relativo <i>al elemento padre</i>



Consideraciones adicionales

- El valor por defecto del tamaño de la fuente es **16px**
- **16px = 1rem**
 - Si se tienen **16px** → **3REM = 48px**
- 10vw = 10% del ancho de la ventana
 - Si **window.width = 1980px** → **15vw = 297px**
- 32vh = 32% del alto de la ventana
 - Si **window.height = 500px** → **2vh = 10px**



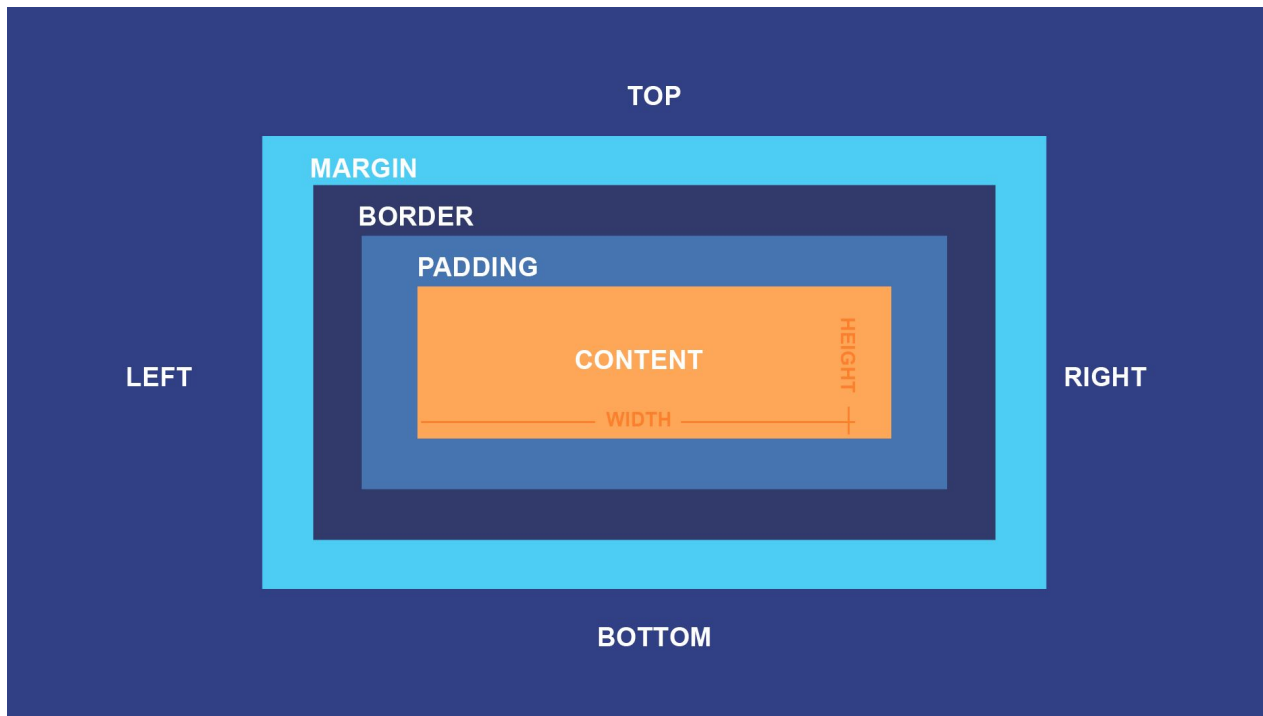
06

MODELO DE CAJA





Modelo de caja (Box model)



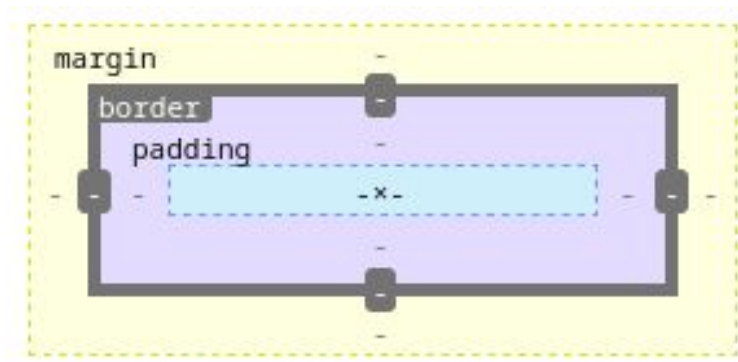


Modelo de caja (Box model)

Todos los elementos de HTML pueden ser mostrados en cajas. Este es un hecho que se considera para el diseño y el layout (acomodo).

Cada caja consiste de:

- Margen (margin)
- Borde/Contorno (border)
- Relleno (padding)
- Contenido (content)





Modelo de caja (Box model)

- **Contenido** (content): Donde aparece el texto e imágenes, definido por **width** y **height**
- **Relleno** (padding **P**): Espacio alrededor del contenido pero antes del borde.
- **Borde/Contorno** (border **B**): Línea fuera del relleno que delimita los contenidos y su espaciado
- **Margen** (margin **M**): Espacio fuera del elemento que puede ser usado para generar distancia entre otros elementos



Visualización de caja

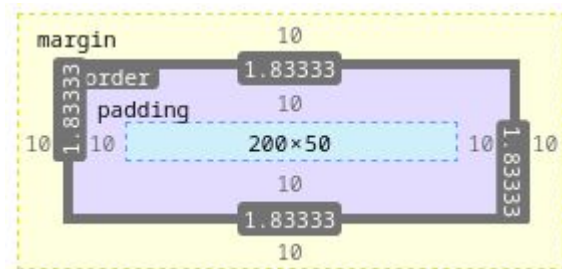
```
div {  
  width: 200px;  
  height: 50px;  
  border: 2px solid navy;  
  padding: 10px;  
  margin: 10px;  
}
```



$$Tw = w + P_l + P_r + B_l + B_r + M_l + M_r$$
$$Th = h + P_t + P_b + B_t + B_b + M_t + M_b$$

$$Tw = 200px + 10px + 10px + 2px + 2px + 10px + 10px = 244px$$

$$Th = 50px + 10px + 10px + 2px + 2px + 10px + 10px = 94px$$





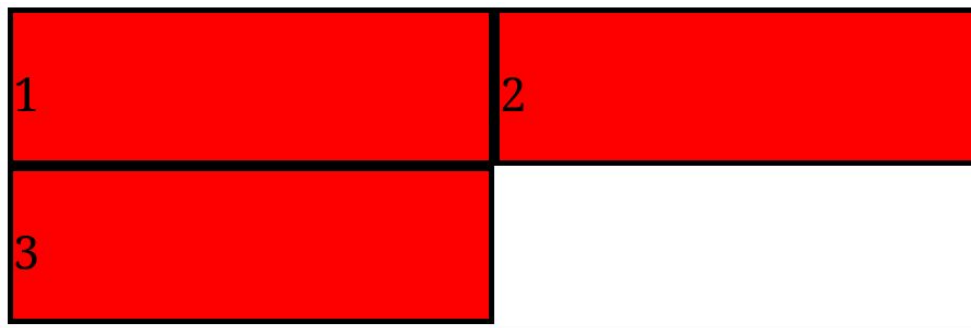
Propiedades adicionales

- **[min,max]-[width, height]:** Otorga medidas máximas o mínimas (útil cuando se quiera tener responsividad y diseños fijos al mismo tiempo)
- **box-sizing:** Define el comportamiento del ancho y alto del elemento
 - *content-box*: Comportamiento por default; Incluye contenido, borde y margen en el cálculo total
 - *border-box*: Las medidas totales serán igual a las del contenido sin considerar lo externo



Caso de estudio: elementos lado a lado

Si se desea que el width ignore las medidas externas, utilizar **box-sizing: border box** ($T_w = w$)





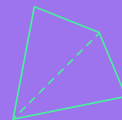
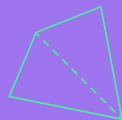
Consideraciones adicionales

- La propiedad ***border*** puede ser usada para “debuggear” el diseño que estemos desarrollando

Contenido de la caja



No sé qué estoy haciendo :(



07

PROPIEDAD DISPLAY





Propiedades *display*

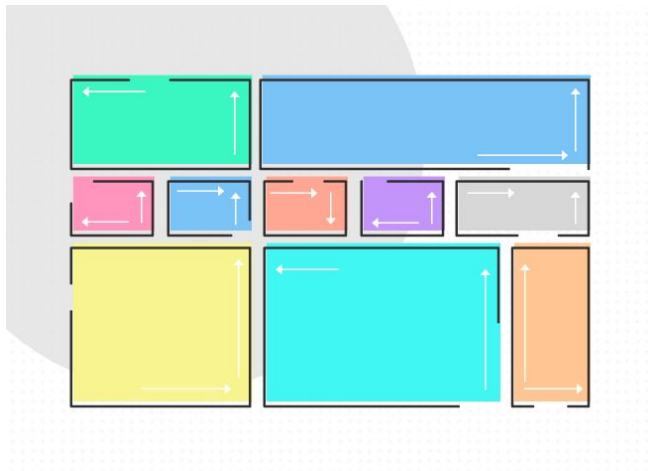
Esta propiedad puede tomar múltiples valores, sin embargo, se presentan 5 relevantes:

1. **none**: El elemento se “remueve” de la vista (afecta el contenido alrededor)
2. **block**: Transforma un elemento inline en uno de bloque (ejemplo: una nueva línea después de una etiqueta ``)
3. **inline**: Transforma un elemento de bloque en uno inline (ejemplo: remover medidas de un `<div>`)



Propiedades *display*

- 4. **inline-block**: Muestra el elemento como si fuera *inline*, pero agregando la posibilidad de modificar su ancho y alto (ejemplo: dar medidas a una etiqueta <p>)
- 5. **flex**: Uso de flexbox (temas siguientes)



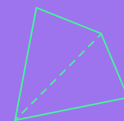
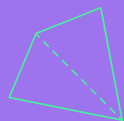


Propiedades *display*

```
.block {  
  display: block;  
}  
  
.inline {  
  display: inline;  
}  
  
.inline-block {  
  display: inline-block;  
}
```

The diagram illustrates three different CSS display properties:

- Block:** A large container box containing three smaller boxes stacked vertically, labeled 'a', 'b', and 'c'.
- Inline:** A horizontal row of four boxes. The first three are small and labeled 'a', 'b', and 'c'. The fourth is a long, empty box.
- Inline-block:** A container box containing three smaller boxes. Two are on the top row (labeled 'a' and 'b') and one is on the bottom row (labeled 'c').



08

POSICIONAMIENTO





Posicionamiento

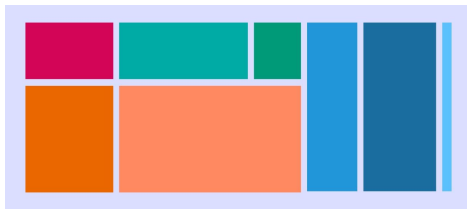
La propiedad **position** puede adquirir alguno de los siguientes valores:

- **static**: Predeterminado, no se ve ningún efecto especial
- **relative**: La esquina superior izquierda del elemento es el origen a partir del cual se puede alterar el objeto
- **fixed**: El elemento se posiciona de manera relativa a la ventana del navegador, es decir, el elemento siempre se encontrará en la misma posición aunque se haga un desplazamiento en cualquier dirección



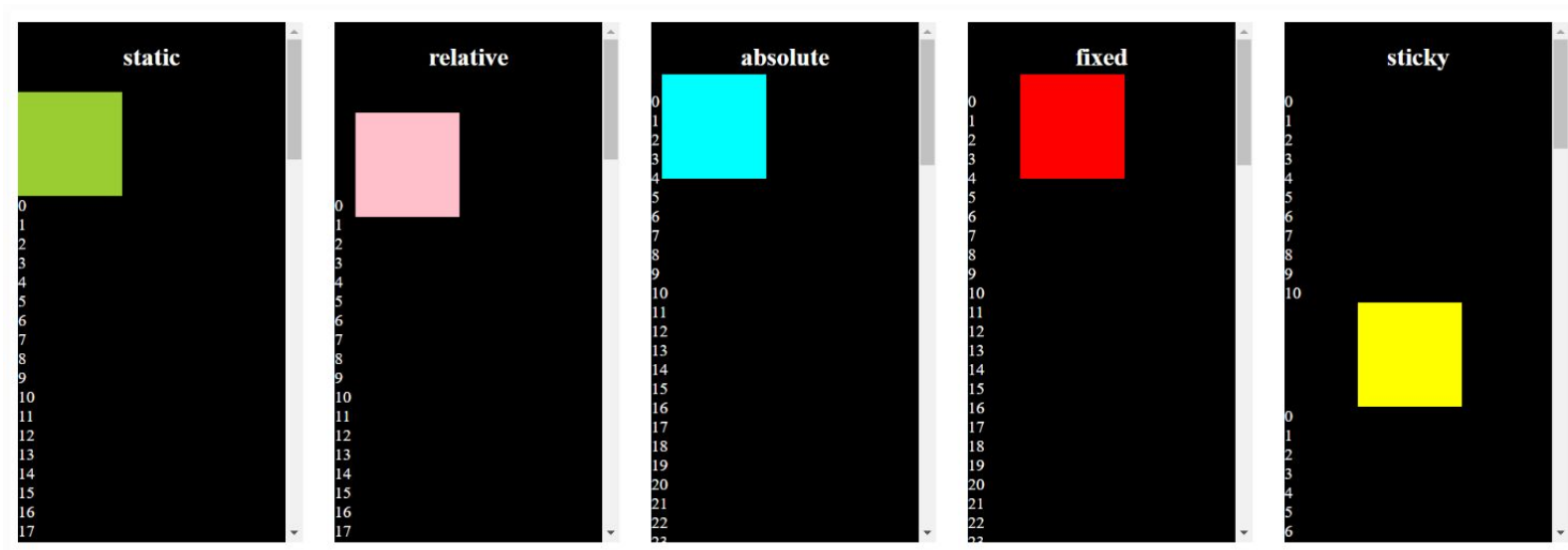
Posicionamiento

- **absolute:** El elemento se posiciona de manera relativa al contenedor con posición relativa más próximo (si no existe, se asume que es <body>)
- **sticky:** El elemento se comporta como si estuviera posicionado de manera relativa hasta que se llega a una determinada posición al realizar un desplazamiento, después, se queda fijo en esa posición





Demostración





Propiedad *float*

Esta propiedad se usa para posicionar y dar formato al contenido. Puede tomar los siguientes valores:

- **left:** los elementos flotan a la izquierda
- **right:** los elementos flotan a la derecha
- **none:** los elementos no flotan (por defecto)
- **inherit:** hereda la propiedad del contenedor padre



imagen

Este texto puede estar a la derecha de la imagen ya que la imagen está flotando a la izquierda.



Propiedad *clear*

Esta propiedad especifica qué elementos pueden flotar a un lado del elemento “limpiado” y en qué lado. Puede tomar alguno de estos valores:

- **none:** permite elementos flotantes en ambos lados (predeterminado)
- **left:** no permite elementos flotantes a la izquierda
- **right:** no permite elementos flotantes a la derecha
- **both:** no permite elementos flotantes a ningún lado
- **inherit:** hereda la propiedad del contenedor padre



El momento que todos esperaban

Ejercicio de posición

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolore ipsum blanditiis nisi odit possimus odio, modi alias officia deleniti corrupti consectetur nostrum qui assumenda animi nemo quaerat? Qui nemo repudiandae praesentium enim temporibus dolor inventore eaque! Nesciunt, quis? Quos, eos?



Lorem ipsum, dolor sit amet consectetur adipisicing elit. Laudantium nam rerum accusantium preferendis quia labore asperiores corporis odit quibusdam, ad reiciendis voluptatum! Atque quisquam adipisci repellendus libero, repellat totam temporibus facilis magnam, qui inventore veniam asperiores id accusamus numquam eius non blanditiis fuga! Quaerat distinctio vitae harum. Eligendi quasi quis dignissimos? Aut eaque eos numquam unde, mollitia provident, sapiente hic in architecto expedita ea enim libero cumque vel? Debitis, aperiam tenetur beatae earum voluptates sed quasi cum omnis dolore voluptatum fuga repudiandae delectus at repellendus ipsa asperiores maiores, maxime obcaecati? Sint et veritatis quae, impedit suscipit sed, optio vel in, totam saepe perspicatis est. Error ducimus dignissimos aperiam, facilis eius eum? Quo provident repellendus autem odio hic obcaecati atque omnis blanditiis impedit! Temporibus numquam beatae fugit veniam voluptatem. Qui incidunt amet molestiae. Totam earum autem impedit sunt, nesciunt mollitia blanditiis qui optio itaque necessitatibus numquam illum rerum? Vitae veritatis obcaecati, odio libero perspicatis maiores soluta deleniti autem reprehenderit cumque porro, minima sequi? Perspicatis voluptate quod quas amet nostrum alias, sapiente distinctio aut obcaecati non, at preferendis fuga nobis voluptates possimus ad nihil exercitationem, itaque reprehenderit similique odio? Nihil, enim asperiores aliquam hic accusamus maxime aliquid assumenda eum vero, fuga voluptatibus? Officiis, vitae necessitatibus? Magnam ea eum itaque vel, dolor doloribus, aliquam, eligendi architecto reiciendis libero pariatur. Iure, necessitatibus voluptatum modi asperiores iusto quia corporis deserunt debitis eveniet doloribus. Modi ad saepe ea repellat? Saepe voluptatum expedita, odit veritatis dolorum dignissimos preferendis nostrum ipsa quisquam consequatur. Maxime tempore labore reiciendis? Mollitia voluptas quidem illo iure praesentium. Esse doloremque atque amet voluptatum in rem libero, quo facilis veniam. Cupiditate sit quod qui, magni laborum blanditiis vel quam, officia, hic deleniti facilis corporis repudiandae? Illum aut dolores voluptas earum ratione ducimus cum vel cumque quidem magni, quod vero quam consectetur. Inventore vitae amet fugit et soluta non dolore quasi necessitatibus aperiam nisi beatae numquam ratione provident nam sed sit quo autem eos, itaque, aliquam labore animi preferendis quidem! Odio odit molestias labore deserunt non qui dicta aliquid assumenda aperiam cumque eius itaque, repellat debitis incidunt dignissimos, dolores minus iste consequuntur. Ut laborum optio ab sunt odit id veritatis preferendis natus enim iusto? Iure, alias illum. Ut tenetur et nostrum quisquam, cumque consequuntur illo molestiae reprehenderit dolorem impedit dolor asperiores ea numquam soluta explicabo laudantium labore fugiat quo ullam? Voluptatibus corporis assumenda rerum molestiae possimus nihil culpa animi expedita dolore iure quo, illum provident eum, asperiores qui magnam reprehenderit in id totam repellat distinctio fugit aut et itaque. Culpa quasi doloremque sapiente consequuntur dolore dolor vitae commodi tempora atque, dolorum esse aspernatur delectus architecto beatae error inventore asperiores porro est quia alias? Expedita ad suscipit, accusantium nam repellat atque. Tenetur ipsum odit optio inventore quo asperiores voluptates, praesentium saepe voluptate distinctio, facilis magni iusto beatae aut, error suscipit nesciunt. Quia placeat reiciendis obcaecati quasi eveniet animi iusto culpa quibusdam modi, sequi tenetur, saepe fuga non dolor officiis eos sit doloribus maxime a ipsum laboriosam accusantium debitis alias vitae! Neque molestias unde inventore obcaecati velit id? Illo provident beatae quo, iste cumque assumenda et delectus porro? Eveniet maiores officia iure, ipsum nemo reiciendis ex voluptas qui beatae sequi itaque dignissimos odit enim laborum vitae harum totam eum. Illum hic a corrupti



Esta página almacena cookies. Lorem ipsum dolor sit amet consectetur adipisicing elit. Nihil ea, rem quaerat maxime numquam quidem. Nemo in error harum aliquid?

Aceptar