



SVELTE • RUNES

\$inspect

ON THIS PAGE



`$inspect` only works during development. In a production build it becomes a noop.

The `$inspect` rune is roughly equivalent to `console.log`, with the exception that it will re-run whenever its argument changes. `$inspect` tracks reactive state deeply, meaning that updating something inside an object or array using fine-grained reactivity will cause it to re-fire ([demo](#)):

```
<script>
  let count = $state(0);
  let message = $state('hello');

  $inspect(count, message); // will console.log when `count` or `message` change
</script>

<button onclick={() => count++}>Increment</button>
<input bind:value={message} />
```



\$inspect(...).with

`$inspect` returns a property `with`, which you can invoke with a callback, which will then be invoked instead of `console.log`. The first argument to the callback is either `"init"` or `"update"`; subsequent arguments are the values passed to `$inspect` ([demo](#)):

```
<script>
  let count = $state(0);
```



```
});  
</script>  
  
<button onclick={() => count++}>Increment</button>
```

A convenient way to find the origin of some change is to pass `console.trace` to `with` :

```
$inspect(stuff).with(console.trace);
```



[✎ Edit this page on GitHub](#)

PREVIOUS

[\\$bindable](#)

NEXT

[\\$host](#)