Components can also dispatch events. To do so, they must create an event dispatcher. Update `Inner.svelte`:

```
<script>
    import { createEventDispatcher } from 'svelte';

    const dispatch = createEventDispatcher();

    function sayHello() {
        dispatch('message', {
            text: 'Hello!'
        });
    }
</script>
```

> *`createEventDispatcher` must be called when the component is first instantiated — you can't do it later inside e.g. a `setTimeout` callback. This links `dispatch` to the component instance.*

Notice that the `App` component is listening to the messages dispatched by `Inner` component thanks to the `on:message` directive. This directive is an attribute prefixed with `on:` followed by the event name that we are dispatching (in this case, `message`).

Without this attribute, messages would still be dispatched, but the App would not react to it. You can try removing the `on:message` attribute and pressing the button again.

> *You can also try changing the event name to something else. For instance, change `dispatch('message')` to `dispatch('myevent')` in `Inner.svelte` and change the attribute name from `on:message` to `on:myevent` in the `App.svelte` component.*