Welcome to the first edition of our "What's new in Svelte" series! We'll try to make this a monthly blog post in which you'll find out about new features, bug fixes, and a showcase of Svelte projects from around the community.

New features

1. `use:obj.method` allows functions defined within objects to be used within actions (Example, **3.26.0**, warning removed in **3.27.0**)
2. `_` is now supported as a "numerical separator", similar to a `.` or `,` (Example, **3.26.0**)
3. `import.meta` now works in template expressions (Example, **3.26.0**)
4. CSS Selectors with `~` and `+` combinators are now supported (Example, **3.27.0**, with a compiler fix in **3.29.0**)
5. The `{#key}` block is now available to key arbitrary content on an expression. Whever the expression changes, the contents inside the `{#key}` block will be destroyed and recreated. For an in-depth explanation and to find out how it's implemented, check out the new blog post of Svelte Team member Tan Li Hau. (More info, **3.29.0**)
6. Slots can now be forwarded through child components! This used to only be possible by adding extra wrapper `<div>` s (More info, **3.29.0**)
7. When using TypeScript, you can now type the `createEventDispatcher` method:

```
<script lang="ts">
    import { createEventDispatcher } from 'svelte';

    const dispatch = createEventDispatcher<{
        /**
         * you can also add docs
         */
        checked: boolean; // Will translate to `CustomEvent<boolean>`
        hello: string;
    }>();

    // ...
</script>
```

This will make sure that you can invoke dispatch only with the specified event names and its types. The Svelte for VS Code extension was also updated to deal with this new feature. It will provide strong typings for these events as well as autocompletion and hover information.

New from Sapper! Sapper 0.28.9 just came out. The highlights from it include much better support for CSP nonces, asset preload support for exported pages, and error details are now available in the `$page` store on error pages.

In addition, Sapper's CSS handling has been rewritten over the course of recent releases in order to fix existing CSS handling bugs, refactor the CSS handling to occur entirely within a Rollup plugin, and remove the need internally to register CSS in the routing system. Congrats and thank you to the folks working on Sapper for all their solid work!

Impactful bug fixes

- CSS compilation will no longer remove rules for the `open` attribute on `<details>` elements (Example, **3.26.0**)
- `prettier-plugin-svelte` will do a better job now at dealing with whitespaces, especially around inline elements. It will also preserve formatting inside `<pre>` tags and will no longer format languages

which are not supported by Prettier, like SASS, Pug or Stylus.

Coming up

- Svelte Summit, Svelte's second global online conference, is taking place on October 18! Sign up for free to get reminders and talk updates!

For all the features and bugfixes see the CHANGELOG for Svelte and Sapper.

Svelte Showcase

- This CustomMenu example demos how to replace the OS right-click menu
- Github Tetris lets you play a Tetris-like game in a git commit history
- Who are my representatives? is a website built with Svelte to help US residents get more info on their congressional representatives
- Pick Palette is a color palette manager made with Svelte!

In-depth learning:

- Svelte 3 Up and Running is a new book about building production-ready static web apps with Svelte 3
- Sapper Tutorial (Crash Course) walks through the ins-and-outs of Sapper, the Svelte-powered application framework
- Svelte Society Day France happened September 27th featuring a wide variety of topics all in French! You can find the full recording here.

Plug-and-play components:

- svelte-zoom brings "nearly native" pan-and-zoom to images on desktop and mobile
- svelte-materialify is a Material component library for Svelte with over 50 components
- svelte-undoable makes it easy to introduce undo and redo functionality using `bind:`
- This Tilt component implements a common UX pattern where the hovered element tilts to follow the mouse

Lots of examples of how use JS tech came out this month:

- Sapper with PostCSS and Tailwind
- PrismJS (Code block syntax highlighting)
- Filepond (Drag-and-drop file upload)
- Ionic (UI Components)
- Pell (WYSIWYG Editor)
- Leaflet (Mapping)

Reminder: There's a Svelte integrations repo that demonstrates ways to incorporate Svelte into your stack (and vice versa). If you've got questions on how to use a particular piece of tech with Svelte, you may find your answer there... and if you've gotten something to work with Svelte, consider contributing!

For more amazing Svelte projects, check out the Svelte Society, Reddit and Discord... and be sure to post your own!

See you next month!

By the way, Svelte now has an OpenCollective! All contributions and all expenses are published in our transparent public ledger. Learn who is donating, how much, where that money is going, submit expenses, get reimbursed and more!