
Cómo aprovechar las variables en CSS

Jorge Aguirre - FrontEnd Developer @ Axis

Contenidos

- 1 - Historia
 - 2 - Sintaxis
 - 3 - Especificación
 - 4 - Errores
 - 5 - Casos de uso
-

Todo está aquí



code-jorge

HISTORIA

Cómo comienza todo

Variables en SASS (y otros)

Antes de tener variables en CSS, teníamos variables en SASS pero ojo, que no funcionan igual.

Se sustituyen durante la compilación

SCSS

```
$height: 40px;

.area {
  height: $height;
}
```



CSS

```
.area {
  height: 40px;
}
```

* En el CSS final no hay ninguna variable

Hoy en día tenemos variables en CSS

```
:root {  
  --height: 40px;  
}
```

CSS nativo, sin necesidad de preprocesadores.
Forma parte del estándar.

¿Y lo podemos utilizar?



Lo podemos utilizar



93% de soporte global

Soy del otro 7%

Cambia de trabajo

Soy del otro 7%

```
@supports(--css: variables) {  
  :root {  
    --height: 40px;  
  }  
}
```

Puedes usar las “CSS feature queries” - pero no vale la pena

SINTAXIS

Cómo defino y uso CSS variables

Declarar variables

```
:root {  
  --red: #dc3545;  
  --blue: #007bff;  
  --99: 99;  
  --🐵: monkey;  
}
```

Las variables empiezan por --

Nombrar variables

```
:root {  
  --red: #dc3545;  
  --Red: #dc3546;  
  --RED: #dc3547;  
}
```

Son sensibles a mayúsculas, CSS entiende que hay 3 propiedades diferentes

Usar variables

```
:root {  
  --red: #dc3545;  
}  
  
.block {  
  color: var(--red);  
}
```

La función **var()** es tu nueva mejor amiga

Valor por defecto

```
.block {  
  color: var(--red, #dc3545);  
}
```

¡Por si la variable no existe!

Solo para valores

```
:root {  
  --property: color  
}  
  
.block {  
  var(--property): #dc3545;  
}
```

This is not the way!

INTRODUCCIÓN

Usando las variables en CSS

Declaramos variables en ficheros CSS

Declaro la variable

```
:root {  
  --color-red: #dc3545;  
  --color-blue: #007bff;  
}
```



Uso la variable

```
.block {  
  border: 3px solid var(--color-red);  
  background-color: var(--color-blue);  
}
```

Cualquier tipo de valor es válido

Uso las variables

Declaro más variables

```
:root {  
  --z-index-modal: 3;  
  --z-index-header: 2;  
  --height-header: 50px;  
  --font-size-title: 22px;  
}
```



```
.header {  
  z-index: var(--z-index-header);  
  height: var(--height-header);  
}  
.title {  
  font-size: var(--font-size-title);  
}  
.modal {  
  z-index: var(--z-index-modal);  
}
```

Y se pueden combinar con otros conceptos de CSS

Declaro la variable

```
:root {  
  --height-header: 60px;  
}
```



Uso la variable

```
.main {  
  height: calc(100vh - var(--height-header));  
  overflow: hidden;  
}
```

Muy útil combinarlo con calc()

Incluso consigo misma

```
.main {  
  color: var(--color-1, var(--color-2, white));  
}
```

Puedo usar otra variable de fallback

Incluso consigo misma

```
:root {  
  --font-size-title: 36px;  
  --font-size-subtitle: calc(0.8 * var(--font-size-title));  
}
```

Puedo usar una variable para calcular otra

Se pueden hacer cosas “raras”

Declaro la variable en HEX y en RGB

```
:root {  
  --color-red: #dc3545;  
  --color-red-rgb: 220, 53, 69;  
}
```



Uso la variable

```
.block {  
  background-color: var(--color-red) ;  
  border-width: 2px;  
  border-color: rgb(var(--color-red-rgb)) ;  
}
```

Es una sustitución, no hace falta que la variable tenga sentido

Se pueden hacer cosas “raras” - la secuela

Declaro las variables

```
:root {  
  --color-red: #dc3545;  
  --color-red-rgb: 220, 53, 69;  
  --opacity-disabled: .7  
}
```



Combino varias variables en 1 mismo atributo

```
.button {  
  background-color: var(--color-red);  
}  
  
.button:disabled {  
  background-color: rgba(  
    var(--color-red-rgb),  
    var(--opacity-disabled)  
  );  
}
```

Pero tampoco vale todo

Declaro las variables

```
:root {  
  --font-size: 30;  
}
```



Intento usar la variable

```
.title {  
  font-size: var(--font-size)px  
}
```

ERRORES

Cuando no va todo bien

Usar variables que no existen

```
block {  
  width: 20px;  
  height: 20px;  
  background-color: var(--red, #ff0000);  
}
```

Usar variables que no existen

```
block {  
  width: 20px;  
  height: 20px;  
  background-color: var(--red, #ff0000);  
}
```

Se utiliza el valor de **fallback**

¿Y si no hay fallback?

```
block {  
  width: 20px;  
  height: 20px;  
  background-color: var(--red);  
}
```

Si no hay fallback...

```
block {  
  width: 20px;  
  height: 20px;  
  background-color: var(--red);  
}
```

La propiedad toma el valor **initial**

Usar variables sin sentido

```
block {  
  --red: 20px;  
  width: 20px;  
  height: 20px;  
  background-color: var(--red);  
}
```

Usar variables sin sentido

```
block {  
  --red: 20px;  
  width: 20px;  
  height: 20px;  
  background-color: var(--red);  
}
```

Si el valor final no tiene sentido, se pone **initial**

¿Y si hay fallback?

```
block {  
  --red: 20px;  
  width: 20px;  
  height: 20px;  
  background-color: var(--red, #ff0000);  
}
```

¿Y si hay fallback?

```
block {  
  --red: 20px;  
  width: 20px;  
  height: 20px;  
  background-color: var(--red, #ff0000);  
}
```

No afecta, el fallback es solo para las variables no definidas

¿Y si tengo otro valor sin var()?

```
block {  
  --red: 20px;  
  width: 20px;  
  height: 20px;  
  background-color: #ff0000;  
  background-color: var(--red, #ff0000);  
}
```

Da igual....

```
block {  
  width: 20px;  
  height: 20px;  
  background-color: #ff0000;  
  background-color: 20px;  
}
```

No hace nada especial por usar var()

CASOS DE USO

Ejemplos chulos de CSS

Caso I: Scoping

Usar las reglas de especificidad de CSS para cambiar valores

Puedo definir variables en cualquier selector

```
h1 {
  --font-size: 28px;
  --font-weight: 700;
}
h2 {
  --font-size: 28px;
  --font-weight: 500;
}
h3 {
  --font-size: 24px;
  --font-weight: 400;
}
```

```
h1, h2, h3 {
  font-size: var(--font-size);
  font-weight: var(--font-weight);
}
```

Vamos a verlo

Con **Instagraph**

Caso II: Partial

Usar variables CSS para descomponer valores complejos

Trocear una propiedad en varias variables

```
.box {  
  --b-width: 6px;  
  --b-style: solid;  
  --b-color: steelblue;  
  border: var(--b-width) var(--b-style) var(--b-color);  
}  
  
.box.pink {  
  --b-color: deeppink;  
}
```

Redefino una variable en vez de redefinir todo el valor

Vamos a verlo

Con Qwitter

Caso III: Dynamics

Variables dinámicas - informadas desde JS

Trocear una propiedad en varias variables

```
// Algo de JS tenía que poner en un JSDay...  
document.documentElement.style.setProperty("--color", <value>);
```

Creo o modifiko una variable desde JS

Vamos a verlo

Con **Fatebook** y con **Snapcat**

**Muchas
gracias a
todos**
