

{css}



Clase 32

Clase 32

<índice>

Variables en CSS

¿Qué son?

Declaración y uso

Scope/ámbito de variables

Ventajas

Caso práctico

¿Qué son?

¿Qué son?

Custom Properties

Las variables CSS, también conocidas como propiedades personalizadas, son entidades definidas por los desarrolladores que contienen valores específicos que se pueden reutilizar en todo el documento CSS

¿Qué son?

Hacen que la hoja de estilos sea más mantenible y los valores fácilmente cambiados en un solo lugar. Estas variables pueden almacenar colores, fuentes, tamaños o cualquier otro valor de CSS que deseas utilizar en varios lugares de tu hoja de estilos

El objetivo principal suele ser evitar escribir múltiples veces ese valor, y en su lugar, ponerle un nombre más lógico y fácil de recordar, que hará referencia al valor real.

Ejemplos de uso:
Tamaños, tipografías, márgenes, paddings etc...

Definición y uso

Definición y uso

Para definir una custom property haremos uso de los dos guiones -- previos al nombre que queramos utilizar. Además, debemos fijarnos en el elemento que definimos la variable, en este ejemplo la pseudoclase :root:

Definición y uso

```
:root {  
  --background-color: black;  
}
```

Definición y uso

La pseudoclase :root hace referencia al elemento raíz del documento, o lo que es lo mismo, al elemento <html>. La diferencia de utilizar html o :root como selector es que este último tiene algo más de especificidad CSS. Mientras que html tiene 001, :root tendría 010.

Definición y uso

Al colocarla en :root estamos definiendo que la custom property estará definida para el ámbito de esa etiqueta <html> (o cualquier elemento hijo), es decir, a todo el documento. Sin embargo, ya veremos que podemos aplicar estas variables sólo a partes concretas del DOM de nuestra página.

Definición y uso

Uso de variables

Definición y uso

```
:root {  
  --background-color: indigo;  
}  
  
body {  
  background: var(--background-color);  
}
```

Definición y uso

En este caso estamos aplicando a la propiedad background el valor que contiene --background-color para el elemento <body>. Esto último es muy importante entenderlo, ya que una custom property puede tener diferentes valores dependiendo del ámbito en el que se encuentra.

Definición y uso

Fallback

Además, es muy recomendable que la expresión var() tenga dos parámetros. El primero de ellos, la custom property en cuestión, el segundo de ellos, el valor por defecto en el caso de que esa propiedad no esté definida en el ámbito actual:

Definición y uso

```
body {  
  background: var(--background-color, blue);  
}
```

Definición y uso

En este caso, la variable --background-color no está definida, por lo que el <body> tendrá siempre el color de fondo azul, ya que es el fallback indicado.

Definición y uso

Demo fallback

Scope / ámbito de variables

Scope / ámbito de variables

El ejemplo anterior es muy básico y puede que no se aprecie el detalle de los ámbitos con las custom properties, así que vamos a verlo con un ejemplo más específico.

Scope / ámbito de variables

```
.parent {  
  --background-color: black;  
  
  color: white;  
}  
  
.first {  
  --background-color: purple;  
}  
  
.child {  
  background: var(--background-color, blue);  
}
```

```
<div class="parent">  
  <div class="first child">First child</div>  
  <div class="second child">Second child</div>  
</div>  
<div class="third child">Third child</div>
```

Scope / ámbito de variables

- Los dos primeros elementos .child tomarán color negro, ya que se le aplica a .parent (e hijos).
- El primer elemento .child se sobreescribe con color púrpura, ya que se le aplica a .first.
- El tercer elemento no tendrá ninguna variable definida, por lo que tomará color azul.

Scope / ámbito de variables

Esto nos permite mucha flexibilidad y potencia a la hora de utilizar custom properties en CSS, ya que son tremadamente útiles y versátiles, permitiéndonos utilizar incluso la cascada de CSS a nuestro favor.

Scope / ámbito de variables

```
:root { --color: blue; }
div { --color: green; }
#alert { --color: red; }
* { color: var(--color); }

<p>I inherited blue from the root element!</p>
<div>I got green set directly on me!</div>
<div id='alert'>
  While I got red set directly on me!
  <p>I'm red too, because of inheritance!</p>
</div>
```

Scope / ámbito de variables

DEMO

Ventajas y usos

Ventajas y usos

Usos típicos

Ventajas y usos

Tamaños de fuente

Puedes establecer variables para diferentes tamaños de fuente que quieras utilizar en todo tu sitio. Por ejemplo, puedes tener variables para tamaños de fuente para encabezados, texto del cuerpo, subtítulos, etc.

```
:root {  
    --font-size-body: 16px;  
    --font-size-heading: 32px;  
    --font-size-subheading: 24px;  
}
```

Ventajas y usos

Familias de Fuentes

Define variables para las familias de fuentes para mantener la consistencia y facilitar los cambios de las fuentes a través del proyecto.

```
:root {  
    --font-family-primary: 'Open Sans', sans-serif;  
    --font-family-secondary: 'Roboto', sans-serif;  
}
```

Ventajas y usos

Peso de Fuentes

Las variables pueden contener los diferentes pesos de fuente que tu diseño requiere, permitiendo un cambio sencillo y coherente en los pesos de fuente en todo el sitio.

```
:root {  
    --font-weight-normal: 400;  
    --font-weight-bold: 700;  
}
```

Ventajas y usos

Altura de línea

Para mantener una legibilidad y espaciado consistentes, puedes usar variables para la altura de línea de los distintos elementos de texto.

```
:root {  
    --line-height-body: 1.5;  
    --line-height-heading: 1.2;  
}
```

Ventajas y usos

Color de texto

Las variables son muy útiles para definir los colores de texto que se usarán en todo el sitio web, facilitando la implementación de un esquema de color consistente.

```
:root {  
    --text-color-primary: #333;  
    --text-color-secondary: #555;  
}
```

Ventajas y usos

```
body {  
    font-family: var(--font-family-primary);  
    font-size: var(--font-size-body);  
    line-height: var(--line-height-body);  
    color: var(--text-color-primary);  
}  
  
h1 {  
    font-size: var(--font-size-heading);  
    line-height: var(--line-height-heading);  
    font-weight: var(--font-weight-bold);  
}
```

Ventajas y usos

Conclusión

Usar variables para la tipografía no solo mejora la coherencia en el diseño de tu sitio web, sino que también hace que sea mucho más fácil hacer ajustes globales a tu tipografía en el futuro.

Ventajas y usos

Ventajas

Ventajas y usos

Tematización

Puedes definir un conjunto de variables para colores, fuentes y otros valores estéticos que hacen un tema. Esto facilita el cambio de tema de un sitio web solo con cambiar los valores de estas variables.

Ventajas y usos

Consistencia de Diseño

Utilizar variables para espaciados (márgenes y paddings), tamaños de fuente, y medidas de elementos garantiza que tu diseño mantenga consistencia en todo el sitio web.

Ventajas y usos

Reutilización de código

Al usar variables, reduces la repetición de valores específicos en tu CSS, lo que hace que tu código sea más limpio y fácil de mantener.

Ventajas y usos

Facilidad de mantenimiento

Cuando necesitas actualizar un valor, como un color principal o un tamaño de fuente, puedes hacerlo en un solo lugar, y el cambio se aplicará en todo tu CSS donde se use esa variable.

Ventajas y usos

Adaptabilidad a Cambios de Contexto

Puedes cambiar el valor de las variables en diferentes contextos usando media queries, lo cual es útil para diseños responsives

Ventajas y usos

Interactividad con JS

Las variables CSS pueden ser leídas y modificadas con JavaScript, lo que te permite cambiar estilos dinámicamente en respuesta a la interacción del usuario.

Ventajas y usos

Mejora la legibilidad

Al nombrar las variables de forma descriptiva, haces que tu código sea más fácil de entender para otros desarrolladores (o para ti mismo en el futuro).

Ventajas y usos

Mejora la legibilidad

Al nombrar las variables de forma descriptiva, haces que tu código sea más fácil de entender para otros desarrolladores (o para ti mismo en el futuro).

Ventajas y usos

Las variables CSS hacen que trabajar con hojas de estilo sea más parecido a programar, con beneficios similares en cuanto a la abstracción y reutilización de código. Su uso se ha vuelto una práctica estándar en el desarrollo front-end moderno.

Caso práctico

Terry's Taco Joint

Pretty Good Tacos!

\$1.99

Tacos

\$3.99

Kombucha

TACOS

BEERS

WINES

DESSERTS

RESERVATIONS



Tacos

Lore ipsum dolor sit amet consectetur
adipisicing elit. Animi, assumenda.



Beer

Lore ipsum dolor sit amet consectetur
adipisicing elit. Animi, assumenda.



Wine

Lore ipsum dolor sit amet consectetur
adipisicing elit. Animi, assumenda.



Music

Lore ipsum dolor sit amet consectetur
adipisicing elit. Animi, assumenda.



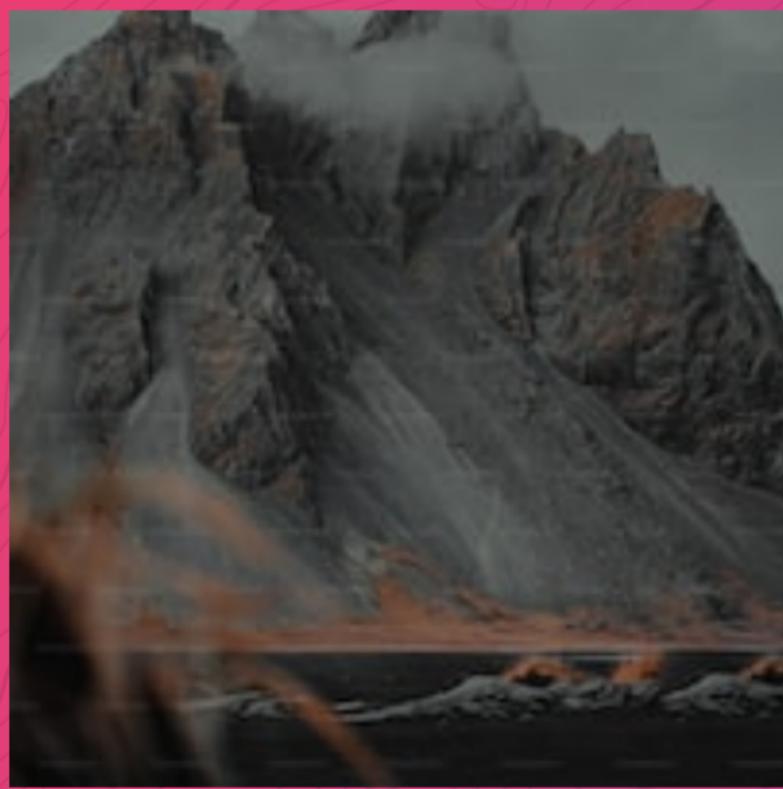
Featured Taco

Slim Profile, easy to hold and loaded with cheese.

This is the one you have been waiting for

[Learn More →](#)

Instant Grams



<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez