

Every component has a *lifecycle* that starts when it is created, and ends when it is destroyed. There are a handful of functions that allow you to run code at key moments during that lifecycle.

The one you'll use most frequently is `onMount`, which runs after the component is first rendered to the DOM. We briefly encountered it [earlier](#) when we needed to interact with a `<canvas>` element after it had been rendered.

We'll add an `onMount` handler that loads some data over the network:

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

  let photos = [];

  onMount(async () => {
    const res = await fetch(`https://jsonplaceholder.typicode.com/photos?
_limit=20`);
    photos = await res.json();
  });
</script>
```

It's recommended to put the `fetch` in `onMount` rather than at the top level of the `<script>` because of server-side rendering (SSR). With the exception of `onDestroy`, lifecycle functions don't run during SSR, which means we can avoid fetching data that should be loaded lazily once the component has been mounted in the DOM.

Lifecycle functions must be called while the component is initialising so that the callback is bound to the component instance — not (say) in a `setTimeout`.

If the `onMount` callback returns a function, that function will be called when the component is destroyed.