SVELTE • TEMPLATE SYNTAX

use:

ON THIS PAGE

Actions are functions that are called when an element is mounted. They are added with the `use:` directive, and will typically use an `$effect` so that they can reset any state when the element is unmounted:

App.svelte

JS TS

```
<script lang="ts">
  import type { Action } from 'svelte/action';

  const myaction: Action = (node) => {
    // the node has been mounted in the DOM

    $effect(() => {
      // setup goes here

      return () => {
        // teardown goes here
      };
    });
  };
</script>

<div use:myaction>...</div>
```

An action can be called with an argument:

App.svelte

JS TS

```
<script lang="ts">
  import type { Action } from 'svelte/action';
```

```
</script>

<div use:myaction={data}>...</div>
```

The action is only called once (but not during server-side rendering) — it will *not* run again if the argument changes.

Legacy mode show all

Typing

The `Action` interface receives three optional type arguments — a node type (which can be `Element`, if the action applies to everything), a parameter, and any custom event handlers created by the action:

App.svelte JS TS

```
<script lang="ts">
  import { on } from 'svelte/events';
  import type { Action } from 'svelte/action';

  const gestures: Action<
    HTMLDivElement,
    null,
    {
      onswiperight: (e: CustomEvent) => void;
      onswipeleft: (e: CustomEvent) => void = (node) => {
    $effect(() => {
      // ...
      node.dispatchEvent(new CustomEvent('swipeleft'));

      // ...
      node.dispatchEvent(new CustomEvent('swiperight'));
    });
  };
</script>
```

```
>...</div>
```

Edit this page on GitHub

PREVIOUS
bind:

NEXT
transition: