



CSS

Hojas de estilo en cascada

<https://developer.mozilla.org/es/docs/Web/CSS>



Juan Vladimir
@juanvladimir13

Agenda

1. CSS
2. Unidades
3. Regla CSS
4. Formas de agregar CSS a HTML
5. Selectores
6. Selectores de combinación
7. Propiedades personalizadas
8. Especificidad
9. Propiedad box-sizing, ...

CSS

Hojas de Estilo en Cascada (del inglés **Cascading Style Sheets**) o **CSS** es el lenguaje de estilos utilizado para *describir la presentación de documentos HTML*.

Es el código que usas para dar estilo a tu página web

Describe como debe ser **renderizado el elemento estructurado en la pantalla**, en papel, en el habla o en otros medios.

Posee una especificación estandarizada por parte del **W3C**



CSS all about boxes

Los **elementos HTML** son como **cajas** apiladas una sobre la otra

Propiedades

padding: Espacio alrededor del contenido

border: Línea que se encuentra fuera del padding

margin: Espacio fuera del elemento que lo separa de los demás

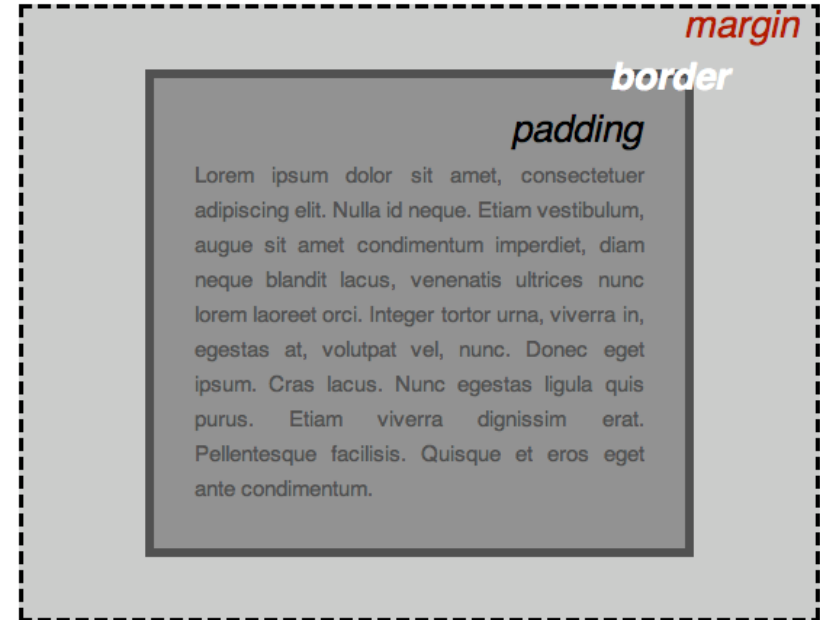
width: ancho del elemento

background-color: Color de fondo del contenido y del padding

color: Color del contenido del elemento

text-shadow: Sombra difuminada en el texto dentro del elemento

display: Modo de visualización para el elemento



Unidades

Longitud absoluta

pt Puntos

px Píxeles

Longitud relativa al tamaño del texto

em : Tamaño calculado en base de su elemento padre.

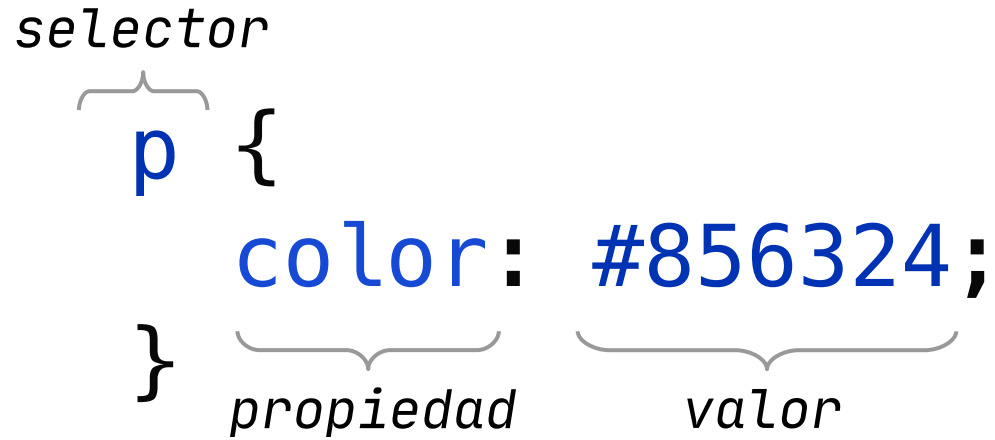
rem : Tamaño calculado en base de su elemento raíz.

Longitud relativa al tamaño de la ventana

vw : % del ancho de la ventana gráfica

vh : % de la altura de la ventana gráfica

Regla CSS

selector
`p {
 color: #856324;
}`

Selector: Selecciona el(los) elemento(s) a dar estilo

Formas de agregar CSS a HTML

```
<html lang="es">
  <head>
    <style>
      p {
        color: #856324;
      }
    </style>
    <title>Tecnico medio</title>
    <link rel="stylesheet" href="base.css">
  </head>
  <body>
    <p style="color: green;">Hola mundo</p>
  </body>
</html>
```

Dentro la etiqueta **<head>** se puede utilizar de las siguiente formas:

- Utilizando la etiqueta **<style>**
- Utilizando la etiqueta **<link>** como un **archivo** local o externo

Como **atributo** en un **elemento HTML**

Selectores

global

```
* {  
}
```

type

```
<h1>Curso</h1>  
  
etiquetaHTML {  
}
```

class

```
<p class="title"></p>  
  
.nombre {  
}
```

id

```
<p id="comment"></p>  
  
#nombre {  
}
```

attribute

```
elementoHTML [attribute] {}  
elementoHTML [attribute="value"] {}
```

https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectors

https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectors#tipos_de_selectores

https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectors/Type_Class_and_ID_Selectors

https://developer.mozilla.org/es/docs/Web/CSS/Attribute_selectors

https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectors/Attribute_selectors

Selectores

Pseudoclase

`elementoHTML: hover {}`

Estado especial del **elemento HTML** seleccionado

- ✓ En relación a factores externos
- ✓ Según estado de su contenido
- ✓ Según posición del ratón

Pseudoelemento

`elementoHTML::first-line {}`

No describen un estado especial,
permiten añadir estilos a una parte concreta del documento HTML

<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>

<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-elements>

https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectors/Pseudo-classes_and_pseudo-elements

Selectores de combinación

Descendientes

`selector1 selector2 {}`

Elementos hijos

`selector1 > selector2 {}`

Adyacente

`elemento1 + elemento2 {}`

Hermanos generales

`elemento1 ~ elemento2 {}`

Propiedades personalizadas

Global

```
/* declaracion */
:root {
  --primary-color: green;
  --fs-18: 18px;
}

/* uso */
p {
  color: var(--primary-color);
  font-size: var(--fs-18);
}
```

Local

```
/* declaracion */
.container {
  --title-color: green;
}

/* uso */
.item {
  color: var(--title-color);
}
```

Especificidad

Es la manera que los navegadores deciden **qué valores de una propiedad CSS son más relevantes** para un elemento y, por lo tanto, serán aplicados.

Cuando varias declaraciones **tienen igual especificidad**, se aplicará al elemento la última declaración encontrada

Valor	Selector
0	Universal (*)
1	Elemento tipo, pseudoelemento
10	Clase, pseudoclase, atributo
100	ID
1000	Atributo style en etiqueta HTML
10000	!important

Propiedad box-sizing

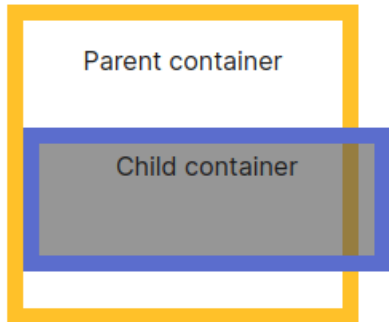
Calcular el **width** y **height** total de un **elemento HTML**

content-box:

width = width of the content

height = height of the content

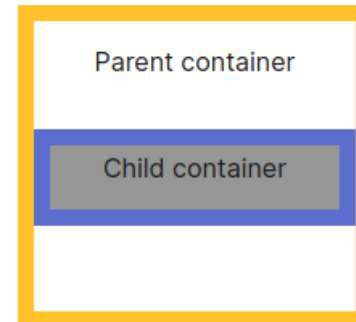
border y **padding** no son incluidos en el cálculo.



border-box

width = border + padding + width of the content

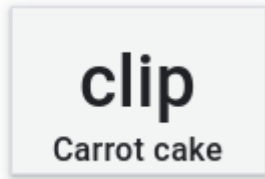
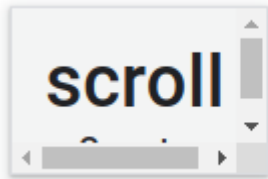
height = border + padding + height of the content



Propiedad overflow

Especifica si:

- ✓ Recortar contenido
- ✓ Dibujar barras de desplazamiento
- ✓ Mostrar el contenido excedente en un elemento a nivel de bloque



visible: El contenido no es recortado, podría ser dibujado fuera de la caja contenedora.

hidden: El contenido es recortado y no se muestran barras de posición.

scroll: El contenido es recortado y usa las barras de desplazamiento.

auto: Depende del navegador, provee desplazamiento si hay contenido excedente.

<https://developer.mozilla.org/en-US/docs/Web/CSS/overflow>

<https://developer.mozilla.org/es/docs/Web/CSS/overflow>

Propiedad text-wrap

Controla cómo se ajusta el texto dentro de un elemento

wrap: Se ajusta entre líneas en los caracteres apropiados para minimizar el desbordamiento

nowrap: Desbordará su elemento contenedor en lugar de saltar a una nueva línea

balance, pretty, stable: Complejos de alto costo computacional o debe ser utilizado en casos concretos.

Propiedad text-overflow

Determina como el contenido que se desborda y que no es mostrado

Puede ser cortado, mostrar una elipsis '...' o mostrar una cadena de texto personalizada

visible
Carrot cake
dessert
jujubes

ellip...
Carrot cak...
Ice cream ...

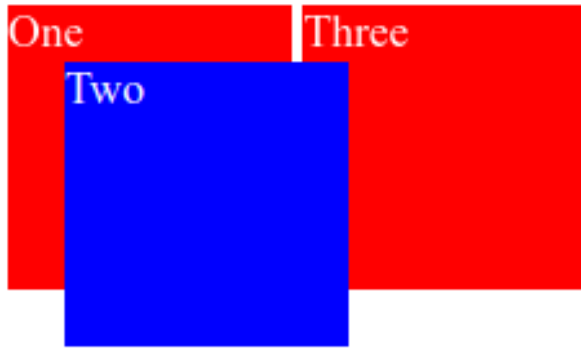
clip: Valor predeterminado, truncará el texto **en el límite del área de contenido**

ellipsis: Mostrará puntos suspensivos para representar texto recortado

```
p {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```


Position

Posición del elemento es en el documento, las propiedades **top**, **right**, **bottom**, y **left** determinan la ubicación final



static: Posición de acuerdo al flujo normal del documento.

relative: Posición de acuerdo al flujo normal del documento, luego desplazado con relación a sí mismo.

absolute: Posición relativo a su ancestro más cercano.

fixed: Posición con relación al bloque contenedor inicial establecido por el viewport

sticky: Es desplazado con relación a su ancestro que se desplace más cercano y su bloque contenedor

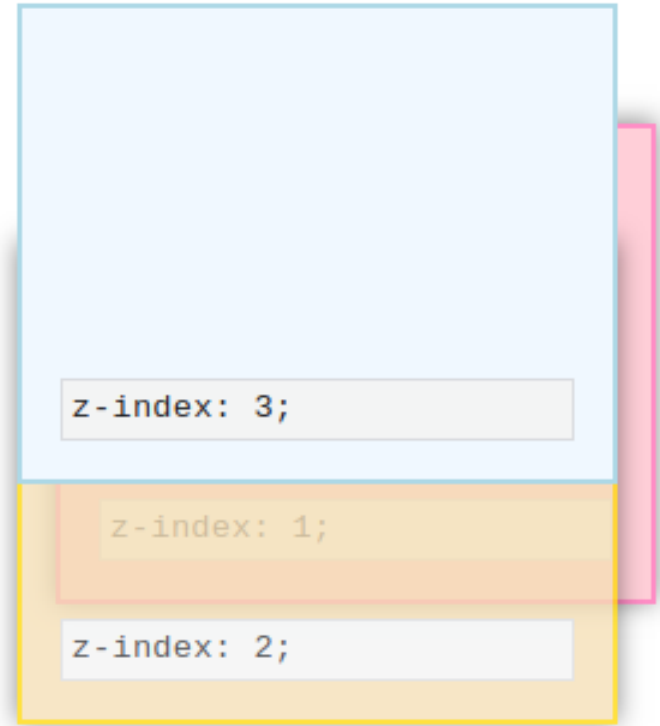
Propiedad z-index

Establece explícitamente un **orden de capas para HTML** basado en el espacio 3D del navegador.

Acepta un valor numérico que puede ser positivo o negativo.

Los elementos aparecerán encima de otro si tienen un valor de **z-index** más alto.

Si z-index no funciona, debes establecer el valor **position** del elemento en un valor distinto de **static**



Herencia

inherit

Toma el valor calculado de la propiedad de su **elemento padre**

initial

Fuerza la utilización del **valor inicial de la propiedad** para el elemento al cual se aplica

unset

Restablece una propiedad a su valor heredado si la propiedad hereda naturalmente de su padre, y a su valor inicial en caso contrario.

Color

Valor hexadecimal

hashtag (#) seguido de seis cifras hexadecimales

```
p { color: #856324; }
```

Valores RGB

Es una función **rgb()** recibe tres parámetros que representan los canales **rojo**, **verde** y **azul**, representado por números decimales entre el 0 y el 255

```
p { color: rgb(2 121 139); }
```

Valores RGBA

Funcionan exactamente de la misma manera que los colores RGB, **hay un cuarto valor** que representa el canal alfa del color, que controla la opacidad.

```
p {  
  color: rgb(2 121 139 / .3);  
}
```