

LMSGI

Bloque 4 - Tema 1 - Introducción a CSS. Selectores - reglas y propiedades revisar

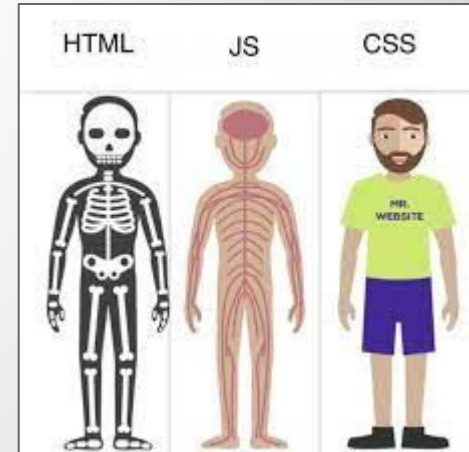


Resumen

- Atributos id y class de HTML
- Introducción
- Especificación
- Herencia en CSS
- ¿Cómo utilizo CSS?
- Tipos de medios
- Reglas
- Selectores
- Cómo indicar color en CSS
- Propiedades básicas color, background-color y opacity
- Un primer ejemplo completo
- Depuración básica de CSS
- Documentación

<https://lenguajehtml.com/p/css/>

[Fuente](#)



Atributos id y class de HTML

- **id** → asigna un nombre a un elemento, tiene que ser **único** en la página (debe comenzar con una letra). Lo hemos utilizado como vínculo interno (#)
- **class** → **asigna un nombre de clase o un conjunto de nombres de clase a un elemento.**
 - Se puede asignar el mismo nombre o nombres de clase a cualquier número de elementos.
 - Se puede indicar que un elemento pertenece a más de una clase → cada nombre se separa por un espacio.

```
<div class="ciudades capitales" id="listaCiudades">  
<h2>Tokyo</h2>  
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous  
metropolitan area in the world.</p>  
<p>It is the seat of the Japanese government and the Imperial Palace, and the home of the Japanese  
Imperial Family.</p>  
<p>The Tokyo prefecture is part of the world's most populous metropolitan area with 38 million people  
and the world's largest urban economy.</p>  
</div>
```

Utilizaremos ambos atributos como selectores en hojas de estilo

Introducción a CSS

- Con CSS aplicamos formato a las páginas HTML (o XML)
- Los formatos aplicados con CSS eran aplicados con etiquetas y atributos HTML (align, font, ... - hoy desaprobadas) y con JavaScript.
- Los documentos HTML son más sencillos facilitando la descarga y legibilidad del mismo
- Se puede aplicar una hoja de estilo a múltiples páginas Web → coherencia en el estilo de todo el sitio Web
- Sin realizar modificaciones en la página HTML se pueden aplicar estilos diferentes en función del medio (papel, pda, pantalla del ordenador, TV, ...)
- Con estilos se consiguen más efectos que los que se podrían conseguir con HTML

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
  color: green;
}
p {
  color: white;
}
</style>
</head>
<body>

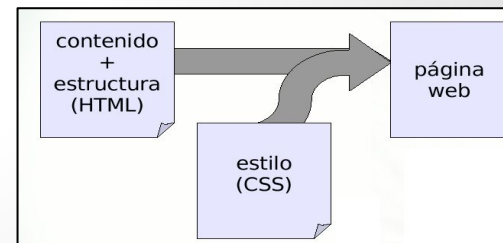
<h1>IES Valle del Jerte -
Plasencia</h1>

<p>Lenguaje de Marcas y Sistemas de
Gestión de la Información</p>

</body>
</html>
```

IES Valle del Jerte - Plasencia

Lenguaje de Marcas y Sistemas de Gestión de la Información



Separar presentación de contenido → con HTML definimos los contenidos y su estructura semántica y con CSS los estilos (formatos)

Especificación

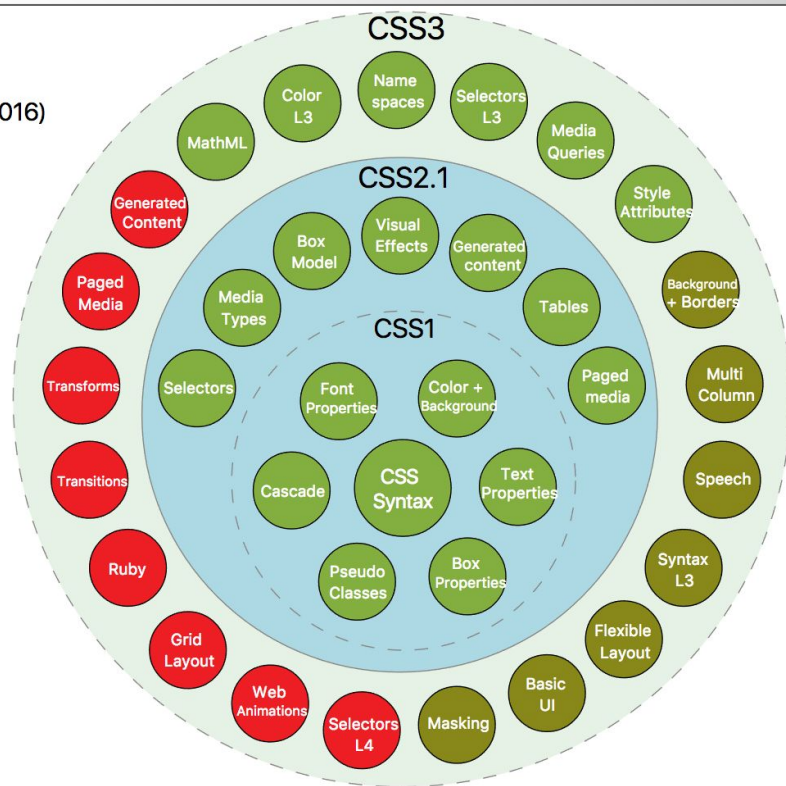
- Normalizado por W3C → <https://www.w3.org/Style/CSS/current-work> → parte del núcleo especificado en CSS 2.1 y lo amplía/modifica mediante **módulos** nuevos → CSS 3
- La mayoría de los navegadores soportan CSS, aunque no todas las propiedades → <http://css3test.com/>, <http://caniuse.com/#search=css>, http://www.w3schools.com/cssref/css3_browsersupport.asp etc
- Validador de código CSS → <https://jigsaw.w3.org/css-validator/>



CSS3

Taxonomy & Status (Dezember 2016)

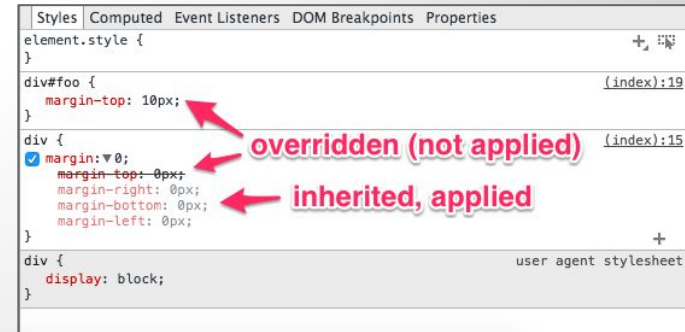
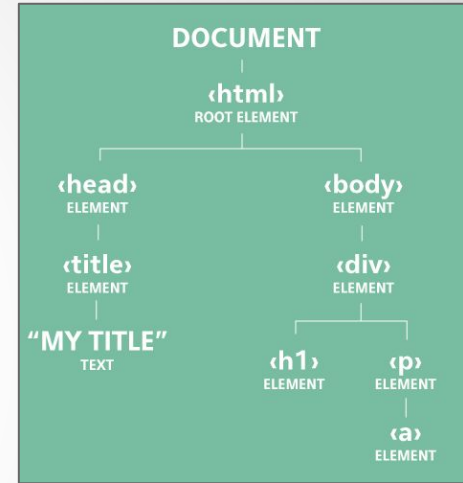
- W3C Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Obsolete or inactive



<https://validator.w3.org/unicorn/> ← validador unificado

Herencia en CSS

- **CSS → Cascading Style Sheets (Hojas de estilo en cascada)** → los estilos creados para un elemento **pueden ser heredados**, es decir, también se aplican, a todos aquellos elementos que se encuentren *<<por debajo>>*.
- Por ejemplo, si se especifica un tipo de letra para el elemento `<body>` todos aquellos elementos que puedan heredar las características se presentarán con el mismo tipo de fuente. Así, el elemento `<p>` heredará el tipo de letra salvo que especifiquemos lo contrario → *hojas de estilo muy compactas, una sola norma en el primer elemento se aplica a todos*.
- Todas las hojas de estilo CSS **son insensibles a las mayúsculas/minúsculas**, excepto para las partes que no están bajo el control de CSS → **es recomendable escribir en minúsculas**



¿Cómo utilizo CSS?

- Estilos en **línea** → en el propio elemento
- Estilos **internos** → en el mismo fichero html encerrados en `<style>`
- Estilos **externos** → en ficheros de **texto** independientes (la **extensión suele ser .css**) → **estilos externos** → los **vinculamos con <link>**
- En caso de contradicción a la hora de aplicar reglas de estilo, la **prioridad** es:
 - Estilos en línea
 - Estilos internos - Estilos externos → **el último en aparecer redefine**
 - Estilos por defectos del navegador

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
    color: green;
}
p {
    color: white;
}
</style>
<link rel="stylesheet" type="text/css"
href="css/estilos.css">
</head>
<body>

<h1>IES Valle del Jerte - Plasencia</h1>

<p><strong>Lenguaje de Marcas</strong> y
Sistemas de Gestión de la Información</p>

<p>Administración de Sistemas Informáticos
en Red</p>

<p style="color:blue">Desarrollo de
Aplicaciones Multiplataforma</p>

</body>
</html>
```

IES Valle del Jerte - Plasencia

Lenguaje de Marcas y Sistemas de Gestión de la Información

Administración de Sistemas Informáticos en Red

Desarrollo de Aplicaciones Multiplataforma

Internos → atributos opcionales de `<style>`:
type (por defecto text/css) y media

Externos → también podemos utilizar el atributo
media → el fichero referenciado es de texto y
contiene las reglas, de forma similar a los internos
(sin incluir `<style>` → **ES LA OPCIÓN
RECOMENDADA → REUTILIZACIÓN Y
SEPARACIÓN CONTENIDOS - FORMATOS** →
Otra opción es con `@import`

```
<style type="text/css" media="screen">
    @import '/css/estilos.css';
</style>
```

En línea → separadas por ; podemos poner todas
las reglas que queramos.

Comentarios en hojas de estilo internas y externas → `/* ... comentario ... */`

Tipos de Medio

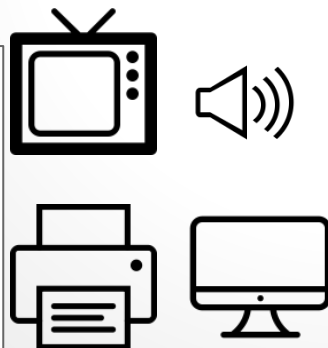
- Podemos indicar en qué **medios** se aplican las reglas:
 - Utilizando el **atributo opcional media** en <link> o en <style>
→ *Lo utilizaremos en este tema*
 - Utilizando **media queries** → [Media queries - CSS | MDN](#)
- Incluso podemos aplicarlas en función de la resolución y otras propiedades del entorno en el que se está visualizando la página web.
- Depurar CSS de media Print en Chrome: [Using Chrome's Element Inspector in Print Preview Mode? - Stack Overflow](#)
- [Cómo crear una hoja de estilo para impresión - HTML & CSS: Curso práctico avanzado](#)

```
@media print {  
  * {  
    color: black;  
  }  
}  
  
@media screen {  
  a {  
    background-color: #bbafed;  
  }  
}
```

Tipos de medios

- all**
Aplicable a todos los dispositivos.
- print**
Destinado material paginado y para documentos visibles en pantalla en modo de vista previa para impresión. Por favor, consulte la sección de medios paginados, y la [sección de Media en el tutorial de Introducción](#) para más información acerca de problemas de formateo específicos para los medios paginados.
- screen**
Destinado a principalmente a pantallas de computadora a color.
- speech**
Destinado a sintetizadores de voz. Nota: CSS2 tenía un tipo de medio similar llamado 'aural' para este propósito. Vea el apéndice sobre hojas de estilo aural para más detalles.

Nota: CSS2.1 y Media Queries 3 definió varios tipos de media adicionales (tty, tv, projection, handheld, braille, embossed, aural), pero fueron discontinuados en CSS Media Queries 4 y no deben ser usados.



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

```
<!-- CSS media query on a link element -->  
<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />  
  
<!-- CSS media query within a stylesheet -->  
<style>  
  @media (max-width: 600px) {  
    .facet_sidebar {  
      display: none;  
    }  
  }  
</style>
```


Reglas

- Los estilos se especifican mediante **reglas** → cada una especifica un conjunto de propiedades relacionadas con el formato.
- **Selector** → elemento o elementos a los que se aplicarán las reglas (h1, p, b, .clase, #id, etc)
- Si al final de una regla ponemos **!important** tiene prioridad en su aplicación

Selector {propiedad-1:valor1; ...;propiedad-n:valor-n}

```
p { font-style: italic ! important }
```

A CSS ruleset (or rule):

```
div p, #id:first-line {  
    background-color : red ;  
    background-style : none  
}
```

Group of selectors

Declarations block

A CSS declarations block:

```
{  
    background-color : red ;  
    background-style : none  
}
```

Declarations

The SEMI-COLON separates the two declarations

The final SEMI-COLON is not necessary, but good practice

Reglas abreviadas y resumidas

- **Forma abreviada de poner propiedades** → Cuando varios selectores comparten las mismas declaraciones, pueden agruparse en una lista separada por comas.

```
H1 { font-family: sans-serif }
H2 { font-family: sans-serif; }
H3 { font-family: sans-serif; }
```



```
H1, H2, H3 { font-family: sans-serif; }
```

Thus, the following rules:

```
h1 { font-weight: bold; }
h1 { font-size: 12px; }
h1 { line-height: 14px; }
h1 { font-family: Helvetica; }
h1 { font-variant: normal; }
h1 { font-style: normal; }
```

are equivalent to:

```
h1 {
  font-weight: bold;
  font-size: 12px;
  line-height: 14px;
  font-family: Helvetica;
  font-variant: normal;
  font-style: normal;
}
```

- **Propiedades resumidas** → nos permite especificar los valores de varias propiedades con una sola propiedad

```
H1 {
  font-weight: bold;
  font-size: 12pt;
  line-height: 14pt;
  font-family: Helvetica;
  font-variant: normal;
  font-style: normal;
  font-stretch: normal;
  font-size-adjust: none;
}
```

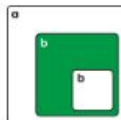
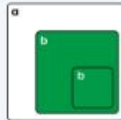




```
H1 { font: bold 12pt/14pt Helvetica; }
```

Selectores: Introducción

Selector/res {propiedad1:valor1; propiedad2: valor2;}

- Identifican a qué partes del documento se aplican las reglas.
- Tipos básicos:
 - **Elementos** → Todas las apariciones de un elemento de HTML específico (p.ej., todos los elementos P, todos los elementos H1, etc.)
 - **Id** → Las apariciones únicas de un elemento de HTML (es decir, cuyo atributo id tome cierto valor).
 - **Clases** → Todas las apariciones de un elemento de HTML que pertenezcan a una clase específica (es decir, cuyo atributo class tome cierto valor).
 - **Selector universal** → * (todos los elementos HTML)
- Listado de selectores CSS 1, 2, 3 y 4:
 - [CSS Selectors Reference](#) ← **IMPORTANTE!!!**
 - [CSS selectors - CSS: Cascading Style Sheets | MDN](#)
 - [Es/Selectores avanzados de CSS - Web Education Community Group](#)
 - [CSS Selectores \(1\)](#) ← *Explicado en Castellano con ejemplos.*
- Están trabajando en la versión 4 de los selectores → [Selectors Level 4](#)

Preview	Selector	Description
	a > b Child Combinator	Select all b elements that are directly inside of a elements.
	a b Descendant Combinator	Select all b elements that are anywhere inside of a elements.
	a + b Adjacent sibling combinator	Select all b elements that are immediately next to a elements.
	a ~ b General sibling combinator	Select all b elements that are

[Fuente](#)

Ejemplo de reglas CSS con selectores ID y CLASS



```
<!DOCTYPE html>
<html>
<head>
<style>
#paral {
    color: blue;
}
.destacado {
    color: green;
    font-weight: bolder;
}
.final {
    text-decoration: underline;
}
p {
    border: 1px solid black;
}
</style>
</head>
<body>

<p id="paral" class="destacado">Lenguaje de Marcas y Sistemas de Gestión
de la Información</p>
<p>Desarrollo de Aplicaciones Multiplataforma</p>
<p class="destacado final">Administración de Sistemas Informáticos en
Red</p>

</body>
</html>
```

Lenguaje de Marcas y Sistemas de Gestión de la Información

Desarrollo de Aplicaciones Multiplataforma

Administración de Sistemas Informáticos en Red

Los selectores de ID
comienzan con # y los de
clases con .

Selector universal *

```
*.noticia { color: green } /* todos los elementos con class~=noticia */
```

```
.noticia { color: green } /* todos los elementos con class~=noticia */
```

```
h1.noticia { color: green } /* elementos h1 con class~=noticia */
```

```
<h1>No aparece en verde</h1> <h1 class="noticia">Sí aparece en verde</h1>
```

```
p.noticia.marino { color: green }
```

```
<p class="noticia rojo azul marino">Párrafo 1, al que SÍ se aplica la regla</p>  
<p class="noticia rojo">Párrafo 2, al que NO se aplica la regla</p>
```


Selectores en CSS 2.1

SELECTORES

Patrón	Aplicado a:	Ejemplo
*	Cualquier elemento	N/D
e	Elemento <code><e></code>	<code><e></code>
e f	Cualquier <code><f></code> que esté incluido en <code><e></code>	<code><e>...<d>...<f>...</d>...</e></code>
e > f	Cualquier <code><f></code> que es hijo de <code><e></code>	<code><e>...<f>...</f>...</e></code>
e:first-child	<code><e></code> cuando es el primer hijo de su elemento padre	<code><f><e></e>...</f></code>
e + f	<code><f></code> si está precedido inmediatamente por un <code><e></code>	<code>...</e> <f>...<f></code>
e[miAtr]	<code><e></code> si posee el atributo "miAtr"	<code><e miAtr="xxx".../></code>
e[miAtr="v"]	<code><e></code> si posee el atributo "miAtr" y este tiene el valor "v"	<code><e miAtr="v".../></code>
e[miAtr~="v"]	<code><e></code> si su atributo "miAtr" es una lista de valores separados por espacios y uno es "v"	<code><e miAtr="x y z v".../></code>
e[lang "es"]	<code><e></code> si su atributo "lang" es una lista de valores separados por guiones y comienza con "es"	<code><e lang="es-ES" .../></code>
e.miClase	<code><e></code> cuya clase es "miClase"	<code><e class="miClase".../></code>
e#miId	<code><e></code> cuyo identificador es miId	<code><e id="miId".../></code>
a:link	Enlaces no visitados	N/D
a:visited	Enlaces visitados	N/D
e:active	<code><e></code> cuando es activado (tiempo entre que se pulsa un botón sobre él y se suelta)	N/D
e:hover	<code><e></code> cuando se posiciona el cursor sobre él pero no se activa	N/D
e:focus	<code><e></code> cuando tiene el foco posicionado en él	N/D
e:lang(c)	<code><e></code> si está marcado con el lenguaje c	N/D
e:first-line	Primera línea de <code><e></code>	N/D
e:first-letter	Primera letra de <code><e></code>	N/D
e:before	Aplica contenido antes de <code><e></code>	N/D
e:after	Aplica contenido después del elemento <code><e></code>	N/D

Se pueden combinar entre ellos. Por ejemplo:

```
body > ol > li p {  
  reglas  
}
```

Las reglas se aplicarían a los p que estén dentro de un li, en cualquier nivel, que sean hijos (primer nivel) de un ol (lista ordenada) que estén dentro de body

¡Importante! - **a:visited** solo puede utilizar un conjunto restringido de propiedades y siempre que hayan sido previamente establecidas en el estado no visitado (a o a:link), es decir no se puede utilizar para agregar propiedades que aún no están presentes en el enlace.

[Más información](#)

pseudo-clases

pseudo-elementos

Los 30 selectores CSS que debes memorizar

Selectores hasta CSS 3

Pattern	Meaning	Described in section	First defined in CSS level
*	any element	Universal selector	2
E	an element of type E	Type selector	1
E[foo]	an E element with a "foo" attribute	Attribute selectors	2
E[foo="bar"]	an E element whose "foo" attribute value is exactly equal to "bar"	Attribute selectors	2
E[foo~="bar"]	an E element whose "foo" attribute value is a list of whitespace-separated values, one of which is exactly equal to "bar"	Attribute selectors	2
E[foo^="bar"]	an E element whose "foo" attribute value begins exactly with the string "bar"	Attribute selectors	3
E[foo\$="bar"]	an E element whose "foo" attribute value ends exactly with the string "bar"	Attribute selectors	3
E[foo*="bar"]	an E element whose "foo" attribute value contains the substring "bar"	Attribute selectors	3
E[foo ="en"]	an E element whose "foo" attribute has a hyphen-separated list of values beginning (from the left) with "en"	Attribute selectors	2
E:root	an E element, root of the document	Structural pseudo-classes	3
E:nth-child(n)	an E element, the n-th child of its parent	Structural pseudo-classes	3
E:nth-last-child(n)	an E element, the n-th child of its parent, counting from the last one	Structural pseudo-classes	3
E:nth-of-type(n)	an E element, the n-th sibling of its type	Structural pseudo-classes	3
E:nth-last-of-type(n)	an E element, the n-th sibling of its type, counting from the last one	Structural pseudo-classes	3
E:first-child	an E element, first child of its parent	Structural pseudo-classes	2
E:last-child	an E element, last child of its parent	Structural pseudo-classes	3
E:first-of-type	an E element, first sibling of its type	Structural pseudo-classes	3
E:last-of-type	an E element, last sibling of its type	Structural pseudo-classes	3
E:only-child	an E element, only child of its parent	Structural pseudo-classes	3
E:only-of-type	an E element, only sibling of its type	Structural pseudo-classes	3
E:empty	an E element that has no children (including text nodes)	Structural pseudo-classes	3
E:link	an E element being the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited)	The link pseudo-classes	1
E:visited			
E:active	an E element during certain user actions	The user action pseudo-classes	1 and 2
E:hover			
E:focus			
E:target	an E element being the target of the referring URI	The target pseudo-class	3
E:lang(fr)	an element of type E in language "fr" (the document language specifies how language is determined)	The :lang() pseudo-class	2
E:enabled	a user interface element E which is enabled or disabled	The UI element states pseudo-classes	3
E:disabled			
E:checked	a user interface element E which is checked (for instance a radio-button or checkbox)	The UI element states pseudo-classes	3
E:first-line	the first formatted line of an E element	The :first-line pseudo-element	1
E:first-letter	the first formatted letter of an E element	The :first-letter pseudo-element	1
E:before	generated content before an E element	The :before pseudo-element	2
E:after	generated content after an E element	The :after pseudo-element	2
E.warning	an E element whose class is "warning" (the document language specifies how class is determined).	Class selectors	1
E#myid	an E element with ID equal to "myid".	ID selectors	1
E:not(s)	an E element that does not match simple selector s	Negation pseudo-class	3
E F	an F element descendant of an E element	Descendant combinator	1
E > F	an F element child of an E element	Child combinator	2
E + F	an F element immediately preceded by an E element	Adjacent sibling combinator	2
E ~ F	an F element preceded by an E element	General sibling combinator	3

No puede haber espacio antes del paréntesis

[Selectors Level 4](#)

[Selectores CSS Nivel 4 Para Tener En Cuenta](#)

Selectores: test

Test Selectores [Try CSS Selector](#)

Click a selector:

.intro
#Lastname
.intro, #Lastname
h1
h1, p
div p
div > p
ul + h3
ul ~ table
*
[id]
[id=my-Address]
[id\$=ess]
[id|=my]
[id^=L]
[title~=beautiful]
[id*=s]
:checked
:disabled
:enabled
:empty
:focus
p:first-child
p::first-letter
p::first-line
p:first-of-type
h1:hover
input:in-range
input:out-of-range
input:invalid
input:valid
p:lang(it)
p:last-child
p:last-of-type
tr:nth-child(even)

Selector:

p::first-letter

The first letter of all
<p> elements.

Result:

```
<div class="intro">
<p>My name is Donald <span id="Lastname">Duck.
</span> </p>

<p id="my-Address">I live in Duckburg </p>

<p>I have many friends: </p>
</div>

<ul id="Listfriends">
  <li>Goofy </li>
  <li>Mickey </li>
  <li>Daisy </li>
  <li>Pluto </li>
</ul>

<p>All my friends are great! <br>
But I really like Daisy! </p>

<p lang="it" title="Hello beautiful">Ciao bella </p>

<h3>We are all animals! </h3>

<p> <b>My latest discoveries have led me to
believe that we are all animals: </b> </p>

<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Type of Animal</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Mickey</td>
      <td>Mouse</td>
    </tr>
    <tr>
      <td>Goofey</td>
      <td>Dog</td>
    </tr>
    <tr>
      <td>Daisy</td>
      <td>Duck</td>
    </tr>
    <tr>
      <td>Pluto</td>
      <td>Dog</td>
    </tr>
  </tbody>
</table>
```

Test soporte selectores CSS

Home News January 20, 2021 · New feature: Cookie Store API Compare browsers About

Can I use ? Settings

50 results found
Canuse (8) MON (44)

CSS3 selectors REC

Advanced element selection using selectors including:
{foo~"bar"}, {foo~"bar"}, {foo~"bar"}, :root, :nth-child(), :nth-last-child(), :nth-of-type(), :nth-last-of-type(), :last-child, :first-of-type, :last-of-type, :only-child, :only-of-type, :empty, :target, :enabled, :disabled, :checked, :not(), :general-sibling

Current align Change relative Data relative Filtered All

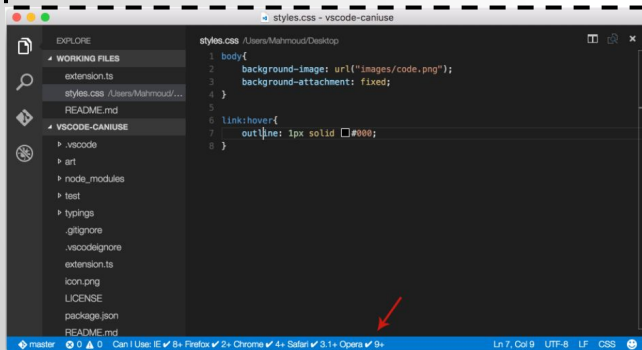
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaIOS Browser
6															
7-8		2-3		3.1											
9-10	12-86	3.5-43	4-86	3.2-13.1	10-71	3.2-13.7		2.1-4.4.4	12-12.1			4-12.0			
11	87	84	87	14	72	14.3	all	81	59	87	83	12.12	13.0	10.4	7.12
		85-86	88-90	TP											2.5

Notes Test on a real browser New Sub-features Known issues (6) Resources (5) Feedback

Can I Use
Selectors

<http://css3test.com/>

Can I Use en VSCode



<http://tools.css3.info/selectors-test/test.html>

From the 41 selectors 41 have passed, 0 are buggy and 0 are unsupported (Passed 574 out of 574 tests)



Selectors Level 3 TR DEV

100%

Selectors

- > Sibling combinator
- > ::before
- > ::after
- > ::first-letter
- > ::first-line
- > [att~val]
- > [att=val]
- > [att=val]
- > Namespaces
- > :target
- > :enabled
- > :disabled
- > :checked
- > :indeterminate
- > :root
- > :nth-child
- > :nth-last-child
- > :nth-of-type
- > :nth-last-of-type
- > :last-child
- > :only-child
- > :first-of-type
- > :last-of-type
- > :only-of-type
- > :empty
- > :not()

Selectors Level 4 TR DEV

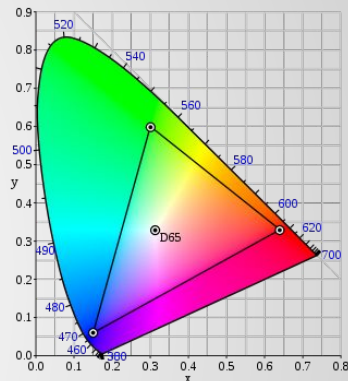
0%

Selectors

- > Descendant combinator

Cómo indicar color en CSS

- El tipo de datos CSS color denota un color en el *sRGB color space*. Un color puede ser descrito en cualquiera de las siguiente maneras:
 - Con una *palabra clave* (*red, blue, ...*). También *transparent*,
 - Usando el sistema RGB cubic-coordinate (usando #-hexadecimal o las notaciones funcionales `rgb()` y `rgba()`) → *rgba(0,0,0,0)* es *transparent*
 - Usando el sistema HSL cylindrical-coordinate, usando las notaciones funcionales `hsl()` y `hsla()`
- Asociado con el color en el espacio sRGB, un valor color también consiste de una **coordenada alpha-channel o valor de transparencia**, indicando cómo se debe mezclar el color con su color de fondo.
- Aunque los valores de colores en CSS son definidos de manera precisa **existe la posibilidad de que parezcan distintos en dispositivos diferentes** → *La mayoría de ellos no están calibrados y algunos navegadores no soportan todos los colores.*
- Color se utiliza en multitud de propiedades CSS: color, background-color, border-color,*
- El valor **currentcolor** identifica el valor de la propiedad color del elemento padre si el propio elemento no lo ha especificado. Útil en caso de que no se haya heredado.
- Especificación → [CSS Color Module Level 3](#)
- Manual que utilizaremos → [<color> - CSS | MDN](#) (ver apartados compatibilidad y especificaciones)
- [Ultimate Guide to Color in UX/UI Design | by Christine Vallaure](#)



Named	Numeric	Color name	Hex rgb	Decimal
		black	#000000	0,0,0
		silver	#C0C0C0	192,192,192
		gray	#808080	128,128,128
		white	#FFFFFF	255,255,255
		maroon	#800000	128,0,0
		red	#FF0000	255,0,0
		purple	#800080	128,0,128
		fuchsia	#FF00FF	255,0,255
		green	#008000	0,128,0
		lime	#00FF00	0,255,0
		olive	#808000	128,128,0
		yellow	#FFFF00	255,255,0
		navy	#000080	0,0,128
		blue	#0000FF	0,0,255
		teal	#008080	0,128,128
		aqua	#00FFFF	0,255,255

La recomendación WCAG 2.0 de la W3C aconseja que no se utilice color como el único medio para especificar una información, acción o resultado concreto. Algunos usuarios tienen problemas para diferenciar colores y existe la posibilidad de que la información transmitida no sea captada. Por supuesto esto no impide que se use el color, sólo su uso como único medio para describir alguna información.

No puede haber espacio antes del paréntesis

color: rgb (11,66,222);

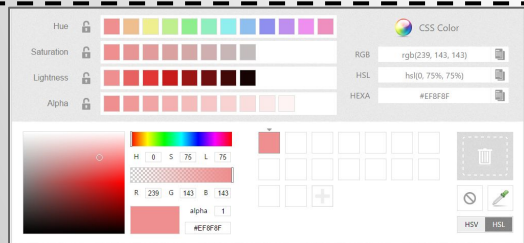
Herramientas online para seleccionar color

- Hay multitud de herramientas: incluidas en los propios IDE (Dreamweaver,...), online, ... para seleccionar color. → [Herramientas para elegir gamas de colores](#), [paletas de colores](#) y [Pantone](#)

[HSL Color Picker](#)



[Herramienta para seleccionar color - CSS | MDN](#)



Ejemplos

```
h2 { color: olive }

em { color: rgb(255,0,0) } /* integer range 0 - 255 */
em { color: rgb(300,0,0) } /* clipped to rgb(255,0,0) */
em { color: rgb(255,-10,0) } /* clipped to rgb(255,0,0) */
em { color: rgb(110%, 0%, 0%) } /* clipped to rgb(100%,0%,0%) */

em { color: rgb(255,0,0) } /* integer range 0 - 255 */
em { color: rgba(255,0,0,1) } /* the same, with explicit opacity of 1 */
em { color: rgb(100%,0%,0%) } /* float range 0.0% - 100.0% */
em { color: rgba(100%,0%,0%,1) } /* the same, with explicit opacity of 1 */
p { color: rgba(0,0,255,0.5) } /* semi-transparent solid blue */
p { color: rgba(100%, 50%, 0%, 0.1) } /* very transparent solid orange */
* { color: hsl(0, 100%, 50%) } /* red */
* { color: hsl(120, 100%, 50%) } /* lime */
* { color: hsl(120, 100%, 25%) } /* dark green */
* { color: hsl(120, 100%, 75%) } /* light green */
* { color: hsl(120, 75%, 75%) } /* pastel green, and so on */
```

Desde inspector de Chrome
podemos buscar un color



Propiedades básicas: color, background-color y opacity

Selector {color:valor1; background-color: valor2;}

- color → con esta **propiedad** indicamos el color del texto de un elemento. Se **hereda** por los descendientes. Se puede aplicar a todos los elementos.
- background-color → propiedad para indicar el color de fondo de un elemento. **No se hereda, aunque podemos hacer que herede utilizando el valor inherit** . Por defecto es transparente.
- opacity → valor entre 0 y 1 → nivel de transparencia.

background-color: color | transparent | inherit

HTML

```
1 <div class="exampleone">
2   Lorem ipsum dolor sit amet, consectetur
3 </div>
4
5 <div class="exampletwo">
6   Lorem ipsum dolor sit amet, consectetur
7 </div>
8
9 <div class="examplethree">
10  Lorem ipsum dolor sit amet, consectetur
11 </div>
```



CSS

```
1 .exampleone {
2   background-color: teal;
3   color: white;
4 }
5
6 .exampletwo {
7   background-color: rgb(153,102,153);
8   color: rgb(255,255,204);
9 }
10
11 .examplethree {
12   background-color: #777799;
13   color: #FFFFFF;
14 }
```



Result

Lorem ipsum dolor sit amet,
consectetur
Lorem ipsum dolor sit amet,
consectetur
Lorem ipsum dolor sit amet,
consectetur

Un primer ejemplo completo

1. Crea un fichero HTML con el contenido de la imagen de la izquierda
2. Crea un fichero CSS de nombre *estilos.css* con el contenido de la imagen de la derecha. → La imagen de la pluma y de la viñeta puede ser cualquiera, en el caso de la pluma ha de ser transparente.
3. Vincula el fichero HTML con el CSS y prueba su funcionamiento.

Ejercicio CSS 1.1 - Mi primera hoja de estilos

BIEN SEA PARA ILUSTRAR LA INFORMACIÓN CONTENIDA EN NUESTRAS PÁGINAS O PARA INTEGRAR UN LARGO TEXTO O, POR QUE NO, POR CUESTIONES ESTÉTICAS, ES PRÁCTICAMENTE IMPOSIBLE EVITAR COLOCAR ALGUNA QUE OTRA IMAGEN EN NUESTRAS PÁGINAS. AL EMPLEAR LAS IMÁGENES DEBEMOS GUIARNOS POR UNAS CUANTAS REGLAS BÁSICAS:

- i. Procurar que las imágenes sirvan para algo, que den información. Esta es una de las razones por las que debemos usar los atributos ALT.
- ii. No pasarnos. La página no debe pesar demasiado. Un visitante que ve como la página tarda en cargarse más de 15 o 20 seg lo más probable es que se vaya y, lo que es peor, que no vuelva.
- iii. Procurar un equilibrio entre tamaño del archivo de imagen y calidad de ésta. A veces una imagen muy grande requiere una compresión tal que al final tenemos una calidad pésima. En estos casos debemos plantearnos el tipo de imagen a usar.

Ejemplo de lista no ordenada:

-  Elemento 1
-  Elemento 2
-  Elemento 3

Texto obtenido de [HTTP://ESPACIOLATINO.COM/](http://espaciolatino.com/)

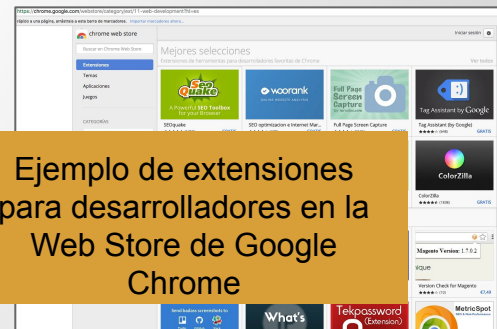
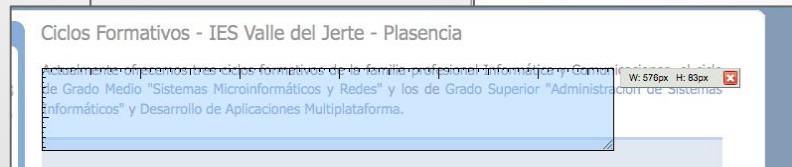
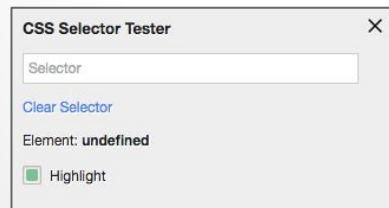
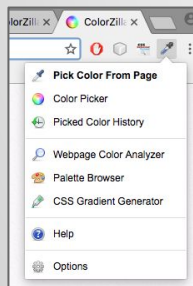
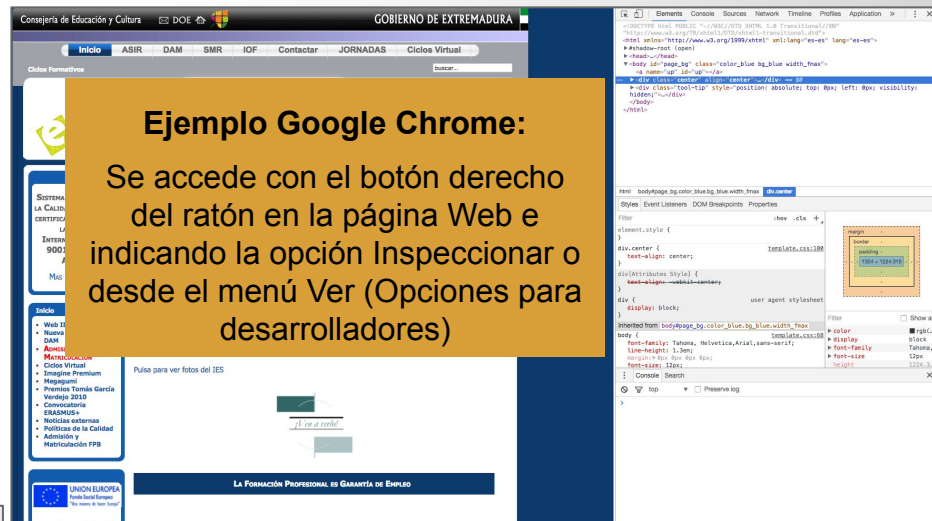
166,386 ptas son 1 €

Web espacio latino

```
#parrafo1 {
    font-size: 16px;
    line-height: 25px;
    font-weight: 500;
    font-variant: small-caps;
    color: #666666;
    background-image: url(../imagenes/pluma.gif);
    background-repeat: repeat;
    border: medium dotted #00FF00;
    padding: 20px 20px 20px 20px;
}
ol {
    list-style-position: inside;
    list-style-type: lower-roman;
}
ul {
    list-style-position: inside;
    list-style-image: url(../imagenes/viñeta.gif)
}
body {
    color: #CA7C35;
}
a:link {
    text-transform: lowercase;
}
a:hover {
    text-transform: uppercase;
}
```

Depuración básica de CSS

- La mayoría de los navegadores incorporan herramientas muy útiles para el desarrollador Web.
- También disponemos de extensiones → [measureIt](#), [colorzilla](#), tras para trabajar resoluciones (responsive), etc. → [30 Chrome extensions for web designers and devs | Creative Blog](#)
- Y herramientas online ← algunas las hemos visto en diapositivas anteriores
- [Firefox DevTools User Docs — Firefox Source Docs documentation](#)



Ejemplo de extensiones para desarrolladores en la Web Store de Google Chrome

Documentación (I)

<https://cssreference.io/>

background-color

Defines the color of the element's background.

`background-color: transparent;`

By default, the background color is **transparent**, basically meaning that there is **no** background color.

`background-color: red;`

You can use one of the **140+** color names.

`background-color: #95f7b0;`

You can use **hexadecimal** color codes.

`background-color: rgb(50, 115, 220);`

You can use **rgb()** color codes:

- the first value is for **red**
- the second value is for **green**
- the third value is for **blue**

Each of them can have a value between **0** and **255**.

`background-color: rgba(50, 115, 220, 0.3);`

You can use **rgba()** color codes:

in collection: Backgrounds

Hello World

Hello World

Hello World

Hello World

Hello World

LMSGI - Antonio Berrocal Piris

Collections: Animations Backgrounds Box model Flexbox **CSS Grid New!** Positioning Transitions Typography

Search 129 CSS properties

align-content	flexbox
align-items	flexbox
align-self	flexbox
animation-delay	animations
animation-direction	animations
animation-duration	animations
animation-fill-mode	animations
animation-iteration-count	animations
animation-name	animations
animation-play-state	animations
animation-timing-function	animations
animation	animations

<https://web.dev/learn/css/>

Documentación (y II)

- [GitHub - eludadev/css-docs:](https://eludadev/css-docs:)
[CSS Selectors, Flexbox, Grid, Box Model, visually explained.](#)
- [CSS en español - Lenguaje CSS](#)
- [The Ultimate Guide to CSS + Cheat Sheets | by Bradley Nice | Level Up! | Medium](#)
- [CSS | MDN](#)
- [DevDocs](#)

<https://www.arkaitzgarro.com/css3/index.html>

CSS3		
ARKAITZ GARRO		
CAPÍTULO 1 INTRODUCCIÓN	6.2 Degradados lineales	11.1 Propiedades 'animables'
1.1 Especificación oficial	6.3 Degradados radiales	11.2 Propiedad transition-property
1.2 Nuevas funcionalidades	6.4 Repetición de degradados	11.3 Propiedad transition-duration
CAPÍTULO 2 MÓDULO SELECTORS	7.1 Sombra en textos	11.4 Propiedad transition-timing-function
2.1 Prefijos específicos de navegadores	7.2 Fuentes web	11.5 Propiedad transition-delay
CAPÍTULO 3 MÓDULO BASIC USER INTERFACE	CAPÍTULO 7 MÓDULO TEXT	11.6 Propiedad transition
3.1 box-sizing	7.1 Sombra en textos	11.7 Listas de valores
3.2 overflow-x y overflow-y	7.2 Fuentes web	11.8 Finalización de una transición
CAPÍTULO 4 MÓDULO COLOR	CAPÍTULO 8 MÓDULO MULTI-COLUMN LAYOUT	11.9 Transiciones y JavaScript
4.1 Propiedad opacity	8.1 Propiedad shorthand	CAPÍTULO 12 MÓDULO ANIMATIONS
4.2 Función rgb()	8.2 Equilibrio de altura	12.1 Configuración
4.3 Función hsl()	8.3 Espacios entre columnas	12.2 Keyframes
4.4 Función hsla()	8.4 Trazos entre columnas	12.3 Ejemplos
CAPÍTULO 5 MÓDULO BACKGROUNDS AND BORDERS	8.5 Texto sobre columnas	CAPÍTULO 13 EJERCICIOS DE CSS 3
5.1 Bordes	CAPÍTULO 9 MÓDULO MEDIA QUERIES	13.1 Capítulo 2
5.2 Sombras	9.1 Sintaxis	13.2 Capítulo 5
5.3 Gradientes	9.2 Operadores lógicos o logical operators	13.3 Capítulo 7
	CAPÍTULO 10 MÓDULO TRANSFORMS	13.4 Capítulo 9
	10.1 Propiedades principales	13.5 Capítulo 10

Referencia CSS

Índice por palabra clave

A	font-stretch	padding-block-end
:active	font-stretch (@font-face)	padding-block-start
additive-symbols (@counter-style)	font-style	padding-bottom
::after (:after)	font-style (@font-face)	padding-inline-end
align-content	font-synthesis	padding-inline-start
align-items	font-variant	padding-left
align-self	font-variant (@font-face)	padding-right
all	font-variant-alternates	padding-top
<angle>	font-variant-caps	@page
animation	font-variant-east-asian	page-break-after
animation-delay	font-variant-ligatures	page-break-before
animation-direction	font-variant-numeric	page-break-inside
animation-duration	font-variant-position	pc
animation-fill-mode	font-weight	%percentage
animation-iteration-count	font-weight (@font-face)	perspective
animation-name	format()	perspective()
animation-play-state	format() (@font-face)	pointer-events
animation-timing-function	fr	polygon()
@annotation	<frequency>	<position>
annotation()	fullscreen	position
attr()		prefix (@counter-style)
	G	pt
	grad	px
B	<gradient>	
::backdrop	grayscale()	
backface-visibility		