

Introduction to Svelte.js

- **What is Svelte.js?:** - A modern front-end framework for building user interfaces. - Created by Rich Harris in 2016. - Emphasizes compile-time optimization.
- **Unique Features and Advantages:** - No virtual DOM. - Small bundle size. - Reactive declarations. - Better performance and simplicity.

Core Concepts of Svelte.js

- **Reactive Programming in Svelte:** - Automatically updates the DOM when state changes. - Uses \$: reactive assignments for declarative updates.
- **Svelte's Compilation Process:** - Converts components into highly optimized imperative code. - No runtime overhead.
- **Component Structure:** - Simplistic approach to components with HTML, CSS, and JavaScript all in one file. - Clear and concise syntax.

Svelte.js vs. Other Frameworks

- **Comparison with React, Vue, and Angular:** - Svelte is a compiler, not a runtime framework. - No virtual DOM, unlike React. - Simplifies learning and development with a minimal API.
- **Performance and Bundle Size Benefits:** - Smaller bundle sizes lead to faster loading times. - Pre-compiled code offers better runtime performance.

Building a Svelte.js Application

- **Setting up a Svelte Project:** - Use Svelte's CLI to create a new project: ``npx degit sveltejs/template my-svelte-project``. - Understand the basic project structure and setup.
- **Basic Example: Creating a Component:** - Write a simple component with HTML, CSS, and JavaScript in one file. - Display dynamic data using Svelte's reactive assignments.
- **Handling State and Props:** - Manage component state using reactive declarations. - Pass data between components using props.

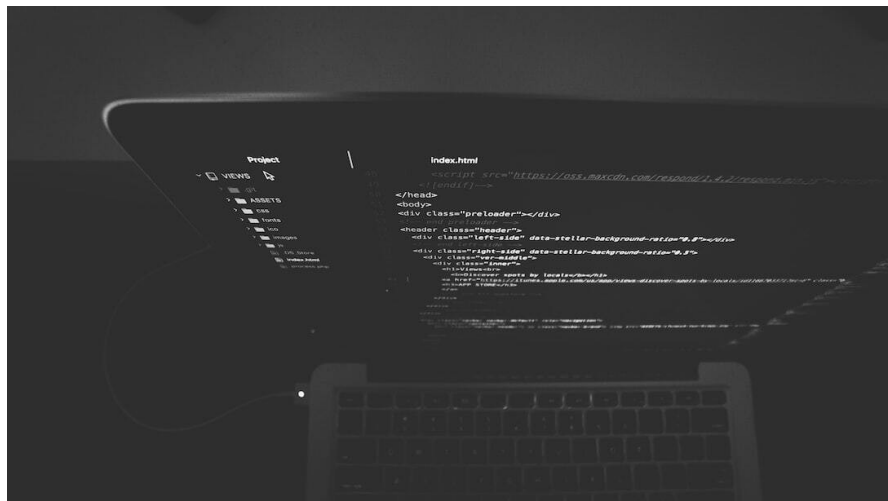


Photo by Nate Grant on Unsplash

Advanced Features of Svelte.js

- **Svelte Stores:** - Centralized state management with writable and readable stores. - Simplifies data sharing and state management across components.
- **Transitions and Animations:** - Built-in support for smooth transitions and keyframe animations. - Create custom animations using `@keyframes` directive.
- **Custom Directives:** - Extend Svelte functionality with custom directives. - Use custom directives for advanced DOM manipulations.

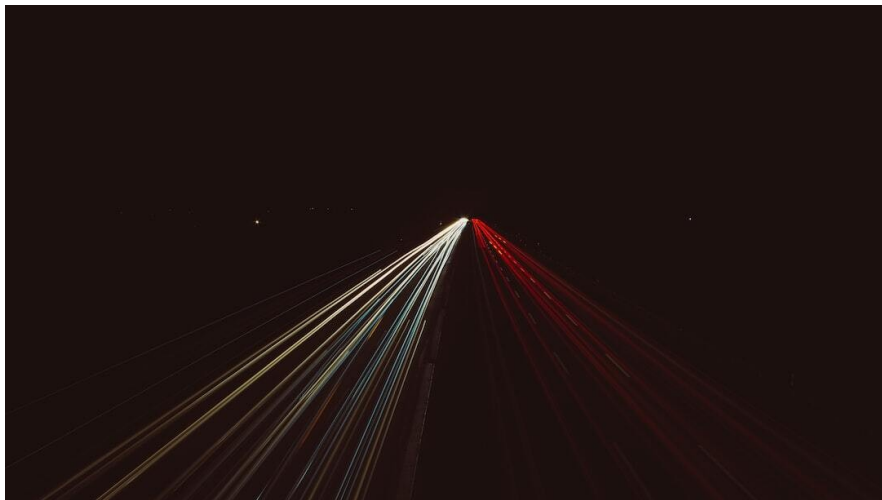


Photo by Markus Spiske on Unsplash

SvelteKit Overview

- **Introduction to SvelteKit:** - A powerful framework for building modern web applications with Svelte. - Provides features like routing, SSR (Server-Side Rendering), and static site generation.
- **Benefits of Using SvelteKit:** - Simplifies the process of building and deploying Svelte applications. - Supports both client-side and server-side rendering out of the box. - Flexible and powerful for developing complex web applications.

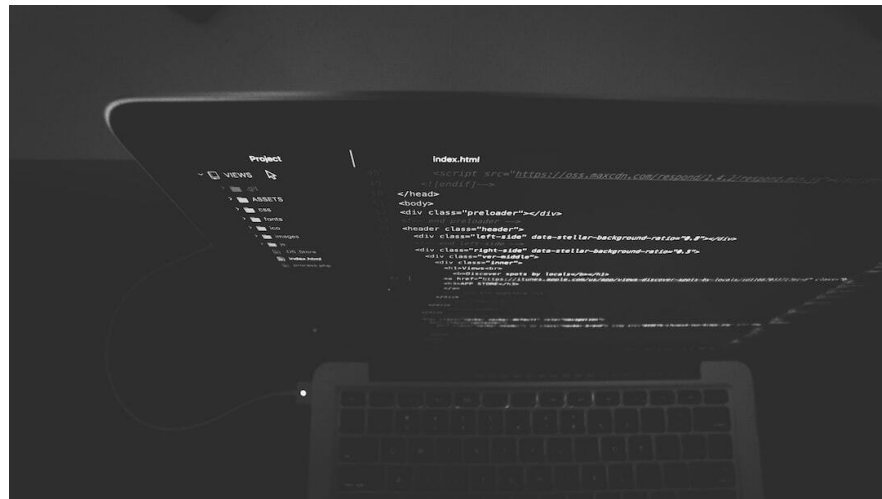


Photo by Nate Grant on Unsplash

Ecosystem and Community

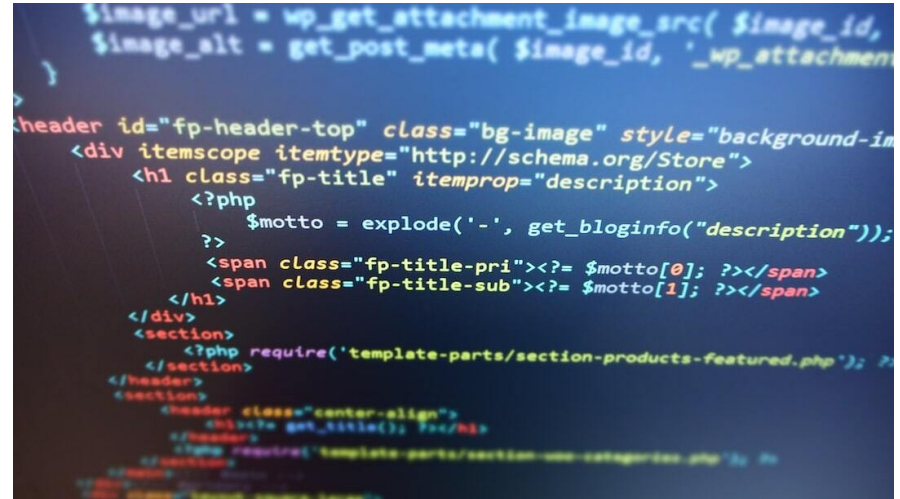
- **Popular Svelte.js Tools and Libraries:** - Svelte Material UI, Svelte Headless UI for building UI components. - Sapper, the predecessor to SvelteKit, for full-stack development. - Svelte DevTools for debugging Svelte applications.
- **Community Resources and Support:** - Official documentation and extensive tutorials on svelte.dev. - Active community on Discord, Reddit, and Stack Overflow. - Numerous blogs, video tutorials, and online courses.



Photo by Nandha Kumar on Unsplash

Conclusion

- **Summary of Key Points:** - Svelte.js provides a unique compile-time approach with no virtual DOM. - It offers smaller bundle sizes and reactive programming for efficient UI updates. - SvelteKit enhances Svelte for building full-stack applications.
- **Future of Svelte.js:** - Growing adoption and community support in web development. - Continuous innovation and expansion of the ecosystem. - Potential to influence future web development practices significantly.



```
$image_url = wp_get_attachment_image_src( $image_id,
$image_alt = get_post_meta( $image_id, '_wp_attachment
)
}
<header id="fp-header-top" class="bg-image" style="background-in
<div itemscope itemtype="http://schema.org/Store">
  <h1 class="fp-title" itemprop="description">
    <?php
      $motto = explode('-', get_bloginfo("description"));
    ?>
    <span class="fp-title-pri"><?= $motto[0]; ?></span>
    <span class="fp-title-sub"><?= $motto[1]; ?></span>
  </h1>
</div>
<section>
  <?php require('template-parts/section-products-featured.php'); ?>
</section>
</header>
<section>
  <header class="center-align">
    <?php get_title(); ?></header>
    <?php require('template-parts/section-see-categories.php'); ?>
  </section>
</section>
</div>
```

Photo by Datahjelpen AS on Unsplash

Performance Metrics

- **Initial Load Time:** - Svelte: 0.9 seconds - React: 1.1 seconds - Vue: 1.0 seconds - Angular: 1.2 seconds
- **Runtime Performance:** - Svelte: Minimal runtime overhead due to pre-compiled code. - Efficient DOM updates leading to faster rendering times.
- **Memory Usage:** - Svelte: Uses less memory during execution. - Optimized rendering process reduces memory footprint compared to React, Vue, and Angular.



Photo by Agence Olloweb on Unsplash

Bundle Size Comparison

- **Svelte:** - Typical bundle size: ~3.6 KB (minified + gzipped)
- **React:** - Typical bundle size: ~43 KB (minified + gzipped)
- **Vue:** - Typical bundle size: ~23 KB (minified + gzipped)
- **Angular:** - Typical bundle size: ~68 KB (minified + gzipped)
- **Implications:** - Smaller bundle sizes lead to faster load times and improved user experience.
 - Svelte's pre-compilation reduces the need for a large runtime library.

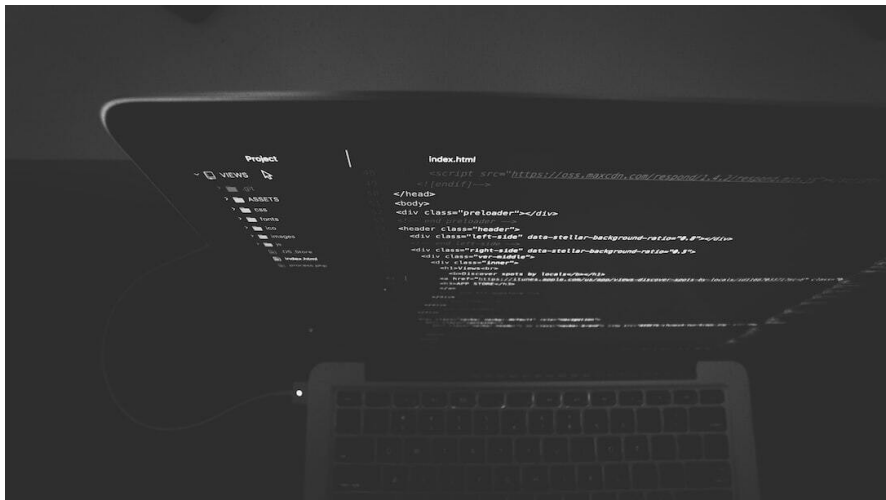


Photo by Nate Grant on Unsplash

Code Snippet Comparison

- **React Code Snippet:**

```
```\nimport React, { useState } from 'react';\nfunction Counter() {\n  const [count, setCount] = useState(0);\n  return (\n    <div>\n      <span>{count}</span>\n    </div>\n  );\n}
```
- **Svelte Code Snippet:**

```
```\n<div>\n  {count}\n</div>
```
- **Key Differences:** - Svelte uses a more declarative approach with reactive assignments. - All logic, styling, and markup are in a single file in Svelte. - React relies on hooks and JSX for state management and UI updates.



Photo by Arnold Francisca on Unsplash