Keyed each blocks

By default, when you modify the value of an `each` block, it will add and remove items at the *end* of the block, and update any values that have changed. That might not be what you want.

It's easier to show why than to explain. Click the 'Remove first thing' button a few times, and notice what happens: It removes the first `<Thing>` component, but the *last* DOM node. Then it updates the `name` value in the remaining DOM nodes, but not the emoji.

Instead, we'd like to remove only the first `<Thing>` component and its DOM node, and leave the others unaffected.

To do that, we specify a unique identifier (or "key") for the `each` block:

```
{#each things as thing (thing.id)}
    <Thing name={thing.name}/>
{/each}
```

Here, `(thing.id)` is the *key*, which tells Svelte how to figure out which DOM node to change when the component updates.

> You can use any object as the key, as Svelte uses a `Map` internally — in other words you could do `(thing)` instead of `(thing.id)`. Using a string or number is generally safer, however, since it means identity persists without referential equality, for example when updating with fresh data from an API server.