



SVELTE • REFERENCE

Compiler errors

animation_duplicate

An element can only have one 'animate' directive

animation_invalid_placement

An element that uses the ``animate:`` directive must be the only child of a keyed ``{#each`` .

animation_missing_key

An element that uses the ``animate:`` directive must be the only child of a keyed ``{#each`` .

attribute_contenteditable_dynamic

'contenteditable' attribute cannot be dynamic if element uses two-way binding

attribute_contenteditable_missing

'contenteditable' attribute is required for `textContent`, `innerHTML` and `innerText` two-way I

attribute_duplicate

Attributes need to be unique



Attribute shorthand cannot be empty

attribute_invalid_event_handler

Event attribute must be a JavaScript expression, not a string

attribute_invalid_multiple

'multiple' attribute must be static if select uses two-way binding

attribute_invalid_name

'%name%' is not a valid attribute name

attribute_invalid_sequence_expression

Sequence expressions are not allowed as attribute/directive values in runes mode, unless v

attribute_invalid_type

'type' attribute must be a static text value if input uses two-way binding

attribute_unquoted_sequence

Attribute values containing `{...}` must be enclosed in quote marks, unless the value only

bind_invalid_expression

bind_invalid_name

```
`bind:%name%` is not a valid binding
```

```
`bind:%name%` is not a valid binding. %explanation%
```

bind_invalid_target

```
`bind:%name%` can only be used with %elements%
```

bind_invalid_value

```
Can only bind to state or props
```

bindable_invalid_location

```
`$bindable()` can only be used inside a `$props()` declaration
```

block_duplicate_clause

```
%name% cannot appear more than once within a block
```

block_invalid_continuation_placement

```
{:...} block is invalid at this position (did you forget to close the preceding element or
```

block_invalid_elseif

block_invalid_placement

```
{#%name% ...} block cannot be %location%
```

block_unclosed

Block was left open

block_unexpected_character

Expected a `"%character%"` character immediately following the opening bracket

block_unexpected_close

Unexpected block closing tag

component_invalid_directive

This type of directive is not valid on components

const_tag_cycle

Cyclical dependency detected: %cycle%

const_tag_invalid_expression

```
{@const ...} must consist of a single variable declaration
```

```
`{@const}` must be the immediate child of `{#snippet}`, `{#if}`, `{:else if}`, `{:else}`,
```

constant_assignment

```
Cannot assign to %thing%
```

constant_binding

```
Cannot bind to %thing%
```

css_empty_declaration

```
Declaration cannot be empty
```

css_expected_identifier

```
Expected a valid CSS identifier
```

css_global_block_invalid_combinator

```
A `:global` selector cannot follow a `%name%` combinator
```

css_global_block_invalid_declaration

```
A top-level `:global {...}` block can only contain rules, not declarations
```

css_global_block_invalid_list

css_global_block_invalid_modifier

A ``:global`` selector cannot modify an existing selector

css_global_block_invalid_modifier_start

A ``:global`` selector can only be modified if it is a descendant of other selectors

css_global_invalid_placement

``:global(...)`` can be at the start or end of a selector sequence, but not in the middle

css_global_invalid_selector

``:global(...)`` must contain exactly one selector

css_global_invalid_selector_list

``:global(...)`` must not contain type or universal selectors when used in a compound selector

css_nesting_selector_invalid_placement

Nesting selectors can only be used inside a rule or as the first selector inside a lone ```

css_selector_invalid

Invalid selector

```
`:global(...)` must not be followed by a type selector
```

debug_tag_invalid_arguments

```
{@debug ...} arguments must be identifiers, not arbitrary expressions
```

declaration_duplicate

```
`%name%` has already been declared
```

declaration_duplicate_module_import

```
Cannot declare a variable with the same name as an import inside `<script module>`
```

derived_invalid_export

```
Cannot export derived state from a module. To expose the current derived value, export a
```

directive_invalid_value

```
Directive value must be a JavaScript expression enclosed in curly braces
```

directive_missing_name

```
`%type%` name cannot be empty
```

dollar_binding_invalid

dollar_prefix_invalid

The `$` prefix is reserved, and cannot be used for variables and imports

each_item_invalid_assignment

Cannot reassign or bind to each block argument in runes mode. Use the array and index variables

effect_invalid_placement

``$effect()`` can only be used as an expression statement

element_invalid_closing_tag

``</%name%>`` attempted to close an element that was not open

element_invalid_closing_tag_autoclosed

``</%name%>`` attempted to close element that was already automatically closed by ``<%reason%>``

element_unclosed

``<%name%>`` was left open

event_handler_invalid_component_modifier

Event modifiers other than `'once'` can only be used on DOM elements

Valid event modifiers are %list%

event_handler_invalid_modifier_combination

The '%modifier1%' and '%modifier2%' modifiers cannot be used together

expected_attribute_value

Expected attribute value

expected_block_type

Expected 'if', 'each', 'await', 'key' or 'snippet'

expected_identifier

Expected an identifier

expected_pattern

Expected identifier or destructure pattern

expected_token

Expected token %token%

expected_whitespace

global_reference_invalid

``%name%`` is an illegal variable name. To reference a global variable called ``%name%``, use

host_invalid_placement

``$host()`` can only be used inside custom element component instances

import_svelte_internal_forbidden

Imports of ``svelte/internal/*`` are forbidden. It contains private runtime code which is s

invalid_arguments_usage

The arguments keyword cannot be used within the template or at the top level of a componen

js_parse_error

`%message%`

legacy_export_invalid

Cannot use ``export let`` in runes mode – use ``$props()`` instead

legacy_props_invalid

legacy_reactive_statement_invalid

``$:`` is not allowed in runes mode, use ``$derived`` or ``$effect`` instead

legacy_rest_props_invalid

Cannot use ``${$restProps}`` in runes mode

let_directive_invalid_placement

``let:`` directive at invalid position

mixed_event_handler_syntaxes

Mixing old (`on:%name%`) and new syntaxes for event handling is not allowed. Use only the one.

module_illegal_default_export

A component cannot have a default export

node_invalid_placement

`%thing%` is invalid inside ``<%parent%>``

HTML restricts where certain elements can appear. In case of a violation the browser will ‘repair’ the HTML in a way that breaks Svelte’s assumptions about the structure of your components. Some examples:

elements)

`<option><div>option a</div></option>` will result in `<option>option a</option>` (the `<div>` is removed)

`<table><tr><td>cell</td></tr></table>` will result in `<table><tbody><tr><td>cell</td></tr></tbody></table>` (a `<tbody>` is auto-inserted)

options_invalid_value

Invalid compiler option: %details%

options_removed

Invalid compiler option: %details%

options_unrecognised

Unrecognised compiler option %keypath%

props_duplicate

Cannot use ``$props()`` more than once

props_illegal_name

Declaring or accessing a prop starting with ``$$`` is illegal (they are reserved for Svelte

props_invalid_identifier

props_invalid_pattern

```
`$props()` assignment must not contain nested properties or computed keys
```

props_invalid_placement

```
`$props()` can only be used at the top level of components as a variable declaration init.
```

reactive_declaration_cycle

```
Cyclical dependency detected: %cycle%
```

render_tag_invalid_call_expression

```
Calling a snippet function using apply, bind or call is not allowed
```

render_tag_invalid_expression

```
`{@render ...}` tags can only contain call expressions
```

render_tag_invalid_spread_argument

```
cannot use spread arguments in `{@render ...}` tags
```

rune_invalid_arguments

```
`%rune%` cannot be called with arguments
```

```
`%rune%` must be called with %args%
```

rune_invalid_computed_property

```
Cannot access a computed property of a rune
```

rune_invalid_name

```
`%name%` is not a valid rune
```

rune_invalid_usage

```
Cannot use `%rune%` rune in non-runes mode
```

rune_missing_parentheses

```
Cannot use rune without parentheses
```

rune_removed

```
The `%name%` rune has been removed
```

rune_renamed

```
`%name%` is now `%replacement%`
```

runes_mode_invalid_import

script_duplicate

A component can have a single top-level `<script>` element and/or a single top-level `<script>`

script_invalid_attribute_value

If the `%name%` attribute is supplied, it must be a boolean attribute

script_invalid_context

If the context attribute is supplied, its value must be "module"

script_reserved_attribute

The `%name%` attribute is reserved and cannot be used

slot_attribute_duplicate

Duplicate slot name '`%name%`' in `<%component%>`

slot_attribute_invalid

slot attribute must be a static value

slot_attribute_invalid_placement

Element with a `slot='...'` attribute must be a child of a component or a descendant of a c

Found default slot content alongside an explicit slot="default"

slot_element_invalid_attribute

`<slot>` can only receive attributes and (optionally) let directives

slot_element_invalid_name

slot attribute must be a static value

slot_element_invalid_name_default

`default` is a reserved word – it cannot be used as a slot name

slot_snippet_conflict

Cannot use `` syntax and `{@render ...}` tags in the same component. Migrate toward:

snippet_conflict

Cannot use explicit children snippet at the same time as implicit children content. Remove:

snippet_invalid_rest_parameter

Snippets do not support rest parameters; use an array instead

snippet_parameter_assignment

snippet_shadowing_prop

This snippet is shadowing the prop ``%prop%`` with the same name

state_invalid_export

Cannot export state from a module if it is reassigned. Either export a function returning

state_invalid_placement

``%rune%(...)`` can only be used as a variable declaration initializer or a class field

store_invalid_scoped_subscription

Cannot subscribe to stores that are not declared at the top level of the component

store_invalid_subscription

Cannot reference store value inside `<script module>`

store_invalid_subscription_module

Cannot reference store value outside a `.svelte`` file

Using a `$` prefix to refer to the value of a store is only possible inside `.svelte` files, where Svelte can automatically create subscriptions when a component is mounted and unsubscribe when the component is unmounted. Consider migrating to runes instead.

```
`style:` directive can only use the `important` modifier
```

style_duplicate

```
A component can have a single top-level `<style>` element
```

svelte_body_illegal_attribute

```
`<svelte:body>` does not support non-event attributes or spread attributes
```

svelte_component_invalid_this

```
Invalid component definition – must be an `{expression}`
```

svelte_component_missing_this

```
`<svelte:component>` must have a 'this' attribute
```

svelte_element_missing_this

```
`<svelte:element>` must have a 'this' attribute with a value
```

svelte_fragment_invalid_attribute

```
`<svelte:fragment>` can only have a slot attribute and (optionally) a let: directive
```

svelte_fragment_invalid_placement

svelte_head_illegal_attribute

```
`<svelte:head>` cannot have attributes nor directives
```

svelte_meta_duplicate

A component can only have one `<%name%>` element

svelte_meta_invalid_content

`<%name%>` cannot have children

svelte_meta_invalid_placement

`<%name%>` tags cannot be inside elements or blocks

svelte_meta_invalid_tag

Valid `<svelte:....>` tag names are `%list%`

svelte_options_deprecated_tag

"tag" option is deprecated – use "customElement" instead

svelte_options_invalid_attribute

```
`<svelte:options>` can only receive static attributes
```

Value must be %list%, if specified

svelte_options_invalid_customelement

"customElement" must be a string literal defining a valid custom element name or an object

svelte_options_invalid_customelement_props

"props" must be a statically analyzable object literal of the form "{ [key: string]: { at

svelte_options_invalid_customelement_shadow

"shadow" must be either "open" or "none"

svelte_options_invalid_tagname

Tag name must be lowercase and hyphenated

See <https://html.spec.whatwg.org/multipage/custom-elements.html#valid-custom-element-name> for more information on valid tag names

svelte_options_reserved_tagname

Tag name is reserved

See <https://html.spec.whatwg.org/multipage/custom-elements.html#valid-custom-element-name> for more information on valid tag names

```
`<svelte:options>` unknown attribute '%name%'
```

svelte_self_invalid_placement

```
`<svelte:self>` components can only exist inside `{#if}` blocks, `{#each}` blocks, `{#snip
```

tag_invalid_name

```
Expected a valid element or component name. Components must have a valid variable name or
```

tag_invalid_placement

```
{@%name% ...} tag cannot be %location%
```

textarea_invalid_content

```
A `<textarea>` can have either a value attribute or (equivalently) child content, but not
```

title_illegal_attribute

```
`<title>` cannot have attributes nor directives
```

title_invalid_content

```
`<title>` can only contain text and {tags}
```

transition_conflict

transition_duplicate

Cannot use multiple `%type%:` directives on a single element

typescript_invalid_feature

TypeScript language features like %feature% are not natively supported, and their use is s

unexpected_eof

Unexpected end of input

unexpected_reserved_word

'%word%' is a reserved word in JavaScript and cannot be used here

void_element_invalid_content

Void elements cannot have children or closing tags

[✎ Edit this page on GitHub](#)

PREVIOUS

[svelte/transition](#)

NEXT

[Compiler warnings](#)

