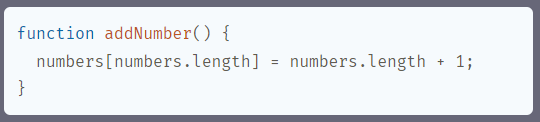
REACTIVITY: UPDATING ARRAYS & OBJECTS

Svelte’s reactivity is triggered by assignments. Methods that mutate arrays or objects will not trigger updates by themselves. You need to assign a declared variable to itself within the array/object to tell the compiler it has changed. An example will be shown below.

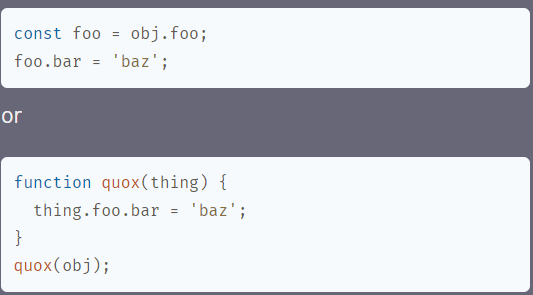
|  |  |
| --- | --- |
| <script>  let numbers = [1, 2, 3, 4];  function addNumber() {  numbers.push(numbers.length + 1);  numbers = numbers;  }  $: sum = numbers.reduce((t, n) => t + n, 0);  </script>  <p>{numbers.join(' + ')} = {sum}</p>  <button on:click={addNumber}>  Add a number  </button> |  |
| Can also write it in ES6 spread syntax |  |

The same rule applies to array methods such as pop, shift, and splice and to object methods such as Map.set, Set.add, etc.

Assignments to properties of arrays and objects work the same way as assignments to the values themselves.



However, indirect assignments to references such as



Won’t trigger reactivity unless you follow it up with a self-referencing statement on a reactive variable. A basic rule of thumb should be that the updated variable must directly appear on the left hand side of the assignment statement.