

Like transitions and animations, an action can take an argument, which the action function will be called with alongside the element it belongs to.

Here, we're using a `longpress` action that fires an event with the same name whenever the user presses and holds the button for a given duration. Right now, if you switch over to the `longpress.js` file, you'll see it's hardcoded to 500ms.

We can change the action function to accept a `duration` as a second argument, and pass that `duration` to the `setTimeout` call:

```
export function longpress(node, duration) {
  // ...

  const handleMousedown = () => {
    timer = setTimeout(() => {
      node.dispatchEvent(
        new CustomEvent('longpress')
      );
    }, duration);
  };

  // ...
}
```

Back in `App.svelte`, we can pass the `duration` value to the action:

```
<button use:longpress={duration}>
```

This *almost* works — the event now only fires after 2 seconds. But if you slide the duration down, it will still take two seconds.

To change that, we can add an `update` method in `longpress.js`. This will be called whenever the argument changes:

```
return {
  update(newDuration) {
    duration = newDuration;
  },
  // ...
};
```

If you need to pass multiple arguments to an action, combine them into a single object, as in `use:longpress={{duration, spiciness}}`