



SVELTE • TEMPLATE SYNTAX

{#each ...}

ON THIS PAGE



```
{#each expression as name}...{/each}
```

```
{#each expression as name, index}...{/each}
```

Iterating over values can be done with an each block. The values in question can be arrays, array-like objects (i.e. anything with a `length` property), or iterables like `Map` and `Set` — in other words, anything that can be used with `Array.from`.

```
<h1>Shopping list</h1>
<ul>
  {#each items as item}
    <li>{item.name} x {item.qty}</li>
  {/each}
</ul>
```



You can use each blocks to iterate over any array or array-like value — that is, any object with a `length` property.

An each block can also specify an *index*, equivalent to the second argument in an `array.map(...)` callback:

```
{#each items as item, i}
  <li>{i + 1}: {item.name} x {item.qty}</li>
{/each}
```



```
{#each expression as name, index (key)}...{/each}
```

If a *key* expression is provided — which must uniquely identify each list item — Svelte will use it to diff the list when data changes, rather than adding or removing items at the end. The key can be any object, but strings and numbers are recommended since they allow identity to persist when the objects themselves change.

```
{#each items as item (item.id)}
  <li>{item.name} x {item.qty}</li>
{/each}

<!-- or with additional index value -->
{#each items as item, i (item.id)}
  <li>{i + 1}: {item.name} x {item.qty}</li>
{/each}
```

You can freely use destructuring and rest patterns in each blocks.

```
{#each items as { id, name, qty }, i (id)}
  <li>{i + 1}: {name} x {qty}</li>
{/each}

{#each objects as { id, ...rest }}
  <li><span>{id}</span><MyComponent {...rest} /></li>
{/each}

{#each items as [id, ...rest]}
  <li><span>{id}</span><MyComponent values={rest} /></li>
{/each}
```

Else blocks

```
{#each expression as name}...{:else}...{/each}
```

```
{#each todos as todo}
  <p>{todo.text}</p>
{:else}
  <p>No tasks today!</p>
{/each}
```



[✎ Edit this page on GitHub](#)

PREVIOUS

[{#if ...}](#)

NEXT

[{#key ...}](#)