SVELTE • TEMPLATE SYNTAX

animate:

ON THIS PAGE

An animation is triggered when the contents of a keyed each block are re-ordered. Animations do not run when an element is added or removed, only when the index of an existing data item within the each block changes. Animate directives must be on an element that is an *immediate* child of a keyed each block.

Animations can be used with Svelte's built-in animation functions or custom animation functions.

```
<!-- When `list` is reordered the animation will run -->
{#each list as item, index (item)}
  <li animate:flip>{item}</li>
{/each}
```

Animation Parameters

As with actions and transitions, animations can have parameters.

(The double `{{curlies}}` aren't a special syntax; this is an object literal inside an expression tag.)

```
{#each list as item, index (item)}
  <li animate:flip={{ delay: 500 }}>{item}</li>
{/each}
```

```
  delay?: number,
  duration?: number,
  easing?: (t: number) => number,
  css?: (t: number, u: number) => string,
  tick?: (t: number, u: number) => void
}
```

Animations can use custom functions that provide the `node`, an `animation` object and any `parameters` as arguments. The `animation` parameter is an object containing `from` and `to` properties each containing a DOMRect describing the geometry of the element in its `start` and `end` positions. The `from` property is the DOMRect of the element in its starting position, and the `to` property is the DOMRect of the element in its final position after the list has been reordered and the DOM updated.

If the returned object has a `css` method, Svelte will create a web animation that plays on the element.

The `t` argument passed to `css` is a value that goes from `0` and `1` after the `easing` function has been applied. The `u` argument is equal to `1 - t`.

The function is called repeatedly *before* the animation begins, with different `t` and `u` arguments.

App.svelte JS TS

```
<script lang="ts">
  import { cubicOut } from 'svelte/easing';

  function whizz(node: HTMLElement, { from, to }: { from: DOMRect; to: DOMRect }, params:
    const dx = from.left - to.left;
    const dy = from.top - to.top;

    const d = Math.sqrt(dx * dx + dy * dy);

    return {
      delay: 0,
      duration: Math.sqrt(d) * 120,
      easing: cubicOut,
      css: (t, u) => `transform: translate(${u * dx}px, ${u * dy}px) rotate(${t * 360}deg
```

```
{#each list as item, index (item)}
  <div animate:whizz>{item}</div>
{/each}
```

A custom animation function can also return a `tick` function, which is called *during* the animation with the same `t` and `u` arguments.

> If it's possible to use `css` instead of `tick`, do so — web animations can run off the main thread, preventing jank on slower devices.

App.svelte

JS TS

```
<script lang="ts">
  import { cubicOut } from 'svelte/easing';

  function whizz(node: HTMLElement, { from, to }: { from: DOMRect; to: DOMRect }, params:
    const dx = from.left - to.left;
    const dy = from.top - to.top;

    const d = Math.sqrt(dx * dx + dy * dy);

    return {
      delay: 0,
      duration: Math.sqrt(d) * 120,
      easing: cubicOut,
      tick: (t, u) => Object.assign(node.style, { color: t > 0.5 ? 'Pink' : 'Blue' })
    };
  }
</script>

{#each list as item, index (item)}
  <div animate:whizz>{item}</div>
{/each}
```

Edit this page on GitHub

PREVIOUS NEXT

Docs