Basics

Welcome to the Svelte tutorial. This will teach you everything you need to know to build fast, small web applications easily.

You can also consult the API docs and the examples, or — if you're impatient to start hacking on your machine locally — the 60-second quickstart.

What is Svelte?

Svelte is a tool for building fast web applications.

It is similar to JavaScript frameworks such as React and Vue, which share a goal of making it easy to build slick interactive user interfaces.

But there's a crucial difference: Svelte converts your app into ideal JavaScript at *build time*, rather than interpreting your application code at *run time*. This means you don't pay the performance cost of the framework's abstractions, and you don't incur a penalty when your app first loads.

You can build your entire app with Svelte, or you can add it incrementally to an existing codebase. You can also ship components as standalone packages that work anywhere, without the overhead of a dependency on a conventional framework.

How to use this tutorial

You'll need to have basic familiarity with HTML, CSS and JavaScript to understand Svelte.

As you progress through the tutorial, you'll be presented with mini exercises designed to illustrate new features. Later chapters build on the knowledge gained in earlier ones, so it's recommended that you go from start to finish. If necessary, you can navigate via the dropdown above (click 'Introduction / Basics').

Each tutorial chapter will have a 'Show me' button that you can click if you get stuck following the instructions. Try not to rely on it too much; you will learn faster by figuring out where to put each suggested code block and manually typing it in to the editor.

Understanding components

In Svelte, an application is composed from one or more *components*. A component is a reusable self-contained block of code that encapsulates HTML, CSS and JavaScript that belong together, written into a `.svelte` file. The ‘hello world’ example in the code editor is a simple component.