

# Package ‘calcite’

May 8, 2026

**Title** Bindings to the Calcite Design System 'JavaScript' Component Library

**Version** 1.0.0

**Description** Provides access to the 'Calcite Design System' 'javascript' components via integration with the 'htmltools' and 'shiny' packages. Pre-built and interactive components can be used to generate either static html or interactive web applications. Learn more about the 'Calcite Design System' at <https://developers.arcgis.com/calcite-design-system/>.

**License** Apache License (>= 2)

**Imports** cli, htmltools, rlang, shiny, utils, yyjsonr

**Suggests** brio, rstudioapi

**URL** <https://r.esri.com/calcite/>

**Encoding** UTF-8

**Language** en

**RoxygenNote** 7.3.3

**NeedsCompilation** yes

**Author** Josiah Parry [aut, cre] (ORCID: <https://orcid.org/0000-0001-9910-865X>)

**Maintainer** Josiah Parry <josiah.parry@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-03-03 10:00:20 UTC

## Contents

calcite_accordion . . . . .	4
calcite_accordion_item . . . . .	6
calcite_action . . . . .	8
calcite_action_bar . . . . .	10
calcite_action_group . . . . .	12
calcite_action_pad . . . . .	13

calcite_alert . . . . .	15
calcite_avatar . . . . .	19
calcite_block . . . . .	20
calcite_block_section . . . . .	22
calcite_button . . . . .	24
calcite_card . . . . .	27
calcite_card_group . . . . .	28
calcite_carousel . . . . .	29
calcite_carousel_item . . . . .	31
calcite_checkbox . . . . .	32
calcite_chip . . . . .	33
calcite_chip_group . . . . .	35
calcite_color_picker . . . . .	36
calcite_combobox . . . . .	37
calcite_combobox_item . . . . .	39
calcite_combobox_item_group . . . . .	40
calcite_date_picker . . . . .	41
calcite_dialog . . . . .	43
calcite_dropdown . . . . .	45
calcite_dropdown_group . . . . .	47
calcite_dropdown_item . . . . .	48
calcite_fab . . . . .	49
calcite_filter . . . . .	50
calcite_flow . . . . .	51
calcite_flow_item . . . . .	52
calcite_icon . . . . .	53
calcite_inline_editable . . . . .	54
calcite_input_date_picker . . . . .	56
calcite_input_file . . . . .	57
calcite_input_message . . . . .	59
calcite_input_number . . . . .	61
calcite_input_text . . . . .	65
calcite_input_time_picker . . . . .	68
calcite_input_time_zone . . . . .	69
calcite_label . . . . .	70
calcite_link . . . . .	72
calcite_list . . . . .	74
calcite_list_item . . . . .	76
calcite_list_item_group . . . . .	77
calcite_loader . . . . .	78
calcite_menu . . . . .	79
calcite_menu_item . . . . .	80
calcite_meter . . . . .	81
calcite_modal . . . . .	83
calcite_navigation . . . . .	84
calcite_navigation_logo . . . . .	85
calcite_navigation_user . . . . .	86
calcite_notice . . . . .	87

calcite_option . . . . .	91
calcite_option_group . . . . .	92
calcite_pagination . . . . .	93
calcite_panel . . . . .	94
calcite_popover . . . . .	98
calcite_progress . . . . .	99
calcite_radio_button . . . . .	100
calcite_radio_button_group . . . . .	101
calcite_rating . . . . .	102
calcite_scrim . . . . .	104
calcite_segmented_control . . . . .	105
calcite_segmented_control_item . . . . .	107
calcite_select . . . . .	109
calcite_sheet . . . . .	111
calcite_shell . . . . .	113
calcite_shell_center_row . . . . .	114
calcite_shell_panel . . . . .	115
calcite_slider . . . . .	117
calcite_split_button . . . . .	121
calcite_stepper . . . . .	123
calcite_stepper_item . . . . .	124
calcite_switch . . . . .	125
calcite_tab . . . . .	128
calcite_table . . . . .	129
calcite_table_header . . . . .	131
calcite_tabs . . . . .	131
calcite_tab_nav . . . . .	132
calcite_tab_title . . . . .	133
calcite_text_area . . . . .	135
calcite_tile . . . . .	136
calcite_tile_group . . . . .	138
calcite_time_picker . . . . .	141
calcite_tip . . . . .	142
calcite_tip_group . . . . .	143
calcite_tip_manager . . . . .	144
calcite_tooltip . . . . .	145
calcite_tree . . . . .	146
calcite_tree_item . . . . .	147
calcite_version . . . . .	148
examples . . . . .	149
page_actionbar . . . . .	149
page_navbar . . . . .	151
page_sidebar . . . . .	151
update_calcite . . . . .	153

---

 calcite\_accordion      *Create a Calcite Accordion Component*


---

**Description**

Accordions are most often used for hiding and showing sections of related content. They can have different selection modes, allowing users to have multiple or single accordion item(s) open.

**Usage**

```
calcite_accordion(
  ...,
  id = NULL,
  appearance = NULL,
  icon_position = NULL,
  icon_type = NULL,
  scale = NULL,
  selection_mode = NULL
)
```

**Arguments**

...	One or more <code>calcite_accordion_item()</code> components
id	Component ID (required for Shiny reactivity)
appearance	Visual style: "solid" or "transparent" (default: "solid")
icon_position	Placement of the icon in the header: "start" or "end" (default: "end")
icon_type	Type of icon in the header: "chevron", "caret", or "plus-minus" (default: "chevron")
scale	Size of the component: "s" (small), "m" (medium), or "l" (large) (default: "m")
selection_mode	Selection mode: "multiple" allows any number of selections, "single" allows only one selection, "single-persist" allows one selection and prevents de-selection (default: "multiple")

**Details****Usage:**

- To organize related text-based content.
- To shorten pages and reduce scrolling when content is not crucial to read in full.

**Best practices:**

- Use Accordion to contain related but distinct, primarily text-based content.
- Use Accordion within the main content area of an application to save space.
- Avoid using Accordion within a Panel. Instead, use one or more Block(s).
- Avoid using Accordion to house interactive controls or form-based components.

**Shiny Integration:**

The accordion tracks which items are expanded based on its selection mode.

**Available properties in input\$id:**

- \$appearance - The visual style
- \$iconPosition - Icon placement
- \$iconType - Icon type
- \$scale - Size
- \$selectionMode - Selection mode

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
# Basic accordion with multiple selection
calcite_accordion(
  id = "my_accordion",
  calcite_accordion_item(
    heading = "Watercraft",
    description = "Yachts, boats, and dinghies",
    icon_start = "embark",
    "Recommended for coastal use"
  ),
  calcite_accordion_item(
    heading = "Automobiles",
    description = "Cars, trucks, and buses",
    icon_start = "car",
    "A good choice for inland adventure"
  )
)

# Single selection accordion
calcite_accordion(
  selection_mode = "single",
  appearance = "transparent",
  icon_type = "plus-minus",
  calcite_accordion_item(
    heading = "Section 1",
    "Content 1"
  ),
  calcite_accordion_item(
    heading = "Section 2",
    "Content 2"
  )
)
```

---

 calcite\_accordion\_item

*Create a Calcite Accordion Item Component*


---

### Description

Creates an individual accordion item that can be used within a calcite-accordion. Accordion items can be expanded and collapsed to show or hide content.

### Usage

```
calcite_accordion_item(
  ...,
  id = NULL,
  heading = NULL,
  description = NULL,
  expanded = NULL,
  heading_level = NULL,
  icon_start = NULL,
  icon_end = NULL,
  icon_flip rtl = NULL,
  actions_start = NULL,
  actions_end = NULL,
  content_start = NULL,
  content_end = NULL
)
```

### Arguments

...	Child content for the accordion item (default slot)
id	Component ID (required for Shiny reactivity)
heading	Heading text for the accordion item
description	Description text displayed in the header
expanded	Whether the item is currently expanded (default: FALSE)
heading_level	Semantic heading level (1-6) for accessibility
icon_start	Icon to display at the start of the header
icon_end	Icon to display at the end of the header
icon_flip rtl	Flip icons in RTL languages: "start", "end", or "both"
actions_start	Content for the actions-start slot
actions_end	Content for the actions-end slot
content_start	Content for the content-start slot
content_end	Content for the content-end slot

## Details

### Shiny Integration:

The accordion item emits events when it's expanded or collapsed.

### Available properties in `input$id`:

- `$expanded` - Whether the item is currently expanded
- `$heading` - The heading text
- `$description` - The description text
- Other component properties

### Basic usage:

```
calcite_accordion_item(  
  id = "item1",  
  heading = "Watercraft",  
  description = "Yachts, boats, and dinghies",  
  icon_start = "embark",  
  # Item content...  
)  
  
# In server  
observeEvent(input$item1$expanded, {  
  if (input$item1$expanded) {  
    message("Item was expanded")  
  } else {  
    message("Item was collapsed")  
  }  
})
```

## Value

An object of class `calcite_component`

## References

[Official Documentation](#)

## Examples

```
# Basic accordion item  
calcite_accordion_item(  
  id = "watercraft",  
  heading = "Watercraft",  
  description = "Yachts, boats, and dinghies",  
  icon_start = "embark",  
  "Content goes here..."  
)
```

---

 calcite\_action

*Create a Calcite Action Component*


---

## Description

Creates an action button that can be used within action bars and action groups. Actions provide a compact way to trigger operations with icon-based buttons.

## Usage

```
calcite_action(
  text,
  icon = NULL,
  id = NULL,
  active = NULL,
  disabled = NULL,
  indicator = NULL,
  text_enabled = NULL,
  scale = NULL,
  alignment = NULL,
  appearance = NULL,
  loading = NULL,
  label = NULL,
  ...
)
```

## Arguments

text	Text label for the action (required, also used for accessibility)
icon	Icon name from the Calcite icon set
id	Optional ID for the action (required for Shiny reactivity)
active	Whether the action is highlighted/selected
disabled	Whether the action is disabled
indicator	Whether to show a visual indicator (e.g., notification badge)
text_enabled	Whether to display the text label alongside the icon
scale	Size of the action: "s" (small), "m" (medium), or "l" (large)
alignment	Text alignment: "start", "center", or "end"
appearance	Visual style: "solid" or "transparent"
loading	Whether to show a loading indicator
label	Custom accessibility label (defaults to text if not provided)
...	Additional attributes passed to the component

## Details

### Shiny Integration:

When used in a Shiny app, `calcite_action()` returns a reactive list containing all component properties. You can observe the entire component state or watch individual properties:

### Available properties:

- `$clicked` - Boolean that toggles on each click (use this to detect clicks)
- `$active` - Whether the action is currently active/highlighted
- `$disabled` - Whether the action is disabled
- `$icon` - The icon name
- `$text` - The text label
- `$indicator` - Whether an indicator is shown
- `$alignment`, `$appearance`, `$scale`, etc. - Other component properties

### Usage in server:

```
# Watch for any change to the action (including clicks)
observeEvent(input$my_action, {
  print("Action changed!")
})

# Watch only the clicked state
observeEvent(input$my_action$clicked, {
  print("Action was clicked!")
})

# Access specific properties
observeEvent(input$my_action, {
  is_active <- input$my_action$active
  click_state <- input$my_action$clicked
})
```

## Value

An object of class `calcite_component`

## References

[Official Documentation](#)

## Examples

```
# Basic action with icon
calcite_action(
  text = "Layers",
  icon = "layers",
  id = "layers-action"
)

# Active action with text label
```

```
calcite_action(  
  text = "Settings",  
  icon = "gear",  
  active = TRUE,  
  text_enabled = TRUE  
)  
  
# Action with indicator  
calcite_action(  
  text = "Notifications",  
  icon = "bell",  
  indicator = TRUE  
)  
  
# Shiny example  
if (interactive()) {  
  library(shiny)  
  
  ui <- calcite_shell(  
    calcite_action(  
      id = "my_action",  
      text = "Click me",  
      icon = "check",  
      text_enabled = TRUE  
    ),  
    verbatimTextOutput("status")  
  )  
  
  server <- function(input, output, session) {  
    # Respond to clicks  
    observeEvent(input$my_action$clicked, {  
      message("Action clicked! State: ", input$my_action$clicked)  
    })  
  
    # Display all properties  
    output$status <- renderPrint({  
      input$my_action  
    })  
  }  
  
  shinyApp(ui, server)  
}
```

### Description

Action Bar is composed of `calcite_action()`s used for core operations in the interface. When given an `id`, it acts as a Shiny input that reports the text of the currently active action as `input$id`.

**Usage**

```
calcite_action_bar(
  ...,
  id = NULL,
  expand_disabled = NULL,
  expanded = NULL,
  floating = NULL,
  layout = NULL,
  overflow_actions_disabled = NULL,
  position = NULL,
  scale = NULL
)
```

**Arguments**

...	<a href="#">calcite_action()</a> or <a href="#">calcite_action_group()</a> components
id	Optional ID. When provided, <code>input\$id</code> will contain the text of the currently active action.
expand_disabled	When TRUE, disables the expand/collapse toggle.
expanded	When TRUE, the action bar is expanded showing text labels.
floating	When TRUE, the component is in a floating state.
layout	Layout direction of the actions: "vertical", "horizontal", or "grid".
overflow_actions_disabled	When TRUE, disables automatic overflowing of actions into menus.
position	Position of the component: "start" or "end".
scale	Size of the expand action: "s", "m", or "l".

**Details****Shiny Integration:**

When `id` is provided, `input$id` returns the text of the currently active [calcite\\_action\(\)](#). The action bar manages active state automatically — clicking an action activates it and deactivates all others.

```
observeEvent(input$my_bar, {
  cat("Active action:", input$my_bar, "\n")
})
```

Use `update_calcite()` to programmatically set the active action by passing the text value of the action to activate.

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
calcite_action_bar(
  id = "my_bar",
  calcite_action(text = "Layers", icon = "layers", active = TRUE),
  calcite_action(text = "Legend", icon = "legend")
)
```

---

calcite\_action\_group *Create a Calcite Action Group Component*

---

**Description**

Groups multiple action components together with consistent layout and spacing. Action groups can be used within action bars to organize related actions.

**Usage**

```
calcite_action_group(
  ...,
  id = NULL,
  layout = NULL,
  columns = NULL,
  scale = NULL,
  expanded = NULL,
  label = NULL,
  slot = NULL
)
```

**Arguments**

...	Child calcite_action() components or other content
id	Optional ID for the action group
layout	Layout style: "vertical", "horizontal", or "grid"
columns	Number of columns when layout is "grid" (1-6)
scale	Size of the group: "s" (small), "m" (medium), or "l" (large)
expanded	Whether the group is expanded
label	Accessibility label for the group
slot	Optional slot name (e.g., "bottom-actions" for action bars)

**Value**

An object of class calcite\_component

## References

[Official Documentation](#)

## Examples

```
# Basic action group with vertical layout
calcite_action_group(
  calcite_action(text = "Add", icon = "plus"),
  calcite_action(text = "Edit", icon = "pencil"),
  calcite_action(text = "Delete", icon = "trash")
)

# Horizontal group with custom scale
calcite_action_group(
  layout = "horizontal",
  scale = "1",
  calcite_action(text = "Save", icon = "save"),
  calcite_action(text = "Cancel", icon = "x")
)

# Grid layout with 2 columns
calcite_action_group(
  layout = "grid",
  columns = 2,
  calcite_action(text = "A", icon = "letter-a"),
  calcite_action(text = "B", icon = "letter-b"),
  calcite_action(text = "C", icon = "letter-c"),
  calcite_action(text = "D", icon = "letter-d")
)
```

---

calcite_action_pad	<i>Create a ActionPad component</i>
--------------------	-------------------------------------

---

## Description

Create a ActionPad component

## Usage

```
calcite_action_pad(...)
```

## Arguments

...            named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
actionsEndGroupLabel	actions-end-group-label	Specifies the accessible label for the last calcite-action-group.
expandDisabled	expand-disabled	When true, the expand-toggling behavior is disabled.
expanded	expanded	When true, the component is expanded.
layout	layout	Indicates the layout of the component.
messageOverrides	NA	Use this property to override individual strings used by the component.
overlayPositioning	overlay-positioning	Determines the type of positioning to use for the overlaid content. Using
position	position	Arranges the component depending on the element's dir property.
scale	scale	Specifies the size of the expand calcite-action.

**Events:**

The following events are observed by shiny:

Event	Description
calciteActionPadToggle	Fires when the expanded property is toggled.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-actions to the component.
expand-tooltip	A slot to set the calcite-tooltip for the expand toggle.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_action_pad()
```

---

calcite_alert	<i>Create a Calcite Alert Component</i>
---------------	-----------------------------------------

---

**Description**

Alerts are meant to quickly convey a message to the user. They are ephemeral, workflow-specific messaging that is positioned absolutely over content.

**Usage**

```
calcite_alert(  
  kind = NULL,  
  ...,  
  id = NULL,  
  label = NULL,  
  open = NULL,  
  icon = TRUE,  
  icon_flip rtl = NULL,  
  auto_close = NULL,  
  auto_close_duration = NULL,  
  placement = NULL,  
  queue = NULL,  
  scale = NULL,  
  numbering_system = NULL,  
  title = NULL,  
  message = NULL,  
  link = NULL,  
  actions_end = NULL  
)
```

```
calcite_alert_brand(  
  ...,  
  id = NULL,  
  label = NULL,  
  open = NULL,  
  icon = TRUE,  
  icon_flip rtl = NULL,  
  auto_close = NULL,  
  auto_close_duration = NULL,  
  placement = NULL,  
  queue = NULL,  
  scale = NULL,  
  numbering_system = NULL,  
  title = NULL,  
  message = NULL,  
  link = NULL,  
  actions_end = NULL
```

```
)

calcite_alert_danger(
    ...,
    id = NULL,
    label = NULL,
    open = NULL,
    icon = TRUE,
    icon_flip_rtl = NULL,
    auto_close = NULL,
    auto_close_duration = NULL,
    placement = NULL,
    queue = NULL,
    scale = NULL,
    numbering_system = NULL,
    title = NULL,
    message = NULL,
    link = NULL,
    actions_end = NULL
)

calcite_alert_info(
    ...,
    id = NULL,
    label = NULL,
    open = NULL,
    icon = TRUE,
    icon_flip_rtl = NULL,
    auto_close = NULL,
    auto_close_duration = NULL,
    placement = NULL,
    queue = NULL,
    scale = NULL,
    numbering_system = NULL,
    title = NULL,
    message = NULL,
    link = NULL,
    actions_end = NULL
)

calcite_alert_success(
    ...,
    id = NULL,
    label = NULL,
    open = NULL,
    icon = TRUE,
    icon_flip_rtl = NULL,
    auto_close = NULL,
```

```

    auto_close_duration = NULL,
    placement = NULL,
    queue = NULL,
    scale = NULL,
    numbering_system = NULL,
    title = NULL,
    message = NULL,
    link = NULL,
    actions_end = NULL
)

calcite_alert_warning(
  ...,
  id = NULL,
  label = NULL,
  open = NULL,
  icon = TRUE,
  icon_flip rtl = NULL,
  auto_close = NULL,
  auto_close_duration = NULL,
  placement = NULL,
  queue = NULL,
  scale = NULL,
  numbering_system = NULL,
  title = NULL,
  message = NULL,
  link = NULL,
  actions_end = NULL
)

```

### Arguments

kind	Specifies the kind of alert: "brand", "danger", "info", "success", or "warning" (default: "brand")
...	Child content for the alert
id	Component ID (required for Shiny reactivity)
label	Accessible name for the component (required)
open	When true, displays and positions the component (default: FALSE)
icon	When TRUE, shows a default recommended icon. Alternatively, pass a Calcite UI Icon name to display a specific icon
icon_flip rtl	When TRUE, the icon will be flipped when the element direction is right-to-left
auto_close	When TRUE, the component closes automatically. Recommended for passive, non-blocking alerts (default: FALSE)
auto_close_duration	Duration before auto-close: "fast", "medium", or "slow" (default: "medium")
placement	Placement of the component: "top", "top-start", "top-end", "bottom", "bottom-start", or "bottom-end" (default: "bottom")

queue	Ordering priority when opened: "immediate", "last", or "next" (default: "last")
scale	Size of the component: "s" (small), "m" (medium), or "l" (large) (default: "m")
numbering_system	Unicode numeral system for localization: "arab", "arabext", or "latn"
title	Content for the title slot
message	Content for the message slot
link	Content for the link slot (typically <code>calcite_link()</code> )
actions_end	Content for the actions-end slot (typically <code>calcite_action()</code> components)

## Details

### Best Practices:

While visually similar to Notice, Alert has distinct capabilities and intended use cases:

#### Alert:

- Best for ephemeral, workflow-specific messaging
- Generally should not be open at page load
- Most often displayed as a result of user interaction
- Positioned absolutely over content

#### Notice:

- Best for informative, contextually relevant information
- Can be open at page load, or displayed as a result of user action
- Can be persisted in the interface or made closable

#### Modal:

- Best for long-form messaging or narrative content
- Can contain media or interactive elements
- Generally presented as a result of user invocation

### Writing and Copy:

Alerts are meant to quickly convey a message to the user. Keep copy short and to the point. Consistent Alert structure, verbiage, and copy ensures users have an expected experience.

### Shiny Integration:

#### Available properties in `input$id`:

- `$open` - Whether the alert is currently displayed
- `$kind` - The alert kind
- Other component properties

#### Events:

- `calciteAlertBeforeClose` - Before closing transition begins
- `calciteAlertBeforeOpen` - Before opening transition begins
- `calciteAlertClose` - After close animation completes
- `calciteAlertOpen` - After open animation completes

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
# Basic success alert
calcite_alert_success(
  label = "Success",
  icon = "smile",
  open = TRUE,
  title = "Everything worked out!",
  message = "Take a moment to reflect on your day"
)

# Alert with link
calcite_alert_info(
  label = "Notification",
  open = TRUE,
  title = "Something interesting just happened",
  message = "We thought you might want to take a look",
  link = calcite_link("Take action", href = "#")
)
```

---

calcite\_avatar

*Create a Avatar component*

---

**Description**

Create a Avatar component

**Usage**

```
calcite_avatar(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
------	-----------	-------------

fullName	full-name	Specifies the full name of the user. When label and thumbnail are not defined, specifies the access
label	label	Specifies alternative text when thumbnail is defined, otherwise specifies an accessible label.
scale	scale	Specifies the size of the component.
thumbnail	thumbnail	Specifies the src to an image (remember to add a token if the user is private).
userId	user-id	Specifies the unique id of the user.
username	username	Specifies the username of the user.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_avatar()
```

---

calcite\_block

*Create a Calcite Block Component*

---

**Description**

Creates a collapsible block component designed to house content and controls within a Panel, most often as part of an application layout with Shell Panels.

**Usage**

```
calcite_block(
  ...,
  id = NULL,
  heading = NULL,
  description = NULL,
  collapsible = TRUE,
  expanded = TRUE,
  disabled = NULL,
  loading = NULL,
  icon_start = NULL,
  icon_end = NULL,
  icon_flip_rtl = NULL,
  scale = NULL,
  heading_level = NULL,
  label = NULL,
  drag_disabled = NULL,
  sort_handle_open = NULL,
```

```

    menu_placement = NULL,
    overlay_positioning = NULL
  )

```

### Arguments

...	Child content for the block
id	Component ID (required for Shiny reactivity)
heading	Header text for the block
description	Description text displayed below the heading
collapsible	Whether the block can be collapsed (default: FALSE)
expanded	Whether the block is currently expanded (default: FALSE)
disabled	Whether interaction is prevented (default: FALSE)
loading	Whether to display a busy indicator (default: FALSE)
icon_start	Icon to display at the start of the header
icon_end	Icon to display at the end of the header
icon_flip_rtl	Flip icons in RTL languages: "start", "end", or "both"
scale	Size of the component: "s" (small), "m" (medium), or "l" (large)
heading_level	Semantic heading level (1-6) for accessibility
label	Accessible name for the component
drag_disabled	Prevent dragging when parent Block Group enables it (default: FALSE)
sort_handle_open	Display and position the sort handle (default: FALSE)
menu_placement	Placement of the action menu
overlay_positioning	Positioning type for overlaid content: "absolute" or "fixed"

### Details

#### Shiny Integration:

The block emits events when it's expanded or collapsed, making it easy to track state.

#### Available properties in `input$id`:

- `$expanded` - Whether the block is currently expanded
- `$collapsible` - Whether the block is collapsible
- `$disabled` - Whether the block is disabled
- `$heading` - The heading text
- Other component properties

#### Basic usage:

```

calcite_block(
  id = "my_block",
  heading = "Layer effects",
  description = "Adjust blur, highlight, and more",

```

```
    collapsible = TRUE,  
    expanded = TRUE,  
    icon_start = "effects",  
    # Block content...  
  )  
  
  # In server  
  observeEvent(input$my_block$expanded, {  
    if (input$my_block$expanded) {  
      message("Block was expanded")  
    } else {  
      message("Block was collapsed")  
    }  
  })  
})
```

#### Update from server:

```
# Programmatically expand or collapse the block  
update_calcite("my_block", expanded = TRUE)  
update_calcite("my_block", expanded = FALSE)
```

#### Value

An object of class `calcite_component`

#### References

[Official Documentation](#)

#### Examples

```
# Basic collapsible block  
calcite_block(  
  id = "effects_block",  
  heading = "Layer effects",  
  description = "Adjust blur, highlight, and more",  
  collapsible = TRUE,  
  icon_start = "effects",  
  "Block content goes here..."  
)
```

---

calcite\_block\_section *Create a BlockSection component*

---

#### Description

Create a `BlockSection` component

**Usage**

```
calcite_block_section(...)
```

**Arguments**

```
...          named attributes passed to htmltools::tag()
```

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
iconEnd	icon-end	Specifies an icon to display at the end of the component.
iconFlipRtl	icon-flip-rtl	Displays the iconStart and/or iconEnd as flipped when the element direction is right-to-left.
iconStart	icon-start	Specifies an icon to display at the start of the component.
messageOverrides	NA	Use this property to override individual strings used by the component.
open	open	When true, expands the component and its contents.
status	status	Displays a status-related indicator icon.
text	text	The component header text.
toggleDisplay	toggle-display	Specifies how the component's toggle is displayed, where: "button" sets the toggle to a button.

**Events:**

The following events are observed by shiny:

Event	Description
calciteBlockSectionToggle	Fires when the header has been clicked.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding custom content.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_block_section()
```

---

calcite_button	<i>Create a Calcite Button Component</i>
----------------	------------------------------------------

---

**Description**

Creates an interactive button component that can be used for user actions. Buttons can display text, icons, or both, and support various visual styles.

**Usage**

```
calcite_button(
  ...,
  id = NULL,
  alignment = NULL,
  appearance = NULL,
  disabled = NULL,
  download = NULL,
  form = NULL,
  href = NULL,
  icon_end = NULL,
  icon_flip_rtl = NULL,
  icon_start = NULL,
  kind = NULL,
  label = NULL,
  loading = NULL,
  name = NULL,
  rel = NULL,
  round = NULL,
  scale = NULL,
  split_child = NULL,
  target = NULL,
  type = NULL,
  width = NULL
)
```

**Arguments**

...	Content for the button (default slot) - typically text or other elements
id	Component ID (required for Shiny reactivity)
alignment	Specifies the alignment of the component's elements: "center", "end", "icon-end-space-between", "icon-start-space-between", "space-between", or "start" (default: "center")
appearance	Specifies the appearance style: "outline", "outline-fill", "solid", or "transparent" (default: "solid")
disabled	When TRUE, interaction is prevented and the component is displayed with lower opacity (default: FALSE)

download	Prompts the user to save the linked URL instead of navigating to it. Can be TRUE or a string filename
form	The id of the form that will be associated with the component
href	Specifies the URL of the linked resource (absolute or relative path)
icon_end	Specifies an icon to display at the end of the component
icon_flip_rtl	Displays the iconStart and/or iconEnd as flipped when the element direction is RTL: "both", "end", or "start"
icon_start	Specifies an icon to display at the start of the component
kind	Specifies the kind of the component: "brand", "danger", "inverse", or "neutral" (default: "brand")
label	Accessible name for the component
loading	When TRUE, a busy indicator is displayed (default: FALSE)
name	Specifies the name of the component on form submission
rel	Defines the relationship between the href value and the current document
round	When TRUE, adds a round style to the component (default: FALSE)
scale	Specifies the size of the component: "s" (small), "m" (medium), or "l" (large) (default: "m")
split_child	Specifies if the component is a child of a calcite-split-button: "primary", "secondary", or boolean
target	Specifies where to open the linked document defined in href
type	Specifies the default behavior: "button", "reset", or "submit" (default: "button")
width	Specifies the width of the component: "auto", "full", or "half" (default: "auto"). Note: "half" is deprecated, use "full" instead.

## Details

### Shiny Integration:

When used in a Shiny app, `calcite_button()` returns a reactive list containing component properties and a click counter. You can observe the entire component state or watch individual properties:

### Available properties:

- `$clicks` - Integer count of how many times the button has been clicked
- `$alignment`, `$appearance`, `$disabled`, `$kind`, `$scale`, etc. - All component properties

### Usage in server:

```
# Watch for button clicks using the clicks counter
observeEvent(input$my_button$clicks, {
  print(paste("Button clicked", input$my_button$clicks, "times"))
})

# Or watch for any change to the button
observeEvent(input$my_button, {
  print("Button changed!")
})
```

```
  })  
  
  # Access specific properties  
  observeEvent(input$my_button, {  
    is_disabled <- input$my_button$disabled  
    click_count <- input$my_button$clicks  
  })
```

### Value

An object of class `calcite_component`

### References

[Official Documentation](#)

### Examples

```
# Basic button  
calcite_button(  
  id = "my_button",  
  "Click Me!"  
)  
  
# Button with icon  
calcite_button(  
  "Save",  
  icon_start = "save",  
  appearance = "solid",  
  kind = "brand"  
)  
  
# Outline button with icon at end  
calcite_button(  
  "Delete",  
  icon_end = "trash",  
  appearance = "outline",  
  kind = "danger"  
)  
  
# Round icon-only button  
calcite_button(  
  icon_start = "plus",  
  round = TRUE,  
  label = "Add item"  
)  
  
# Shiny example  
if (interactive()) {  
  library(shiny)  
  
  ui <- div(  
    calcite_button("Save", icon_start = "save", appearance = "solid", kind = "brand"),  
    calcite_button("Delete", icon_end = "trash", appearance = "outline", kind = "danger"),  
    calcite_button("Add item", icon_start = "plus", round = TRUE, label = "Add item")  
  )  
}
```

```

    calcite_button(
      id = "test_button",
      "Click Me!",
      appearance = "solid",
      kind = "brand"
    ),
    verbatimTextOutput("click_count")
  )

server <- function(input, output, session) {
  # Watch for button clicks
  output$click_count <- renderText({
    paste("Button clicked", input$test_button$clicks, "times")
  })
}

shinyApp(ui, server)
}

```

---

 calcite\_card

*Create a Card component*


---

## Description

Create a Card component

## Usage

```
calcite_card(...)
```

## Arguments

...                    named attributes passed to `htmltools::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
label	label	Accessible name for the component.
loading	loading	When true, a busy indicator is displayed.
messageOverrides	NA	Use this property to override individual strings used by the component.
selectable	selectable	When true, the component is selectable.
selected	selected	When true, the component is selected.
thumbnailPosition	thumbnail-position	Sets the placement of the thumbnail defined in the thumbnail slot.

**Events:**

The following events are observed by shiny:

Event	Description
calciteCardSelect	Fires when the deprecated selectable is true, or selectionMode set on parent calcite-card-group is

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding content.
title	<b>Deprecated</b> use heading instead. A slot for adding a heading.
subtitle	<b>Deprecated</b> use description instead. A slot for adding a description.
thumbnail	A slot for adding a thumbnail.
heading	A slot for adding a heading.
description	A slot for adding a description.
footer-start	A slot for adding a leading footer.
footer-end	A slot for adding a trailing footer.

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_card()
```

---

calcite\_card\_group     *Create a CardGroup component*

---

**Description**

Create a CardGroup component

**Usage**

```
calcite_card_group(...)
```

**Arguments**

...                    named attributes passed to htmltools::tag()

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
label	label	Accessible name for the component.
selectedItems	NA	Specifies the component's selected items.
selectionMode	selection-mode	Specifies the selection mode of the component.

**Events:**

The following events are observed by shiny:

Event	Description
calciteCardGroupSelect	Emits when the component's selection changes and the selectionMode is not none.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding one or more calcite-cards.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_card_group()
```

---

calcite_carousel	<i>Create a Carousel component</i>
------------------	------------------------------------

---

**Description**

Create a Carousel component

**Usage**

```
calcite_carousel(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>arrowType</code>	<code>arrow-type</code>	Specifies how and if the "previous" and "next" arrows are displayed.
<code>autoplay</code>	<code>autoplay</code>	When true, the carousel will autoplay and controls will be displayed. When "paused"
<code>autoplayDuration</code>	<code>autoplay-duration</code>	When <code>autoplay</code> is true, specifies in milliseconds the length of time to display each
<code>controlOverlay</code>	<code>control-overlay</code>	Specifies if the component's controls are positioned absolutely on top of slotted Car
<code>disabled</code>	<code>disabled</code>	When true, interaction is prevented and the component is displayed with lower opa
<code>label</code>	<code>label</code>	Accessible name for the component.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>selectedItem</code>	NA	The component's selected <code>calcite-carousel-item</code> .

**Events:**

The following events are observed by shiny:

Event	Description
<code>calciteCarouselChange</code>	Fires when the selected <code>calcite-carousel-item</code> changes.
<code>calciteCarouselPause</code>	Fires when the carousel autoplay state pauses due to a user hovering over the component or focusing
<code>calciteCarouselPlay</code>	Fires when the carousel autoplay is invoked by the user.
<code>calciteCarouselResume</code>	Fires when the carousel autoplay state resumes due to a user no longer hovering over the component
<code>calciteCarouselStop</code>	Fires when the carousel autoplay state is stopped by a user.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding <code>calcite-carousel-items</code> .

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_carousel()
```

---

calcite\_carousel\_item *Create a CarouselItem component*

---

## Description

Create a CarouselItem component

## Usage

```
calcite_carousel_item(...)
```

## Arguments

...                    named attributes passed to `htmltools::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description	Values	Reflects to Attribute
label	label	Accessible name for the component.	string	FALSE
selected	selected	When true, the component is selected.	boolean	TRUE

### Slots:

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding content.

## Value

an object of class `calcite_component` which is a subclass of `shiny.tag`

## References

[Official Documentation](#)

## Examples

```
calcite_carousel_item()
```

---

calcite\_checkbox      *Create a Calcite Checkbox Component*

---

### Description

A checkbox allows users to make a binary selection, typically within a form or as a standalone toggle. Supports indeterminate state for tri-state behavior.

### Usage

```
calcite_checkbox(  
  id = NULL,  
  checked = NULL,  
  disabled = NULL,  
  form = NULL,  
  indeterminate = NULL,  
  label = NULL,  
  label_text = NULL,  
  name = NULL,  
  required = NULL,  
  scale = NULL,  
  status = NULL,  
  value = NULL  
)
```

### Arguments

id	Component ID (required for Shiny reactivity)
checked	When TRUE, the component is checked. Default: FALSE
disabled	When TRUE, interaction is prevented and the component is displayed with lower opacity. Default: FALSE
form	Specifies the id of the component's associated form.
indeterminate	When TRUE, the component is initially indeterminate, independent from its checked value. Visual only. Default: FALSE
label	Specifies an accessible label for the component.
label_text	Specifies the component's label text.
name	Specifies the name of the component. Required to pass the component's value on form submission.
required	When TRUE and the component resides in a form, the component must have a value in order for the form to submit. Default: FALSE
scale	Specifies the size of the component: "s", "m", or "l". Default: "m"
status	Specifies the status of the input field: "idle", "invalid", or "valid". Default: "idle"
value	The component's value.

**Details****Shiny Integration:**

When an id is provided, the checkbox emits its state whenever it changes.

**Available properties in input\$id:**

- \$checked - Whether the checkbox is checked
- \$indeterminate - Whether the checkbox is in indeterminate state
- \$value - The component's value
- \$disabled - Whether the checkbox is disabled
- \$name - The component's name
- \$scale - The component's scale
- \$status - The component's status

**Events:**

- calciteCheckboxChange - Fires when the checked status changes

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
calcite_checkbox(id = "agree", label_text = "I agree to the terms")
```

```
calcite_checkbox(
  id = "opt_in",
  label_text = "Subscribe to newsletter",
  checked = TRUE,
  scale = "l"
)
```

---

calcite\_chip

*Create a Chip component*

---

**Description**

Create a Chip component

**Usage**

```
calcite_chip(...)
```

**Arguments**

...            named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
appearance	appearance	Specifies the appearance style of the component.
closable	closable	When true, a close button is added to the component.
closed	closed	When true, hides the component.
closeOnDelete	close-on-delete	When true, the component closes when the Delete or Backspace key is pressed while
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
icon	icon	Specifies an icon to display.
iconFlipRtl	icon-flip-rtl	When true, the icon will be flipped when the element direction is right-to-left ("rtl")
kind	kind	Specifies the kind of the component, which will apply to border and background if applicable.
label	label	Accessible name for the component.
messageOverrides	NA	Use this property to override individual strings used by the component.
scale	scale	Specifies the size of the component. When contained in a parent calcite-chip-group
selected	selected	When true, the component is selected.
value	value	The component's value.

**Events:**

The following events are observed by shiny:

Event	Description
calciteChipClose	Fires when the component's close button is selected.
calciteChipSelect	Fires when the selected state of the component changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding text.
image	A slot for adding an image.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_chip()
```

---

calcite\_chip\_group      *Create a ChipGroup component*

---

**Description**

Create a ChipGroup component

**Usage**

```
calcite_chip_group(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
label	label	Accessible name for the component.
scale	scale	Specifies the size of the component. Child calcite-chips inherit the component's value.
selectedItems	NA	Specifies the component's selected items.
selectionMode	selection-mode	Specifies the selection mode of the component, where: "multiple" allows any number of

**Events:**

The following events are observed by shiny:

Event	Description
calciteChipGroupSelect	Fires when the component's selection changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding one or more calcite-chips.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_chip_group()
```

---

```
calcite_color_picker Create a ColorPicker component
```

---

**Description**

Create a ColorPicker component

**Usage**

```
calcite_color_picker(...)
```

**Arguments**

...                    named attributes passed to `htmltools:::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>allowEmpty</code>	<code>allow-empty</code>	When <code>true</code> , an empty color ( <code>null</code> ) will be allowed as a value. When <code>false</code> , a color is required.
<code>alphaChannel</code>	<code>alpha-channel</code>	When <code>true</code> , the component will allow updates to the color's alpha value.
<code>channelsDisabled</code>	<code>channels-disabled</code>	When <code>true</code> , hides the RGB/HSV channel inputs.
<code>clearable</code>	<code>clearable</code>	When <code>true</code> , an empty color ( <code>null</code> ) will be allowed as a value. When <code>false</code> , a color is required.
<code>disabled</code>	<code>disabled</code>	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.
<code>format</code>	<code>format</code>	The format of value. When "auto", the format will be inferred from value when possible.
<code>hexDisabled</code>	<code>hex-disabled</code>	When <code>true</code> , hides the hex input.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>numberingSystem</code>	<code>numbering-system</code>	Specifies the Unicode numeral system used by the component for localization.
<code>savedDisabled</code>	<code>saved-disabled</code>	When <code>true</code> , hides the saved colors section.
<code>scale</code>	<code>scale</code>	Specifies the size of the component.
<code>storageId</code>	<code>storage-id</code>	Specifies the storage ID for colors.
<code>value</code>	<code>value</code>	The component's value, where the value can be a CSS color string, or a RGB, HSL, or hex string.

**Events:**

The following events are observed by shiny:

Event	Description
calciteColorPickerChange	Fires when the color value has changed.
calciteColorPickerInput	Fires as the color value changes. Similar to the calciteColorPickerChange event with the exception that it fires as the user is typing.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_color_picker()
```

---

calcite_combobox	<i>Create a Combobox component</i>
------------------	------------------------------------

---

**Description**

Create a Combobox component

**Usage**

```
calcite_combobox(...)
```

**Arguments**

... named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
allowCustomValues	allow-custom-values	When <code>true</code> , allows entry of custom values, which are not in the original set of values.
clearDisabled	clear-disabled	When <code>true</code> , the value-clearing will be disabled.
disabled	disabled	When <code>true</code> , interaction is prevented and the component is displayed with a disabled appearance.
filteredItems	NA	Specifies the component's filtered items.
filterProps	NA	Specifies the properties to match against when filtering. If not set, all properties are used.
filterText	filter-text	Text for the component's filter input field.
flipPlacements	NA	Specifies the component's fallback slotted content placement when it's in a dropdown menu.
form	form	The id of the form that will be associated with the component. When not set, the component will be associated with the form of the parent component.
label	label	Accessible name for the component.
maxItems	max-items	Specifies the maximum number of <code>calcite-combobox-items</code> (including the filter input field) to display.

messageOverrides	NA	Use this property to override individual strings used by the component.
name	name	Specifies the name of the component. Required to pass the component's
open	open	When true, displays and positions the component.
overlayPositioning	overlay-positioning	Determines the type of positioning to use for the overlaid content. Using
placeholder	placeholder	Specifies the placeholder text for the input.
placeholderIcon	placeholder-icon	Specifies the placeholder icon for the input.
placeholderIconFlipRtl	placeholder-icon-flip-rtl	When true, the icon will be flipped when the element direction is right-to
readOnly	read-only	When true, the component's value can be read, but controls are not acces
required	required	When true and the component resides in a form, the component must ha
scale	scale	Specifies the size of the component.
selectedItems	NA	Specifies the component's selected items.
selectionDisplay	selection-display	When selectionMode is "ancestors" or "multiple", specifies the disp
selectionMode	selection-mode	Specifies the selection mode of the component, where: "multiple" allow
status	status	Specifies the status of the input field, which determines message and icon
validationIcon	validation-icon	Specifies the validation icon to display under the component.
validationMessage	validation-message	Specifies the validation message to display under the component.
validity	NA	The current validation state of the component.
value	value	The component's value(s) from the selected calcite-combobox-item(s)

**Events:**

The following events are observed by shiny:

Event	Description
calciteComboboxBeforeClose	Fires when the component is requested to be closed, and before the closing transition begins.
calciteComboboxBeforeOpen	Fires when the component is added to the DOM but not rendered, and before the opening tra
calciteComboboxChange	Fires when the selected item(s) changes.
calciteComboboxChipClose	Fires when a selected item in the component is closed via its calcite-chip.
calciteComboboxClose	Fires when the component is closed and animation is complete.
calciteComboboxFilterChange	Fires when text is added to filter the options list.
calciteComboboxOpen	Fires when the component is open and animation is complete.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-combobox-items.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_combobox()
```

---

```
calcite_combobox_item Create a ComboboxItem component
```

---

**Description**

Create a ComboboxItem component

**Usage**

```
calcite_combobox_item(...)
```

**Arguments**

```
...          named attributes passed to htmltools::tag()
```

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
active	active	When true, the component is active.
ancestors	NA	Specifies the parent and grandparent items, which are set on calcite-combobox.
description	description	A description for the component, which displays below the heading.
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
filterDisabled	filter-disabled	When true, omits the component from the calcite-combobox filtered search results.
guid	guid	The id attribute of the component. When omitted, a globally unique identifier is used.
heading	heading	The component's text.
icon	icon	Specifies an icon to display.
iconFlipRtl	icon-flip-rtl	When true, the icon will be flipped when the element direction is right-to-left ("rtl").
label	label	The component's label.
metadata	NA	Provides additional metadata to the component used in filtering.
selected	selected	When true, the component is selected.
shortHeading	short-heading	The component's short heading. When provided, the short heading will be displayed in the component.
textLabel	text-label	The component's text.
value	value	The component's value.

**Events:**

The following events are observed by shiny:

Event	Description
calciteComboboxItemChange	Fires whenever the component is selected or unselected.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding nested calcite-combobox-items.
content-end	A slot for adding non-actionable elements after the component's content.
content-start	A slot for adding non-actionable elements before the component's content.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_combobox_item()
```

---

```
calcite_combobox_item_group
  Create a ComboboxItemGroup component
```

---

**Description**

Create a ComboboxItemGroup component

**Usage**

```
calcite_combobox_item_group(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
ancestors	NA	Specifies the parent and grandparent calcite-combobox-items, which are set on calcite-combobox-items.
label	label	Specifies the title of the component.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-combobox-items.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_combobox_item_group()
```

---

`calcite_date_picker`    *Create a Calcite Date Picker Component*

---

**Description**

Provides a calendar interface for selecting dates. Supports single date selection or date range selection with various customization options.

**Usage**

```
calcite_date_picker(  
  id = NULL,  
  value = NULL,  
  range = NULL,  
  min = NULL,  
  max = NULL,  
  scale = NULL,  
  layout = NULL,  
  calendars = NULL,  
  active_range = NULL,  
  heading_level = NULL,  
  month_style = NULL,  
  numbering_system = NULL,  
  proximity_selection_disabled = NULL,  
  ...  
)
```

**Arguments**

<code>id</code>	Optional ID for the date picker (required for Shiny reactivity)
<code>value</code>	Selected date as a string ("yyyy-mm-dd"), or array of strings for range values
<code>range</code>	When TRUE, activates range mode to allow start and end dates (default: FALSE)
<code>min</code>	Earliest allowed date ("yyyy-mm-dd")
<code>max</code>	Latest allowed date ("yyyy-mm-dd")
<code>scale</code>	Specifies size of the component: "s" (small), "m" (medium), or "l" (large)
<code>layout</code>	Defines the layout: "horizontal" or "vertical"
<code>calendars</code>	Number of calendars displayed when range is TRUE: 1 or 2 (default: 2)
<code>active_range</code>	When range is TRUE, specifies active range: "start" or "end"
<code>heading_level</code>	Heading level for document structure (1-6)
<code>month_style</code>	Month display style: "wide" (default) or "abbreviated"
<code>numbering_system</code>	Unicode numeral system: "latn", "arab", or "arabext"
<code>proximity_selection_disabled</code>	When TRUE, disables default range narrowing behavior (default: FALSE)
<code>...</code>	Additional attributes passed to the component

**Details****Date Formats:**

- Dates should be provided as strings in "yyyy-mm-dd" format
- For range mode, provide an array: `c("2024-01-15", "2024-01-20")`
- The component always returns dates as an array of strings, even in single date mode

**Shiny Integration:**

When used in a Shiny app with an `id`, `calcite_date_picker()` returns a reactive list containing component properties.

**Available properties:**

- `$value` - Array of selected date strings (always an array, even for single dates)
- `$range` - Whether range mode is enabled
- `$min` - Minimum allowed date
- `$max` - Maximum allowed date
- Other component properties

**Usage in server:**

```
# Single date mode
observeEvent(input$my_date$value, {
  selected_date <- input$my_date$value[1] # Get first (and only) date
  message("Selected: ", selected_date)
})

# Range mode
```

```
observeEvent(input$my_date_range$value, {
  start_date <- input$my_date_range$value[1]
  end_date <- input$my_date_range$value[2]
  message("Range: ", start_date, " to ", end_date)
})
```

## Value

An object of class `calcite_component`

## References

[Official Documentation](#)

## Examples

```
# Basic date picker
calcite_date_picker(
  id = "my_date"
)

# Date picker with initial value
calcite_date_picker(
  id = "event_date",
  value = "2024-12-25"
)

# Date picker with min/max constraints
calcite_date_picker(
  id = "booking_date",
  min = "2024-01-01",
  max = "2024-12-31"
)

# Range date picker
calcite_date_picker(
  id = "date_range",
  range = TRUE,
  value = c("2024-01-15", "2024-01-20")
)
```

---

calcite\_dialog

*Create a Dialog component*

---

## Description

Create a Dialog component

**Usage**

```
calcite_dialog(...)
```

**Arguments**

```
...          named attributes passed to htmltools::tag()
```

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
beforeClose	NA	Passes a function to run before the component closes.
closeDisabled	close-disabled	When true, disables the component's close button.
description	description	A description for the component.
dragEnabled	drag-enabled	When true, the component is draggable.
escapeDisabled	escape-disabled	When true, disables the default close on escape behavior. By default, an open component will close on escape.
heading	heading	The component header text.
headingLevel	heading-level	Specifies the heading level of the component's heading for proper document structure.
kind	kind	Specifies the kind of the component, which will style the top border.
loading	loading	When true, a busy indicator is displayed.
menuOpen	menu-open	When true, the action menu items in the header-menu-actions slot are open.
messageOverrides	NA	Use this property to override individual strings used by the component.
modal	modal	When true, displays a scrim blocking interaction underneath the component.
open	open	When true, displays and positions the component.
outsideCloseDisabled	outside-close-disabled	When true, disables the closing of the component when clicked outside.
overlayPositioning	overlay-positioning	Determines the type of positioning to use for the overlaid content. Using "absolute" will allow the component to be positioned over other elements.
placement	placement	Specifies the placement of the dialog.
resizable	resizable	When true, the component is resizable.
scale	scale	Specifies the size of the component.
width	width	Specifies the width of the component.
widthScale	width-scale	Specifies the width of the component.

**Events:**

The following events are observed by shiny:

Event	Description
calciteDialogBeforeClose	Fires when the component is requested to be closed and before the closing transition begins.
calciteDialogBeforeOpen	Fires when the component is added to the DOM but not rendered, and before the opening transition begins.
calciteDialogClose	Fires when the component is closed and animation is complete.
calciteDialogOpen	Fires when the component is open and animation is complete.
calciteDialogScroll	Fires when the content is scrolled.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding content.
content	<b>Deprecated</b> Use custom-content slot instead.
custom-content	A slot for displaying custom content. Will prevent the rendering of any default Dialog UI, except for b
action-bar	A slot for adding a calcite-action-bar to the component.
alerts	A slot for adding calcite-alerts to the component.
content-bottom	A slot for adding content below the unnamed (default) slot and - if populated - the footer slot.
content-top	A slot for adding content above the unnamed (default) slot and - if populated - below the action-bar
header-actions-start	A slot for adding actions or content to the starting side of the component's header.
header-actions-end	A slot for adding actions or content to the ending side of the component's header.
header-content	A slot for adding custom content to the component's header.
header-menu-actions	A slot for adding an overflow menu with actions inside a calcite-dropdown.
fab	A slot for adding a calcite-fab (floating action button) to perform an action.
footer	A slot for adding custom content to the component's footer. Should not be used with the "footer-sta
footer-end	A slot for adding a trailing footer custom content. Should not be used with the "footer" slot.
footer-start	A slot for adding a leading footer custom content. Should not be used with the "footer" slot.

## Value

an object of class `calcite_component` which is a subclass of `shiny.tag`

## References

[Official Documentation](#)

## Examples

```
calcite_dialog()
```

---

calcite\_dropdown      *Create a Dropdown component*

---

## Description

Create a Dropdown component

## Usage

```
calcite_dropdown(...)
```

## Arguments

...      named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
closeOnSelectDisabled	close-on-select-disabled	When true, the component will remain open after a selection is made. If
disabled	disabled	When true, interaction is prevented and the component is displayed with
flipPlacements	NA	Specifies the component's fallback calcite-dropdown-item placement
maxItems	max-items	Specifies the maximum number of calcite-dropdown-items to display
open	open	When true, displays and positions the component.
overlayPositioning	overlay-positioning	Determines the type of positioning to use for the overlaid content. Using
placement	placement	Determines where the component will be positioned relative to the contain
scale	scale	Specifies the size of the component.
selectedItems	NA	Specifies the component's selected items.
type	type	Specifies the action to open the component from the container element.
width	width	Specifies the width of the component.
widthScale	width-scale	Specifies the width of the component.

**Events:**

The following events are observed by shiny:

Event	Description
calciteDropdownBeforeClose	Fires when the component is requested to be closed and before the closing transition begins.
calciteDropdownBeforeOpen	Fires when the component is added to the DOM but not rendered, and before the opening tran
calciteDropdownClose	Fires when the component is closed and animation is complete.
calciteDropdownOpen	Fires when the component is open and animation is complete.
calciteDropdownSelect	Fires when a calcite-dropdown-item's selection changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-dropdown-group elements. Every calcite-dropdown-item must have a pa
trigger	A slot for the element that triggers the calcite-dropdown.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_dropdown()
```

---

 calcite\_dropdown\_group

*Create a DropdownGroup component*


---

**Description**

Create a DropdownGroup component

**Usage**

```
calcite_dropdown_group(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
groupTitle	group-title	Specifies and displays a group title.
selectionMode	selection-mode	Specifies the selection mode of the component, where: "multiple" allows any number of

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-dropdown-items.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_dropdown_group()
```

---

calcite\_dropdown\_item *Create a DropdownItem component*

---

### Description

Create a DropdownItem component

### Usage

```
calcite_dropdown_item(...)
```

### Arguments

...                    named attributes passed to `htmltools::tag()`

### Details

#### Properties:

The following properties are provided by this component:

Name	Attribute	Description
disabled	disabled	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.
href	href	Specifies the URL of the linked resource, which can be set as an absolute or relative path. Determin
iconEnd	icon-end	Specifies an icon to display at the end of the component.
iconFlipRtl	icon-flip-rtl	Displays the <code>iconStart</code> and/or <code>iconEnd</code> as flipped when the element direction is right-to-left (" <code>r</code> ")
iconStart	icon-start	Specifies an icon to display at the start of the component.
label	label	Accessible name for the component.
rel	rel	Specifies the relationship to the linked document defined in <code>href</code> .
selected	selected	When <code>true</code> , the component is selected.
target	target	Specifies the frame or window to open the linked document.

#### Events:

The following events are observed by shiny:

Event	Description
calciteDropdownItemSelect	Fires when the component is selected.

#### Slots:

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding text.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_dropdown_item()
```

---

calcite_fab	<i>Create a Fab component</i>
-------------	-------------------------------

---

**Description**

Create a Fab component

**Usage**

```
calcite_fab(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description	Value
appearance	appearance	Specifies the appearance style of the component.	"outl
disabled	disabled	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.	boole
icon	icon	Specifies an icon to display.	string
iconFlipRtl	icon-flip-rtl	When <code>true</code> , the icon will be flipped when the element direction is right-to-left (" <code>rtl</code> ").	boole
kind	kind	Specifies the kind of the component, which will apply to border and background.	"bran
label	label	Accessible name for the component.	string
loading	loading	When <code>true</code> , a busy indicator is displayed.	boole
scale	scale	Specifies the size of the component.	"l"
text	text	Specifies text to accompany the component's icon.	string
textEnabled	text-enabled	When <code>true</code> , displays the text value in the component.	boole

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_fab()
```

---

calcite_filter	<i>Create a Filter component</i>
----------------	----------------------------------

---

**Description**

Create a Filter component

**Usage**

```
calcite_filter(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
filteredItems	NA	The component's resulting items after filtering.
filterProps	NA	Specifies the properties to match against when filtering. This will only apply when value is not NA.
items	NA	Defines the items to filter. The component uses the values as the starting point, and returns the filtered items.
label	label	Specifies an accessible name for the component.
messageOverrides	NA	Use this property to override individual strings used by the component.
placeholder	placeholder	Specifies placeholder text for the input element.
scale	scale	Specifies the size of the component.
value	value	The component's value.

**Events:**

The following events are observed by shiny:

Event	Description
calciteFilterChange	Fires when the filter text changes.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_filter()
```

---

calcite_flow	<i>Create a Flow component</i>
--------------	--------------------------------

---

**Description**

Create a Flow component

**Usage**

```
calcite_flow(...)
```

**Arguments**

...            named attributes passed to `htmltools::tag()`

**Details****Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding <code>calcite-flow-item</code> elements to the component.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_flow()
```

---

calcite\_flow\_item      *Create a FlowItem component*

---

## Description

Create a FlowItem component

## Usage

```
calcite_flow_item(...)
```

## Arguments

...                  named attributes passed to `htmltools::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
<code>beforeBack</code>	NA	When provided, the method will be called before it is removed from its parent <code>calcite_flow_item</code> .
<code>beforeClose</code>	NA	Passes a function to run before the component closes.
<code>closable</code>	<code>closable</code>	When true, displays a close button in the trailing side of the component's header.
<code>closed</code>	<code>closed</code>	When true, the component will be hidden.
<code>collapsed</code>	<code>collapsed</code>	When true, hides the component's content area.
<code>collapsible</code>	<code>collapsible</code>	When true, the component is collapsible.
<code>description</code>	<code>description</code>	A description for the component.
<code>disabled</code>	<code>disabled</code>	When true, interaction is prevented and the component is displayed with lower opacity.
<code>heading</code>	<code>heading</code>	The component header text.
<code>headingLevel</code>	<code>heading-level</code>	Specifies the heading level of the component's heading for proper document structure.
<code>loading</code>	<code>loading</code>	When true, a busy indicator is displayed.
<code>menuOpen</code>	<code>menu-open</code>	When true, the action menu items in the <code>header-menu-actions</code> slot are open.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>overlayPositioning</code>	<code>overlay-positioning</code>	Determines the type of positioning to use for the overlaid content. Using "absolute" will position the content relative to the component's bounding box. Using "fixed" will position the content relative to the browser window.
<code>scale</code>	<code>scale</code>	Specifies the size of the component.
<code>selected</code>	<code>selected</code>	When true, flow-item is displayed within a parent flow.
<code>showBackButton</code>	<code>show-back-button</code>	When true, displays a back button in the component's header.

### Events:

The following events are observed by shiny:

Event	Description
<code>calciteFlowItemBack</code>	Fires when the back button is clicked.
<code>calciteFlowItemClose</code>	Fires when the close button is clicked.
<code>calciteFlowItemScroll</code>	Fires when the content is scrolled.

calciteFlowItemToggle Fires when the collapse button is clicked.

### Slots:

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding custom content.
action-bar	A slot for adding a calcite-action-bar to the component.
alerts	A slot for adding calcite-alerts to the component.
content-top	A slot for adding content above the unnamed (default) slot and below the action-bar slot (if populated)
content-bottom	A slot for adding content below the unnamed (default) slot and above the footer slot (if populated)
header-actions-start	A slot for adding calcite-actions or content to the start side of the component's header.
header-actions-end	A slot for adding calcite-actions or content to the end side of the component's header.
header-content	A slot for adding custom content to the component's header.
header-menu-actions	A slot for adding an overflow menu with calcite-actions inside a calcite-dropdown.
fab	A slot for adding a calcite-fab (floating action button) to perform an action.
footer	A slot for adding custom content to the component's footer. Should not be used with the "footer-start" slot.
footer-actions	<b>Deprecated</b> Use the "footer" slot instead. A slot for adding calcite-buttons to the component's footer.
footer-end	A slot for adding a trailing footer custom content. Should not be used with the "footer" slot.
footer-start	A slot for adding a leading footer custom content. Should not be used with the "footer" slot.

### Value

an object of class calcite\_component which is a subclass of shiny.tag

### References

[Official Documentation](#)

### Examples

```
calcite_flow_item()
```

---

calcite\_icon *Create a Icon component*

---

### Description

Create a Icon component

### Usage

```
calcite_icon(...)
```

### Arguments

... named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
flipRtl	flip-rtl	When true, the icon will be flipped when the element direction is right-to-left ("rtl").
icon	icon	Displays a specific icon.
preload	preload	When true, it loads preloads the icon data.
scale	scale	Specifies the size of the component.
textLabel	text-label	Accessible name for the component. It is recommended to set this value if your icon is semantic.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_icon()
```

---

```
calcite_inline_editable
```

*Create a InlineEditable component*

---

**Description**

Create a `InlineEditable` component

**Usage**

```
calcite_inline_editable(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
afterConfirm	NA	Specifies a callback to be executed prior to disabling editing via the controls. When pro
controls	controls	When true and editingEnabled is true, displays save and cancel controls on the co
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacit
editingEnabled	editing-enabled	When true, inline editing is enabled on the component.
loading	loading	When true, a busy indicator is displayed.
messageOverrides	NA	Use this property to override individual strings used by the component.
scale	scale	Specifies the size of the component. Defaults to the scale of the wrapped calcite-inp

**Events:**

The following events are observed by shiny:

Event	Description
calciteInlineEditableEditCancel	Emits when the component's "cancel editing" button is pressed.
calciteInlineEditableEditConfirm	Emits when the component's "confirm edits" button is pressed.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding a calcite-input.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_inline_editable()
```

---

 calcite\_input\_date\_picker

*Create a InputDatePicker component*


---

## Description

Create a InputDatePicker component

## Usage

```
calcite_input_date_picker(...)
```

## Arguments

...                    named attributes passed to `htmltools:::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
<code>disabled</code>	<code>disabled</code>	When true, interaction is prevented and the component is displayed in a disabled state.
<code>flipPlacements</code>	NA	Specifies the component's fallback calcite-date-picker placements.
<code>focusTrapDisabled</code>	<code>focus-trap-disabled</code>	When true, prevents focus trapping.
<code>form</code>	<code>form</code>	The id of the form that will be associated with the component.
<code>headingLevel</code>	<code>heading-level</code>	Specifies the heading level of the component's heading for programmatic accessibility.
<code>layout</code>	<code>layout</code>	Defines the layout of the component.
<code>max</code>	<code>max</code>	When the component resides in a form, specifies the latest allowed date as a full date object.
<code>maxAsDate</code>	NA	Specifies the latest allowed date as a full date object.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>min</code>	<code>min</code>	When the component resides in a form, specifies the earliest allowed date as a full date object.
<code>minAsDate</code>	NA	Specifies the earliest allowed date as a full date object.
<code>monthStyle</code>	<code>month-style</code>	Specifies the monthStyle used by the component.
<code>name</code>	<code>name</code>	Specifies the name of the component. Required to pass the component to the form.
<code>numberingSystem</code>	<code>numbering-system</code>	Specifies the Unicode numeral system used by the component for displaying numbers.
<code>open</code>	<code>open</code>	When true, displays the calcite-date-picker component.
<code>overlayPositioning</code>	<code>overlay-positioning</code>	Determines the type of positioning to use for the overlaid content.
<code>placement</code>	<code>placement</code>	Specifies the placement of the calcite-date-picker relative to the component.
<code>proximitySelectionDisabled</code>	<code>proximity-selection-disabled</code>	When true, disables the default behavior on the third click of the component.
<code>range</code>	<code>range</code>	When true, activates a range for the component.
<code>readOnly</code>	<code>read-only</code>	When true, the component's value can be read, but controls are disabled.
<code>required</code>	<code>required</code>	When true and the component resides in a form, the component is required.
<code>scale</code>	<code>scale</code>	Specifies the size of the component.
<code>status</code>	<code>status</code>	Specifies the status of the input field, which determines message placement.
<code>validationIcon</code>	<code>validation-icon</code>	Specifies the validation icon to display under the component.
<code>validationMessage</code>	<code>validation-message</code>	Specifies the validation message to display under the component.

validity	NA	The current validation state of the component.
value	value	Selected date as a string in ISO format ("yyyy-mm-dd").
valueAsDate	NA	The component's value as a full date object.

**Events:**

The following events are observed by shiny:

Event	Description
calciteInputDatePickerBeforeClose	Fires when the component is requested to be closed and before the closing transition begins.
calciteInputDatePickerBeforeOpen	Fires when the component is added to the DOM but not rendered, and before the opening transition begins.
calciteInputDatePickerChange	Fires when the component's value changes.
calciteInputDatePickerClose	Fires when the component is closed and animation is complete.
calciteInputDatePickerOpen	Fires when the component is open and animation is complete.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_input_date_picker()
```

---

`calcite_input_file`     *Create a Calcite File Input Component*

---

**Description**

Creates a file input component for selecting files from the user's device. Use the `accept` argument to restrict which file types can be selected. This component works by syncing with Shiny's native file upload mechanism to handle file transfers to the server.

**Usage**

```
calcite_input_file(
  id = NULL,
  accept = NULL,
  multiple = FALSE,
  disabled = NULL,
  label = NULL,
  label_text = NULL,
  required = NULL,
```

```

    scale = NULL,
    validation_icon = NULL,
    validation_message = NULL
  )

```

### Arguments

<code>id</code>	Component ID (required for Shiny reactivity)
<code>accept</code>	A character vector of accepted file extensions without the leading period (e.g. <code>c("csv", "tsv", "txt")</code> ). MIME types are also accepted (e.g. <code>"text/plain"</code> ). When <code>NULL</code> , all file types are accepted.
<code>multiple</code>	When <code>TRUE</code> , the user can select more than one file. Must be a scalar logical. Default <code>FALSE</code> .
<code>disabled</code>	When <code>TRUE</code> , prevents interaction and decreases opacity.
<code>label</code>	Accessible label for the component.
<code>label_text</code>	Label text displayed on the component.
<code>required</code>	When <code>TRUE</code> , a file must be selected for form submission.
<code>scale</code>	Size of the component: <code>"s"</code> , <code>"m"</code> , or <code>"l"</code> .
<code>validation_icon</code>	Validation icon to display ( <code>TRUE</code> or an icon name).
<code>validation_message</code>	Validation message to display under the component.

### Details

#### Shiny Integration:

When used in a Shiny app, `calcite_input_file()` works like `shiny::fileInput()` and returns a data frame via `input$id` with one row per uploaded file.

#### Columns in `input$id`:

- `name` - The filename provided by the browser
- `size` - File size in bytes
- `type` - MIME type (e.g., `"text/csv"`)
- `datapath` - Path to a temp file containing the uploaded data. Use this path to read the file contents (e.g., `read.csv(input$my_file$datapath[1])`)

The uploaded files are stored in a temporary directory and will be deleted when the Shiny session ends.

### Value

An object of class `calcite_component`

### References

[Official Documentation](#)

## Examples

```
# Accept CSV files only
calcite_input_file(
  id = "my_file",
  accept = "csv",
  label_text = "Upload a CSV file"
)

# Accept multiple image formats
calcite_input_file(
  id = "my_file",
  accept = c("png", "jpg", "jpeg", "gif"),
  multiple = TRUE,
  label_text = "Upload images"
)

if (interactive()) {
  library(shiny)

  ui <- calcite_shell(
    calcite_panel(
      heading = "File Upload",
      calcite_input_file(
        id = "csv_upload",
        accept = "csv",
        label_text = "Upload CSV"
      ),
      tableOutput("contents")
    )
  )

  server <- function(input, output, session) {
    output$contents <- renderTable({
      req(input$csv_upload)
      read.csv(input$csv_upload$datapath[1])
    })
  }

  shinyApp(ui, server)
}
```

---

calcite\_input\_message *Create a Calcite Input Message Component*

---

## Description

Creates a message component that displays validation messages, hints, and other contextual feedback for input components.

**Usage**

```
calcite_input_message(
  ...,
  status = NULL,
  icon = NULL,
  icon_flip_rtl = NULL,
  scale = NULL,
  id = NULL
)
```

**Arguments**

...	Content for the message (default slot) - typically text or textOutput()
status	Status of the message: "idle", "valid", or "invalid" (default: "idle")
icon	Specifies an icon to display (TRUE for default or icon name)
icon_flip_rtl	When TRUE, icon is flipped in RTL direction (default: FALSE)
scale	Size of the component: "s", "m", or "l" (default: "m")
id	Component ID (optional)

**Details****Usage:**

Input messages are typically used inside a `calcite_label()` component alongside an input to provide contextual feedback, validation messages, or hints.

**Status Options:**

- **"valid"**: Green checkmark icon, success message
- **"invalid"**: Red warning icon, error message
- **"idle"**: Default state, informational message

**Dynamic Messages:**

Use `textOutput()` in the slot and `renderText()` in the server to create dynamic validation messages. Use `update_calcite()` to change the status/icon properties.

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

## Examples

```
# Static message
calcite_input_message(
  "Username is available",
  status = "valid",
  icon = TRUE
)

# Dynamic message with textOutput
calcite_input_message(
  shiny::textOutput("validation_msg"),
  status = "invalid",
  icon = "exclamation-mark-circle",
  id = "msg"
)

# With input in a label
calcite_label(
  label = "Username",
  calcite_input_text(
    id = "username",
    placeholder = "Enter username"
  ),
  calcite_input_message(
    "Username must be 3-20 characters",
    status = "idle"
  )
)
```

---

calcite\_input\_number *Create a Calcite Input Number Component*

---

## Description

Creates a number input component for form entry where users can enter numeric values. Supports integer-only mode, min/max constraints, and number formatting.

## Usage

```
calcite_input_number(
  id = NULL,
  value = NULL,
  name = NULL,
  placeholder = NULL,
  label = NULL,
  label_text = NULL,
  alignment = NULL,
  autocomplete = NULL,
  clearable = NULL,
```

```

disabled = NULL,
form = NULL,
group_separator = NULL,
icon = NULL,
icon_flip rtl = NULL,
integer = NULL,
loading = NULL,
max = NULL,
min = NULL,
number_button_type = NULL,
numbering_system = NULL,
prefix_text = NULL,
read_only = NULL,
required = NULL,
scale = NULL,
status = NULL,
step = NULL,
suffix_text = NULL,
validation_icon = NULL,
validation_message = NULL,
action = NULL,
label_content = NULL
)

```

### Arguments

id	Component ID (required for Shiny reactivity)
value	The component's value
name	Specifies the name of the component (required for form submission)
placeholder	Specifies placeholder text for the component
label	Accessible name for the component
label_text	When provided, displays label text on the component
alignment	Text alignment of the value: "start", "center", or "end" (default: "start")
autocomplete	Type of content to autocomplete for forms
clearable	When TRUE, shows a clear button when the component has a value (default: FALSE)
disabled	When TRUE, interaction is prevented and displayed with lower opacity (default: FALSE)
form	The id of the form associated with the component
group_separator	When TRUE, number values display with a group separator corresponding to the language and country format (default: FALSE)
icon	Specifies an icon to display (TRUE for default or icon name)
icon_flip rtl	When TRUE, icon is flipped in RTL direction (default: FALSE)

integer	When TRUE, restricts the component to integer numbers only and disables exponential notation (default: FALSE)
loading	When TRUE, displays a busy indicator (default: FALSE)
max	Maximum value for the input
min	Minimum value for the input
number_button_type	Placement of the increment/decrement buttons: "vertical", "horizontal", or "none" (default: "vertical")
numbering_system	Unicode numeral system: "arab", "arabext", or "latn"
prefix_text	Text to display at the start of the component
read_only	When TRUE, value can be read but not modified (default: FALSE)
required	When TRUE, must have a value for form submission (default: FALSE)
scale	Size of the component: "s", "m", or "l" (default: "m")
status	Status of the input field: "idle", "valid", or "invalid" (default: "idle")
step	Granularity the value must adhere to (can be "any" or a number)
suffix_text	Text to display at the end of the component
validation_icon	Validation icon to display (TRUE or icon name)
validation_message	Validation message to display
action	Content for the action slot (typically a calcite_action)
label_content	Content for the label-content slot

## Details

### Shiny Integration:

When used in a Shiny app, `calcite_input_number()` returns a reactive list containing:

### Available properties:

- `$value` - The current numeric value (as string)
- `$validity` - Object containing validation state (valid, badInput, rangeOverflow, etc.)
- `$status` - Current status ("idle", "valid", "invalid")
- Other component properties

### Usage in server:

```
# Watch for value changes
observeEvent(input$age$value, {
  age_num <- as.numeric(input$age$value)
  print(paste("Age:", age_num))
})
```

```
# Check validity
observeEvent(input$age, {
```

```
    if (input$age$validity$valid) {
      print("Valid number entered")
    }
    if (input$age$validity$rangeOverflow) {
      print("Number is too large")
    }
  })
})
```

### Value

An object of class `calcite_component`

### References

[Official Documentation](#)

### Examples

```
# Basic number input
calcite_input_number(
  id = "quantity",
  placeholder = "Enter quantity"
)

# Integer-only input with constraints
calcite_input_number(
  id = "age",
  integer = TRUE,
  min = 0,
  max = 120,
  step = 1,
  required = TRUE
)

# Price input with formatting
calcite_input_number(
  id = "price",
  prefix_text = "$",
  group_separator = TRUE,
  step = 0.01,
  min = 0
)

# Percentage input
calcite_input_number(
  id = "percent",
  suffix_text = "%",
  min = 0,
  max = 100,
  step = 1
)
```

---

calcite\_input\_text      *Create a Calcite Input Text Component*

---

### Description

Creates a text input component for form entry where users can enter text values. Supports validation, length constraints, and pattern matching.

### Usage

```
calcite_input_text(  
  id = NULL,  
  value = NULL,  
  name = NULL,  
  placeholder = NULL,  
  label = NULL,  
  label_text = NULL,  
  alignment = NULL,  
  autocomplete = NULL,  
  clearable = NULL,  
  disabled = NULL,  
  form = NULL,  
  icon = NULL,  
  icon_flip rtl = NULL,  
  loading = NULL,  
  max_length = NULL,  
  min_length = NULL,  
  pattern = NULL,  
  prefix_text = NULL,  
  read_only = NULL,  
  required = NULL,  
  scale = NULL,  
  status = NULL,  
  suffix_text = NULL,  
  validation_icon = NULL,  
  validation_message = NULL,  
  action = NULL,  
  label_content = NULL  
)
```

### Arguments

id	Component ID (required for Shiny reactivity)
value	The component's value
name	Specifies the name of the component (required for form submission)
placeholder	Specifies placeholder text for the component

label	Accessible name for the component
label_text	When provided, displays label text on the component
alignment	Text alignment of the value: "start", "center", or "end" (default: "start")
autocomplete	Type of content to autocomplete for forms
clearable	When TRUE, shows a clear button when the component has a value (default: FALSE)
disabled	When TRUE, interaction is prevented and displayed with lower opacity (default: FALSE)
form	The id of the form associated with the component
icon	Specifies an icon to display (TRUE for default or icon name)
icon_flip rtl	When TRUE, icon is flipped in RTL direction (default: FALSE)
loading	When TRUE, displays a busy indicator (default: FALSE)
max_length	Maximum length of text for the component's value
min_length	Minimum length of text for the component's value
pattern	Regular expression pattern the value must match for validation
prefix_text	Text to display at the start of the component
read_only	When TRUE, value can be read but not modified (default: FALSE)
required	When TRUE, must have a value for form submission (default: FALSE)
scale	Size of the component: "s", "m", or "l" (default: "m")
status	Status of the input field: "idle", "valid", or "invalid" (default: "idle")
suffix_text	Text to display at the end of the component
validation_icon	Validation icon to display (TRUE or icon name)
validation_message	Validation message to display
action	Content for the action slot (typically a calcite_action)
label_content	Content for the label-content slot

## Details

### Shiny Integration:

When used in a Shiny app, `calcite_input_text()` returns a reactive list containing:

### Available properties:

- `$value` - The current text value
- `$validity` - Object containing validation state (valid, tooLong, tooShort, patternMismatch, etc.)
- `$status` - Current status ("idle", "valid", "invalid")
- Other component properties

### Usage in server:

```
# Watch for value changes
observeEvent(input$username$value, {
  print(paste("Username:", input$username$value))
})

# Check validity
observeEvent(input$username, {
  if (input$username$validity$valid) {
    print("Valid username entered")
  }
  if (input$username$validity$tooShort) {
    print("Username is too short")
  }
})
```

## Value

An object of class `calcite_component`

## References

[Official Documentation](#)

## Examples

```
# Basic text input
calcite_input_text(
  id = "username",
  placeholder = "Enter username"
)

# Text input with length constraints
calcite_input_text(
  id = "username",
  min_length = 3,
  max_length = 20,
  required = TRUE
)

# Text input with pattern validation
calcite_input_text(
  id = "zip_code",
  pattern = "\\d{5}",
  placeholder = "12345"
)

# Search input with icon
calcite_input_text(
  id = "search",
  icon = "search",
  clearable = TRUE,
```

```
    placeholder = "Search..."
  )
```

---

```
calcite_input_time_picker
```

*Create a InputTimePicker component*

---

## Description

Create a InputTimePicker component

## Usage

```
calcite_input_time_picker(...)
```

## Arguments

...                    named attributes passed to `htmltools:::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
<code>disabled</code>	<code>disabled</code>	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.
<code>focusTrapDisabled</code>	<code>focus-trap-disabled</code>	When <code>true</code> , prevents focus trapping.
<code>form</code>	<code>form</code>	The <code>id</code> of the form that will be associated with the component. When not set, the component will be associated with the form containing the component.
<code>hourFormat</code>	<code>hour-format</code>	Specifies the component's hour format, where: "user" displays the user's locale format, "12" displays 12-hour format, and "24" displays 24-hour format.
<code>max</code>	<code>max</code>	When the component resides in a form, specifies the maximum value.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>min</code>	<code>min</code>	When the component resides in a form, specifies the minimum value.
<code>name</code>	<code>name</code>	Specifies the name of the component on form submission.
<code>numberingSystem</code>	<code>numbering-system</code>	Specifies the Unicode numeral system used by the component for localization.
<code>open</code>	<code>open</code>	When <code>true</code> , displays the <code>calcite-time-picker</code> component.
<code>overlayPositioning</code>	<code>overlay-positioning</code>	Determines the type of positioning to use for the overlaid content. Using "absolute" will allow the component to be positioned anywhere on the page. Using "popper" will use the Popper.js library to position the component.
<code>placement</code>	<code>placement</code>	Determines where the popover will be positioned relative to the input.
<code>readOnly</code>	<code>read-only</code>	When <code>true</code> , the component's value can be read, but controls are not accessible and the component is displayed with lower opacity.
<code>required</code>	<code>required</code>	When <code>true</code> and the component resides in a form, the component must have a value when the form is submitted.
<code>scale</code>	<code>scale</code>	Specifies the size of the component.
<code>status</code>	<code>status</code>	Specifies the status of the input field, which determines message and icons.
<code>step</code>	<code>step</code>	Specifies the granularity the component's value must adhere to (in seconds).
<code>validationIcon</code>	<code>validation-icon</code>	Specifies the validation icon to display under the component.
<code>validationMessage</code>	<code>validation-message</code>	Specifies the validation message to display under the component.
<code>validity</code>	NA	The current validation state of the component.
<code>value</code>	<code>value</code>	The time value in ISO (24-hour) format.

**Events:**

The following events are observed by shiny:

Event	Description
calciteInputTimePickerBeforeClose	Fires when the component is requested to be closed and before the closing transition begins.
calciteInputTimePickerBeforeOpen	Fires when the component is added to the DOM but not rendered, and before the opening transition begins.
calciteInputTimePickerChange	Fires when the component's value is changed.
calciteInputTimePickerClose	Fires when the component is closed and animation is complete.
calciteInputTimePickerOpen	Fires when the component is open and animation is complete.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_input_time_picker()
```

---

```
calcite_input_time_zone
```

*Create a InputTimeZone component*

---

**Description**

Create a InputTimeZone component

**Usage**

```
calcite_input_time_zone(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
clearable	clearable	When true, an empty value (null) will be allowed as a value. When false, an empty value is not allowed.
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.

form	form	The id of the form that will be associated with the component. When not set, the
maxItems	max-items	Specifies the component's maximum number of options to display before displaying
messageOverrides	NA	Use this property to override individual strings used by the component.
mode	mode	This specifies the type of value and the associated options presented to the user:
name	name	Specifies the name of the component. Required to pass the component's value on
offsetStyle	offset-style	Specifies how the offset will be displayed, where "user" uses UTC or GMT dependi
open	open	When true, displays and positions the component.
overlayPositioning	overlay-positioning	Determines the type of positioning to use for the overlaid content. Using "absolu
readOnly	read-only	When true, the component's value can be read, but controls are not accessible and
referenceDate	reference-date	This date will be used as a reference to Daylight Savings Time when creating tim
scale	scale	Specifies the size of the component.
status	status	Specifies the status of the input field, which determines message and icons.
validationIcon	validation-icon	Specifies the validation icon to display under the component.
validationMessage	validation-message	Specifies the validation message to display under the component.
validity	NA	The current validation state of the component.
value	value	The component's value, where the value is the time zone offset or the difference, i

**Events:**

The following events are observed by shiny:

Event	Description
calciteInputTimeZoneBeforeClose	Fires when the component is requested to be closed and before the closing transition beg
calciteInputTimeZoneBeforeOpen	Fires when the component is added to the DOM but not rendered, and before the opening
calciteInputTimeZoneChange	Fires when the component's value changes.
calciteInputTimeZoneClose	Fires after the component is closed and animation is complete.
calciteInputTimeZoneOpen	Fires after the component is opened and animation is complete.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_input_time_zone()
```

---

calcite\_label

*Create a Calcite Label Component*

---

**Description**

Creates a label component that wraps and provides accessible text for form controls and other interactive elements.

**Usage**

```
calcite_label(
  label,
  ...,
  target_id = NULL,
  id = NULL,
  alignment = NULL,
  layout = NULL,
  scale = NULL
)
```

**Arguments**

label	The label text (required)
...	Child components to wrap within the label (e.g., calcite_input_text())
target_id	The id of the component this label is bound to (only needed when the labeled component is outside the label). Maps to the for HTML attribute
id	Component ID (optional)
alignment	Specifies the text alignment of the component: "start", "center", or "end" (default: "start")
layout	Defines the layout of the label: "block", "default", "inline", or "inline-space-between" (default: "default"). Note: "default" is deprecated, use "block" instead
scale	Specifies the size of the component: "s" (small), "m" (medium), or "l" (large) (default: "m")

**Value**

An object of class calcite\_component

**References**

[Official Documentation](#)

**Examples**

```
# Label wrapping an input
calcite_label(
  label = "Username",
  calcite_input_text(
    id = "username",
    placeholder = "Enter username"
  )
)

# Label with inline layout
calcite_label(
  label = "Subscribe",
  layout = "inline",
```

```

    calcite_checkbox(id = "subscribe")
  )

  # Label with external component (using target_id)
  htmltools::tagList(
    calcite_label(
      label = "Password",
      target_id = "password"
    ),
    calcite_input_text(id = "password", placeholder = "Enter password")
  )

```

---

 calcite\_link

*Create a Calcite Link Component*


---

### Description

Creates a link component for navigation or performing actions. Links are useful for tertiary-level actions or inline navigation within text.

### Usage

```

calcite_link(
  text,
  href,
  id = NULL,
  disabled = NULL,
  download = NULL,
  icon_end = NULL,
  icon_flip_rtl = NULL,
  icon_start = NULL,
  rel = NULL,
  target = NULL
)

```

### Arguments

text	The link text to display (required)
href	The URL of the linked resource (required) - can be absolute or relative path
id	Component ID (optional)
disabled	When TRUE, interaction is prevented and the component is displayed with lower opacity (default: FALSE)
download	Prompts the user to save the linked URL instead of navigating to it. Can be TRUE or a string filename
icon_end	Specifies an icon to display at the end of the link
icon_flip_rtl	Displays the iconStart and/or iconEnd as flipped when the element direction is RTL: "both", "end", or "start"

icon_start	Specifies an icon to display at the start of the link
rel	Specifies the relationship to the linked document (e.g., "noopener", "noreferrer")
target	Specifies the frame or window to open the linked document (e.g., "_blank", "_self", "_parent", "_top")

## Details

### Usage Guidelines:

#### Use Links for:

- Navigation to another page or section
- Tertiary-level actions in a group
- Inline actions within sentences

#### Use Buttons instead for:

- Primary actions like "Create", "Submit", "Upload"
- Task completion actions
- Prominent calls to action

#### Best Practices:

- Avoid making entire sentences into links
- Link text should be descriptive of the destination
- Use target = "\_blank" with rel = "noopener noreferrer" for external links

## Value

An object of class `calcite_component`

## References

[Official Documentation](#)

## Examples

```
# Basic link
calcite_link(
  text = "View documentation",
  href = "https://developers.arcgis.com/calcite-design-system/"
)

# Link with icon
calcite_link(
  text = "External site",
  href = "https://example.com",
  icon_end = "launch",
  target = "_blank",
  rel = "noopener noreferrer"
)

# Download link
```

```

calcite_link(
  text = "Download report",
  href = "/files/report.pdf",
  download = "monthly-report.pdf",
  icon_start = "download"
)

# Disabled link
calcite_link(
  text = "Coming soon",
  href = "#",
  disabled = TRUE
)

```

---

calcite\_list

*Create a List component*


---

## Description

A general purpose list that enables users to construct list items that conform to Calcite styling.

## Usage

```
calcite_list(...)
```

## Arguments

...                    named attributes passed to `htmltools::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
canPull	NA	When provided, the method will be called to determine whether the element can be pulled.
canPut	NA	When provided, the method will be called to determine whether the element can be put.
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
displayMode	display-mode	Specifies the nesting behavior of <code>calcite-list-items</code> , where "flat" displays items side-by-side.
dragEnabled	drag-enabled	When true, <code>calcite-list-items</code> are sortable via a draggable button.
filteredData	NA	The currently filtered <code>calcite-list-item</code> data.
filteredItems	NA	The currently filtered <code>calcite-list-items</code> .
filterEnabled	filter-enabled	When true, an input appears at the top of the component that can be used by end users to filter items.
filterLabel	filter-label	Specifies an accessible name for the filter input field.
filterPlaceholder	filter-placeholder	Placeholder text for the component's filter input field.
filterPredicate	NA	Specifies a function to handle filtering.
filterProps	NA	Specifies the properties to match against when filtering. If not set, all properties are used.
filterText	filter-text	Text for the component's filter input field.

group	group	The list's group identifier. To drag elements from one list into another, both lists must have the same group identifier.
interactionMode	interaction-mode	Specifies the interaction mode of the component. "interactive" allows interactive dragging, "passive" allows only passive dragging, and "disabled" disables dragging.
label	label	Specifies an accessible name for the component. When dragEnabled is true and the component is focused, this label is used to describe the component to screen readers.
loading	loading	When true, a busy indicator is displayed.
messageOverrides	NA	Use this property to override individual strings used by the component.
numberingSystem	numbering-system	Specifies the Unicode numeral system used by the component for localization.
scale	scale	Specifies the size of the component.
selectedItems	NA	The currently selected items.
selectionAppearance	selection-appearance	Specifies the selection appearance - "icon" (displays a checkmark or dot) or "text" (displays the text of the selected item).
selectionMode	selection-mode	Specifies the selection mode of the component, where: "multiple" allows any number of items to be selected, "single" allows only one item to be selected, and "none" does not allow any items to be selected.

**Events:**

The following events are observed by shiny:

Event	Description
calciteListChange	Fires when the component's selected items have changed.
calciteListDragEnd	Fires when the component's dragging has ended.
calciteListDragStart	Fires when the component's dragging has started.
calciteListFilter	Fires when the component's filter has changed.
calciteListOrderChange	Fires when the component's item order changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-list-item and calcite-list-item-group elements.
filter-actions-start	A slot for adding actionable calcite-action elements before the filter component.
filter-actions-end	A slot for adding actionable calcite-action elements after the filter component.
filter-no-results	When filterEnabled is true, a slot for adding content to display when no results are found.

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_list()
```

---

calcite\_list\_item      *Create a ListItem component*

---

## Description

Create a ListItem component

## Usage

```
calcite_list_item(...)
```

## Arguments

...                  named attributes passed to `htmltools::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
closable	closable	When true, a close button is added to the component.
closed	closed	When true, hides the component.
description	description	A description for the component. Displays below the label text.
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
dragDisabled	drag-disabled	When true, the item is not draggable.
iconEnd	icon-end	Specifies an icon to display at the end of the component.
iconFlipRtl	icon-flip-rtl	Displays the iconStart and/or iconEnd as flipped when the element direction is right-to-left.
iconStart	icon-start	Specifies an icon to display at the start of the component.
label	label	The label text of the component. Displays above the description text.
messageOverrides	NA	Use this property to override individual strings used by the component.
metadata	NA	Provides additional metadata to the component. Primary use is for a filter on the parent component.
open	open	When true, the item is open to show child components.
scale	scale	Specifies the size of the component.
selected	selected	When true and the parent calcite-list's selectionMode is "single", "single-p", or "multiple", the component is selected.
sortHandleOpen	sort-handle-open	When true, displays and positions the sort handle.
unavailable	unavailable	When true, the component's content appears inactive.
value	value	The component's value.

### Events:

The following events are observed by shiny:

Event	Description
calciteListItemClose	Fires when the close button is clicked.
calciteListItemSelect	Fires when the component is selected.
calciteListItemSortHandleBeforeClose	Fires when the sort handle is requested to be closed and before the closing transition.

calciteListItemSortHandleBeforeOpen	Fires when the sort handle is added to the DOM but not rendered, and before the open button is clicked.
calciteListItemSortHandleClose	Fires when the sort handle is closed and animation is complete.
calciteListItemSortHandleOpen	Fires when the sort handle is open and animation is complete.
calciteListItemToggle	Fires when the open button is clicked.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-list, calcite-list-item and calcite-list-item-group elements.
actions-start	A slot for adding actionable calcite-action elements before the content of the component.
content-start	A slot for adding non-actionable elements before the label and description of the component.
content	A slot for adding non-actionable, centered content in place of the label and description of the component.
content-end	A slot for adding non-actionable elements after the label and description of the component.
actions-end	A slot for adding actionable calcite-action elements after the content of the component.
content-bottom	A slot for adding content below the component's label and description.

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_list_item()
```

---

```
calcite_list_item_group
```

*Create a ListItemGroup component*

---

**Description**

Create a ListItemGroup component

**Usage**

```
calcite_list_item_group(...)
```

**Arguments**

...            named attributes passed to htmltools::tag()

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description	Values
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.	boolean
heading	heading	The header text for all nested calcite-list-item rows.	string
scale	scale	Specifies the size of the component.	"l"   "m"   "s"

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-list-item and calcite-list-item-group elements.

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_list_item_group()
```

---

calcite\_loader      *Create a Loader component*

---

**Description**

Create a Loader component

**Usage**

```
calcite_loader(...)
```

**Arguments**

...      named attributes passed to htmltools::tag()

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
inline	inline	When true, displays smaller and appears to the left of the text.
label	label	Accessible name for the component.
scale	scale	Specifies the size of the component.
text	text	Text that displays under the component's indicator.
type	type	Specifies the component type. Use "indeterminate" if finding actual progress value is impossible. Other
value	value	The component's value. Valid only for "determinate" indicators. Percent complete of 100.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_loader()
```

---

calcite_menu	<i>Create a Menu component</i>
--------------	--------------------------------

---

**Description**

Create a Menu component

**Usage**

```
calcite_menu(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description	Values
label	label	Accessible name for the component.	string
layout	layout	Specifies the layout of the component.	"horizontal"   "vert
messageOverrides	NA	Use this property to override individual strings used by the component.	Check API referenc

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_menu()
```

---

calcite\_menu\_item      *Create a MenuItem component*

---

**Description**

Create a MenuItem component

**Usage**

```
calcite_menu_item(...)
```

**Arguments**

...                  named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
active	active	When true, the component is highlighted.
breadcrumb	breadcrumb	When true, the component displays a breadcrumb trail for use as a navigational aid.
href	href	Specifies the URL destination of the component, which can be set as an absolute or relative
iconEnd	icon-end	Specifies an icon to display at the end of the component.

iconFlipRtl	icon-flip-rtl	Displays the iconStart and/or iconEnd as flipped when the element direction is right-to-
iconStart	icon-start	Specifies an icon to display at the start of the component.
label	label	Accessible name for the component.
messageOverrides	NA	Use this property to override individual strings used by the component.
open	open	When true, the component will display any slotted calcite-menu-item in an open over
rel	rel	Defines the relationship between the href value and the current document.
target	target	Specifies where to open the linked document defined in the href property.
text	text	Specifies the text to display.

**Events:**

The following events are observed by shiny:

Event	Description
calciteMenuItemSelect	Emits when the component is selected.

**Slots:**

The following slots are provided by this component:

Slot	Description
submenu-item	A slot for adding calcite-menu-items in a submenu.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_menu_item()
```

---

calcite\_meter      *Create a Meter component*

---

**Description**

Create a Meter component

**Usage**

```
calcite_meter(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>appearance</code>	<code>appearance</code>	Specifies the appearance style of the component.
<code>disabled</code>	<code>disabled</code>	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.
<code>fillType</code>	<code>fill-type</code>	Specifies the component's display, where "single" displays a single color and "range" displays a range of colors.
<code>form</code>	<code>form</code>	The id of the form that will be associated with the component. When not set, the component is associated with the form of the parent component.
<code>groupSeparator</code>	<code>group-separator</code>	When <code>true</code> , number values are displayed with a group separator corresponding to the locale.
<code>high</code>	<code>high</code>	Specifies a high value. When <code>fillType</code> is "range", displays a different color when the value is high.
<code>label</code>	<code>label</code>	Accessible name for the component.
<code>low</code>	<code>low</code>	Specifies a low value. When <code>fillType</code> is "range", displays a different color when the value is low.
<code>max</code>	<code>max</code>	Specifies the highest allowed value of the component.
<code>min</code>	<code>min</code>	Specifies the lowest allowed value of the component.
<code>name</code>	<code>name</code>	Specifies the name of the component. Required to pass the component's value on form submission.
<code>numberingSystem</code>	<code>numbering-system</code>	Specifies the Unicode numeral system used by the component for localization.
<code>rangeLabels</code>	<code>range-labels</code>	When <code>true</code> , displays the values of high, low, min, and max.
<code>rangeLabelType</code>	<code>range-label-type</code>	When <code>rangeLabels</code> is <code>true</code> , specifies the format of displayed labels.
<code>scale</code>	<code>scale</code>	Specifies the size of the component.
<code>unitLabel</code>	<code>unit-label</code>	When <code>rangeLabelType</code> is "units" and either <code>valueLabel</code> or <code>rangeLabels</code> are <code>true</code> , displays the unit label.
<code>value</code>	<code>value</code>	Specifies the current value of the component.
<code>valueLabel</code>	<code>value-label</code>	When <code>true</code> , displays the current value.
<code>valueLabelType</code>	<code>value-label-type</code>	When <code>valueLabel</code> is <code>true</code> , specifies the format of displayed label.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_meter()
```

---

calcite\_modal      *Create a Modal component*

---

### Description

Use the calcite-dialog component instead.

### Usage

```
calcite_modal(...)
```

### Arguments

...      named attributes passed to `htmltools::tag()`

### Details

#### Properties:

The following properties are provided by this component:

Name	Attribute	Description
<code>beforeClose</code>	NA	Passes a function to run before the component closes.
<code>closeButtonDisabled</code>	<code>close-button-disabled</code>	When <code>true</code> , disables the component's close button.
<code>docked</code>	<code>docked</code>	When <code>true</code> , prevents the component from expanding to the entire screen on
<code>escapeDisabled</code>	<code>escape-disabled</code>	When <code>true</code> , disables the default close on escape behavior.
<code>focusTrapDisabled</code>	<code>focus-trap-disabled</code>	When <code>true</code> , prevents focus trapping.
<code>fullscreen</code>	<code>fullscreen</code>	Sets the component to always be fullscreen. Overrides <code>widthScale</code> and <code>-ca</code>
<code>kind</code>	<code>kind</code>	Specifies the kind of the component, which will apply to top border.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>open</code>	<code>open</code>	When <code>true</code> , displays and positions the component.
<code>outsideCloseDisabled</code>	<code>outside-close-disabled</code>	When <code>true</code> , disables the closing of the component when clicked outside.
<code>scale</code>	<code>scale</code>	Specifies the size of the component.
<code>widthScale</code>	<code>width-scale</code>	Specifies the width of the component.

#### Events:

The following events are observed by shiny:

Event	Description
<code>calciteModalBeforeClose</code>	Fires when the component is requested to be closed and before the closing transition begins.
<code>calciteModalBeforeOpen</code>	Fires when the component is added to the DOM but not rendered, and before the opening transition
<code>calciteModalClose</code>	Fires when the component is closed and animation is complete.
<code>calciteModalOpen</code>	Fires when the component is open and animation is complete.

**Slots:**

The following slots are provided by this component:

Slot	Description
header	A slot for adding header text.
content	A slot for adding the component's content.
content-top	A slot for adding content to the component's sticky header, where content remains at the top of the component.
content-bottom	A slot for adding content to the component's sticky footer, where content remains at the bottom of the component.
primary	A slot for adding a primary button.
secondary	A slot for adding a secondary button.
back	A slot for adding a back button.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_modal()
```

---

`calcite_navigation`     *Create a Navigation component*

---

**Description**

Create a Navigation component

**Usage**

```
calcite_navigation(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
label	label	When <code>navigationAction</code> is true, specifies the label of the calcite-action.
navigationAction	navigation-action	When true, displays a calcite-action and emits a <code>calciteNavActionSelect</code> event.

**Events:**

The following events are observed by shiny:

Event	Description
calciteNavigationActionSelect	When navigationAction is true, emits when the displayed action selection changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
logo	A slot for adding a calcite-logo component to the primary navigation level.
user	A slot for adding a calcite-user component to the primary navigation level.
progress	A slot for adding a calcite-progress component to the primary navigation level.
navigation-action	A slot for adding a calcite-action component to the primary navigation level.
content-start	A slot for adding a calcite-menu, calcite-action, or other interactive elements in the start position.
content-center	A slot for adding a calcite-menu, calcite-action, or other interactive elements in the center position.
content-end	A slot for adding a calcite-menu, calcite-action, or other interactive elements in the end position.
navigation-secondary	A slot for adding a calcite-navigation component in the secondary navigation level. Components rendered here are visible only to users with the secondary navigation level.
navigation-tertiary	A slot for adding a calcite-navigation component in the tertiary navigation level. Components rendered here are visible only to users with the tertiary navigation level.

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_navigation()
```

```
calcite_navigation_logo
```

*Create a NavigationLogo component*

**Description**

Create a NavigationLogo component

**Usage**

```
calcite_navigation_logo(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>active</code>	<code>active</code>	When <code>true</code> , the component is highlighted.
<code>description</code>	<code>description</code>	A description for the component, which displays below the heading.
<code>heading</code>	<code>heading</code>	Specifies heading text for the component, such as a product or organization name.
<code>headingLevel</code>	<code>heading-level</code>	Specifies the heading level of the component's heading for proper document structure, without
<code>href</code>	<code>href</code>	Specifies the URL destination of the component, which can be set as an absolute or relative p
<code>icon</code>	<code>icon</code>	Specifies an icon to display.
<code>iconFlipRtl</code>	<code>icon-flip-rtl</code>	When <code>true</code> , the icon will be flipped when the element direction is right-to-left (" <code>rtl</code> ").
<code>label</code>	<code>label</code>	Describes the appearance or function of the thumbnail. If no label is provided, context will r
<code>rel</code>	<code>rel</code>	Defines the relationship between the <code>href</code> value and the current document.
<code>target</code>	<code>target</code>	Specifies where to open the linked document defined in the <code>href</code> property.
<code>thumbnail</code>	<code>thumbnail</code>	Specifies the <code>src</code> to an image.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_navigation_logo()
```

---

```
calcite_navigation_user
```

*Create a NavigationUser component*

---

**Description**

Create a NavigationUser component

**Usage**

```
calcite_navigation_user(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>active</code>	<code>active</code>	When true, the component is highlighted.
<code>fullName</code>	<code>full-name</code>	Specifies the full name of the user.
<code>label</code>	<code>label</code>	Describes the appearance of the avatar. If no label is provided, context will not be provided to a
<code>textDisabled</code>	<code>text-disabled</code>	When true, hides the <code>fullName</code> and <code>username</code> contents.
<code>thumbnail</code>	<code>thumbnail</code>	Specifies the <code>src</code> to an image (remember to add a token if the user is private).
<code>userId</code>	<code>user-id</code>	Specifies the unique id of the user.
<code>username</code>	<code>username</code>	Specifies the username of the user.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_navigation_user()
```

---

calcite\_notice

*Create a Calcite Notice Component*

---

**Description**

Creates a notice component for displaying informative, contextually relevant messages. Best for persistent information that can be open at page load or displayed as a result of user action.

**Usage**

```
calcite_notice(
  ...,
  title = NULL,
  message = NULL,
  link = NULL,
  actions_end = NULL,
  id = NULL,
```

```

    open = NULL,
    closable = NULL,
    icon = NULL,
    icon_flip_rtl = NULL,
    kind = NULL,
    scale = NULL,
    width = NULL
  )

```

## Arguments

...	Additional content passed to the component
title	Content for the title slot
message	Content for the message slot
link	Content for the link slot (e.g. a <code>calcite_link()</code> )
actions_end	Content for the actions-end slot
id	Component ID (required for Shiny reactivity)
open	Whether the notice is visible (default: FALSE)
closable	Whether to show a close button (default: FALSE)
icon	Show default icon (TRUE) or specific icon name (string)
icon_flip_rtl	Flip icon in RTL languages (default: FALSE)
kind	Type of notice: "brand", "danger", "info", "success", or "warning"
scale	Size of the component: "s" (small), "m" (medium), or "l" (large)
width	Width behavior: "auto" or "full" (note: "half" is deprecated)

## Details

### Shiny Integration:

The notice emits events when opened or closed, making it easy to track state and respond to user dismissals.

### Available properties in `input$id`:

- `$open` - Whether the notice is currently visible
- `$closable` - Whether the notice can be closed
- `$kind` - The type of notice
- Other component properties

### Basic usage:

```

calcite_notice(
  id = "my_notice",
  open = TRUE,
  closable = TRUE,
  kind = "success",
  icon = TRUE,
  title = "Success!",

```

```
    message = "Your changes have been saved."
  )

# In server - detect when user closes the notice
observeEvent(input$my_notice$open, {
  if (!input$my_notice$open) {
    message("User dismissed the notice")
  }
})
```

#### Show/hide from server:

```
# Show a notice
update_calcite("my_notice", open = TRUE)

# Hide a notice
update_calcite("my_notice", open = FALSE)
```

#### Slots:

The notice supports several slots for organizing content:

- **title:** The notice title
- **message:** The notice message
- **link:** A calcite-action for links like "Read more"
- **actions-end:** Additional actions (recommended: 2 or fewer)

#### Best Practices:

- Use for informative, contextually relevant information
- Can be open at page load or shown based on user action
- Can be persistent or closable
- Use appropriate kind to convey message severity

#### Value

An object of class `calcite_component`

#### References

[Official Documentation](#)

#### Examples

```
# Basic notice
calcite_notice(
  open = TRUE,
  icon = TRUE,
  closable = TRUE,
  title = "New feature available",
  message = "Check out the reporting dashboard"
)
```

```

# Notice with specific icon and kind
calcite_notice(
  open = TRUE,
  kind = "danger",
  icon = "exclamation-mark-triangle",
  title = "Error in form",
  message = "Please correct the highlighted fields"
)

# Notice with action link
calcite_notice(
  open = TRUE,
  icon = "layers-reference",
  title = "Try this trick",
  message = "Select multiple layers at once",
  link = calcite_link(text = "Read more", href = "#")
)

# Shiny example with server control
if (interactive()) {
  library(shiny)

  ui <- calcite_shell(
    calcite_panel(
      heading = "Notice Demo",

      calcite_notice(
        id = "my_notice",
        open = FALSE,
        closable = TRUE,
        kind = "success",
        icon = TRUE,
        title = "Success!",
        message = "Your action completed successfully"
      ),

      calcite_button(
        id = "show_notice",
        "Show Notice"
      ),

      verbatimTextOutput("notice_status")
    )
  )

  server <- function(input, output, session) {
    observeEvent(input$show_notice$clicks, {
      update_calcite("my_notice", open = TRUE)
    })

    output$notice_status <- renderPrint({
      input$my_notice
    })
  }
}

```

```
    })  
  }  
  
  shinyApp(ui, server)  
}
```

---

**calcite\_option***Create a Calcite Option Component*

---

### Description

Option provides a selectable item within a `calcite_select()` component. Each option represents a single choice in the select dropdown menu.

### Usage

```
calcite_option(label, value, disabled = NULL, selected = NULL, ...)
```

### Arguments

label	Text label for the option (required for accessibility)
value	The component's value (can be any type)
disabled	When true, interaction is prevented and the component is displayed with lower opacity
selected	When true, the component is selected
...	Additional attributes passed to the component

### Details

Options are used within `calcite_select()` to define the available choices. The value parameter determines what value will be sent to Shiny when the option is selected.

### Value

An object of class `calcite_component`

### References

[Official Documentation](#)

## Examples

```
# Basic option
calcite_option(
  label = "First Option",
  value = "1"
)

# Selected option
calcite_option(
  label = "Default Selection",
  value = "default",
  selected = TRUE
)

# Disabled option
calcite_option(
  label = "Unavailable",
  value = "unavailable",
  disabled = TRUE
)

# Use with calcite_select
calcite_select(
  id = "my_select",
  label = "Choose one",
  calcite_option(label = "Option 1", value = "1"),
  calcite_option(label = "Option 2", value = "2", selected = TRUE),
  calcite_option(label = "Option 3", value = "3")
)
```

---

calcite\_option\_group *Create a OptionGroup component*

---

## Description

Create a OptionGroup component

## Usage

```
calcite_option_group(...)
```

## Arguments

...                    named attributes passed to `htmltools::tag()`

**Details**

**Properties:**

The following properties are provided by this component:

Name	Attribute	Description	Values
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.	boolean
label	label	Accessible name for the component.	string

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-options.

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_option_group()
```

---

calcite_pagination	<i>Create a Pagination component</i>
--------------------	--------------------------------------

---

**Description**

Create a Pagination component

**Usage**

```
calcite_pagination(...)
```

**Arguments**

...            named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
groupSeparator	group-separator	When true, number values are displayed with a group separator corresponding to the locale.
messageOverrides	NA	Use this property to override individual strings used by the component.
numberingSystem	numbering-system	Specifies the Unicode numeral system used by the component for localization.
pageSize	page-size	Specifies the number of items per page.
scale	scale	Specifies the size of the component.
startItem	start-item	Specifies the starting item number.
totalItems	total-items	Specifies the total number of items.

**Events:**

The following events are observed by shiny:

Event	Description
calcitePaginationChange	Emits when the selected page changes.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_pagination()
```

---

calcite\_panel

*Create a Calcite Panel Component*

---

**Description**

Panel is a container that can house interactions as well as live within other Calcite Components. It provides a header with heading text and a summary, and slots to add Actions and Icons to the start and end of the header.

**Usage**

```

calcite_panel(
  ...,
  id = NULL,
  heading = NULL,
  description = NULL,
  closable = NULL,
  closed = NULL,
  collapsed = NULL,
  collapse_direction = NULL,
  collapsible = NULL,
  disabled = NULL,
  heading_level = NULL,
  icon = NULL,
  icon_flip rtl = NULL,
  loading = NULL,
  menu_flip_placements = NULL,
  menu_open = NULL,
  menu_placement = NULL,
  message_overrides = NULL,
  overlay_positioning = NULL,
  scale = NULL,
  action_bar = NULL,
  alerts = NULL,
  content_bottom = NULL,
  content_top = NULL,
  header_actions_start = NULL,
  header_actions_end = NULL,
  header_content = NULL,
  header_menu_actions = NULL,
  fab = NULL,
  footer = NULL,
  footer_start = NULL,
  footer_end = NULL
)

```

**Arguments**

...	Main content for the panel (default slot)
id	Component ID (required for Shiny reactivity)
heading	Header text for the panel
description	Description text displayed below the heading
closable	Whether to display a close button in the header (default: FALSE)
closed	Whether the component is hidden (default: FALSE)
collapsed	Whether the content area is hidden (default: FALSE)
collapse_direction	Direction of collapse icon: "down" or "up" (default: "down")

<code>collapsible</code>	Whether the panel can be collapsed (default: FALSE)
<code>disabled</code>	Whether interaction is prevented (default: FALSE)
<code>heading_level</code>	Semantic heading level (1-6) for accessibility
<code>icon</code>	Icon to display in the header
<code>icon_flip rtl</code>	Flip icon in RTL languages (default: FALSE)
<code>loading</code>	Whether to display a busy indicator (default: FALSE)
<code>menu_flip_placements</code>	Fallback placements for the menu when space is insufficient
<code>menu_open</code>	Whether the action menu items are open (default: FALSE)
<code>menu_placement</code>	Placement of the action menu (default: "bottom-end")
<code>message_overrides</code>	Override individual strings used by the component
<code>overlay_positioning</code>	Positioning type for overlaid content: "absolute" or "fixed" (default: "absolute")
<code>scale</code>	Size of the component: "s" (small), "m" (medium), or "l" (large) (default: "m")
<code>action_bar</code>	Content for the action-bar slot (typically <code>calcite_action_bar()</code> )
<code>alerts</code>	Content for the alerts slot (typically <code>calcite_alert()</code> components)
<code>content_bottom</code>	Content below the main slot and above the footer
<code>content_top</code>	Content above the main slot and below the action-bar
<code>header_actions_start</code>	Actions or content at the start of the header
<code>header_actions_end</code>	Actions or content at the end of the header
<code>header_content</code>	Custom content for the header
<code>header_menu_actions</code>	Overflow menu with actions (typically in <code>calcite_dropdown()</code> )
<code>fab</code>	Floating action button (typically <code>calcite_fab()</code> )
<code>footer</code>	Custom content for the footer (don't use with <code>footer_start/footer_end</code> )
<code>footer_start</code>	Leading footer content (don't use with footer slot)
<code>footer_end</code>	Trailing footer content (don't use with footer slot)

## Details

### Shiny Integration:

The panel emits events when it's closed, collapsed, expanded, scrolled, or toggled.

### Available properties in `input$id`:

- `$closable` - Whether the close button is displayed
- `$closed` - Whether the panel is hidden
- `$collapsed` - Whether the content area is collapsed
- `$collapsible` - Whether the panel is collapsible
- `$disabled` - Whether the panel is disabled

- Other component properties

**Events:**

- calcitePanelClose - Fires when the close button is clicked
- calcitePanelCollapse - Fires when the content area is collapsed
- calcitePanelExpand - Fires when the content area is expanded
- calcitePanelScroll - Fires when the content is scrolled
- calcitePanelToggle - Fires when the collapse button is clicked

**Value**

An object of class calcite\_component

**References**

[Official Documentation](#)

**Examples**

```
# Basic panel with heading
calcite_panel(
  id = "my_panel",
  heading = "Map Options",
  "Panel content goes here..."
)

# Collapsible panel with icon
calcite_panel(
  heading = "Layer effects",
  description = "Adjust blur, highlight, and more",
  collapsible = TRUE,
  icon = "effects",
  "Panel content"
)

# Panel with header actions and footer
calcite_panel(
  heading = "Settings",
  header_actions_start = calcite_action(
    icon = "question",
    text = "Help",
    slot = "header-actions-start"
  ),
  header_actions_end = calcite_action(
    icon = "save",
    text = "Save",
    slot = "header-actions-end"
  ),
  footer = htmltools::tagList(
    calcite_button(width = "half", appearance = "outline", "Cancel"),
    calcite_button(width = "half", "Save")
  ),
)
```

```
    "Panel content"
  )
```

---

calcite\_popover      *Create a Popover component*

---

## Description

Create a Popover component

## Usage

```
calcite_popover(...)
```

## Arguments

```
...                    named attributes passed to htmltools::tag()
```

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
<code>autoClose</code>	<code>auto-close</code>	When <code>true</code> , clicking outside of the component automatically closes open c
<code>closable</code>	<code>closable</code>	When <code>true</code> , displays a close button within the component.
<code>flipDisabled</code>	<code>flip-disabled</code>	When <code>true</code> , prevents flipping the component's placement when overlapping
<code>flipPlacements</code>	NA	Specifies the component's fallback placement when it's initial or specified
<code>focusTrapDisabled</code>	<code>focus-trap-disabled</code>	When <code>true</code> , prevents focus trapping.
<code>heading</code>	<code>heading</code>	The component header text.
<code>headingLevel</code>	<code>heading-level</code>	Specifies the heading level of the component's heading for proper docume
<code>initialFocusTrapFocus</code>	<code>initial-focus-trap-focus</code>	Specifies whether focus should move to the popover when the focus trap is
<code>label</code>	<code>label</code>	Accessible name for the component.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>offsetDistance</code>	<code>offset-distance</code>	Offsets the position of the popover away from the <code>referenceElement</code> .
<code>offsetSkidding</code>	<code>offset-skidding</code>	Offsets the position of the component along the <code>referenceElement</code> .
<code>open</code>	<code>open</code>	When <code>true</code> , displays and positions the component.
<code>overlayPositioning</code>	<code>overlay-positioning</code>	Determines the type of positioning to use for the overlaid content. Using "
<code>placement</code>	<code>placement</code>	Determines where the component will be positioned relative to the referen
<code>pointerDisabled</code>	<code>pointer-disabled</code>	When <code>true</code> , removes the caret pointer.
<code>referenceElement</code>	<code>reference-element</code>	The <code>referenceElement</code> used to position the component according to its pl
<code>scale</code>	<code>scale</code>	Specifies the size of the component.
<code>triggerDisabled</code>	<code>trigger-disabled</code>	When <code>true</code> , disables automatically toggling the component when its refer

**Events:**

The following events are observed by shiny:

Event	Description
calcitePopoverBeforeClose	Fires when the component is requested to be closed and before the closing transition begins.
calcitePopoverBeforeOpen	Fires when the component is added to the DOM but not rendered, and before the opening transition begins.
calcitePopoverClose	Fires when the component is closed and animation is complete.
calcitePopoverOpen	Fires when the component is open and animation is complete.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding custom content.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_popover()
```

---

calcite\_progress      *Create a Progress component*

---

**Description**

Create a Progress component

**Usage**

```
calcite_progress(...)
```

**Arguments**

...      named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
label	label	Accessible name for the component.
reversed	reversed	When true and for "indeterminate" progress bars, reverses the animation direction.
text	text	Text that displays under the component's indicator.
type	type	Specifies the component type. Use "indeterminate" if finding actual progress value is impossible.
value	value	When type is "determinate", specifies the component's value with a range of 0 to 100.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_progress()
```

---

```
calcite_radio_button Create a RadioButton component
```

---

**Description**

Create a `RadioButton` component

**Usage**

```
calcite_radio_button(...)
```

**Arguments**

```
...          named attributes passed to htmltools::tag()
```

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
checked	checked	When true, the component is checked.
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.

form	form	The id of the form that will be associated with the component. When not set, the component will be associated with the parent form.
name	name	Specifies the name of the component. Can be inherited from calcite-radio-button-group. Required.
required	required	When true and the component resides in a form, the component must have a value selected from the component.
scale	scale	Specifies the size of the component inherited from the calcite-radio-button-group.
value	value	The component's value.

**Events:**

The following events are observed by shiny:

Event	Description
calciteRadioButtonChange	Fires only when the radio button is checked. This behavior is identical to the native HTML input type="radio".

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_radio_button()
```

---

```
calcite_radio_button_group
```

*Create a RadioButtonGroup component*

---

**Description**

Create a RadioButtonGroup component

**Usage**

```
calcite_radio_button_group(...)
```

**Arguments**

... named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
layout	layout	Defines the layout of the component.
name	name	Specifies the name of the component on form submission. Must be unique to other components on the page.
required	required	When true and the component resides in a form, the component must have a value.
scale	scale	Specifies the size of the component.
selectedItem	NA	Specifies the component's selected item.
status	status	Specifies the status of the validation message.
validationIcon	validation-icon	Specifies the validation icon to display under the component.
validationMessage	validation-message	Specifies the validation message to display under the component.

**Events:**

The following events are observed by shiny:

Event	Description
calciteRadioButtonGroupChange	Fires when the component has changed.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-radio-buttons.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_radio_button_group()
```

---

calcite\_rating      *Create a Rating component*

---

**Description**

Create a Rating component

**Usage**

```
calcite_rating(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
average	average	Specifies a cumulative average from previous ratings to display.
count	count	Specifies the number of previous ratings to display.
disabled	disabled	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.
form	form	The id of the form that will be associated with the component. When not set, the component is associated with the form of the parent component.
messageOverrides	NA	Use this property to override individual strings used by the component.
name	name	Specifies the name of the component. Required to pass the component's value on the wire.
readOnly	read-only	When <code>true</code> , the component's value can be read, but cannot be modified.
scale	scale	Specifies the size of the component.
showChip	show-chip	When <code>true</code> , and if available, displays the average and/or count data summary in a chip.
status	status	Specifies the status of the input field, which determines message and icons.
validationIcon	validation-icon	Specifies the validation icon to display under the component.
validationMessage	validation-message	Specifies the validation message to display under the component.
validity	NA	The current validation state of the component.
value	value	The component's value.

**Events:**

The following events are observed by shiny:

Event	Description
<code>calciteRatingChange</code>	Fires when the component's value changes.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_rating()
```

---

`calcite_scrim`*Create a Calcite Scrim Component*

---

## Description

Scrim is an overlay placed on top of content to show disabled or loading states. By default, Scrim positions to the extent of its closest parent. To display a Scrim with custom positioning, add `position: relative` styling to its closest parent container.

## Usage

```
calcite_scrim(..., id = NULL, loading = FALSE)
```

## Arguments

<code>...</code>	Main content for the scrim (default slot — primarily loading information)
<code>id</code>	Component ID (required for Shiny reactivity)
<code>loading</code>	When TRUE, a busy indicator is displayed in the center. Default: FALSE.

## Details

### Shiny Integration:

When given an `id`, the scrim reports its state as `input$id` — a named list:

- `$loading` — logical, whether the loading indicator is shown

Use `update_calcite(id, loading = TRUE/FALSE)` to toggle the loading state from the server.

### Positioning:

Scrim fills its closest positioned parent. Wrap the area you want to cover in a container with `position: relative` so the scrim covers the right region.

## Value

An object of class `calcite_component` which is a subclass of `shiny.tag`

## References

[Official Documentation](#)

## Examples

```
calcite_scrim(id = "my_scrim", loading = TRUE)
```

---

 calcite\_segmented\_control

*Create a Calcite Segmented Control Component*


---

## Description

Creates a segmented control for selecting between multiple options. Similar to radio buttons but with a more compact, segmented button appearance.

## Usage

```
calcite_segmented_control(
  ...,
  id = NULL,
  value = NULL,
  appearance = NULL,
  disabled = NULL,
  layout = NULL,
  scale = NULL,
  width = NULL,
  name = NULL,
  label_text = NULL,
  status = NULL,
  validation_message = NULL,
  required = NULL
)
```

## Arguments

...	Additional attributes passed to the component
id	Component ID (required for Shiny reactivity)
value	Initial selected value (should match a child item's value)
appearance	Visual style: "solid", "outline", or "outline-fill"
disabled	Whether the control is disabled (default: FALSE)
layout	Orientation: "horizontal" or "vertical"
scale	Size of the control: "s" (small), "m" (medium), or "l" (large)
width	Width behavior: "auto" or "full"
name	Name for form submission
label_text	Label displayed on the component
status	Validation state: "idle", "valid", or "invalid"
validation_message	Message displayed for validation feedback
required	Whether selection is required

**Details****Shiny Integration:**

The segmented control emits a `calciteSegmentedControlChange` event when the selected item changes.

**Available properties in `input$id`:**

- `$value` - Currently selected value (matches a child item's value attribute)
- `$disabled` - Whether the control is disabled
- `$scale` - Current scale setting
- Other component properties

**Basic usage:**

```
calcite_segmented_control(
  id = "effect_type",
  width = "full",
  calcite_segmented_control_item(value = "blur"),
  calcite_segmented_control_item(value = "highlight", checked = TRUE),
  calcite_segmented_control_item(value = "party")
)

# In server
observeEvent(input$effect_type, {
  selected <- input$effect_type$value
  print(paste("Selected:", selected))
})
```

**Update from server:**

```
update_calcite("effect_type", value = "blur")
```

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
# Basic segmented control
calcite_segmented_control(
  id = "view_mode",
  calcite_segmented_control_item(value = "list", icon_start = "list"),
  calcite_segmented_control_item(value = "grid", icon_start = "grid", checked = TRUE),
  calcite_segmented_control_item(value = "table", icon_start = "table")
)

# Full width with text
calcite_segmented_control(
```

```

    id = "effect",
    width = "full",
    calcite_segmented_control_item(value = "Blur"),
    calcite_segmented_control_item(value = "Highlight", checked = TRUE),
    calcite_segmented_control_item(value = "Party mode")
  )

# Shiny example
if (interactive()) {
  library(shiny)

  ui <- calcite_shell(
    calcite_card(
      heading = "Segmented Control Example",
      calcite_label(
        "Choose an option",
        calcite_segmented_control(
          id = "my_control",
          width = "full",
          calcite_segmented_control_item(value = "option1"),
          calcite_segmented_control_item(value = "option2", checked = TRUE),
          calcite_segmented_control_item(value = "option3")
        )
      ),
      verbatimTextOutput("selected_value")
    )
  )

  server <- function(input, output, session) {
    output$selected_value <- renderPrint({
      paste("Selected:", input$my_control$value)
    })
  }

  shinyApp(ui, server)
}

```

---

calcite\_segmented\_control\_item

*Create a Calcite Segmented Control Item Component*

---

## Description

Creates an individual item within a segmented control. Must be used as a child of `calcite_segmented_control()`.

## Usage

```

calcite_segmented_control_item(
  value,
  label = value,

```

```

    checked = NULL,
    icon_start = NULL,
    icon_end = NULL,
    icon_flip_rtl = NULL,
    ...
)

```

### Arguments

value	The value this item represents (required)
label	Text label displayed for the item (defaults to value)
checked	Whether this item is initially selected (default: FALSE)
icon_start	Icon to display at the start of the item
icon_end	Icon to display at the end of the item
icon_flip_rtl	Whether to flip the icon in RTL languages (default: FALSE)
...	Additional content or attributes (text label if provided as unnamed argument)

### Value

An object of class `calcite_component`

### References

[Official Documentation](#)

### Examples

```

# Item with text label
calcite_segmented_control_item(value = "option1")

# Item with icon
calcite_segmented_control_item(
  value = "list",
  icon_start = "list"
)

# Initially selected item
calcite_segmented_control_item(
  value = "default",
  checked = TRUE
)

```

---

 calcite\_select

 Create a Calcite Select Component
 

---

## Description

Select provides a single-selection dropdown menu for forms. It displays a list of options that users can choose from. You can either provide values and labels vectors for convenience, or manually construct `calcite_option()` components.

## Usage

```
calcite_select(
  ...,
  values = NULL,
  labels = NULL,
  id = NULL,
  label,
  disabled = NULL,
  form = NULL,
  label_text = NULL,
  message_overrides = NULL,
  name = NULL,
  required = NULL,
  scale = NULL,
  status = NULL,
  validation_icon = NULL,
  validation_message = NULL,
  value = NULL,
  width = NULL,
  label_content = NULL
)
```

## Arguments

...	calcite-option components (default slot). Ignored if values is provided.
values	Character vector of option values. When provided, options are automatically generated from values and labels. Takes precedence over manually constructed options in ...
labels	Character vector of option labels (display text). Must be same length as values. If NULL, uses values as labels.
id	Component ID (required for Shiny reactivity)
label	Accessible name for the component (required)
disabled	When true, interaction is prevented and the component is displayed with lower opacity
form	The id of the form that will be associated with the component

label_text	When provided, displays label text on the component
message_overrides	Override individual strings used by the component
name	Specifies the name of the component. Required to pass the component's value on form submission
required	When true, the component must have a value for the form to submit
scale	Specifies the size of the component: "s" (small), "m" (medium), or "l" (large)
status	Specifies the status of the input field: "idle", "invalid", or "valid"
validation_icon	Specifies the validation icon to display under the component
validation_message	Specifies the validation message to display under the component
value	The component's selectedOption value
width	Specifies the width of the component: "auto", "full", or "half" (half is deprecated)
label_content	Content to render next to the component's labelText (slot)

## Details

### Shiny Integration:

When used in a Shiny app, `calcite_select()` returns a reactive list containing the component's state. The component emits a `calciteSelectChange` event when the selection changes.

### Available properties in `input$id`:

- `$value` - The value of the selected option
- `$disabled` - Whether the select is disabled
- `$scale` - The size of the component
- `$status` - The validation status
- Other component properties

### Events:

- `calciteSelectChange` - Fires when the selected option changes

## Value

An object of class `calcite_component`

## References

[Official Documentation](#)

## Examples

```
# Using values and labels vectors (convenient)
calcite_select(
  id = "my_select",
  label = "Choose terrain type",
  values = c("mountains", "rivers", "lakes", "buttes", "fjords"),
  labels = c("Mountains", "Rivers", "Lakes", "Buttes", "Fjords")
)

# Using values only (labels default to values)
calcite_select(
  id = "simple_select",
  label = "Choose an option",
  values = c("option1", "option2", "option3")
)

# Manually constructing options (more control)
calcite_select(
  id = "fruit_select",
  label = "Choose a fruit",
  label_text = "Select your favorite fruit",
  scale = "m",
  calcite_option(label = "Apple", value = "apple"),
  calcite_option(label = "Banana", value = "banana", selected = TRUE),
  calcite_option(label = "Orange", value = "orange")
)

# Required select with validation
calcite_select(
  id = "required_select",
  label = "Required field",
  label_text = "Make a selection",
  required = TRUE,
  status = "invalid",
  validation_message = "Please select an option",
  calcite_option(label = "Choose...", value = ""),
  calcite_option(label = "Option A", value = "a"),
  calcite_option(label = "Option B", value = "b")
)
```

---

calcite\_sheet

*Create a Sheet component*

---

## Description

Create a Sheet component

## Usage

```
calcite_sheet(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>beforeClose</code>	NA	Passes a function to run before the component closes.
<code>displayMode</code>	<code>display-mode</code>	Specifies the display mode - "float" (content is separated detached), or "overlaid"
<code>escapeDisabled</code>	<code>escape-disabled</code>	When true, disables the default close on escape behavior.
<code>focusTrapDisabled</code>	<code>focus-trap-disabled</code>	When true, prevents focus trapping.
<code>height</code>	<code>height</code>	Specifies the height of the component.
<code>heightScale</code>	<code>height-scale</code>	When position is "block-start" or "block-end", specifies the height of the component.
<code>label</code>	<code>label</code>	Specifies the label of the component.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>open</code>	<code>open</code>	When true, displays and positions the component.
<code>outsideCloseDisabled</code>	<code>outside-close-disabled</code>	When true, disables the closing of the component when clicked outside.
<code>position</code>	<code>position</code>	Determines where the component will be positioned.
<code>resizable</code>	<code>resizable</code>	When true, the component is resizable.
<code>width</code>	<code>width</code>	Specifies the width of the component.
<code>widthScale</code>	<code>width-scale</code>	When position is "inline-start" or "inline-end", specifies the width of the component.

**Events:**

The following events are observed by shiny:

Event	Description
<code>calciteSheetBeforeClose</code>	Fires when the component is requested to be closed and before the closing transition begins.
<code>calciteSheetBeforeOpen</code>	Fires when the component is added to the DOM but not rendered, and before the opening transition begins.
<code>calciteSheetClose</code>	Fires when the component is closed and animation is complete.
<code>calciteSheetOpen</code>	Fires when the component is open and animation is complete.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_sheet()
```

---

 calcite\_shell

 Create a Calcite Shell Layout
 

---

## Description

The Shell is a foundational layout component in Calcite, enabling the creation of rich, interactive experiences. It provides structured slots for headers, footers, side panels, and main content.

## Usage

```
calcite_shell(
  ...,
  header = NULL,
  footer = NULL,
  panel_start = NULL,
  panel_end = NULL,
  panel_top = NULL,
  panel_bottom = NULL,
  modals = NULL,
  dialogs = NULL,
  alerts = NULL,
  sheets = NULL
)
```

## Arguments

...	Main content to display in the default slot (between panels)
header	Content for the header slot (top of shell). Typically a <a href="#">calcite_navigation()</a> component.
footer	Content for the footer slot (bottom of shell)
panel_start	Content for the start/left panel. Typically a <a href="#">calcite_shell_panel()</a> wrapping a <a href="#">calcite_panel()</a> .
panel_end	Content for the end/right panel. Typically a <a href="#">calcite_shell_panel()</a> wrapping a <a href="#">calcite_panel()</a> .
panel_top	Content for the top panel (below header)
panel_bottom	Content for the bottom panel (above footer)
modals	Slot for <a href="#">calcite_modal()</a> components
dialogs	Slot for <a href="#">calcite_dialog()</a> components
alerts	Slot for <a href="#">calcite_alert()</a> components
sheets	Slot for <a href="#">calcite_sheet()</a> components

## Details

### Shell Structure:

The shell organizes your application into distinct regions:

- **header:** Top navigation bar
- **panel-start/panel-end:** Side panels (left/right)
- **panel-top/panel-bottom:** Top/bottom panels
- **Default content:** Main application area (maps, charts, etc.)
- **footer:** Bottom information/links
- **Overlay slots:** modals, dialogs, alerts, sheets

When embedded within other content, the overlay slots (modals, dialogs, alerts, sheets) facilitate placement of these components relative to the Shell, constraining them to the shell's boundaries rather than the full page.

## Value

An object of class `calcite_component` which is a subclass of `shiny.tag`

## References

[Official Documentation](#)

## Examples

```
# Basic shell with header and content
calcite_shell(
  header = calcite_navigation(
    calcite_navigation_logo(slot = "logo", heading = "My App")
  ),
  "Main content goes here"
)

# Shell with sidebar panel
calcite_shell(
  panel_start = calcite_shell_panel(
    calcite_panel(heading = "Layers")
  ),
  calcite_panel(heading = "Map View")
)
```

---

`calcite_shell_center_row`

*Create a ShellCenterRow component*

---

## Description

Use the `calcite-shell-panel` component instead.

**Usage**

```
calcite_shell_center_row(...)
```

**Arguments**

```
...          named attributes passed to htmltools::tag()
```

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
detached	detached	When true, the content area displays like a floating panel.
heightScale	height-scale	Specifies the maximum height of the component.
position	position	Specifies the component's position. Will be flipped when the element direction is right-to-left ("r").

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding content to the calcite-shell-panel.
action-bar	A slot for adding a calcite-action-bar to the calcite-shell-panel.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_shell_center_row()
```

---

calcite\_shell\_panel    *Create a Calcite Shell Panel Component*

---

**Description**

Shell Panel is a layout container that slots into `calcite_shell()` via `panel_start` or `panel_end`. It sits adjacent to or on top of the main content area. Pass a `calcite_panel()` (typically containing `calcite_block()` components) as the default content, and optionally a `calcite_action_bar()` in the `action_bar` slot.

**Usage**

```
calcite_shell_panel(
  ...,
  collapsed = NULL,
  display_mode = NULL,
  height = NULL,
  message_overrides = NULL,
  resizable = NULL,
  width = NULL,
  action_bar = NULL
)
```

**Arguments**

...	Default slot content. Typically a single <a href="#">calcite_panel()</a> .
collapsed	When TRUE, hides the component's content area. Default: FALSE
display_mode	Specifies the display mode: <ul style="list-style-type: none"> <li>• "dock" (default): full height, pushes center content aside</li> <li>• "overlay": full height, displays on top of center content</li> <li>• "float-content": not full height, content detached from action bar, on top of center content</li> <li>• "float-all": detaches both panel and action bar on top of center content</li> </ul>
height	Specifies the component's height.
message_overrides	Override individual strings used by the component.
resizable	When TRUE and display_mode is "dock" or "overlay", the content area is resizable. Default: FALSE
width	Specifies the component's width: "s", "m", or "l".
action_bar	Content for the action-bar slot. Typically a <a href="#">calcite_action_bar()</a> component.

**Details****Usage:**

Shell Panel is the recommended wrapper when placing panels inside [calcite\\_shell\(\)](#)'s `panel_start` or `panel_end` slots. It controls display mode, width, collapsibility, and action bar placement.

Multiple Shell Panels can be open simultaneously, unlike [calcite\\_sheet\(\)](#). It does not provide a page-blocking scrim and is not intended to interrupt the main application workflow.

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
calcite_shell(  
  panel_start = calcite_shell_panel(  
    calcite_panel(  
      heading = "Layers",  
      calcite_block(  
        heading = "Options",  
        collapsible = TRUE,  
        expanded = TRUE  
      )  
    )  
  ),  
  calcite_panel(heading = "Map View")  
)
```

---

`calcite_slider`*Create a Calcite Slider Component*

---

**Description**

Creates a slider input for selecting numeric values. Supports both single-value and range selection (dual handles).

**Usage**

```
calcite_slider(  
  id = NULL,  
  value = NULL,  
  min = 0,  
  max = 100,  
  step = 1,  
  min_value = NULL,  
  max_value = NULL,  
  label_handles = NULL,  
  label_ticks = NULL,  
  label_text = NULL,  
  ticks = NULL,  
  disabled = NULL,  
  required = NULL,  
  scale = NULL,  
  snap = NULL,  
  precise = NULL,  
  mirrored = NULL,  
  fill_placement = NULL,  
  histogram = NULL,  
  histogram_stops = NULL,  
  group_separator = NULL,
```

```

    page_step = NULL,
    min_label = NULL,
    max_label = NULL,
    name = NULL,
    form = NULL,
    numbering_system = NULL,
    status = NULL,
    validation_icon = NULL,
    validation_message = NULL,
    ...
  )

```

### Arguments

<code>id</code>	Component ID (required for Shiny reactivity)
<code>value</code>	Initial value (for single slider) or NULL
<code>min</code>	Minimum selectable value (default: 0)
<code>max</code>	Maximum selectable value (default: 100)
<code>step</code>	Increment step for up/down arrows and keyboard (default: 1)
<code>min_value</code>	For range sliders, the lower bound value
<code>max_value</code>	For range sliders, the upper bound value
<code>label_handles</code>	Whether to display numeric labels on handles (default: FALSE)
<code>label_ticks</code>	Whether to display numeric labels on tick marks (default: FALSE)
<code>label_text</code>	Accessible label text for the component
<code>ticks</code>	Interval for displaying tick marks on the number line
<code>disabled</code>	Whether the slider is disabled (default: FALSE)
<code>required</code>	Whether a value is required for form submission (default: FALSE)
<code>scale</code>	Size of the slider: "s" (small), "m" (medium), or "l" (large)
<code>snap</code>	Whether to enable snap-to-step on mouse interaction (default: FALSE)
<code>precise</code>	Whether to use finer positioning for handles (default: FALSE)
<code>mirrored</code>	Whether to mirror the slider (default: FALSE)
<code>fill_placement</code>	Where to display the fill: "start", "end", or "none"
<code>histogram</code>	A 2-column numeric matrix or data.frame for histogram display
<code>histogram_stops</code>	Color stops for histogram display
<code>group_separator</code>	Whether to display thousand separators in numbers (default: FALSE)
<code>page_step</code>	Interval to move with Page Up/Down keys
<code>min_label</code>	Accessible label for the minimum handle (for screen readers)
<code>max_label</code>	Accessible label for the maximum handle (for screen readers)
<code>name</code>	Name attribute for form submission

form	Associated form element ID
numbering_system	Numbering system to use: "arab", "arabext", or "latn"
status	Validation state: "idle", "valid", or "invalid"
validation_icon	Icon to display for validation feedback
validation_message	Message displayed for validation feedback
...	Additional attributes passed to the component

## Details

### Shiny Integration:

The slider emits two types of events:

- `calciteSliderChange` - Fires when the user releases the handle (debounced, final value)
- `calciteSliderInput` - Fires continuously during drag (use with caution for expensive operations)

### Available properties in `input$id`:

- `$value` - Current value (single slider) or array of [`minValue`, `maxValue`] (range slider)
- `$min / $max` - Slider bounds
- `$step` - Step increment
- `$disabled` - Whether disabled
- Other component properties

### Single-value slider:

```
observeEvent(input$my_slider, {
  current_value <- input$my_slider$value
  print(paste("Slider value:", current_value))
})
```

### Range slider (dual handles):

```
# Provide both min_value and max_value to create a range slider
calcite_slider(
  id = "range_slider",
  min = 0,
  max = 100,
  min_value = 20,
  max_value = 80
)

observeEvent(input$range_slider, {
  lower <- input$range_slider$minValue
  upper <- input$range_slider$maxValue
  print(paste("Range:", lower, "to", upper))
})
```

**Update from server:**

```
# Update single value
update_calcite("my_slider", value = 50)

# Update range
update_calcite("range_slider", minValue = 30, maxValue = 70)
```

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
# Basic slider
calcite_slider(
  id = "my_slider",
  value = 50,
  min = 0,
  max = 100,
  step = 5
)

# Slider with ticks and labels
calcite_slider(
  id = "temperature",
  value = 72,
  min = 32,
  max = 212,
  step = 1,
  ticks = 10,
  label_handles = TRUE,
  label_ticks = TRUE
)

# Range slider (dual handles)
calcite_slider(
  id = "price_range",
  min = 0,
  max = 1000,
  min_value = 100,
  max_value = 500,
  step = 10,
  label_handles = TRUE
)

# Shiny example
if (interactive()) {
```

```
library(shiny)

ui <- calcite_shell(
  calcite_card(
    heading = "Slider Example",
    calcite_label(
      "Choose a value",
      calcite_slider(
        id = "my_slider",
        value = 50,
        min = 0,
        max = 100,
        step = 5,
        label_handles = TRUE
      )
    ),
    verbatimTextOutput("slider_value")
  )
)

server <- function(input, output, session) {
  output$slider_value <- renderPrint({
    paste("Current value:", input$my_slider$value)
  })
}

shinyApp(ui, server)
}
```

---

calcite\_split\_button *Create a SplitButton component*

---

### Description

Create a SplitButton component

### Usage

```
calcite_split_button(...)
```

### Arguments

...                    named attributes passed to `htmltools::tag()`

### Details

#### Properties:

The following properties are provided by this component:

Name	Attribute	Description
appearance	appearance	Specifies the appearance style of the component.
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
dropdownIconType	dropdown-icon-type	Specifies the icon used for the dropdown menu.
dropdownLabel	dropdown-label	Accessible name for the dropdown menu.
flipPlacements	NA	Specifies the component's fallback slotted content placement when it's initial placement is not available.
kind	kind	Specifies the kind of the component, which will apply to border and background colors.
loading	loading	When true, a busy indicator is displayed on the primary button.
overlayPositioning	overlay-positioning	Determines the type of positioning to use for the overlaid content. Using "absolute" will allow you to position the component in the page, regardless of the container element's positioning.
placement	placement	Determines where the component will be positioned relative to the container element.
primaryIconEnd	primary-icon-end	Specifies an icon to display at the end of the primary button.
primaryIconFlipRtl	primary-icon-flip-rtl	Displays the primaryIconStart and/or primaryIconEnd as flipped when the element is right-to-left.
primaryIconStart	primary-icon-start	Specifies an icon to display at the start of the primary button.
primaryLabel	primary-label	Accessible name for the primary button.
primaryText	primary-text	Text displayed in the primary button.
scale	scale	Specifies the size of the component.
width	width	Check API reference

**Events:**

The following events are observed by shiny:

Event	Description
calciteSplitButtonPrimaryClick	Fires when the primary button is clicked.
calciteSplitButtonSecondaryClick	Fires when the dropdown menu is clicked.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-dropdown content.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_split_button()
```

---

calcite_stepper	<i>Create a Stepper component</i>
-----------------	-----------------------------------

---

**Description**

Create a Stepper component

**Usage**

```
calcite_stepper(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
icon	icon	When true, displays a status icon in the <code>calcite-stepper-item</code> heading.
layout	layout	Defines the layout of the component.
messageOverrides	NA	Use this property to override individual strings used by the component.
numbered	numbered	When true, displays the step number in the <code>calcite-stepper-item</code> heading.
numberingSystem	numbering-system	Specifies the Unicode numeral system used by the component for localization.
scale	scale	Specifies the size of the component.
selectedItem	NA	Specifies the component's selected item.

**Events:**

The following events are observed by shiny:

Event	Description
calciteStepperChange	Fires when the active <code>calcite-stepper-item</code> changes.
calciteStepperItemChange	Fires when the active <code>calcite-stepper-item</code> changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding <code>calcite-stepper-item</code> elements.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_stepper()
```

---

calcite\_stepper\_item *Create a StepperItem component*

---

**Description**

Create a StepperItem component

**Usage**

```
calcite_stepper_item(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
complete	complete	When true, the step has been completed.
description	description	A description for the component. Displays below the header text.
disabled	disabled	When true, interaction is prevented and the component is displayed with lower opacity.
error	error	When true, the component contains an error that requires resolution from the user.
heading	heading	The component header text.
iconFlipRtl	icon-flip-rtl	When true, the icon will be flipped when the element direction is right-to-left ("rtl").
messageOverrides	NA	Use this property to override individual strings used by the component.
selected	selected	When true, the component is selected.

**Events:**

The following events are observed by shiny:

Event	Description
calciteStepperItemSelect	Fires when the active <code>calcite-stepper-item</code> changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding custom content.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_stepper_item()
```

---

`calcite_switch`      *Create a Calcite Switch Component*

---

**Description**

Creates a switch component for "on" and "off" choices. Switches should be used for boolean choices and can be useful for opt-in and additive actions.

**Usage**

```
calcite_switch(  
  id = NULL,  
  checked = NULL,  
  disabled = NULL,  
  form = NULL,  
  label = NULL,  
  label_text_end = NULL,  
  label_text_start = NULL,  
  name = NULL,  
  scale = NULL,  
  value = NULL,  
  ...  
)
```

**Arguments**

id	Optional ID for the switch (required for Shiny reactivity)
checked	When TRUE, the component is checked
disabled	When TRUE, interaction is prevented and the component is displayed with lower opacity
form	The id of the form that will be associated with the component
label	Accessible name for the component
label_text_end	When provided, displays label text at the end of the component
label_text_start	When provided, displays label text at the start of the component
name	Specifies the name of the component. Required to pass the component's value on form submission
scale	Specifies the size of the component: "s" (small), "m" (medium), or "l" (large)
value	The component's value
...	Additional attributes passed to the component

**Details****Shiny Integration:**

When used in a Shiny app, `calcite_switch()` returns a reactive list containing all component properties. You can observe the entire component state or watch individual properties:

**Available properties:**

- `$checked` - Whether the switch is currently checked (on/off)
- `$disabled` - Whether the switch is disabled
- `$value` - The switch's value
- `$name` - The component's name
- Other component properties

**Usage in server:**

```
# Watch for changes to the switch
observeEvent(input$my_switch, {
  print(paste("Switch is:", if(input$my_switch$checked) "ON" else "OFF"))
})

# Access the checked state directly
observeEvent(input$my_switch$checked, {
  if (input$my_switch$checked) {
    print("Switch turned ON")
  } else {
    print("Switch turned OFF")
  }
})
```

**Value**

An object of class `calcite_component`

**References**

[Official Documentation](#)

**Examples**

```
# Basic switch
calcite_switch(
  id = "my_switch",
  label = "Enable 3D mode"
)

# Switch with label text on both sides
calcite_switch(
  id = "theme_switch",
  label_text_start = "Light",
  label_text_end = "Dark",
  label = "Theme toggle"
)

# Checked switch at large scale
calcite_switch(
  checked = TRUE,
  scale = "l",
  label = "Notifications"
)

# Shiny example
if (interactive()) {
  library(shiny)

  ui <- calcite_shell(
    calcite_panel(
      heading = "Switch Example",

      calcite_label(
        layout = "inline",
        "3D Off",
        calcite_switch(id = "mode_switch"),
        "3D On"
      ),

      verbatimTextOutput("status")
    )
  )

  server <- function(input, output, session) {
    # Display switch state
    output$status <- renderPrint({
```

```

      input$mode_switch
    })
  }

  shinyApp(ui, server)
}

```

---

 calcite\_tab

*Create a Tab component*


---

## Description

Create a Tab component

## Usage

```
calcite_tab(...)
```

## Arguments

...                    named attributes passed to `htmltools::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
selected	selected	When <code>true</code> , the component's contents are selected. Only one tab can be selected within the <code>calcite-tab</code> .
tab	tab	Specifies a unique name for the component. When specified, use the same value on the <code>calcite-tab</code> .

### Slots:

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding custom content.

## Value

an object of class `calcite_component` which is a subclass of `shiny.tag`

## References

[Official Documentation](#)

## Examples

```
calcite_tab()
```

---

calcite_table	<i>Create a Calcite Table Component</i>
---------------	-----------------------------------------

---

## Description

Creates a table component from a data.frame with automatic row and cell generation. The table provides pagination, numbering, and various styling options.

## Usage

```
calcite_table(
  data,
  ...,
  id = NULL,
  caption,
  header = NULL,
  alignment = "start",
  bordered = NULL,
  numbered = NULL,
  page_size = NULL,
  scale = NULL,
  striped = NULL,
  numbering_system = NULL
)
```

## Arguments

data	A data.frame to display in the table (required)
...	Additional attributes passed to the component
id	Component ID (required for Shiny reactivity)
caption	Accessible title for the table (required)
header	Custom header row created with <code>calcite_table_header()</code> . If NULL, column names from data are used
alignment	Alignment for all cells: "start", "center", or "end" (default: "start")
bordered	When TRUE, displays borders (default: FALSE)
numbered	When TRUE, displays row numbers (default: FALSE)
page_size	Page size for pagination. When > 0, renders pagination controls (default: 0)
scale	Size of the component: "s" (small), "m" (medium), or "l" (large) (default: "m")
striped	When TRUE, displays striped styling (default: FALSE)
numbering_system	Unicode numeral system: "arab", "arabext", or "latn"

## Details

### Shiny Integration:

The table emits events for page changes and selection changes (when selection is enabled).

### Available properties in `input$id`:

- `$pageSize` - Current page size
- `$scale` - Component scale
- `$bordered` - Whether borders are displayed
- `$numbered` - Whether row numbers are displayed
- `$striped` - Whether striped styling is applied
- Other component properties

### Events:

- `calciteTablePageChange` - Fires when the page changes

## Value

An object of class `calcite_component`

## References

[Official Documentation](#)

## Examples

```
# Basic table from data frame
calcite_table(
  data = mtcars[1:5, 1:4],
  id = "my_table",
  caption = "Motor Trend Car Data",
  bordered = TRUE,
  striped = TRUE
)

# Table with pagination
calcite_table(
  data = iris,
  id = "iris_table",
  caption = "Iris Dataset",
  page_size = 10,
  numbered = TRUE
)

# Table with custom headers
calcite_table(
  data = mtcars[1:5, 1:3],
  caption = "Cars",
  header = list(
    calcite_table_header(heading = "Miles/Gallon", description = "Fuel efficiency"),
    calcite_table_header(heading = "Cylinders", description = "Number of cylinders"),
    calcite_table_header(heading = "Displacement", description = "Engine size")
  )
)
```

```
)
)
```

---

calcite\_table\_header    *Create a Calcite Table Header*

---

### Description

Creates a header cell for use in calcite\_table headers.

### Usage

```
calcite_table_header(heading, description = NULL, alignment = "start")
```

### Arguments

heading	Header text to display
description	Optional description text below heading
alignment	Alignment: "start", "center", or "end" (default: "start")

### Value

A calcite-table-header tag

### Examples

```
calcite_table_header(
  heading = "Name",
  description = "Person's full name"
)
```

---

calcite\_tabs                    *Create a Tabs component*

---

### Description

Create a Tabs component

### Usage

```
calcite_tabs(...)
```

### Arguments

...	named attributes passed to <code>htmltools::tag()</code>
-----	----------------------------------------------------------

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
bordered	bordered	When true, the component will display with a folder style menu.
layout	layout	Specifies the layout of the calcite-tab-nav, justifying the calcite-tab-titles to the start ("inline-block") or right ("float-right").
position	position	Specifies the position of calcite-tab-nav and calcite-tab-title components in relation to the calcite-tab-content.
scale	scale	Specifies the size of the component.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-tabs.
title-group	A slot for adding a calcite-tab-nav.

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_tabs()
```

---

calcite\_tab\_nav      *Create a TabNav component*

---

**Description**

Create a TabNav component

**Usage**

```
calcite_tab_nav(...)
```

**Arguments**

...      named attributes passed to htmltools::tag()

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
messageOverrides	NA	Use this property to override individual strings used by the component.
position	position	Specifies the position of calcite-tab-nav and calcite-tab-title components in relation to the parent component.
selectedTitle	NA	Specifies the component's selected calcite-tab-title.
storageId	storage-id	Specifies the name when saving selected calcite-tab data to localStorage.
syncId	sync-id	Specifies text to update multiple components to keep in sync if one changes.

**Events:**

The following events are observed by shiny:

Event	Description
calciteTabChange	Emits when the selected calcite-tab changes.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding calcite-tab-titles.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_tab_nav()
```

---

```
calcite_tab_title      Create a TabTitle component
```

---

**Description**

Tab-titles are optionally individually closable.

**Usage**

```
calcite_tab_title(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>closable</code>	<code>closable</code>	When true, a close button is added to the component.
<code>closed</code>	<code>closed</code>	When true, does not display or position the component.
<code>disabled</code>	<code>disabled</code>	When true, interaction is prevented and the component is displayed with lower opacity.
<code>iconEnd</code>	<code>icon-end</code>	Specifies an icon to display at the end of the component.
<code>iconFlipRtl</code>	<code>icon-flip-rtl</code>	Displays the <code>iconStart</code> and/or <code>iconEnd</code> as flipped when the element direction is right-to-left.
<code>iconStart</code>	<code>icon-start</code>	Specifies an icon to display at the start of the component.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>position</code>	<code>position</code>	Specifies the position of <code>calcite-tab-nav</code> and <code>calcite-tab-title</code> components in relation to the component.
<code>selected</code>	<code>selected</code>	When true, the component and its respective <code>calcite-tab</code> contents are selected. Only one component can be selected at a time.
<code>tab</code>	<code>tab</code>	Specifies a unique name for the component. When specified, use the same value on the <code>calcite-tab</code> component.

**Events:**

The following events are observed by shiny:

Event	Description
<code>calciteTabsActivate</code>	Fires when a <code>calcite-tab</code> is selected.
<code>calciteTabsClose</code>	Fires when a <code>calcite-tab</code> is closed.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding text.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_tab_title()
```

---

calcite\_text\_area      *Create a TextArea component*

---

## Description

Create a TextArea component

## Usage

```
calcite_text_area(...)
```

## Arguments

...                    named attributes passed to `htmltools::tag()`

## Details

### Properties:

The following properties are provided by this component:

Name	Attribute	Description
columns	columns	Specifies the component's number of columns.
disabled	disabled	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.
form	form	The id of the form that will be associated with the component. When not set, the component is associated with the form of the parent component.
groupSeparator	group-separator	When <code>true</code> , number values are displayed with a group separator corresponding to the locale.
label	label	Accessible name for the component.
limitText	limit-text	Check API reference
maxLength	max-length	When the component resides in a form, specifies the maximum number of characters that can be entered.
messageOverrides	NA	Use this property to override individual strings used by the component.
minLength	min-length	When the component resides in a form, specifies the minimum number of characters that can be entered.
name	name	Specifies the name of the component.
numberingSystem	numbering-system	Specifies the Unicode numeral system used by the component for localization.
placeholder	placeholder	Specifies the placeholder text for the component.
readOnly	read-only	When <code>true</code> , the component's value can be read, but cannot be modified.
required	required	When <code>true</code> and the component resides in a form, the component must have a value.
resize	resize	Specifies if the component is resizable.
rows	rows	Specifies the component's number of rows.
scale	scale	Specifies the size of the component.
status	status	Specifies the status of the input field, which determines message and icons.
validationIcon	validation-icon	Specifies the validation icon to display under the component.
validationMessage	validation-message	Specifies the validation message to display under the component.
validity	NA	The current validation state of the component.
value	value	The component's value.
wrap	wrap	Specifies the wrapping mechanism for the text.

**Events:**

The following events are observed by shiny:

Event	Description
calciteTextAreaChange	Fires each time a new value is typed and committed.
calciteTextAreaInput	Fires each time a new value is typed.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding text.
footer-start	A slot for adding content to the start of the component's footer.
footer-end	A slot for adding content to the end of the component's footer.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_text_area()
```

---

calcite\_tile      *Create a Calcite Tile Component*

---

**Description**

Tiles are presentational components useful for presenting consequential information in a compact, consistent format. They can contain supportive icons, a heading, and a description.

**Usage**

```
calcite_tile(  
  ...,  
  id = NULL,  
  heading = NULL,  
  description = NULL,  
  icon = NULL,  
  href = NULL,  
  active = NULL,  
  selected = NULL,
```

```

disabled = NULL,
alignment = NULL,
scale = NULL,
icon_flip_rtl = NULL,
label = NULL,
content_top = NULL,
content_bottom = NULL
)

```

## Arguments

...	Child content for the tile
id	Optional ID for the tile (required for Shiny reactivity)
heading	The component header text
description	A description for the component, which displays below the heading
icon	Specifies an icon to display
href	When provided, the URL for the component (makes tile a link)
active	When TRUE, the component is active (default: FALSE)
selected	When TRUE and parent's selectionMode allows it, component is selected (default: FALSE)
disabled	When TRUE, interaction is prevented and component displays with lower opacity (default: FALSE)
alignment	Specifies alignment of tile's content: "start" or "center"
scale	Specifies size of the component: "s" (small), "m" (medium), or "l" (large)
icon_flip_rtl	When TRUE, icon will be flipped when element direction is RTL (default: FALSE)
label	Accessible name for the component
content_top	Slot for adding non-interactive elements above the component's content
content_bottom	Slot for adding non-interactive elements below the component's content

## Details

### Best Practices:

- Tiles are best used to represent one of a limited number of options, actions, or choices
- The component is wholly focusable - it should not contain slotted focusable elements
- Text should be concise (heading max ~50 chars, description max ~175 chars)
- Use sentence case for heading and description
- End description with proper punctuation

### Shiny Integration:

When used in a Shiny app with an id, `calcite_tile()` returns a reactive list containing component properties.

### Available properties:

- `$active` - Whether the tile is currently active
- `$selected` - Whether the tile is selected
- `$disabled` - Whether the tile is disabled
- `$heading` - The heading text
- `$description` - The description text
- `$icon` - The icon name
- Other component properties

### Value

An object of class `calcite_component`

### References

[Official Documentation](#)

### Examples

```
# Basic tile with icon and description
calcite_tile(
  icon = "3d-glasses",
  heading = "Special viewing glasses",
  description = "Great for eclipses and optical illusions"
)

# Tile with content in bottom slot
calcite_tile(
  icon = "rangefinder",
  heading = "Rangefinder",
  description = "A time-tested tool for field engineers",
  content_bottom = calcite_chip("214 in use")
)

# Active tile with link
calcite_tile(
  icon = "data",
  heading = "Data Analysis",
  href = "https://example.com/data",
  active = TRUE
)
```

---

`calcite_tile_group`      *Create a Calcite Tile Group Component*

---

### Description

Tile Group can be used to organize multiple Tile components within a layout. It supports workflows and patterns using more than one Tile, with opt-in selection modes for interactive workflows.

**Usage**

```
calcite_tile_group(
  ...,
  id = NULL,
  label = NULL,
  layout = NULL,
  alignment = NULL,
  scale = NULL,
  selection_mode = NULL,
  selection_appearance = NULL,
  disabled = NULL
)
```

**Arguments**

...	Child <code>calcite_tile()</code> components
<code>id</code>	Optional ID for the tile group (required for Shiny reactivity to track selection)
<code>label</code>	Accessible name for the component (required for accessibility)
<code>layout</code>	Defines the layout: "horizontal" for rows, "vertical" for a single column
<code>alignment</code>	Specifies alignment of each tile's content: "start" or "center"
<code>scale</code>	Specifies size of the component: "s" (small), "m" (medium), or "l" (large)
<code>selection_mode</code>	Specifies the selection mode: "none" (default), "single", "single-persist", or "multiple"
<code>selection_appearance</code>	Specifies selection appearance: "icon" (checkmark/dot) or "border"
<code>disabled</code>	When TRUE, interaction is prevented and component displays with lower opacity (default: FALSE)

**Details****Selection Modes:**

- "none" - No selection allowed (default)
- "single" - Only one tile can be selected at a time
- "single-persist" - Only one tile selected, prevents de-selection
- "multiple" - Any number of tiles can be selected

**Shiny Integration:**

When used in a Shiny app with an `id`, `calcite_tile_group()` returns a reactive list containing component properties and emits events when selection changes.

**Available properties:**

- `$selectedItems` - Array of selected tile IDs
- `$selectionMode` - Current selection mode
- `$layout` - Current layout
- `$disabled` - Whether the group is disabled

- Other component properties

#### Usage in server:

```
# Watch for selection changes
observeEvent(input$my_tile_group$selectedItems, {
  selected <- input$my_tile_group$selectedItems
  message("Selected tiles: ", paste(selected, collapse = ", "))
})
```

#### Value

An object of class `calcite_component`

#### References

[Official Documentation](#)

#### Examples

```
# Basic tile group
calcite_tile_group(
  label = "Role selection",
  calcite_tile(
    icon = "rangefinder",
    heading = "Field operator",
    description = "Create and edit Reports in the field"
  ),
  calcite_tile(
    icon = "knowledge-graph-dashboard",
    heading = "Office coordinator",
    description = "View and analyze Reports from the office"
  )
)

# Tile group with multiple selection
calcite_tile_group(
  id = "role_selector",
  label = "Select roles",
  selection_mode = "multiple",
  selection_appearance = "border",
  layout = "vertical",
  calcite_tile(
    id = "construction",
    icon = "wrench",
    heading = "Construction Worker",
    description = "Manage construction projects and coordinate teams"
  ),
  calcite_tile(
    id = "engineer",
    icon = "rangefinder",
    heading = "Civil Engineer",
    description = "Design infrastructure and ensure compliance"
```

```
)
)
```

calcite\_time\_picker *Create a TimePicker component*

**Description**

Create a TimePicker component

**Usage**

```
calcite_time_picker(...)
```

**Arguments**

```
...          named attributes passed to htmltools::tag()
```

**Details**

**Properties:**

The following properties are provided by this component:

Name	Attribute	Description
hourFormat	hour-format	Specifies the component's hour format, where: "user" displays the user's locale fo
messageOverrides	NA	Use this property to override individual strings used by the component.
numberingSystem	numbering-system	Specifies the Unicode numeral system used by the component for localization.
scale	scale	Specifies the size of the component.
step	step	Specifies the granularity the value must adhere to (in seconds).
value	value	The component's value in UTC (always 24-hour format).

**Events:**

The following events are observed by shiny:

Event	Description
calciteTimePickerChange	

**Value**

an object of class calcite\_component which is a subclass of shiny.tag

**References**

[Official Documentation](#)

**Examples**

```
calcite_time_picker()
```

---

 calcite\_tip

 Create a Tip component
 

---

### Description

Use the `calcite-card`, `calcite-notice`, `calcite-panel`, or `calcite-tile` component instead.

### Usage

```
calcite_tip(...)
```

### Arguments

```
...          named attributes passed to htmltools::tag()
```

### Details

#### Properties:

The following properties are provided by this component:

Name	Attribute	Description
<code>closed</code>	<code>closed</code>	When true, the component does not display.
<code>closeDisabled</code>	<code>close-disabled</code>	When true, the close button is not present on the component.
<code>heading</code>	<code>heading</code>	The component header text.
<code>headingLevel</code>	<code>heading-level</code>	Specifies the heading level of the component's heading for proper document structure.
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.
<code>selected</code>	<code>selected</code>	When true, the component is selected if it has a parent <code>calcite-tip-manager</code> . Only o

#### Events:

The following events are observed by shiny:

Event	Description
<code>calciteTipDismiss</code>	Emits when the component has been closed.

#### Slots:

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding text and a hyperlink.
<code>thumbnail</code>	A slot for adding an HTML image element.

### Value

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**[Official Documentation](#)**Examples**`calcite_tip()`


---

<code>calcite_tip_group</code>	<i>Create a TipGroup component</i>
--------------------------------	------------------------------------

---

**Description**

Use the `calcite-carousel` and `calcite-carousel-item` components instead.

**Usage**`calcite_tip_group(...)`**Arguments**`...` named attributes passed to `htmltools::tag()`**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description	Values	Reflects to Attribute
<code>groupTitle</code>	<code>group-title</code>	The component header text for all nested <code>calcite-tips</code> .	string	FALSE

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding <code>calcite-tips</code> .

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**[Official Documentation](#)**Examples**`calcite_tip_group()`

---

calcite\_tip\_manager    *Create a TipManager component*

---

### Description

Use the `calcite-carousel` and `calcite-carousel-item` components instead.

### Usage

```
calcite_tip_manager(...)
```

### Arguments

...                    named attributes passed to `htmltools::tag()`

### Details

#### Properties:

The following properties are provided by this component:

Name	Attribute	Description
<code>closed</code>	<code>closed</code>	When <code>true</code> , does not display or position the component.
<code>headingLevel</code>	<code>heading-level</code>	Specifies the heading level of the component's heading for proper document structure, v
<code>messageOverrides</code>	NA	Use this property to override individual strings used by the component.

#### Events:

The following events are observed by shiny:

Event	Description
<code>calciteTipManagerClose</code>	Emits when the component has been closed.

#### Slots:

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding <code>calcite-tips</code> .

### Value

an object of class `calcite_component` which is a subclass of `shiny.tag`

### References

[Official Documentation](#)

**Examples**

```
calcite_tip_manager()
```

---

calcite_tooltip	<i>Create a Tooltip component</i>
-----------------	-----------------------------------

---

**Description**

Create a Tooltip component

**Usage**

```
calcite_tooltip(...)
```

**Arguments**

...                    named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
<code>closeOnClick</code>	<code>close-on-click</code>	Closes the component when the <code>referenceElement</code> is clicked.
<code>label</code>	<code>label</code>	Accessible name for the component.
<code>offsetDistance</code>	<code>offset-distance</code>	Offset the position of the component away from the <code>referenceElement</code> .
<code>offsetSkidding</code>	<code>offset-skidding</code>	Offset the position of the component along the <code>referenceElement</code> .
<code>open</code>	<code>open</code>	When true, the component is open.
<code>overlayPositioning</code>	<code>overlay-positioning</code>	Determines the type of positioning to use for the overlaid content. Using "absolute" will position the component relative to the <code>referenceElement</code> .
<code>placement</code>	<code>placement</code>	Determines where the component will be positioned relative to the <code>referenceElement</code> .
<code>referenceElement</code>	<code>reference-element</code>	The <code>referenceElement</code> to position the component according to its "placement".

**Events:**

The following events are observed by shiny:

Event	Description
<code>calciteTooltipBeforeClose</code>	Fires when the component is requested to be closed and before the closing transition begins.
<code>calciteTooltipBeforeOpen</code>	Fires when the component is added to the DOM but not rendered, and before the opening transition begins.
<code>calciteTooltipClose</code>	Fires when the component is closed and animation is complete.
<code>calciteTooltipOpen</code>	Fires when the component is open and animation is complete.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding text.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_tooltip()
```

---

calcite_tree	<i>Create a Tree component</i>
--------------	--------------------------------

---

**Description**

Create a Tree component

**Usage**

```
calcite_tree(...)
```

**Arguments**

...            named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description
lines	lines	When true, displays indentation guide lines.
scale	scale	Specifies the size of the component.
selectedItems	NA	Specifies the component's selected items.
selectionMode	selection-mode	Specifies the selection mode of the component, where: "ancestors" displays with a check

**Events:**

The following events are observed by shiny:

Event	Description
calciteTreeSelect	Fires when the user selects/deselects calcite-tree-items.

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for calcite-tree-item elements.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_tree()
```

---

calcite\_tree\_item      *Create a TreeItem component*

---

**Description**

Create a TreeItem component

**Usage**

```
calcite_tree_item(...)
```

**Arguments**

...                  named attributes passed to `htmltools::tag()`

**Details****Properties:**

The following properties are provided by this component:

Name	Attribute	Description	Value
disabled	disabled	When <code>true</code> , interaction is prevented and the component is displayed with lower opacity.	boolean
expanded	expanded	When <code>true</code> , the component is expanded.	boolean
iconFlipRtl	icon-flip-rtl	When <code>true</code> , the icon will be flipped when the element direction is right-to-left (" <code>rtl</code> ").	"both"
iconStart	icon-start	Specifies an icon to display at the start of the component.	string
label	label	Accessible name for the component.	string
selected	selected	When <code>true</code> , the component is selected.	boolean

**Slots:**

The following slots are provided by this component:

Slot	Description
Default (unnamed)	A slot for adding text.
children	A slot for adding nested <code>calcite-tree</code> elements.
actions-end	A slot for adding actions to the end of the component. It is recommended to use two or fewer actions.

**Value**

an object of class `calcite_component` which is a subclass of `shiny.tag`

**References**

[Official Documentation](#)

**Examples**

```
calcite_tree_item()
```

---

calcite_version	<i>Calcite components version</i>
-----------------	-----------------------------------

---

**Description**

Reports the currently used calcite component version.

**Usage**

```
calcite_version()
```

**Value**

a character scalar

**Examples**

```
calcite_version()
```

---

examples

*Calcite examples*

---

**Description**

Calcite examples

**Usage**

```
list_examples()
```

```
run_example(name)
```

```
open_example(name = NULL)
```

**Arguments**

name            the example name (without .R extension). See [list\\_examples\(\)](#).

**Examples**

```
list_examples()

if (interactive()) {
  open_example()
  run_example("calcite-checkbox")
}
```

---

page\_actionbar

*Create a Shell with Action Bar Layout*

---

**Description**

A convenience function that creates a common layout pattern: a shell with a navigation header, an action bar, and a side panel. This is ideal for map-based applications or tools with multiple layers/options.

**Usage**

```

page_actionbar(
  ...,
  title = NULL,
  header_actions = NULL,
  actions = NULL,
  panel_content = NULL,
  panel_position = c("start", "end"),
  panel_width = c("m", "s", "l"),
  footer = NULL
)

```

**Arguments**

...	Main content area (typically a map or primary view)
title	Application title shown in the navigation header
header_actions	Optional actions for the header (e.g., user menu, settings). These will be placed in the content-end slot of the navigation.
actions	A <code>calcite_action_bar()</code> component to slot alongside the panel.
panel_content	A <code>calcite_panel()</code> shown when an action is selected.
panel_position	Position of the panel: "start" (left) or "end" (right).
panel_width	Width of the shell panel: "s", "m", or "l".
footer	Optional footer content

**Value**

An object of class `calcite_component` containing a configured shell

**Examples**

```

page_actionbar(
  title = "Wildlife Areas",
  actions = calcite_action_bar(
    calcite_action(text = "Add", icon = "plus"),
    calcite_action(text = "Layers", icon = "layers", active = TRUE)
  ),
  panel_content = calcite_panel(
    heading = "Layers",
    "Layer controls here"
  ),
  "Map content here"
)

```

---

`page_navbar`*Create a Shell with Navigation Bar Layout*

---

**Description**

A convenience function for creating a simple layout with a navigation header and main content area. This is ideal for dashboard-style applications.

**Usage**

```
page_navbar(..., title = NULL, logo = NULL, nav_actions = NULL, footer = NULL)
```

**Arguments**

<code>...</code>	Main content area
<code>title</code>	Application title shown in the navigation header
<code>logo</code>	Optional logo component for the navigation
<code>nav_actions</code>	Optional actions for the navigation bar (placed in content-end)
<code>footer</code>	Optional footer content

**Value**

An object of class `calcite_component` containing a configured shell

**Examples**

```
page_navbar(  
  title = "My Dashboard",  
  nav_actions = list(  
    calcite_action(icon = "gear", text = "Settings"),  
    calcite_action(icon = "sign-out", text = "Logout")  
  ),  
  "Dashboard content here"  
)
```

---

`page_sidebar`*Create a Shell with Sidebar Panel Layout*

---

**Description**

Similar to `bslib::page_sidebar()`, this creates a layout with a sidebar panel and main content area — the easiest way to build a standard Calcite app layout. Pass a `calcite_panel()` as sidebar; it will be wrapped in a `calcite_shell_panel()` automatically.

**Usage**

```

page_sidebar(
  ...,
  sidebar = NULL,
  title = NULL,
  position = c("start", "end"),
  width = c("m", "s", "l"),
  display_mode = c("dock", "overlay", "float-content", "float-all"),
  footer = NULL
)

```

**Arguments**

...	Main content area (typically a <code>calcite_panel()</code> or map view)
sidebar	A <code>calcite_panel()</code> for the sidebar. Typically contains <code>calcite_block()</code> components with controls.
title	Optional application title shown in a navigation header.
position	Position of sidebar: "start" (left) or "end" (right).
width	Width of the sidebar: "s", "m", or "l".
display_mode	Display mode for the shell panel: "dock", "overlay", "float-content", or "float-all". Default: "dock".
footer	Optional footer content

**Value**

An object of class `calcite_component` containing a configured shell

**Examples**

```

page_sidebar(
  title = "Data Explorer",
  sidebar = calcite_panel(
    heading = "Filters",
    calcite_block(
      heading = "Options",
      collapsible = TRUE,
      expanded = TRUE
    )
  ),
  calcite_panel(heading = "Results")
)

```

---

update_calcite	<i>Update Calcite Component Properties</i>
----------------	--------------------------------------------

---

**Description**

Updates the properties of the javascript Calcite component.

**Usage**

```
update_calcite(id, ..., session = shiny::getDefaultReactiveDomain())
```

**Arguments**

id	the html ID of the element to update. Must be a character scalar.
...	named properties to be updated in the component. The names must match the property name. Otherwise, will fail silently.
session	a shiny session object. Default <code>shiny::getDefaultReactiveDomain()</code> .

**Value**

NULL. Sends a message to the shiny session object.

**Examples**

```
# this cannot work outside of shiny
if (interactive()) {
  library(shiny)
  ui <- calcite_shell(
    calcite_card(
      heading = "Content",
      calcite_label(
        layout = "inline",
        calcite_checkbox(id = "checked"),
        "Click me"
      )
    ),
    calcite_notice(
      id = "initial-note",
      div(slot = "title", "Nice!"),
      div(slot = "message", "This is a success message")
    )
  )

  server <- function(input, output, session) {
    observeEvent(input$checked_checked, {
      checked <- input$checked_checked$values
      # Update the `initial-note` property here
      update_calcite("initial-note", open = checked)
    })
  }
}
```

```
}  
# shinyApp(ui, server)  
}
```

# Index

`bslib::page_sidebar()`, [151](#)

`calcite_accordion`, [4](#)  
`calcite_accordion_item`, [6](#)  
`calcite_accordion_item()`, [4](#)  
`calcite_action`, [8](#)  
`calcite_action()`, [10](#), [11](#), [18](#)  
`calcite_action_bar`, [10](#)  
`calcite_action_bar()`, [96](#), [115](#), [116](#), [150](#)  
`calcite_action_group`, [12](#)  
`calcite_action_group()`, [11](#)  
`calcite_action_pad`, [13](#)  
`calcite_alert`, [15](#)  
`calcite_alert()`, [96](#), [113](#)  
`calcite_alert_brand` (`calcite_alert`), [15](#)  
`calcite_alert_danger` (`calcite_alert`), [15](#)  
`calcite_alert_info` (`calcite_alert`), [15](#)  
`calcite_alert_success` (`calcite_alert`), [15](#)  
`calcite_alert_warning` (`calcite_alert`), [15](#)

`calcite_avatar`, [19](#)  
`calcite_block`, [20](#)  
`calcite_block()`, [115](#), [152](#)  
`calcite_block_section`, [22](#)  
`calcite_button`, [24](#)  
`calcite_card`, [27](#)  
`calcite_card_group`, [28](#)  
`calcite_carousel`, [29](#)  
`calcite_carousel_item`, [31](#)  
`calcite_checkbox`, [32](#)  
`calcite_chip`, [33](#)  
`calcite_chip_group`, [35](#)  
`calcite_color_picker`, [36](#)  
`calcite_combobox`, [37](#)  
`calcite_combobox_item`, [39](#)  
`calcite_combobox_item_group`, [40](#)  
`calcite_date_picker`, [41](#)  
`calcite_dialog`, [43](#)  
`calcite_dialog()`, [113](#)

`calcite_dropdown`, [45](#)  
`calcite_dropdown()`, [96](#)  
`calcite_dropdown_group`, [47](#)  
`calcite_dropdown_item`, [48](#)  
`calcite_fab`, [49](#)  
`calcite_fab()`, [96](#)  
`calcite_filter`, [50](#)  
`calcite_flow`, [51](#)  
`calcite_flow_item`, [52](#)  
`calcite_icon`, [53](#)  
`calcite_inline_editable`, [54](#)  
`calcite_input_date_picker`, [56](#)  
`calcite_input_file`, [57](#)  
`calcite_input_message`, [59](#)  
`calcite_input_number`, [61](#)  
`calcite_input_text`, [65](#)  
`calcite_input_time_picker`, [68](#)  
`calcite_input_time_zone`, [69](#)  
`calcite_label`, [70](#)  
`calcite_link`, [72](#)  
`calcite_link()`, [18](#)  
`calcite_list`, [74](#)  
`calcite_list_item`, [76](#)  
`calcite_list_item_group`, [77](#)  
`calcite_loader`, [78](#)  
`calcite_menu`, [79](#)  
`calcite_menu_item`, [80](#)  
`calcite_meter`, [81](#)  
`calcite_modal`, [83](#)  
`calcite_modal()`, [113](#)  
`calcite_navigation`, [84](#)  
`calcite_navigation()`, [113](#)  
`calcite_navigation_logo`, [85](#)  
`calcite_navigation_user`, [86](#)  
`calcite_notice`, [87](#)  
`calcite_option`, [91](#)  
`calcite_option()`, [109](#)  
`calcite_option_group`, [92](#)  
`calcite_pagination`, [93](#)

calcite\_panel, 94  
calcite\_panel(), 113, 115, 116, 150–152  
calcite\_popover, 98  
calcite\_progress, 99  
calcite\_radio\_button, 100  
calcite\_radio\_button\_group, 101  
calcite\_rating, 102  
calcite\_scrim, 104  
calcite\_segmented\_control, 105  
calcite\_segmented\_control\_item, 107  
calcite\_select, 109  
calcite\_select(), 91  
calcite\_sheet, 111  
calcite\_sheet(), 113, 116  
calcite\_shell, 113  
calcite\_shell(), 115, 116  
calcite\_shell\_center\_row, 114  
calcite\_shell\_panel, 115  
calcite\_shell\_panel(), 113, 151  
calcite\_slider, 117  
calcite\_split\_button, 121  
calcite\_stepper, 123  
calcite\_stepper\_item, 124  
calcite\_switch, 125  
calcite\_tab, 128  
calcite\_tab\_nav, 132  
calcite\_tab\_title, 133  
calcite\_table, 129  
calcite\_table\_header, 131  
calcite\_table\_header(), 129  
calcite\_tabs, 131  
calcite\_text\_area, 135  
calcite\_tile, 136  
calcite\_tile\_group, 138  
calcite\_time\_picker, 141  
calcite\_tip, 142  
calcite\_tip\_group, 143  
calcite\_tip\_manager, 144  
calcite\_tooltip, 145  
calcite\_tree, 146  
calcite\_tree\_item, 147  
calcite\_version, 148

Deprecated, 28, 45, 53

examples, 149

list\_examples (examples), 149  
list\_examples(), 149

open\_example (examples), 149

page\_actionbar, 149  
page\_navbar, 151  
page\_sidebar, 151

run\_example (examples), 149

shiny::fileInput(), 58  
shiny::getDefaultReactiveDomain(), 153

update\_calcite, 153