Svelte

Il framework UI veloce, potente e sorprendente

Bacaro Tech

CODE AND FUN

PRESENTAZIONI

Michele Scarpa

Software Engineer

@michele-scarpa

michele-scarpa-90-arco

JS TS

Bacaro
Tech
CODE AND FUN

BACAROTECH

Michele Scarpa
Full stack developer

Giorgio Basile
FE developer

Moreno Frigo Turco
BE developer

Bacarotech si espande: presto altri talenti stanno per brindare con noi

Let me introduce you BACAROTECH

Bacaro Tech is an initiative that brings the joyful, communal spirit of the Venetian bacaro to the world of information technology, through social media outreach, events, and workshops.

Bacaro Tech

CODE AND FUN

Bacaro
Tech
CODE AND FUN

SOCIAL
BACAROTECH

@BACAROTECHOFFICIAL

Key Point

What is Svelte

How it works

Comparisons

SSR

Bacaro
Tech
CODE AND FUN

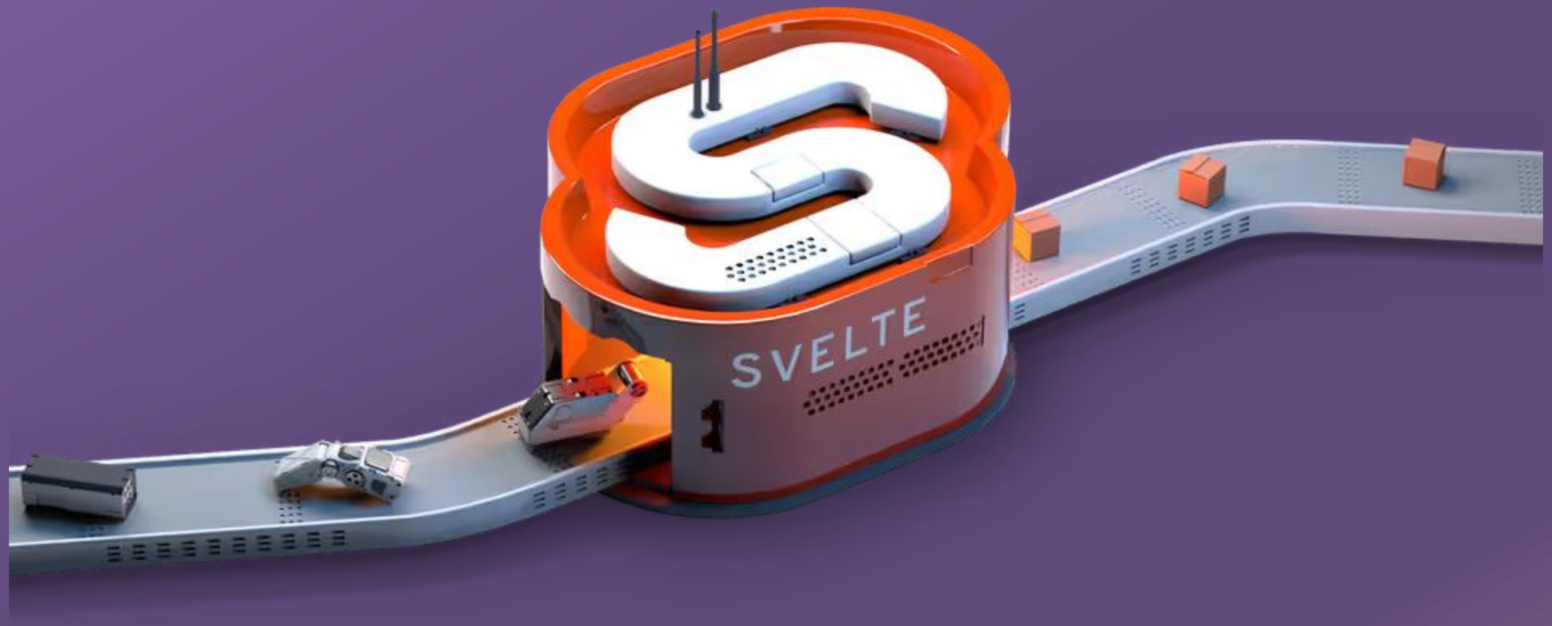
Console.warn(...)

New Tools

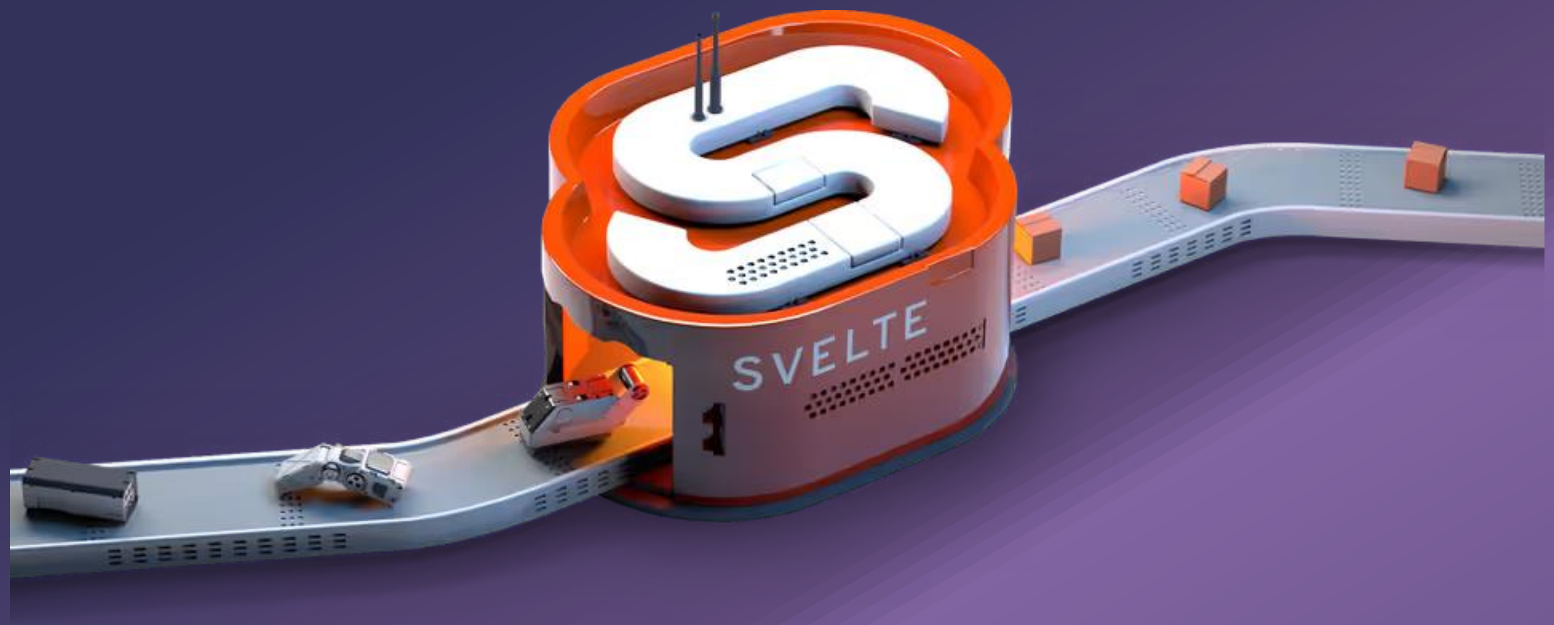
'the site'

Framework completed

Let's start with the introductions of the main character

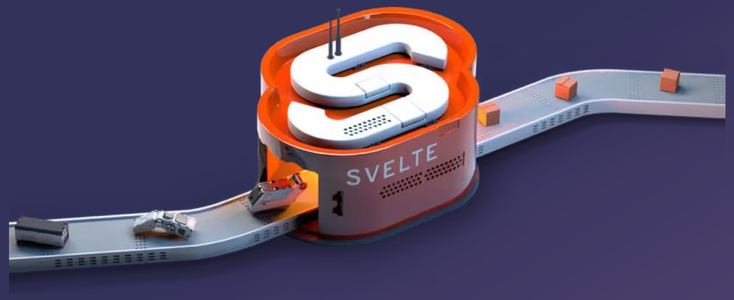


Bacaro
Tech
CODE AND FUN



What is

Svelte is a UI framework that uses a **compiler** to let you write breathtakingly concise components that do **minimal work in the browser**, using languages you already know — **HTML, CSS and JS (TS)**. It’s a love letter to web development.



Where and when

2016 - Rich Harris, from the world of publishing, creates a framework to address many of the inefficiencies of its predecessors.

Compilation at Build Time

Elimination of Runtime Overhead

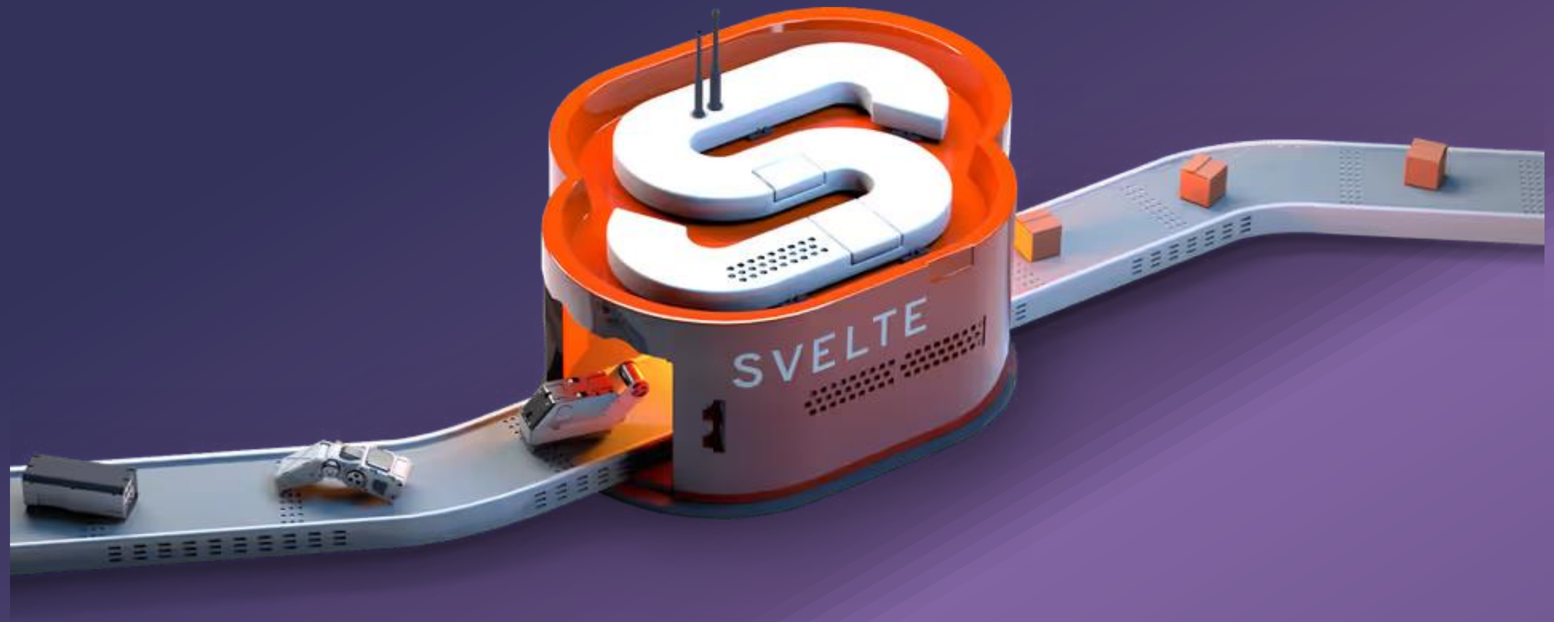
Simplified Reactivity

Clean and Readable Syntax

Runtime

Compile-time

Bacaro Tech
CODE AND FUN



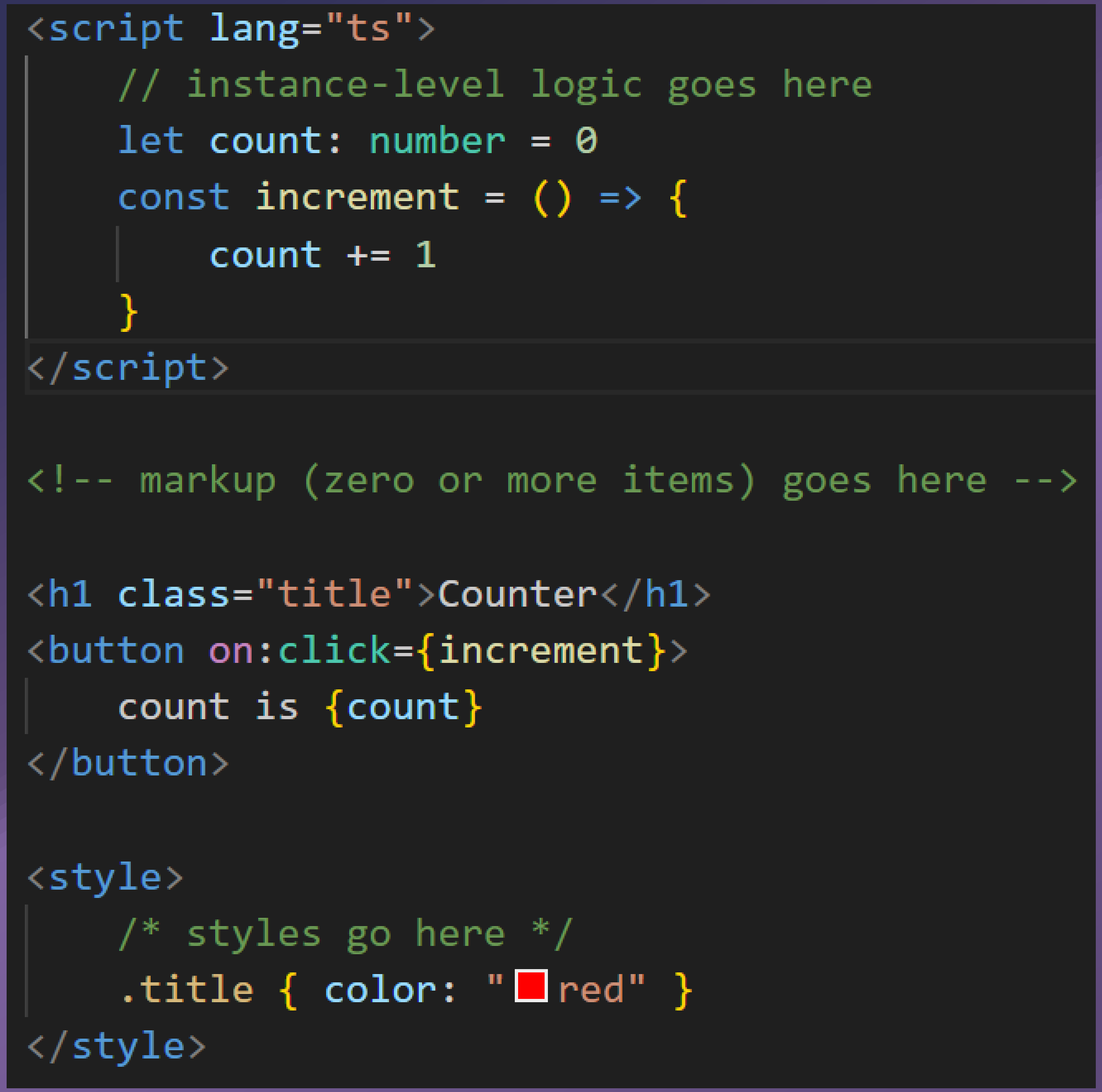
What is a Framework

FRAMEWORKS
are not tools for organizing your code,
they **are tools**
for organizing
your mind

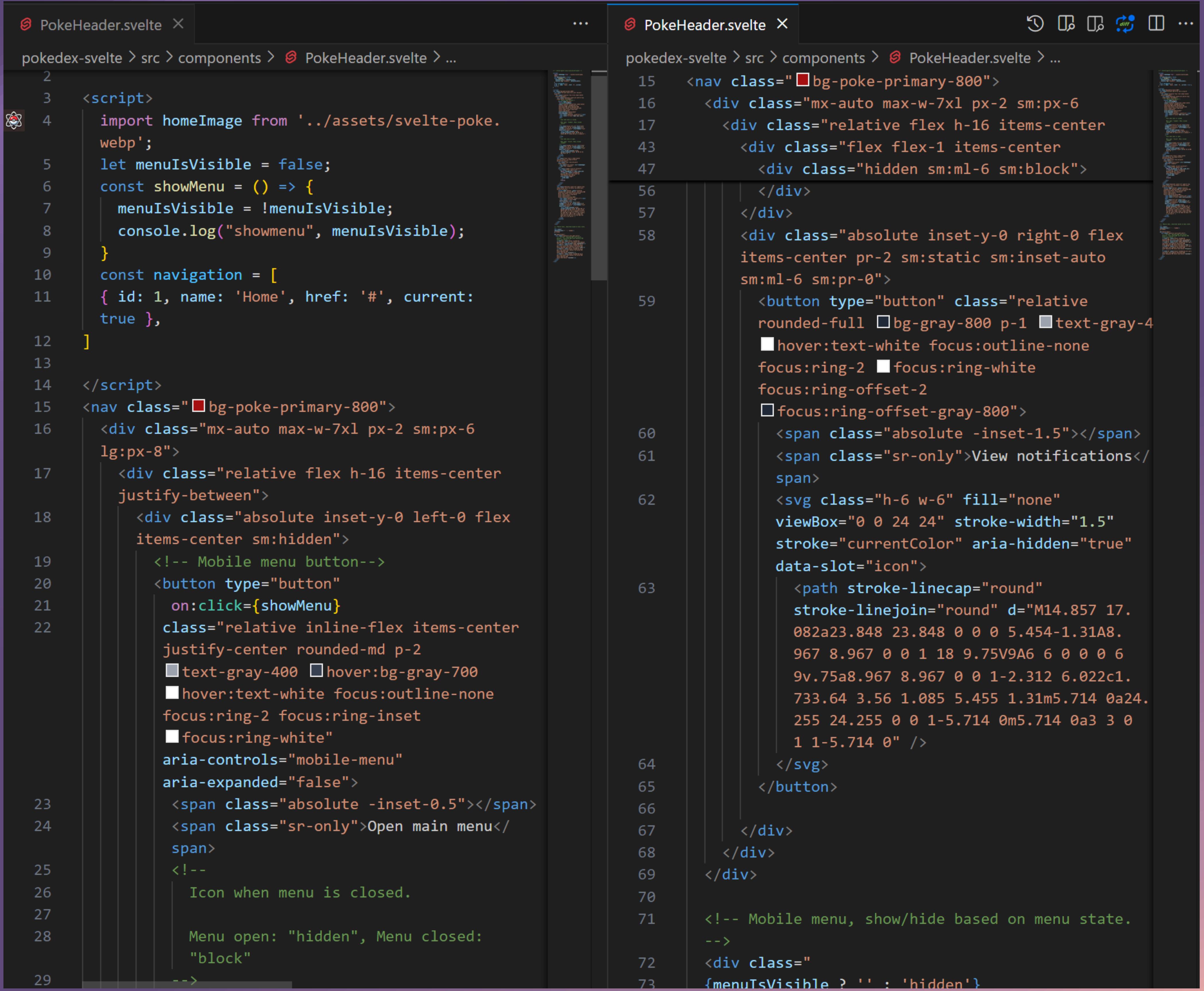
Bacaro
Tech

CODE AND FUN

Components



```
<script lang="ts">
    // instance-level logic goes here
    let count: number = 0
    const increment = () => {
        count += 1
    }
</script>

<!-- markup (zero or more items) goes here -->

<h1 class="title">Counter</h1>
<button on:click={increment}>
    count is {count}
</button>

<style>
    /* styles go here */
    .title { color: "🟥red" }
</style>
```

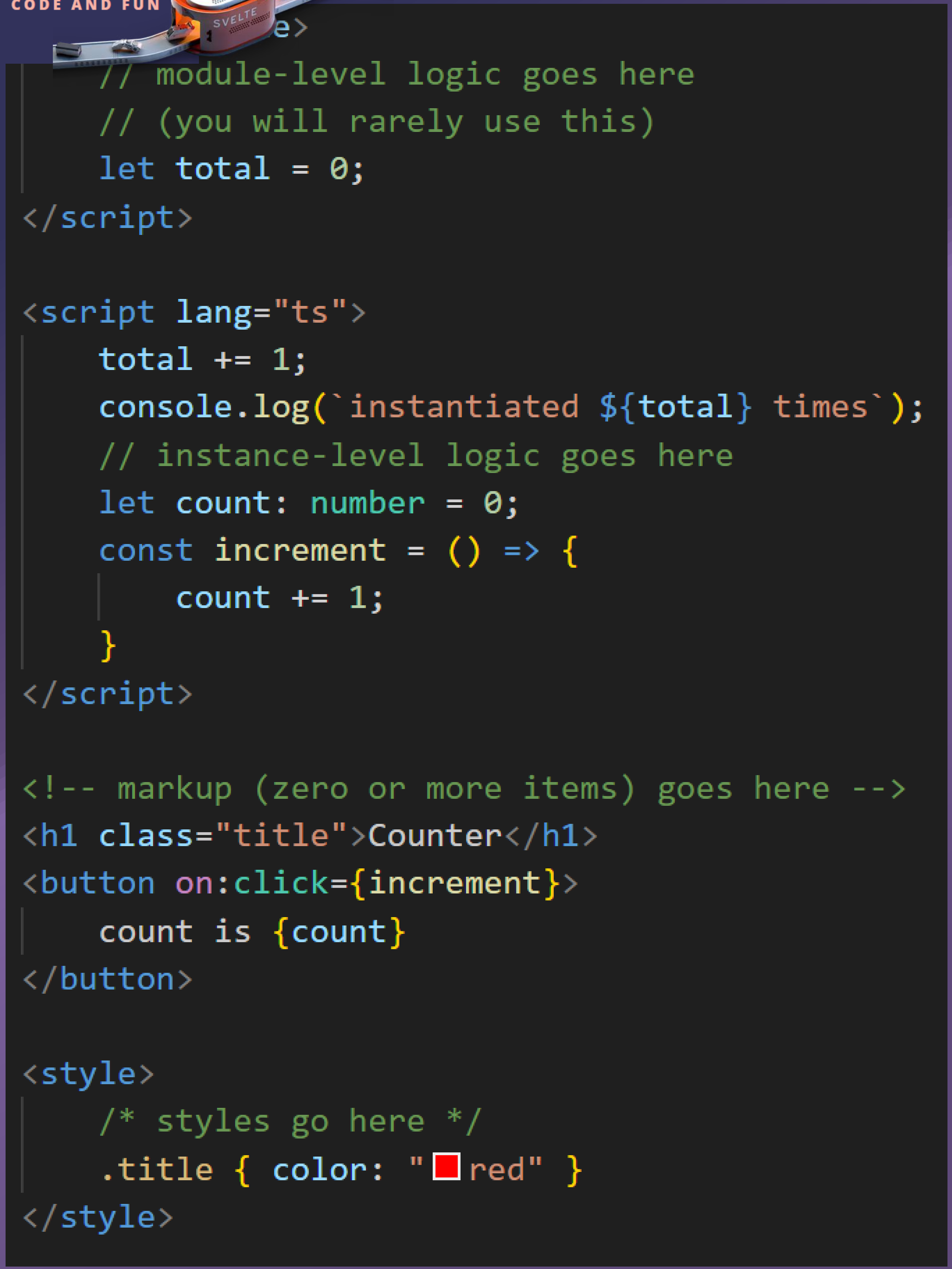


```

<script>
  import homeImage from '../assets/svelte-poke.
  webp';
  let menuIsVisible = false;
  const showMenu = () => {
    menuIsVisible = !menuIsVisible;
    console.log("showmenu", menuIsVisible);
  }
  const navigation = [
  { id: 1, name: 'Home', href: '#', current:
  true },
]

</script>
<nav class="🟥bg-poke-primary-800">
  <div class="mx-auto max-w-7xl px-2 sm:px-6
  lg:px-8">
    <div class="relative flex h-16 items-center
    justify-between">
      <div class="absolute inset-y-0 left-0 flex
      items-center sm:hidden">
        <!-- Mobile menu button-->
        <button type="button"
         on:click={showMenu}
        class="relative inline-flex items-center
        justify-center rounded-md p-2
        ■text-gray-400 ■hover:bg-gray-700
        □hover:text-white focus:outline-none
        focus:ring-2 focus:ring-inset
        □focus:ring-white"
        aria-controls="mobile-menu"
        aria-expanded="false">
         <span class="absolute -inset-0.5"></span>
         <span class="sr-only">Open main menu</
         span>
         <!--
           Icon when menu is closed.

           Menu open: "hidden", Menu closed:
           "block"
         -->
```

```
<nav class="🟥bg-poke-primary-800">
  <div class="mx-auto max-w-7xl px-2 sm:px-6
    <div class="relative flex h-16 items-center
      <div class="flex flex-1 items-center
        <div class="hidden sm:ml-6 sm:block">
        </div>
      </div>
      <div class="absolute inset-y-0 right-0 flex
      items-center pr-2 sm:static sm:inset-auto
      sm:ml-6 sm:pr-0">
        <button type="button" class="relative
        rounded-full ■bg-gray-800 p-1 ■text-gray-4
        □hover:text-white focus:outline-none
        focus:ring-2 □focus:ring-white
        focus:ring-offset-2
        ■focus:ring-offset-gray-800">
          <span class="absolute -inset-1.5"></span>
          <span class="sr-only">View notifications</
          span>
          <svg class="h-6 w-6" fill="none"
          viewBox="0 0 24 24" stroke-width="1.5"
          stroke="currentColor" aria-hidden="true"
          data-slot="icon">
            <path stroke-linecap="round"
            stroke-linejoin="round" d="M14.857 17.
            082a23.848 23.848 0 0 0 5.454-1.31A8.
            967 8.967 0 0 1 18 9.75V9A6 6 0 0 0 6
            9v.75a8.967 8.967 0 0 1-2.312 6.022c1.
            733.64 3.56 1.085 5.455 1.31m5.714 0a24.
            255 24.255 0 0 1-5.714 0m5.714 0a3 3 0
            1 1-5.714 0" />
          </svg>
        </button>

      </div>
    </div>
  </div>

  <!-- Mobile menu, show/hide based on menu state.
  -->
  <div class="
  {menuIsVisible ? '' : 'hidden'}
```

Bacaro
Tech
CODE AND FUN

First Component



```
                e>
    // module-level logic goes here
    // (you will rarely use this)
    let total = 0;
</script>

<script lang="ts">
    total += 1;
    console.log(`instantiated ${total} times`);
    // instance-level logic goes here
    let count: number = 0;
    const increment = () => {
        count += 1;
    }
</script>

<!-- markup (zero or more items) goes here -->
<h1 class="title">Counter</h1>
<button on:click={increment}>
    count is {count}
</button>

<style>
    /* styles go here */
    .title { color: "🟥red" }
</style>
```

<script module>

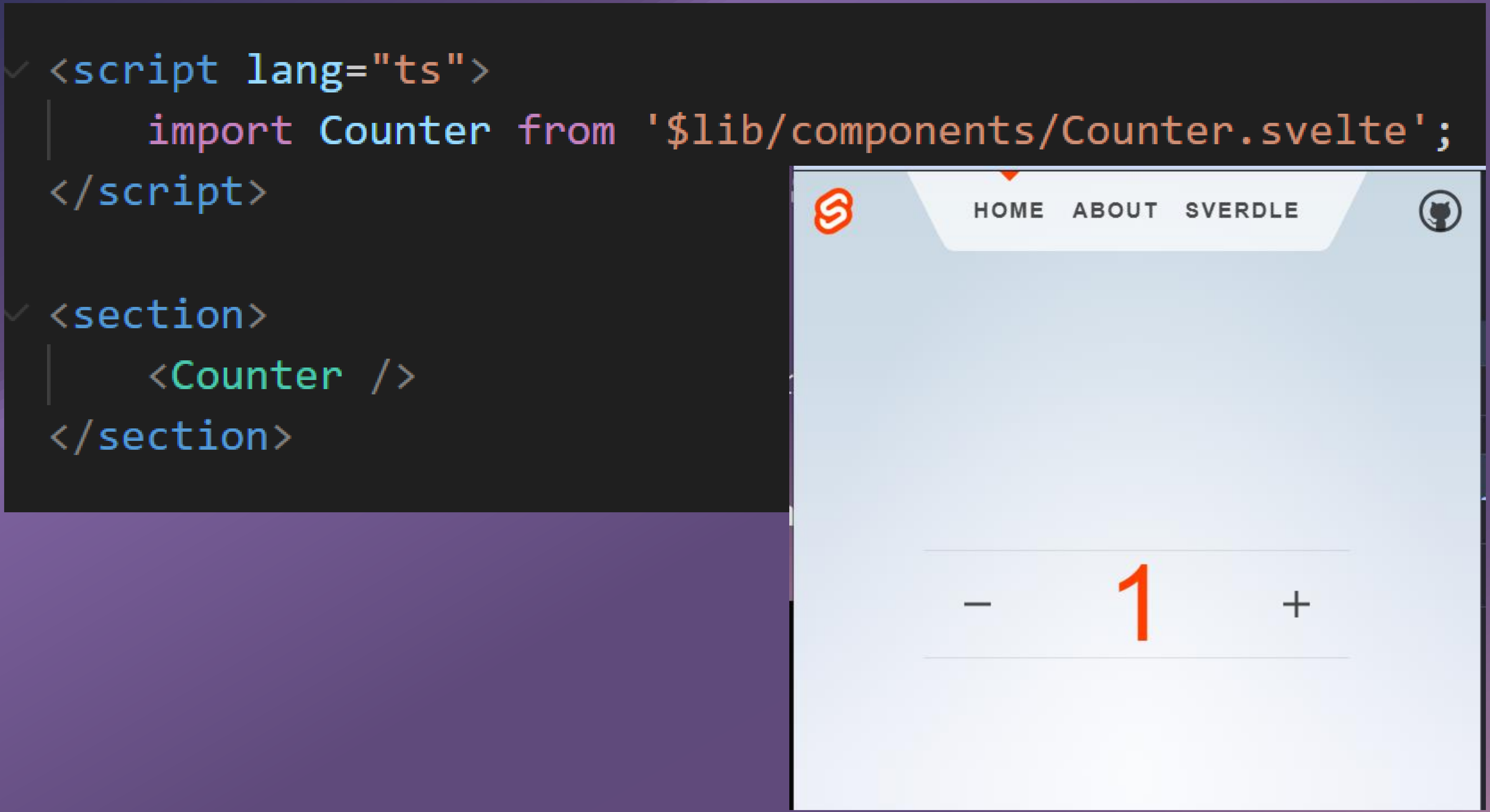
<script> Contains JS/TS

<! – Markup --> html elements

<style> component scoped

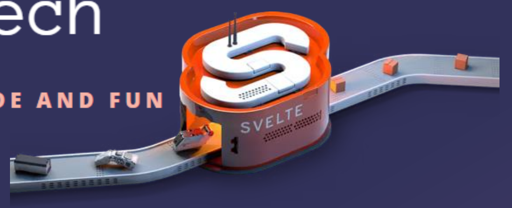
Bacaro
Tech
CODE AND FUN

How to use a component



```
<script lang="ts">
    import Counter from '$lib/components/Counter.svelte';
</script>

<section>
    <Counter />
</section>
```

Bacaro
Tech
CODE AND FUN



How to create First Svelte App Project

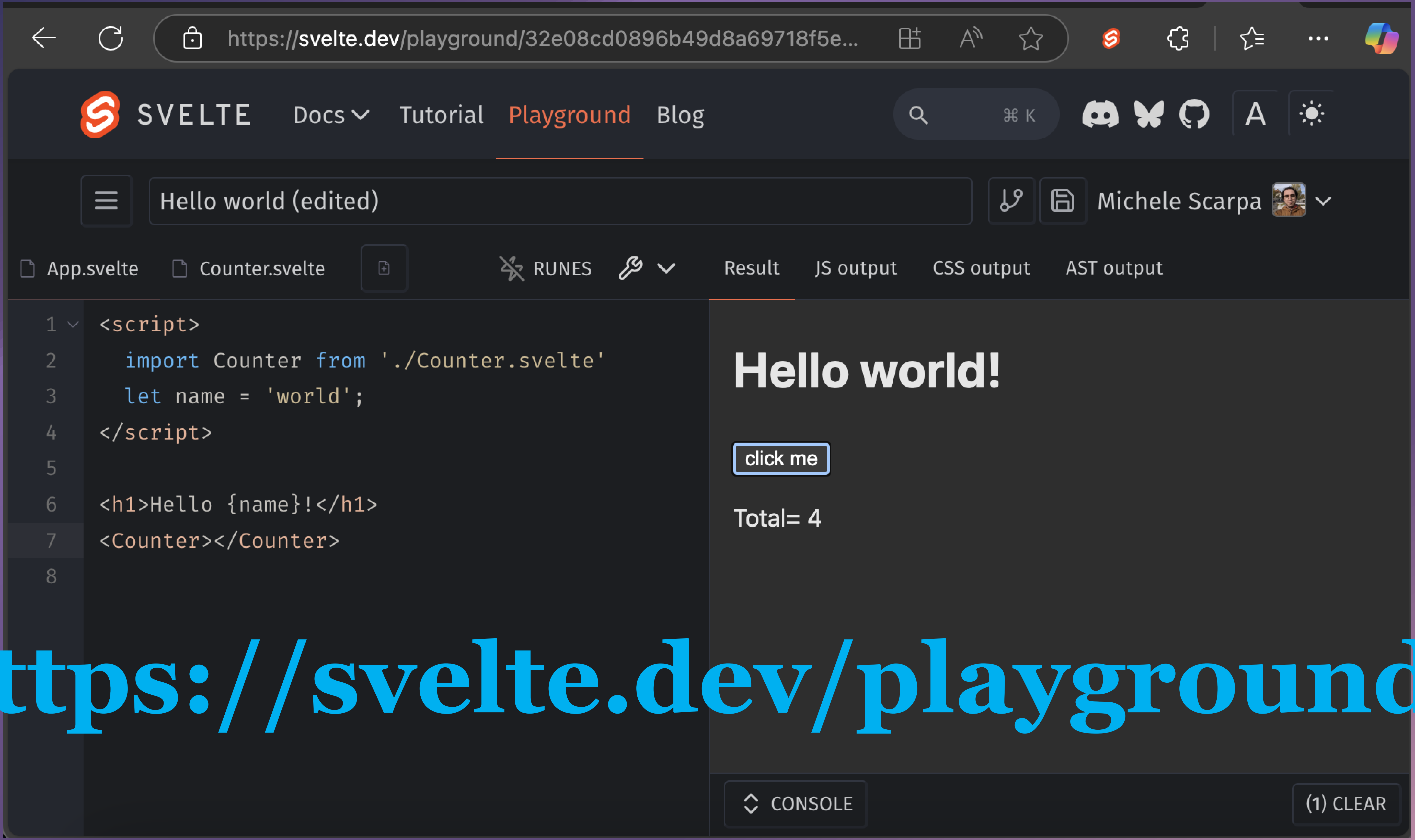
```
npx sv create bacaro-svelte
```

MINIMAL barebones scaffolding for your new app

DEMO showcase app with a word guessing game that works without JavaScript

LIBRAY setup with svelte-package to help correctly package your library

How to try



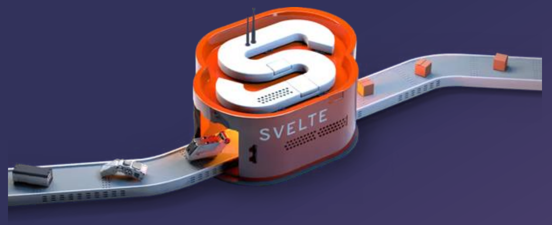
```
<script>
  import Counter from './Counter.svelte'
  let name = 'world';
</script>

<h1>Hello {name}!</h1>
<Counter></Counter>
```

Hello world!

click me

Total= 4

https://svelte.dev/playground/



How to write

Declare Variable

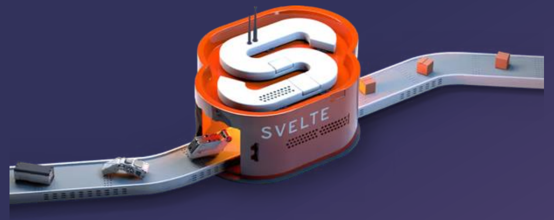
```
let count: number = 0;
```

Interpolation

```
<div>Count is {count}</div>
```

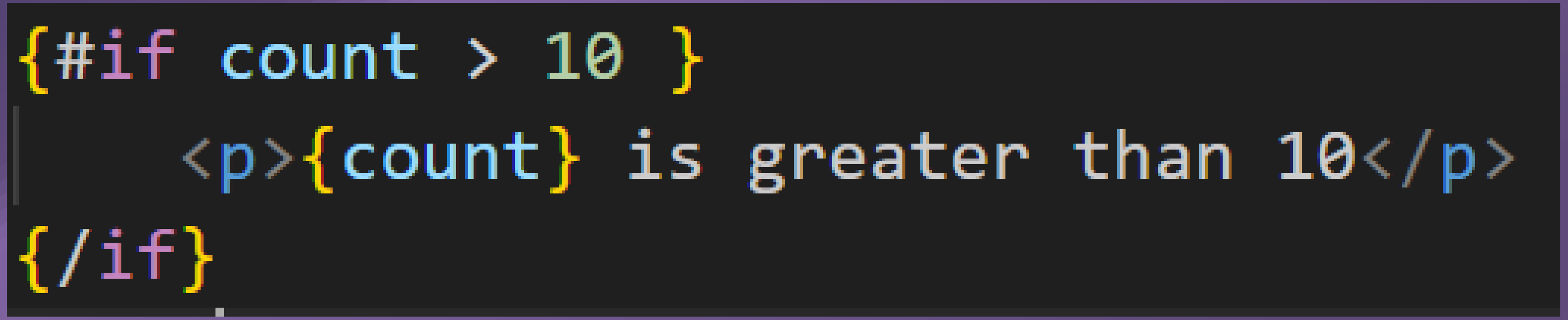
$ Reactivity

```
$: double = count * 2; // svelte 3 - 4
let count = 0;
let name = $state(‘’); // svelte 5: runes
let greeting = $derived(name ?
`Ciao,${name}!` : 'Inserisci il tuo nome’
);
```



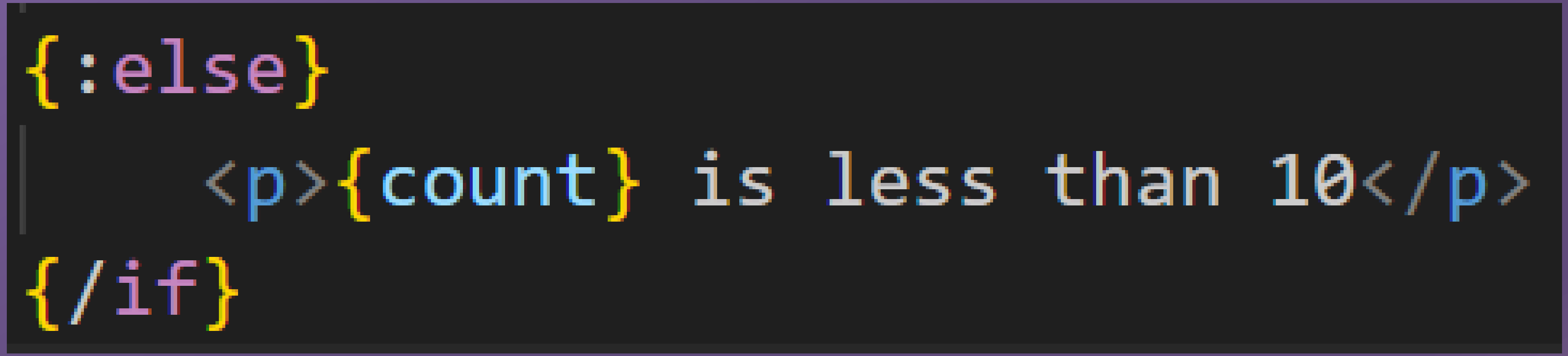
Template: IF

{#if} {/if}



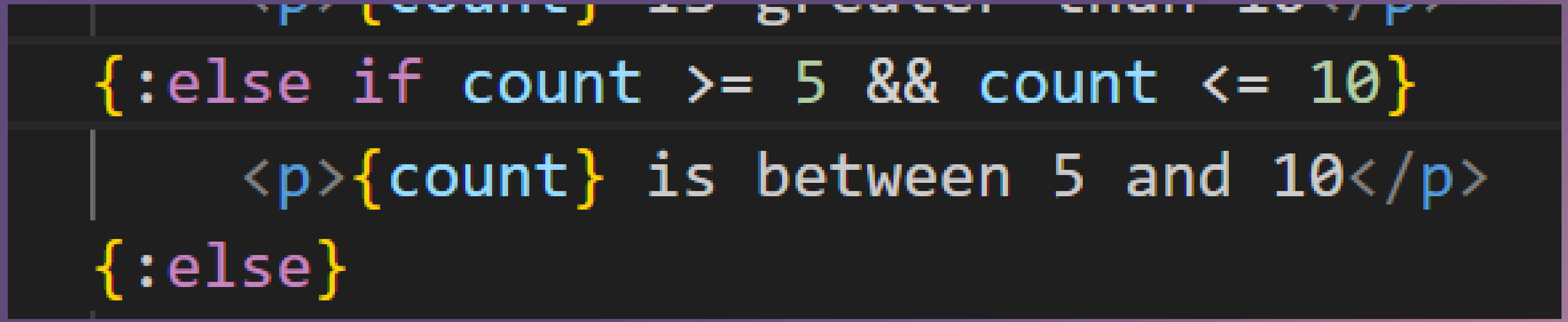
```
{#if count > 10 }
    <p>{count} is greater than 10</p>
{/if}
```

{:else}



```
{:else}
    <p>{count} is less than 10</p>
{/if}
```

{:else if}



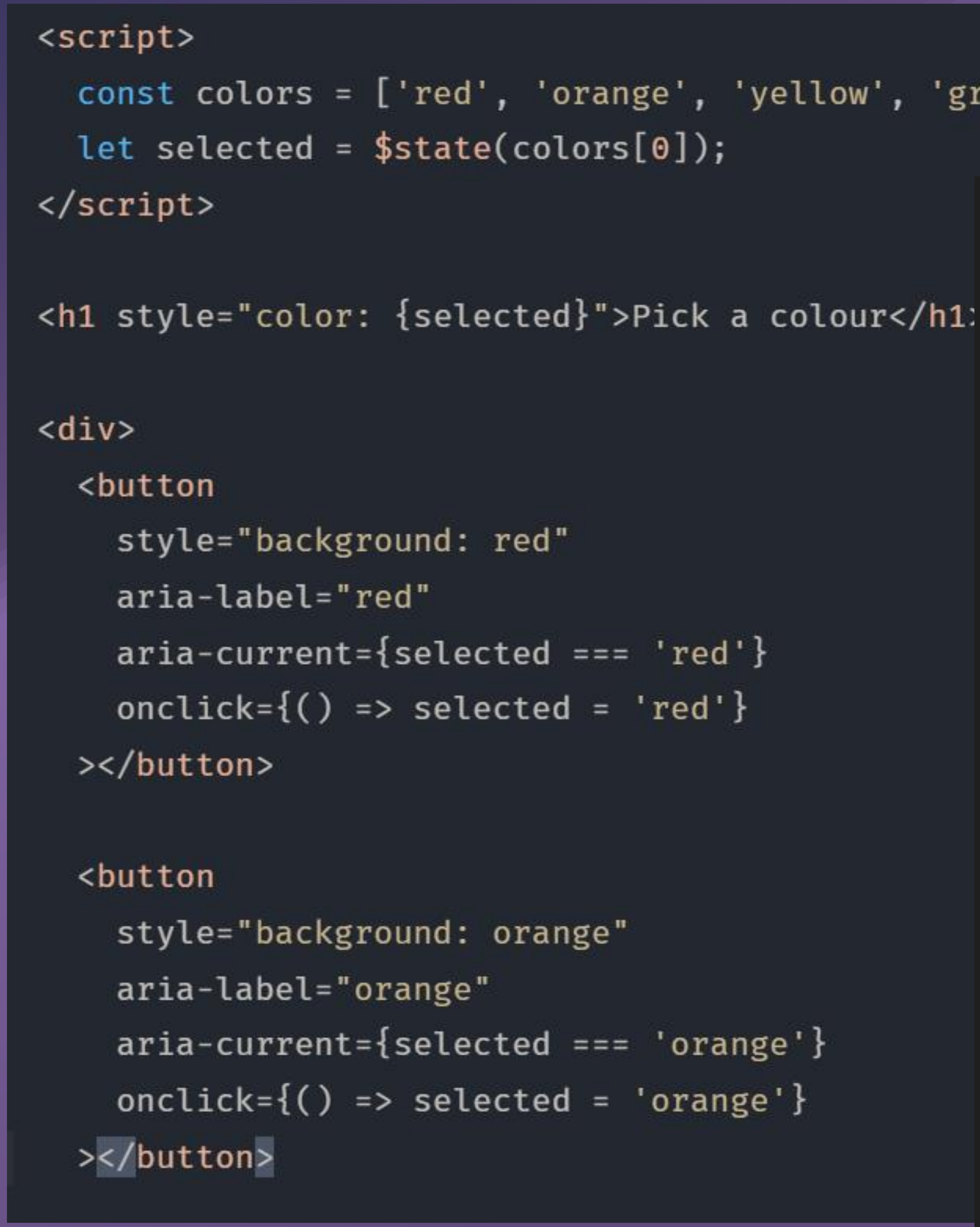
```
        <p>{count} is greater than 10</p>
    {:else if count >= 5 && count <= 10}
        <p>{count} is between 5 and 10</p>
    {:else}
```

z

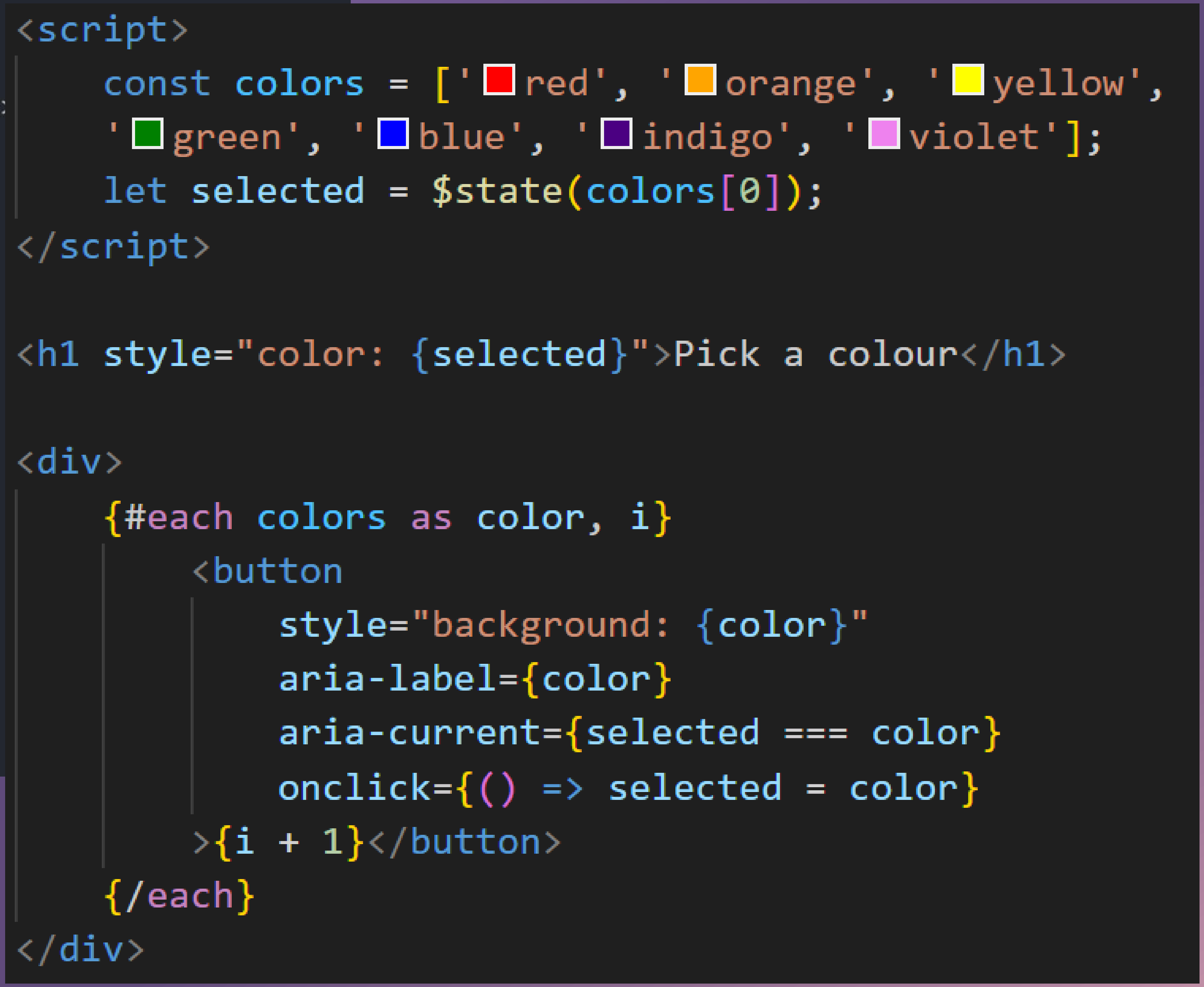
Bacaro
Tech
CODE AND FUN

Template: EACH

{#each }
{/each}



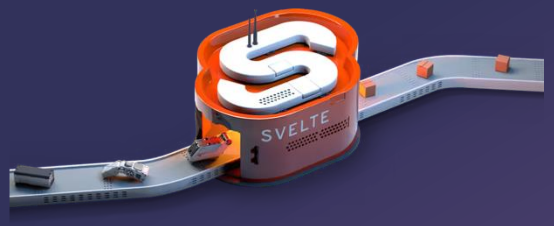
```
<script>
  const colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo',
  let selected = $state(colors[0]);
</script>

<h1 style="color: {selected}">Pick a colour</h1

<div>
  <button
    style="background: red"
    aria-label="red"
    aria-current={selected === 'red'}
    onclick={() => selected = 'red'}
  ></button>

  <button
    style="background: orange"
    aria-label="orange"
    aria-current={selected === 'orange'}
    onclick={() => selected = 'orange'}
  ></button>
```



```
<script>
    const colors = ['red', 'orange', 'yellow',
    'green', 'blue', 'indigo', 'violet'];
    let selected = $state(colors[0]);
</script>

<h1 style="color: {selected}">Pick a colour</h1>

<div>
    {#each colors as color, i}
        <button
            style="background: {color}"
            aria-label={color}
            aria-current={selected === color}
            onclick={() => selected = color}
        >{i + 1}</button>
    {/each}
</div>
```

Z

Bacaro
Tech
CODE AND FUN



Template: await

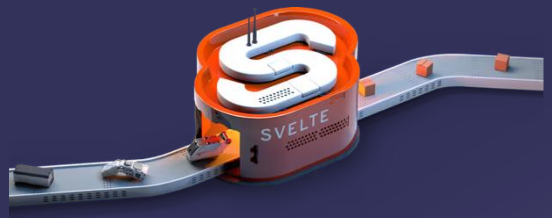
{#await }

{#then }

{#catch }

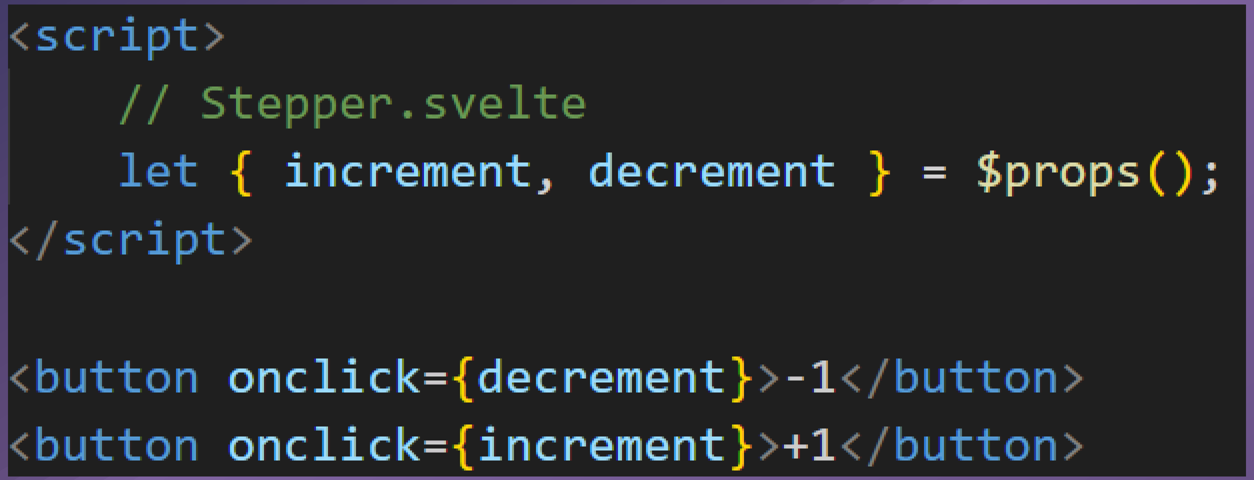
{/await}

```
<script>
  import { roll } from './utils.js';

  let promise = $state(roll());
</script>

<button onclick={() => promise = roll()}>
  roll the dice
</button>

{#await promise}
  <p>...rolling</p>
{:then number}
  <p>you rolled a {number}!</p>
{:catch error}
  <p style="color: red">{error.message}</p>
{/await}
```

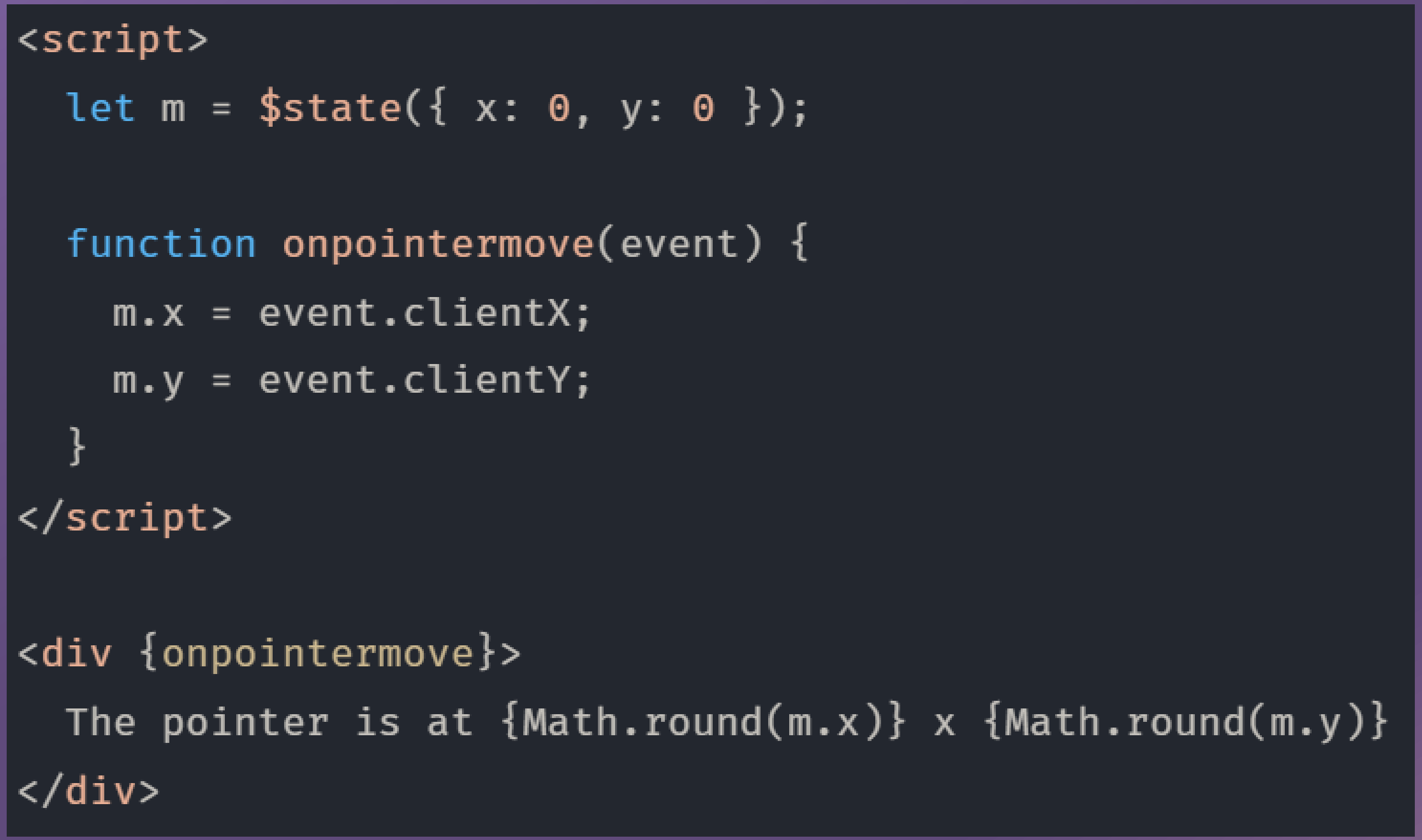


Events

HTML Events

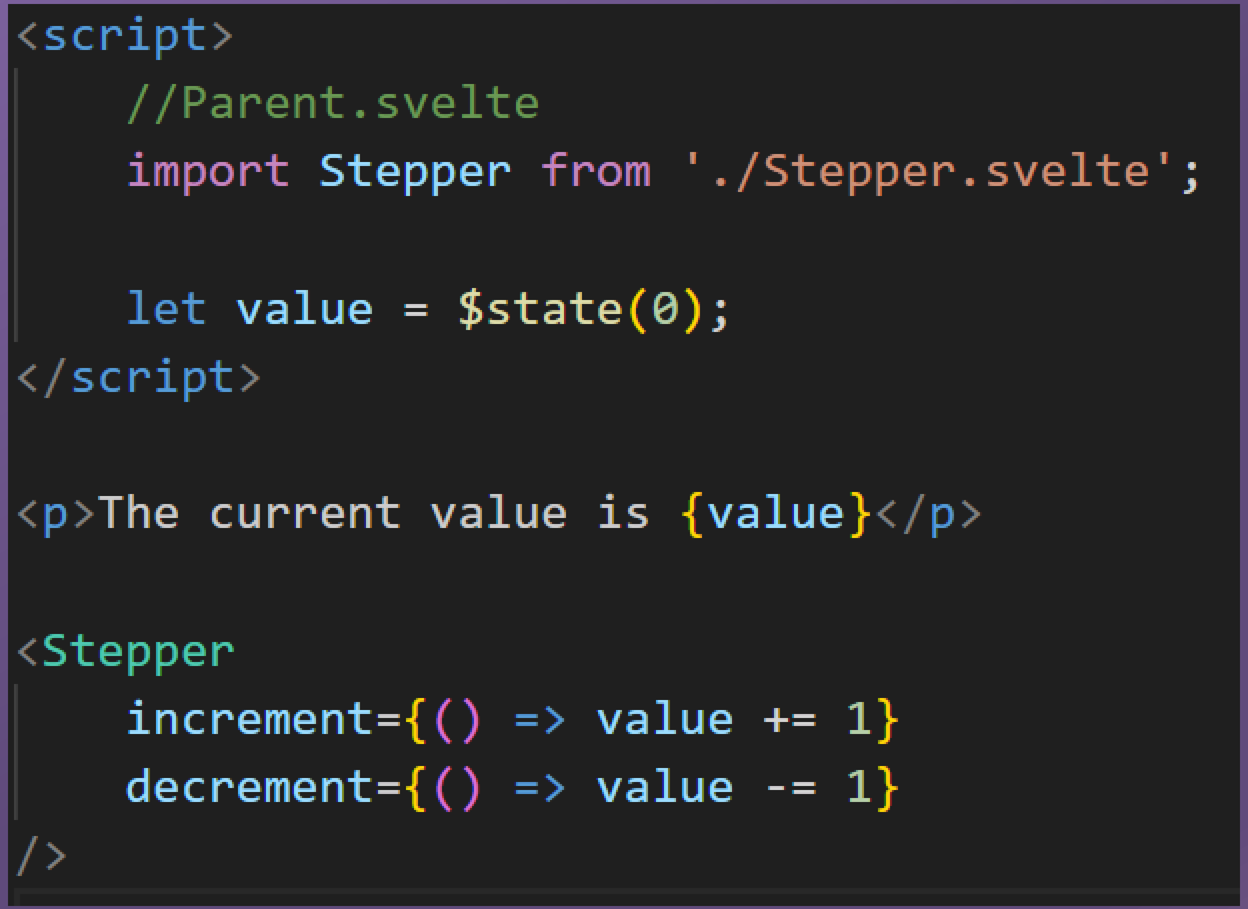


```
<script>
    // Stepper.svelte
    let { increment, decrement } = $props();
</script>

<button onclick={decrement}>-1</button>
<button onclick={increment}>+1</button>
```

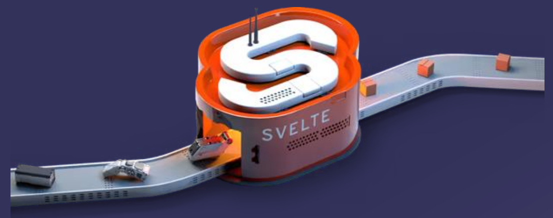


```
<script>
  let m = $state({ x: 0, y: 0 });

  function onpointermove(event) {
    m.x = event.clientX;
    m.y = event.clientY;
  }
</script>

<div {onpointermove}>
  The pointer is at {Math.round(m.x)} x {Math.round(m.y)}
</div>
```

Component Events



```
<script>
    //Parent.svelte
    import Stepper from './Stepper.svelte';

    let value = $state(0);
</script>

<p>The current value is {value}</p>

<Stepper
    increment={() => value += 1}
    decrement={() => value -= 1}
/>
```

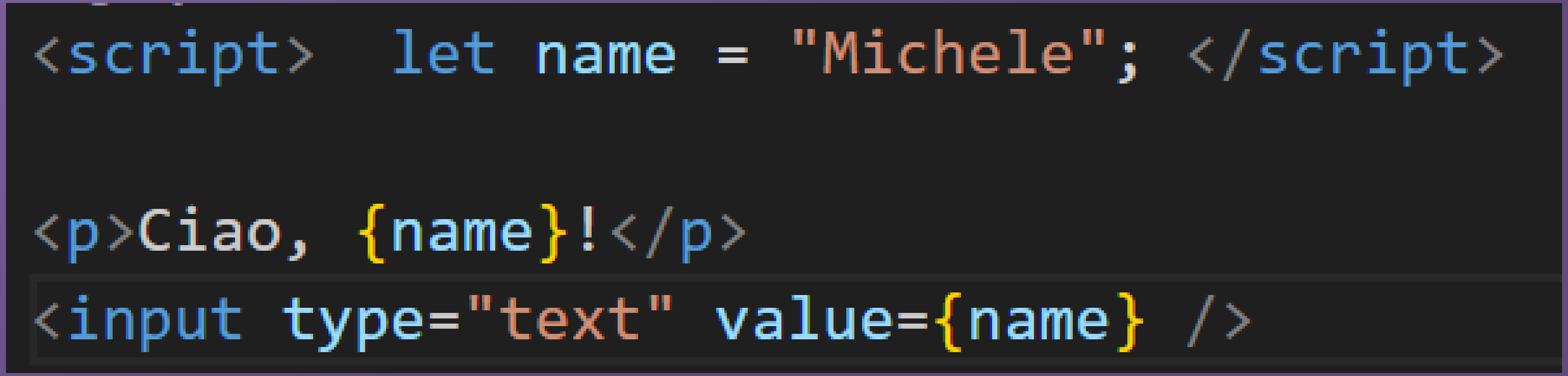


Data binding

Data Binding : synchronized among<script> and UI <html>

Svelte support Two-Way Binding

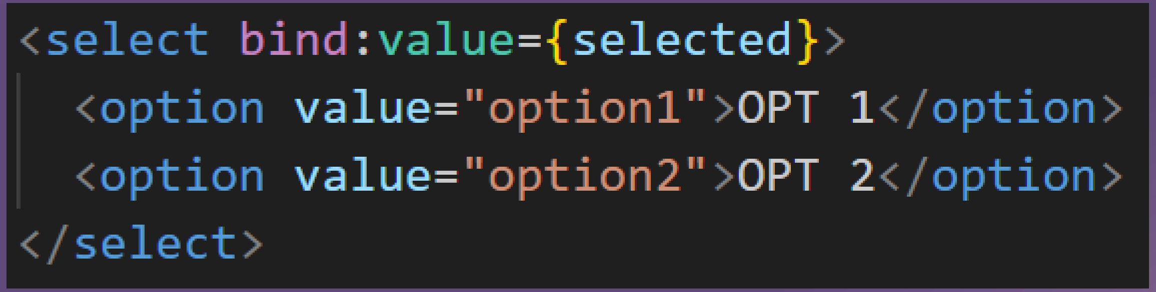
From model to UI



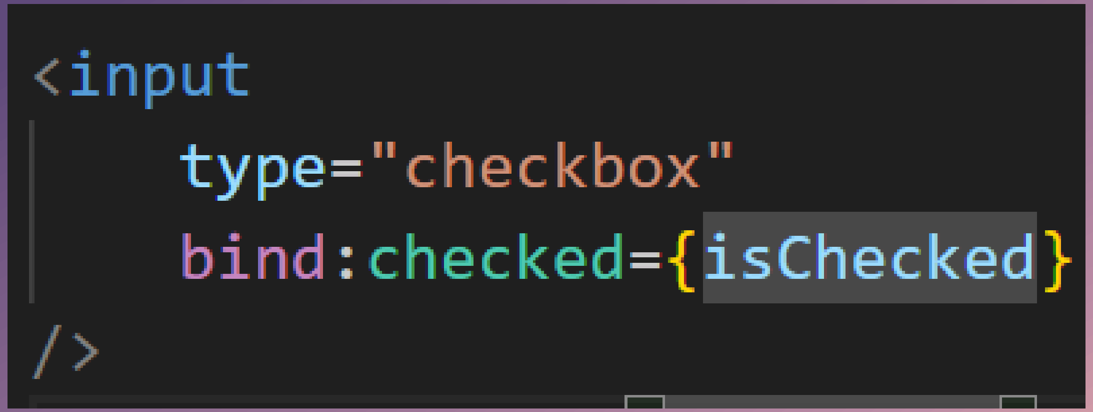
```
<script>  let name = "Michele"; </script>

<p>Ciao, {name}!</p>
<input type="text" value={name} />
```

From UI to model (automatically)

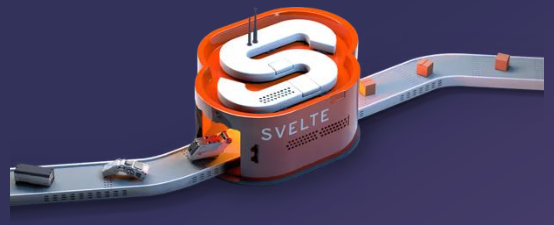
```
<input type="text" bind:value={name} />
```



```
<select bind:value={selected}>
  <option value="option1">OPT 1</option>
  <option value="option2">OPT 2</option>
</select>
```



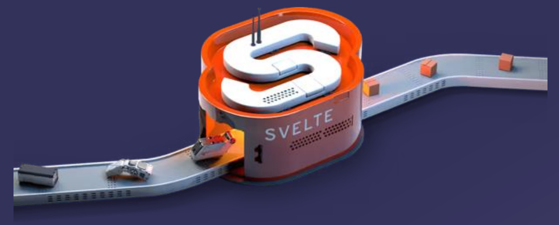
```
<input
    type="checkbox"
    bind:checked={isChecked}
/>
```



How it works

z

Bacaro
Tech
CODE AND FUN



Compile-Time vs Execution Time

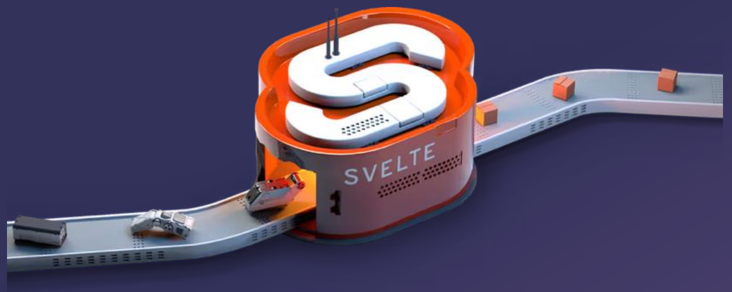
Compile Time

Transforms source code into optimized code before execution

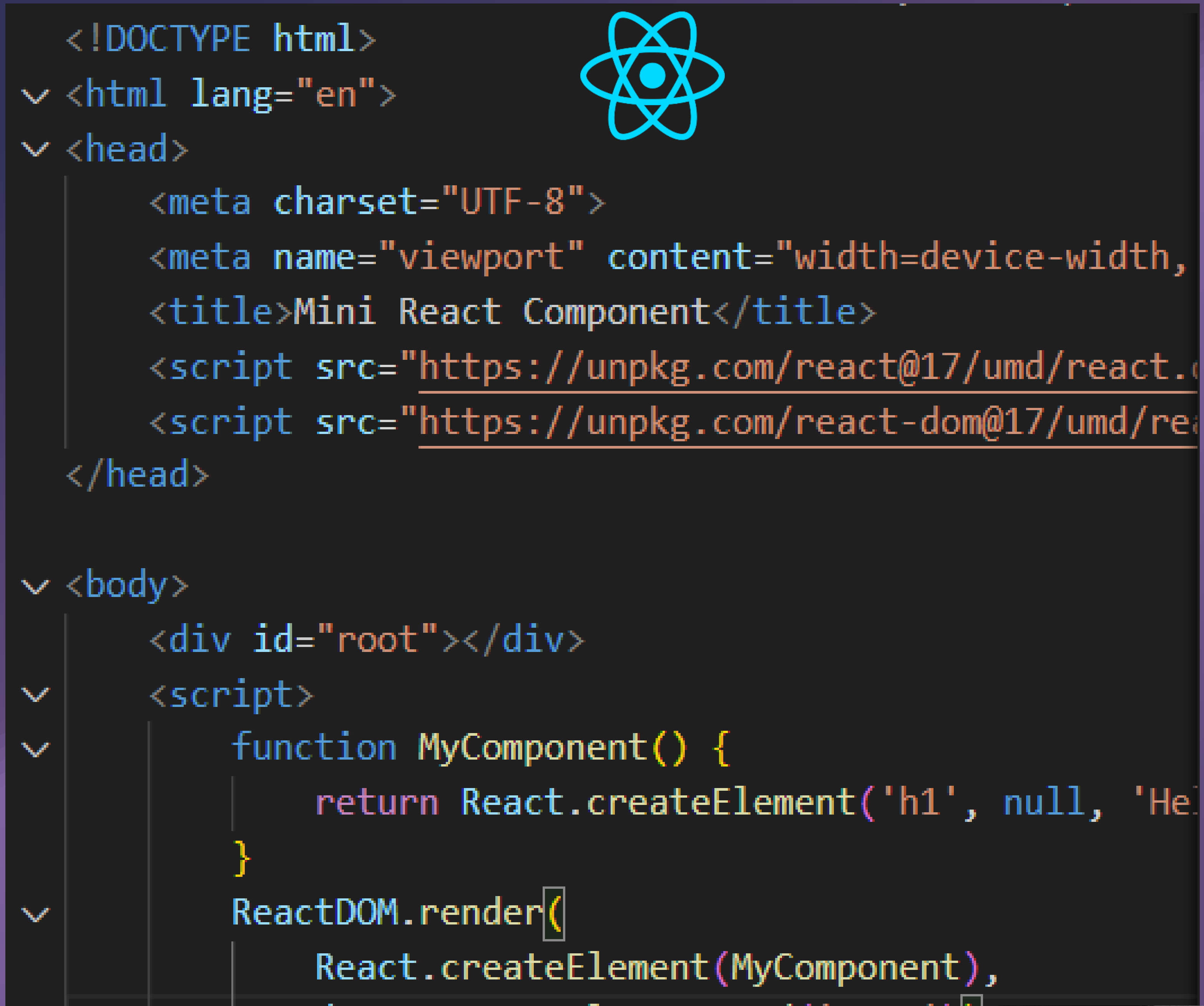
Exeution Time

It executes the code and maneges the behavior ot the application in the browser

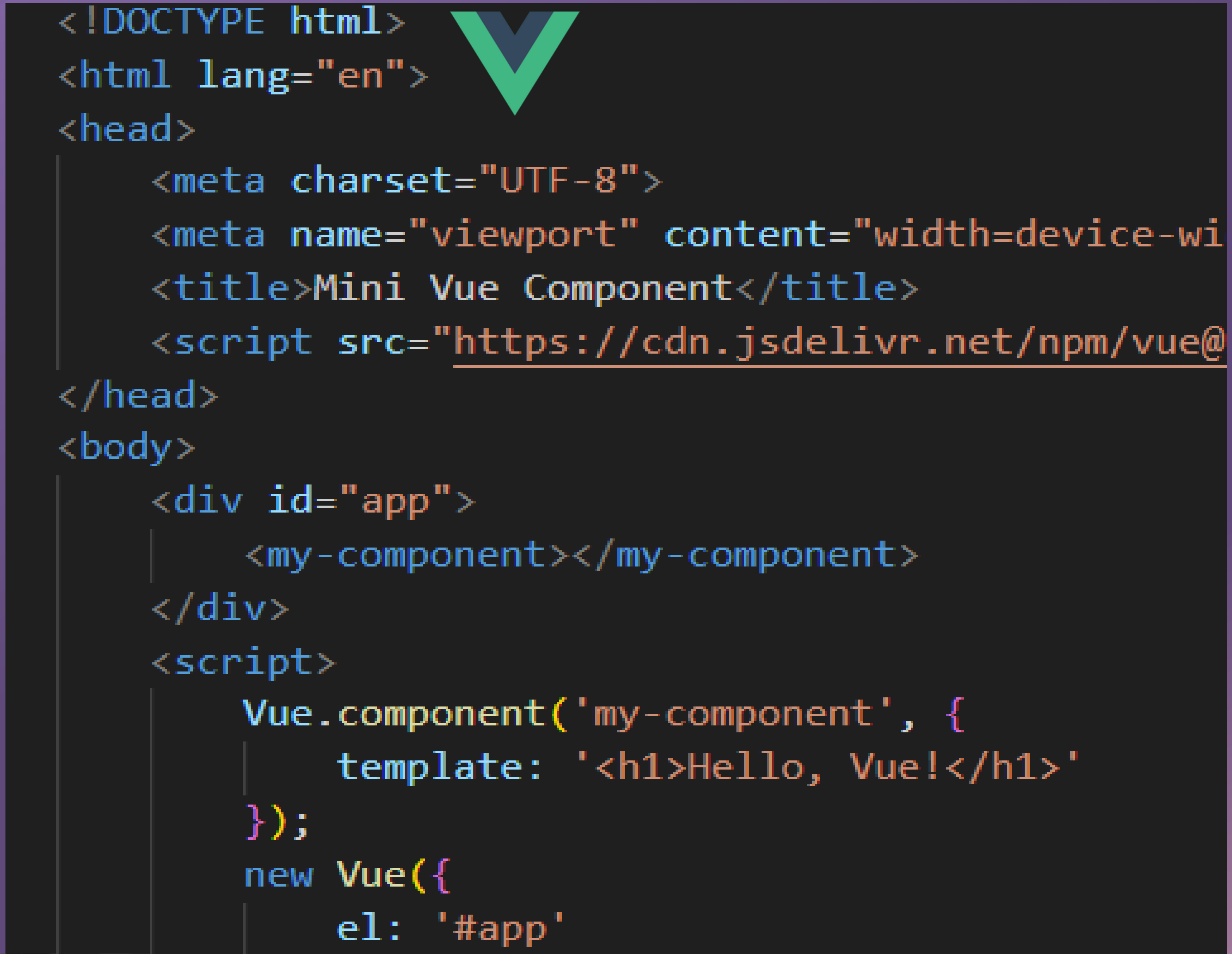
Bacaro
Tech
CODE AND FUN



Execution-time

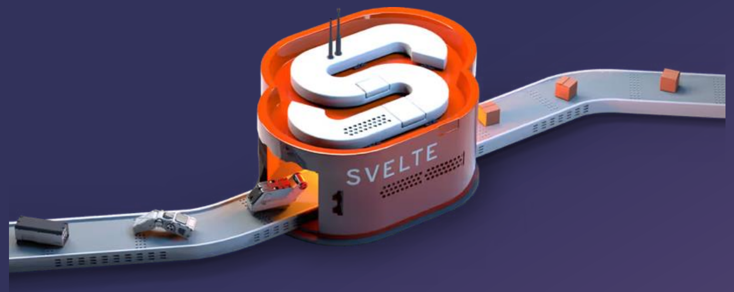


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    <title>Mini React Component</title>
    <script src="https://unpkg.com/react@17/umd/react.
    <script src="https://unpkg.com/react-dom@17/umd/rea
</head>

<body>
    <div id="root"></div>
    <script>
        function MyComponent() {
            return React.createElement('h1', null, 'He
        }
        ReactDOM.render(
            React.createElement(MyComponent),
            document.getElementById('root'));
    </script>
</body>
</html>
```



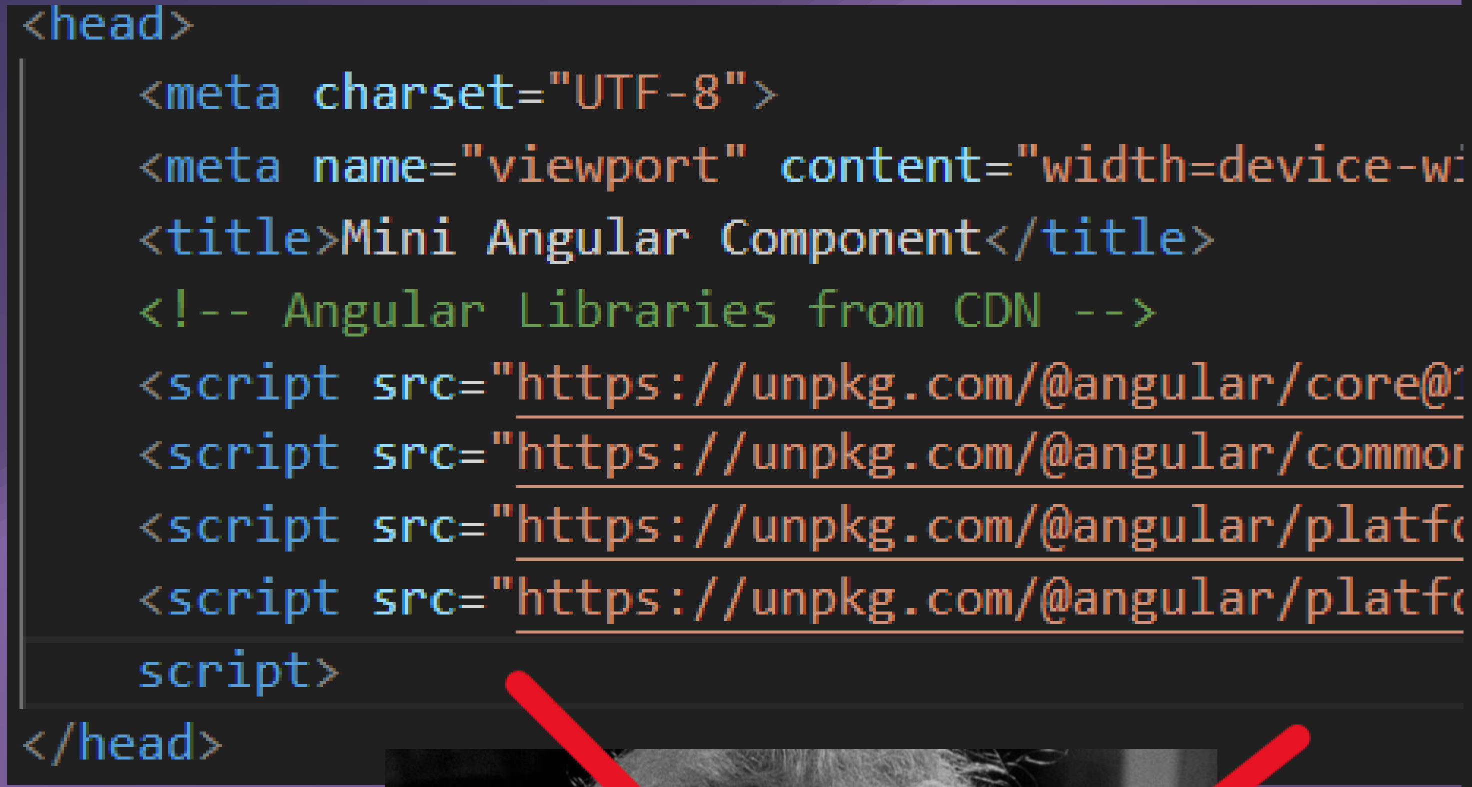
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-wi
    <title>Mini Vue Component</title>
    <script src="https://cdn.jsdelivr.net/npm/vue@
</head>
<body>
    <div id="app">
        <my-component></my-component>
    </div>
    <script>
        Vue.component('my-component', {
            template: '<h1>Hello, Vue!</h1>'
        });
        new Vue({
            el: '#app'
        });
    </script>
dy>
ml>
```

Bacaro
Tech
CODE AND FUN



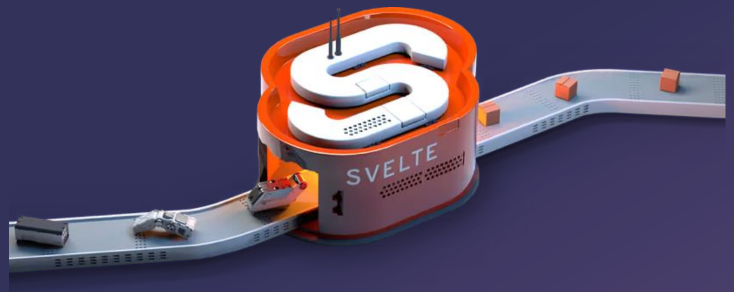
Angular

ZoneJS

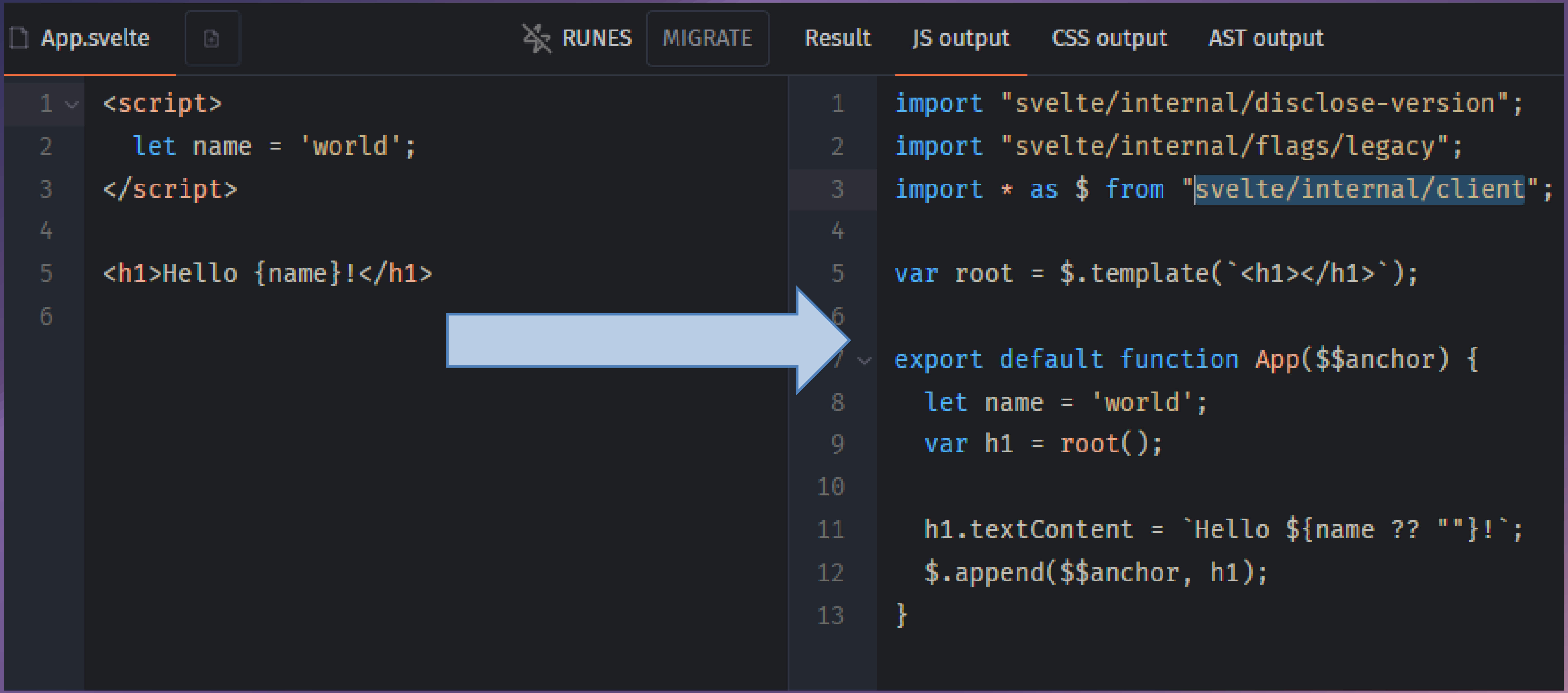


```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-wi
    <title>Mini Angular Component</title>
    <!-- Angular Libraries from CDN -->
    <script src="https://unpkg.com/@angular/core@1
    <script src="https://unpkg.com/@angular/commor
    <script src="https://unpkg.com/@angular/platfc
    <script src="https://unpkg.com/@angular/platfc
    script>
</head>
```

```
<script>
    const { Component, NgModule } = ng.core;
    const { BrowserModule } = ng.platformBrowser;
    const { platformBrowserDynamic } = ng.platformBı

    @Component({
        selector: 'my-app',
        template: `<h1>Hello, Angular!</h1>`
    })
    class AppComponent { }

    @NgModule({
        declarations: [AppComponent],
        imports: [BrowserModule],
        bootstrap: [AppComponent]
    })
    class AppModule { }

    platformBrowserDynamic().bootstrapModule(AppModu
        .catch(err => console.error(err));
</script>
```



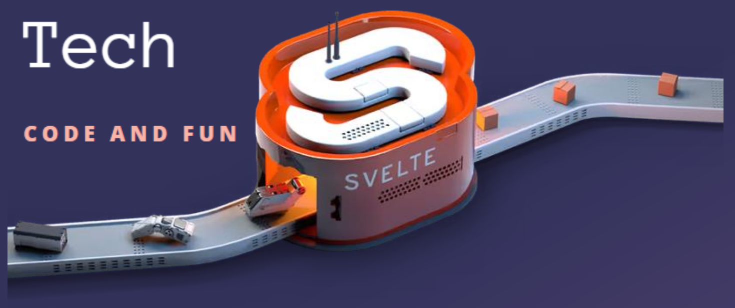
Svelte: Compile-time



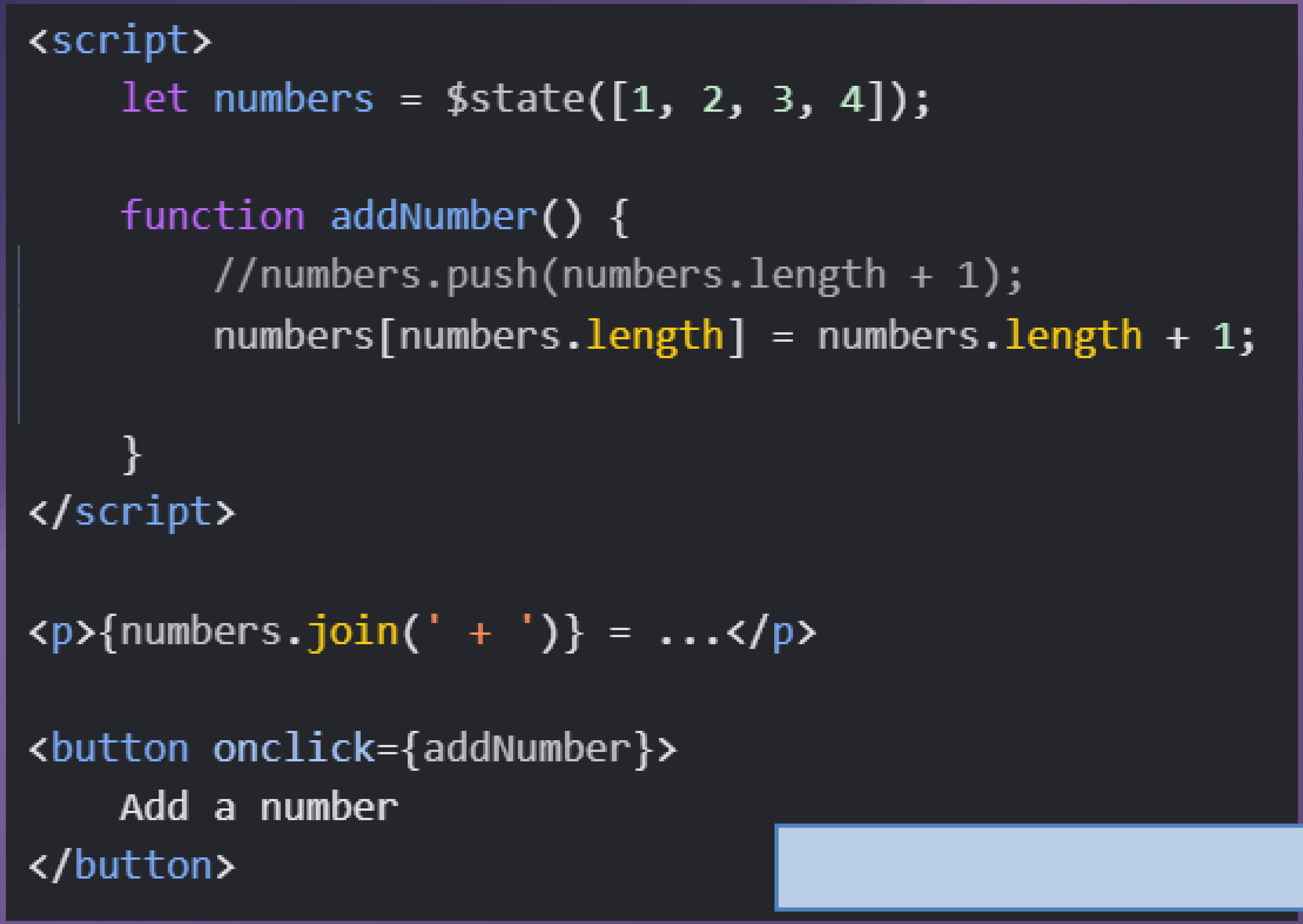
```svelte
<script>
  let name = 'world';
</script>

<h1>Hello {name}!</h1>
```

```js
import "svelte/internal/disclose-version";
import "svelte/internal/flags/legacy";
import * as $ from "svelte/internal/client";

var root = $.template(`<h1></h1>`);

export default function App($$anchor) {
  let name = 'world';
  var h1 = root();

  h1.textContent = `Hello ${name ?? ""}!`;
  $.append($$anchor, h1);
}
```

Bacaro
Tech
CODE AND FUN



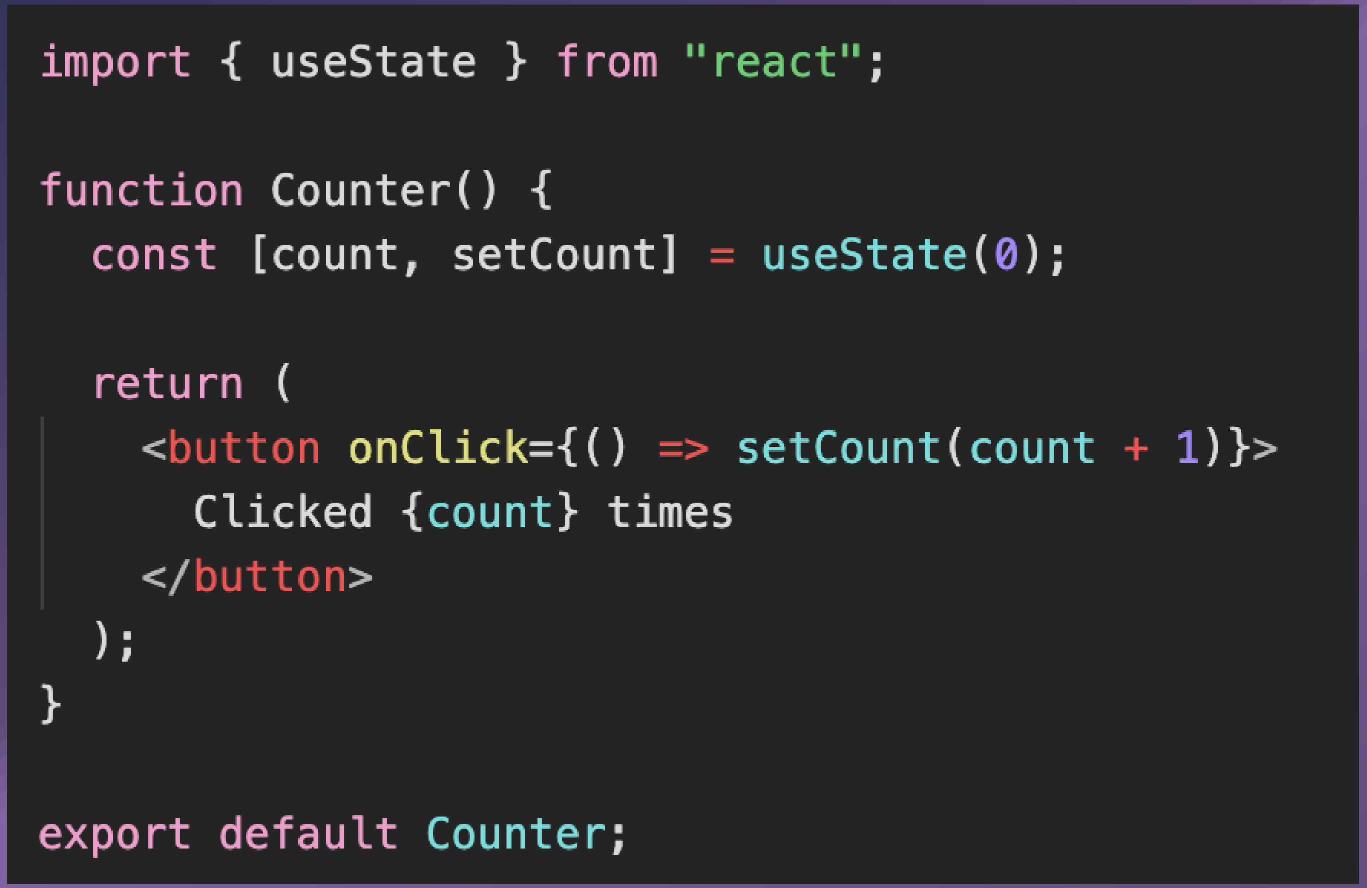
Compiled Component



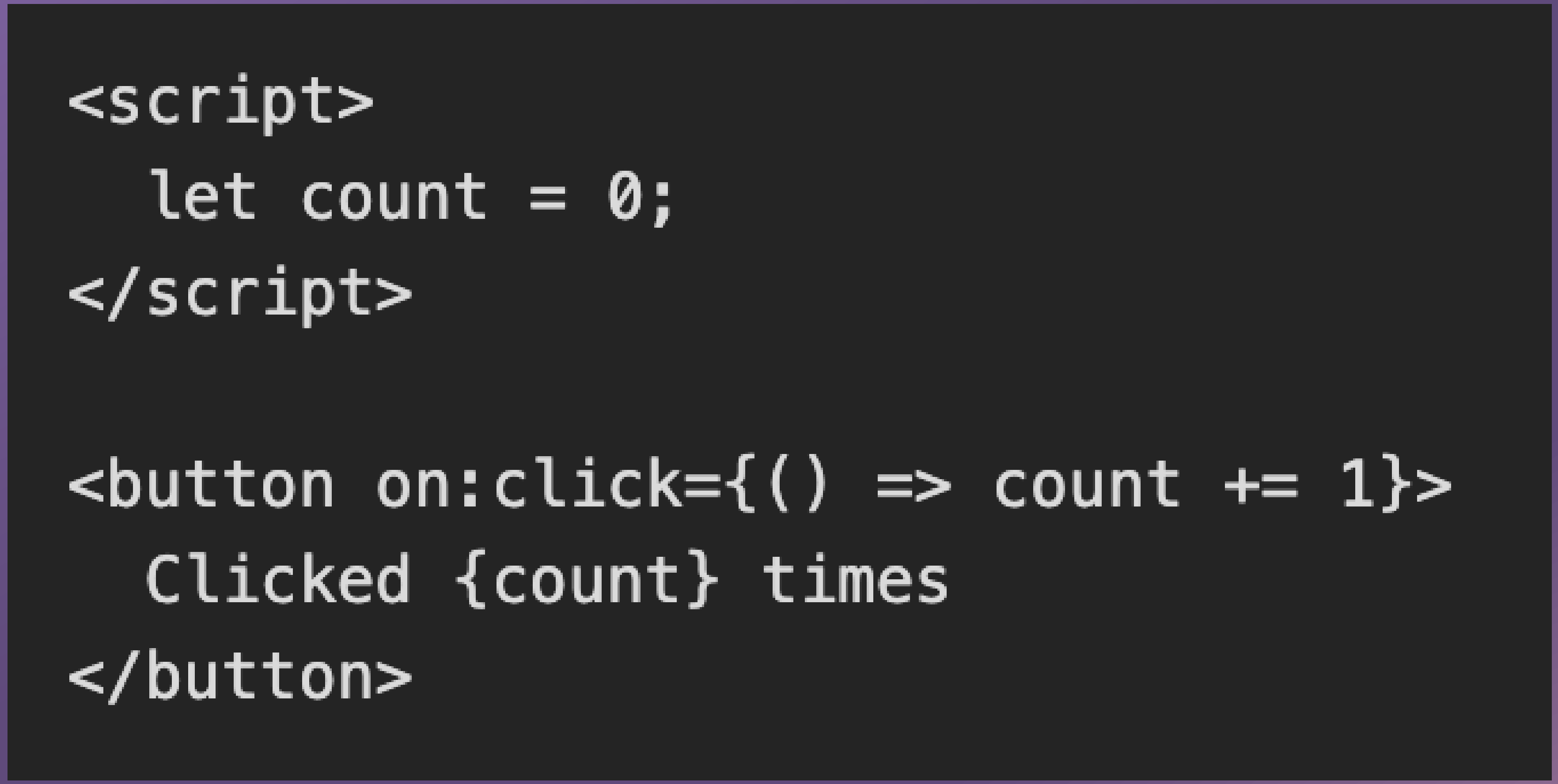
```
<script>
    let numbers = $state([1, 2, 3, 4]);

    function addNumber() {
        //numbers.push(numbers.length + 1);
        numbers[numbers.length] = numbers.length + 1;

    }
</script>

<p>{numbers.join(' + ')} = ...</p>

<button onclick={addNumber}>
    Add a number
</button>
```

```
import { createHotContext as __vite__createHotContext } from "/@vite/client";import.meta.

$.mark_module_start();
AddNumber[$.FILENAME] = 'src/lib/components/AddNumber.svelte';

import * as $ from "/node_modules/.vite/deps/svelte_internal_client.js?v=0dcc777b";

function addNumber(_, numbers) {
    //numbers.push(numbers.length + 1);
    numbers[numbers.length] = numbers.length + 1;
}

var root = $.add_locations($.template(`<p> </p> <button>Add a number</button>`, 1), AddNu

function AddNumber($$anchor, $$props) {
    $.check_target(new.target);
    $.push($$props, true, AddNumber);

    let numbers = $.proxy([1, 2, 3, 4]);
    var fragment = root();
    var p = $.first_child(fragment);
    var text = $.child(p);

    $.reset(p);

    var button = $.sibling(p, 2);

    button.__click = [addNumber, numbers];
    $.template_effect(($0) => $.set_text(text, `${$0 ?? ''} = ...`), [() => numbers.join(
    $.append($$anchor, fragment);
    return $.pop({ ...$.legacy_api() });
}

if (import.meta.hot) {
    AddNumber = $.hmr(AddNumber, () => AddNumber[$.HMR].source);

    import.meta.hot.acceptExports(["default"],(module) => {
        module.default[$.HMR].source = AddNumber[$.HMR].source;
        $.set(AddNumber[$.HMR].source, module.default[$.HMR].original);
    });
}

export default AddNumber;

$.mark_module_end(AddNumber);
$.delegate(['click']);
//# sourceMappingURL=data:application/json;base64,eyJ2ZXJzaW9uIjozLCJtYXBwaW5ncyI6Ijs7Ozs
```

Compilazione vs runtime

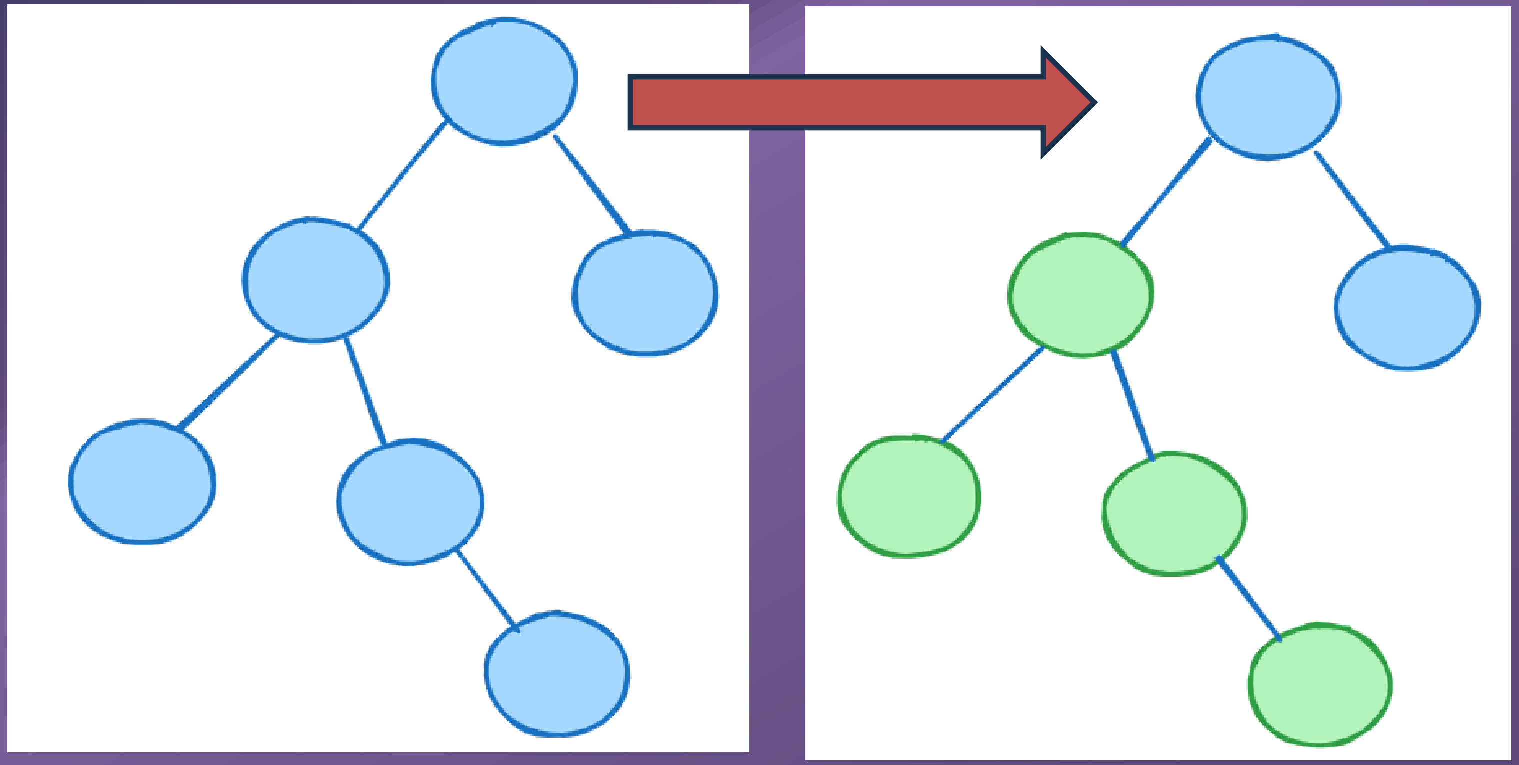


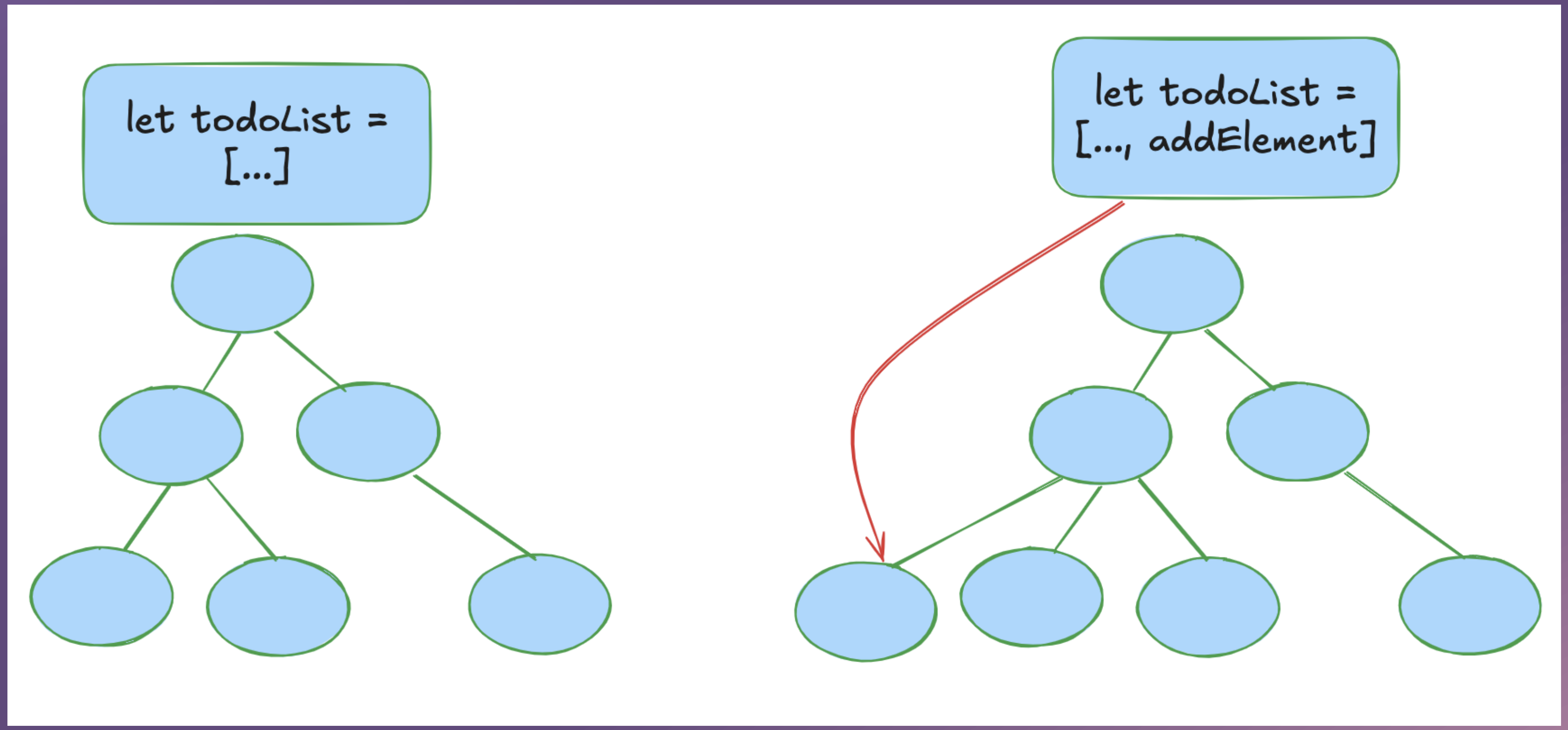
```
import { useState } from "react";

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <button onClick={() => setCount(count + 1)}>
      Clicked {count} times
    </button>
  );
}

export default Counter;
```

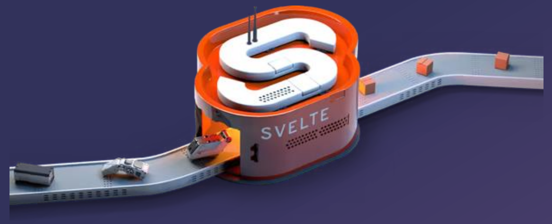


```
<script>
  let count = 0;
</script>

<button on:click={() => count += 1}>
  Clicked {count} times
</button>
```

Virtual Dom vs Aggiornamento diretto





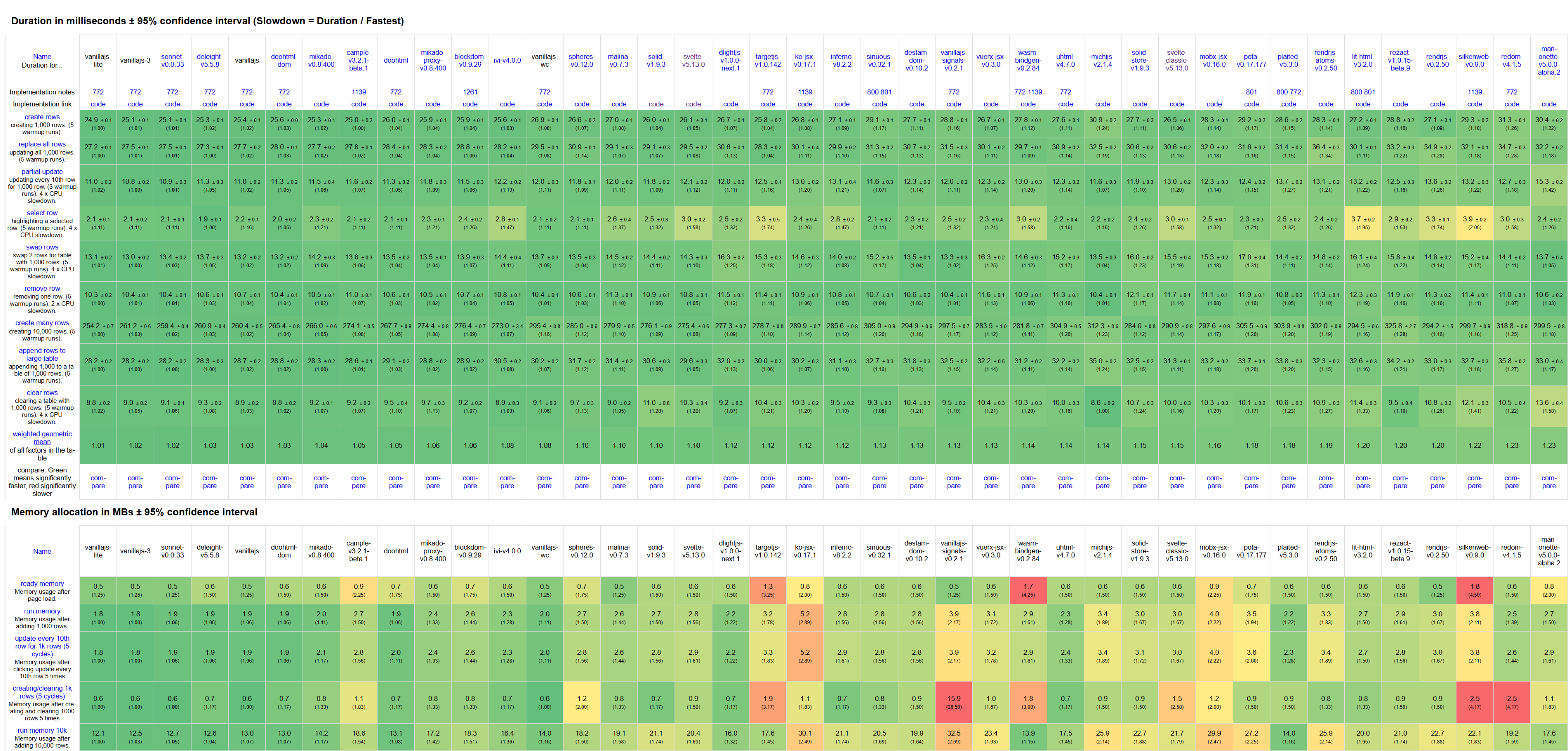


Comparisons

Bacaro
Tech
CODE AND FUN

Banchmark

https://github.com/krausest/js-framework-benchmark.git



Keyed implementations create an association between the domain data and a dom element by assigning a 'key'. If data changes the dom element with that key will be updated. In consequence inserting or deleting an element in the data array causes a corresponding change to the dom.

Duration in milliseconds ± 95% confidence interval (Slowdown = Duration / Fastest)

Name Duration for...	vanillajs-lite	vanillajs-3	sonnet-v0.0.33	deleight-v5.5.8	vanillajs	doohtml-dom	mikado-v0.8.400	cample-v3.2.1-beta.1	doohtml	mikado-proxy-v0.8.400
Implementation notes	772	772	772	772	772	772		1139	772	
Implementation link	code	code	code	code	code	code	code	code	code	code
create rows creating 1,000 rows. (5 warmup runs).	24.9 ± 0.1 (1.00)	25.1 ± 0.1 (1.01)	25.1 ± 0.1 (1.01)	25.3 ± 0.1 (1.02)	25.4 ± 0.1 (1.02)	25.6 ± 0.0 (1.03)	25.3 ± 0.1 (1.02)	25.0 ± 0.2 (1.00)	26.0 ± 0.1 (1.04)	25.9 ± 0.1 (1.04)
replace all rows updating all 1,000 rows. (5 warmup runs).	27.2 ± 0.1 (1.00)	27.5 ± 0.1 (1.01)	27.5 ± 0.1 (1.01)	27.3 ± 0.1 (1.00)	27.7 ± 0.2 (1.02)	28.0 ± 0.1 (1.03)	27.7 ± 0.2 (1.02)	27.8 ± 0.1 (1.02)	28.4 ± 0.1 (1.04)	28.3 ± 0.2 (1.04)
partial update updating every 10th row for 1,000 row. (3 warmup runs). 4 x CPU slowdown.	11.0 ± 0.2 (1.02)	10.8 ± 0.2 (1.00)	10.9 ± 0.3 (1.01)	11.3 ± 0.3 (1.05)	11.0 ± 0.2 (1.02)	11.3 ± 0.2 (1.05)	11.5 ± 0.4 (1.06)	11.6 ± 0.2 (1.07)	11.3 ± 0.2 (1.05)	11.8 ± 0.3 (1.09)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	2.1 ± 0.1 (1.11)	2.1 ± 0.2 (1.11)	2.1 ± 0.1 (1.11)	1.9 ± 0.1 (1.00)	2.2 ± 0.1 (1.16)	2.0 ± 0.2 (1.05)	2.3 ± 0.2 (1.21)	2.1 ± 0.2 (1.11)	2.1 ± 0.1 (1.11)	2.3 ± 0.1 (1.21)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	13.1 ± 0.2 (1.01)	13.0 ± 0.2 (1.00)	13.4 ± 0.2 (1.03)	13.7 ± 0.3 (1.05)	13.2 ± 0.2 (1.02)	13.2 ± 0.2 (1.02)	14.2 ± 0.3 (1.09)	13.8 ± 0.3 (1.06)	13.5 ± 0.2 (1.04)	13.5 ± 0.1 (1.04)
remove row removing one row. (5 warmup runs). 2 x CPU slowdown.	10.3 ± 0.2 (1.00)	10.4 ± 0.1 (1.01)	10.4 ± 0.1 (1.01)	10.6 ± 0.1 (1.03)	10.7 ± 0.1 (1.04)	10.4 ± 0.1 (1.01)	10.5 ± 0.1 (1.02)	11.0 ± 0.1 (1.07)	10.6 ± 0.1 (1.03)	10.5 ± 0.1 (1.02)
create many rows creating 10,000 rows. (5 warmup runs).	254.2 ± 0.7 (1.00)	261.2 ± 0.6 (1.03)	259.4 ± 0.4 (1.02)	260.9 ± 0.4 (1.03)	260.4 ± 0.5 (1.02)	265.4 ± 0.8 (1.04)	266.0 ± 0.6 (1.05)	274.1 ± 0.5 (1.08)	267.7 ± 0.9 (1.05)	274.4 ± 0.8 (1.08)
append rows to large table appending 1,000 to a table of 1,000 rows. (5 warmup runs).	28.2 ± 0.2 (1.00)	28.2 ± 0.2 (1.00)	28.2 ± 0.2 (1.00)	28.3 ± 0.3 (1.00)	28.7 ± 0.2 (1.02)	28.8 ± 0.2 (1.02)	28.3 ± 0.2 (1.00)	28.6 ± 0.1 (1.01)	29.1 ± 0.2 (1.03)	28.8 ± 0.2 (1.02)
clear rows clearing a table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	8.8 ± 0.2 (1.02)	9.0 ± 0.2 (1.05)	9.1 ± 0.1 (1.06)	9.3 ± 0.2 (1.08)	8.9 ± 0.2 (1.03)	8.8 ± 0.2 (1.02)	9.2 ± 0.1 (1.07)	9.2 ± 0.2 (1.07)	9.5 ± 0.4 (1.10)	9.7 ± 0.3 (1.13)
weighted geometric mean of all factors in the table	1.01	1.02	1.02	1.03	1.03	1.03	1.04	1.05	1.05	1.06
compare: Green means significantly faster, red significantly slower	compare	compare	compare	compare	compare	compare	compare	compare	compare	compare

Name Duration for...	blockdom-v0.9.29	ivi-v4.0.0	vanillajs-wc	spheres-v0.12.0	malina-v0.7.3	solid-v1.9.3	svelte-v5.13.0	dlightjs-v1.0.0-next.1	targetjs-v1.0.142	ko-jsx-v0.17.1
Implementation notes	1261		772						772	1139
Implementation link	code	code	code	code	code	code	code	code	code	code
create rows creating 1,000 rows. (5 warmup runs).	25.9 ± 0.1 (1.04)	25.6 ± 0.1 (1.03)	26.9 ± 0.1 (1.08)	26.6 ± 0.2 (1.07)	27.0 ± 0.1 (1.08)	26.0 ± 0.1 (1.04)	26.1 ± 0.1 (1.05)	26.7 ± 0.1 (1.07)	25.8 ± 0.2 (1.04)	26.8 ± 0.1 (1.08)
replace all rows updating all 1,000 rows. (5 warmup runs).	28.8 ± 0.1 (1.06)	28.2 ± 0.1 (1.04)	29.5 ± 0.1 (1.08)	30.9 ± 0.1 (1.14)	29.1 ± 0.3 (1.07)	29.1 ± 0.3 (1.07)	29.5 ± 0.2 (1.08)	30.6 ± 0.1 (1.13)	28.3 ± 0.2 (1.04)	30.1 ± 0.4 (1.11)
partial update updating every 10th row for 1,000 row. (3 warmup runs). 4 x CPU slowdown.	11.5 ± 0.3 (1.06)	12.2 ± 0.2 (1.13)	12.0 ± 0.3 (1.11)	11.8 ± 0.1 (1.09)	12.0 ± 0.2 (1.11)	11.8 ± 0.2 (1.09)	12.1 ± 0.2 (1.12)	12.0 ± 0.2 (1.11)	12.5 ± 0.1 (1.16)	13.0 ± 0.2 (1.20)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	2.4 ± 0.2 (1.26)	2.8 ± 0.1 (1.47)	2.1 ± 0.2 (1.11)	2.1 ± 0.1 (1.11)	2.6 ± 0.4 (1.37)	2.5 ± 0.3 (1.32)	3.0 ± 0.2 (1.58)	2.5 ± 0.2 (1.32)	3.3 ± 0.5 (1.74)	2.4 ± 0.4 (1.26)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	13.9 ± 0.3 (1.07)	14.4 ± 0.4 (1.11)	13.7 ± 0.3 (1.05)	13.5 ± 0.3 (1.04)	14.5 ± 0.2 (1.12)	14.4 ± 0.2 (1.11)	14.3 ± 0.3 (1.10)	16.3 ± 0.2 (1.25)	15.3 ± 0.3 (1.18)	14.6 ± 0.3 (1.12)
remove row removing one row. (5 warmup runs). 2 x CPU slowdown.	10.7 ± 0.1 (1.04)	10.8 ± 0.1 (1.05)	10.4 ± 0.1 (1.01)	10.6 ± 0.1 (1.03)	11.3 ± 0.1 (1.10)	10.9 ± 0.1 (1.06)	10.8 ± 0.1 (1.05)	11.5 ± 0.1 (1.12)	11.4 ± 0.1 (1.11)	10.9 ± 0.1 (1.06)
create many rows creating 10,000 rows. (5 warmup runs).	276.4 ± 0.7 (1.09)	273.0 ± 3.4 (1.07)	295.4 ± 0.8 (1.16)	285.0 ± 0.6 (1.12)	279.9 ± 0.5 (1.10)	276.1 ± 0.9 (1.09)	275.4 ± 0.6 (1.08)	277.3 ± 0.7 (1.09)	278.7 ± 0.8 (1.10)	289.9 ± 0.7 (1.14)
append rows to large table appending 1,000 to a table of 1,000 rows. (5 warmup runs).	28.9 ± 0.2 (1.02)	30.5 ± 0.2 (1.08)	30.2 ± 0.2 (1.07)	31.7 ± 0.2 (1.12)	31.4 ± 0.2 (1.11)	30.6 ± 0.3 (1.09)	29.6 ± 0.3 (1.05)	32.0 ± 0.2 (1.13)	30.0 ± 0.3 (1.06)	30.2 ± 0.3 (1.07)
clear rows clearing a table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	9.2 ± 0.2 (1.07)	8.9 ± 0.3 (1.03)	9.1 ± 0.2 (1.06)	9.7 ± 0.3 (1.13)	9.0 ± 0.2 (1.05)	11.0 ± 0.8 (1.28)	10.3 ± 0.4 (1.20)	9.2 ± 0.3 (1.07)	10.4 ± 0.3 (1.21)	10.3 ± 0.2 (1.20)
weighted geometric mean of all factors in the table	1.06	1.08	1.08	1.10	1.10	1.10	1.10	1.12	1.12	1.12
compare: Green means significantly faster, red significantly slower	compare	compare	compare	compare	compare	compare	compare	compare	compare	compare

Name Duration for...	inferno-v8.2.2	sinuous-v0.32.1	destam-dom-v0.10.2	vanillajs-signals-v0.2.1	vuerx-jsx-v0.3.0	wasm-bindgen-v0.2.84	uhtml-v4.7.0	michijs-v2.1.4	solid-store-v1.9.3	svelte-classic-v5.13.0
Implementation notes		800 801		772		772 1139	772			
Implementation link	code	code	code	code	code	code	code	code	code	code
create rows creating 1,000 rows. (5 warmup runs).	27.1 ± 0.1 (1.09)	29.1 ± 0.1 (1.17)	27.7 ± 0.1 (1.11)	28.8 ± 0.1 (1.16)	26.7 ± 0.1 (1.07)	27.8 ± 0.1 (1.12)	27.6 ± 0.1 (1.11)	30.9 ± 0.2 (1.24)	27.7 ± 0.3 (1.11)	26.5 ± 0.1 (1.06)
replace all rows updating all 1,000 rows. (5 warmup runs).	29.9 ± 0.2 (1.10)	31.3 ± 0.2 (1.15)	30.7 ± 0.2 (1.13)	31.5 ± 0.3 (1.16)	30.1 ± 0.2 (1.11)	29.7 ± 0.1 (1.09)	30.9 ± 0.2 (1.14)	32.5 ± 0.2 (1.19)	30.6 ± 0.2 (1.13)	30.6 ± 0.2 (1.13)
partial update updating every 10th row for 1,000 row. (3 warmup runs). 4 x CPU slowdown.	13.1 ± 0.4 (1.21)	11.6 ± 0.3 (1.07)	12.3 ± 0.2 (1.14)	12.0 ± 0.2 (1.11)	12.3 ± 0.2 (1.14)	13.0 ± 0.3 (1.20)	12.3 ± 0.2 (1.14)	11.6 ± 0.3 (1.07)	11.9 ± 0.3 (1.10)	13.0 ± 0.2 (1.20)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	2.8 ± 0.2 (1.47)	2.1 ± 0.2 (1.11)	2.3 ± 0.2 (1.21)	2.5 ± 0.2 (1.32)	2.3 ± 0.4 (1.21)	3.0 ± 0.2 (1.58)	2.2 ± 0.4 (1.16)	2.2 ± 0.2 (1.16)	2.4 ± 0.2 (1.26)	3.0 ± 0.1 (1.58)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	14.0 ± 0.2 (1.08)	15.2 ± 0.5 (1.17)	13.5 ± 0.1 (1.04)	13.3 ± 0.3 (1.02)	16.3 ± 0.2 (1.25)	14.6 ± 0.3 (1.12)	15.2 ± 0.3 (1.17)	13.5 ± 0.3 (1.04)	16.0 ± 0.2 (1.23)	15.5 ± 0.4 (1.19)
remove row removing one row. (5 warmup runs). 2 x CPU slowdown.	10.8 ± 0.1 (1.05)	10.7 ± 0.1 (1.04)	10.6 ± 0.2 (1.03)	10.4 ± 0.1 (1.01)	11.6 ± 0.1 (1.13)	10.9 ± 0.1 (1.06)	11.3 ± 0.1 (1.10)	10.4 ± 0.1 (1.01)	12.1 ± 0.1 (1.17)	11.7 ± 0.1 (1.14)
create many rows creating 10,000 rows. (5 warmup runs).	285.6 ± 0.8 (1.12)	305.0 ± 0.9 (1.20)	294.9 ± 0.6 (1.16)	297.5 ± 0.7 (1.17)	283.5 ± 1.0 (1.12)	281.8 ± 0.7 (1.11)	304.9 ± 0.5 (1.20)	312.3 ± 0.6 (1.23)	284.0 ± 0.8 (1.12)	290.9 ± 0.8 (1.14)
append rows to large table appending 1,000 to a table of 1,000 rows. (5 warmup runs).	31.1 ± 0.3 (1.10)	32.7 ± 0.3 (1.16)	31.8 ± 0.3 (1.13)	32.5 ± 0.2 (1.15)	32.2 ± 0.5 (1.14)	31.2 ± 0.2 (1.11)	32.2 ± 0.2 (1.14)	35.0 ± 0.2 (1.24)	32.5 ± 0.2 (1.15)	31.3 ± 0.1 (1.11)
clear rows clearing a table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	9.5 ± 0.2 (1.10)	9.3 ± 0.3 (1.08)	10.4 ± 0.3 (1.21)	9.5 ± 0.2 (1.10)	10.4 ± 0.3 (1.21)	10.3 ± 0.3 (1.20)	10.0 ± 0.3 (1.16)	8.6 ± 0.2 (1.00)	10.7 ± 0.3 (1.24)	10.0 ± 0.3 (1.16)
weighted geometric mean of all factors in the table	1.12	1.13	1.13	1.13	1.13	1.14	1.14	1.14	1.15	1.15
compare: Green means significantly faster, red significantly slower	compare	compare	compare	compare	compare	compare	compare	compare	compare	compare

Name Duration for...	mobx-jsx-v0.16.0	pota-v0.17.177	plaited-v5.3.0	rendrjs-atoms-v0.2.50	lit-html-v3.2.0	rezact-v1.0.15-beta.9	rendrjs-v0.2.50	silkenweb-v0.9.0	redom-v4.1.5	mari-onette-v5.0.0-alpha.2
Implementation notes		801	800 772		800 801			1139	772	
Implementation link	code	code	code	code	code	code	code	code	code	code
create rows creating 1,000 rows. (5 warmup runs).	28.3 ± 0.1 (1.14)	29.2 ± 0.2 (1.17)	28.6 ± 0.2 (1.15)	28.3 ± 0.1 (1.14)	27.2 ± 0.1 (1.09)	28.8 ± 0.2 (1.16)	27.1 ± 0.1 (1.09)	29.3 ± 0.2 (1.18)	31.3 ± 0.1 (1.26)	30.4 ± 0.2 (1.22)
replace all rows updating all 1,000 rows. (5 warmup runs).	32.0 ± 0.2 (1.18)	31.6 ± 0.2 (1.16)	31.4 ± 0.2 (1.15)	36.4 ± 0.3 (1.34)	30.1 ± 0.1 (1.11)	33.2 ± 0.3 (1.22)	34.9 ± 0.2 (1.28)	32.1 ± 0.1 (1.18)	34.7 ± 0.3 (1.28)	32.2 ± 0.2 (1.18)
partial update updating every 10th row for 1,000 row. (3 warmup runs). 4 x CPU slowdown.	12.3 ± 0.3 (1.14)	12.4 ± 0.2 (1.15)	13.7 ± 0.2 (1.27)	13.1 ± 0.2 (1.21)	13.2 ± 0.2 (1.22)	12.5 ± 0.3 (1.16)	13.6 ± 0.2 (1.26)	13.2 ± 0.3 (1.22)	12.7 ± 0.3 (1.18)	15.3 ± 0.2 (1.42)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	2.5 ± 0.1 (1.32)	2.3 ± 0.3 (1.21)	2.5 ± 0.2 (1.32)	2.4 ± 0.2 (1.26)	3.7 ± 0.2 (1.95)	2.9 ± 0.2 (1.53)	3.3 ± 0.1 (1.74)	3.9 ± 0.2 (2.05)	3.0 ± 0.3 (1.58)	2.4 ± 0.2 (1.26)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	15.3 ± 0.2 (1.18)	17.0 ± 0.4 (1.31)	14.4 ± 0.2 (1.11)	14.8 ± 0.2 (1.14)	16.1 ± 0.4 (1.24)	15.8 ± 0.4 (1.22)	14.8 ± 0.2 (1.14)	15.2 ± 0.4 (1.17)	14.4 ± 0.2 (1.11)	13.7 ± 0.4 (1.05)
remove row removing one row. (5 warmup runs). 2 x CPU slowdown.	11.1 ± 0.1 (1.08)	11.9 ± 0.1 (1.16)	10.8 ± 0.2 (1.05)	11.3 ± 0.1 (1.10)	12.3 ± 0.3 (1.19)	11.9 ± 0.1 (1.16)	11.3 ± 0.2 (1.10)	11.4 ± 0.1 (1.11)	11.0 ± 0.1 (1.07)	10.6 ± 0.2 (1.03)
create many rows creating 10,000 rows. (5 warmup runs).	297.6 ± 0.9 (1.17)	305.5 ± 0.9 (1.20)	303.9 ± 0.8 (1.20)	302.0 ± 0.9 (1.19)	294.5 ± 0.6 (1.16)	325.8 ± 2.7 (1.28)	294.2 ± 1.5 (1.16)	299.7 ± 0.9 (1.18)	318.8 ± 0.9 (1.25)	299.5 ± 0.8 (1.18)
append rows to large table appending 1,000 to a table of 1,000 rows. (5 warmup runs).	33.2 ± 0.2 (1.18)	33.7 ± 0.1 (1.20)	33.8 ± 0.3 (1.20)	32.3 ± 0.3 (1.15)	32.6 ± 0.3 (1.16)	34.2 ± 0.2 (1.21)	33.0 ± 0.3 (1.17)	32.7 ± 0.3 (1.16)	35.8 ± 0.2 (1.27)	33.0 ± 0.4 (1.17)
clear rows clearing a table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	10.3 ± 0.3 (1.20)	10.1 ± 0.2 (1.17)	10.6 ± 0.3 (1.23)	10.9 ± 0.3 (1.27)	11.4 ± 0.3 (1.33)	9.5 ± 0.4 (1.10)	10.8 ± 0.2 (1.26)	12.1 ± 0.3 (1.41)	10.5 ± 0.4 (1.22)	13.6 ± 0.4 (1.58)
weighted geometric mean of all factors in the table	1.16	1.18	1.18	1.19	1.20	1.20	1.20	1.22	1.23	1.23
compare: Green means significantly faster, red significantly slower	compare	compare	compare	compare	compare	compare	compare	compare	compare	compare

Memory allocation in MBs ± 95% confidence interval

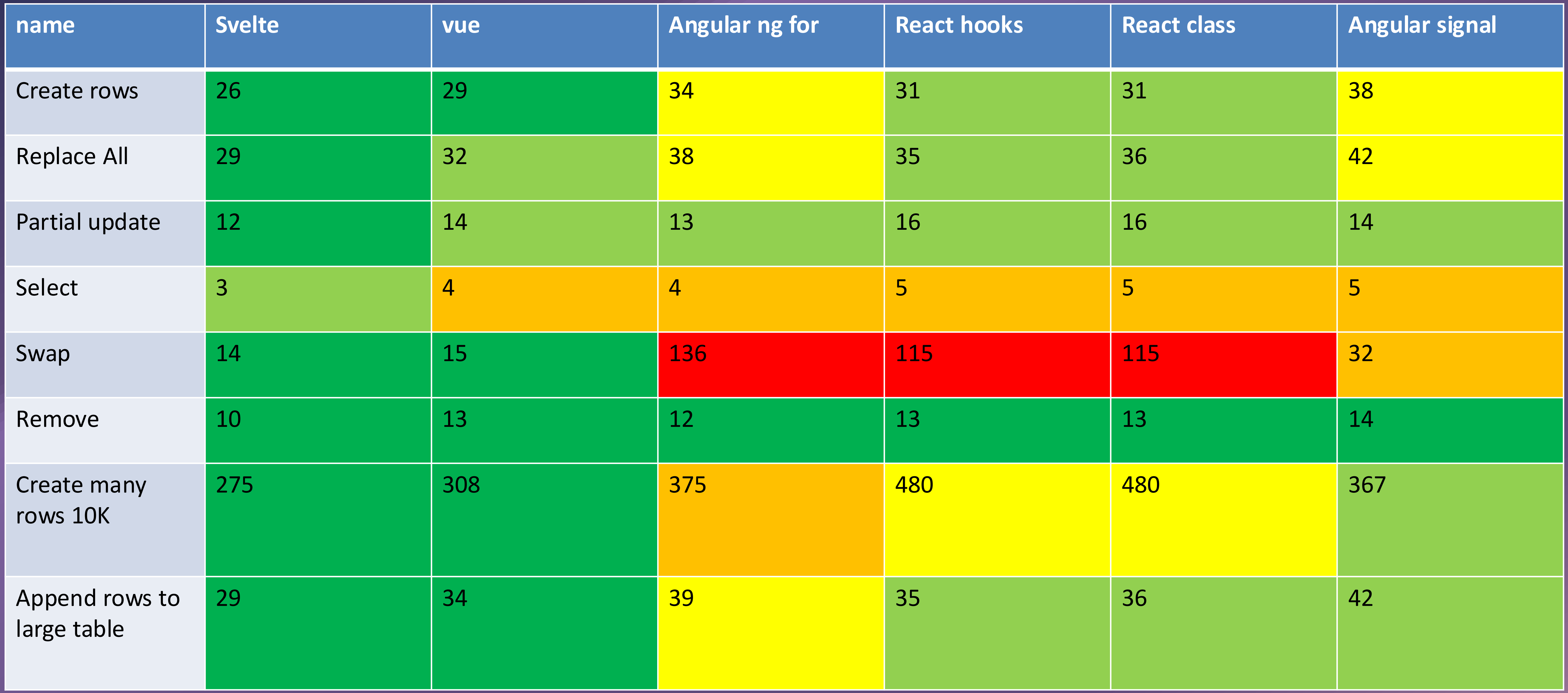
Name	vanillajs-lite	vanillajs-3	sonnet-v0.0.33	deleight-v5.5.8	vanillajs	doohtml-dom	mikado-v0.8.400	cample-v3.2.1-beta.1	doohtml	mikado-proxy-v0.8.400
ready memory Memory usage after page load.	0.5 (1.25)	0.5 (1.25)	0.5 (1.25)	0.6 (1.50)	0.5 (1.25)	0.6 (1.50)	0.6 (1.50)	0.9 (2.25)	0.7 (1.75)	0.6 (1.50)
run memory Memory usage after adding 1,000 rows.	1.8 (1.00)	1.8 (1.00)	1.9 (1.06)	1.9 (1.06)	1.9 (1.06)	1.9 (1.06)	2.0 (1.11)	2.7 (1.50)	1.9 (1.06)	2.4 (1.33)
update every 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	1.8 (1.00)	1.8 (1.00)	1.9 (1.06)	1.9 (1.06)	1.9 (1.06)	1.9 (1.06)	2.1 (1.17)	2.8 (1.56)	2.0 (1.11)	2.4 (1.33)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	0.6 (1.00)	0.6 (1.00)	0.6 (1.00)	0.7 (1.17)	0.6 (1.00)	0.7 (1.17)	0.8 (1.33)	1.1 (1.83)	0.7 (1.17)	0.8 (1.33)
run memory 10k Memory usage after adding 10,000 rows.	12.1 (1.00)	12.5 (1.03)	12.7 (1.05)	12.6 (1.04)	13.0 (1.07)	13.0 (1.07)	14.2 (1.17)	18.6 (1.54)	13.1 (1.08)	17.2 (1.42)
geometric mean of all factors in the table	1.05	1.05	1.08	1.15	1.08	1.16	1.25	1.71	1.21	1.38

Name	blockdom-v0.9.29	ivi-v4.0.0	vanillajs-wc	spheres-v0.12.0	malina-v0.7.3	solid-v1.9.3	svelte-v5.13.0	dlightjs-v1.0.0-next.1	targetjs-v1.0.142	ko-jsx-v0.17.1
ready memory Memory usage after page load.	0.7 (1.75)	0.6 (1.50)	0.5 (1.25)	0.7 (1.75)	0.5 (1.25)	0.6 (1.50)	0.6 (1.50)	0.6 (1.50)	1.3 (3.25)	0.8 (2.00)
run memory Memory usage after adding 1,000 rows.	2.6 (1.44)	2.3 (1.28)	2.0 (1.11)	2.7 (1.50)	2.6 (1.44)	2.7 (1.50)	2.8 (1.56)	2.2 (1.22)	3.2 (1.78)	5.2 (2.89)
update every 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	2.6 (1.44)	2.3 (1.28)	2.0 (1.11)	2.8 (1.56)	2.6 (1.44)	2.8 (1.56)	2.9 (1.61)	2.2 (1.22)	3.3 (1.83)	5.2 (2.89)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	0.8 (1.33)	0.7 (1.17)	0.6 (1.00)	1.2 (2.00)	0.8 (1.33)	0.7 (1.17)	0.9 (1.50)	0.7 (1.17)	1.9 (3.17)	1.1 (1.83)
run memory 10k Memory usage after adding 10,000 rows.	18.3 (1.51)	16.4 (1.36)	14.0 (1.16)	18.2 (1.50)	19.1 (1.58)	21.1 (1.74)	20.4 (1.69)	16.0 (1.32)	17.6 (1.45)	30.1 (2.49)
geometric mean of all factors in the table	1.49	1.31	1.12	1.65	1.41	1.48	1.57	1.28	2.18	2.38

Name	inferno-v8.2.2	sinuous-v0.32.1	destam-dom-v0.10.2	vanillajs-signals-v0.2.1	vuerx-jsx-v0.3.0	wasm-bindgen-v0.2.84	uhtml-v4.7.0	michijs-v2.1.4	solid-store-v1.9.3	svelte-classic-v5.13.0
ready memory Memory usage after page load.	0.6 (1.50)	0.6 (1.50)	0.6 (1.50)	0.5 (1.25)	0.6 (1.50)	1.7 (4.25)	0.6 (1.50)	0.6 (1.50)	0.6 (1.50)	0.6 (1.50)
run memory Memory usage after adding 1,000 rows.	2.8 (1.56)	2.8 (1.56)	2.8 (1.56)	3.9 (2.17)	3.1 (1.72)	2.9 (1.61)	2.3 (1.28)	3.4 (1.89)	3.0 (1.67)	3.0 (1.67)
update every 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	2.9 (1.61)	2.8 (1.56)	2.8 (1.56)	3.9 (2.17)	3.2 (1.78)	2.9 (1.61)	2.4 (1.33)	3.4 (1.89)	3.1 (1.72)	3.0 (1.67)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	0.7 (1.17)	0.8 (1.33)	0.9 (1.50)	15.9 (26.50)	1.0 (1.67)	1.8 (3.00)	0.7 (1.17)	0.9 (1.50)	0.9 (1.50)	1.5 (2.50)
run memory 10k Memory usage after adding 10,000 rows.	21.1 (1.74)	20.5 (1.69)	19.9 (1.64)	32.5 (2.69)	23.4 (1.93)	13.9 (1.15)	17.5 (1.45)	25.9 (2.14)	22.7 (1.88)	21.7 (1.79)
geometric mean of all factors in the table	1.50	1.52	1.55	3.34	1.71	2.07	1.34	1.77	1.65	1.80

Name	mobx-jsx-v0.16.0	pota-v0.17.177	plaited-v5.3.0	rendrjs-atoms-v0.2.50	lit-html-v3.2.0	rezact-v1.0.15-beta.9	rendrjs-v0.2.50	silkenweb-v0.9.0	redom-v4.1.5	mari-onette-v5.0.0-alpha.2
ready memory Memory usage after page load.	0.9 (2.25)	0.7 (1.75)	0.6 (1.50)	0.6 (1.50)	0.6 (1.50)	0.6 (1.50)	0.5 (1.25)	1.8 (4.50)	0.6 (1.50)	0.8 (2.00)
run memory Memory usage after adding 1,000 rows.	4.0 (2.22)	3.5 (1.94)	2.2 (1.22)	3.3 (1.83)	2.7 (1.50)	2.9 (1.61)	3.0 (1.67)	3.8 (2.11)	2.5 (1.39)	2.7 (1.50)
update every 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	4.0 (2.22)	3.6 (2.00)	2.3 (1.28)	3.4 (1.89)	2.7 (1.50)	2.8 (1.56)	3.0 (1.67)	3.8 (2.11)	2.6 (1.44)	2.9 (1.61)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	1.2 (2.00)	0.9 (1.50)	0.9 (1.50)	0.8 (1.33)	0.8 (1.33)	0.9 (1.50)	0.9 (1.50)	2.5 (4.17)	2.5 (4.17)	1.1 (1.83)
run memory 10k Memory usage after adding 10,000 rows.	29.9 (2.47)	27.2 (2.25)	14.0 (1.16)	25.9 (2.14)	20.0 (1.65)	21.0 (1.74)	22.7 (1.88)	22.1 (1.83)	19.2 (1.59)	17.6 (1.45)
geometric mean of all factors in the table	2.23	1.87	1.32	1.71	1.49	1.58	1.58	2.73	1.82	1.67

Banchmark

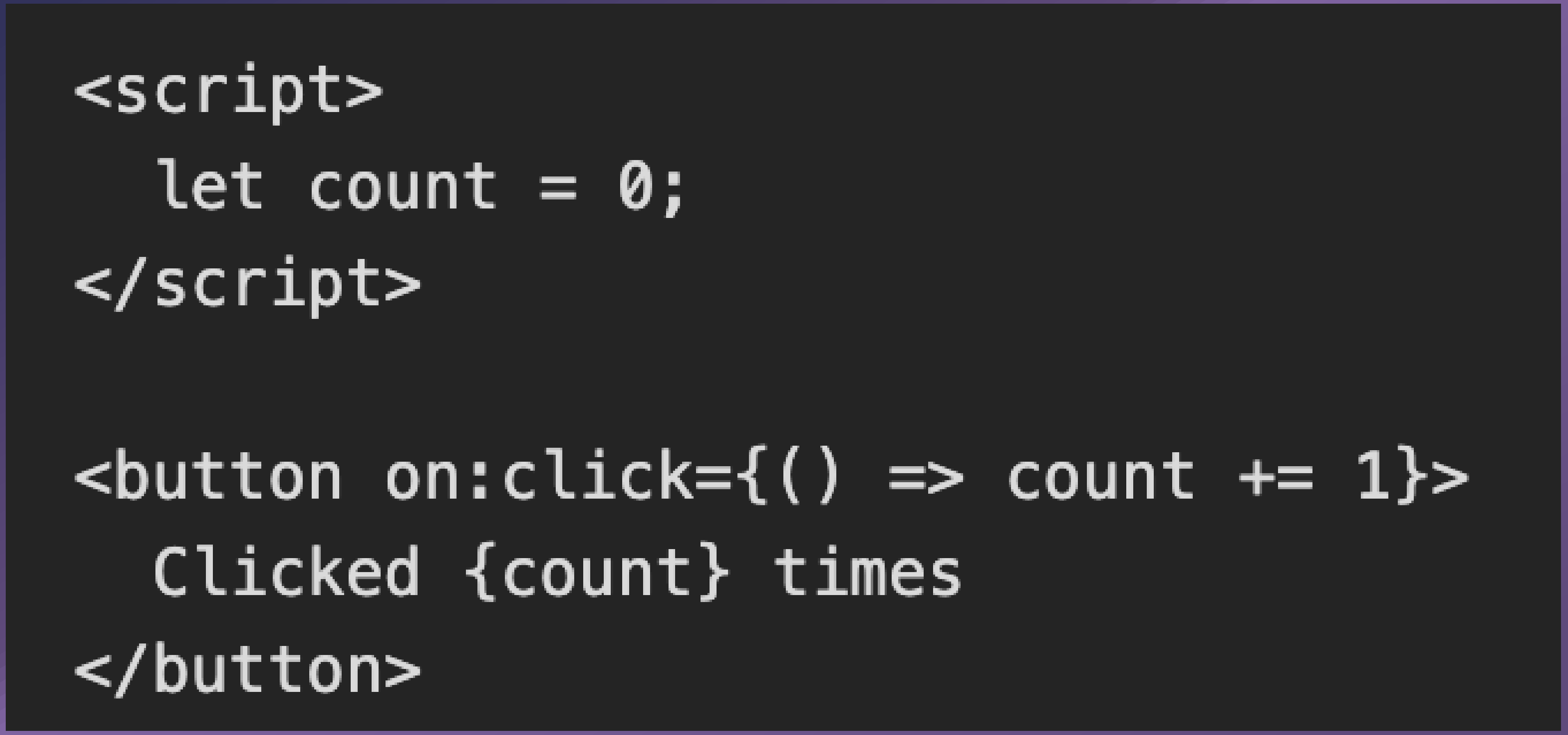


name	Svelte	vue	Angular ng for	React hooks	React class	Angular signal
Create rows	26	29	34	31	31	38
Replace All	29	32	38	35	36	42
Partial update	12	14	13	16	16	14
Select	3	4	4	5	5	5
Swap	14	15	136	115	115	32
Remove	10	13	12	13	13	14
Create many rows 10K	275	308	375	480	480	367
Append rows to large table	29	34	39	35	36	42

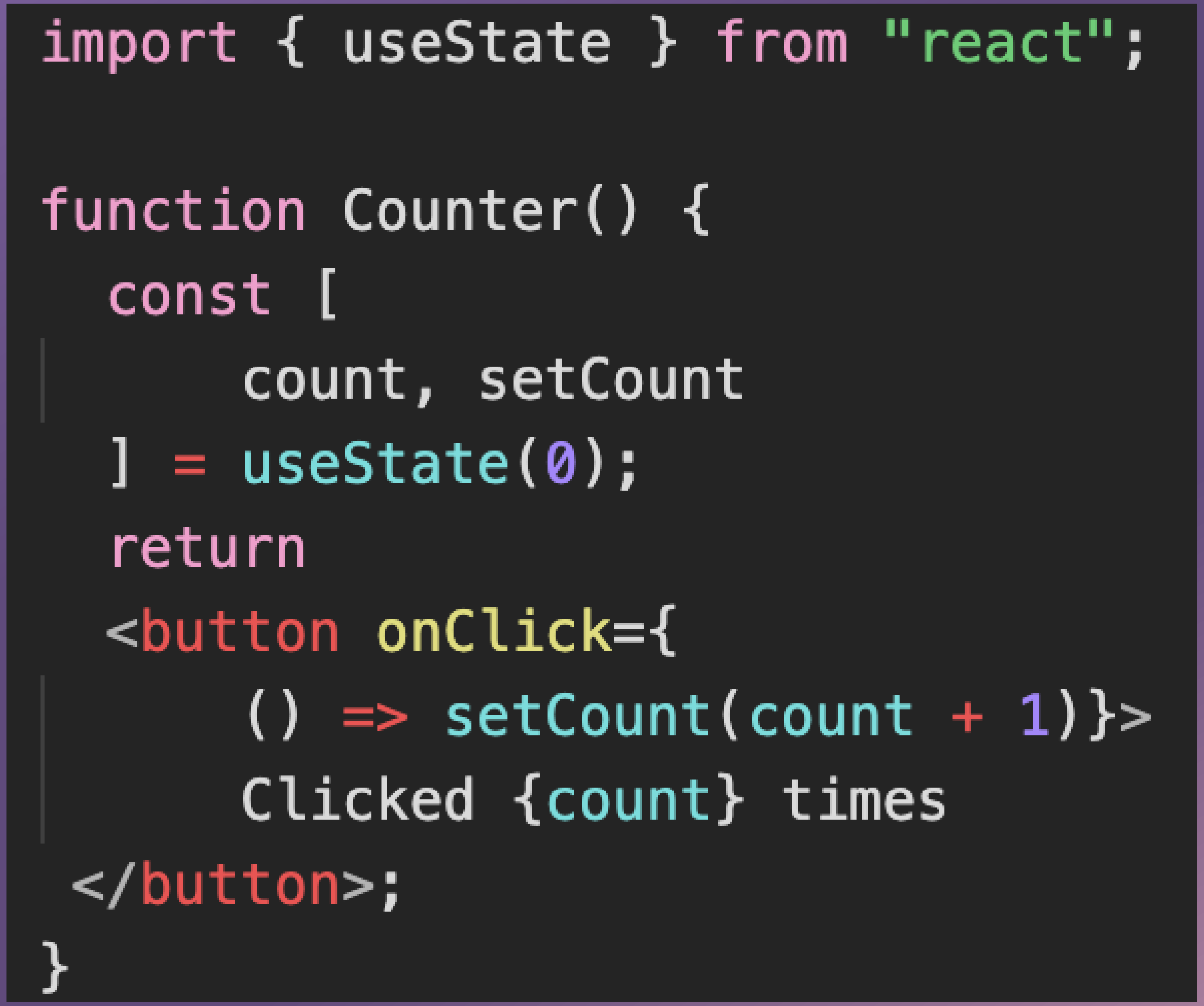
Architectures

SvelteKit	**Next.js**	**Angular**
Compilato, SSR/SPA	Runtime	Runtime
SSR/SPA/SSG	SSR/SPA/SSG	SPA, SSR, SSG
è il più leggero ed efficiente, ma ancora giovane.	offre un buon bilanciamento tra performance ed ecosistema.	è più adatto a progetti enterprise di grandi dimensioni.

Reactivity



```
<script>
  let count = 0;
</script>

<button on:click={() => count += 1}>
  Clicked {count} times
</button>
```



```
import { useState } from "react";

function Counter() {
  const [
      count, setCount
  ] = useState(0);
  return
  <button onClick={
      () => setCount(count + 1)}>
      Clicked {count} times
 </button>;
}
```

Reactivity

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-counter',
  template: `<button (click)="increment()">
  Clicked {{ count }} times</button>`
})
export class CounterComponent {
  count = 0;
  increment() { this.count += 1; }
}
```

```
import { Component } from '@angular/core';
import { signal } from '@angular/core';

@Component({
  selector: 'app-counter',
  template: `<button (click)="increment()">
    Clicked {{ count() }} times
  </button>`
})
export class CounterComponent {
  count = signal(0);

  increment() {
    this.count.update(value => value + 1);
  }
}
```

Performance e Bundle Size

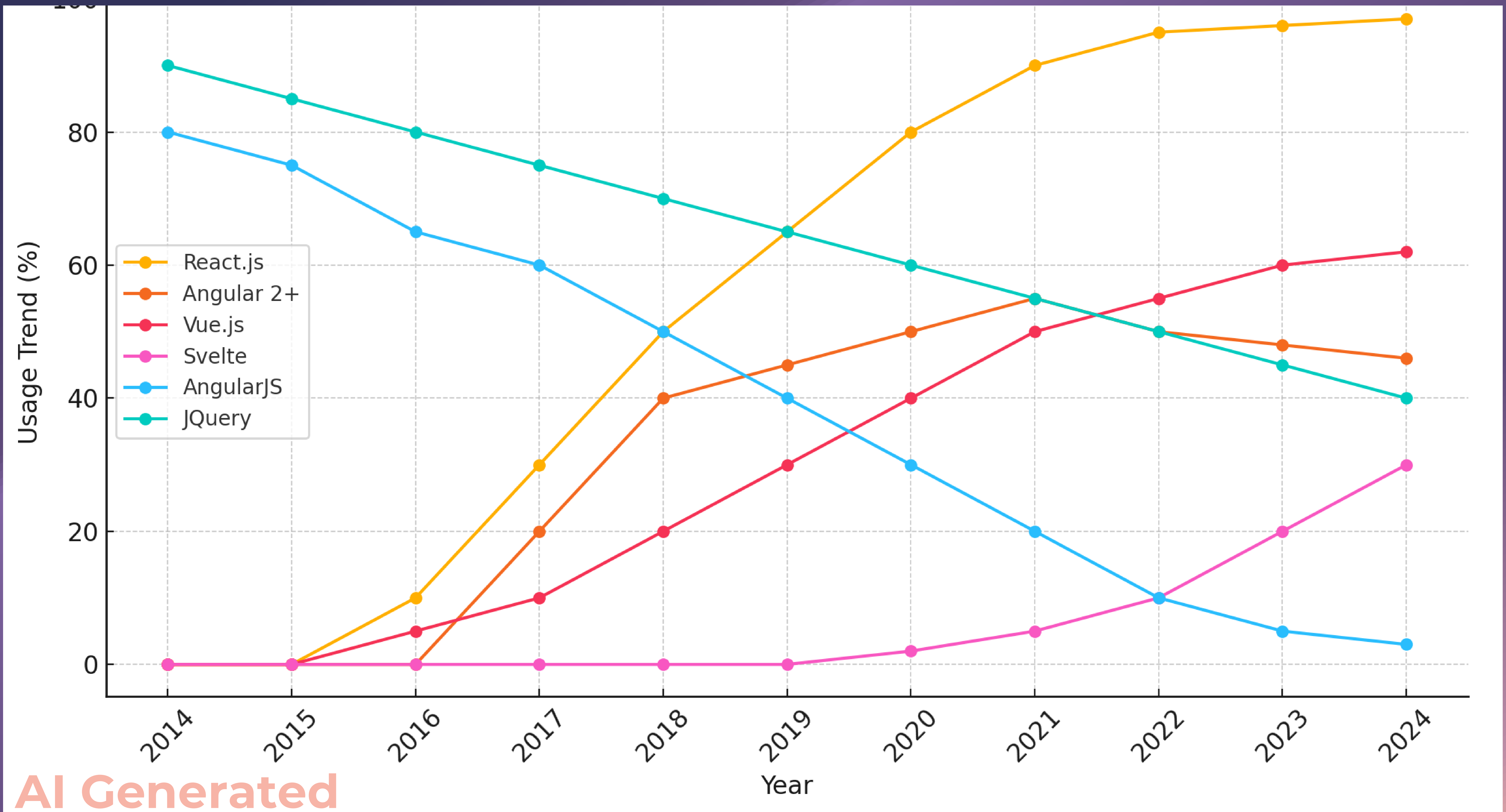
	SvelteKit	Next.js	Angular
Aggiornamento UI	⚡ Istantaneo	🐌 Più lento (Virtual DOM)	🏗️ Più lento (Zone.js)
Bundle Size	📉 5-10 KB	📈 40-50 KB	🏗️ 100+ KB
Risultato	è il più veloce e leggero.	è un compromesso tra flessibilità e performance.	ha il bundle più grande e il ciclo di aggiornamento più pesante

Use Case & DX

	SvelteKit	Next.js	Angular
Code			
App leggere e veloci	✅ Ottimo	🟠 Buono	❌ No
Progetti full-stack	🟠 Buono	✅ Ottimo	❌ No
App enterprise	❌ No	🟠 Buono	✅ Ottimo
Ecosistema e community	🟠 In crescita	✅ Maturo	✅ Maturo
Facilità di sviluppo	✅ Facile	🟠 Media	❌ Complesso

Bacaro
Tech
CODE AND FUN

JS Framework world



AI Generated

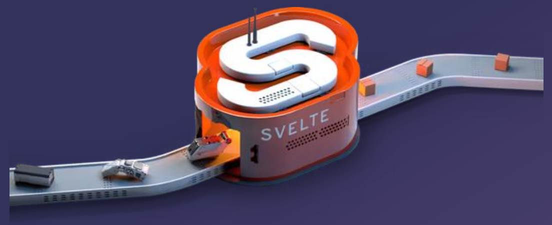
Ecosystem & Community

Google
Enterprise

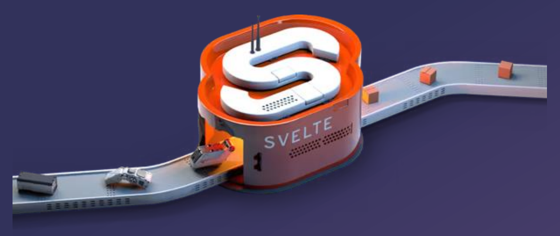
Meta
Largest community

Strong Community
Enterprise and small project

Ecosystem: Grows fast
Dedicated Community



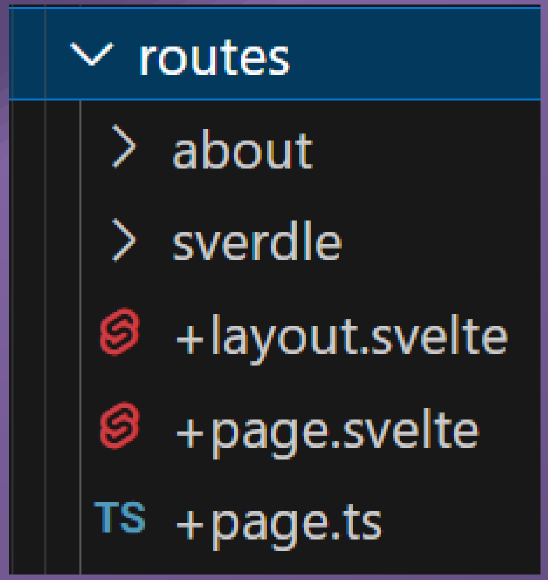
SSR SSG CSR



Routing

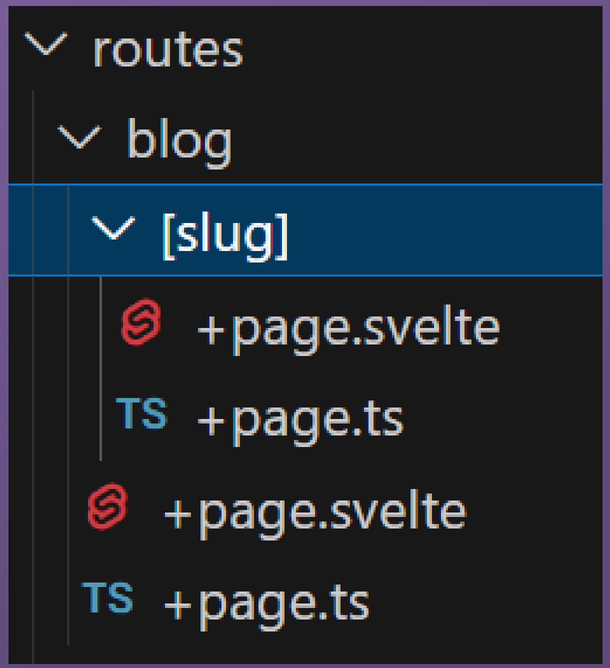
Svelte KIT

Route



- routes
  - about
  - sverdle
  - +layout.svelte
  - +page.svelte
  - +page.ts

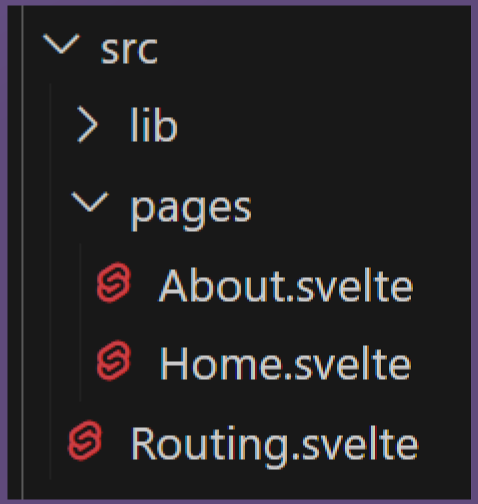
Sub Route



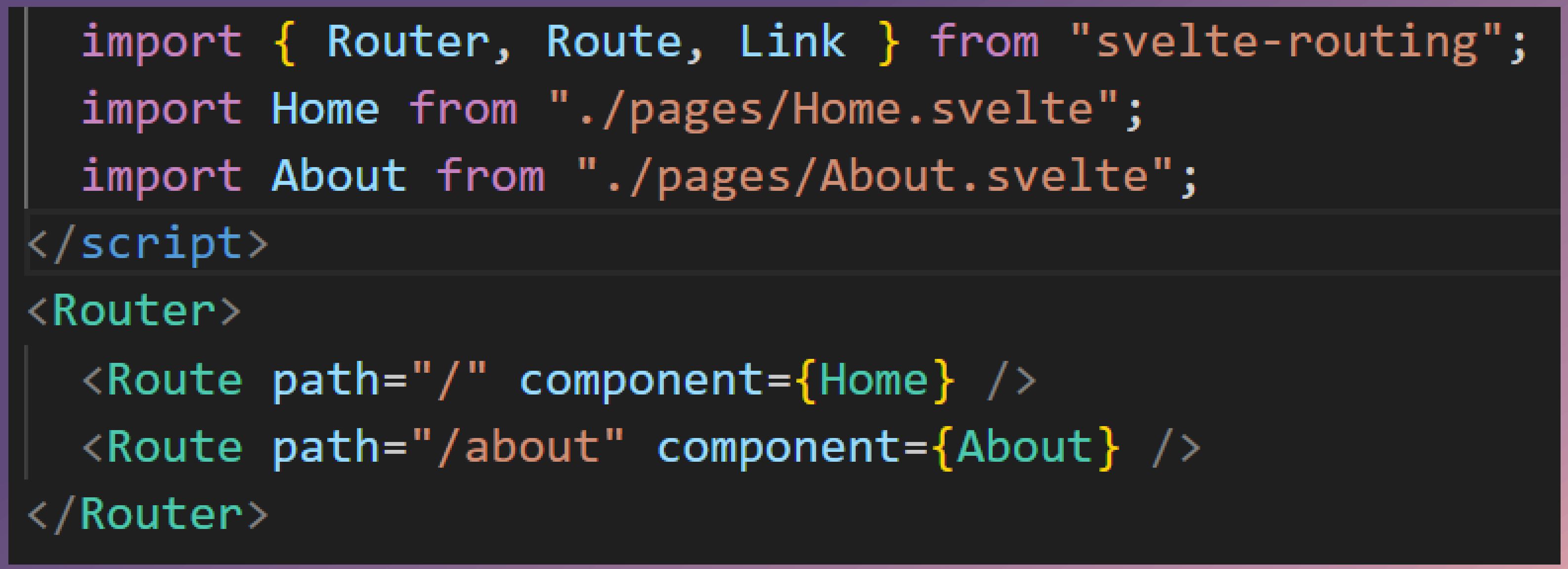
- routes
  - blog
    - [slug]
      - +page.svelte
      - +page.ts
    - +page.svelte
    - +page.ts

External Library

svelte-routing

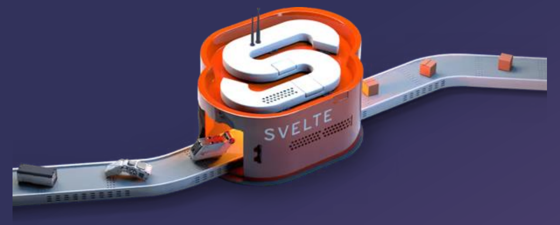


- src
  - lib
  - pages
    - About.svelte
    - Home.svelte
  - Routing.svelte



```
  import { Router, Route, Link } from "svelte-routing";
  import Home from "./pages/Home.svelte";
  import About from "./pages/About.svelte";
</script>
<Router>
  <Route path="/" component={Home} />
  <Route path="/about" component={About} />
</Router>
```

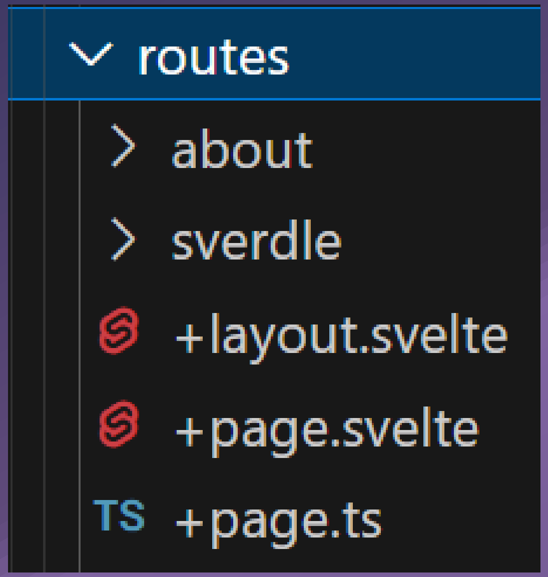
Bacaro
Tech
CODE AND FUN



Routing

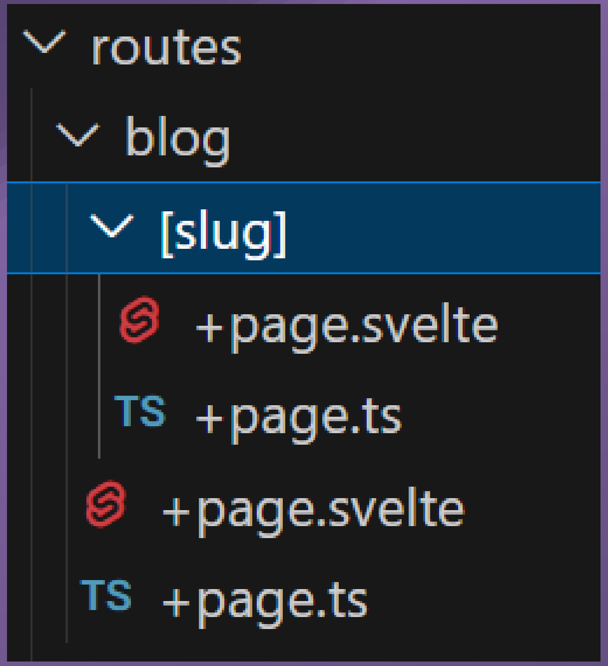
Svelte KIT

Route

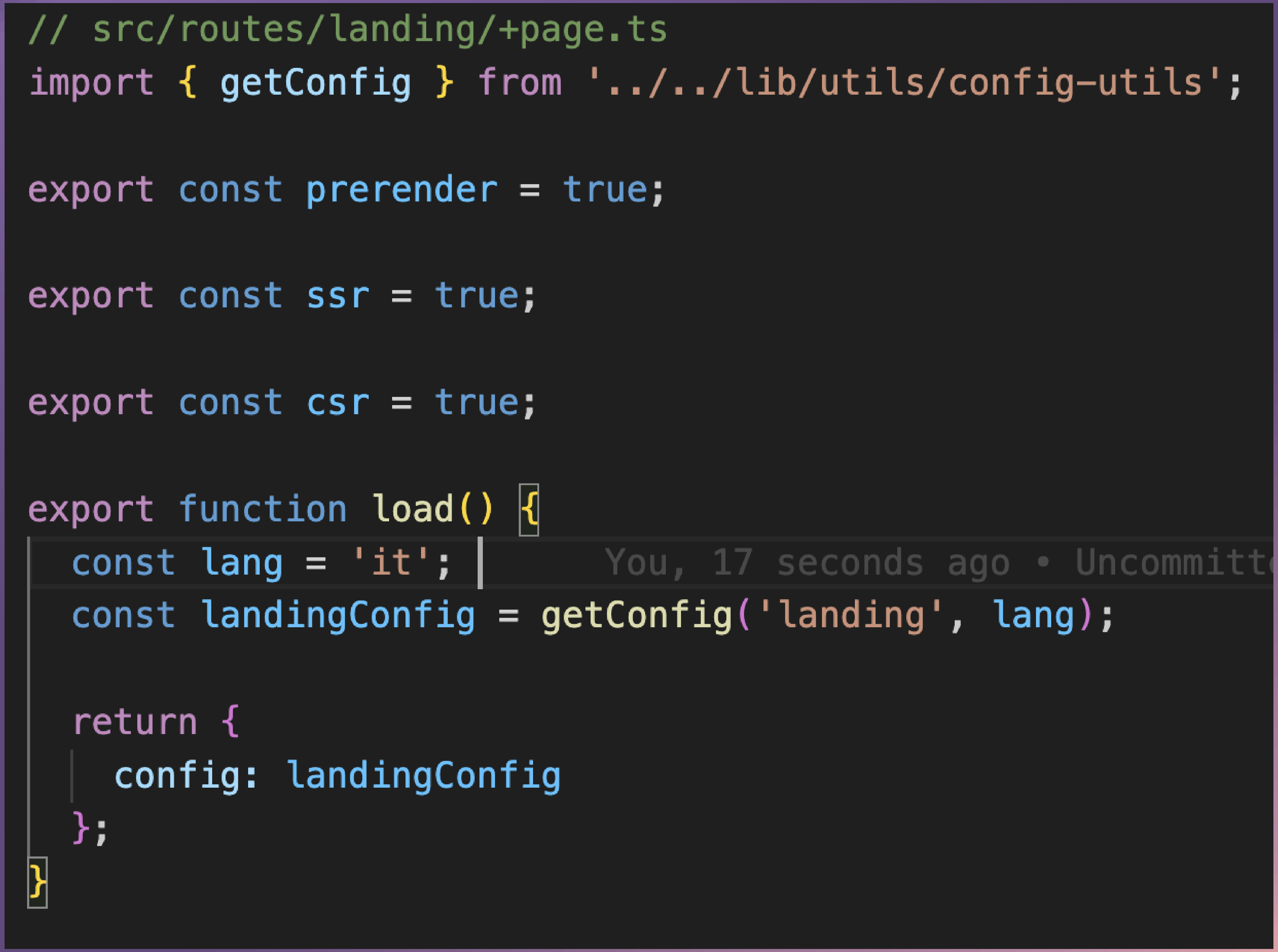


- routes
  - about
  - sverdle
  - +layout.svelte
  - +page.svelte
  - +page.ts

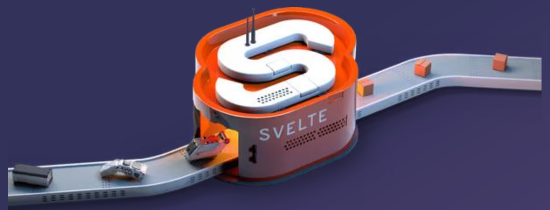
Sub Route



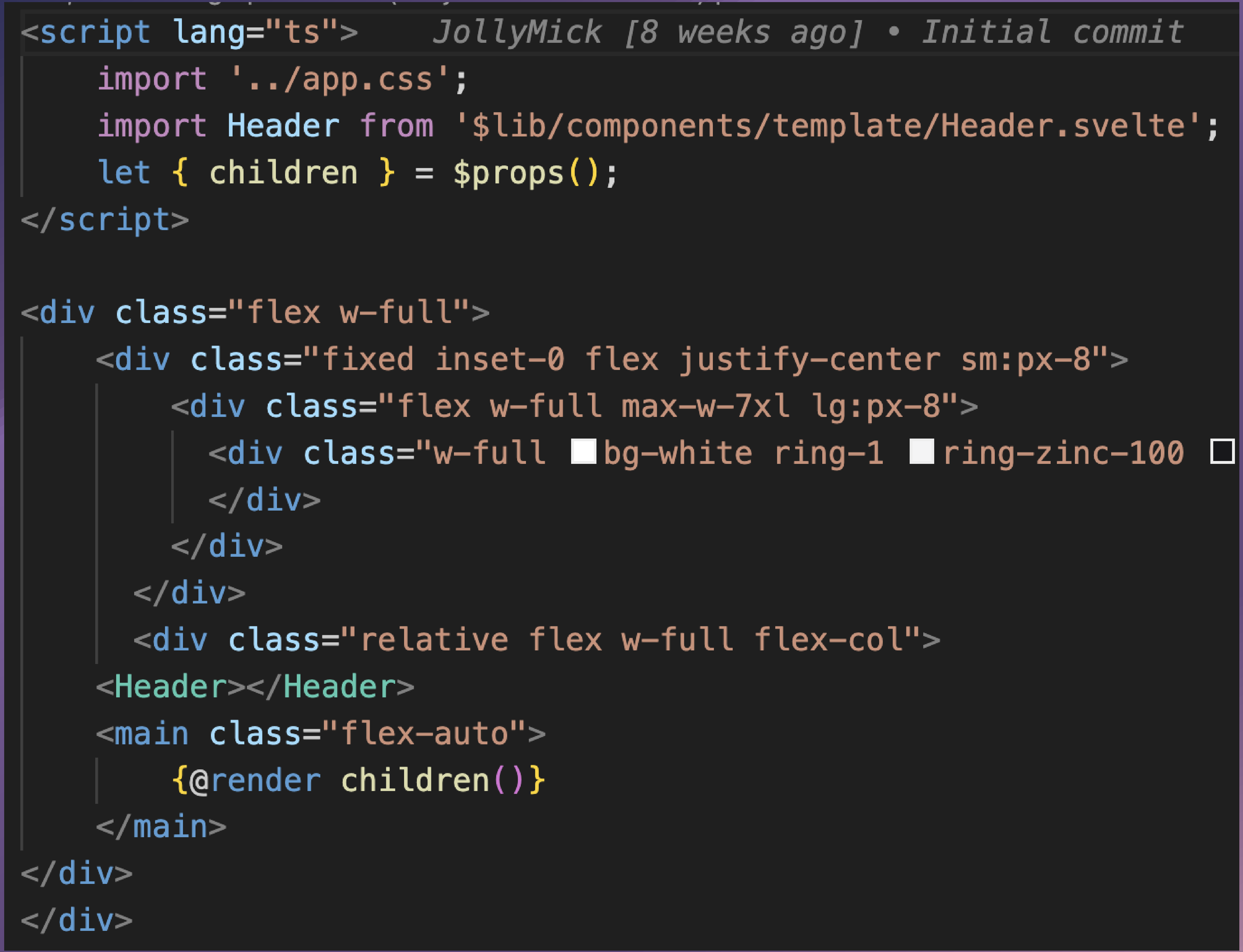
- routes
  - blog
    - [slug]
      - +page.svelte
      - +page.ts
    - +page.svelte
    - +page.ts



```
// src/routes/landing/+page.ts
import { getConfig } from '../../lib/utils/config-utils';

export const prerender = true;

export const ssr = true;

export const csr = true;

export function load() {
  const lang = 'it';        You, 17 seconds ago • Uncommitte
  const landingConfig = getConfig('landing', lang);

  return {
    config: landingConfig
  };
}
```



Layout



```
<script lang="ts">    JollyMick [8 weeks ago] • Initial commit
    import '../app.css';
    import Header from '$lib/components/template/Header.svelte';
    let { children } = $props();
</script>

<div class="flex w-full">
    <div class="fixed inset-0 flex justify-center sm:px-8">
        <div class="flex w-full max-w-7xl lg:px-8">
          <div class="w-full ■bg-white ring-1 ■ring-zinc-100 □
          </div>
        </div>
      </div>
      <div class="relative flex w-full flex-col">
    <Header></Header>
    <main class="flex-auto">
        {@render children()}
    </main>
</div>
</div>
```

CSR SSR SSG

```
export const csr = true;
export const ssr = true;
export const prerender =
true;
```

Rendering client side

Rendering server side

Prerendering

Bacaro
Tech
CODE AND FUN

CSR SSR SSG

SPA -> all client side

```
export const csr = true, ssr = false, prerender = false;
```

SSR puro-> all server side

```
export const csr = false, ssr = true, prerender = false;
```

SSR + Hydration -> start server side, then client side

```
export const csr = true, ssr = true, prerender = false;
```

SSG puro -> prebuild

```
export const csr = false, ssr = false, prerender = true;
```

GRAZIE DELLA VOSTRA ATTENZIONE!