En este video, quiero hablar sobre valores reactivos y declaraciones reactivas. Pero, para explicar esto de manera más fácil de entender, voy a usar un ejemplo. Actualmente, solo tenemos dos valores: “name” (nombre) y “belt color” (color del cinturón). Lo que quiero hacer es expandir esto para tener un “first name” (primer nombre) y un “last name” (apellido).

Voy a crear esos dos valores. Primero, el “first name” lo voy a cambiar a “Jimmy” y luego voy a añadir un “last name” que voy a establecer como “Hendrix”. Entonces, ahora tengo un “first name” y un “last name”. Además, voy a crear un campo de entrada para cada uno, para que un usuario pueda elegir qué valores poner, al igual que tenemos el campo de entrada para el color del cinturón.

Voy a eliminar parte del código que no necesito, como el botón y algunas cosas de arriba. Luego, voy a mantener esta parte, que es donde se muestra el color del cinturón. A continuación, voy a agregar un par de entradas para los nombres. Primero, importamos el “first name” y lo vinculamos con el atributo bind, de esta forma:

<input bind:value={firstName} />

Luego, duplicamos esto y cambiamos a “last name”:

<input bind:value={lastName} />

Así que ahora tenemos tres entradas: una para el primer nombre, otra para el apellido y otra para el color del cinturón. Estamos mostrando el color del cinturón aquí, pero en vez de eso, voy a eliminar esa línea y guardar el archivo, de manera que ahora solo mostramos el color del cinturón sin el texto que lo describe.

Si voy a la aplicación y cambio el color del cinturón a “rojo”, se actualiza automáticamente, pero si cambio el primer o el segundo nombre, no pasa nada porque no estamos mostrando esos valores aún. Entonces, si quiero mostrar el nombre completo, tendría que concatenar el primer nombre y el apellido. Podría hacer algo como esto:

firstName + " " + lastName

Esto funciona, y si lo guardamos, veremos que dice “Jimmy Hendrix black belt”. Si cambiamos el primer nombre a “Jimmy Foo”, se actualiza correctamente. Sin embargo, sería mucho más cómodo si tuviéramos un valor llamado “full name” (nombre completo) que se actualice automáticamente cuando cambian el primer o el apellido. Así, no tendríamos que concatenar estos dos valores cada vez.

Para hacer esto, podemos usar lo que se llama un **valor reactivo**. Los valores reactivos en Svelte son valores que se actualizan automáticamente cuando cambian los datos de los que dependen. Es como si fueran propiedades computadas en Vue.js, si estás familiarizado con ellas.

El valor reactivo lo creamos con la sintaxis $: de la siguiente manera:

$: fullName = `${firstName} ${lastName}`;

Esto es lo que hace que “fullName” se actualice automáticamente cada vez que cambian “firstName” o “lastName”. Ahora, en vez de concatenar el nombre completo en la vista, solo tenemos que mostrar el valor de fullName:

<h1>{fullName}</h1>

Ahora, siempre que cambiemos el primer o el apellido, el nombre completo se actualizará automáticamente. Esto es más eficiente porque si tuviéramos que mostrar el nombre completo en varios lugares, no tendríamos que escribir la concatenación en cada uno de ellos.

Este es un ejemplo sencillo, pero la idea es mostrarte cómo funcionan los valores reactivos. En ejemplos más complejos, los valores reactivos serán más útiles. Ahora vamos a guardar y ver cómo funciona:

Puedes cambiar el primer nombre, el apellido y el color del cinturón, y todo se actualiza correctamente.

### Valores Reactivos vs. Declaraciones Reactivas

Además de los valores reactivos, existe algo llamado **declaraciones reactivas**, que son similares pero en lugar de contener solo un valor, ejecutan un bloque de código cada vez que los valores dentro de ellos cambian. Para crear una declaración reactiva, usamos la misma sintaxis $: pero seguida de un bloque de código.

Por ejemplo, si quiero imprimir en consola el color del cinturón cada vez que cambie, haría lo siguiente:

$: console.log(beltColor);

Esto ejecutará el console.log cada vez que el color del cinturón cambie. Si guardamos el archivo y cambiamos el color en la interfaz, veremos cómo se imprime el nuevo color en la consola cada vez.

Si quiero hacer algo más complejo, como mostrar tanto el color del cinturón como el nombre completo, puedo usar un bloque de código:

$: {  
 console.log(beltColor);  
 console.log(fullName);  
}

Este bloque de código se ejecutará cada vez que cambien los valores que usamos dentro de él (en este caso, beltColor y fullName). No importa si cambian ambos, solo basta con que uno de ellos cambie para que se ejecute todo el bloque.

Con esto, cada vez que cambiamos el color del cinturón o el nombre completo, se imprimen ambos en la consola.

En resumen, hoy vimos dos conceptos importantes: los **valores reactivos** y las **declaraciones reactivas**. Estos son muy poderosos en Svelte y te permiten reaccionar a cambios en el estado de la aplicación de manera eficiente. Este fue solo un ejemplo simple, pero con el tiempo los usaremos de manera más compleja para manejar estados más avanzados.