

# UD 2

# Interfaces web utilizando estilos CSS

DISEÑO DE INTERFACES WEB

**Técnico de Grado Superior Desarrollo de Aplicaciones Web**

**2024-25**

# Resultados de aprendizaje

## 2. Crea interfaces Web homogéneas definiendo y aplicando estilos.

1. Se han reconocido las posibilidades de modificar las etiquetas HTML.
2. Se han definido estilos de forma directa.
3. Se han definido y asociado estilos globales en hojas externas.
4. Se han definido hojas de estilos alternativas.
5. Se han redefinido estilos.
6. Se han identificado las distintas propiedades de cada elemento.
7. Se han creado clases de estilos.
8. Se han utilizado herramientas de validación de hojas de estilos.
9. Se ha utilizado y actualizado la guía de estilo.

# Contenidos

- Introducción.
- Selectores.
- Unidades de medida y colores.
- Modelo de cajas (box model).
- Posicionamiento y visualización.
- Texto.
- Enlaces.
- Imágenes.
- Listas.
- Tablas.
- Formularios.
- Layout.

# Introducción

CSS no es realmente un lenguaje de programación, tampoco es un lenguaje de marcado.

Es un *lenguaje de hojas de estilo*, es decir, te permite aplicar estilos de manera selectiva a elementos en documentos HTML.

# Introducción

style.css

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

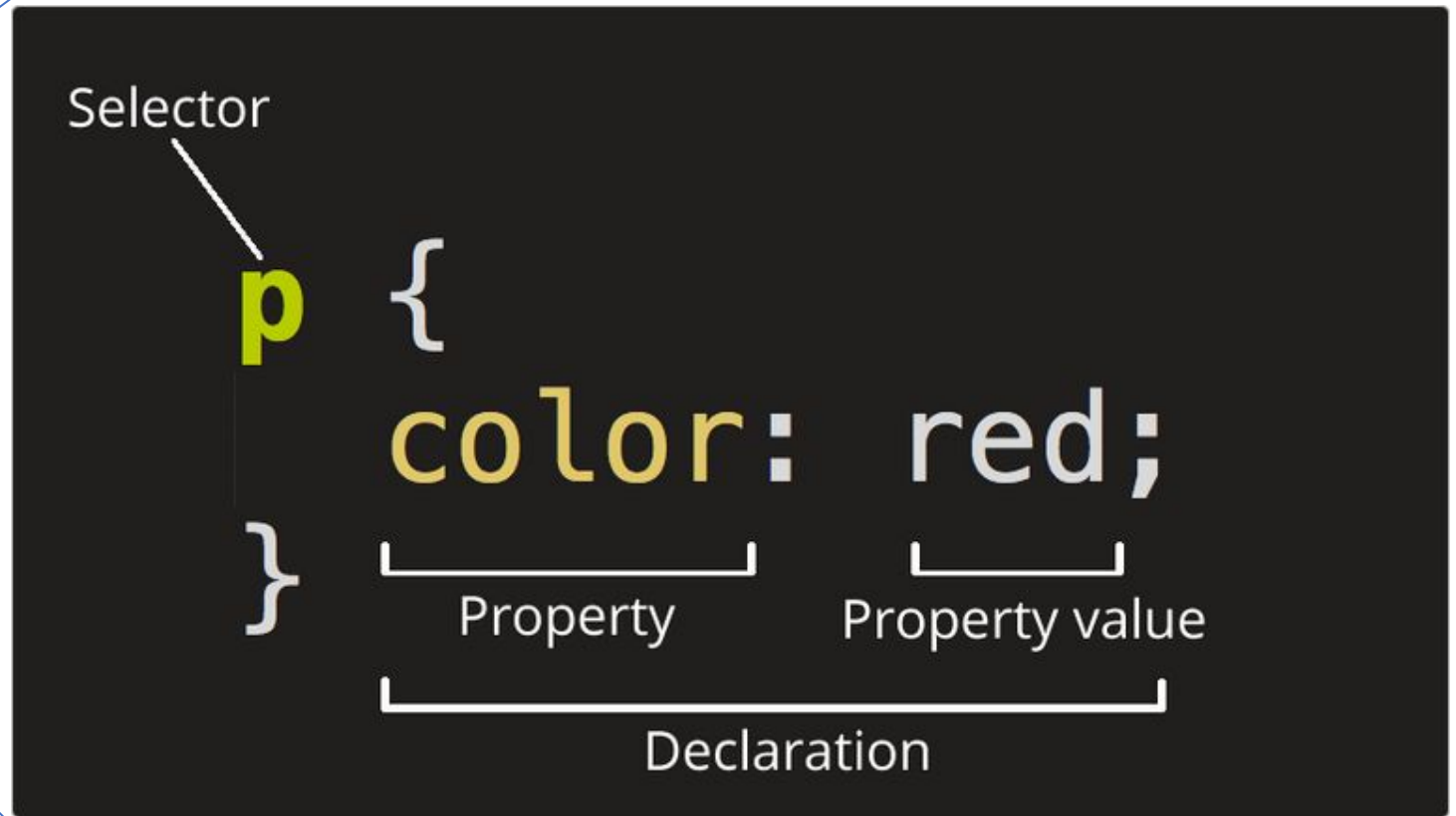
<head>

```
<link href="styles/style.css" rel="stylesheet" type="text/css" />
```

</head>

# Selectores

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



# Selectores

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

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

```
p,  
li,  
h1 {  
  color: red;  
}
```

# Selectores

¿Qué ocurre cuando a un elemento le aplica más de un selector?

```
.special {  
  color: red;  
}
```

```
p {  
  color: blue;  
}
```

```
<p class="special">¿De qué color soy?</p>
```

El que prevalece lo determinan las reglas de cascada y especificidad

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

```
p {  
  color: blue;  
}
```



# Selectores

Estos son algunos de los más comunes, pero hay más

Nombre del selector	Qué selecciona	Ejemplo
Selector de elemento (llamado algunas veces selector de etiqueta o tipo)	Todos los elementos HTML del tipo especificado.	<code>p</code> Selecciona <code>&lt;p&gt;</code>
Selector de identificación (ID)	El elemento en la página con el ID especificado (en una página HTML dada, solo se permite un único elemento por ID).	<code>#mi-id</code> Selecciona <code>&lt;p id="mi-id"&gt;</code> y <code>&lt;a id="mi-id"&gt;</code>
Selector de clase	Los elementos en la página con la clase especificada (una clase puede aparecer varias veces en una página).	<code>.mi-clase</code> Selecciona <code>&lt;p class="mi-clase"&gt;</code> y <code>&lt;a class="mi-clase"&gt;</code>
Selector de atributo	Los elementos en una página con el atributo especificado.	<code>img[src]</code> Selecciona <code>&lt;img src="mimagen.png"&gt;</code> pero no <code>&lt;img&gt;</code>
Selector de pseudoclase	Los elementos especificados, pero solo cuando esté en el estado especificado, por ejemplo cuando el puntero esté sobre él.	<code>a:hover</code> Selecciona <code>&lt;a&gt;</code> , pero solo cuando el puntero esté sobre el enlace.

# Selectores

Selector	Ejemplo
<a href="#">Selector de tipo</a>	<code>h1 { }</code>
<a href="#">Selector universal</a>	<code>* { }</code>
<a href="#">Selector de clase</a>	<code>.box { }</code>
<a href="#">Selector de ID</a>	<code>#unique { }</code>
<a href="#">Selector de atributo</a>	<code>a[title] { }</code>
<a href="#">Pseudoclase</a>	<code>p:first-child { }</code>
<a href="#">Pseudoelemento</a>	<code>p::first-line { }</code>
<a href="#">Operadores de combinación descendentes</a>	<code>article p</code>
<a href="#">Operador de combinación de elementos hijo</a>	<code>article &gt; p</code>
<a href="#">Operador de combinación de elementos hermanos adyacentes</a>	<code>h1 + p</code>
<a href="#">Operador de combinación general de elementos hermanos</a>	<code>h1 ~ p</code>

# Unidades de medida y colores

```
h1 {  
  color: black;  
  background-color: rgb(197, 93, 161);  
}
```

<integer>

<number>

<dimension>

<percentage>

# Unidades de medida

Unidad	Nombre
cm	Centímetros
mm	Milímetros
Q	Cuartos de milímetros
in	Pulgadas
pc	Picas
pt	Puntos
px	Píxeles

Unidad	Relativa a
em	Tamaño de letra del elemento padre, en el caso de propiedades tipográficas como <code>font-size</code> , y tamaño de la fuente del propio elemento en el caso de otras propiedades, como <code>width</code> .
ex	Altura x de la fuente del elemento.
ch	La medida de avance (ancho) del glifo "0" de la letra del elemento.
rem	Tamaño de la letra del elemento raíz.
lh	Altura de la línea del elemento.
vw	1% del ancho de la ventana gráfica.
vh	1% de la altura de la ventana gráfica.
vmin	1% de la dimensión más pequeña de la ventana gráfica.
vmax	1% de la dimensión más grande de la ventana gráfica.

# Colores

#02798b

#c55da1

#128a7d

```
.one {  
  background-color: #02798b;  
}
```

```
.two {  
  background-color: #c55da1;  
}
```

```
.three {  
  background-color: #128a7d;  
}
```

```
<div class="wrapper">  
  <div class="box one">#02798b</div>  
  <div class="box two">#c55da1</div>  
  <div class="box three">#128a7d</div>  
</div>
```

# Colores

rgb(2 121 139)

rgb(197 93 161)

rgb(18 138 125)

```
.one {  
  background-color: rgb(2 121 139);  
}  
  
.two {  
  background-color: rgb(197 93 161);  
}  
  
.three {  
  background-color: rgb(18 138 125);  
}
```

```
<div class="wrapper">  
  <div class="box one">rgb(2 121 139)</div>  
  <div class="box two">rgb(197 93 161)</div>  
  <div class="box three">rgb(18 138 125)</div>  
</div>
```

# Colores



```
.one {  
  background-color: rgb(2 121 139 / .3);  
}  
  
.two {  
  background-color: rgb(197 93 161 / .7);  
}  
  
.three {  
  background-color: rgb(18 138 125 / .9);  
}
```

```
<div class="wrapper">  
  <div class="box one">rgb(2 121 139 / .3)</div>  
  <div class="box two">rgb(197 93 161 / .7)</div>  
  <div class="box three">rgb(18 138 125 / .9)</div>  
</div>
```

# Colores

¿Se puede utilizar el model HSL?

¡Pruébalo!



# Modelo de cajas

*box model*

**Inline**

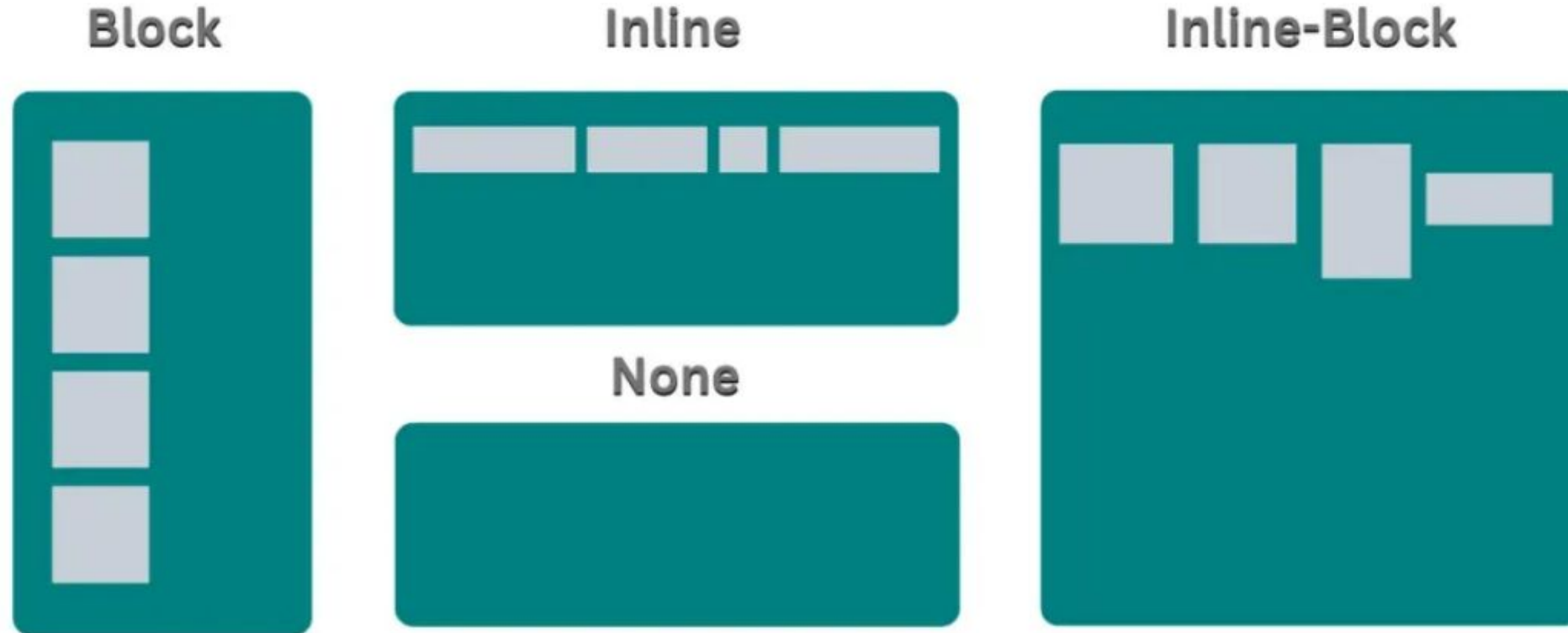


**Block**



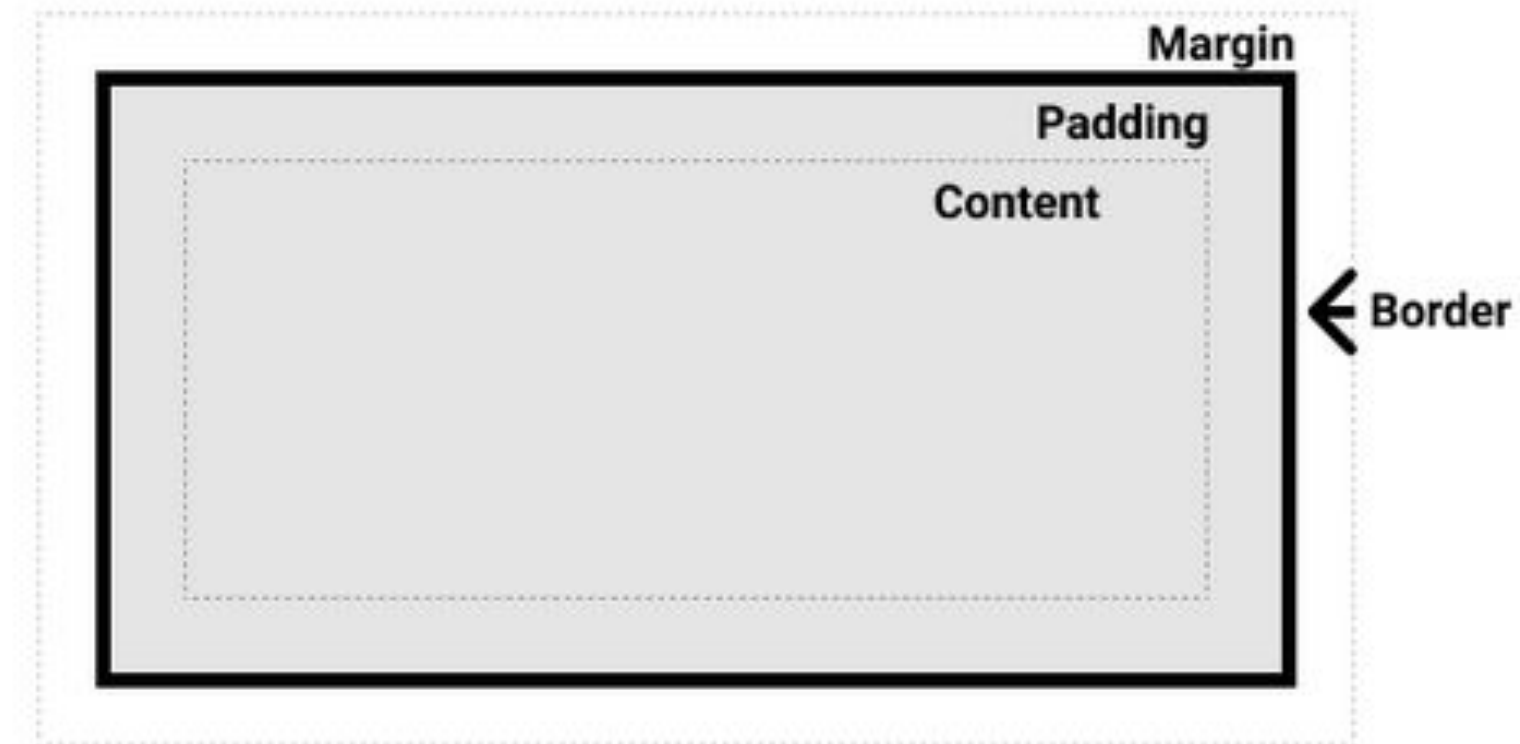
# Modelo de cajas

*box model*




# Modelo de cajas

*box model*



# Modelo de cajas

## *box model*

 mdn web docs

ReferencesGuidesPlusCurriculumBlogTools

Theme

Log in

Sign up for free

Bloques de construcción CSS > El modelo de caja

Español

Esta página ha sido traducida del inglés por la comunidad. Aprende más y únete a la comunidad de MDN Web Docs.

Filter

HTML — Structuring the web

Introduction to HTML

Multimedia and embedding

HTML tables

CSS — Styling the web

CSS first steps

CSS building blocks

Bloques de construcción CSS

Selectores CSS

Selectores de tipo, clase e ID

Selectores de atributo

El modelo de caja

Anterior

Overview: Bloques de construcción CSS

Siguiente

Todo en CSS tiene una caja alrededor, y comprender estas cajas es clave para poder crear diseños con CSS o para alinear elementos con otros elementos. En este artículo, echaremos un vistazo más de cerca al *modelo de cajas* en CSS con el que vas a poder crear diseños de compaginación más complejos con una comprensión de cómo funciona y la terminología relacionada.

Prerrequisitos:	Conocimientos básicos de informática, tener el <a href="#">software básico</a> instalado, conocimientos básicos de cómo <a href="#">trabajar con archivos</a> , conocimientos básicos de HTML (véase <a href="#">Introducción al HTML</a> ) y nociones de CSS (véase <a href="#">Primeros pasos con CSS</a> ).
Objetivo:	Aprender sobre el modelo de cajas en CSS, en qué consiste el modelo de cajas y cómo cambiar al modelo alternativo.

[https://developer.mozilla.org/es/docs/Learn/CSS/Building\\_blocks/The\\_box\\_model](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/The_box_model)

# Para practicar

- MDN web docs CSS

[https://developer.mozilla.org/es/docs/Learn/CSS/First\\_steps/Styling\\_a\\_biography\\_page](https://developer.mozilla.org/es/docs/Learn/CSS/First_steps/Styling_a_biography_page)

- w3schools

[https://www.w3schools.com/css/css\\_display\\_visibility.asp](https://www.w3schools.com/css/css_display_visibility.asp)

- Tareas de Moodle centros

**Jane Doe**

Web Developer

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean.

A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.

### **Contact information**

- Email: [jane@example.com](mailto:jane@example.com)
- Web: <http://example.com>
- Tel: 123 45678

# Posicionamiento y visualización

Basado en el *box model*

La propiedad **position** especifica cómo un elemento es posicionado en el documento.

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

CSS Demo: position

RESET


```
position: static;
```

```
position: relative;  
top: 40px; left: 40px;
```

```
position: absolute;  
top: 40px; left: 40px;
```

```
position: sticky;  
top: 20px;
```

In this demo you can control the position property for the yellow box.



To see the effect of sticky positioning, select the position: sticky option and scroll this container.

The element will scroll along with its container, until it is at the top of the container

# Posicionamiento y visualización

Tipos de posicionamiento:

Cuando el valor computado de `position` es ~~`static`~~, `relative`, `absolute`, `fixed`, o `sticky`.

- **relative:**

Las propiedades `top` y `bottom` especifican el desplazamiento vertical desde su posición original; las propiedades `left` y `right` especifican su desplazamiento horizontal.

- **absolute** o **fixed:**

Las propiedades `top`, `right`, `bottom`, y `left` especifican el desplazamiento desde los bordes del bloque contenedor del elemento. El elemento establece un nuevo contexto de formato de bloque para su contenido

- **sticky:**

Es tratado como un elemento posicionado relativamente hasta que su bloque contenedor cruza un límite establecido (como por ejemplo dando a `top` cualquier valor distinto de `auto`), dentro de su flujo principal (o el contenedor dentro del cual se mueve), desde el cual es tratado como "fijo" hasta que alcance el borde opuesto de su bloque contenedor.

# Posicionamiento y visualización

## Z-index

[https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_positioned\\_layout/Understanding\\_z-index](https://developer.mozilla.org/es/docs/Web/CSS/CSS_positioned_layout/Understanding_z-index)

## Entendiendo la propiedad CSS z-index

Usualmente se puede considerar que las páginas HTML tienen dos dimensiones, porque el texto, las imágenes y otros elementos son organizados en la página sin superponerse. Hay un solo flujo de renderizado, y todos los elementos son concientes del espacio ocupado por otros. El atributo `z-index` te permite ajustar el orden de las capas de los objetos cuando el contenido está siendo renderizado.



# Posicionamiento y visualización

## display

### Propiedad display

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

[https://www.w3schools.com/cssref/pr\\_class\\_display.php](https://www.w3schools.com/cssref/pr_class_display.php)

The `display` CSS property sets whether an element is treated as a [block or inline box](#) and the layout used for its children, such as [flow layout](#), [grid](#) or [flex](#).

Formally, the `display` property sets an element's inner and outer *display types*. The outer type sets an element's participation in [flow layout](#); the inner type sets the layout of children. Some values of `display` are fully defined in their own individual specifications; for example the detail of what happens when `display: flex` is declared is defined in the CSS Flexible Box Model specification.

### Try it

CSS Demo: display

RESET

display: block;

display: inline-block;

display: none;

display: flex;

display: grid;

Apply different `display` values on the dashed orange-bordered `div`, which contains three child elements.

Some text A.

Child 1Child 2Child 3

Some text B.

Más adelante

# Posicionamiento y visualización

Propiedades `visibility`, `overflow`, `opacity`



¿Diferencia con  
`display:none`?

```
.elemento-visible {  
    visibility: visible; /* El elemento es visible */  
}  
  
.elemento-oculto {  
    visibility: hidden; /* El elemento es invisible pero sigue ocupando espacio */  
}
```

# Posicionamiento y visualización

Propiedades `visibility`, `overflow`, `opacity`

visible,  
hidden,  
scroll, auto

```
.contenedor {  
  width: 200px;  
  height: 100px;  
  overflow: auto;  
  border: 1px solid black;  
}
```

CSS Demo: visibility

RESET

visibility: visible

visibility: hidden;

visibility: collapse;

Hide  
me

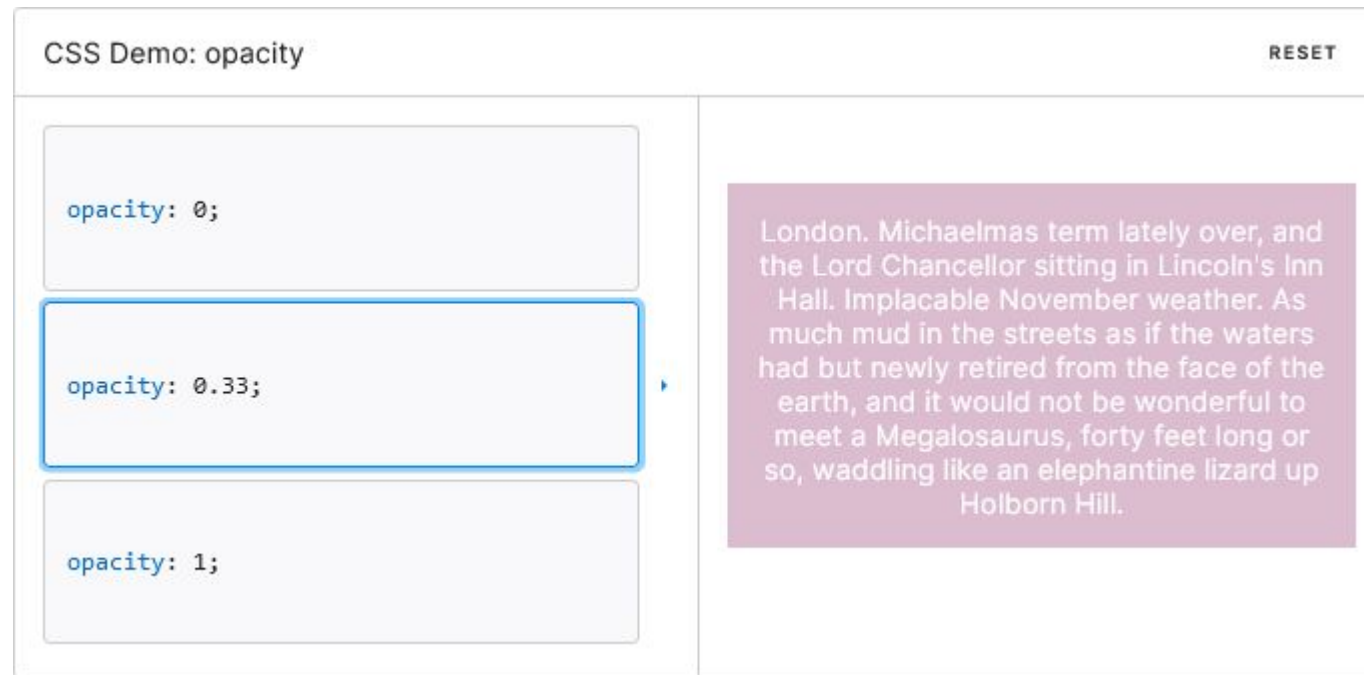
Item 2

Item 3

# Posicionamiento y visualización

Propiedades `visibility`, `overflow`, `opacity`

1: opaco  
0: transparente



# Posicionamiento y visualización

Propiedades `visibility`, `overflow`, `opacity`

- `visibility` vs `display`:  
Con `visibility: hidden`, el elemento sigue ocupando espacio, mientras que `display: none` lo elimina completamente del flujo.
- `opacity`:  
Mantiene el elemento visible pero parcialmente transparente (útil para efectos visuales), mientras que `visibility: hidden` lo oculta sin transparencia.
- `overflow`:  
Controla cómo gestionar el contenido que excede el tamaño del contenedor; por ejemplo, barras de desplazamiento o contenido oculto si se desborda.

# Posicionamiento y visualización


Propiedades `visibility`, `overflow`, `opacity`

Este es un texto largo que debería desbordar el área del contenedor, por lo que se controlará con `overflow: auto`.

Este texto es semi-transparente.

```
.contenedor {  
  width: 200px;  
  height: 100px;  
  border: 2px solid black;  
  overflow: auto; /* Solo mostrará barras de desplazamiento si es necesario */  
}  
  
.oculto {  
  visibility: hidden; /* Elemento invisible pero sigue ocupando espacio */  
}  
  
.transparente {  
  opacity: 0.5; /* Elemento con 50% de transparencia */  
}
```

# Bordes

```
border-width: 5px;  
border-style: solid;  
border-color:  aqua;  
border-radius: 15px;
```

CSS Demo: border RESET

border: solid;

border: dashed red;

border: 1rem solid;

border: thick double #32a1ce;

border: 4mm ridge rgba(211, 220, 50, .6)

This is a box with a border  
around it.

# Texto y fuente

```
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
font-style: italic;  
font-variant: small-caps;  
font-weight: bold;  
font-size: larger;
```

```
text-align: justify;  
letter-spacing: .15em;  
text-decoration: underline;  
word-spacing: 0.25em;  
text-transform: uppercase;  
text-indent: 0.5cm;  
line-height: normal;
```



# Texto y fuente

- [font-family](#)
- [font-size](#)
- [font-stretch](#)
- [font-style](#)
- [font-variant](#)
- [font-weight](#)
- [line-height](#)

CSS Demo: font-style

RESET

font-style: normal;

font-style: italic;

font-style: oblique;

font-style: oblique 40deg;

London. Michaelmas term lately over,  
and the Lord Chancellor sitting in  
Lincoln's Inn Hall. Implacable  
November weather. As much mud in  
the streets as if the waters had but  
newly retired from the face of the  
earth, and it would not be wonderful  
to meet a Megalosaurus, forty feet long  
or so, waddling like an elephantine  
lizard up Holborn Hill.

CSS Demo: font-family

RESET

font-family: Georgia, serif;

font-family: "Gill Sans", sans-serif;

font-family: sans-serif;

font-family: serif;

font-family: cursive;

font-family: system-ui;

London. Michaelmas term lately over,  
and the Lord Chancellor sitting in  
Lincoln's Inn Hall. Implacable  
November weather. As much mud in the  
streets as if the waters had but newly  
retired from the face of the earth, and it  
would not be wonderful to meet a  
Megalosaurus, forty feet long or so,  
waddling like an elephantine lizard up  
Holborn Hill.

CSS Demo: text-shadow

text-shadow: 1px 1px 2px pink;

text-shadow: #FC0 1px 0 10px;

text-shadow: 5px 5px #558ABB;

text-shadow: red 2px 5px;

text-shadow: 5px 10px;

text-shadow: 1px 1px 2px red, 0 0 1em blue;

Far out in the uncharted  
backwaters of the unfashionable  
end of the western spiral arm of  
the Galaxy...

# Texto y fuente (ejemplo)

```
/* Estilo para texto */  
h1 {  
    font-family: 'Arial', sans-serif;  
    font-size: 32px;  
    text-align: center;  
    text-transform: uppercase;  
    color: #333;  
}  
  
p {  
    font-size: 16px;  
    line-height: 1.6;  
    color: #555;  
    text-align: justify;  
    letter-spacing: 1px;  
}
```

# Enlaces

estado (pseudo-clases)

```
a:link {  
    color: blue;  
    text-decoration: none;  
}  
  
a:visited {  
    color: purple;  
}  
  
a:hover {  
    color: green;  
    text-decoration: underline;  
}  
  
a:active {  
    color: red;  
}
```

# Enlaces

## Algunos efectos interesantes

```
a {  
  color: blue; /* Establece el color de los enlaces */  
}  
  
a:hover {  
  color: red; /* Cambia el color cuando el mouse pasa por encima */  
}
```

```
a {  
  text-decoration: none; /* Elimina el subrayado predeterminado */  
}  
  
a:hover {  
  text-decoration: underline; /* Añade el subrayado al pasar el mouse */  
}
```

# Enlaces

## Enlaces con iconos

```
<a href="#">
   Enlace con icono
</a>
```

```
a img {
  width: 20px;
  height: 20px;
  vertical-align: middle; /* Alinea el icono con el texto */
}
```

# Enlaces

## **Reto 1:**

Cambiar el fondo y bordes en los enlaces, añada también *padding*

## **Reto 2:**

Hacer que el enlace tenga apariencia de un botón con estilo bloque

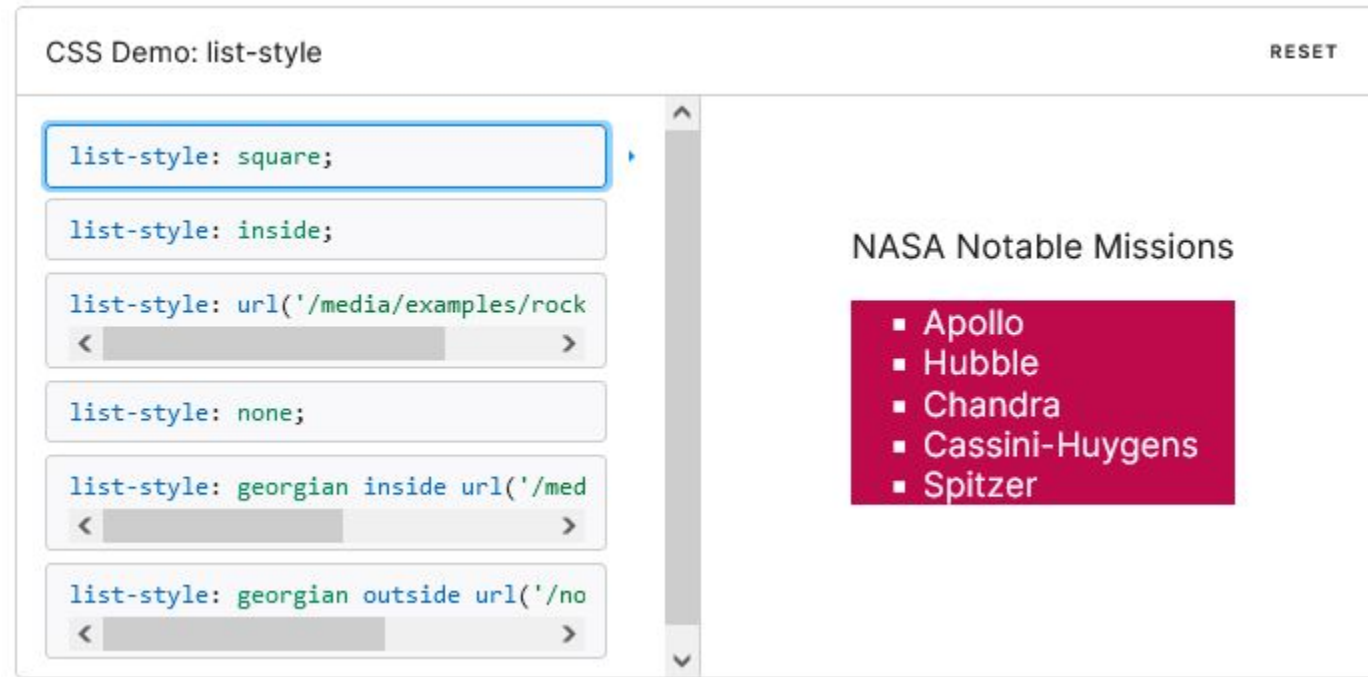
## **Reto 3:**

Cambiar el cursor al pasar por encima del enlace

# Imágenes

```
/* Estilo para imágenes */  
img {  
    width: 100%;  
    height: auto;  
    border: 3px solid #000;  
    border-radius: 15px;  
    object-fit: cover;  
}  
  
div {  
    background-image: url('fondo.jpg');  
    background-size: cover;  
    background-position: center;  
}
```

# Listas



```
list-style-type: square;  
list-style-position: 20px;  
list-style-image: url(image.gif);  
marker-offset: auto;
```



# Listas

```
/* Estilo para listas */  
ul {  
    list-style-type: square;  
    list-style-position: inside;  
    padding: 10px;  
}  
  
ol {  
    list-style-type: decimal;  
    padding-left: 20px;  
}  
  
li {  
    margin-bottom: 8px;  
}
```

# Tablas

```
border-collapse: separate;
empty-cells: show;
border-spacing: 2px;
table-layout: auto;
caption-side: bottom;

/* Columns */
column-count: 10;
column-gap: 5px;
column-rule-width: medium;
column-rule-style: dotted ;
column-rule-color: □black;
column-width: 10px;
column-span: all;
```

# Tablas (ejemplo)

```
/* Estilo para tablas */
table {
    width: 100%;
    border-collapse: collapse;
}

th, td {
    padding: 12px;
    border: 1px solid #ddd;
    text-align: left;
}

th {
    background-color: #f2f2f2;
}

tr:nth-child(even) {
    background-color: #f9f9f9;
}
```

# Formularios

```
textarea {  
    resize: vertical;  
    height: 150px;  
}  
  
input[type="submit"],  
button {  
    background-color: #4CAF50;  
    color: white;  
    padding: 12px 20px;  
    border: none;  
    border-radius: 5px;  
    cursor: pointer;  
    font-size: 16px;  
}  
  
input[type="submit"]:hover,  
button:hover {  
    background-color: #45a049;  
}
```

```
/* Estilo para formularios */  
form {  
    max-width: 600px;  
    margin: 20px auto;  
    padding: 20px;  
    border: 1px solid #ccc;  
    border-radius: 10px;  
    background-color: #f9f9f9;  
}  
  
label {  
    display: block;  
    margin-bottom: 8px;  
    font-weight: bold;  
    color: #333;  
}  
  
input[type="text"],  
input[type="email"],  
input[type="password"],  
textarea,  
select {  
    width: 100%;  
    padding: 10px;  
    margin-bottom: 15px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
    font-size: 16px;  
}
```

# Formularios



¿Podemos validar un formulario  
con CSS?

# Layout

*cómo organizar los elementos en una página*

```
float: none;  
clear: both;  
display: block;  
height: fit-content;  
width: auto;  
margin: top right bottom left;  
padding: top right bottom left;  
overflow: hidden;  
visibility: visible;
```

En una unidad posterior veremos cómo Grid y Flexbox han reemplazado a float y clear

El box model con contenido, margen, relleno y borde

El control de la posición (static, relative, absolute, fixed) y la visibilidad permite controlar los elementos

# Fondo

## *background*

## background

✓ Baseline Widely available



The `background` [shorthand CSS](#) property sets all background style properties at once, such as color, image, origin and size, or repeat method. Component properties not set in the `background` shorthand property value declaration are set to their default values.

## Try it

CSS Demo: background

RESET

```
background: green;
```

```
background: content-box radial-gradient(c
```

```
background: no-repeat url("../media/ex
```

```
t-x url("../media/examples/star.png");
```

```
background: center / contain no-repeat ur  
#eee 35% url("../media/exa
```



# Propiedades de *shorthand*

*propiedades que te permiten establecer simultáneamente valores de múltiples propiedades distintas*

- Font Shorthand
- Border Shorthand
- Padding Shorthand
- Background Shorthand
- Margin Shorthand
- List-Style Shorthand
- ... y hay algunos más

```
margin-top: 20px;
```

```
margin-right: 30px;
```

```
margin-bottom: 70px;
```

```
margin-left: 80px;
```



```
margin: 20px 30px 70px 80px;
```



# Versión para imprimir

CSS permite:

- crear hojas de estilo específicas para la impresión

```
<link rel="stylesheet" type="text/css" href="estilo.css">  
<link rel="stylesheet" type="text/css" href="estilo_imprimir.css" media="print">  
</head>
```

- usar la regla `@media print` para definir un conjunto reglas de estilos que afectarían solamente a la impresión

```
@media print {  
  body {  
    font-size: 11pt;  
    background-color: #fff;  
  }  
  table {  
    border: none;  
  }  
}
```

# Personalizar el cursor

## cursor

The `cursor` [CSS](#) property sets the mouse cursor, if any, to show when the mouse pointer is over an element.

The cursor setting should inform users of the mouse operations that can be performed at the current location, including: text selection, activating help or context menus, copying content, resizing tables, and so on. You can specify either the type of cursor using a keyword, or load a specific icon to use (with optional fallback images and mandatory keyword as a final fallback).

## Try it

CSS Demo: cursor

RESET

cursor: help;

cursor: wait;

cursor: crosshair;

cursor: not-allowed;

cursor: zoom-in;

cursor: grab;

Move over this element to see the cursor style.

Para el comportamiento *responsive*

## pointer

✓ Baseline Widely available



The `pointer` [CSS media feature](#) tests whether the user has a pointing device (such as a mouse), and if so, how accurate the *primary* pointing device is.

**Note:** If you want to test the accuracy of any pointing device, use `any-pointer` instead.

## Syntax

The `pointer` feature is specified as a keyword value chosen from the list below.

`none`

The primary input mechanism does not include a pointing device.

`coarse`

The primary input mechanism includes a pointing device of limited accuracy, such as a finger on a touchscreen.

`fine`

The primary input mechanism includes an accurate pointing device, such as a mouse.

# Personalizar el cursor

```
<button class="enviar">Enviar</button>  
<button class="deshabilitado" disabled>No disponible</button>
```

```
.enviar {  
    cursor: pointer;  
}  
  
.deshabilitado {  
    cursor: not-allowed;  
}
```

# Prioridad en las declaraciones CSS

La prioridad o *especificidad* en CSS dicta qué estilos se aplican cuando hay conflictos entre las reglas.

- Especificidad: Depende del tipo de selector utilizado:
  - Selectores de ID (#id) tienen mayor prioridad.
  - Selectores de clase (.clase) tienen menor prioridad que los de ID pero mayor que los de etiqueta (h1).
- `!important`: Se utiliza para forzar una propiedad, pero debe ser usado con cautela.

# Prioridad en las declaraciones CSS

```
p {  
    color: blue;  
}  
  
#especial {  
    color: green;  
}  
  
p#importante {  
    color: red;  
}
```

Corregir el siguiente código para que el color del texto en p sea rojo, utilizando la especificidad adecuada.

```
p#importante {  
    color: red !important;  
}
```

# Validador

Deutsch English Español Français 한국어 Italiano Nederlands 日本語 Polski Português Русский فارسی Svenska Български Українська Čeština Romanian Magyar Ελληνικά हिन्दी 简体中文



## CSS Validation Service

Verifica Hojas de Estilo en Cascada (CSS) y documentos (X)HTML con hojas de estilo

mediante URI

mediante Carga de Archivo

mediante Entrada directa

### Validar mediante URI

Introduce la URI de un documento (HTML con CSS o sólo CSS) que desees validar:

Dirección:

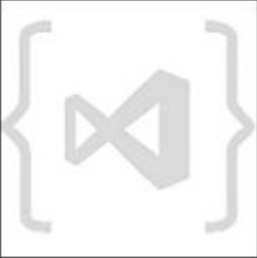
► Más opciones

Check

# Validador

Visual Studio | Marketplace

Visual Studio Code > Programming Languages > CSSTree validator



## CSSTree validator


Sergey Melyukov | 📄 129,081 installs | ★★★★★ (11) | Free

Validate CSS according to W3C specs and browser implementations

[Install](#) [Trouble Installing?](#)

Visual Studio | Marketplace

Visual Studio Code > Linters > W3C Web Validator



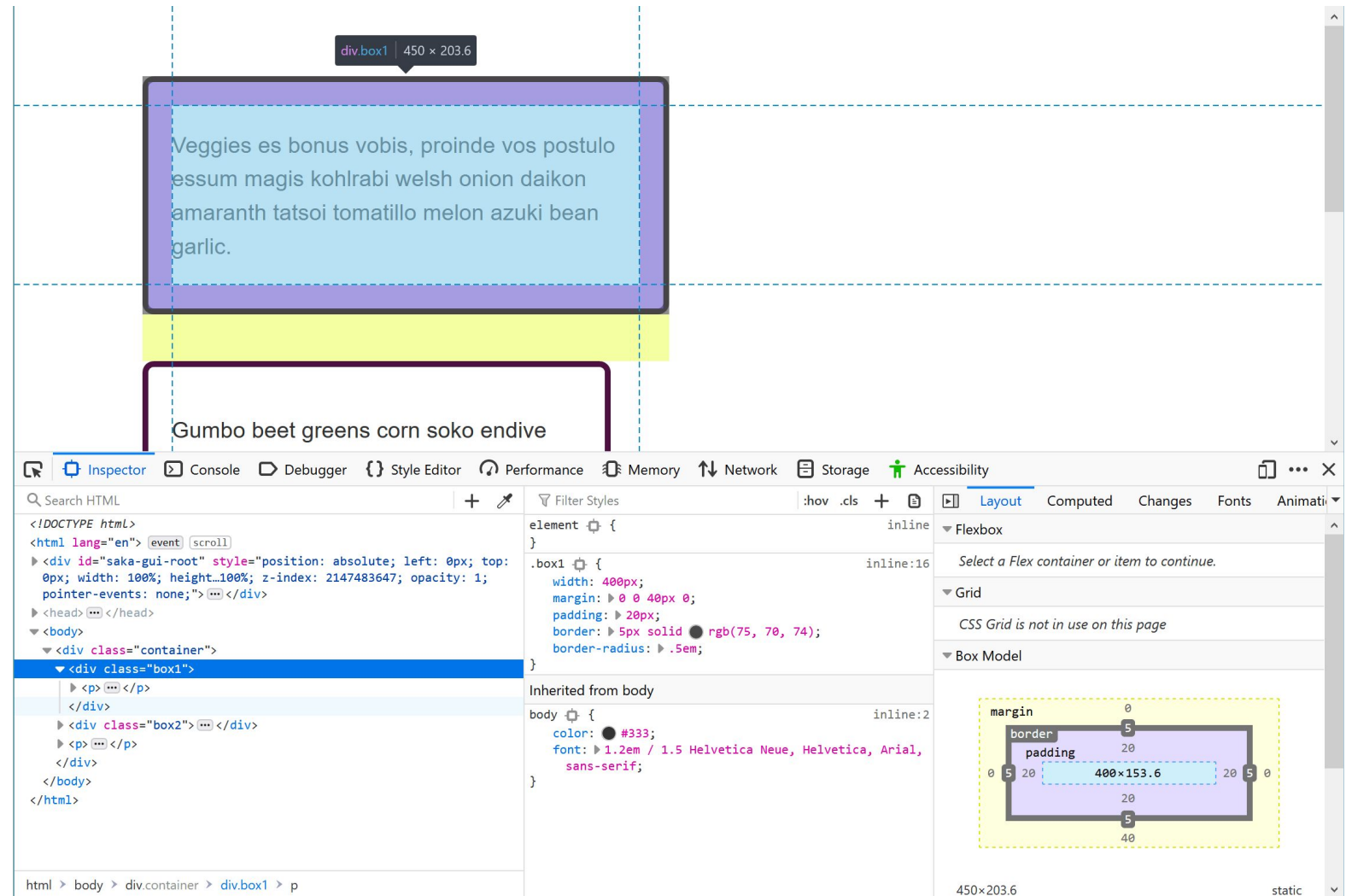
## W3C Web Validator

Celian Riboulet | 📄 163,678 installs | ★★★★★ (12) | Free

Check the validity of your HTML/CSS files in one click

[Install](#) [Trouble Installing?](#)

# Debug

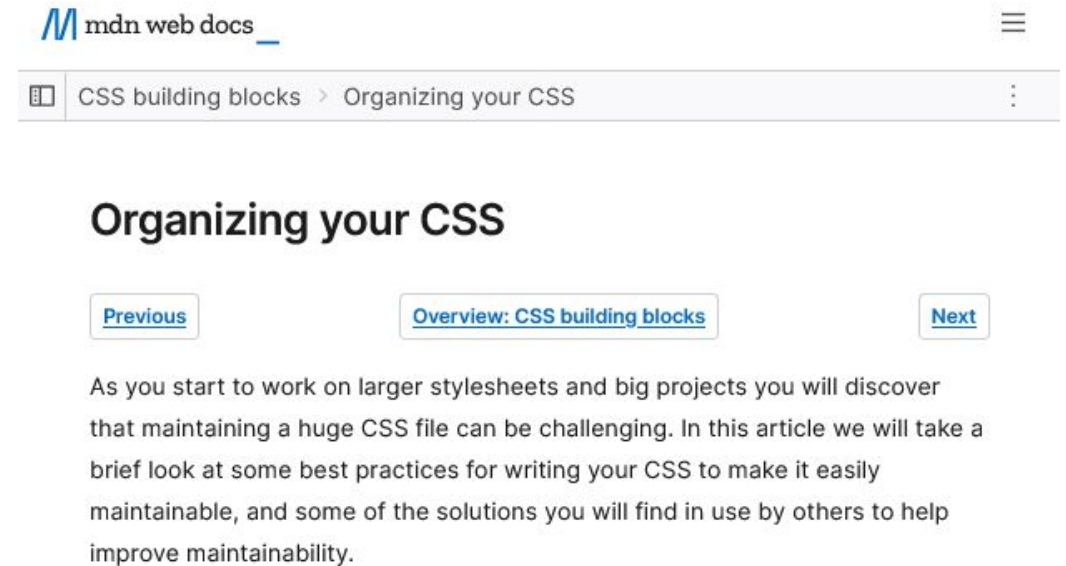




# Recomendaciones generales

- Utilizar *shorthand* cuando sea posible.
- Usar unidades relativas
- Organizar el código para facilitar la lectura (usar comentarios, indentación, etc.)
- Usar clases en lugar de id
- Evitar el CSS *inline*
- Usa una nomenclatura coherente

Además hay Convenciones de Nomenclatura como BEM: *Bloque, Elemento y Modificador.*



# Recursos útiles

- Cheatsheets! en la plataforma
- [css-tricks](https://css-tricks)
- Juego <https://flukeout.github.io/>

