

Svelte changelog

Unreleased

- Add `Symbol` as a known global (#7419)

3.46.6

- Actually include action TypeScript interface in published package (#7407)

3.46.5

- Add TypeScript interfaces for typing actions (#6538)
- Do not generate `unused-export-let` warning inside `<script context="module">` blocks (#7055)
- Do not collapse whitespace-only CSS vars (#7152)
- Add `aria-description` to the list of allowed ARIA attributes (#7301)
- Fix attribute escaping during SSR (#7327)
- Prevent `.innerHTML` optimization from being used when `style:` directive is present (#7386)

3.46.4

- Avoid `maximum call stack size exceeded` errors on large components (#4694)
- Preserve leading space with `preserveWhitespace: true` (#4731)
- Preserve leading space in `<pre>` tags (#6437)
- Improve error message when trying to use `style:` directives on inline components (#7177)
- Add `FormData` as a known global (#7199)
- Mark `css/instance/module` AST properties as optional in types (#7204)

3.46.3

- Ignore whitespace in `{#each}` blocks when containing elements with `animate:` (#5477)
- Throw compiler error when variable in `context="instance"` collides with import in `context="module"` (#7090)
- Fix compiler crash when `{@const}` contains arrow functions (#7134)

3.46.2

- Export `FlipParams` interface from `svelte/animate` (#7103)
- Fix `style:` directive reactivity inside `{#each}` block (#7136)

3.46.1

- Handle `style:kebab-case` directives (#7122)
- Improve AST produced for `style:` directives (#7127)

3.46.0

- Implement `{@const}` tag (RFC #33, #6413)
- Implement `style:` directive (RFC #42, #5923)
- Fix style manager conflicts when using multiple Svelte instances (#7026)
- Fix hydration when using `{@html}` (#7115)

3.45.0

- Fix non-boolean attribute rendering in SSR to render truthy values as-is (#6121)
- Fix binding to a member expression also invalidating the member property (#6921)
- Fix default values in `{#each}`/etc. destructurings not being considered references for the purposes of compiler warnings (#6964)
- Fix `{:else if}` value incorrectly being cached (#7043)
- Add `all-no-redundant-roles` warning (#7067)
- Fix code generation error with arrow functions whose bodies are object destructuring assignments (#7087)

3.44.3

- Fix `bind:this` binding inside `onMount` for manually instantiated component (#6760)
- Prevent cursor jumps with one-way binding for other `type="text"`-like `<input>`s (#6941)
- Exclude `async` loops from `loopGuardTimeout` (#6945)

3.44.2

- Fix overly restrictive preprocessor types (#6904)
- More specific typing for `crossfade` function - returns a tuple, not an array (#6926)
- Add `URLSearchParams` as a known global (#6938)
- Add `types` field to `exports` map (#6939)

3.44.1

- Fix code generation when a multi-line `return` statement contains comments (code-red#36)
- Fix code generation when `for/if/while` statements have empty bodies (#6884)

3.44.0

- Add `enableSourcemap` compiler option (#6835)

3.43.2

- Fix regression where user-specified `imports` were not rewritten according to the `sveltePath` option (#6834)

3.43.1

- Prevent a rejecting promise used in `{#await}` during SSR from appearing as an unhandled rejection (#6789)

3.43.0

- Use export map to expose no-op versions of lifecycle functions for SSR (#6743)
- Prefer `context` passed to component constructor, even when running synchronously in another component (#6753)
- Handle preprocessors that return empty sourcemaps (#6757)

3.42.6

- Hide private preprocess typings (#6622)
- Fix reactive function in `{:else if}` expression not being properly re-run (#6727)

3.42.5

- In `draw` transition, account for `stroke-linecap` in determining length (#4540)
- Fix regression with destructuring assignments with default values (#6699)

3.42.4

- Only apply optimized `src` attribute handling when in an `html` namespace (#6575)
- Fix styles for transitions and animations being attached to the wrong document in `<iframe>`s (#6637)
- Fix `<select>` with a `{...spread}` attribute that didn't provide a value key getting its value improperly unset (#6675)

3.42.3

- Add `BigInt` as a known global (#6671)

- Fix regression where `onDestroy` in `svelte/ssr` was improperly a no-op (#6676)

3.42.2

- Collapse whitespace in `class` and `style` attributes (#6004)
- Deselect all `<option>`s in a `<select>` where the bound `value` doesn't match any of them (#6126)
- In hydrated components, only rely on helpers for creating the types of elements present in the component (#6555)
- Add `HTMLElement` and `SVGElement` as known globals (#6643)
- Account for scaling in `flip` animations (#6657)

3.42.1

- Fix regression with reordering keyed `{#each}` blocks when compiling with hydration enabled (#6561)

3.42.0

- Allow `use:actions` to be used on `<svelte:body>` (#3163)
- Improve parser errors for certain invalid components (#6259, #6288)
- Fix paths in generator JS sourcemaps to be relative (#6598)
- Fix overzealous warnings about `context="module"` variables not being reactive (#6606)

3.41.0

- Support `export { ... } from` syntax in components (#2214)
- Support `export let { ... } =` syntax in components (#5612)
- Support `{#await ... then/catch}` without a variable for the resolved/rejected value (#6270)

3.40.3

- Fix `<slot>` data when a transition is cancelled before completing (#5394)
- Fix destructuring into variables beginning with `$` so that they result in store updates (#5653)
- Fix `in:` transition configuration not properly updating when it's changed after its initial creation (#6505)
- Fix applying `:global()` for `>` selector combinator (#6550)
- Fix mounting component at detached DOM node (#6567)

3.40.2

- Fix dynamic `autofocus={...}` attribute handling (#4995)

- Add filename to combined source map if needed (#6089)
- In AST, parse empty attribute values as an empty string (#6286)
- Fix tracking whether transition has started (#6399)
- Fix incorrect scoping of `:global()` selectors (#6550)

3.40.1

- Fix store reactivity regression when using reactive statements (#6557)

3.40.0

- Support rendering a component in a shadow DOM (#5869)
- Fix `:root` selector being erroneously scoped to component (#4767)
- Fix `.end` in AST for expressions inside attributes (#6258)
- Fix one-way `<select>` binding when it has a spread attribute (#6433)
- Various hydration improvements and fixes (#6449)
- Use smaller versions of internal helpers when compiling without hydration support (#6462)
- Fix two-way binding of values when updating through synchronous component accessors (#6502)

3.39.0

- Support `bind:group` in SSR (#4621)
- Add ally warning `ally-mouse-events-have-key-events` which checks that `mouseover/mouseout` are accompanied by `focus/blur` event handlers (#5938)
- Make it possible to silence more warnings (#5954)
- Add `!trusted` event modifier (#6137)
- Add `varsReport` compiler option to include all variables reference in the component in the `variables` report (#6192)
- Add `errorMode` compiler option to try to continue compiling when an error is detected (#6194)
- Expose `svelte/ssr` which exports lifecycle methods as no-ops (#6416)
- Add `getAllContexts` (#6447)
- Throw proper error for `export default function() {}` and `export default class {}` rather than crashing the compiler (#3275)
- Fix SSR rendering of falsy `input` values (#4551)
- Fix `preserveComments` in SSR mode (#4730)
- Do not warn if `context="module"` variables are not the only dependencies in reactive statements (#5954)
- Stop checking `ally-media-has-caption` ally warning on `<audio>` elements (#6054)
- Fix erroneous “unknown prop” warning when using slot on a component (#6065)
- Add sourcemaps to all HTML elements (#6092)

- Relax **derived** function signature (#6178)
- Throw compiler error when passing empty directive names (#6299)
- Fix compiler error when using **:where()** inside **:global()** (#6434)
- Fix ordering of elements in keyed **{#each}** (#6444)
- Remove deprecated ally warning **ally-no-onchange warning** (#6457)
- Fix **:global()** with pseudo element not being seen as global (#6468)
- Allow **:global()** to contain multiple selectors when it is not part of a larger selector (#6477)
- Make **<script>** and **<style>** end tag parsing more robust (#6511)

3.38.3

- Speed up hydration by reducing amount of element reorderings (#4308)
- Fix escaping attribute values when using a spread in SSR (#5756)
- Throw compiler error when **:global()** contains multiple selectors (#5907)
- Give explicit error rather than crashing when an attribute shorthand **{}** is empty (#6086)
- Make **<textarea>** end tag parsing more robust (#6276)
- Fix **:global(...):some-pseudoclass** selectors not being seen as global (#6306)
- Fix type signatures of **writable** and **readable** so it's possible to call them without arguments (#6291, #6345)
- Preserve **this** in bubbled events (#6310)
- Fix slot props not updating when transition is aborted (#6386)
- Fix generic props relationship in **SvelteComponentTyped** (#6400)

3.38.2

- Revert hydration optimisation for the time being (#6279)

3.38.1

- Fix hydration regression (#6274)

3.38.0

- Avoid recreating DOM elements during hydration (#1067)
- Support passing CSS custom properties to component (#5628)
- Support **:global()** as part of compound CSS selectors (#6222)
- Fix updating **<slot>** contents when there's an aborted transition (#3542)
- Fix setting boolean attributes on custom elements (#5951)
- Add missing function overload for **derived** to allow explicitly setting an initial value for non-async derived stores (#6172)
- Fix dynamic **href** values erroneously triggering ally warnings (#5990)
- Fix scope leak when updating an **{#await}** block (#6173)
- Pass full markup source to **script/style** preprocessors (#6169)

- Fix `crossfade` types to mark `fallback` as optional (#6201)
- Add missing “context” typing to `SvelteComponent` constructor options (#6236)
- Don’t automatically switch to `svg` namespace when in `foreign` namespace (#6257)

3.37.0

- Allow root-level context to be passed to the component constructor (#6032)

3.36.0

- Add `this: void` typing to store functions (#6094)
- Export `Spring`, `Tweened` and `EasingFunction` interfaces (#6070, #6056)
- Export interfaces for transition parameters (#5207)
- Export store’s useful TypeScript definitions (#5864)
- Fix previous breaking change to `svelte/preprocess` types location (#6100)
- Fix missing slotted elements in AST (#6066)

3.35.0

- Implement slotted components and `<svelte:fragment slot=“...”>` (#1037, #2079)
- Fix reactivity bug where `slot=“...”` is specified after attributes that should be reactive (#5626)

3.34.0

- Add a `cssHash` option for controlling the classname used for CSS scoping (#570)

3.33.0

- In custom elements, call `onMount` functions when connecting and clean up when disconnecting (#1152, #2227, #4522)
- Allow destructured defaults to refer to other variables (#5066)
- Do not emit `contextual-store` warnings for function parameters or declared variables (#6008)

3.32.3

- Fix removal of lone `:host` selectors (#5982)

3.32.2

- Fix unnecessary additional invalidation with `<Component bind:prop={obj.foo}/>` (#3075, #4447, #5555)
- Fix scoping of selectors with `:global()` and `~` sibling combinators (#5499)
- Fix removal of `:host` selectors as unused when compiling to a custom element (#5946)

3.32.1

- Warn when using `module` variables reactively, and close weird reactivity loophole (#5847)
- Throw a parser error for `class:` directives with an empty class name (#5858)
- Fix extraneous store subscription in SSR mode (#5883)
- Don't emit update code for `class:` directives whose expression is not dynamic (#5919)
- Fix type inference for derived stores (#5935)
- Make parameters of built-in animations and transitions optional (#5936)
- Make `SvelteComponentDev` typings more forgiving (#5937)
- Fix `foreign` elements incorrectly disallowing `bind:this` (#5942)

3.32.0

- Allow multiple instances of the same action on an element (#5516)
- Support `foreign` namespace, which disables certain HTML5-specific behaviour and checks (#5652)
- Support inline comment sourcemaps in code from preprocessors (#5854)

3.31.2

- Rework SSR store handling to subscribe and unsubscribe as in DOM mode (#3375, #3582, #3636)
- Fix error when removing elements that are already transitioning out (#5789, #5808)
- Fix duplicate content race condition with `{#await}` blocks and out transitions (#5815)
- Deconflict variable names used for contextual actions (#5834)

3.31.1

- Fix scrolling of element with resize listener by making the `<iframe>` have `z-index: -1` (#5448)
- Fix location of automatically declared reactive variables (#5749)
- Warn when using `className` or `htmlFor` attributes (#5777)
- Fix checkbox `bind:group` in keyed `{#each}` where the array can be re-ordered (#5779)

- Fix checkbox `bind:group` in nested `{#each}` contexts (#5811)
- Add graphics roles as known ARIA roles (#5822)
- Fix local transitions if a parent has a cancelled outro transition (#5829)
- Support `use:obj.some.deep.function` as actions (#5844)

3.31.0

- Use a separate `SvelteComponentTyped` interface for typed components (#5738)

3.30.1

- Support consuming decoded sourcemaps as created by the `source-map` library's `SourceMapGenerator` (#5722)
- Actually export `hasContext` (#5726)

3.30.0

- Add a typed `SvelteComponent` interface (#5431)
- Support spread into `<slot>` props (#5456)
- Fix setting reactive dependencies which don't appear in the template to `undefined` (#5538)
- Support preprocessor sourcemaps during compilation (#5584)
- Fix ordering of elements when using `{#if}` inside `{#key}` (#5680)
- Add `hasContext` lifecycle function (#5690)
- Fix missing walk types in `svelte/compiler` (#5696)

3.29.7

- Include `./register` in exports map (#5670)

3.29.6

- Include `./package.json` in export map (#5659)

3.29.5

- Fix `$$props` and `$$restProps` when compiling to a custom element (#5482)
- Include an export map in `package.json` (#5556)
- Fix function calls in `<slot>` props that use contextual values (#5565)
- Fix handling aborted transitions in `{:else}` blocks (#5573)
- Add `Element` and `Node` to known globals (#5586)
- Fix `$$slots` when compiling to custom elements (#5594)
- Fix internal `imports` so that we're exposing a valid ES module (#5617)

3.29.4

- Fix code generation error with `??` alongside logical operators (#5558)

3.29.3

- Hopefully actually republish with proper UMD build for use in the REPL

3.29.2

- Republish with proper UMD build for use in the REPL

3.29.1

- Fix compiler hanging on `<slot slot="...">` (#5475)
- Fix types on `get` function in `svelte/store` (#5483)
- Add missing `end` field on ASTs for non-top-level `<style>` elements (#5487)
- Fix `{#if}` inside `{#await}` with destructuring (#5508)
- Fix types on lifecycle hooks (#5529)

3.29.0

- Support `<slot slot="...">` (#2079)
- Fix unmounting components with a bidirectional transition with a delay (#4954)
- Add types to `get` function in `svelte/store` (#5269)
- Add a warning when a component looks like it's trying to use another component without beginning with a capital letter (#5302)
- Add `EventSource` to known globals (#5463)
- Fix compiler exception with `~/+` combinators and `{...spread}` attributes (#5465)

3.28.0

- Add `{#key}` block for keying arbitrary content on an expression (#1469)

3.27.0

- Add `|nonpassive` event modifier, explicitly passing `passive: false` (#2068)
- Scope CSS selectors with `~` and `+` combinators (#3104)
- Fix keyed `{#each}` not reacting to key changing (#5444)
- Fix destructuring into store values (#5449)
- Fix erroneous `missing-declaration` warning with `use:obj.method` (#5451)

3.26.0

- Support `use:obj.method` as actions (#3935)
- Support `_` as numeric separator (#5407)
- Fix assignments to properties on store values (#5412)
- Add special style scoping handling of `[open]` selectors on `<details>` elements (#5421)
- Support `import.meta` in template expressions (#5422)

3.25.1

- Fix specificity of certain styles involving a child selector (#4795)
- Fix transitions that are parameterised with stores (#5244)
- Fix scoping of styles involving child selector and `*` (#5370)
- Fix destructuring which reassigns stores (#5388)
- Fix `{#await}`s with no `{:catch}` getting stuck unresolved if the promise rejects (#5401)

3.25.0

- Use `null` rather than `undefined` for coerced bound value of `<input type="number">` (#1701)
- Expose object of which slots have received content in `$$slots` (#2106)
- Correctly disallow using lifecycle hooks after synchronous component initialisation (#4259, #4899)
- Re-throw an unhandled rejection when an `{#await}` block with no `{:catch}` gets a rejection (#5129)
- Add types to `createEventDispatcher` (#5211)
- In SSR mode, do not automatically declare variables for reactive assignments to member expressions (#5247)
- Include selector in message of `unused-css-selector` warning (#5252)
- Fix using `<Namespaced.Component/>`s in child `{#await}/{#each}` contexts (#5255)
- Fix using `<svelte:component>` in `{:catch}` (#5259)
- Fix setting one-way bound `<input>` value to `undefined` when it has spread attributes (#5270)
- Fix deep two-way bindings inside an `{#each}` involving a store (#5286)
- Use valid XHTML for elements that are optimised and inserted with `.innerHTML` (#5315)
- Fix reactivity of `$$props` in slot fallback content (#5367)

3.24.1

- Prevent duplicate invalidation with certain two-way component bindings (#3180, #5117, #5144)
- Fix reactivity when passing `$$props` to a `<slot>` (#3364)

- Fix transitions on `{#each} { :else }` (#4970)
- Fix unneeded invalidation of `$$props` and `$$restProps` (#4993, #5118)
- Provide better compiler error message when mismatched tags are due to autoclosing of tags (#5049)
- Add `a11y-label-has-associated-control` warning (#5074)
- Add `a11y-media-has-caption` warning (#5075)
- Fix `bind:group` when using contextual reference (#5174)

3.24.0

- Support nullish coalescing (`??`) and optional chaining (`?.`) operators (#1972)
- Support `import.meta` (#4379)
- Fix only setting `<input>` values when they're changed when there are spread attributes (#4418)
- Fix placement of `{@html}` when used at the root of a slot, at the root of a component, or in `<svelte:head>` (#5012, #5071)
- Fix certain handling of two-way bound `contenteditable` elements (#5018)
- Fix handling of `imported` value that is used as a store and is also mutated (#5019)
- Do not display `a11y-missing-content` warning on elements with `contenteditable` bindings (#5020)
- Fix handling of `this` in inline function expressions in the template (#5033)
- Fix collapsing HTML with static content (#5040)
- Prevent use of `$store` at compile time when top-level `store` has been shadowed (#5048)
- Update `<select>` with one-way `value` binding when the available `<option>`s change (#5051)
- Fix published `tweened` types so the `.set()` and `.update()` options are optional (#5062)
- Fix contextual `bind:this` inside `{#each}` block (#5067)
- Preprocess self-closing `<script>` and `<style>` tags (#5080)
- Fix types for animation- and transition-related param objects so each param is optional (#5083)

3.23.2

- Fix `bind:group` inside `{#each}` (#3243)
- Don't crash when using an arrow function as a statement (#4617)
- Deconflict `bind:this` variable (#4636)

3.23.1

- Fix checkbox `bind:group` when multiple options have the same value (#4397)
- Fix `bind:this` to the value of an `{#each}` block (#4517)

- Fix reactivity when assigning to contextual `{#each}` variable (#4574, #4744)
- Fix binding to contextual `{#each}` values that shadow outer names (#4757)
- Work around EdgeHTML DOM issue when removing attributes during hydration (#4911)
- Throw CSS parser error when `:global()` does not contain a selector (#4930)

3.23.0

- Update `<select>` with `bind:value` when the available `<option>`s change (#1764)
- Add `muted` binding for media elements (#2998)
- Fix inconsistencies when setting a two-way bound `<input>` to `undefined` (#3569)
- Fix setting `<select multiple>` when there are spread attributes (#4392)
- Fix let-less `<slot>` with context overflow (#4624)
- Fix resize listening on certain older browsers (#4752)
- Add `a11y-no-onchange` warning (#4788)
- Fix `use:` actions being recreated when a keyed `{#each}` is reordered (#4693)
- Fix `{@html}` when using tags that can only appear inside certain tags (#4852)
- Fix reactivity when binding directly to `{#each}` context (#4879)

3.22.3

- Support default values and trailing commas in destructuring `{#await}` (#4560, #4810)
- Fix handling of `tweened` store when set using `duration: 0` (#4799, #4846)
- Fix setting `value` attribute with `bind:group` and attribute spread (#4803)
- Fix issue with compound `{#if}` block involving static condition, dynamic condition, and inline component (#4840)
- Update a11y warnings per ARIA 1.2 working draft (#4844)

3.22.2

- Fix compiler exception with `a11y-img-redundant-alt` and value-less `alt` attribute (#4777)

3.22.1

- Fix compiler exception with `a11y-img-redundant-alt` and dynamic `alt` attribute (#4770)

3.22.0

- Fix misaligned line numbers in source maps (#3906)
- Make setting a `tweened` store using `duration: 0` instantly update the value (#4399)
- Fix reactivity with imported values that are then mutated (#4555)
- Fix contextual dynamic `bind:this` inside `{#each}` block (#4686)
- Do not display `ally` warning about missing `href` for `<a>` with `name` or `id` (#4697)
- Disable infinite loop guard inside generators (#4698)
- Display `ally-invalid-attribute` warning for `href="javascript:..."` (#4733)
- Implement `ally-img-redundant-alt` warning (#4750)
- Fix variable name conflict with component called `<Anchor>` (#4768)

3.21.0

- Support dimension bindings in cross-origin environments (#2147)
- Fix several related outro bugs (#3202, #3410, #3685, #4620, #4630)
- Try using `globalThis` rather than `globals` for the benefit of non-Node servers and web workers (#3561, #4545)
- Support `{#await ... catch ...}` syntax shorthand (#3623)
- Fix attaching of JS debugging comments to HTML comments (#4565)
- Fix `<svelte:component/>` within `<slot/>` (#4597)
- Fix bug with updating simple `{#if}` blocks (#4629)
- Fix issues with `<input type="number">` updates (#4631, #4687)
- Prevent illegal attribute names (#4648)
- Fix `{#if}` block directly within `<slot/>` (#4703)

3.20.1

- Fix compiler regression with slots (#4562)

3.20.0

- Allow destructuring in `{#await}` blocks (#1851)
- Allow `<svelte:self>` to be used in a slot (#2798)
- Expose object of unknown props in `$$restProps` (#2930)
- Prevent passing named slots other than from the top level within a component (#3385)
- Allow transitions and animations to work within iframes (#3624)
- Fix initialising slot fallbacks when unnecessary (#3763)
- Disallow binding directly to `const` variables (#4479)
- Fix re-attaching event handlers on keyed `{#each}` blocks (#4491)
- Fix updating keyed `{#each}` blocks with `{:else}` (#4536, #4549)
- Fix hydration of top-level content (#4542)

3.19.2

- In **dev** mode, display a runtime warning when a component is passed an unexpected slot (#1020, #1447)
- In **vars** array, correctly indicate whether **module** variables are **mutated** or **reassigned** (#3215)
- Fix spread props not updating in certain situations (#3521, #4480)
- Use the fallback content for slots if they are passed only whitespace (#4092)
- Fix bitmask overflow for **{#if}** blocks (#4263)
- In **dev** mode, check for unknown props even if the component has no writable props (#4323)
- Exclude global variables from **\$capture_state** (#4463)
- Fix bitmask overflow for slots (#4481)

3.19.1

- Do not treat modifications to **\$\$props** as updates to a store called **\$props** (#4368)
- Deconflict **value** parameter name used in contextual bindings (#4445)
- Fix dev mode validation of **{#each}** blocks using strings (#4450)

3.19.0

- Fix indirect bindings involving elements with spreads (#3680)
- **\$capture_state/\$inject_state** now act on the component's entire state, rather than its props (#3822)
- Warn when using **<Foo/>** and **Foo** is dynamic (#4331)
- Display compilation warnings in **svelte/register** in dev mode (#4364)
- Fix unneeded updating of keyed each blocks (#4373)
- Throw runtime error in dev mode for non-array-like values in **{#each}** blocks (#4408)

3.18.2

- Fix binding to module-level variables (#4086)
- Improve parsing error messages when there is a pending unclosed tag (#4131)
- Disallow attribute/prop names from matching two-way-bound names or **{shorthand}** attribute/prop names (#4325)
- Improve performance of **flush()** by not using **.shift()** (#4356)
- Permit reserved keywords as destructuring keys in **{#each}** (#4372)
- Disallow reserved keywords in **{expressions}** (#4372)
- Fix code generation error with precedence of arrow functions (#4384)
- Fix event handlers that are dynamic via reactive declarations or stores (#4388)
- Fix invalidation in expressions like **++foo.bar** (#4393)

3.18.1

- Fix code generation error with adjacent inline and block comments (#4312)
- Fix detection of unused CSS selectors that begin with a `:global()` but contain a scoped portion (#4314)

3.18.0

- Fix infinite loop when instantiating another component during `onMount` (#3218)
- Make autosubscribing to a nullish store a no-op (#2181)

3.17.3

- Fix updating a `<slot>` inside an `{#if}` or other block (#4292)
- Fix using RxJS observables in `derived` stores (#4298)
- Add dev mode check to disallow duplicate keys in a keyed `{#each}` (#4301)
- Fix hydration of `<title>` when starting from SSR-generated code with `hydratable: true` (#4310)

3.17.2

- Fix removing attributes during hydration (#1733)
- Disallow two-way binding to a variable declared by an `{#await}` block (#4012)
- Allow access to `let:` variables in sibling attributes on slot root (#4173)
- Fix `~` and class selector matching against values separated by any whitespace characters (#4242)
- Fix code generation for `awaited` expressions that need parentheses (#4267)
- Preserve JavaScript comments from the original component source where possible (#4268)
- Add some more known globals (#4276)
- Correctly apply event modifiers to `<svelte:body>` events (#4278)

3.17.1

- Only attach SSR mode markers to a component's `<head>` elements when compiling with `hydratable: true` (#4258)

3.17.0

- Remove old `<head>` elements during hydration so they aren't duplicated (#1607)
- Prevent text input cursor jumping in Safari with one-way binding (#3449)
- Expose compiler version in dev events (#4047)
- Don't run actions before their element is in the document (#4166)

- Fix reactive assignments with destructuring and stores where the destructured value should be undefined (#4170)
- Fix hydrating `{:else}` in `{#each}` (#4202)
- Do not automatically declare variables in reactive declarations when assigning to a member expression (#4212)
- Fix stringifying of attributes in SSR mode when there are spread attributes (#4240)
- Only render one `<title>` in SSR mode when multiple components provide one (#4250)

3.16.7

- Also apply actions in the order they're given along with other directives (#2446, #4156)
- Check whether a dynamic event handler is a function before calling it (#4090)
- Correctly mark event handlers as dynamic when they involve an expression used in a `bind:` elsewhere (#4155)

3.16.6

- Fix CSS specificity bug when encapsulating styles (#1277)
- Apply directives in the order they're given (#2446)
- Fix destructuring in `let:` directives (#2751)
- Preserve whitespace around `<tspan>`s in `<svg>`s (#3998)

3.16.5

- Better fix for cascading invalidations and fix some regressions (#4098, #4114, #4120)

3.16.4

- Fix slots with props not propagating through to inner slots (#4061)
- Fix noting autosubscribed stores as **referenced** in **vars** for tooling (#4081)
- Fix cascading invalidations in certain situations (#4094)

3.16.3

- Fix bitmask overflow when using slotted components (#4077)
- Remove unnecessary `$$invalidate` calls from init block (#4018)

3.16.2

- Handle slot updates when parent component has a bitmask overflow (#4078)

3.16.1

- Fix unused export warning for props used as stores (#4021)
- Fix `{:then}` without resolved value containing `{#each}` (#4022)
- Fix incorrect code generated with `loopGuardTimeout` (#4034)
- Fix handling of bitmask overflow and globals (#4037)
- Fix `{:then}` containing `{#if}` (#4044)
- Fix bare imports in `format: 'cjs'` output mode (#4055)
- Warn when using a known global as a component name (#4070)

3.16.0

- Use bitmasks to track changes (#3945)
- Fix heisenbug with component styles (#3977)
- Do not warn about missing expected props for `export function foo()` `{}` (#3954)
- Fix `context="module"` exports with the same name as an instance variable (#3983)
- Fix binding to contextual values from `{#each}` blocks referring to global variables (#3992)
- Use `requestAnimationFrame` callback argument for smoother transitions (#4014)
- Fix `listen_dev` argument order (#4016)

3.15.0

- Hide commented sections from preprocessors (#3894)
- Add `seeking` and `ended` bindings to media elements (#3650)
- Add `videoWidth` and `videoHeight` bindings to video elements (#3927)
- Fix for dynamic event handlers (#3934)
- Handle scale transforms when using the `flip` animation (#3555)
- Fix some code generation bugs (#3929, #3939)
- Add `aria-hidden="true"` to objects generated when adding resize-listeners, to improve accessibility (#3948)

3.14.1

- Deconflict block method names with other variables (#3900)
- Fix entity encoding issue in text nodes with constant expressions (#3911)
- Make code for unknown prop warnings compatible with older js engines (#3914)

3.14.0

- Add `loopGuardTimeout` option that augments `for/while` loops to prevent infinite loops, primarily for use in the REPL (#3887)

- Keep component bindings in sync when changed in reactive statements (#3382)
- Update attributes before bindings (#3857)
- Prevent variable naming conflict (#3899)

3.13.0

- New structured code generation, which eliminates a number of edge cases and obscure bugs (#3539)

Also:

- Fix `{#each}` context not shadowing outer scope when using `bind:` (#1565)
- Fix edge cases in matching selectors against elements (#1710)
- Fix several bugs related to interaction of `{...spread}` attributes with other features (#2721, #2916, #3421, #3681, #3764, #3790)
- Allow exiting a reactive block early with `break $` (#2828)
- Fix binding to props that have been renamed with `export { ... as ... }` (#3508)
- Fix application of style scoping class in cases of ambiguity (#3544)
- Check attributes have changed before setting them to avoid image flicker (#3579)
- Fix generating malformed code for `{@debug}` tags with no dependencies (#3588)
- Fix generated code in specific case involving compound ifs and child components (#3595)
- Fix `bind:this` binding to a store (#3591)
- Use safer `HTMLElement` check before extending class (#3608)
- Add `location` as a known global (#3619)
- Support `{#await}` with `{:catch}` but no `{:then}` (#3623)
- Clean up dead code emitted for `<slot/>`s (#3631)
- Fix tracking of dependencies of compound assignments in reactive statements (#3634)
- Flush changes in newly attached block when using `{#await}` (#3660)
- Throw exception immediately when calling `createEventDispatcher()` after component instantiation (#3667)
- Fix globals shadowing contextual template scope (#3674)
- Fix `<svelte:window>` bindings to stores (#3832)
- Deconflict generated var names with builtins (#3724)
- Allow spring/tweened values to be initially undefined (#3761)
- Warn if using `<svelte:options tag="...">` without `customElement: true` option (#3782)
- Add `Event` to list of known globals (#3810)
- Throw helpful error on empty CSS declaration (#3801)
- Support `easing` param on `fade` transition (#3823)
- Generate valid names from filenames with unicode characters (#3845)
- Don't generate any code for markup-less components (#2200)

- Deconflict with internal name `block` (#3854)
- Set attributes before bindings, to prevent erroneous assignments to `input.files` (#3828)
- Smarter unused CSS detection (#3825)
- Allow dynamic event handlers (#3040)
- Prevent erroneous `"undefined"` class name (#3876)
- Prevent resetting of `src` attribute unless changed (#3579)
- Prevent hydration of void element 'children' (#3882)
- Hoist globals even if mentioned in `<script>` block (#3745)

3.12.1

- Escape `@` symbols in props, again (#3545)

3.12.0

- Fire events on `document` in development to facilitate dev tooling (#3005)
- Remove old props when the keys in spread props are removed (#2282)

3.11.0

- `$capture_state` and `$inject_state` HMR hooks in dev mode (#3148)
- Allow unclosed tags inside `if/each/etc` blocks (#2807)
- Invalidate unreferenced store values inside `<script>` (#3537)
- Print `null` text when hydrating (#3379)

3.10.1

- Preserve reactivity inside `if` block heads etc (#3512)
- Fix store bindings inside `each` blocks (#3455)
- Generate correct code for `if-else` blocks with static conditions (#3505)
- Avoid generating unnecessary component update code (#3526)
- Make `bind:currentTime` more reliable (#3524)
- Prevent errors when setting spread props on SVG elements (#3522)

3.10.0

- Add `blur` transition (#3477)
- Prevent `<input type="number">` edge case with spread props (#3426)
- Robustify cyclical dependency detection, improve errors (#3459)

3.9.2

- Fix handling of additional `@`-rules in style blocks (#2995)
- Fix `if` blocks with complex but static conditions (#3447)

3.9.1

- Only update style properties if necessary (#3433)
- Only update if/await blocks if necessary (#2355)
- Set context correctly inside await blocks (#2443)
- Handle **!important** inline styles (#1834)
- Make index references reactive in event handlers inside keyed each blocks (#2569)

3.9.0

- Support **is** attribute on elements, with a warning (#3182)
- Handle missing slot prop (#3322)
- Don't set undefined/null input values, unless previous value exists (#1233)
- Fix style attribute optimisation bailout (#1830)

3.8.1

- Set SVG namespace for slotted elements (#3321)

3.8.0

- Add **self** event modifier (#3372)
- Generate valid code when spreading literal (#3185)
- Coerce tag values to string before checking equality (#2290)

3.7.1

- Assume **let** variables are dynamic for slots (#3354)
- Allow transition functions to return nothing (#2246)

3.7.0

- Disable warnings via **svelte-ignore** comments (#3351)
- Throw if **\$** or **\$\$...** is referenced as global (#3272)
- Remount HTML tags correctly (#3329)
- Treat data attributes like other attributes (#3337)

3.6.11

- Handle reassigned RxJS observables (#3304)
- Remove commas from HTMLified attributes with multiple chunks (#3341)
- Prevent **class** on element with scoped styles from rendering as **undefined** (#3283)
- Allow references to index in key expression (#3274)
- Mark attribute selectors as used if corresponding binding exists (#3281)
- Preserve **async/*** when hoisting functions (#3179)

- Make **raf** a noop on server (#3324)
- Prevent erroneous ally warning for image input with alt attribute (#3331)
- Add several well-known globals (#3316)

3.6.10

- Use **change** event for file inputs (#3226)
- Always fire reactive declarations with **\$\$props** (#3286)
- More conservative spread prop updates (#3289)
- Quote props if necessary in SSR mode (#3312)

3.6.9

- Always update derived stores with a derived input whose value does not change (#3191)

3.6.8

- Preserve global keyframes that don't match local elements (#3228)
- Fix spread/**class**: combination (#3242)
- Never scope **:root** selector (#3250)
- Prevent trailing commas in function arguments (#3255)

3.6.7

- Prevent corruption of outro callbacks with nested keyed each blocks (#3209)
- Prevent cursor jumping in bound input in Safari (#3199)
- Make resize listener object unfocusable (#3206)

3.6.6

- Prevent dynamic components being detached twice (#3113, #2086)

3.6.5

- Handle RxJS-style observables with **get** (#3153)
- Pass **let** values to bindings (#3140)
- Escape **@** symbols in props (#3173)
- Scale crossfaded elements (#3175)

3.6.4

- Run **onMount** functions in correct order, and before initial **afterUpdate** functions (#2281)
- Fix code transformation for shorthand methods (#2906)
- Fix assignments in inline functions (#3038)

3.6.3

- Fix await block mounting inside removed if block (#1496)
- Update when element references are removed (#2034)
- Don't attempt to serialize non-string values in server-rendered bindings (#2135)
- Recognise dependencies in function expressions (#2693)
- Scope pseudo-class selectors without class/type (#1705)
- Allow nested at-rules (#3135)
- Allow attributes to contain = characters (#3149)

3.6.2

- Fix placement of each-else block (#2917)
- Make context accessible to `bind:this` (#2806)
- Pass hoisted values to slots (#2586)

3.6.1

- Fix escaping of `@` in dev mode debug filename (#3114)

3.6.0

- Add `innerHTML` and `textContent` bindings for `contenteditable` elements (#2996)
- Fix destructuring assignments where targets are member expressions (#3092)
- Deconflict with used globals (#2963)
- Always run `onDestroy` functions, not just for detaching components (#3058)
- Fix scope analysis around catch clauses (#3064)
- Add error constructors to known globals (#3064)
- Always bail out of hoisting on encountering local state in function definition (#3044)
- Fix incorrect merging of top-level text nodes (#3027)
- Handle removal of components in each blocks without props (#3035)
- Only call subscriber once when resubscribing to a store (#3022)
- Check for existence of dynamic component before introing (#3054)
- Sanitize names of bubbled event handlers (#2923)

3.5.4

- Preserve whitespace at the boundaries of `{#each}` blocks (#713)
- Fix dynamic `bind:this` on components (#2333)
- Fix binding to values in a component when it uses `$$props` (#2725)
- Fix parsing ambiguous HTML entities (#3071)

3.5.3

- Don't double-destroy keyed each blocks with outros (#3055)

3.5.2

- Prevent duplicated outros causing errors (#3001)
- Fix automatic name generation (#2843)
- Fix .d.ts stubs (#3009)
- Don't strip non-breaking spaces (#3014)
- Fix `requestAnimationFrame` context (#2933)
- Allow space before attribute value (#3026)
- Remove null/undefined attributes (#1434)
- Fix whitespace in static markup (#3030)

3.5.1

- Accommodate webpack idiosyncracies

3.5.0

- Update package folder structure (#2887)
- Support `once` modifier on component events (#2654)
- Allow empty `<title>` tags (#2980)
- Render textarea binding values inside element (#2975)
- Fix delayed animation glitch (#2871)
- Solve diamond dependencies problem with stores (#2660)
- Fix missing outros inside each blocks (#2689)
- Support animations without transitions (#2908)
- Add missing transition events (#2912)

3.4.4

- Publish type declaration files (#2874)
- Don't trigger updates for unreferenced values (#2865)
- Omit readonly bindings from SSR output (#2339)
- Prevent outdated animation CSS (#2871)
- Repair dynamic `{@html ...}` in head (#2880)
- Don't create unknown prop warnings for internal props, or if component has `$$props` (#2881)

3.4.3

- Add type declaration files for everything (#2842)
- Prevent `svelte/store` being bundled (#2786)
- Warn on unknown props in dev mode (#2840)
- Treat `requestAnimationFrame` as a no-op on the server (#2856)

- Add `raw` property to AST's `Text` nodes (#2714)
- Add `<details bind:open>` (#2854)

3.4.2

- Use empty string for empty data attributes (#2804)
- Support `customElement: true` with no `<svelte:options>` (#2821)
- Add docstrings to `svelte/store` (#2795)

3.4.1

- Handle non-falsy non-function return values from derivers (#2780)
- Allow `spring` to work server-side (#2773)

3.4.0

- Allow custom element to be defined without a `tag` (#2417)
- Fix parsing of quote marks inside attribute values (#2715)
- Convert `svelte/store` to TypeScript (#2733)
- Allow `debug` tags to include hoisted values (#2764)
- Parse error if attribute name is missing = (#1513)
- Allow reactive declarations to depend on mutated `const` values (#2728)

3.3.0

- Allow multiple event listeners on a single node (#2688)
- Allow derivers to return a cleanup function (#2553)
- Support namespaced components (`<Foo.Bar/>`) (#2743)

3.2.2

- Add `window` and `document` to expected globals (#2722)
- Prevent hoisting of functions that depend on reactive state (#2703)
- Generate correct code when slot has no changes (#2697)
- Prevent `Object.prototype`-related bugs (#2696)

3.2.1

- Use same comparison logic for `derived` as for other stores (#2644)
- Invalidate dependencies of reactive declarations (#2444)
- Fix instrumentation of auto-subscription self-assignments (#2681)
- Warn on non-top-level or module-context statements labeled with `$:` (#2176)

3.2.0

- Improve `spring` animations, and add `hard/soft` options (#2627)

- Expose `parse` and `walk` functions (#2661, #2534)
- Support array/object rest in `each` block destructuring patterns (#2647, #2658)
- Use `setAttribute` to change `form` property on form elements (#1742)
- Fix ally warning when `<figcaption>` is non-direct descendant of `<figure>` (#2582)
- Squelch erroneous ‘empty block’ warnings (#1716)
- Fix IE9/10 error with `insertBefore` (#2573)
- Prevent `$$scope` from being spread onto an element (#2520)
- Resubscribe to stores that are assigned to in `<script>` (#2435)
- Allow reactive declarations to depend on `const` variables (#2285)
- Trigger store changes on `UpdateExpression` (#2625)
- Squelch missing prop warning if variable is initialised (#2635)
- Add `alert`, `confirm` and `prompt` to known globals (#2648)

3.1.0

- Allow store subscribe functions to return an object with an `unsubscribe` method, providing native RxJS support (#2549)

3.0.1

- Prevent text input cursor jumping in Safari (#2506)
- Allow assignments to member expressions (#2510)
- Prevent mutually dependent functions causing an infinite during hoisting (#2542)
- Reuse scheduler promise instead of creating new one each time (#2555)
- Various site/docs fixes

3.0.0

- Everything

2.15.4

- IE `classList` fix (#1868)

2.15.3

- Don’t mutate AST

2.15.2

- Expose `stats.props` (#1837)

2.15.1

- Don't throw missing store error when store is declared in component (#1828)

2.15.0

- Event modifiers (#1088)
- Wheel and touch events are passive by default (#1088)
- Add `<svelte:document>` tag (#1484)
- Include binding values in server-rendered HTML (#1205)
- Remove attributes when value is undefined/null (#1434)
- Initialise window scroll from component data (#938)
- Remove references to unused properties in generated code (#1187)
- Add TypeScript definitions for store (#1207)
- Better error for missing store (#1807)

2.14.3

- Account for directive dependencies (#1793)
- Detach each block iterations in each blocks with no update method (#1795)

2.14.2

- Fix issue with nested `{#if}` blocks (#1780)

2.14.1

- Fix block insertion order regression (#1778)
- Fix blocks inside `<svelte:head>` (#1774)
- Better attribute parsing (#1772)
- Fix parse errors inside directives (#1788)

2.14.0

- Refactor internals (#1678)
- Deprecate `onerror` option (#1745)
- Handle edge cases where `destroy` is called before `mount` (#1653)
- Make `scroll` binding more efficient (#1579)
- Make 'readonly property' store error more informative (#1761)

2.13.5

- Fix missing dependencies in shorthand class directives (#1739)

2.13.4

- Support dynamic `import()` in template expressions

2.13.3

- Fix bug with keyed each blocks and nested components (#1706)

2.13.2

- Coalesce simultaneous store/component updates (#1520)
- Fix nested transitions preventing each block item removal (#1617)
- Add `class` directive shorthand and encapsulate styles (#1695)
- Prevent erroneous updates of bound inputs (#1699)

2.13.1

- Coerce second argument to `toggleClass` (#1685)

2.13.0

- Add `class` directive (#890)
- Remove sourcemaps from npm package (#1690)

2.12.1

- Allow actions to take any expression (#1676)
- Run transitions in component context (#1675)
- Correctly set select value on mount (#1666)
- Support `{@debug}` in SSR (#1659)
- Don't treat ` ` as empty whitespace (#1658)
- Fix outros for if blocks with no else (#1688)
- Set `style.cssText` in spread attributes (#1684)

2.12.0

- Initialise actions on mount rather than hydrate (#1653)
- Allow non-existent components to be destroyed (#1677)
- Pass AMD ID from CLI correctly (#1672)
- Minor AST tweaks (#1673, #1674)
- Reduce code duplication in component initialisation (#1670)

2.11.0

- Add `--shared` CLI option (#1649)
- Run first `onstate before` fragment is rendered (#1522)
- Exclude current computed prop from state object (#1544)

2.10.1

- Add sourcemaps to `{@debug}` tags (#1647)

2.10.0

- Add a `{@debug}` tag, for inspecting values in templates in dev mode (#1635)
- Fix dimension bindings in iOS (#1642)

2.9.11

- Pass props to custom elements rather than setting attributes, where appropriate (#875)
- Handle whitespace in lists consistently between SSR and DOM renderers (#1637)
- Improve error for invalid `ref` names (#1613)

2.9.10

- Handle `null` consistently in tags (#1598)
- Support object rest in computed properties (#1540)
- Always update dynamic components when expression changes (#1621)
- Encapsulate local styles inside global styles (#1618)

2.9.9

- Fix attribute name regex (#1623)

2.9.8

- Sanitize spread attributes in SSR — fixes vulnerability CVE-2018-6341 (#1623)

2.9.7

- Allow `<input type=file bind:files>` (#1608)
- Ensure child window exists before removing listener in `addResizeHandler` (#1600)
- Handle transitions in `else` block (#1589)

2.9.6

- Provide more useful error if SSR component attempts to render non-SSR component (#1605)

2.9.5

- Null out refs to dynamic components (#1596)

2.9.4

- Make identifier optional for `then` and `catch` blocks (#1507)
- Group outros correctly (#1575)

2.9.3

- Fix bug when an each block contains transitions but its else branch does not (#1559)
- If an event handler throws an exception, don't block all future calls to that handler (#1573)

2.9.2

- Fix conflict when using multiple if-else blocks, some of which use outros and some of which do not (#1580)
- Fix some cases where `.innerHTML` was being used to create child elements when it shouldn't (#1581)

2.9.1

- Use `template.content` instead of `template` where appropriate (#1571)

2.9.0

- Play outro transitions on `<svelte:component>` if `nestedTransitions` is true (#1568)
- Allow illegal identifiers to be component prop names, for e.g. spreading `data-foo` props (#887)
- Abort transition when node is detached (#1561)
- Only include `transitionManager` when necessary (#1514)

2.8.1

- Fix prefixed animation name replacement (#1556)

2.8.0

- Correctly set store on nested components (to parent store, not root store) (#1538)

2.7.2

- Prevent unnecessary remounts (#1527)
- Allow `refs.*` as callee (#1526)
- Handle empty lists when outroing (#1532)

2.7.1

- Fix spread props with multiple dependencies (#1515)

2.7.0

- Add `__svelte_meta` object to elements in dev mode, containing source info (#1499)
- Fix `bind:online` in dev mode (#1502)
- Update v1 warnings/errors (#1508)
- Transform prefixed keyframes (#1504)

2.6.6

- Fix nested transition bug (#1497)

2.6.5

- Handle cases where only some `if` block branches have outros (#1492)

2.6.4

- Web worker support (#1487)
- Update dynamic component bindings when component changes (#1489)

2.6.3

- Nested transitions respect `skipIntroByDefault` (#1460)
- Always create outro for top-level block (#1470)

2.6.2

- Fix spread+bindings on dynamic components (#1433)
- Abort in-progress animations, if a new one starts (#1458)
- Allow animations to be parameterised (#1462)

2.6.1

- Absolutely position outroing animated nodes (#1457)

2.6.0

- Add `animate` directive (#1454)
- Add `skipIntroByDefault` compiler option and `intro: true` init option (#1448)
- Add `nestedTransitions` compiler option (#1451)
- Component outros, if `nestedTransitions` is true (#1211)
- Allow transition functions to return a function, for inter-transition coordination (#1453)
- Pass `1 - t` as second argument to transition functions (#1452)

2.5.1

- Add new ARIA attributes (#1436)
- Add `Promise` to whitelisted globals (#1441)
- Allow spaces around reserved keyword attributes (#1445)

2.5.0

- Support transitions in `await` blocks (#956)
- Abort outros if block is recreated (#1425)
- Wait until transitions have completed before removing styles (#648)
- Support event shorthand on dynamic components (#1427)
- Various codegen improvements (#1419, #1421, #1422, #1424)
- Correctly handle `await` blocks with no dynamic content (#1417)
- Allow spread props on elements with static attribute tests (#1429)

2.4.4

- Declare missing variable in Store (#1415)
- Always declare spread levels (#1413)

2.4.3

- `ref` directives prevent HTMLified content (#1407)
- Store computed properties update components immediately upon declaration (#1327)

2.4.2

- Evaluate `each` key in child scope (#1397)
- Prevent false negatives and positives when detecting cyclical computed store properties (#1399)
- Only update dynamic component props (#1394)

2.4.1

- Fix DOM event context (#1390)

2.4.0

- Integrate CLI (#1360)
- Allow arbitrary destructuring for each block items, with binding (#1385)
- Each block keys can use arbitrary expressions (#703)
- `bind:offsetWidth`, `bind:offsetHeight`, `bind:clientWidth` and `bind:clientHeight` (#984)
- Leaner generated code for `each` blocks (#1287)

2.3.0

- Allow computed properties to have entire state object as dependency (#1303)
- Fix `stats` when `options.generate` is `false` (#1368)
- Assign custom methods to custom elements (#1369)
- Fix `this` value in custom event handlers (#1297)
- Re-evaluate `each` values lazily (#1286)
- Preserve outer context in `await` blocks (#1251)

2.2.0

- Internal refactoring (#1367)

2.1.1

- Report initial `changed` based on state, not expected props (#1356)
- Set state to empty object, not null, on destroy (#1354)
- Prevent stale state in component event handlers (#1353)

2.1.0

- Allow shorthand imports (#1038)
- Update spread props inside each blocks (#1337)

2.0.0

See the blog post for information on how to upgrade your apps

- New template syntax (#1318)
- Emit ES2015 code, not ES5 (#1348)
- Add `onstate` and `onupdate` hooks, remove `component.observe` method (#1197)
- Use destructuring syntax for computed properties (#1069)

- Change signature of `svelte.compile` (#1298)
- Remove `validate` and `Stylesheet` from public API (#1348)
- Don't typecast numeric attributes (#657)
- Always compile with `Store` support, and cascading disabled (#1348)
- Remove unused `hash` property from AST (#1348)
- Rename `loc` property to `start` in warnings and errors (#1348)

1.64.1

- Fix computed properties in SSR renderer (#1349)

1.64.0

- Deprecate passing a string argument to `component.get` (#1347)

1.63.1

- Allow `observe` method to be overwritten

1.63.0

- Add `onstate` and `onupdate` lifecycle hooks and deprecate `component.observe` (#1197)
- Add `on` and `fire` to `Store`, deprecate `onchange` and `observe` (#1344)
- Require computed properties to have destructured argument in v2 mode (#1069)

1.62.0

- Add a `code` field to errors and warnings (#474)
- When using v2 syntax, do not use interpolation in non-root `<style>` tags (#1339)

1.61.0

- Support v2 syntax with `parser: 'v2'` option (#1318)

1.60.3

- Fix validation of `multiple` attributes on bound `<select>` elements (#1331)

1.60.2

- Fix order of insertions for keyed each blocks with siblings (#1306)
- Bail out of CSS DCE if element has spread attribute (#1300)
- Allow `console` etc in component events (#1278)

- Deconflict against inherited contexts (#1275)
- Make CSS DCE case insensitive (#1269)
- Error on dynamic **multiple** attribute for bound select (#1270)
- Allow custom events on `<:Window>` (#1268)

1.60.1

- Fix spread updates on dynamic components (#1307)

1.60.0

- Spread properties (#195)
- `svelte.compile` returns an object with `{ js, css, ast }` properties, where `js` and `css` are `{ code, map }` objects (#1298)
- Fixed broken compile errors when using Rollup (#1296)

1.59.0

- Deprecate `teardown` in custom event handlers (#531)
- Allow static content in keyed `each` block (#1291)
- Allow empty content in keyed `each` block (#1295)
- Only delete applicable transitions (#1290)

1.58.5

- Allow backtick string literals for `svg`, `tag`, and `props` properties (#1284)
- Fix removal of transition styles under Firefox (#1288)

1.58.4

- Fix initial state regression (#1283)

1.58.3

- Actions run in the context of the component (#1279)
- Set refs when mounting dynamic components (#1280)

1.58.2

- (1.58.1 failed to publish)

1.58.1

- Actions (#1247)
- Support `preserveComments` option in SSR mode (#1265)
- Fix performance regression (#1274)

1.58.0

- Fast row swapping (#588)
- Better error messages for invalid directives (#1242)
- Fix local context variable bugs (#1240, #1254)
- Skip missing property warnings for computed/global properties in dev mode (#1246)
- Add end position to warnings (#1250)

1.57.4

- Deconflict context names (#1229)
- Use `setAttribute` to set input types (#1209)
- Scale transition duration correctly (#1221)

1.57.3

- Fix scoped CSS on static child elements (#1223)

1.57.2

- Fix scoped CSS on SVG elements (#1224)

1.57.1

- Add `each_value` to `contextProps` (#1206)

1.57.0

- Use classes (not attributes) for style encapsulation, and base36-encode hashes (#1118)

1.56.4

- Allow `component` and `state` to be context names (#1213)
- Don't remove `@supports` rules when `cascade: false` (#1215)

1.56.3

- Top-level transitions work inside nested components (#1188)
- Always use internal `_mount` method (#1201)

1.56.2

- Null out `key` for children of keyed each blocks (#1202)

1.56.1

- Fix if-in-each bug (#1195)
- Cross-browser `scrollX`/`scrollY` support (#1175)

1.56.0

- Internal refactor (#1122)
- Use correct context for component events (#1184)
- Allow observing `$foo` in dev mode (#1181)
- Handle dynamic data in default slot (#1144)

1.55.1

- Fix cancellation of store `onchange` handlers (#1177)
- Write `["default"]` instead of `.default` in legacy mode (#1166)
- Upgrade Acorn (#1182)
- Don't warn about capitalisation if `options.name` begins with non-alphabetical character (#1179)

1.55.0

- Add `immutable` compiler option for Svelte and runtime option for `Store` (#1146)
- Fix component store bindings (#1100)
- Fire `oncreate` when custom element is attached (#1117)
- Downgrade empty blocks to a warning (#1156)
- Error on unclosed comment (#1156)

1.54.2

- Prevent `await` blocks using stale state (#1131)
- Prevent erroneous missing data warnings for custom elements (#1065)
- Remove empty selectors in prod mode (#1138)
- Escape attribute values in SSR mode (#1155)
- Remove `<noscript>` elements in DOM mode (#1108)
- Allow hydration of non-root `<script>/<style>` tags (#1163)
- Allow interpolation in non-root `<style>` tags (#1163)

1.54.1

- Hoist destructured references (#1139)
- Add `bind:volume` for media elements (#1143)

1.54.0

- Run `oncreate` hooks depth-first, top-to-bottom (#1135)

- Render boolean attributes correctly in SSR mode (#1109)
- Add **feed** aria role to expected roles when doing ally checks (#1124)
- More complete fix for case sensitive attributes (#1062)
- Handle CLRF line endings in await block comments (#1132)

1.53.0

- Base scoping selectors on **<style>** contents alone (#1091)

1.52.0

- Deconflict referenced globals (#1079)
- Validate contents of **await** blocks (#1061)
- Fire **oncreate** for components in **await** blocks (#1061)
- Automatically fix attribute casing (#1062)
- Fix escaping in **<script>** and **<style>** (#1082)
- Error if invalid characters are used in computed properties, and allow any valid identifier in props (#1083)
- Don't run ally tests on components (#1110)
- Respect **store** option in SSR mode (#1107)

1.51.1

- Only escape **<**, **>** and **&** characters (#1082)

1.51.0

- Lock **scroll** bindings (#1071)
- Escape HTML entities when compiling to static HTML (#1066)
- Apply ally warnings to SVG **<a>** attributes with **xlink:href** (#1008)

1.50.1

- Prevent main fragment being created twice in custom elements (#1064)

1.50.0

- Detect unused/misplaced components (#1039)
- Warn on unused event definitions/transitions (#1051)
- Remove whitespace inside **<:Head>** (#1026)
- Optimise **<title>** (#1027)
- Add **bind: false** option to disable two-way binding (#54)

1.49.3

- Return **html** from SSR compiler **render().toString()** methods (#1044)
- Correctly reinitialise dynamic components (#1040)

- Allow `<option>` outside `<select>` (#1022)
- Fix data references in await-block event handlers (#1032)

1.49.2

- Add `store.umd.js` (#967)
- Warn on use of `this` inside computed properties (#1033)

1.49.1

- Pass `store` to children in SSR mode (#1029)

1.49.0

- Support `store` as a component property (#1028)

1.48.0

- Added `<:Head>` component for injecting contents into document head (#1013)
- SSR `render(...)` method now returns a `{ html, css: { code, map }, head }` object (#1013)
- SSR `renderCss(...)` method is deprecated (#1013)
- Add a `preload` function to components (#1015)
- Expose `this.root` on nested components (#1023)

1.47.2

- Deconflict computed properties against arguments to `_recompute` (#1012)
- Allow `await` blocks in slots (#1018)
- Allow components without slots to have whitespace as only child (#1007)
- Correctly set `toString` on `CompileError` (#1016)

1.47.1

- Sanitize filenames in SSR mode (#1005)

1.47.0

- Support dynamic `import(...)` inside `<script>` tags (#1003)

1.46.1

- `await...then` shorthand (#957)
- Allow dynamic components inside elements (#993)
- Don't use `dataset` on SVG nodes (#982)
- Fix erroneous `<figcaption>` ally warning (#991)

- Handle empty classes when pruning unused selectors (#978)
- Better trimming of SSR'd output (#976)
- Don't add `event` to `expectedProperties` (#972)
- Emit error on bad `set` arguments in dev mode (#990)

1.46.0

- Pass `filename` through to preprocessors (#983)

1.45.0

- Dynamic components (#971)

1.44.2

- Fix `await` blocks with siblings (#974)
- Fix `await` blocks inside `if` blocks (#975)

1.44.1

- Fix bidirectional transitions that reference state (#962)

1.44.0

- Add `svelte.preprocess` (#181, #876)
- Add `{{#await ...}}` blocks (#654)

1.43.1

- Fix parameterised transitions (#962)
- Prevent boolean attributes breaking estree-walker expectations (#961)
- Throw error on cyclical store computations (#964)

1.43.0

- Export `Store` class to manage global state (#930)
- Recognise `aria-current` (#953)
- Support SSR register options including `extensions` (#939)
- Friendlier error for illegal contexts (#934)
- Remove whitespace around `<:Window>` components (#943)

1.42.1

- Correctly append items inside a slotted `each` block (#932)
- Fix `<:Window bind:online/>` (#936)
- Attach globals to state upon initialisation (#908)

1.42.0

- Implement **indeterminate** binding for checkbox inputs (#910)
- Use **<option>** children as **value** attribute if none exists (#928)
- Allow quoted property names in default export and sub-properties (#914)
- Various improvements to generated code for bindings

1.41.4

- Handle self-destructive bindings (#917)
- Prevent **innerHTML** with **<option>** elements (#915)
- Use **dataset** unless **legacy** is true (#858)
- Add **prepare** script to facilitate installing from git (#923)

1.41.3

- Prevent argument name clashes (#911)
- Fix UMD factory arguments (#918)
- Don't attempt to set computed values (#893)
- Fix TypeScript build error (#919)

1.41.2

- Handle attribute selectors with no value (#905)
- Retain **async** keyword when extracting functions (#904)
- Shallow clone **data** on initialisation (#891)

1.41.1

- Fix updates of destructured each blocks (#897)
- Don't warn on **options.*** event handler callee (#900)

1.41.0

- **onwarn** and **onerror** receive default handlers as second arguments (#883)
- Recognise **muted** as boolean property on **<audio>** elements (#886)
- Array destructuring for **each** block contexts (#889)

1.40.2

- Ignore **@apply** and similar in CSS (#871)
- Properly escape CSS in custom elements (#872)

1.40.1

- Always use explicit closing tags with **innerHTML** (#866)
- Escape text in **textContent** and **innerHTML** expressions (#868)

1.40.0

- Short fragment method names (#863)
- Extract declarations out of default export (#756)

1.39.4

- Extract shared init logic (#855)
- Allow `console.*` calls in event handlers (#782)
- Marker comments in output (#823)
- Use `textContent` and `innerHTML` where appropriate (#23)
- Various improvements to generated code

1.39.3

- Allow `slot='...'` inside custom elements (#827)
- Disallow `slot='...'` inside if/each blocks (#849)
- Use correct parent node for slotted if blocks (#850)

1.39.2

- Escape CSS in shadow DOM (#840)
- Fix missing anchor bug inside SVG elements (#843)

1.39.1

- Always use anchors for slotted content (#822)
- Prevent ES6 in helpers (#838)
- Correctly determine whether to use `timeRangesToArray` (#837)

1.39.0

- Always attach fragment to shadow root (#821)
- Add `buffered`, `seekable`, `played` bindings to media elements (#819)
- Quote `class` properties in legacy mode (#830)
- Warn on missing `lang` attribute on `<html>` (#828)

1.38.0

- Compile-time ally warnings (#815)
- Remove redundant input blowback guards (#645)
- Use component name in debugging messages (#781)

1.37.0

- Experimental support for compiling to custom elements (#797)

1.36.0

- Optimize `style` attributes where possible (#455)

1.35.0

- `set` and `get` continue to work until `destroy` is complete (#788)
- Observers of unchanged bound values don't fire incorrectly (#804)
- Nested components with slotted content render correctly in SSR mode (#801)
- More efficient rendering of raw and slotted content (#637)
- Handle unquoted attributes in attribute selectors (#798)

1.34.0

- Support nested `<slot>` elements (#790)
- Attach `options` to instance (#550, #777)
- Error if transitions are applied to component (#791)
- Handle CSS variables in `<style>` tag (#757)

1.33.0

- Replace `{{yield}}` with `<slot>` — adds fallback content named slots, and `options.slots` (#763)

1.32.0

- Allow `anchor` initialisation option, alongside `target` (#784)
- Remove leading CSS selectors safely (#783)

1.31.0

- Add `legacy` compile option, which adds IE9 compatibility (#773)

1.30.0

- Update all component bindings simultaneously (#760)
- Fix `@keyframes` atrules with `from` and `to` selectors (#774)

1.29.3

- Only recompute tag and attribute values if they could have changed (#768)
- Fix CSS scoping with multiple levels of descendant selectors (#767)

1.29.2

- Preserve space before non-parenthesized media query expression (#759)
- Call `create()` on new iterations of static each blocks (#762)
- Use `change` events as well as `input` events to bind range inputs (#740)

1.29.1

- Replace `set` and `destroy` with `noop` when component is destroyed (#744)

1.29.0

- Add static `setup` method (#578)
- Hoist if block selectors (#751)
- More sigil escaping fixes (#750)

1.28.1

- Fix unescaping of special words in SSR mode (#741)

1.28.0

- Support `ref:foo` as a CSS selector (#693)
- Prevent magic-string bugs by only minifying CSS combinators if `cascade: false` (#743)
- Don't throw an error if component is destroyed twice (#643)

1.27.0

- Minify CSS and remove unused styles (#697)
- Optimize ternary expressions when excluding unused CSS (#696)
- Clear refs after `ondestroy` callbacks fire (#706)
- Prevent certain forms of component binding blowback (#721)
- Use helper to encapsulate styles (#375)
- Event propagation shorthand — `on:foo` equals `on:foo='fire("foo", event)'` (#638)
- Allow `refs.*` in event handlers, and validate them (#686)

1.26.2

- Unescape `#` characters (#722)

1.26.1

- Fix select binding regression (#724)

1.26.0

- Enforce correct order of operations when initialising (#708 and #714)
- Ensure data is up-to-date when re-rendering yield block (#711)
- Fix unescaping of strings, preserve at-rules in CSS (#712)
- Preserve whitespace at end of each blocks (#713)

1.25.1

- Better CSS sourcemaps (#716)

1.25.0

- Fix hoisted event handlers (#699)
- Fire `intro.start` and `outro.start` events (#702)
- Preserve order of components in keyed each blocks (#700)
- Add `cssMap` property to compiler output (#698)

1.24.0

- Deconflict names with imports in SSR compiler (#655)
- Improved transition performance (#670)
- Run transitions on initial render (#651)
- Add dev mode warning if `hydrate` is true but `hydratable` was false (#664)
- Manipulate sourcemap to make missing loop values obvious (#683)
- Only add CSS scoping attributes where necessary (#679)
- Warn on unused CSS selectors (#678)
- Fix `<select>` binding in loop (#685)
- Prevent bindings from calling `oncreate` functions prematurely (#694)
- Simpler codegen (#673)

1.23.4

- Don't recreate if blocks incorrectly (#669)

1.23.3

- Pass parameters to `get_block` (#667)

1.23.2

- Fix if blocks being recreated on update (#665)

1.23.1

- Fix each-else blocks that are empty on initial render (#662)

1.23.0

- Hydration (#649)
- Correctly transform CSS selectors with pseudo-elements (#658)

1.22.5

- Fix nested component unmounting bug (#643)

1.22.4

- Include `ast` in `svelte.compile` return value (#632)
- Set initial value of `<select>` binding, if unspecified (#639)
- Mark indirect dependencies of `<select>` bindings (i.e. the dependencies of their `<option>` values) (#639)

1.22.3

- Fix nested component unmounting bug (#625)
- Allow components to have computed member expression bindings (#624)
- Handle empty `<style>` tags (#634)
- Warn on missing component (#623)
- Allow dynamic `type` attribute for unbound inputs (#620)
- Rename `addEventListener` and `removeEventListener` directives (#621)

1.22.2

- Escape template strings correctly in SSR output (#616)
- Prevent magic-string deprecation warning (#617)

1.22.1

- Sanitise event handler names (#612)

1.22.0

- Symmetry between `mount` and `unmount`. This is potentially a breaking change if your components import other components that were precompiled with an earlier version of Svelte (#592)
- Add `cascade` option, which prevents styles affecting child components if `false`, unless selectors are wrapped in `:global(...)` and keyframe declaration IDs are prefixed with `-global-`. This will become the default behaviour in v2 (#583)
- Support binding to computed member expressions (#602)
- Coerce empty string in `number/range` inputs to `undefined`, not `0` (#584)
- Fix insert location of DOM elements in each/if/nested component edge cases (#610)

1.21.0

- Always use `helpers` if referenced, not just for call expressions (#575)
- Fix parsing of `<textarea>` children (#599)
- Treat `<textarea>` value attributes and children as equivalent, and fail validation if both are present (#599)
- Fix `<textarea>` SSR (#599)
- Apply CSS transition styles immediately if transition has delay (#574)
- Ensure `transitionManager` is treeshakeable (#593)
- Fix for environments where `node.style.animation` is undefined (#587)
- Fix order of operations when dealing with `<select>` elements (#590)
- Downgrade ‘invalid callee’ to a warning (#579)
- Convert to TypeScript (#573)

1.20.2

- Fix destruction of compound if-blocks with outros (#572)

1.20.1

- Fix insertion order of `if` blocks and their anchors (#569)

1.20.0

- Faster, better updates of keyed each blocks (#373, #543)
- Use element IDs to robustly track dynamically injected `<style>` tags (#554)
- Abort outros before corresponding intros (#546)
- Generate less code for `if` blocks with `else` blocks (#540)
- Ensure `{{yield}}` block content is injected into the right place (#561)
- Simpler, more readable codegen code (#559)
- Validate transition directives (#564)
- Apply delays to bidirectional transitions (#562)
- Handle all valid HTML entities (#565)
- Fix outros on compound `if` blocks (#565)
- Validation for `<:Window>` tags (#565)
- Increased test coverage (#565)

1.19.1

- Export `generateKeyframes`, so that CSS transitions work

1.19.0

- Experimental support for transitions (#7)
- Use `querySelector(':checked')` instead of `selectedOptions` (#539)
- Stringify helpers before bundling them, to avoid renaming errors (#538)

1.18.2

- Parenthesize if-block conditions (#532)
- Fix parsing of parenthesized expressions (#534)
- Fix error on `bind:checked` that doesn't belong to a checkbox input (#529)

1.18.1

- Allow `destroy()` in event handlers (#523)
- Fix bug with `{{yield}}` blocks following elements (#524)

1.18.0

- Visit `<select>` attributes after children, to ensure options are in the right state (#521)
- Use sibling elements as anchors rather than creating comment nodes wherever possible (#3)

1.17.2

- Replace bad characters when creating variable names based on element names (#516)

1.17.1

- Fixes for static each-else and yield blocks (#509), (#514)
- Code generation tweaks (#504), (#507)

1.17.0

- Add `currentTime`, `duration` and `paused` bindings for media elements (#406)
- Don't treat helpers as dependencies (#492)
- Allow `<:Window>` event handlers to access component state (#497)
- Allow two-way binding to properties named 'component' (#495)
- Group checkbox bindings correctly, to avoid erroneously unchecking siblings (#498)
- Validate two-way bindings (#494)
- Allow dynamic each-block to have static else-block (#501)
- Initialise `<select>` value correctly (#502)

1.16.0

- Better code generation (#489), (#490), (#491)
- Prevent binding blowback on initial render (#488)

1.15.1

- Clone data before merging it with state (#479)
- Apply binding event handlers before user event handlers (#486)

1.15.0

- Dev mode — downgrade ‘missing data’ to a warning, and ignore whitelisted globals (#475)
- Fix `<select>` value binding when options are updated late (#476)
- Throw at compile time if event handler references invalid callee (#473)
- Check for helper function purity (#473)
- Validate `namespace` option (#473)

1.14.1

- Replace bad characters when creating variable names based on attributes (#470)

1.14.0

- Better guard against naming conflicts (#465)
- Better error if getters and setters are used with `methods` (#425)
- Don’t create whitespace nodes inside elements that can’t use them (#189)
- Collapse consecutive `if` statements with the same condition (#450)
- Window `scroll` bindings are bidirectional (#404)
- Add `bind:online` to window (#404)
- In dev mode, throw if read-only properties are set (#404)
- Prevent conflicts with component name (#464)
- Ensure event handler names are deconflicted (#466)

1.13.7

- Fix observers — `defer: true` now triggers callback after DOM is updated (#441)
- Handle empty `computed` property (#452)
- Correctly bind one-way `<select>` value attributes with objects (#423)
- Hoist event handlers inside each blocks, where possible (#456)
- Don’t bind event handler callbacks (#433)
- Internal refactoring and neater code generation (#453)

1.13.6

- Use `assign` helper instead of `Object.assign` for better performance and legacy compatibility (#431)
- Improved code generation (#419), (#440), (#442)

1.13.5

- Read **range** and **number** input values as numbers (#436)
- Better error for **bind:value='{{foo}}'** (#437)

1.13.4

- Prevent unclosed **<script>** tag causing infinite loop (#435)

1.13.3

- Correctly handle **{{true}}**, **{{false}}** and **{{null}}** (#424)
- Update **<select>** value attributes correctly (#423)
- Bind custom event handler callbacks (#428)
- Disallow **import root** (#430)
- Prevent component bindings mutating the wrong object (#432)

1.13.2

- Fix deep component bindings (#420)
- Include **css** property in compiler output (#409)
- Treat functions as mutable objects when recomputing (#413)
- Include magic-string in bundle (#410)
- Disable unneeded Bubl  transformations for slimmer output (#411)

1.13.1

- Prevent infinite loops with pathological component bindings (#398)
- More robust deconflicting (#401)

1.13.0

- Add **<:Window>** meta tag with event listeners, and a handful of bindings (#371)
- Don't uncheck radios incorrectly (#399)

1.12.1

- Deconflict non-helper functions (**addCss** etc) (#388)
- Allow reserved words in tags, e.g. **{{class}}** (#383)

1.12.0

- Shorthand attributes — **<Widget :foo/>** is equivalent to **<Widget foo='{{foo}}' />** (#384)
- Support **bind:group** for radio and checkbox inputs (#311, #312)
- Better sourcemap support for two-way bindings

1.11.4

- Dev mode warning for bad `component.observe` arguments (#369)
- Translate `component.on('teardown', ...)` to `component.on('destroy', ...)` and add dev warning (#365)
- Use shared prototype to save bytes (#378)

1.11.3

- Undo CSS behaviour change in 1.11.2 (#372)
- Pin version of `css-tree` (#370)

1.11.2

- Add component CSS to each document a component is rendered to (#331)

1.11.1

- Fix two-way binding for components inside `each` blocks (#356)

1.11.0

- Add `format: 'eval'` and `svelte.create`, to create components directly from source code (#345)
- Node 4 compatibility (#109)

1.10.3

- Prevent `'</script>'` string occurrence breaking pages (#349)
- Allow reference to whitelisted globals without properties (#333)
- Don't remove ` ` incorrectly (#348)
- `let -> var` in `addCss` block (#351)

1.10.2

- Accept any case for doctype declarations (#336)
- Allow non-top-level `<script>` and `<style>` tags to pass through without processing (#335)

1.10.1

- typecheck argument in `__set` when in dev mode (#342)
- Prevent duplicate helpers in non-shared mode (#337)

1.10.0

- Component self-references with `<:Self/>` (#51)
- Two-way binding with `<select multiple>` (#313)

1.9.1

- Better error for malformed event handlers (#220)
- Allow function expressions in tags (#269)

1.9.0

- Add development warnings (#13), (#320), (#177), (#249)
- Better error if parser encounters an unmatched closing tag (#321)

1.8.1

- Allow implicitly closed elements (#318)
- More informative error messages for unclosed elements/blocks (#258)
- Deprecate `onrender` and `ontearardown` in favour of `oncreate` and `ondestroy` (#40)

1.8.0

- Prevent duplicate imports (#308)
- Use `input` events (not `change`) for all input elements other than `checkbox` and `radio`, and textareas (#309)
- Encapsulate keyframe declarations (#245)

1.7.1

- Deconflict imports and shared helpers (#222)
- Deconflict each-block contexts and reserved words (#222)
- Allow shorthand properties in expressions (#296)

1.7.0

- Include CSS AST in `svelte.parse` output (#302)
- Better handling of CSS parse errors (#302)
- Initialise `<select>` elements with two-way binding correctly (#301)
- Allow local context in component event handlers inside `each` blocks (#290)
- Fix two-way binding for components inside `each` blocks (#290)

1.6.11

- Initialise dynamic `<option>` value correctly (#291)

1.6.10

- Ensure **sources** and **sourcesContent** are populated in sourcemaps, even if none of the original code is used (#295)
- Add **outputFilename** option to populate **file** and **sources** sourcemap properties correctly (#295)

1.6.9

- Don't trigger bindings for torn-down components (#277)
- SSR: Handle two-way bindings (#275)
- Improve performance by checking data has changed before updates (#279)
- Parse CSS with css-tree to prevent transformation errors with unconventional styles (#288)

1.6.8

- Always trigger **onrender**, including when change initiator is a nested component (#263)
- Handle default function parameters in computations (#274)

1.6.7

- SSR: Fix apostrophes (#267)
- Add **xmlns** attributes to SVGs (#262)

1.6.6

- Omit text from comment anchors (#247)
- Handle **xlink** attributes (#264)

1.6.5

- Handle **<!doctype>** declarations (#255)

1.6.4

- Fix updates of yields inside each blocks (20e1b05)
- SSR: Handle attributes with values that begin with a number (#248)
- Handle multiline comments in CSS (#252)

1.6.3

- Fix **{{yield}}** bugs for components inside **if** and **each** blocks (#230, #231)
- Set attributes on **<svg>** elements correctly (#233)
- Add **svelte.VERSION** property to compiler

1.6.2

- Use helpers for `addEventListener`, `removeEventListener`, `setAttribute` (#227)
- Escape `sharedPath` (#229)
- Handle attributes with values that begin with a number (#234)
- Update dependencies

1.6.1

- SSR: Handle component directives at arbitrary positions (#221)
- Provide useful feedback on invalid void closing tag (#224)

1.6.0

- Replace `standalone: false` with `shared: true`, or `shared: 'custom/path/to/shared.js'` (#218)
- Include `shared.js` in package

1.5.0

- Implement `standalone: false` (#9)
- SSR: Handle component directives (216)

1.4.0

- Keyed `each` blocks (#81)

1.3.1

- Remove file extensions from AMD dependencies (#144)
- Throw if `options.name` is illegal (#102)

1.3.0

- SSR compiler: Support `format` option (#196)
- SSR compiler: Don't self-close 'normal' elements (#200)
- Remove leading spaces from scoped CSS (#140)
- Internal refactoring

1.2.5

- Allow whitelisted globals in templates (#185)
- Intercept parse errors with `options.onerror`

1.2.4

- SSR compiler: Implement `{{triples}}` (#197)
- SSR compiler: Escape HTML in tags (#197)

1.2.3

- Add support for `namespace` declaration for SVG (etc) components (#147)
- Throw error if methods or lifecycle hooks are arrow functions that use `this` or `arguments` (#179)
- Use `setAttribute()` for `list` attributes, to preserve link to `<datalist>` (#178)
- Throw error if default export is not an object literal (#190)
- Internal refactoring

1.2.2

- Omit directives in server-side rendering (#163)
- Handle comments in SSR (#165)
- Support calling methods of `event`/`this` in event handlers (#162)
- Remove `mount` from public API (#150)

1.2.1

- Server-side rendering is available as a compiler option (`generate: 'ssr'`) (#159)
- Allow call expressions where function is not in `helpers` (#163)

1.2.0

- Server-side rendering of HTML (#148) and CSS (#154)

1.1.3

- Handle `xmlns` attributes correctly (#142)
- Error on duplicate `<style>/<script>` tags rather than failing silently (#142)
- Don't create whitespace text nodes inside SVG elements (#142)
- Require void elements to be lowercase, to eliminate confusion with components (#137)

1.1.2

- Deconflict variable names (#88, #126)

1.1.1

- Support multiple SVG elements in a component (#130)

1.1.0

- Separate fragment creation from `mount` (#91)
- Trigger `onrender` hook at correct time for nested components (#103)
- Fix keypath dynamic attributes in components (#46)
- Implement `{{yield}}` (#112)
- Optimise teardown (#99)
- Require computed properties to have at least one dependency (#115)
- Support `{{#each ...}}...{{else}}...{{/each}}` (#90)
- Triple mustaches (#35)

1.0.7

- Correctly escape HTML entities (#85)

1.0.6

- Generate useful sourcemaps (#60)

1.0.5

- Ensure compiler only generates ES5 code (#75)
- `get()` without arguments returns entire state object (#73)

1.0.4

- Handle empty attributes in elements and components (#63)
- Detach top-level text nodes inside departing each blocks (#62)

1.0.3

- Better generated code for `if` blocks, especially with `else/elseif` chains (#28)
- Trim unnecessary whitespace from `else/elseif` blocks (#49)
- Handle trailing comments in script tags (#64)

1.0.2

Set `style.cssText` rather than `style` (#44)

1.0.1

- Preserve SVG namespace inside each blocks

- Always use `setAttribute` with SVG elements

1.0.0

- Generate AMD, CJS, IIFE and UMD builds
- Correctly insert text nodes before anchors (#31)

0.3.0

- Fix bug where departing element would unset `ref` incorrectly (#24)
- Basic template validation (#6)
- Fire `onrender` hooks once component is in DOM (#18)
- Only detach nodes when necessary to do so (#26)

0.2.2

- On second thoughts, don't transpile build. Was only really for Uglify's benefit, which is daft

0.2.1

- Transpile build

0.2.0

- Only generate UMD build, for now
- Include dependencies in the build, treat as `devDependencies`
- Faster initial render
- Parent data overrides child defaults
- Remove top-level text nodes on teardown
- Handle `readUntil` errors in parser
- Basic `<select>` binding
- Handle missing data
- Prevent infinite set/observe loops
- Add `bind:foo` shorthand
- `else` and `elseif` blocks
- Hoist imports

0.1.1

- Support unquoted attributes
- Handle entities in attributes
- Inline nested components
- `fire` and `on` methods

0.1.0

- Breaking change – Svelte compiler now generates constructor functions rather than factories (#2)
- SVG support

0.0.2

- First release capable of building TodoMVC

0.0.1

- Just squatting on the package name