

Introducción a CSS

DigitalHouse >
Coding School

Índice

1. [Vinculación con HTML](#)
2. [Las reglas de CSS](#)
3. [Los selectores de CSS](#)

1

Vinculación con HTML

CSS quiere decir...

Cascading Style Sheet

¿Para qué sirven las hojas de estilo?

Estas sirven **para estilizar nuestro contenido HTML**.

Con CSS podemos cambiar colores, fondos, tipografías, anchos, altos, y mucho más. Así como también generar animaciones y transiciones.

Contamos con 3 métodos para vincular nuestros archivos CSS con el documento HTML.



Vinculación externa



Esta es la más utilizada ya que podemos escribir todos nuestros estilos en un archivo CSS y vincularlos al **HTML** usando la etiqueta `<link>` dentro del `<head>` de nuestro documento.

Atributo + valor

Indica qué relación hay entre los documentos a enlazar. El valor siempre será el mismo.

html

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

Atributo + valor

Ruta de la ubicación de mi hoja de estilo.

Vinculación interna



A través de la etiqueta `<style>` dentro del `<head>`.

html

```
<head>
  <meta charset="UTF-8">
  <style>
    body {background: blue}
  </style>
</head>
```

Vinculación en línea



Usando el atributo `style` en cada elemento de nuestro HTML.

```
html
```

```
<p style="color: red">...</p>
```


2 | Las reglas de CSS

¿Qué son las reglas de CSS?

Una regla de CSS es un conjunto de instrucciones que se aplican a un elemento determinado para agregarle estilos.

Selector

Indica sobre qué elemento o elementos aplicaremos la regla.



```
css  body { font-family: sans-serif }
```



propiedad

Define qué propiedad del elemento o de los elementos vamos a modificar.



valor

Especifica el comportamiento de la propiedad.

¿Cómo se escriben?

Primero escribimos el o los selectores, seguidos de las llaves `{ }` que indican el comienzo de las reglas.

Cada regla se compone de una propiedad y su correspondiente valor, separados por dos puntos `:`.

Finalmente, podemos tener más de una regla por elemento y estas se separan con punto y coma `;`.

CSS

```
body {  
  background-color: purple;  
  font-family: sans-serif;  
  text-align: center;  
}
```

Las propiedades de CSS

Existen muchas propiedades de CSS que nos permiten manipular los elementos del HTML a nuestro antojo.

Si tuviéramos que agrupar las propiedades, los grupos serían más o menos así:

- Tipografías
- Fondos
- Tamaños
- Posicionamiento
- Visualización
- Comportamiento
- Interfaz
- Otros

3

Los selectores de CSS

Selectores de ID

Este selector va a atrapar al elemento **HTML** que tenga asignado el atributo `id` con el valor correspondiente.

Se recomienda usar nombres únicos para cada elemento y no repetirlos a lo largo del documento.

```
html    <h3 id="saludo">¡Hola!</h3>
```

Para llamarlo desde el CSS usamos el `#` seguido del **nombre del ID**.

```
css     #saludo { color: blue }
```

Selectores de clase

Este selector va a atrapar al elemento **HTML** que tenga asignado el atributo `class` con el valor correspondiente.

Podemos asignarle la cantidad de clases que queramos a un mismo elemento. Para hacerlo, solo hace falta separarlas con un espacio.

```
html    <h3 class="noticia destacada">Una noticia</h3>
```

Para llamarlo desde el CSS usamos el `.` seguido del **nombre de la clase**.

```
css     .noticia { font-size: 22px }
```

Selectores de etiqueta

Este selector va a atrapar al elemento **HTML** con el mismo nombre de etiqueta que llamemos desde nuestro **CSS**.

```
html    <p>Ad lorem ipsum dolor sit amet</p>
```

Para llamarlo desde el CSS usamos el **nombre de la etiqueta**.

```
css    p { color: gray }
```


Selectores combinados

Como su nombre lo indica, son la combinación de todos los anteriores, también podemos usar varios selectores del mismo tipo.

En este ejemplo combinamos un selector de etiqueta con uno de clase.

```
html    <h2 class="subtitulo">Un subtítulo</h2>
```

Para llamarlos desde el CSS solo hace falta agregar un selector al lado del otro, cada uno con la sintaxis que le corresponda. Recordemos que van sin espacios.

```
css     h2.subtitulo { color: yellow }
```

Selectores descendentes

Estos selectores sirven para agregar especificidad. En el ejemplo vamos a atrapar al elemento `li` que esté dentro del `ul` con el `id lista`.

Para llamarlos desde el CSS escribimos los selectores separados por un espacio (el de la derecha siempre será el que está dentro del de la izquierda).

html

```
<ul id="lista">  
  <li>Primer ítem</li>  
</ul>
```

css

```
ul#lista li { text-align: center }
```

“

El CSS **siempre va a priorizar** a los selectores más específicos para aplicar los estilos.

”



DigitalHouse>
Coding School

Propiedades tipográficas

Propiedades tipográficas

Existen muchas propiedades de CSS que nos permiten manipular los elementos del HTML a nuestro antojo.

Si tuviéramos que agrupar dichas propiedades, podríamos hacerlo así:

- **Tipografías**
- **Fondos**
- **Tamaños**
- **Posicionamiento**
- **Visualización**
- **Comportamiento**
- **Interfaz**
- **Otros**

font-family

Permite elegir la **familia tipográfica** que queremos usar. Como valor recibe el nombre de la tipografía que queramos usar.

Para que funcione, la tipografía debe existir en la computadora del usuario, o bien debemos usar una webfont.

Podemos poner más de una tipografía, separando las adicionales por comas. En caso de que la primera no esté disponible, se cargará la segunda y así sucesivamente.

```
css  p {  
      font-family: Arial, sans-serif;  
    }
```

font-size

Permite definir el **tamaño tipográfico**.

Recibe un valor numérico acompañado de la unidad de medida. Las unidades de medida más habituales suelen ser: px, em y rem.

css

```
p {  
  font-size: 23px;  
}
```


font-style

Define el **estilo** de la **tipografía**. Recibe los valores italic, normal y oblique.

Para algunos elementos, como ``, el valor por defecto será italic.

css

```
p {  
  font-style: normal;  
}
```

font-weight

Define el **peso** de la **tipografía**. Recibe los valores **bold**, lighter, normal, entre otros. También puede recibir un valor numérico que se irá incrementando de 100 en 100. Para algunos tags el valor por defecto será bold.

css

```
p {  
  font-weight: 500;  
}
```

text-align

Permite definir la **alineación** del texto. Los valores que recibe son center, left, right, inherit y justify. El valor por defecto para todos los elementos es left.

css

```
p {  
  text-align: justify;  
}
```

text-decoration

Permite elegir un tipo de decoración para el texto. Recibe los valores line-through, underline, overline y none.

Para algunos elementos, como los enlaces, el valor por defecto será underline.

css

```
p {  
  text-decoration: underline;  
}
```

line-height

Permite definir el alto de cada línea de textos. Esto también suele llamarse interlineado.

Recibe un valor numérico acompañado de la unidad de medida y, por lo general, está relacionado con el tamaño de la letra o **font-size**.

```
css  p {  
      line-height: 20px;  
    }
```

DigitalHouse>
Coding School

Colores

DigitalHouse >
Coding School

Formatos de color

Los siguientes formatos se pueden aplicar en cualquier propiedad de CSS que reciba color:

Nombre (podemos explorar el resto en [esta página](#))

purple

Hexadecimal

#f05331

RGB

rgb(255, 100, 50)

RGBA

rgba(122, 50, 125, **0.5**). El último número representa la opacidad que tendrá el elemento. Va del 0 al 1 y mientras menor el número, mayor la transparencia.

El color de texto

El atributo `color` nos permite asignarle un **color al texto de un elemento**.

Recibe como valor cualquiera de los formatos de color permitidos.

```
css  h2 { color: purple }
```

```
css  h2 { color: #3459ff }
```

```
css  p { color: rgb(12, 34, 32) }
```

El color de fondo

El atributo `background-color` nos permite asignarle un **color de fondo** a un elemento.

Recibe como valor cualquiera de los formatos de color permitidos.

```
css  p { background-color: tomato }
```

```
css  p { background-color: #3459ff }
```

```
css  p { background-color: rgb(12, 34, 32) }
```

La opacidad

Mediante el atributo `opacity` le otorgamos transparencia a todo el elemento.

```
css  p { opacity: 0.5 }
```



Valor

El valor representa el porcentaje de opacidad que le queremos dar al elemento.

Los valores van desde 0 a 1. Eso quiere decir que 0.5 representa una opacidad del 50%.

DigitalHouse>
Coding School

Fondos

DigitalHouse >
Coding School

Índice

1. [Colores de fondo](#)
2. [Imágenes de fondo](#)

1

Colores de fondo

background-color

Nos permite asignarle un **color de fondo** a un elemento.

Recibe como valor cualquiera de los formatos de color permitidos.

```
css  p { background-color: tomato }
```

```
css  p { background-color: #3459ff }
```

```
css  p { background-color: rgb(12, 34, 32) }
```

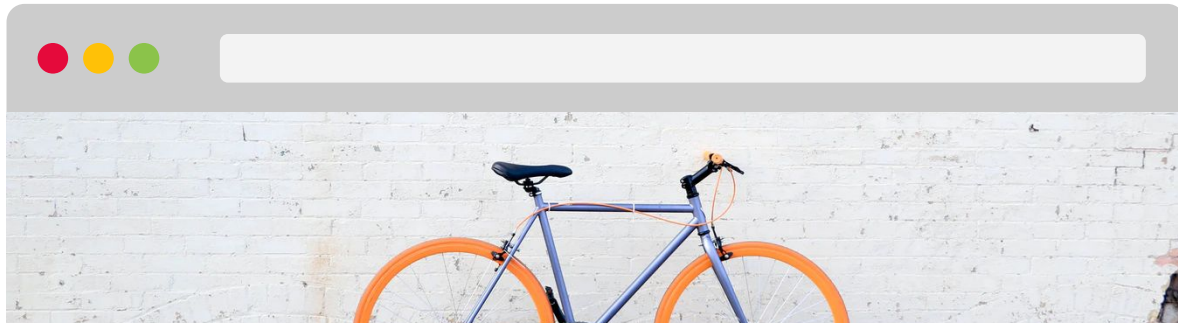

2 | Imágenes de fondo

background-image

Nos permite asignarle una **imagen de fondo** al elemento, definiendo la ruta a través de la URL.

css

```
body { background-image: url('../img/bici.jpg') }
```

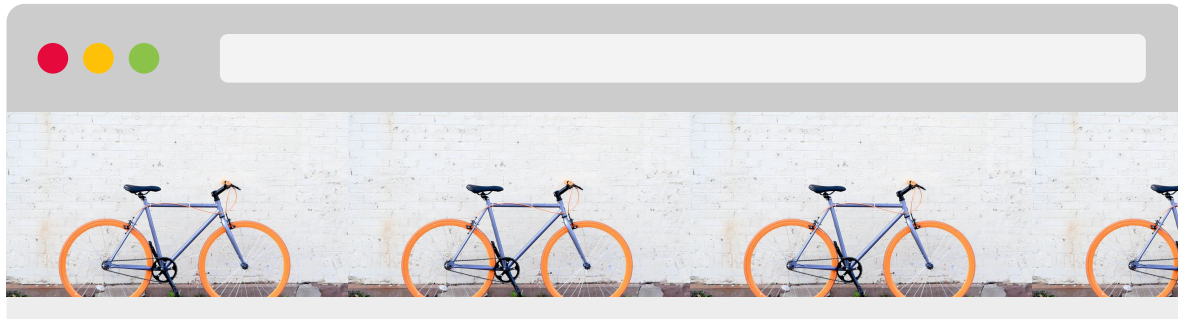


background-repeat

Nos permite controlar si se va a repetir, y de qué manera, la imagen dispuesta. Recibe los valores repeat, no repeat, repeat-x, repeat-y, round y space.

css

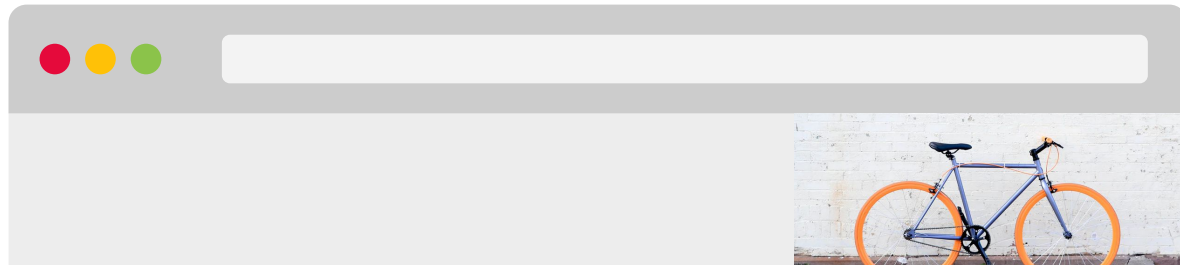
```
body { background-repeat: repeat-x }
```



background-position

Nos permite mover la imagen dentro del elemento y decidir dónde colocarla. Recibe como valores tamaños en píxeles y porcentajes, así como también right, bottom, left, etcétera. Podemos asignarle uno o dos valores. El primero para especificar la posición en el eje x y el segundo, la posición en el eje y.

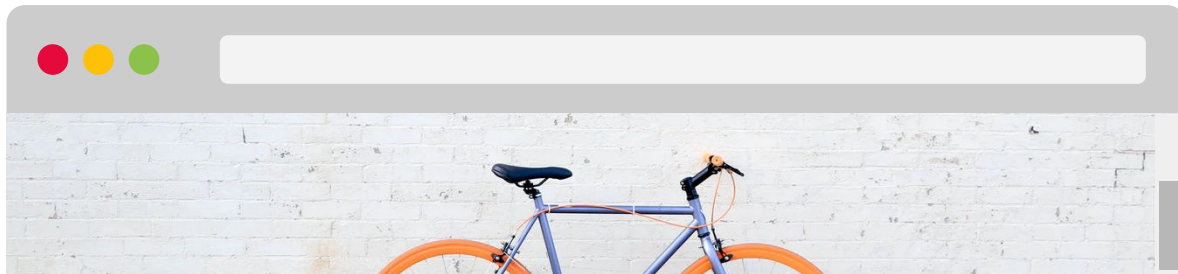
```
css  body { background-position: right top }
```



background-attachment

Nos permite establecer si la imagen de fondo se va a mover junto con la página al hacer scroll o si se va a quedar fija. Recibe como valor `fixed`, `scroll`, `inherit` e `initial`.

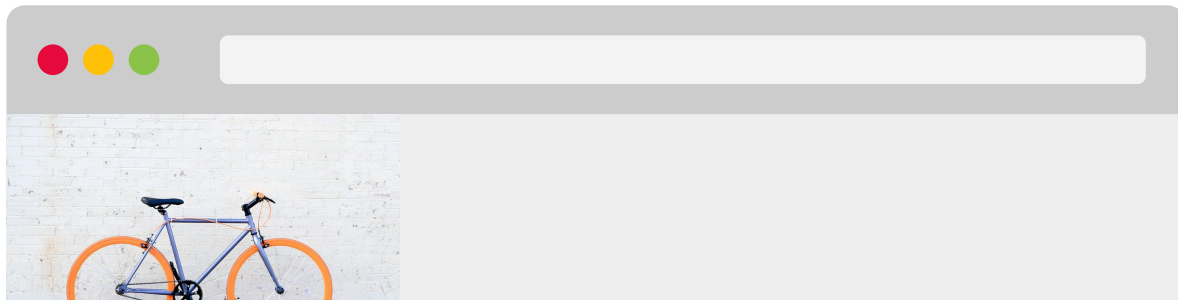
```
css  body { background-attachment: fixed }
```



background-size

Nos permite establecer el tamaño de la imagen de fondo. Recibe como valor contain, cover, inherit; así como también tamaños en pixeles y porcentajes, indicando con el primer valor el ancho y con el segundo el alto.

```
css  body { background-size: 130px }
```



DigitalHouse>
Coding School