

SVELTE 5

What's new, and why did it change?

WHY??

REACTIVITY ABSTRACTION - FINE GRAINED REACTIVITY?

```
$:  
<!----->
```

CAN YOU SPOT THE BUG?

```
1 <script>
2   let count = 1;
3   $: double = count * 2;
4   console.log(double);
5 </script>
6
7 <button on:click={() => {
8   count++;
9   console.log(count, double);
10 }}>count: {count}, double: {double}</button><!-- -->
```

Solution

CAN YOU SPOT THE BUG?

```
1 <script>
2   let count = 1;
3   $: double = count * 2;
4   console.log(double);
5 </script>
6
7 <button on:click={() => {
8   count++;
9   console.log(count, double);
10 }}>count: {count}, double: {double}</button><!-- -->
```

Solution

CAN YOU SPOT THE BUG?

```
1 <script>
2   let count = 1;
3   $: double = count * 2;
4   console.log(double);
5 </script>
6
7 <button on:click={() => {
8   count++;
9   console.log(count, double);
10 }}>count: {count}, double: {double}</button><!-- -->
```

Solution

RUNES

A letter or mark used as a mystical or magic symbol.

Runes have a "\$" prefix and look like functions:

```
<script>  
  let message = $state('hello');  
</script>  
<!-- -->
```

Now you can also use svelte runes in ".svelte.js" or ".svelte.ts", it's not only limited to ".svelte" files

RUNES STATE - CLASSES

```
1  class Todo {  
2    done = $state(false);  
3    text = $state();  
4  
5    constructor(text) {  
6      this.text = text;  
7    }  
8  
9    reset() {  
10     this.text = '';  
11     this.done = false;  
12   }  
13 }  
14 <!-- -->
```


SIDE EFFECT & DERIVED STATE

In Svelte 5, there is a separation between side effects
and derived state

Prior we used the "\$:" to express side effects and
derived states

RUNES DERIVED

```
1 <script>
2   let count = $state(0);
3   let doubled = $derived(count * 2);
4 </script>
5
6 <button onclick={() => count++}>
7   {doubled}
8 </button>
9
10 <p>{count} doubled is {doubled}</p>
11 <!-- -->
```

Back to problem

RUNES EFFECTS

```
1 <script>
2   let size = $state(50);
3   let color = $state('#ff3e00');
4
5   let canvas;
6
7   $effect(() => {
8     const context = canvas.getContext('2d');
9     context.clearRect(0, 0, canvas.width, canvas.height);
10
11     // this will re-run whenever 'color' or 'size' change
12     context.fillStyle = color;
13     context.fillRect(0, 0, size, size);
14   });
```

avoid overusing it!!

SNIPPETS REPLACE "SLOTS"

```
1 <script>
2     let { children } = $props();
3 </script>
4
5 {@render children?.()}
6 <!-- -->
```

SNIPPETS CREATE REUSABLE CHUNKS OF MARKUP

```
1 <script>
2   let { message = 'it's great to see you!' } = $props();
3 </script>
4
5 {#snippet hello(name)}
6   <p>hello {name}! {message}</p>
7 {/snippet}
8
9 {@render hello('alice')}
10 {@render hello('bob')}
11 <!-- -->
```

EVENTS

Pump.svelte



```
<script>
  import { createEventDispatcher } from 'svelte';
  const dispatch = createEventDispatcher();

  let { inflate, deflate } = $props();
  let power = $state(5);
</script>

<button onclick={() => dispatch('inflate', power)inflate(power)}>
  inflate
</button>
<button onclick={() => dispatch('deflate', power)deflate(power)}>
  deflate
</button>
<button onclick={() => power--}>-</button>
Pump power: {power}
<button onclick={() => power++}>+</button>
```

COMPATIBILITY

It is possible to use Svelte 3 components in Svelte 5
(but probably not in Svelte 6)

```
"devDependencies": {  
  "@sveltejs/vite-plugin-svelte": "^3.0.0",  
  "svelte": "^4",  
  // ...  
}  
}<!-- -->
```

```
"devDependencies": {  
  "@sveltejs/vite-plugin-svelte": "^4.0.0",  
  "svelte": "^5",  
  // ...  
}  
}<!-- -->
```

MIGRATION

From Svelte 4 to Svelte 5, there is a migration script

```
1 npx sv migrate  
2 <!-- -->
```


PERFORMANCE

Efficient code with signals

Efficient code with signals

[illegible]