

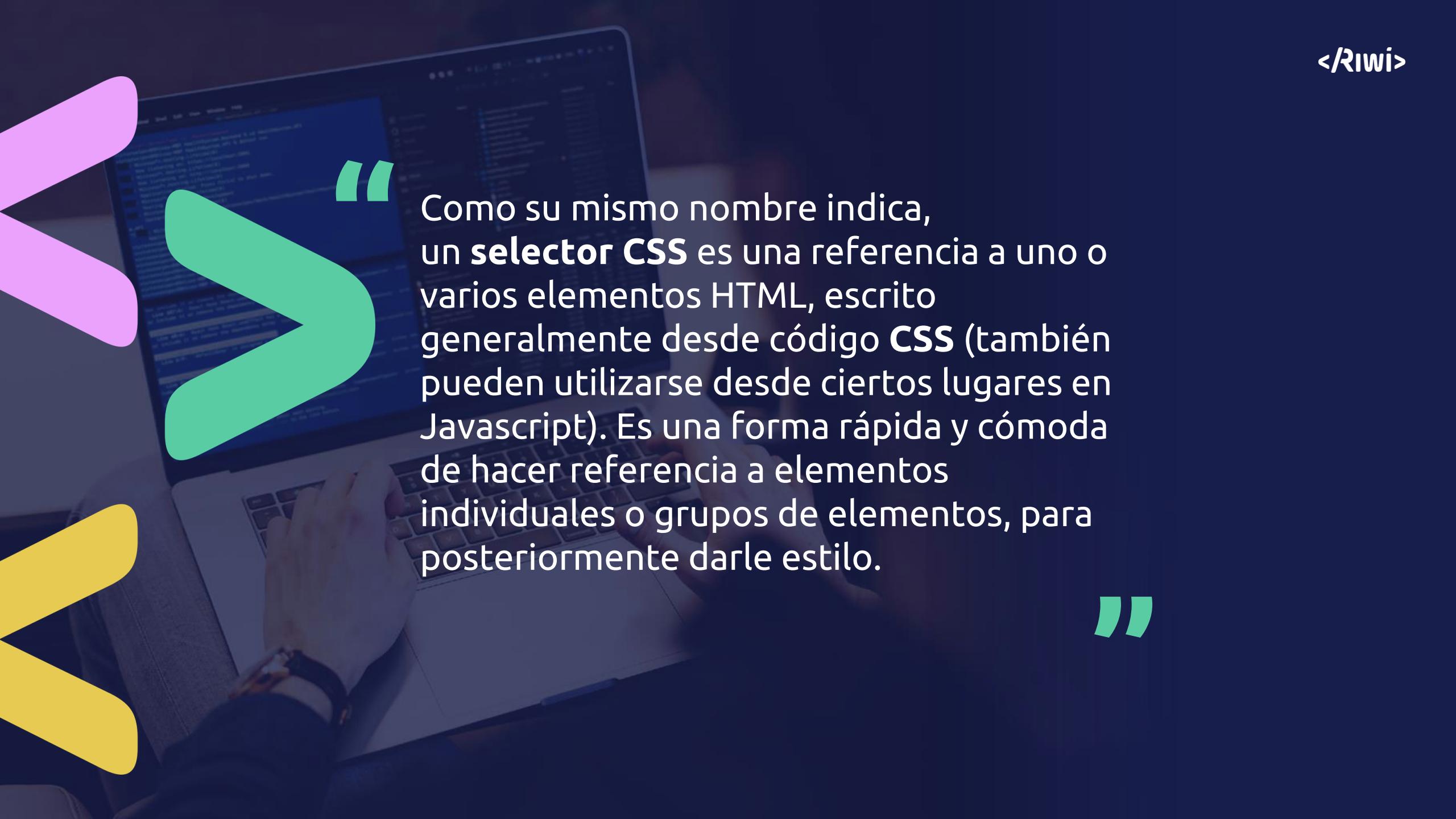


Somos un **ecosistema** de desarrolladores de software

Introducción a CSS (Cascading Style Sheets)



8"> <!--
ation">
 .html">Home
vents.html">Home Events
col-menu.html">Multiple Column Men
ren">
all-button-header.html">Tall But
age-logo.html">Image Logo</li
ve Tall-Logo.html">Ta
f="#">Carousels
le-width-slider.html">Variat
le-width-slider.html">Testimon



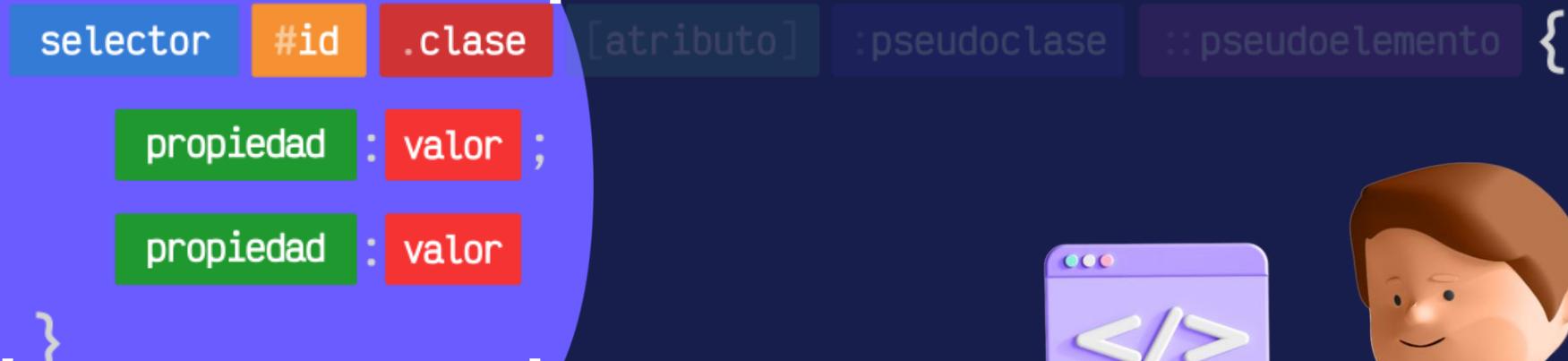
Como su mismo nombre indica, un **selector CSS** es una referencia a uno o varios elementos HTML, escrito generalmente desde código **CSS** (también pueden utilizarse desde ciertos lugares en Javascript). Es una forma rápida y cómoda de hacer referencia a elementos individuales o grupos de elementos, para posteriormente darle estilo.

”

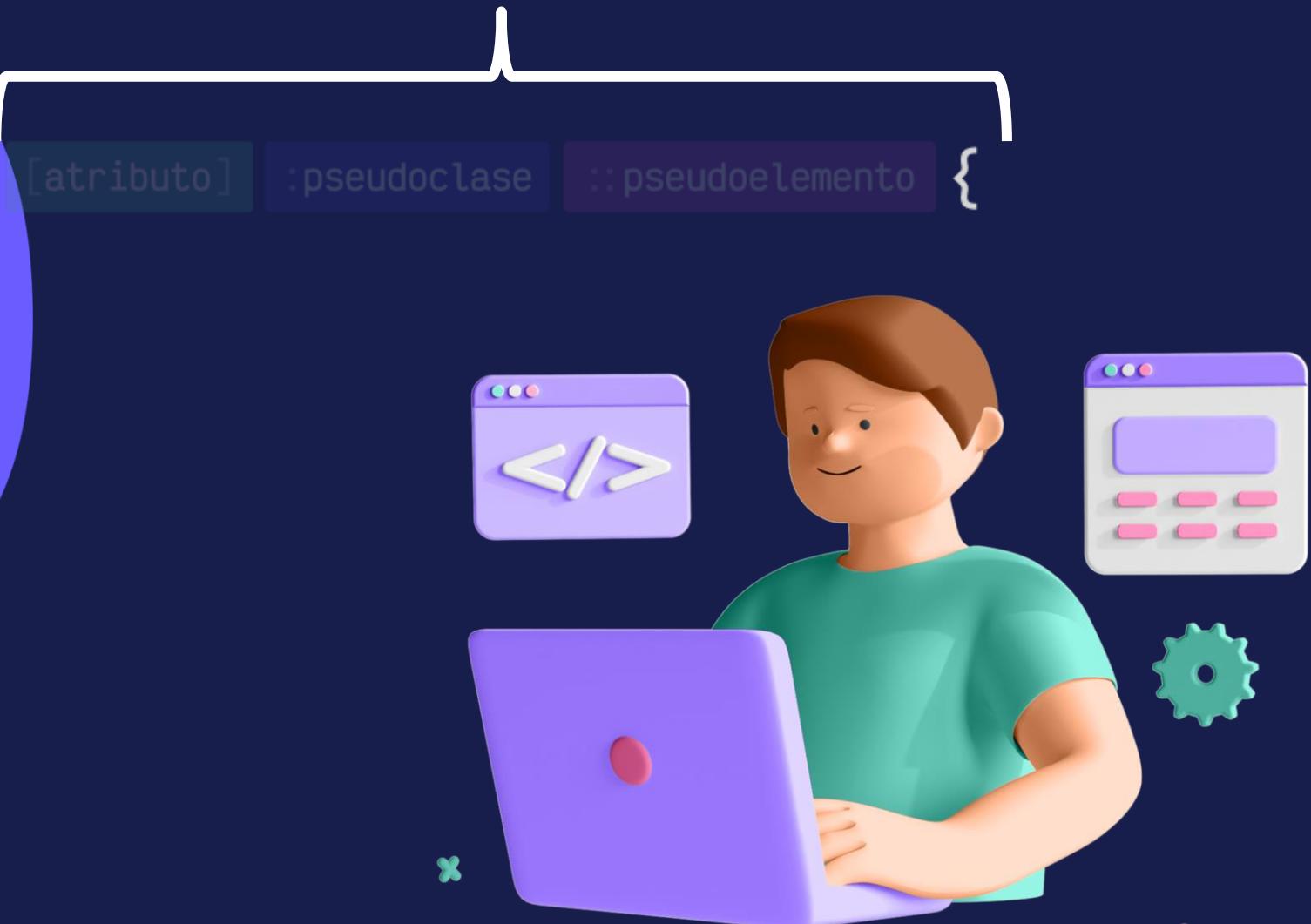
Selectores

Básicos

Avanzado



Básico



Selectores

Propiedades

Selector	Descripción
div	Seleccionamos por etiqueta. Aplicamos estilos a todas las etiquetas que coincidan.
#page	Seleccionamos por ID. Aplicamos estilos al elemento que tenga ese id. Debería ser un único elemento.
.primary	Seleccionamos por clase. Aplicamos estilos a los elementos que tengan dicha class. Pueden existir varias.
button.primary	Mixto. Seleccionamos por múltiples criterios. En este caso, por elemento y por clase.

Selectores

Seleccionar por etiquetas

```
div {  
| background-color: #blueviolet;  
|}  
|
```

Este es un título

Este es un párrafo de ejemplo.

hola como estas?

Aplicamos cambios a todos los elementos seleccionados en CSS.

```
strong {  
| color: #aqua;  
|}  
|
```

Selectores

Selección por ID (únicos)

¡Hola, visitante! ¡Bienvenido a esta página!

En esta página encontrarás los siguientes temas:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Documento de ejemplo</title>
    <link rel="stylesheet" type="text/css" href="estilos.css">
  </head>
  <body>
    <div id="saludo">
      <p>¡Hola, visitante! ¡Bienvenido a esta página!</p>
    </div>

    <div id="main">
      <p>En esta página encontrarás los siguientes temas:</p>
    </div>
  </body>
</html>
```

Selectores

Seleccionar por clases

```
<!DOCTYPE html>
<html>
<head>
    <title>Documento de ejemplo</title>
    <link rel="stylesheet" type="text/css" href="estilos.css">
</head>
<body>
    <div id="main">
        <p>Escoge uno de los siguientes botones:</p>

        <button class="classic">Opción 1</button>
        <button class="classic">Opción 2</button>
        <button class="classic">Opción 3</button>
        <button class="back">Volver</button>
    </div>
</body>
</html>
```

```
.classic {
    background-color: □ green;
    color: □ white;
}
```

Comment Code

```
.back {
    background-color: □ orange;
    color: □ white;
}
```

Selectores

Selección mixta

```
<button class="classic primary large">INICIO</button>
```

En este ejemplo, sólo se aplicarán los estilos cuando el elemento HTML cumpla las siguientes condiciones (observa que no existen espacios entre las clases classic, primary y large):

- Sea un elemento HTML `<button>`
- Tenga una clase `.classic`
- Tenga una clase `.primary`
- Tenga una clase `.large`

Se debe cumplir todas y cada una de ellas. Si falta alguna, no se seleccionará el elemento y no se aplicará el estilo.

```
button.classic.primary.large {  
  
    background-color: #aqua;  
    color: black  
}
```

Selectores

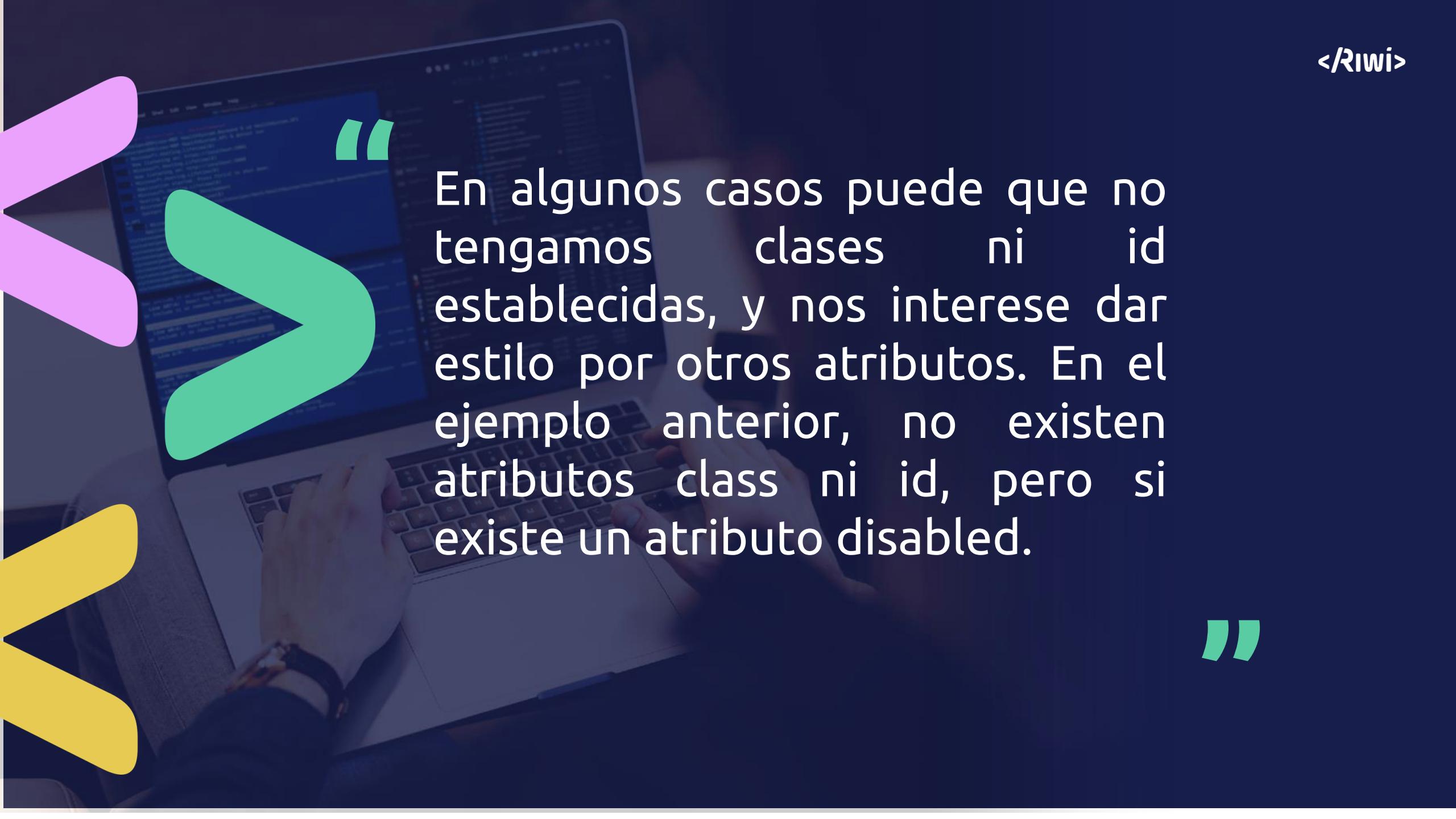
Atributos en CSS

```
<button>Botón activo</button>
<button disabled>Botón desactivado</button>
```

Botón activo

Botón desactivado

```
button {
    background-color: #FF5733; /* Color de fondo para todos los botones */
    color: white; /* Color de texto para todos los botones */
}
Comment Code
button[disabled] {
    background-color: #c36666; /* Color de fondo para botones con atributo disabled */
    color: #ff0000; /* Color de texto para botones con atributo disabled */
}
Comment Code
[disabled] {
    opacity: 0.5; /* Cambiar la opacidad para cualquier elemento con atributo disabled */
}
```



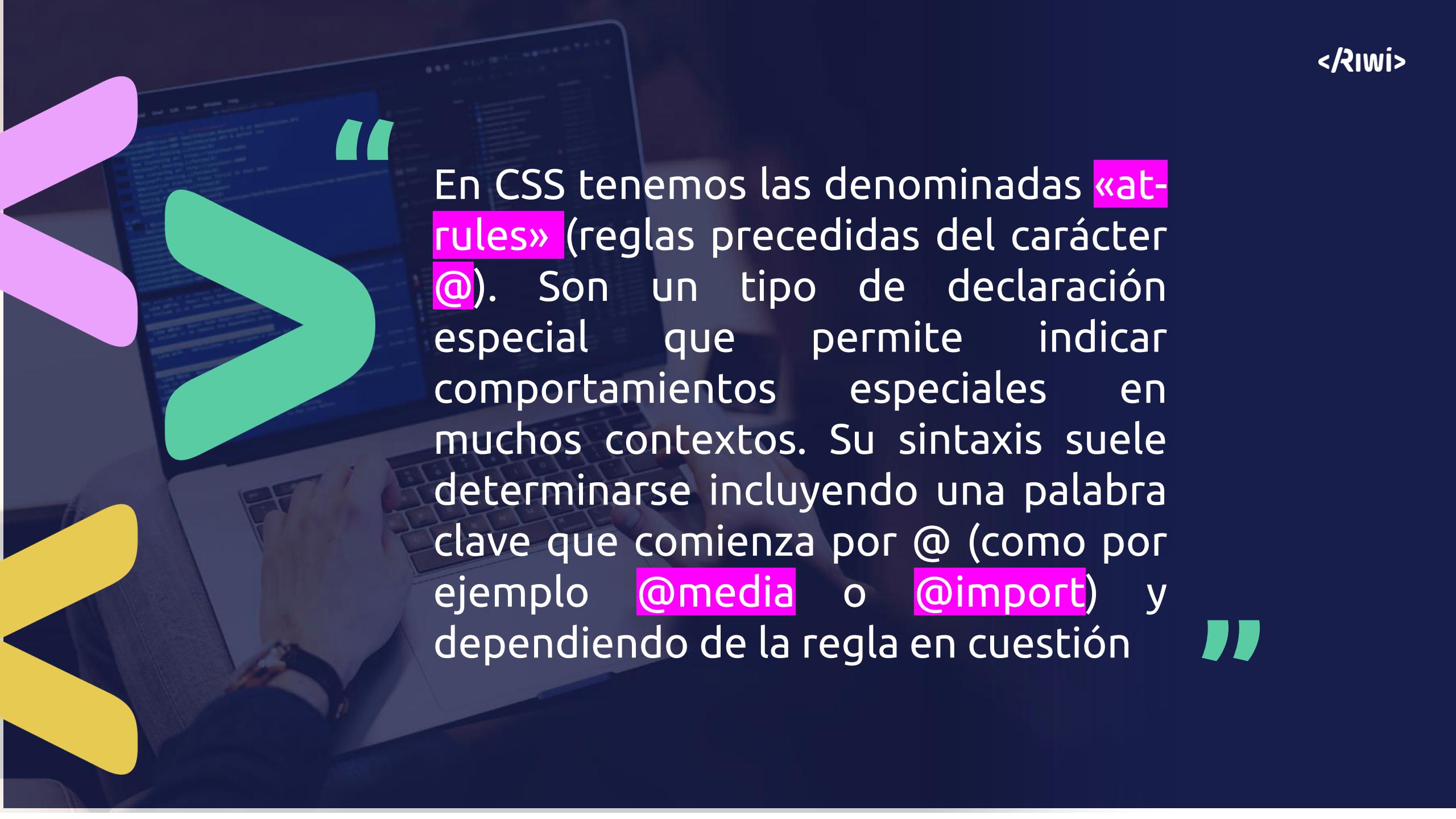
“En algunos casos puede que no tengamos clases ni id establecidas, y nos interese dar estilo por otros atributos. En el ejemplo anterior, no existen atributos class ni id, pero si existe un atributo disabled.

”

Selectores

Atributos

Atributo	¿Cuándo se aplica el estilo al elemento?
Básicos	
[href]	Si tiene un atributo href.
[href="#top"]	Si tiene un atributo href y su valor es #top.
[class~= "manzdev"]	Si tiene un atributo class con una lista de valores y uno de ellos es manzdev.
[lang ="es"]	Si tiene un atributo lang con una lista de valores, donde uno empieza por es-.
Avanzados	
[href^="https://"]	Si tiene un atributo href y su valor comienza por https://.
[href\$=".pdf"]	Si tiene un atributo href y su valor termina por .pdf (un enlace a un PDF).
[href*="manzdev"]	Si tiene un atributo href y su valor contiene manzdev.



“En CSS tenemos las denominadas «at-rules» (reglas precedidas del carácter @). Son un tipo de declaración especial que permite indicar comportamientos especiales en muchos contextos. Su sintaxis suele determinarse incluyendo una palabra clave que comienza por @ (como por ejemplo @media o @import) y dependiendo de la regla en cuestión”

Reglas

```
/* Regla @import para importar estilos CSS de otro fichero */
@import url("index.css");

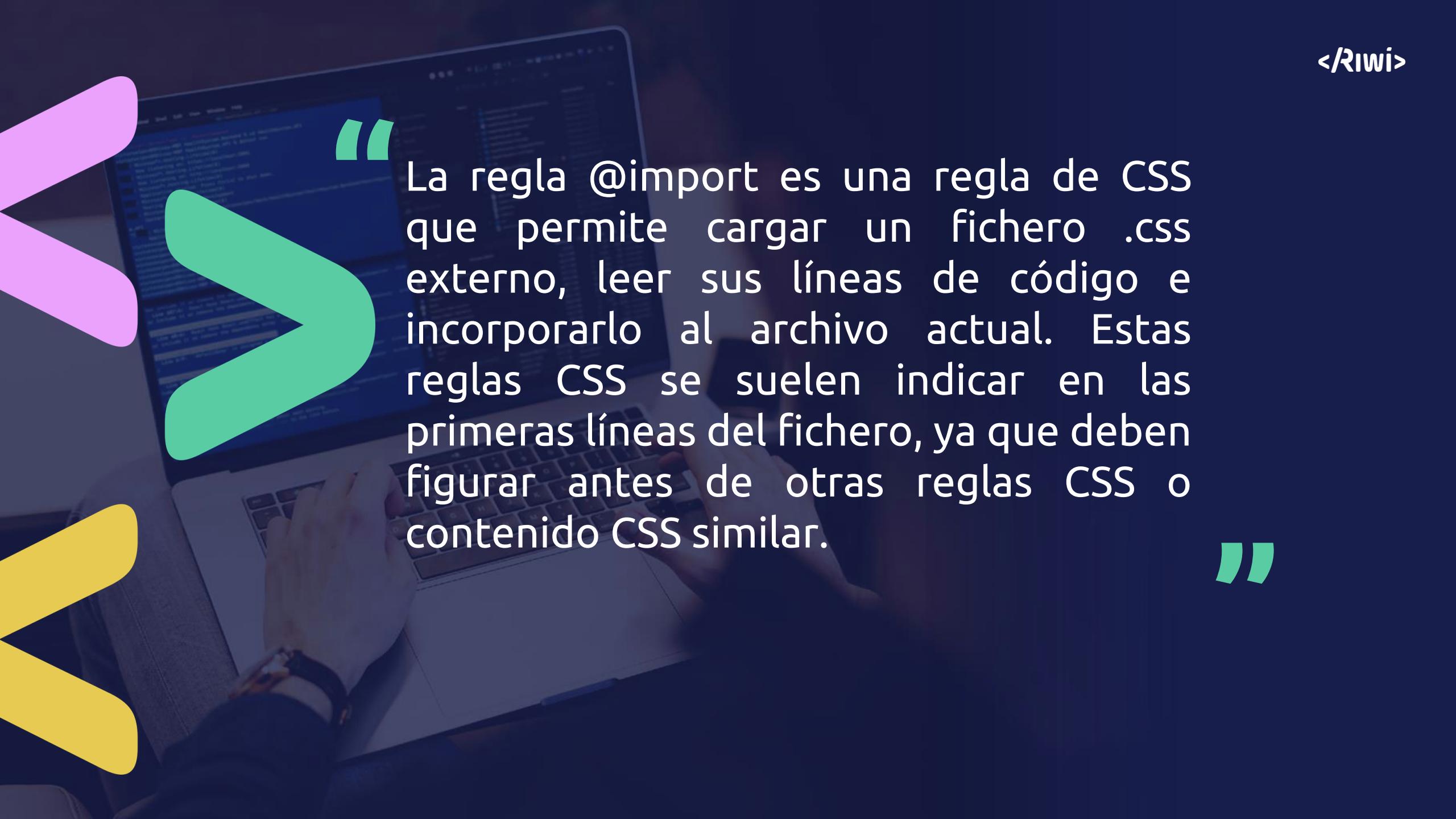
/* Media query para definir estilos de móvil */
@media screen and (max-width: 640px) {
    .page {
        width: 100%;
    }
}
```

En general, se pueden colocar en cualquier parte de la hoja de estilos, tanto al principio, al medio como al final. Sin embargo, algunas reglas tienen ciertas restricciones. Por ejemplo, como comentaremos, en el caso de la regla `@import` debe colocarse en las primeras líneas del fichero.

Reglas

</Riwí>

Regla	Descripción	Enlace
@import	Incluye una hoja de estilos externa, indicando su URL.	Ver la regla @import
@charset	Indicaba la codificación de caracteres usada en el fichero .css actual.	.
@supports	Establece un bloque de estilos que se aplicará si se cumple la condición.	Ver la regla @supports
@media	Aplica los estilos si cumple las condiciones indicadas. Usado para responsive.	Ver Media Queries
@keyframes	Crea los fotogramas clave de una animación.	Ver regla @keyframes
@scope	Aplica estilos con un límite de ámbito en elementos descendientes.	Ver regla @scope
@font-face	Indica una tipografía externa que el navegador debe descargar.	Ver tipografías externas
@font-feature-values	Activa/desactiva características especiales en una tipografía	Ver caract. personalizadas
@when / @else	Establece unos estilos si se cumple la condición y otros si no se cumple.	
@page	Modifica propiedades relacionadas con una página impresa.	Ver Medios paginados
@counter-style	Indica el estilo que se utilizará en los contadores CSS.	Ver Contadores CSS
@property	Indica el tipo de dato de una variable CSS, aplicando ciertas ventajas.	Ver regla @property
@layer	Establece que estilos se incluirán en una capa (que se fusionará más tarde).	Ver la regla @layer



“ La regla @import es una regla de CSS que permite cargar un fichero .css externo, leer sus líneas de código e incorporarlo al archivo actual. Estas reglas CSS se suelen indicar en las primeras líneas del fichero, ya que deben figurar antes de otras reglas CSS o contenido CSS similar. ”

Reglas

@font-feature-values

Regla	Descripción	Enlace
@import	Incluye una hoja de estilos externa, indicando su URL.	Ver la regla @import
@charset	Indicaba la codificación de caracteres usada en el fichero .css actual.	
@supports	Establece un bloque de estilos que se aplicará si se cumple la condición.	Ver la regla @supports
@media	Aplica los estilos si cumple las condiciones indicadas. Usado para responsive.	Ver Media Queries
@keyframes	Crea los fotogramas clave de una animación.	Ver regla @keyframes
@scope	Aplica estilos con un límite de ámbito en elementos descendientes.	Ver regla @scope
@font-face	Indica una tipografía externa que el navegador debe descargar.	Ver tipografías externas
@font-feature-values	Activa/desactiva características especiales en una tipografía	Ver caract. personalizadas
@when / @else	Establece unos estilos si se cumple la condición y otros si no se cumple.	
@page	Modifica propiedades relacionadas con una página impresa.	Ver Medios paginados
@counter-style	Indica el estilo que se utilizará en los contadores CSS.	Ver Contadores CSS
@property	Indica el tipo de dato de una variable CSS, aplicando ciertas ventajas.	Ver regla @property
@layer	Establece que estilos se incluirán en una capa (que se fusionará más tarde).	Ver la regla @layer

Reglas

@import

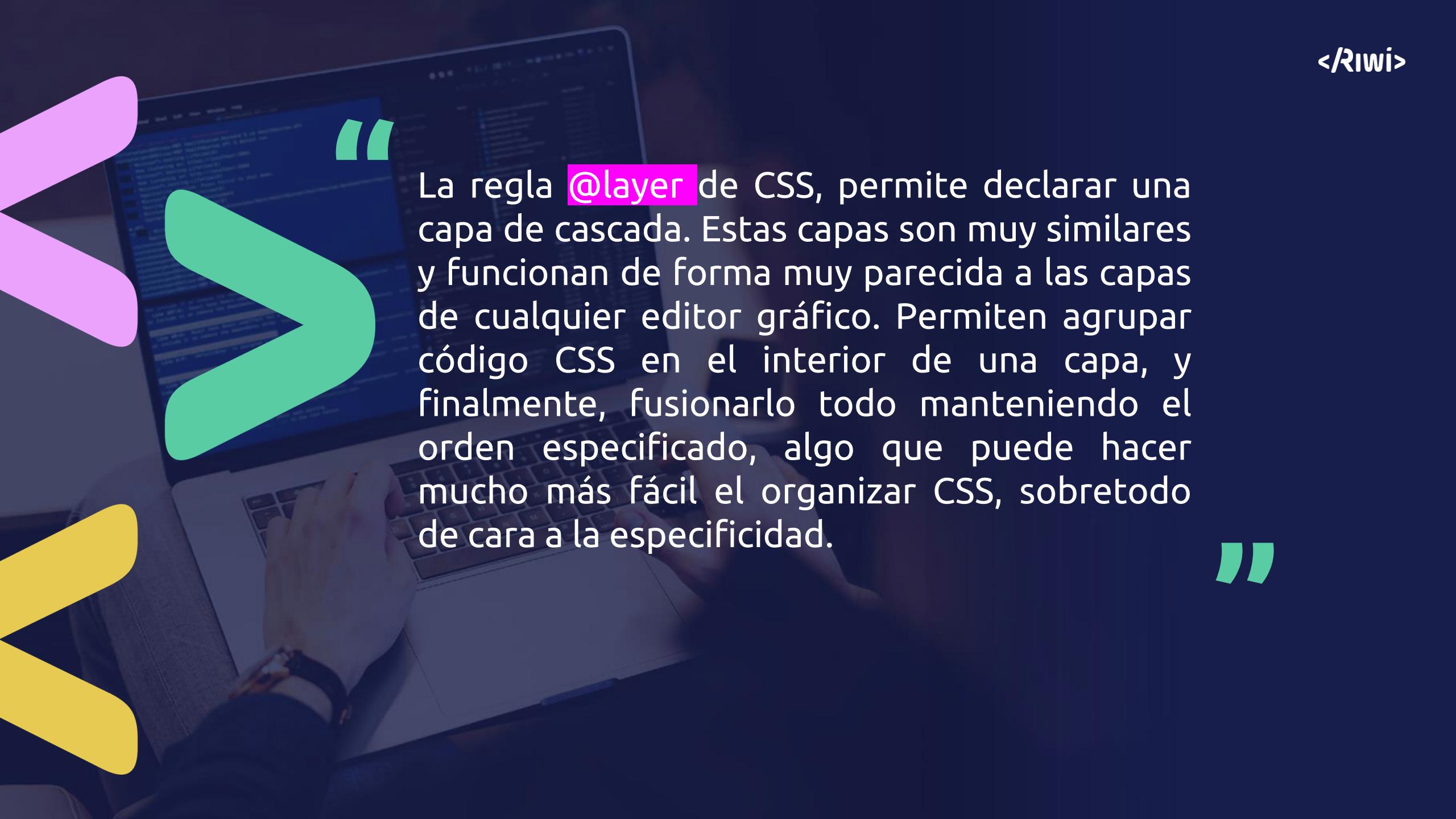
Tipo de importación	¿Qué importamos?	Soporte
@import	Una hoja de estilos CSS externa.	<input checked="" type="checkbox"/> Buen soporte.
@import media query	Una hoja de estilos CSS si coincide con el media query indicado.	<input checked="" type="checkbox"/> Buen soporte.
@import supports(condición)	Una hoja de estilos CSS sólo si el navegador soporta la condición.	 Cuidado.
@import layer(nombre)	Una hoja de estilos CSS y la colocamos en la capa nombre.	<input checked="" type="checkbox"/> Buen soporte.
@import layer()	Una hoja de estilos CSS y la colocamos en una nueva capa anónima.	<input checked="" type="checkbox"/> Buen soporte.

Reglas

Formato @import

Formato	Descripción
@import url("fichero.css")	Importa el fichero.css utilizando la función url() de CSS.
@import "fichero.css"	Importa el fichero.css utilizando un STRING

```
1  @import url("fineprint.css") print;
2  @import url("bluish.css") projection, tv;
3  @import 'custom.css';
4  @import "common.css" screen, projection;
5 + @import url('landscape.css') screen and (orientation:landscape);
```



“ La regla `@layer` de CSS, permite declarar una capa de cascada. Estas capas son muy similares y funcionan de forma muy parecida a las capas de cualquier editor gráfico. Permiten agrupar código CSS en el interior de una capa, y finalmente, fusionarlo todo manteniendo el orden especificado, algo que puede hacer mucho más fácil el organizar CSS, sobretodo de cara a la especificidad. ”

Reglas

@layer

```
@layer reset {  
    /* Comment Code */  
    body {  
        margin: 0;  
        box-sizing: border-box;  
    }  
}
```

Formato	Descripción
@layer	Crea una capa CSS anónima (sin nombre).
@layer nombre	Crea una capa CSS con el nombre indicado.
@layer nombre.subcapa	Crea una capa CSS con nombre, con una subcapa anidada subcapa.
@layer nombre1, nombre2, nombre3...	Declara y establece un orden para varias capas CSS.
@import(fichero.css) layer(nombre)	Importa un archivo CSS y lo introduce en la capa indicada.

Reglas

@layer

Utilizando la regla @layer podemos crear una capa CSS con un nombre determinado

```
@layer utils {  
    Comment Code  
    .main {  
        background: #34a;  
        border: 2px outset #6381db;  
        color: #fff;  
        padding: 5px 10px;  
        border-radius: 6px;  
    }  
}
```

Reglas

Orden de las capas

Podemos cambiar el orden de las capas, si establecemos una regla @layer con las diferentes capas separadas por coma.

```
@layer reset, texts, theme;

@layer reset {
    Comment Code
    button {
        padding: 30px;
    }
}

@layer theme {
    Comment Code
    .primary {
        background: □#34a;
        border: 2px outset □#6381db;
        color: ■#fff;
        padding: 5px 10px;
        border-radius: 6px;
    }
}

@layer texts {
    Comment Code
    .primary {
        color: ■red;
    }
}
```

Reglas

Capas anidadas

Dentro de las capas de CSS, también es posible crear capas dentro de otras.

```
@layer base {  
    @layer reset {  
        Comment Code  
        body {  
            margin: 0;  
            box-sizing: border-box;  
        }  
    }  
}
```

</Be a
coder>