Package: shiny.webawesome (via r-universe)

October 20, 2026

Title 'shiny' Bindings for 'Web Awesome' Components

Version 1.0.0.9000

Description Provides an 'R' and 'shiny' interface to the 'Web Awesome' component library. The package is generator-driven, exposing 'Web Awesome' web components as 'R' functions that produce HTML tags and integrate with the reactive model that 'shiny' uses.

URL https://github.com/mbanand/shiny.webawesome, https://www.shiny-webawesome.org

BugReports https://github.com/mbanand/shiny.webawesome/issues

License MIT + file LICENSE

Encoding UTF-8

Imports htmltools, shiny

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests chromote, covr, devtools, digest, document, jsonlite, knitr, lintr, processx, pkgdown, rmarkdown, rhub, shinytest2, styler, testthat (>= 3.0.0), urlchecker, withr, yaml

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

Repository https://mbanand.r-universe.dev

Date/Publication 2026-04-23 18:45:16 UTC

RemoteUrl https://github.com/mbanand/shiny.webawesome

RemoteRef HEAD

RemoteSha 3cffb090a795776f66efc1341fac66c49c056b33

Contents

wa_animated_image . . . 3
wa_animation . . . 4
wa_avatar . . . 6
wa_badge . . . 7
wa_breadcrumb . . . 8
wa_breadcrumb_item . . . 9
wa_button . . . 10
wa_button_group . . . 13
wa_call_method . . . 14
wa_callout . . . 15
wa_card . . . 17
wa_carousel . . . 18
wa_carousel_item . . . 20
wa_checkbox . . . 21
wa_color_picker . . . 22
wa_comparison . . . 25
wa_container . . . 26
wa_copy_button . . . 27
wa_details . . . 28
wa_dialog . . . 30
wa_divider . . . 31
wa_drawer . . . 32
wa_dropdown . . . 33
wa_dropdown_item . . . 35
wa_format_bytes . . . 36
wa_format_date . . . 37
wa_format_number . . . 39
wa_icon . . . 40
wa_include . . . 42
wa_input . . . 43
wa_intersection_observer . . . 46
wa_js . . . 48
wa_markdown . . . 50
wa_mutation_observer . . . 51
wa_number_input . . . 52
wa_option . . . 55
wa_page . . . 56
wa_popover . . . 58
wa_popup . . . 60
wa_progress_bar . . . 63
wa_progress_ring . . . 64
wa_qr_code . . . 65
wa_radio . . . 66
wa_radio_group . . . 67
wa_rating . . . 69
wa_relative_time . . . 70

wa_resize_observer 72
wa_scroller 73
wa_select 74
wa_set_property 76
wa_skeleton 78
wa_slider 79
wa_spinner 81
wa_split_panel 82
wa_switch 84
wa_tab 85
wa_tab_group 86
wa_tab_panel 87
wa_tag 88
wa_textarea 90
wa_tooltip 93
wa_tree 94
wa_tree_item 95
wa_version 96
wa_zoomable_frame 97
webawesomePage 98

Index **100**

`wa_animated_image` *Create a* `wa-animated-image` *component*

Description

Generated wrapper for the Web Awesome `wa-animated-image` component. Generated from Web Awesome metadata.

Usage

```
wa_animated_image(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  alt = NULL,
  dir = NULL,
  lang = NULL,
  play = NULL,
  src = NULL,
  pause_icon = NULL,
  play_icon = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
alt	String. A description of the image used by assistive devices.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
play	Boolean. Default: FALSE. Plays the animation. When this attribute is remove, the animation will pause.
src	String. The path to the image to load.
pause_icon	Optional pause icon to use instead of the default. Works best with <wa-icon>.
play_icon	Optional play icon to use instead of the default. Works best with <wa-icon>.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_animation *Create a* wa-animation *component*

Description

Generated wrapper for the Web Awesome wa-animation component. Generated from Web Awesome metadata.

Usage

```
wa_animation(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  name = NULL,
  delay = NULL,
  dir = NULL,
  direction = NULL,
  duration = NULL,
  easing = NULL,
  end_delay = NULL,
```

```
  fill = NULL,
  iteration_start = NULL,
  iterations = NULL,
  lang = NULL,
  play = NULL,
  playback_rate = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
name	String. Default: none. The name of the built-in animation to use. For custom animations, use the keyframes prop.
delay	Number. Default: 0. The number of milliseconds to delay the start of the animation.
dir	String. Optional Web Awesome attribute.
direction	Default: 'normal'. Determines the direction of playback as well as the behavior when reaching the end of an iteration. Learn more
duration	Number. Default: 1000. The number of milliseconds each iteration of the animation takes to complete.
easing	String. Default: linear. The easing function to use for the animation. This can be a Web Awesome easing function or a custom easing function such as cubic-bezier(0, 1, .76, 1.14).
end_delay	Number. Default: 0. The number of milliseconds to delay after the active period of an animation sequence.
fill	Default: 'auto'. Sets how the animation applies styles to its target before and after its execution.
iteration_start	Number. Default: 0. The offset at which to start the animation, usually between 0 (start) and 1 (end).
iterations	Number. Default: Infinity. The number of iterations to run before the animation completes. Defaults to Infinity, which loops.
lang	String. Optional Web Awesome attribute.
play	Boolean. Default: FALSE. Plays the animation. When omitted, the animation will be paused. This attribute will be automatically removed when the animation finishes or gets canceled.
playback_rate	Number. Default: 1. Sets the animation's playback rate. The default is 1, which plays the animation at a normal speed. Setting this to 2, for example, will double the animation's speed. A negative value can be used to reverse the animation. This value can be changed without causing the animation to restart.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_avatar` *Create a* `wa-avatar` *component*

Description

Generated wrapper for the Web Awesome `wa-avatar` component. Generated from Web Awesome metadata.

Usage

```
wa_avatar(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  label = NULL,
  dir = NULL,
  image = NULL,
  initials = NULL,
  lang = NULL,
  loading = NULL,
  shape = NULL,
  icon = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`label`	String. Default: `""`. A label to use to describe the avatar to assistive devices.
`dir`	String. Optional Web Awesome attribute.
`image`	String. Default: `""`. The image source to use for the avatar.
`initials`	String. Default: `""`. Initials to use as a fallback when no image is available (1-2 characters max recommended).
`lang`	String. Optional Web Awesome attribute.

loading	Enumerated string. Allowed values: eager, lazy. Default: eager. Indicates how the browser should load the image.
shape	Enumerated string. Allowed values: circle, rounded, square. Default: circle. The shape of the avatar.
icon	The default icon to use when no image or initials are present. Works best with <wa-icon>.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_badge *Create a* wa-badge *component*

Description

Generated wrapper for the Web Awesome wa-badge component. Generated from Web Awesome metadata.

Usage

```
wa_badge(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  appearance = NULL,
  attention = NULL,
  dir = NULL,
  lang = NULL,
  pill = NULL,
  variant = NULL,
  end = NULL,
  start = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.

appearance	Enumerated string. Allowed values: accent, filled, filled-outlined, outlined. Default: accent. The badge's visual appearance.
attention	Enumerated string. Allowed values: bounce, none, pulse. Default: none. Adds an animation to draw attention to the badge.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
pill	Boolean. Default: FALSE. Draws a pill-style badge with rounded edges.
variant	Enumerated string. Allowed values: brand, danger, neutral, success, warning. Default: brand. The badge's theme variant. Defaults to brand if not within another element with a variant.
end	An element, such as <wa-icon>, placed after the label.
start	An element, such as <wa-icon>, placed before the label.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_breadcrumb *Create a* wa-breadcrumb *component*

Description

Generated wrapper for the Web Awesome wa-breadcrumb component. Generated from Web Awesome metadata.

Usage

```
wa_breadcrumb(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  label = NULL,
  dir = NULL,
  lang = NULL,
  separator = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
label	String. Default: "". The label to use for the breadcrumb control. This will not be shown on the screen, but it will be announced by screen readers and other assistive devices to provide more context for users.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
separator	The separator to use between breadcrumb items. Works best with <wa-icon>.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_breadcrumb_item *Create a* wa-breadcrumb-item *component*

Description

Generated wrapper for the Web Awesome wa-breadcrumb-item component. Generated from Web Awesome metadata.

Usage

```
wa_breadcrumb_item(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  href = NULL,
  lang = NULL,
  rel = NULL,
  target = NULL,
  end = NULL,
  separator = NULL,
  start = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
href	String. Optional URL to direct the user to when the breadcrumb item is activated. When set, a link will be rendered internally. When unset, a button will be rendered instead.
lang	String. Optional Web Awesome attribute.
rel	String. Default: `noreferrer noopener`. The `rel` attribute to use on the link. Only used when `href` is set.
target	Enumerated string. Allowed values: `_blank`, `_parent`, `_self`, `_top`. Tells the browser where to open the link. Only used when `href` is set.
end	An element, such as `<wa-icon>`, placed after the label.
separator	The separator to use for the breadcrumb item. This will only change the separator for this item. If you want to change it for all items in the group, set the separator on `<wa-breadcrumb>` instead.
start	An element, such as `<wa-icon>`, placed before the label.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_button *Create a* `wa-button` *component*

Description

Generated wrapper for the Web Awesome `wa-button` component. When used as a Shiny input, the component exposes action semantics through `input$<input_id>`. The Shiny input invalidates on each committed action rather than publishing a durable value payload. Each committed action publishes a numeric action value. Generated from Web Awesome metadata.

Usage

```
wa_button(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  name = NULL,
  appearance = NULL,
  custom_error = NULL,
  dir = NULL,
  download = NULL,
  formaction = NULL,
  formenctype = NULL,
  formmethod = NULL,
  formnovalidate = NULL,
  formtarget = NULL,
  href = NULL,
  lang = NULL,
  loading = NULL,
  pill = NULL,
  rel = NULL,
  size = NULL,
  target = NULL,
  title = NULL,
  type = NULL,
  variant = NULL,
  with_caret = NULL,
  with_end = NULL,
  with_start = NULL,
  end = NULL,
  start = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. The value of the button, submitted as a pair with the button's name as part of the form data, but only when this button is the submitter. This attribute is ignored when href is present.
disabled	Boolean. Default: FALSE. Disables the button.

name	String. Default: null. The name of the button, submitted as a name/value pair with form data, but only when this button is the submitter. This attribute is ignored when href is present.
appearance	Enumerated string. Allowed values: accent, filled, filled-outlined, outlined, plain. Default: accent. The button's visual appearance.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
download	String. Tells the browser to download the linked file as this filename. Only used when href is present.
formaction	String. Used to override the form owner's action attribute. This wrapper argument sets the HTML formaction attribute, which maps to the component's formAction field/property.
formenctype	Enumerated string. Allowed values: application/x-www-form-urlencoded, multipart/form-data, text/plain. Used to override the form owner's enctype attribute. This wrapper argument sets the HTML formenctype attribute, which maps to the component's formEnctype field/property.
formmethod	Enumerated string. Allowed values: get, post. Used to override the form owner's method attribute. This wrapper argument sets the HTML formmethod attribute, which maps to the component's formMethod field/property.
formnovalidate	Boolean. Default: FALSE. Used to override the form owner's novalidate attribute. This wrapper argument sets the HTML formnovalidate attribute, which maps to the component's formNoValidate field/property.
formtarget	Used to override the form owner's target attribute. This wrapper argument sets the HTML formtarget attribute, which maps to the component's formTarget field/property.
href	String. When set, the underlying button will be rendered as an <a> with this href instead of a <button>.
lang	String. Optional Web Awesome attribute.
loading	Boolean. Default: FALSE. Draws the button in a loading state.
pill	Boolean. Default: FALSE. Draws a pill-style button with rounded edges.
rel	String. When using href, this attribute will map to the underlying link's rel attribute.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. The button's size.
target	Enumerated string. Allowed values: _blank, _parent, _self, _top. Tells the browser where to open the link. Only used when href is present.
title	String. Default: "". Optional Web Awesome attribute.
type	Enumerated string. Allowed values: button, reset, submit. Default: button. The type of button. Note that the default value is button instead of submit, which is opposite of how native <button> elements behave. When the type is submit, the button will submit the surrounding form.
variant	Enumerated string. Allowed values: brand, danger, neutral, success, warning. Default: neutral. The button's theme variant. Defaults to neutral if not within another element with a variant.

with_caret	Boolean. Default: FALSE. Draws the button with a caret. Used to indicate that the button triggers a dropdown menu or similar behavior.
with_end	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in an end element so the server-rendered markup includes the end slot before the component hydrates on the client.
with_start	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a start element so the server-rendered markup includes the start slot before the component hydrates on the client.
end	An element, such as <wa-icon>, placed after the label.
start	An element, such as <wa-icon>, placed before the label.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> uses action semantics and invalidates on each committed action rather than publishing a durable value payload. The Shiny action value is returned as a numeric action value.

---

wa_button_group *Create a* wa-button-group *component*

Description

Generated wrapper for the Web Awesome wa-button-group component. Generated from Web Awesome metadata.

Usage

```
wa_button_group(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  label = NULL,
  dir = NULL,
  lang = NULL,
  orientation = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.

style	Optional inline CSS style string.
label	String. Default: "". A label to use for the button group. This won't be displayed on the screen, but it will be announced by assistive devices when interacting with the control and is strongly recommended.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
orientation	Enumerated string. Allowed values: horizontal, vertical. Default: horizontal. The button group's orientation.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_call_method *Call one browser-side method on a target element*

Description

Sends a one-way command from the Shiny server that invokes a browser-side method on the element identified by DOM id.

Usage

```
wa_call_method(
  id,
  method,
  args = list(),
  session = shiny::getDefaultReactiveDomain()
)
```

Arguments

id	DOM id of the target browser element.
method	Scalar method name to invoke on the target element.
args	Optional list of positional arguments to pass to the method. These should be serializable through Shiny's custom-message transport. In practice, prefer JSON-like scalar values or nested lists that map cleanly to browser values. Do not expect R functions, language objects, or HTML tags to serialize into useful method arguments.
session	Shiny session object. Defaults to the current reactive domain.

Details

wa_call_method() is a narrow package-level escape hatch for advanced cases where a Web Awesome component method needs to be triggered from server logic but is not covered by a generated helper. It does not validate whether the requested method exists for the targeted component.

This helper is complementary to generated component bindings and update helpers. It is not part of upstream component coverage and does not expand the generated per-component API surface.

On the server side, wa_call_method() validates only its R helper inputs, such as the target id, method name, argument list, and session. It does not validate whether the requested method exists on the browser-side element.

In the browser runtime, the command layer validates that the target DOM id resolves to an element, that a method name was supplied, and that the named member is callable on the target element before invoking it. Command-layer warnings are controlled by the package warning registry, especially the command_layer key. For option details, see the Package Options article.

Value

Invisibly returns NULL.

Examples

```
library(shiny)
library(shiny.webawesome)

ui <- webawesomePage(
  wa_button("Show dialog", id = "show_dialog"),
  wa_dialog("Dialog body", id = "dialog")
)

server <- function(input, output, session) {
  observeEvent(input$show_dialog, {
    wa_call_method("dialog", "show", session = session)
  })
}

app <- shinyApp(ui, server)
stopifnot(inherits(app, "shiny.appobj"))

# Run `app` interactively to launch this example application.
```

---

wa_callout *Create a* wa-callout *component*

Description

Generated wrapper for the Web Awesome wa-callout component. Generated from Web Awesome metadata.

Usage

```
wa_callout(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  appearance = NULL,
  dir = NULL,
  lang = NULL,
  size = NULL,
  variant = NULL,
  icon = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
appearance	Enumerated string. Allowed values: `accent`, `filled`, `filled-outlined`, `outlined`, `plain`. The callout's visual appearance.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
size	Enumerated string. Allowed values: `large`, `medium`, `small`. Default: `medium`. The callout's size.
variant	Enumerated string. Allowed values: `brand`, `danger`, `neutral`, `success`, `warning`. Default: `brand`. The callout's theme variant. Defaults to `brand` if not within another element with a variant.
icon	An icon to show in the callout. Works best with `<wa-icon>`.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_card *Create a* wa-card *component*

Description

Generated wrapper for the Web Awesome wa-card component. Generated from Web Awesome metadata.

Usage

```
wa_card(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  appearance = NULL,
  dir = NULL,
  lang = NULL,
  orientation = NULL,
  with_footer = NULL,
  with_header = NULL,
  with_media = NULL,
  actions = NULL,
  footer = NULL,
  footer_actions = NULL,
  header = NULL,
  header_actions = NULL,
  media = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
appearance	Enumerated string. Allowed values: accent, filled, filled-outlined, outlined, plain. Default: outlined. The card's visual appearance.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
orientation	Enumerated string. Allowed values: horizontal, vertical. Default: vertical. Renders the card's orientation *
with_footer	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a footer element so the server-rendered markup includes the footer before the component hydrates on the client.

with_header	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a header element so the server-rendered markup includes the header before the component hydrates on the client.
with_media	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a media element so the server-rendered markup includes the media before the component hydrates on the client.
actions	An optional actions section to render at the end for the horizontal card.
footer	An optional footer for the card.
footer_actions	An optional actions section to render in the footer of the vertical card.
header	An optional header for the card.
header_actions	An optional actions section to render in the header of the vertical card.
media	An optional media section to render at the start of the card.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_carousel` *Create a* `wa-carousel` *component*

Description

Generated wrapper for the Web Awesome `wa-carousel` component. When used as a Shiny input, `input$<input_id>` reflects the component's current semantic `activeSlide` state. The Shiny value is returned as a numeric value. This index is 0-based. Generated from Web Awesome metadata.

Usage

```
wa_carousel(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  autoplay = NULL,
  autoplay_interval = NULL,
  current_slide = NULL,
  dir = NULL,
  lang = NULL,
  loop = NULL,
  mouse_dragging = NULL,
  navigation = NULL,
```

```
  orientation = NULL,
  pagination = NULL,
  slides = NULL,
  slides_per_move = NULL,
  slides_per_page = NULL,
  next_icon = NULL,
  previous_icon = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
autoplay	Boolean. Default: FALSE. When set, the slides will scroll automatically when the user is not interacting with them.
autoplay_interval	Number. Default: 3000. Specifies the amount of time, in milliseconds, between each automatic scroll.
current_slide	Number. Default: 0. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
loop	Boolean. Default: FALSE. When set, allows the user to navigate the carousel in the same direction indefinitely.
mouse_dragging	Boolean. Default: FALSE. When set, it is possible to scroll through the slides by dragging them with the mouse.
navigation	Boolean. Default: FALSE. When set, show the carousel's navigation.
orientation	Enumerated string. Allowed values: horizontal, vertical. Default: horizontal. Specifies the orientation in which the carousel will lay out.
pagination	Boolean. Default: FALSE. When set, show the carousel's pagination indicators.
slides	Number. Default: 0. Optional Web Awesome attribute.
slides_per_move	Number. Default: 1. Specifies the number of slides the carousel will advance when scrolling, useful when specifying a slides-per-page greater than one. It can't be higher than slides-per-page.
slides_per_page	Number. Default: 1. Specifies how many slides should be shown at a given time.
next_icon	Optional next icon to use instead of the default. Works best with <wa-icon>.
previous_icon	Optional previous icon to use instead of the default. Works best with <wa-icon>.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current semantic activeSlide state. The Shiny value is returned as a numeric value. This index is 0-based.

`wa_carousel_item` *Create a* `wa-carousel-item` *component*

Description

Generated wrapper for the Web Awesome `wa-carousel-item` component. Generated from Web Awesome metadata.

Usage

```
wa_carousel_item(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  lang = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`dir`	String. Optional Web Awesome attribute.
`lang`	String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_checkbox *Create a* wa-checkbox *component*

Description

Generated wrapper for the Web Awesome wa-checkbox component. When used as a Shiny input, input$<input_id> reflects the component's current value value. The Shiny value is returned as a logical value. Generated from Web Awesome metadata.

Usage

```
wa_checkbox(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  checked = NULL,
  disabled = NULL,
  hint = NULL,
  name = NULL,
  custom_error = NULL,
  dir = NULL,
  indeterminate = NULL,
  lang = NULL,
  required = NULL,
  size = NULL,
  title = NULL,
  hint_slot = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. The value of the checkbox, submitted as a name/value pair with form data.
checked	Boolean. Default: FALSE. The default value of the form control. Primarily used for resetting the form control. This wrapper argument sets the HTML checked attribute, which maps to the component's defaultChecked field/property rather than its live checked property.
disabled	Boolean. Default: FALSE. Disables the checkbox.

hint	String. Default: "". The checkbox's hint. If you need to display HTML, use the hint slot instead.
name	String. Default: null. The name of the checkbox, submitted as a name/value pair with form data.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
indeterminate	Boolean. Default: FALSE. Draws the checkbox in an indeterminate state. This is usually applied to checkboxes that represents a "select all/none" behavior when associated checkboxes have a mix of checked and unchecked states.
lang	String. Optional Web Awesome attribute.
required	Boolean. Default: FALSE. Makes the checkbox a required field.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. The checkbox's size.
title	String. Default: "". Optional Web Awesome attribute.
hint_slot	Text that describes how to use the checkbox. Alternatively, you can use the hint attribute.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current value value. The Shiny value is returned as a logical value.

---

wa_color_picker *Create a* wa-color-picker *component*

Description

Generated wrapper for the Web Awesome wa-color-picker component. When used as a Shiny input, input$<input_id> reflects the component's current value value. Generated from Web Awesome metadata.

Usage

```
wa_color_picker(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
```

```
  hint = NULL,
  name = NULL,
  custom_error = NULL,
  dir = NULL,
  format = NULL,
  lang = NULL,
  opacity = NULL,
  open = NULL,
  placement = NULL,
  required = NULL,
  size = NULL,
  swatches = NULL,
  uppercase = NULL,
  with_hint = NULL,
  with_label = NULL,
  without_format_toggle = NULL,
  hint_slot = NULL,
  label_slot = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. The default value of the form control. Primarily used for resetting the form control. This wrapper argument sets the HTML value attribute, which maps to the component's defaultValue field/property rather than its live value property.
disabled	Boolean. Default: FALSE. Disables the color picker.
label	String. Default: "". The color picker's label. This will not be displayed, but it will be announced by assistive devices. If you need to display HTML, you can use the label slot‘ instead.
hint	String. Default: "". The color picker's hint. If you need to display HTML, use the hint slot instead.
name	String. Default: null. The name of the form control, submitted as a name/value pair with form data.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
format	Enumerated string. Allowed values: hex, hsl, hsv, rgb. Default: hex. The format to use. If opacity is enabled, these will translate to HEXA, RGBA, HSLA, and HSVA respectively. The color picker will accept user input in any format (including CSS color names) and convert it to the desired format.

lang	String. Optional Web Awesome attribute.
opacity	Boolean. Default: FALSE. Shows the opacity slider. Enabling this will cause the formatted value to be HEXA, RGBA, or HSLA.
open	Boolean. Default: FALSE. Indicates whether or not the popup is open. You can toggle this attribute to show and hide the popup, or you can use the show() and hide() methods and this attribute will reflect the popup's open state.
placement	Enumerated string. Allowed values: bottom, bottom-end, bottom-start, left, left-end, left-start, right, right-end, right-start, top, top-end, top-start. Default: bottom-start. The preferred placement of the color picker's popup. Note that the actual placement will vary as configured to keep the panel inside of the viewport.
required	Boolean. Default: FALSE. Makes the color picker a required field.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. Determines the size of the color picker's trigger
swatches	Default: "". One or more predefined color swatches to display as presets in the color picker. Can include any format the color picker can parse, including HEX(A), RGB(A), HSL(A), HSV(A), and CSS color names. Each color must be separated by a semicolon (;). Alternatively, you can pass an array of color values or an array of { color, label } objects to this property using JavaScript. When using objects with labels, the label will be used for the swatch's accessible name instead of the raw color value.
uppercase	Boolean. Default: FALSE. By default, values are lowercase. With this attribute, values will be uppercase instead.
with_hint	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.
with_label	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a label element so the server-rendered markup includes the label before the component hydrates on the client.
without_format_toggle	Boolean. Default: FALSE. Removes the button that lets users toggle between format.
hint_slot	The color picker's form hint. Alternatively, you can use the hint attribute.
label_slot	The color picker's form label. Alternatively, you can use the label attribute.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current value value.

wa_comparison *Create a* wa-comparison *component*

Description

Generated wrapper for the Web Awesome wa-comparison component. When used as a Shiny input, input$<input_id> reflects the component's current value value. Generated from Web Awesome metadata.

Usage

```
wa_comparison(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  dir = NULL,
  lang = NULL,
  position = NULL,
  after = NULL,
  before = NULL,
  handle = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
position	Number. Default: 50. The position of the divider as a percentage.
after	The after content, often an <img> or <svg> element.
before	The before content, often an <img> or <svg> element.
handle	The icon used inside the handle.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current value value.

`wa_container` *Construct a shiny.webawesome container*

Description

Creates a convenience container for layouts and utility-class usage within the shiny.webawesome package API. `wa_container()` is complementary to `htmltools::tags$div()` and is not a wrapper for an upstream Web Awesome component.

Usage

```
wa_container(..., id = NULL, class = NULL, style = NULL)
```

Arguments

`...`	UI children and additional HTML attributes for the container.
`id`	Optional DOM `id` attribute.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.

Details

The helper attaches the shiny.webawesome dependency so Web Awesome utility classes can be used even when no generated component wrappers appear in the same UI subtree.

Value

A tag with the shiny.webawesome dependency attached.

Examples

```
container <- wa_container(
  class = "wa-stack",
  wa_card("First card"),
  wa_card("Second card")
)
```

wa_copy_button *Create a* wa-copy-button *component*

Description

Generated wrapper for the Web Awesome wa-copy-button component. Generated from Web Awesome metadata.

Usage

```
wa_copy_button(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  copy_label = NULL,
  dir = NULL,
  error_label = NULL,
  feedback_duration = NULL,
  from = NULL,
  lang = NULL,
  success_label = NULL,
  tooltip_placement = NULL,
  copy_icon = NULL,
  error_icon = NULL,
  success_icon = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. Default: "". The text value to copy.
disabled	Boolean. Default: FALSE. Disables the copy button.
copy_label	String. Default: "". A custom label to show in the tooltip.
dir	String. Optional Web Awesome attribute.
error_label	String. Default: "". A custom label to show in the tooltip when a copy error occurs.
feedback_duration	Number. Default: 1000. The length of time to show feedback before restoring the default trigger.

from	String. Default: "". An id that references an element in the same document from which data will be copied. If both this and value are present, this value will take precedence. By default, the target element's textContent will be copied. To copy an attribute, append the attribute name wrapped in square brackets, e.g. from="elvalue". To copy a property, append a dot and the property name, e.g. from="el.value".
lang	String. Optional Web Awesome attribute.
success_label	String. Default: "". A custom label to show in the tooltip after copying.
tooltip_placement	Enumerated string. Allowed values: bottom, left, right, top. Default: top. The preferred placement of the tooltip.
copy_icon	The icon to show in the default copy state. Works best with <wa-icon>.
error_icon	The icon to show when a copy error occurs. Works best with <wa-icon>.
success_icon	The icon to show when the content is copied. Works best with <wa-icon>.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_details` *Create a* `wa-details` *component*

Description

Generated wrapper for the Web Awesome `wa-details` component. When used as a Shiny input, `input$<input_id>` reflects the component's current semantic `open` state. The Shiny value is returned as a logical value. Generated from Web Awesome metadata.

Usage

```
wa_details(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  disabled = NULL,
  name = NULL,
  appearance = NULL,
  dir = NULL,
  icon_placement = NULL,
  lang = NULL,
  open = NULL,
```

```
  summary = NULL,
  collapse_icon = NULL,
  expand_icon = NULL,
  summary_slot = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
disabled	Boolean. Default: FALSE. Disables the details so it can't be toggled.
name	String. Groups related details elements. When one opens, others with the same name will close.
appearance	Enumerated string. Allowed values: filled, filled-outlined, outlined, plain. Default: outlined. The element's visual appearance.
dir	String. Optional Web Awesome attribute.
icon_placement	Enumerated string. Allowed values: end, start. Default: end. The location of the expand/collapse icon.
lang	String. Optional Web Awesome attribute.
open	Boolean. Default: FALSE. Indicates whether or not the details is open. You can toggle this attribute to show and hide the details, or you can use the show() and hide() methods and this attribute will reflect the details' open state.
summary	String. The summary to show in the header. If you need to display HTML, use the summary slot instead.
collapse_icon	Optional collapse icon to use instead of the default. Works best with <wa-icon>.
expand_icon	Optional expand icon to use instead of the default. Works best with <wa-icon>.
summary_slot	The details' summary. Alternatively, you can use the summary attribute.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current semantic open state. The Shiny value is returned as a logical value.

wa_dialog *Create a* wa-dialog *component*

Description

Generated wrapper for the Web Awesome wa-dialog component. When used as a Shiny input, input$<input_id> reflects the component's current semantic open state. The Shiny value is returned as a logical value. Generated from Web Awesome metadata.

Usage

```
wa_dialog(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  label = NULL,
  dir = NULL,
  lang = NULL,
  light_dismiss = NULL,
  open = NULL,
  with_footer = NULL,
  without_header = NULL,
  footer = NULL,
  header_actions = NULL,
  label_slot = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
label	String. Default: "". The dialog's label as displayed in the header. You should always include a relevant label, as it is required for proper accessibility. If you need to display HTML, use the label slot instead.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
light_dismiss	Boolean. Default: FALSE. When enabled, the dialog will be closed when the user clicks outside of it.
open	Boolean. Default: FALSE. Indicates whether or not the dialog is open. Toggle this attribute to show and hide the dialog.

with_footer	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a footer element so the server-rendered markup includes the footer before the component hydrates on the client.
without_header	Boolean. Default: FALSE. Disables the header. This will also remove the default close button.
footer	The dialog's footer, usually one or more buttons representing various options.
header_actions	Optional actions to add to the header. Works best with <wa-button>.
label_slot	The dialog's label. Alternatively, you can use the label attribute.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current semantic open state. The Shiny value is returned as a logical value.

---

wa_divider *Create a* wa-divider *component*

Description

Generated wrapper for the Web Awesome wa-divider component. Generated from Web Awesome metadata.

Usage

```
wa_divider(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  lang = NULL,
  orientation = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
orientation	Enumerated string. Allowed values: horizontal, vertical. Default: horizontal. Sets the divider's orientation.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_drawer` *Create a* `wa-drawer` *component*

Description

Generated wrapper for the Web Awesome `wa-drawer` component. When used as a Shiny input, `input$<input_id>` reflects the component's current semantic `open` state. The Shiny value is returned as a logical value. Generated from Web Awesome metadata.

Usage

```
wa_drawer(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  label = NULL,
  dir = NULL,
  lang = NULL,
  light_dismiss = NULL,
  open = NULL,
  placement = NULL,
  with_footer = NULL,
  without_header = NULL,
  footer = NULL,
  header_actions = NULL,
  label_slot = NULL
)
```

Arguments

`input_id`	Shiny input id for the component. This is also used as the rendered DOM `id` attribute.
`...`	Child content for the component's default slot.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`label`	String. Default: `""`. The drawer's label as displayed in the header. You should always include a relevant label, as it is required for proper accessibility. If you need to display HTML, use the `label` slot instead.

dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
light_dismiss	Boolean. Default: TRUE. When enabled, the drawer will be closed when the user clicks outside of it.
open	Boolean. Default: FALSE. Indicates whether or not the drawer is open. Toggle this attribute to show and hide the drawer.
placement	Enumerated string. Allowed values: bottom, end, start, top. Default: end. The direction from which the drawer will open.
with_footer	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a footer element so the server-rendered markup includes the footer before the component hydrates on the client.
without_header	Boolean. Default: FALSE. Disables the header. This will also remove the default close button.
footer	The drawer's footer, usually one or more buttons representing various options.
header_actions	Optional actions to add to the header. Works best with <wa-button>.
label_slot	The drawer's label. Alternatively, you can use the label attribute.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current semantic open state. The Shiny value is returned as a logical value.

---

wa_dropdown *Create a* wa-dropdown *component*

Description

Generated wrapper for the Web Awesome wa-dropdown component. When used as a Shiny input, action semantics are exposed through input$<input_id> and the latest selected dropdown item value is exposed through input$<input_id>_value. The action input increments on every selection, including repeated selection of the same item. The companion value input reflects the selected item's value, returns NULL when the selected item has no value, and preserves an explicit empty string "" when that is the selected item's value. Each committed action publishes a numeric action value. Generated from Web Awesome metadata.

Usage

```
wa_dropdown(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  dir = NULL,
  distance = NULL,
  lang = NULL,
  open = NULL,
  placement = NULL,
  size = NULL,
  skidding = NULL,
  trigger = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
distance	Number. Default: 0. The distance of the dropdown menu from its trigger.
lang	String. Optional Web Awesome attribute.
open	Boolean. Default: FALSE. Opens or closes the dropdown.
placement	Enumerated string. Allowed values: bottom, bottom-end, bottom-start, left, left-end, left-start, right, right-end, right-start, top, top-end, top-start. Default: bottom-start. The placement of the dropdown menu in reference to the trigger. The menu will shift to a more optimal location if the preferred placement doesn't have enough room.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. The dropdown's size.
skidding	Number. Default: 0. The offset of the dropdown menu along its trigger.
trigger	The element that triggers the dropdown, such as a <wa-button> or <button>.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> uses action semantics and invalidates on each committed action, including repeated selection of the same item. The Shiny action value is returned as a numeric action value.

input$<input_id>_value reflects the latest selected item's value value, returns NULL when the selected item has no value , and preserves an explicit empty string "" when that is the selected item's value. The payload value is returned as a character string or NULL.

---

wa_dropdown_item *Create a* wa-dropdown-item *component*

Description

Generated wrapper for the Web Awesome wa-dropdown-item component. Generated from Web Awesome metadata.

Usage

```
wa_dropdown_item(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  checked = NULL,
  disabled = NULL,
  dir = NULL,
  lang = NULL,
  submenu_open = NULL,
  type = NULL,
  variant = NULL,
  details = NULL,
  icon = NULL,
  submenu = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. An optional value for the menu item. This is useful for determining which item was selected when listening to the dropdown's wa-select event.
checked	Boolean. Default: FALSE. Set to TRUE to check the dropdown item. Only valid when type is checkbox.
disabled	Boolean. Default: FALSE. Disables the dropdown item.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.

submenu_open	Boolean. Default: FALSE. Whether the submenu is currently open.
type	Enumerated string. Allowed values: checkbox, normal. Default: normal. Set to checkbox to make the item a checkbox.
variant	Enumerated string. Allowed values: danger, default. Default: default. The type of menu item to render.
details	Additional content or details to display after the label.
icon	An optional icon to display before the label.
submenu	Submenu items, typically <wa-dropdown-item> elements, to create a nested menu.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_format_bytes *Create a* wa-format-bytes *component*

Description

Generated wrapper for the Web Awesome wa-format-bytes component. Generated from Web Awesome metadata.

Usage

```
wa_format_bytes(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  dir = NULL,
  display = NULL,
  lang = NULL,
  unit = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.

style	Optional inline CSS style string.
value	Number. Default: 0. The number to format in bytes.
dir	String. Optional Web Awesome attribute.
display	Enumerated string. Allowed values: long, narrow, short. Default: short. Determines how to display the result, e.g. "100 bytes", "100 b", or "100b".
lang	String. Optional Web Awesome attribute.
unit	Enumerated string. Allowed values: bit, byte. Default: byte. The type of unit to display.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_format_date` *Create a* `wa-format-date` *component*

Description

Generated wrapper for the Web Awesome `wa-format-date` component. Generated from Web Awesome metadata.

Usage

```
wa_format_date(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  date = NULL,
  day = NULL,
  dir = NULL,
  era = NULL,
  hour = NULL,
  hour_format = NULL,
  lang = NULL,
  minute = NULL,
  month = NULL,
  second = NULL,
  time_zone = NULL,
  time_zone_name = NULL,
  weekday = NULL,
  year = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
date	Default: `new Date()`. The date/time to format. If not set, the current date and time will be used. When passing a string, it's strongly recommended to use the ISO 8601 format to ensure timezones are handled correctly. To convert a date to this format in JavaScript, use `date.toISOString()`.
day	Enumerated string. Allowed values: `2-digit`, `numeric`. The format for displaying the day.
dir	String. Optional Web Awesome attribute.
era	Enumerated string. Allowed values: `long`, `narrow`, `short`. The format for displaying the era.
hour	Enumerated string. Allowed values: `2-digit`, `numeric`. The format for displaying the hour.
hour_format	Enumerated string. Allowed values: `12`, `24`, `auto`. Default: `auto`. The format for displaying the hour.
lang	String. Optional Web Awesome attribute.
minute	Enumerated string. Allowed values: `2-digit`, `numeric`. The format for displaying the minute.
month	Enumerated string. Allowed values: `2-digit`, `long`, `narrow`, `numeric`, `short`. The format for displaying the month.
second	Enumerated string. Allowed values: `2-digit`, `numeric`. The format for displaying the second.
time_zone	String. The time zone to express the time in.
time_zone_name	Enumerated string. Allowed values: `long`, `short`. The format for displaying the time.
weekday	Enumerated string. Allowed values: `long`, `narrow`, `short`. The format for displaying the weekday.
year	Enumerated string. Allowed values: `2-digit`, `numeric`. The format for displaying the year.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_format_number *Create a* wa-format-number *component*

Description

Generated wrapper for the Web Awesome wa-format-number component. Generated from Web Awesome metadata.

Usage

```
wa_format_number(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  currency = NULL,
  currency_display = NULL,
  dir = NULL,
  lang = NULL,
  maximum_fraction_digits = NULL,
  maximum_significant_digits = NULL,
  minimum_fraction_digits = NULL,
  minimum_integer_digits = NULL,
  minimum_significant_digits = NULL,
  type = NULL,
  without_grouping = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Number. Default: 0. The number to format.
currency	String. Default: USD. The ISO 4217 currency code to use when formatting.
currency_display	Enumerated string. Allowed values: code, name, narrowSymbol, symbol. Default: symbol. How to display the currency.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
maximum_fraction_digits	Number. The maximum number of fraction digits to use. Possible values are 0-100.

maximum_significant_digits
: Number. The maximum number of significant digits to use,. Possible values are 1-21.

minimum_fraction_digits
: Number. The minimum number of fraction digits to use. Possible values are 0-100.

minimum_integer_digits
: Number. The minimum number of integer digits to use. Possible values are 1-21.

minimum_significant_digits
: Number. The minimum number of significant digits to use. Possible values are 1-21.

type
: Enumerated string. Allowed values: `currency`, `decimal`, `percent`. Default: `decimal`. The formatting style to use.

without_grouping
: Boolean. Default: `FALSE`. Turns off grouping separators.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_icon` *Create a* `wa-icon` *component*

Description

Generated wrapper for the Web Awesome `wa-icon` component. Generated from Web Awesome metadata.

Usage

```
wa_icon(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  label = NULL,
  name = NULL,
  animation = NULL,
  auto_width = NULL,
  dir = NULL,
  family = NULL,
  flip = NULL,
```

```
  lang = NULL,
  library = NULL,
  rotate = NULL,
  src = NULL,
  swap_opacity = NULL,
  variant = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
label	String. Default: `""`. An alternate description to use for assistive devices. If omitted, the icon will be considered presentational and ignored by assistive devices.
name	String. The name of the icon to draw. Available names depend on the icon library being used.
animation	Enumerated string. Allowed values: `beat`, `beat-fade`, `bounce`, `fade`, `flip`, `shake`, `spin`, `spin-pulse`, `spin-reverse`. Sets the animation for the icon
auto_width	Boolean. Default: `FALSE`. Sets the width of the icon to match the cropped SVG viewBox. This operates like the Font `fa-width-auto` class.
dir	String. Optional Web Awesome attribute.
family	String. The family of icons to choose from. For Font Awesome Free, valid options include `classic` and `brands`. For Font Awesome Pro subscribers, valid options include, `classic`, `sharp`, `duotone`, `sharp-duotone`, and `brands`. A valid kit code must be present to show pro icons via CDN. You can set `<html data-fa-kit-code="...">` to provide one.
flip	Enumerated string. Allowed values: `both`, `x`, `y`. Sets the flip direction of the icon along the 'x' (horizontal), 'y' (vertical), or 'both' axes.
lang	String. Optional Web Awesome attribute.
library	String. Default: `default`. The name of a registered custom icon library.
rotate	Number. Default: `0`. Sets the rotation degree of the icon
src	String. An external URL of an SVG file. Be sure you trust the content you are including, as it will be executed as code and can result in XSS attacks.
swap_opacity	Boolean. Default: `FALSE`. Swaps the opacity of duotone icons.
variant	String. The name of the icon's variant. For Font Awesome, valid options include `thin`, `light`, `regular`, and `solid` for the `classic` and `sharp` families. Some variants require a Font Awesome Pro subscription. Custom icon libraries may or may not use this property.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_include` *Create a* `wa-include` *component*

Description

Generated wrapper for the Web Awesome `wa-include` component. Generated from Web Awesome metadata.

Usage

```
wa_include(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  allow_scripts = NULL,
  dir = NULL,
  lang = NULL,
  mode = NULL,
  src = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`allow_scripts`	Boolean. Default: `FALSE`. Allows included scripts to be executed. Be sure you trust the content you are including as it will be executed as code and can result in XSS attacks.
`dir`	String. Optional Web Awesome attribute.
`lang`	String. Optional Web Awesome attribute.
`mode`	Enumerated string. Allowed values: `cors`, `no-cors`, `same-origin`. Default: `cors`. The fetch mode to use.
`src`	String. The location of the HTML file to include. Be sure you trust the content you are including as it will be executed as code and can result in XSS attacks.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_input` *Create a* `wa-input` *component*

Description

Generated wrapper for the Web Awesome `wa-input` component. When used as a Shiny input, `input$<input_id>` reflects the component's current `value` value. Generated from Web Awesome metadata.

Generated update helper for the Web Awesome `wa-input` component.

Usage

```
wa_input(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  hint = NULL,
  name = NULL,
  appearance = NULL,
  autocapitalize = NULL,
  autocomplete = NULL,
  autocorrect = NULL,
  autofocus = NULL,
  custom_error = NULL,
  dir = NULL,
  enterkeyhint = NULL,
  inputmode = NULL,
  lang = NULL,
  max = NULL,
  maxlength = NULL,
  min = NULL,
  minlength = NULL,
  password_toggle = NULL,
  password_visible = NULL,
  pattern = NULL,
  pill = NULL,
  placeholder = NULL,
  readonly = NULL,
  required = NULL,
```

```
  size = NULL,
  spellcheck = NULL,
  step = NULL,
  title = NULL,
  type = NULL,
  with_clear = NULL,
  with_hint = NULL,
  with_label = NULL,
  without_spin_buttons = NULL,
  clear_icon = NULL,
  end = NULL,
  hide_password_icon = NULL,
  hint_slot = NULL,
  label_slot = NULL,
  show_password_icon = NULL,
  start = NULL
)

update_wa_input(
  session,
  input_id,
  value = NULL,
  label = NULL,
  hint = NULL,
  disabled = NULL
)
```

Arguments

input_id	Input id of the component to update.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Optional value to send to the component.
disabled	Optional logical disabled state to send to the component.
label	Optional label text to send to the component.
hint	Optional hint text to send to the component.
name	String. Default: null. The name of the input, submitted as a name/value pair with form data.
appearance	Enumerated string. Allowed values: filled, filled-outlined, outlined. Default: outlined. The input's visual appearance.
autocapitalize	Enumerated string. Allowed values: characters, none, off, on, sentences, words. Controls whether and how text input is automatically capitalized as it is entered by the user.
autocomplete	String. Specifies what permission the browser has to provide assistance in filling out form field values. Refer to this page on MDN for available values.

autocorrect — Boolean. Default: FALSE. Indicates whether the browser's autocorrect feature is on or off. When set as an attribute, use "off" or "on". When set as a property, use TRUE or FALSE. Because the upstream implementation treats attributes and properties differently for this field, this argument accepts TRUE, FALSE, "on", "off", NULL. TRUE emits autocorrect="on" and FALSE emits autocorrect="off". NULL omits the attribute. Accepted string values: "on" is emitted unchanged; "off" is emitted unchanged.

autofocus — Boolean. Default: FALSE. Indicates that the input should receive focus on page load.

custom_error — String. Default: null. Optional Web Awesome attribute.

dir — String. Optional Web Awesome attribute.

enterkeyhint — Enumerated string. Allowed values: done, enter, go, next, previous, search, send. Used to customize the label or icon of the Enter key on virtual keyboards.

inputmode — Enumerated string. Allowed values: decimal, email, none, numeric, search, tel, text, url. Tells the browser what type of data will be entered by the user, allowing it to display the appropriate virtual keyboard on supportive devices.

lang — String. Optional Web Awesome attribute.

max — The input's maximum value. Only applies to date and number input types.

maxlength — Number. The maximum length of input that will be considered valid.

min — The input's minimum value. Only applies to date and number input types.

minlength — Number. The minimum length of input that will be considered valid.

password_toggle — Boolean. Default: FALSE. Adds a button to toggle the password's visibility. Only applies to password types.

password_visible — Boolean. Default: FALSE. Determines whether or not the password is currently visible. Only applies to password input types.

pattern — String. A regular expression pattern to validate input against.

pill — Boolean. Default: FALSE. Draws a pill-style input with rounded edges.

placeholder — String. Default: "". Placeholder text to show as a hint when the input is empty.

readonly — Boolean. Default: FALSE. Makes the input readonly.

required — Boolean. Default: FALSE. Makes the input a required field.

size — Enumerated string. Allowed values: large, medium, small. Default: medium. The input's size.

spellcheck — Boolean. Default: TRUE. Enables spell checking on the input.

step — Specifies the granularity that the value must adhere to, or the special value any which means no stepping is implied, allowing any numeric value. Only applies to date and number input types.

title — String. Default: "". Optional Web Awesome attribute.

type — Enumerated string. Allowed values: date, datetime-local, email, number, password, search, tel, text, time, url. Default: text. The type of input. Works the same as a native <input> element, but only a subset of types are supported. Defaults to text.

with_clear Boolean. Default: FALSE. Adds a clear button when the input is not empty.

with_hint Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.

with_label Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a label element so the server-rendered markup includes the label before the component hydrates on the client.

without_spin_buttons
Boolean. Default: FALSE. Hides the browser's built-in increment/decrement spin buttons for number inputs.

clear_icon An icon to use in lieu of the default clear icon.

end An element, such as <wa-icon>, placed at the end of the input control.

hide_password_icon
An icon to use in lieu of the default hide password icon.

hint_slot Text that describes how to use the input. Alternatively, you can use the hint attribute.

label_slot The input's label. Alternatively, you can use the label attribute.

show_password_icon
An icon to use in lieu of the default show password icon.

start An element, such as <wa-icon>, placed at the start of the input control.

session Shiny session object.

Value

An HTML tag for the component.

Invisibly returns NULL.

Shiny Bindings

input$<input_id> reflects the component's current value value.

---

wa_intersection_observer

Create a wa-intersection-observer *component*

Description

Generated wrapper for the Web Awesome wa-intersection-observer component. Generated from Web Awesome metadata.

Usage

```
wa_intersection_observer(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  disabled = NULL,
  dir = NULL,
  intersect_class = NULL,
  lang = NULL,
  once = NULL,
  root = NULL,
  root_margin = NULL,
  threshold = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`disabled`	Boolean. Default: `FALSE`. Deactivates the intersection observer functionality.
`dir`	String. Optional Web Awesome attribute.
`intersect_class`	String. Default: `""`. CSS class applied to elements during intersection. Automatically removed when elements leave the viewport, enabling pure CSS styling based on visibility state.
`lang`	String. Optional Web Awesome attribute.
`once`	Boolean. Default: `FALSE`. If enabled, observation ceases after initial intersection.
`root`	String. Default: `null`. Element ID to define the viewport boundaries for tracked targets.
`root_margin`	String. Default: `0px`. Offset space around the root boundary. Accepts values like CSS margin syntax.
`threshold`	String. Default: `0`. One or more space-separated values representing visibility percentages that trigger the observer callback.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_js *Register a small JavaScript snippet with the page*

Description

Adds one inline JavaScript snippet to the page and attaches the shiny.webawesome dependency.

Usage

```
wa_js(code)
```

Arguments

code	One non-missing JavaScript string to inject into the page.

Details

`wa_js()` is a small package-level helper for app-local browser glue that should remain easy to see from the surrounding Shiny code. Typical uses include listening for browser-side events, reading live component properties, and publishing values back to Shiny with `Shiny.setInputValue()`.

A typical property-read pattern is:

1. find the browser element by DOM `id`
2. read the live property in JavaScript
3. publish the value with `Shiny.setInputValue()`
4. consume the published value in Shiny as an ordinary input

This helper is intentionally narrow. It accepts a scalar JavaScript string and injects it into the rendered page. For larger or shared scripts, prefer standard Shiny asset patterns such as `www/` files and `tags$script()`.

Value

A `<script>` tag with the shiny.webawesome dependency attached.

Examples

```
if (interactive()) {
  ui <- webawesomePage(
    wa_js("
      function publishDetailsOpen() {
        const details = document.getElementById('details');

        if (!details ||
            !window.Shiny ||
            typeof window.Shiny.setInputValue !== 'function') {
          return;
        }
```

```
          window.Shiny.setInputValue(
            'details_open_state',
            details.open,
            { priority: 'event' }
          );
        }

        document.addEventListener('shiny:connected', publishDetailsOpen, {
          once: true
        });

        document.addEventListener('wa-show', function(event) {
          if (event.target.id === 'details') {
            publishDetailsOpen();
          }
        });

        document.addEventListener('wa-hide', function(event) {
          if (event.target.id === 'details') {
            publishDetailsOpen();
          }
        });
      "),
      wa_details(
        input_id = "details",
        summary = "More information",
        "Details body"
      ),
      shiny::verbatimTextOutput("details_state")
    )

    server <- function(input, output, session) {
      output$details_state <- shiny::renderPrint({
        input$details_open_state
      })
    }

    shiny::shinyApp(ui, server)
  }

  if (interactive()) {
    js_file <- tempfile(fileext = ".js")
    writeLines(
      c(
        "document.addEventListener('wa-show', function(event) {",
        "  if (event.target.id !== 'details') {",
        "    return;",
        "  }",
        "",
        "  if (window.Shiny &&",
        "      typeof window.Shiny.setInputValue === 'function') {",
        "    window.Shiny.setInputValue(",
```

```
      "      'details_open_state',",
      "      event.target.open,",
      "      { priority: 'event' }",
      "    );",
      "  }",
      "});"
    ),
    js_file
  )

  js_code <- paste(readLines(js_file, warn = FALSE), collapse = "\n")

  ui <- webawesomePage(
    wa_js(js_code),
    wa_details(
      input_id = "details",
      summary = "More information",
      "Details body"
    )
  )
}
```

---

wa_markdown *Create a* wa-markdown *component*

Description

Generated wrapper for the Web Awesome wa-markdown component. Generated from Web Awesome metadata.

Usage

```
wa_markdown(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  lang = NULL,
  tab_size = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.

dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
tab_size	Number. Default: 4. The tab stop width used when converting leading tabs to spaces during whitespace normalization.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_mutation_observer *Create a* wa-mutation-observer *component*

Description

Generated wrapper for the Web Awesome wa-mutation-observer component. Generated from Web Awesome metadata.

Usage

```
wa_mutation_observer(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  disabled = NULL,
  attr = NULL,
  attr_old_value = NULL,
  char_data = NULL,
  char_data_old_value = NULL,
  child_list = NULL,
  dir = NULL,
  lang = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
disabled	Boolean. Default: FALSE. Disables the observer.

attr — String. Watches for changes to attributes. To watch only specific attributes, separate them by a space, e.g. attr="class id title". To watch all attributes, use *.

attr_old_value — Boolean. Default: FALSE. Indicates whether or not the attribute's previous value should be recorded when monitoring changes.

char_data — Boolean. Default: FALSE. Watches for changes to the character data contained within the node.

char_data_old_value — Boolean. Default: FALSE. Indicates whether or not the previous value of the node's text should be recorded.

child_list — Boolean. Default: FALSE. Watches for the addition or removal of new child nodes.

dir — String. Optional Web Awesome attribute.

lang — String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_number_input *Create a* wa-number-input *component*

Description

Generated wrapper for the Web Awesome wa-number-input component. When used as a Shiny input, input$<input_id> reflects the component's current value value. Generated from Web Awesome metadata.

Generated update helper for the Web Awesome wa-number-input component.

Usage

```
wa_number_input(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  hint = NULL,
  name = NULL,
```

```
  appearance = NULL,
  autocomplete = NULL,
  autofocus = NULL,
  custom_error = NULL,
  dir = NULL,
  enterkeyhint = NULL,
  inputmode = NULL,
  lang = NULL,
  max = NULL,
  min = NULL,
  pill = NULL,
  placeholder = NULL,
  readonly = NULL,
  required = NULL,
  size = NULL,
  step = NULL,
  title = NULL,
  with_hint = NULL,
  with_label = NULL,
  without_steppers = NULL,
  decrement_icon = NULL,
  end = NULL,
  hint_slot = NULL,
  increment_icon = NULL,
  label_slot = NULL,
  start = NULL
)

update_wa_number_input(
  session,
  input_id,
  value = NULL,
  label = NULL,
  hint = NULL,
  disabled = NULL
)
```

Arguments

input_id	Input id of the component to update.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Optional value to send to the component.
disabled	Optional logical disabled state to send to the component.
label	Optional label text to send to the component.
hint	Optional hint text to send to the component.

name	String. Default: null. The name of the input, submitted as a name/value pair with form data.
appearance	Enumerated string. Allowed values: filled, filled-outlined, outlined. Default: outlined. The input's visual appearance.
autocomplete	String. Specifies what permission the browser has to provide assistance in filling out form field values. Refer to this page on MDN for available values.
autofocus	Boolean. Default: FALSE. Indicates that the input should receive focus on page load.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
enterkeyhint	Enumerated string. Allowed values: done, enter, go, next, previous, search, send. Used to customize the label or icon of the Enter key on virtual keyboards.
inputmode	Enumerated string. Allowed values: decimal, numeric. Default: numeric. Tells the browser what type of data will be entered by the user, allowing it to display the appropriate virtual keyboard on supportive devices.
lang	String. Optional Web Awesome attribute.
max	Number. The input's maximum value.
min	Number. The input's minimum value.
pill	Boolean. Default: FALSE. Draws a pill-style input with rounded edges.
placeholder	String. Default: "". Placeholder text to show as a hint when the input is empty.
readonly	Boolean. Default: FALSE. Makes the input readonly.
required	Boolean. Default: FALSE. Makes the input a required field.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. The input's size.
step	Default: 1. Specifies the granularity that the value must adhere to, or the special value any which means no stepping is implied, allowing any numeric value.
title	String. Default: "". Optional Web Awesome attribute.
with_hint	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.
with_label	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a label element so the server-rendered markup includes the label before the component hydrates on the client.
without_steppers	Boolean. Default: FALSE. Hides the increment/decrement stepper buttons.
decrement_icon	An icon to use in lieu of the default decrement icon.
end	An element, such as <wa-icon>, placed at the end of the input control (before steppers).
hint_slot	Text that describes how to use the input. Alternatively, you can use the hint attribute.
increment_icon	An icon to use in lieu of the default increment icon.
label_slot	The input's label. Alternatively, you can use the label attribute.
start	An element, such as <wa-icon>, placed at the start of the input control.
session	Shiny session object.

Value

An HTML tag for the component.

Invisibly returns NULL.

Shiny Bindings

input$<input_id> reflects the component's current value value.

---

wa_option *Create a* wa-option *component*

Description

Generated wrapper for the Web Awesome wa-option component. Generated from Web Awesome metadata.

Usage

```
wa_option(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  dir = NULL,
  lang = NULL,
  selected = NULL,
  end = NULL,
  start = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. Default: "". The option's value. When selected, the containing form control will receive this value. The value must be unique from other options in the same group. Values may not contain spaces, as spaces are used as delimiters when listing multiple values.
disabled	Boolean. Default: FALSE. Draws the option in a disabled state, preventing selection.

label	String. The option's plain text label. Usually automatically generated, but can be useful to provide manually for cases involving complex content.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
selected	Boolean. Default: FALSE. Selects an option initially. This wrapper argument sets the HTML selected attribute, which maps to the component's defaultSelected field/property.
end	An element, such as <wa-icon>, placed after the label.
start	An element, such as <wa-icon>, placed before the label.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_page *Create a* wa-page *component*

Description

Generated wrapper for the Web Awesome wa-page component. Generated from Web Awesome metadata.

Usage

```
wa_page(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  disable_navigation_toggle = NULL,
  lang = NULL,
  mobile_breakpoint = NULL,
  nav_open = NULL,
  navigation_placement = NULL,
  view = NULL,
  aside = NULL,
  banner = NULL,
  footer = NULL,
  header = NULL,
  main_footer = NULL,
  main_header = NULL,
```

```
  menu = NULL,
  navigation = NULL,
  navigation_footer = NULL,
  navigation_header = NULL,
  navigation_toggle = NULL,
  navigation_toggle_icon = NULL,
  skip_to_content = NULL,
  subheader = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
disable_navigation_toggle	Boolean. Default: FALSE. Determines whether or not to hide the default hamburger button. This will automatically flip to "TRUE" if you add an element with data-toggle-nav anywhere in the element light DOM. Generally this will be set for you and you don't need to do anything, unless you're using SSR, in which case you should set this manually for initial page loads.
lang	String. Optional Web Awesome attribute.
mobile_breakpoint	String. Default: 768px. At what page width to hide the "navigation" slot and collapse into a hamburger button. Accepts both numbers (interpreted as px) and CSS lengths (e.g. 50em), which are resolved based on the root element.
nav_open	Boolean. Default: FALSE. Whether or not the navigation drawer is open. Note, the navigation drawer is only "open" on mobile views.
navigation_placement	Enumerated string. Allowed values: end, start. Default: start. Where to place the navigation when in the mobile viewport.
view	Enumerated string. Allowed values: desktop, mobile. Default: desktop. The view is a reflection of the "mobileBreakpoint", when the page is larger than the mobile-breakpoint (768px by default), it is considered to be a "desktop" view. The view is merely a way to distinguish when to show/hide the navigation. You can use additional media queries to make other adjustments to content as necessary. The default is "desktop" because the "mobile navigation drawer" isn't accessible via SSR due to drawer requiring JS.
aside	Content to be shown on the right side of the page. Typically contains a table of contents, ads, etc. This section "sticks" to the top as the page scrolls.
banner	The banner that gets display above the header. The banner will not be shown if no content is provided.
footer	The content to display in the footer. This is always displayed underneath the viewport so will always make the page "scrollable".

header — The header to display at the top of the page. If a banner is present, the header will appear below the banner. The header will not be shown if there is no content.

main_footer — Footer to display inline below the main content.

main_header — Header to display inline above the main content.

menu — The left side of the page. If you slot an element in here, you will override the default navigation slot and will be handling navigation on your own. This also will not disable the fallback behavior of the navigation button. This section "sticks" to the top as the page scrolls.

navigation — The main content to display in the navigation area. This is displayed on the left side of the page, if menu is not used. This section "sticks" to the top as the page scrolls.

navigation_footer — The footer for a navigation area. On mobile this will be the footer for <wa-drawer>.

navigation_header — The header for a navigation area. On mobile this will be the header for <wa-drawer>.

navigation_toggle — Use this slot to slot in your own button + icon for toggling the navigation drawer. By default it is a <wa-button> + a 3 bars <wa-icon>

navigation_toggle_icon — Use this to slot in your own icon for toggling the navigation drawer. By default it is 3 bars <wa-icon>.

skip_to_content — The "skip to content" slot. You can override this If you would like to override the Skip to content button and add additional "Skip to X", they can be inserted here.

subheader — A subheader to display below the header. This is a good place to put things like breadcrumbs.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_popover *Create a* wa-popover *component*

Description

Generated wrapper for the Web Awesome wa-popover component. Generated from Web Awesome metadata.

Usage

```
wa_popover(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  distance = NULL,
  `for` = NULL,
  lang = NULL,
  open = NULL,
  placement = NULL,
  skidding = NULL,
  without_arrow = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
distance	Number. Default: `8`. The distance in pixels from which to offset the popover away from its target.
for	String. Default: `null`. The ID of the popover's anchor element. This must be an interactive/focusable element such as a button.
lang	String. Optional Web Awesome attribute.
open	Boolean. Default: `FALSE`. Shows or hides the popover.
placement	Enumerated string. Allowed values: `bottom`, `bottom-end`, `bottom-start`, `left`, `left-end`, `left-start`, `right`, `right-end`, `right-start`, `top`, `top-end`, `top-start`. Default: `top`. The preferred placement of the popover. Note that the actual placement may vary as needed to keep the popover inside of the viewport.
skidding	Number. Default: `0`. The distance in pixels from which to offset the popover along its target.
without_arrow	Boolean. Default: `FALSE`. Removes the arrow from the popover.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_popup *Create a* wa-popup *component*

Description

Generated wrapper for the Web Awesome wa-popup component. Generated from Web Awesome metadata.

Usage

```
wa_popup(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  active = NULL,
  anchor = NULL,
  arrow = NULL,
  arrow_padding = NULL,
  arrow_placement = NULL,
  auto_size = NULL,
  auto_size_padding = NULL,
  auto_size_boundary = NULL,
  boundary = NULL,
  dir = NULL,
  distance = NULL,
  flip = NULL,
  flip_fallback_placements = NULL,
  flip_fallback_strategy = NULL,
  flip_padding = NULL,
  flip_boundary = NULL,
  hover_bridge = NULL,
  lang = NULL,
  placement = NULL,
  shift = NULL,
  shift_padding = NULL,
  shift_boundary = NULL,
  skidding = NULL,
  sync = NULL,
  anchor_slot = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.

class — Optional CSS class string.

style — Optional inline CSS style string.

active — Boolean. Default: FALSE. Activates the positioning logic and shows the popup. When this attribute is removed, the positioning logic is torn down and the popup will be hidden.

anchor — The element the popup will be anchored to. If the anchor lives outside of the popup, you can provide the anchor element id, a DOM element reference, or a VirtualElement. If the anchor lives inside the popup, use the anchor slot instead.

arrow — Boolean. Default: FALSE. Attaches an arrow to the popup. The arrow's size and color can be customized using the --arrow-size and --arrow-color custom properties. For additional customizations, you can also target the arrow using ::part(arrow) in your stylesheet.

arrow_padding — Number. Default: 10. The amount of padding between the arrow and the edges of the popup. If the popup has a border-radius, for example, this will prevent it from overflowing the corners.

arrow_placement — Enumerated string. Allowed values: anchor, center, end, start. Default: anchor. The placement of the arrow. The default is anchor, which will align the arrow as close to the center of the anchor as possible, considering available space and arrow-padding. A value of start, end, or center will align the arrow to the start, end, or center of the popover instead.

auto_size — Enumerated string. Allowed values: both, horizontal, vertical. When set, this will cause the popup to automatically resize itself to prevent it from overflowing.

auto_size_padding — Number. Default: 0. The amount of padding, in pixels, to exceed before the auto-size behavior will occur.

auto_size_boundary — The auto-size boundary describes clipping element(s) that overflow will be checked relative to when resizing. By default, the boundary includes overflow ancestors that will cause the element to be clipped. If needed, you can change the boundary by passing a reference to one or more elements to this property.

boundary — Enumerated string. Allowed values: scroll, viewport. Default: viewport. The bounding box to use for flipping, shifting, and auto-sizing.

dir — String. Optional Web Awesome attribute.

distance — Number. Default: 0. The distance in pixels from which to offset the panel away from its anchor.

flip — Boolean. Default: FALSE. When set, placement of the popup will flip to the opposite site to keep it in view. You can use flipFallbackPlacements to further configure how the fallback placement is determined.

flip_fallback_placements — String. Default: "". If the preferred placement doesn't fit, popup will be tested in these fallback placements until one fits. Must be a string of any number of placements separated by a space, e.g. "top bottom left". If no placement fits, the flip fallback strategy will be used instead.

flip_fallback_strategy
: Enumerated string. Allowed values: `best-fit`, `initial`. Default: `best-fit`. When neither the preferred placement nor the fallback placements fit, this value will be used to determine whether the popup should be positioned using the best available fit based on available space or as it was initially preferred.

flip_padding
: Number. Default: `0`. The amount of padding, in pixels, to exceed before the flip behavior will occur.

flip_boundary
: The flip boundary describes clipping element(s) that overflow will be checked relative to when flipping. By default, the boundary includes overflow ancestors that will cause the element to be clipped. If needed, you can change the boundary by passing a reference to one or more elements to this property.

hover_bridge
: Boolean. Default: `FALSE`. When a gap exists between the anchor and the popup element, this option will add a "hover bridge" that fills the gap using an invisible element. This makes listening for events such as `mouseenter` and `mouseleave` more sane because the pointer never technically leaves the element. The hover bridge will only be drawn when the popover is active.

lang
: String. Optional Web Awesome attribute.

placement
: Enumerated string. Allowed values: `bottom`, `bottom-end`, `bottom-start`, `left`, `left-end`, `left-start`, `right`, `right-end`, `right-start`, `top`, `top-end`, `top-start`. Default: `top`. The preferred placement of the popup. Note that the actual placement will vary as configured to keep the panel inside of the viewport.

shift
: Boolean. Default: `FALSE`. Moves the popup along the axis to keep it in view when clipped.

shift_padding
: Number. Default: `0`. The amount of padding, in pixels, to exceed before the shift behavior will occur.

shift_boundary
: The shift boundary describes clipping element(s) that overflow will be checked relative to when shifting. By default, the boundary includes overflow ancestors that will cause the element to be clipped. If needed, you can change the boundary by passing a reference to one or more elements to this property.

skidding
: Number. Default: `0`. The distance in pixels from which to offset the panel along its anchor.

sync
: Enumerated string. Allowed values: `both`, `height`, `width`. Syncs the popup's width or height to that of the anchor element.

anchor_slot
: The element the popup will be anchored to. If the anchor lives outside of the popup, you can use the `anchor` attribute or property instead.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_progress_bar *Create a* wa-progress-bar *component*

Description

Generated wrapper for the Web Awesome wa-progress-bar component. Generated from Web Awesome metadata.

Usage

```
wa_progress_bar(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  label = NULL,
  dir = NULL,
  indeterminate = NULL,
  lang = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Number. Default: 0. The current progress as a percentage, 0 to 100.
label	String. Default: "". A custom label for assistive devices.
dir	String. Optional Web Awesome attribute.
indeterminate	Boolean. Default: FALSE. When TRUE, percentage is ignored, the label is hidden, and the progress bar is drawn in an indeterminate state.
lang	String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_progress_ring *Create a* wa-progress-ring *component*

Description

Generated wrapper for the Web Awesome wa-progress-ring component. Generated from Web Awesome metadata.

Usage

```
wa_progress_ring(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  label = NULL,
  dir = NULL,
  lang = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Number. Default: 0. The current progress as a percentage, 0 to 100.
label	String. Default: "". A custom label for assistive devices.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_qr_code *Create a* wa-qr-code *component*

Description

Generated wrapper for the Web Awesome wa-qr-code component. Generated from Web Awesome metadata.

Usage

```
wa_qr_code(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  label = NULL,
  background = NULL,
  dir = NULL,
  error_correction = NULL,
  fill = NULL,
  lang = NULL,
  radius = NULL,
  size = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. Default: "". The QR code's value.
label	String. Default: "". The label for assistive devices to announce. If unspecified, the value will be used instead.
background	String. Default: "". The background color. This can be any valid CSS color or transparent. It cannot be a CSS custom property.
dir	String. Optional Web Awesome attribute.
error_correction	Enumerated string. Allowed values: H, L, M, Q. Default: H. The level of error correction to use. Learn more
fill	String. Default: "". The fill color. This can be any valid CSS color, but not a CSS custom property.
lang	String. Optional Web Awesome attribute.

radius	Number. Default: 0. The edge radius of each module. Must be between 0 and 0.5.
size	Number. Default: 128. The size of the QR code, in pixels.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_radio *Create a* wa-radio *component*

Description

Generated wrapper for the Web Awesome wa-radio component. Generated from Web Awesome metadata.

Usage

```
wa_radio(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  name = NULL,
  appearance = NULL,
  custom_error = NULL,
  dir = NULL,
  lang = NULL,
  size = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. The radio's value. When selected, the radio group will receive this value.
disabled	Boolean. Default: FALSE. Disables the radio.

name	String. Default: null. The name of the input, submitted as a name/value pair with form data.
appearance	Enumerated string. Allowed values: button, default. Default: default. The radio's visual appearance.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
size	Enumerated string. Allowed values: large, medium, small. The radio's size. When used inside a radio group, the size will be determined by the radio group's size, which will override this attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_radio_group *Create a* wa-radio-group *component*

Description

Generated wrapper for the Web Awesome wa-radio-group component. When used as a Shiny input, input$<input_id> reflects the component's current value value. Generated from Web Awesome metadata.

Usage

```
wa_radio_group(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  hint = NULL,
  name = NULL,
  custom_error = NULL,
  dir = NULL,
  lang = NULL,
  orientation = NULL,
  required = NULL,
  size = NULL,
```

```
  with_hint = NULL,
  with_label = NULL,
  hint_slot = NULL,
  label_slot = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. The default value of the form control. Primarily used for resetting the form control. This wrapper argument sets the HTML value attribute, which maps to the component's defaultValue field/property rather than its live value property.
disabled	Boolean. Default: FALSE. Disables the radio group and all child radios.
label	String. Default: "". The radio group's label. Required for proper accessibility. If you need to display HTML, use the label slot instead.
hint	String. Default: "". The radio groups's hint. If you need to display HTML, use the hint slot instead.
name	String. Default: null. The name of the radio group, submitted as a name/value pair with form data.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
orientation	Enumerated string. Allowed values: horizontal, vertical. Default: vertical. The orientation in which to show radio items.
required	Boolean. Default: FALSE. Ensures a child radio is checked before allowing the containing form to submit.
size	Enumerated string. Allowed values: large, medium, small. The radio group's size. When present, this size will be applied to all <wa-radio> items inside.
with_hint	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.
with_label	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a label element so the server-rendered markup includes the label before the component hydrates on the client.
hint_slot	Text that describes how to use the radio group. Alternatively, you can use the hint attribute.
label_slot	The radio group's label. Required for proper accessibility. Alternatively, you can use the label attribute.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current value value.

---

wa_rating *Create a* wa-rating *component*

Description

Generated wrapper for the Web Awesome wa-rating component. When used as a Shiny input, input$<input_id> reflects the component's current value value. The Shiny value is returned as a numeric value. Generated from Web Awesome metadata.

Usage

```
wa_rating(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  name = NULL,
  custom_error = NULL,
  default_value = NULL,
  dir = NULL,
  get_symbol = NULL,
  lang = NULL,
  max = NULL,
  precision = NULL,
  readonly = NULL,
  required = NULL,
  size = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.

value	Number. Default: 0. The current rating.
disabled	Boolean. Default: FALSE. Disables the rating.
label	String. Default: "". A label that describes the rating to assistive devices.
name	String. Default: null. The name of the rating, submitted as a name/value pair with form data.
custom_error	String. Default: null. Optional Web Awesome attribute.
default_value	Number. Default: 0. The default value of the form control. Used to reset the rating to its initial value.
dir	String. Optional Web Awesome attribute.
get_symbol	A function that customizes the symbol to be rendered. The first and only argument is the rating's current value. The function should return a string containing trusted HTML of the symbol to render at the specified value. Works well with <wa-icon> elements.
lang	String. Optional Web Awesome attribute.
max	Number. Default: 5. The highest rating to show.
precision	Number. Default: 1. The precision at which the rating will increase and decrease. For example, to allow half-star ratings, set this attribute to 0.5.
readonly	Boolean. Default: FALSE. Makes the rating readonly.
required	Boolean. Default: FALSE. Makes the rating a required field.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. The component's size.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current value value. The Shiny value is returned as a numeric value.

---

wa_relative_time *Create a* wa-relative-time *component*

Description

Generated wrapper for the Web Awesome wa-relative-time component. Generated from Web Awesome metadata.

Usage

```
wa_relative_time(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  date = NULL,
  dir = NULL,
  format = NULL,
  lang = NULL,
  numeric = NULL,
  sync = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`date`	Default: `new Date()`. The date from which to calculate time from. If not set, the current date and time will be used. When passing a string, it's strongly recommended to use the ISO 8601 format to ensure timezones are handled correctly. To convert a date to this format in JavaScript, use `date.toISOString()`.
`dir`	String. Optional Web Awesome attribute.
`format`	Enumerated string. Allowed values: `long`, `narrow`, `short`. Default: `long`. The formatting style to use.
`lang`	String. Optional Web Awesome attribute.
`numeric`	Enumerated string. Allowed values: `always`, `auto`. Default: `auto`. When `auto`, values such as "yesterday" and "tomorrow" will be shown when possible. When `always`, values such as "1 day ago" and "in 1 day" will be shown.
`sync`	Boolean. Default: `FALSE`. Keep the displayed value up to date as time passes.

Value

An HTML tag for the component.

Shiny Bindings

None.

`wa_resize_observer` *Create a* `wa-resize-observer` *component*

Description

Generated wrapper for the Web Awesome `wa-resize-observer` component. Generated from Web Awesome metadata.

Usage

```
wa_resize_observer(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  disabled = NULL,
  dir = NULL,
  lang = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`disabled`	Boolean. Default: `FALSE`. Disables the observer.
`dir`	String. Optional Web Awesome attribute.
`lang`	String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_scroller *Create a* wa-scroller *component*

Description

Generated wrapper for the Web Awesome wa-scroller component. Generated from Web Awesome metadata.

Usage

```
wa_scroller(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  lang = NULL,
  orientation = NULL,
  without_scrollbar = NULL,
  without_shadow = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
orientation	Enumerated string. Allowed values: horizontal, vertical. Default: horizontal. The scroller's orientation.
without_scrollbar	Boolean. Default: FALSE. Removes the visible scrollbar.
without_shadow	Boolean. Default: FALSE. Removes the shadows.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_select *Create a* wa-select *component*

Description

Generated wrapper for the Web Awesome wa-select component. When used as a Shiny input, input$<input_id> reflects the component's current value value. The Shiny value is returned as a character string for single-select usage, or a character vector when multiple is TRUE. Generated from Web Awesome metadata.

Generated update helper for the Web Awesome wa-select component.

Usage

```
wa_select(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  hint = NULL,
  name = NULL,
  appearance = NULL,
  custom_error = NULL,
  dir = NULL,
  lang = NULL,
  max_options_visible = NULL,
  multiple = NULL,
  open = NULL,
  pill = NULL,
  placeholder = NULL,
  placement = NULL,
  required = NULL,
  size = NULL,
  with_clear = NULL,
  with_hint = NULL,
  with_label = NULL,
  clear_icon = NULL,
  end = NULL,
  expand_icon = NULL,
  hint_slot = NULL,
  label_slot = NULL,
  start = NULL
)
```

```
update_wa_select(
  session,
  input_id,
  value = NULL,
  label = NULL,
  hint = NULL,
  disabled = NULL
)
```

Arguments

input_id	Input id of the component to update.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Optional value to send to the component.
disabled	Optional logical disabled state to send to the component.
label	Optional label text to send to the component.
hint	Optional hint text to send to the component.
name	String. Default: `""`. The name of the select, submitted as a name/value pair with form data.
appearance	Enumerated string. Allowed values: `filled`, `filled-outlined`, `outlined`. Default: `outlined`. The select's visual appearance.
custom_error	String. Default: `null`. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
max_options_visible	Number. Default: `3`. The maximum number of selected options to show when `multiple` is TRUE. After the maximum, "+n" will be shown to indicate the number of additional items that are selected. Set to 0 to remove the limit.
multiple	Boolean. Default: `FALSE`. Allows more than one option to be selected.
open	Boolean. Default: `FALSE`. Indicates whether or not the select is open. You can toggle this attribute to show and hide the menu, or you can use the `show()` and `hide()` methods and this attribute will reflect the select's open state.
pill	Boolean. Default: `FALSE`. Draws a pill-style select with rounded edges.
placeholder	String. Default: `""`. Placeholder text to show as a hint when the select is empty.
placement	Enumerated string. Allowed values: `bottom`, `top`. Default: `bottom`. The preferred placement of the select's menu. Note that the actual placement may vary as needed to keep the listbox inside of the viewport.
required	Boolean. Default: `FALSE`. The select's required attribute.
size	Enumerated string. Allowed values: `large`, `medium`, `small`. Default: `medium`. The select's size.

with_clear	Boolean. Default: FALSE. Adds a clear button when the select is not empty.
with_hint	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.
with_label	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a label element so the server-rendered markup includes the label before the component hydrates on the client.
clear_icon	An icon to use in lieu of the default clear icon.
end	An element, such as <wa-icon>, placed at the end of the combobox.
expand_icon	The icon to show when the control is expanded and collapsed. Rotates on open and close.
hint_slot	Text that describes how to use the input. Alternatively, you can use the hint attribute.
label_slot	The input's label. Alternatively, you can use the label attribute.
start	An element, such as <wa-icon>, placed at the start of the combobox.
session	Shiny session object.

Value

An HTML tag for the component.

Invisibly returns NULL.

Shiny Bindings

input$<input_id> reflects the component's current value value. The Shiny value is returned as a character string for single-select usage, or a character vector when multiple is TRUE.

---

wa_set_property *Set one live property on a browser-side element*

Description

Sends a one-way command from the Shiny server that sets a live browser-side property on the element identified by DOM id.

Usage

```
wa_set_property(
  id,
  property,
  value,
  session = shiny::getDefaultReactiveDomain()
)
```

Arguments

id	DOM id of the target browser element.
property	Scalar property name to assign on the target element.
value	Value to assign. This should be serializable through Shiny's custom-message transport. In practice, prefer JSON-like values such as strings, numbers, logicals, NULL, vectors/lists of scalars, and named lists that map cleanly to browser objects. Do not expect R functions, language objects, or HTML tags to serialize into useful browser-side values.
session	Shiny session object. Defaults to the current reactive domain.

Details

wa_set_property() is a narrow package-level escape hatch for advanced cases where a Web Awesome component property needs to be updated from server logic but is not covered by a generated update helper. It does not validate whether the requested property exists for the targeted component.

This helper is complementary to generated component bindings and update helpers. It is not part of upstream component coverage and does not expand the generated per-component API surface.

On the server side, wa_set_property() validates only its R helper inputs, such as the target id, property name, and session. It does not validate whether the requested property exists on the browser-side element.

In the browser runtime, the command layer validates that the target DOM id resolves to an element and that a property name was supplied, then assigns the value directly. Command-layer warnings are controlled by the package warning registry, especially the command_layer key. For option details, see the Package Options article.

Value

Invisibly returns NULL.

Examples

```
library(shiny)
library(shiny.webawesome)

ui <- webawesomePage(
  wa_button("Open dialog", id = "open_dialog"),
  wa_dialog("Dialog body", id = "dialog")
)

server <- function(input, output, session) {
  observeEvent(input$open_dialog, {
    wa_set_property("dialog", "open", TRUE, session = session)
  })
}

app <- shinyApp(ui, server)
stopifnot(inherits(app, "shiny.appobj"))
```

```
# Run `app` interactively to launch this example application.
```

`wa_skeleton` *Create a* `wa-skeleton` *component*

Description

Generated wrapper for the Web Awesome `wa-skeleton` component. Generated from Web Awesome metadata.

Usage

```
wa_skeleton(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  dir = NULL,
  effect = NULL,
  lang = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`dir`	String. Optional Web Awesome attribute.
`effect`	Enumerated string. Allowed values: `none`, `pulse`, `sheen`. Default: `none`. Determines which effect the skeleton will use.
`lang`	String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_slider *Create a* wa-slider *component*

Description

Generated wrapper for the Web Awesome wa-slider component. When used as a Shiny input, input$<input_id> reflects the component's current value value. The Shiny value is returned as a numeric value. Generated from Web Awesome metadata.

Generated update helper for the Web Awesome wa-slider component.

Usage

```
wa_slider(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  hint = NULL,
  name = NULL,
  autofocus = NULL,
  custom_error = NULL,
  dir = NULL,
  indicator_offset = NULL,
  lang = NULL,
  max = NULL,
  max_value = NULL,
  min = NULL,
  min_value = NULL,
  orientation = NULL,
  range = NULL,
  readonly = NULL,
  size = NULL,
  step = NULL,
  tooltip_distance = NULL,
  tooltip_placement = NULL,
  with_hint = NULL,
  with_label = NULL,
  with_markers = NULL,
  with_tooltip = NULL,
  hint_slot = NULL,
  label_slot = NULL,
  reference = NULL
)
```

```
update_wa_slider(
  session,
  input_id,
  value = NULL,
  label = NULL,
  hint = NULL,
  disabled = NULL
)
```

Arguments

input_id	Input id of the component to update.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Optional value to send to the component.
disabled	Optional logical disabled state to send to the component.
label	Optional label text to send to the component.
hint	Optional hint text to send to the component.
name	String. Default: `null`. The name of the slider. This will be submitted with the form as a name/value pair.
autofocus	Boolean. Default: `FALSE`. Tells the browser to focus the slider when the page loads or a dialog is shown.
custom_error	String. Default: `null`. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
indicator_offset	Number. The starting value from which to draw the slider's fill, which is based on its current value.
lang	String. Optional Web Awesome attribute.
max	Number. Default: `100`. The maximum value allowed.
max_value	Number. Default: `50`. The maximum value of a range selection. Used only when range attribute is set.
min	Number. Default: `0`. The minimum value allowed.
min_value	Number. Default: `0`. The minimum value of a range selection. Used only when range attribute is set.
orientation	Enumerated string. Allowed values: `horizontal`, `vertical`. Default: `horizontal`. The orientation of the slider.
range	Boolean. Default: `FALSE`. Converts the slider to a range slider with two thumbs.
readonly	Boolean. Default: `FALSE`. Makes the slider a read-only field.
size	Enumerated string. Allowed values: `large`, `medium`, `small`. Default: `medium`. The slider's size.

step	Number. Default: 1. The granularity the value must adhere to when incrementing and decrementing.
tooltip_distance	Number. Default: 8. The distance of the tooltip from the slider's thumb.
tooltip_placement	Enumerated string. Allowed values: bottom, left, right, top. Default: top. The placement of the tooltip in reference to the slider's thumb.
with_hint	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.
with_label	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a label element so the server-rendered markup includes the label before the component hydrates on the client.
with_markers	Boolean. Default: FALSE. Draws markers at each step along the slider.
with_tooltip	Boolean. Default: FALSE. Draws a tooltip above the thumb when the control has focus or is dragged.
hint_slot	Text that describes how to use the input. Alternatively, you can use the hint attribute. instead.
label_slot	The slider label. Alternatively, you can use the label attribute.
reference	One or more reference labels to show visually below the slider.
session	Shiny session object.

Value

An HTML tag for the component.

Invisibly returns NULL.

Shiny Bindings

input$<input_id> reflects the component's current value value. The Shiny value is returned as a numeric value.

---

wa_spinner *Create a* wa-spinner *component*

Description

Generated wrapper for the Web Awesome wa-spinner component. Generated from Web Awesome metadata.

Usage

```
wa_spinner(..., id = NULL, class = NULL, style = NULL, dir = NULL, lang = NULL)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_split_panel *Create a* wa-split-panel *component*

Description

Generated wrapper for the Web Awesome wa-split-panel component. Generated from Web Awesome metadata.

Usage

```
wa_split_panel(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  disabled = NULL,
  dir = NULL,
  lang = NULL,
  orientation = NULL,
  position = NULL,
  position_in_pixels = NULL,
  primary = NULL,
  snap = NULL,
  snap_threshold = NULL,
  divider = NULL,
  end = NULL,
  start = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
disabled	Boolean. Default: FALSE. Disables resizing. Note that the position may still change as a result of resizing the host element.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
orientation	Enumerated string. Allowed values: horizontal, vertical. Default: horizontal. Sets the split panel's orientation.
position	Number. Default: 50. The current position of the divider from the primary panel's edge as a percentage 0-100. Defaults to 50% of the container's initial size.
position_in_pixels	Number. The current position of the divider from the primary panel's edge in pixels.
primary	Enumerated string. Allowed values: end, start. If no primary panel is designated, both panels will resize proportionally when the host element is resized. If a primary panel is designated, it will maintain its size and the other panel will grow or shrink as needed when the host element is resized.
snap	String. One or more space-separated values at which the divider should snap. Values can be in pixels or percentages, e.g. "100px 50%".
snap_threshold	Number. Default: 12. How close the divider must be to a snap point until snapping occurs.
divider	The divider. Useful for slotting in a custom icon that renders as a handle.
end	Content to place in the end panel.
start	Content to place in the start panel.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_switch *Create a* wa-switch *component*

Description

Generated wrapper for the Web Awesome wa-switch component. When used as a Shiny input, input$<input_id> reflects the component's current value value. The Shiny value is returned as a logical value. Generated from Web Awesome metadata.

Usage

```
wa_switch(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  checked = NULL,
  disabled = NULL,
  hint = NULL,
  name = NULL,
  custom_error = NULL,
  dir = NULL,
  lang = NULL,
  required = NULL,
  size = NULL,
  title = NULL,
  with_hint = NULL,
  hint_slot = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	String. The value of the switch, submitted as a name/value pair with form data.
checked	Boolean. Default: FALSE. The default value of the form control. Primarily used for resetting the form control. This wrapper argument sets the HTML checked attribute, which maps to the component's defaultChecked field/property rather than its live checked property.
disabled	Boolean. Default: FALSE. Disables the switch.

hint	String. Default: "". The switch's hint. If you need to display HTML, use the hint slot instead.
name	String. Default: null. The name of the switch, submitted as a name/value pair with form data.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
required	Boolean. Default: FALSE. Makes the switch a required field.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. The switch's size.
title	String. Default: "". Optional Web Awesome attribute.
with_hint	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.
hint_slot	Text that describes how to use the switch. Alternatively, you can use the hint attribute.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current value value. The Shiny value is returned as a logical value.

---

wa_tab *Create a* wa-tab *component*

Description

Generated wrapper for the Web Awesome wa-tab component. Generated from Web Awesome metadata.

Usage

```
wa_tab(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  disabled = NULL,
  dir = NULL,
  lang = NULL,
  panel = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
disabled	Boolean. Default: FALSE. Disables the tab and prevents selection.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
panel	String. Default: "". The name of the tab panel this tab is associated with. The panel must be located in the same tab group.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_tab_group *Create a* wa-tab-group *component*

Description

Generated wrapper for the Web Awesome wa-tab-group component. When used as a Shiny input, input$<input_id> reflects the component's current semantic active state. The Shiny value is returned as a character string. Generated from Web Awesome metadata.

Usage

```
wa_tab_group(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  activation = NULL,
  active = NULL,
  dir = NULL,
  lang = NULL,
  placement = NULL,
  without_scroll_controls = NULL,
  nav = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
activation	Enumerated string. Allowed values: auto, manual. Default: auto. When set to auto, navigating tabs with the arrow keys will instantly show the corresponding tab panel. When set to manual, the tab will receive focus but will not show until the user presses spacebar or enter.
active	String. Default: "". Sets the active tab.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
placement	Enumerated string. Allowed values: bottom, end, start, top. Default: top. The placement of the tabs.
without_scroll_controls	Boolean. Default: FALSE. Disables the scroll arrows that appear when tabs overflow.
nav	Used for grouping tabs in the tab group. Must be <wa-tab> elements. Note that <wa-tab> will set this slot on itself automatically.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current semantic active state. The Shiny value is returned as a character string.

---

wa_tab_panel *Create a* wa-tab-panel *component*

Description

Generated wrapper for the Web Awesome wa-tab-panel component. Generated from Web Awesome metadata.

Usage

```
wa_tab_panel(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  name = NULL,
  active = NULL,
  dir = NULL,
  lang = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
name	String. Default: "". The tab panel's name.
active	Boolean. Default: FALSE. When TRUE, the tab panel will be shown.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_tag *Create a* wa-tag *component*

Description

Generated wrapper for the Web Awesome wa-tag component. Generated from Web Awesome metadata.

Usage

```
wa_tag(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  appearance = NULL,
  dir = NULL,
  lang = NULL,
  pill = NULL,
  size = NULL,
  variant = NULL,
  with_remove = NULL
)
```

Arguments

`...`	Child content for the component's default slot.
`id`	Optional DOM id attribute for HTML, CSS, and JS targeting.
`class`	Optional CSS class string.
`style`	Optional inline CSS style string.
`appearance`	Enumerated string. Allowed values: `accent`, `filled`, `filled-outlined`, `outlined`. Default: `filled-outlined`. The tag's visual appearance.
`dir`	String. Optional Web Awesome attribute.
`lang`	String. Optional Web Awesome attribute.
`pill`	Boolean. Default: `FALSE`. Draws a pill-style tag with rounded edges.
`size`	Enumerated string. Allowed values: `large`, `medium`, `small`. Default: `medium`. The tag's size.
`variant`	Enumerated string. Allowed values: `brand`, `danger`, `neutral`, `success`, `warning`. Default: `neutral`. The tag's theme variant. Defaults to `neutral` if not within another element with a variant.
`with_remove`	Boolean. Default: `FALSE`. Makes the tag removable and shows a remove button.

Value

An HTML tag for the component.

Shiny Bindings

None.

wa_textarea *Create a* wa-textarea *component*

Description

Generated wrapper for the Web Awesome wa-textarea component. When used as a Shiny input, input$<input_id> reflects the component's current value value. Generated from Web Awesome metadata.

Generated update helper for the Web Awesome wa-textarea component.

Usage

```
wa_textarea(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  value = NULL,
  disabled = NULL,
  label = NULL,
  hint = NULL,
  name = NULL,
  appearance = NULL,
  autocapitalize = NULL,
  autocomplete = NULL,
  autocorrect = NULL,
  autofocus = NULL,
  custom_error = NULL,
  dir = NULL,
  enterkeyhint = NULL,
  inputmode = NULL,
  lang = NULL,
  maxlength = NULL,
  minlength = NULL,
  placeholder = NULL,
  readonly = NULL,
  required = NULL,
  resize = NULL,
  rows = NULL,
  size = NULL,
  spellcheck = NULL,
  title = NULL,
  with_count = NULL,
  with_hint = NULL,
  with_label = NULL,
  hint_slot = NULL,
```

```
  label_slot = NULL
)

update_wa_textarea(
  session,
  input_id,
  value = NULL,
  label = NULL,
  hint = NULL,
  disabled = NULL
)
```

Arguments

input_id	Input id of the component to update.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
value	Optional value to send to the component.
disabled	Optional logical disabled state to send to the component.
label	Optional label text to send to the component.
hint	Optional hint text to send to the component.
name	String. Default: null. The name of the textarea, submitted as a name/value pair with form data.
appearance	Enumerated string. Allowed values: filled, filled-outlined, outlined. Default: outlined. The textarea's visual appearance.
autocapitalize	Enumerated string. Allowed values: characters, none, off, on, sentences, words. Controls whether and how text input is automatically capitalized as it is entered by the user.
autocomplete	String. Specifies what permission the browser has to provide assistance in filling out form field values. Refer to this page on MDN for available values.
autocorrect	Boolean. Default: FALSE. Indicates whether the browser's autocorrect feature is on or off. When set as an attribute, use "off" or "on". When set as a property, use TRUE or FALSE. Because the upstream implementation treats attributes and properties differently for this field, this argument accepts TRUE, FALSE, "on", "off", NULL. TRUE emits autocorrect="on" and FALSE emits autocorrect="off". NULL omits the attribute. Accepted string values: "on" is emitted unchanged; "off" is emitted unchanged.
autofocus	Boolean. Default: FALSE. Indicates that the input should receive focus on page load.
custom_error	String. Default: null. Optional Web Awesome attribute.
dir	String. Optional Web Awesome attribute.
enterkeyhint	Enumerated string. Allowed values: done, enter, go, next, previous, search, send. Used to customize the label or icon of the Enter key on virtual keyboards.

inputmode	Enumerated string. Allowed values: decimal, email, none, numeric, search, tel, text, url. Tells the browser what type of data will be entered by the user, allowing it to display the appropriate virtual keyboard on supportive devices.
lang	String. Optional Web Awesome attribute.
maxlength	Number. The maximum length of input that will be considered valid.
minlength	Number. The minimum length of input that will be considered valid.
placeholder	String. Default: "". Placeholder text to show as a hint when the input is empty.
readonly	Boolean. Default: FALSE. Makes the textarea readonly.
required	Boolean. Default: FALSE. Makes the textarea a required field.
resize	Enumerated string. Allowed values: auto, both, horizontal, none, vertical. Default: vertical. Controls how the textarea can be resized.
rows	Number. Default: 4. The number of rows to display by default.
size	Enumerated string. Allowed values: large, medium, small. Default: medium. The textarea's size.
spellcheck	Boolean. Default: TRUE. Enables spell checking on the textarea.
title	String. Default: "". Optional Web Awesome attribute.
with_count	Boolean. Default: FALSE. Shows a character count below the textarea. When maxlength is set, shows remaining characters instead.
with_hint	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a hint element so the server-rendered markup includes the hint before the component hydrates on the client.
with_label	Boolean. Default: FALSE. Only required for SSR. Set to TRUE if you're slotting in a label element so the server-rendered markup includes the label before the component hydrates on the client.
hint_slot	Text that describes how to use the input. Alternatively, you can use the hint attribute.
label_slot	The textarea's label. Alternatively, you can use the label attribute.
session	Shiny session object.

Value

An HTML tag for the component.

Invisibly returns NULL.

Shiny Bindings

input$<input_id> reflects the component's current value value.

wa_tooltip *Create a* wa-tooltip *component*

Description

Generated wrapper for the Web Awesome wa-tooltip component. Generated from Web Awesome metadata.

Usage

```
wa_tooltip(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  disabled = NULL,
  dir = NULL,
  distance = NULL,
  `for` = NULL,
  hide_delay = NULL,
  lang = NULL,
  open = NULL,
  placement = NULL,
  show_delay = NULL,
  skidding = NULL,
  trigger = NULL,
  without_arrow = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
disabled	Boolean. Default: FALSE. Disables the tooltip so it won't show when triggered.
dir	String. Optional Web Awesome attribute.
distance	Number. Default: 8. The distance in pixels from which to offset the tooltip away from its target.
for	String. Default: null. Optional Web Awesome attribute.
hide_delay	Number. Default: 0. The amount of time to wait before hiding the tooltip when the user mouses out.
lang	String. Optional Web Awesome attribute.

open	Boolean. Default: FALSE. Indicates whether or not the tooltip is open. You can use this in lieu of the show/hide methods.
placement	Enumerated string. Allowed values: bottom, bottom-end, bottom-start, left, left-end, left-start, right, right-end, right-start, top, top-end, top-start. Default: top. The preferred placement of the tooltip. Note that the actual placement may vary as needed to keep the tooltip inside of the viewport.
show_delay	Number. Default: 150. The amount of time to wait before showing the tooltip when the user mouses in.
skidding	Number. Default: 0. The distance in pixels from which to offset the tooltip along its target.
trigger	String. Default: hover focus. Controls how the tooltip is activated. Possible options include click, hover, focus, and manual. Multiple options can be passed by separating them with a space. When manual is used, the tooltip must be activated programmatically.
without_arrow	Boolean. Default: FALSE. Removes the arrow from the tooltip.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_tree *Create a* wa-tree *component*

Description

Generated wrapper for the Web Awesome wa-tree component. When used as a Shiny input, input$<input_id> reflects the component's current semantic selectedItemIds state. The Shiny value is returned as a character vector. For stable Shiny selection values, selectable descendant wa-tree-item elements should have DOM id attributes. Generated from Web Awesome metadata.

Usage

```
wa_tree(
  input_id,
  ...,
  class = NULL,
  style = NULL,
  dir = NULL,
  lang = NULL,
  selection = NULL,
  collapse_icon = NULL,
  expand_icon = NULL
)
```

Arguments

input_id	Shiny input id for the component. This is also used as the rendered DOM id attribute.
...	Child content for the component's default slot.
class	Optional CSS class string.
style	Optional inline CSS style string.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
selection	Enumerated string. Allowed values: leaf, multiple, single. Default: single. The selection behavior of the tree. Single selection allows only one node to be selected at a time. Multiple displays checkboxes and allows more than one node to be selected. Leaf allows only leaf nodes to be selected.
collapse_icon	The icon to show when the tree item is collapsed. Works best with <wa-icon>.
expand_icon	The icon to show when the tree item is expanded. Works best with <wa-icon>.

Value

An HTML tag for the component.

Shiny Bindings

input$<input_id> reflects the component's current semantic selectedItemIds state. The Shiny value is returned as a character vector.

---

wa_tree_item *Create a* wa-tree-item *component*

Description

Generated wrapper for the Web Awesome wa-tree-item component. Generated from Web Awesome metadata.

Usage

```
wa_tree_item(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  disabled = NULL,
  dir = NULL,
  expanded = NULL,
  lang = NULL,
  lazy = NULL,
  selected = NULL,
```

```
  collapse_icon = NULL,
  expand_icon = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
disabled	Boolean. Default: FALSE. Disables the tree item.
dir	String. Optional Web Awesome attribute.
expanded	Boolean. Default: FALSE. Expands the tree item.
lang	String. Optional Web Awesome attribute.
lazy	Boolean. Default: FALSE. Enables lazy loading behavior.
selected	Boolean. Default: FALSE. Draws the tree item in a selected state.
collapse_icon	The icon to show when the tree item is collapsed.
expand_icon	The icon to show when the tree item is expanded.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

wa_version *Return the bundled Web Awesome version*

Description

Reports the Web Awesome version bundled with the current shiny.webawesome installation.

Usage

```
wa_version()
```

Value

A length-1 character vector containing the bundled Web Awesome version string.

Examples

```
wa_version()
```

wa_zoomable_frame *Create a* wa-zoomable-frame *component*

Description

Generated wrapper for the Web Awesome wa-zoomable-frame component. Generated from Web Awesome metadata.

Usage

```
wa_zoomable_frame(
  ...,
  id = NULL,
  class = NULL,
  style = NULL,
  allowfullscreen = NULL,
  dir = NULL,
  lang = NULL,
  loading = NULL,
  referrerpolicy = NULL,
  sandbox = NULL,
  src = NULL,
  srcdoc = NULL,
  with_theme_sync = NULL,
  without_controls = NULL,
  without_interaction = NULL,
  zoom = NULL,
  zoom_levels = NULL,
  zoom_in_icon = NULL,
  zoom_out_icon = NULL
)
```

Arguments

...	Child content for the component's default slot.
id	Optional DOM id attribute for HTML, CSS, and JS targeting.
class	Optional CSS class string.
style	Optional inline CSS style string.
allowfullscreen	Boolean. Default: FALSE. Allows fullscreen mode.
dir	String. Optional Web Awesome attribute.
lang	String. Optional Web Awesome attribute.
loading	Enumerated string. Allowed values: eager, lazy. Default: eager. Controls iframe loading behavior.
referrerpolicy	String. Controls referrer information.

sandbox	String. Security restrictions for the iframe.
src	String. The URL of the content to display.
srcdoc	String. Inline HTML to display.
with_theme_sync	Boolean. Default: FALSE. Enables automatic theme syncing (light/dark mode and theme selector classes) from the host document to the iframe.
without_controls	Boolean. Default: FALSE. Removes the zoom controls.
without_interaction	Boolean. Default: FALSE. Disables interaction when present.
zoom	Number. Default: 1. The current zoom of the frame, e.g. 0 = 0% and 1 = 100%.
zoom_levels	String. Default: 25% 50% 75% 100% 125% 150% 175% 200%. The zoom levels to step through when using zoom controls. This does not restrict programmatic changes to the zoom.
zoom_in_icon	The slot that contains the zoom in icon.
zoom_out_icon	The slot that contains the zoom out icon.

Value

An HTML tag for the component.

Shiny Bindings

None.

---

webawesomePage *Construct a shiny.webawesome page*

Description

Creates a minimal full-page HTML scaffold for Shiny applications that use Web Awesome components. webawesomePage() attaches the package dependency once at page level and temporarily suppresses duplicate wrapper-level attachment while evaluating its children. This function serves an entirely different role from the upstream component wa_page().

Usage

```
webawesomePage(..., title = NULL, lang = NULL, class = NULL, body_class = NULL)
```

Arguments

`...`	UI content to place in the page body.
`title`	Optional page title.
`lang`	Optional HTML `lang` attribute for the page.
`class`	Optional CSS class string for the root HTML element. This is useful for page-level Web Awesome theme, palette, and brand classes such as `"wa-theme-default wa-palette-default wa-brand-blue"` or `"wa-palette-default"`. When `class` includes a recognized non-default theme class such as `"wa-theme-awesome"` or `"wa-theme-shoelace"`, `webawesomePage()` also loads the matching bundled theme stylesheet automatically.
`body_class`	Optional CSS class string for the page body. This is useful for app-shell layout or body-level styling hooks that should stay separate from root HTML theme classes.

Details

This is a package-level Shiny helper that follows Shiny's page-helper model (`fluidPage()`, etc.) for dependency attachment and full-page scaffolding. It is intentionally separate from the upstream `wa-page` component wrapper.

Value

An HTML page scaffold with the shiny.webawesome dependency attached.

Examples

```
if (interactive()) {
  webawesomePage(
    title = "shiny.webawesome",
    class = "wa-theme-default wa-palette-default wa-brand-blue",
    wa_card("Hello from Web Awesome")
  )
}
```

Index

update_wa_input *(wa_input)*, 43
update_wa_number_input *(wa_number_input)*, 52
update_wa_select *(wa_select)*, 74
update_wa_slider *(wa_slider)*, 79
update_wa_textarea *(wa_textarea)*, 90

wa_animated_image, 3
wa_animation, 4
wa_avatar, 6
wa_badge, 7
wa_breadcrumb, 8
wa_breadcrumb_item, 9
wa_button, 10
wa_button_group, 13
wa_call_method, 14
wa_callout, 15
wa_card, 17
wa_carousel, 18
wa_carousel_item, 20
wa_checkbox, 21
wa_color_picker, 22
wa_comparison, 25
wa_container, 26
wa_copy_button, 27
wa_details, 28
wa_dialog, 30
wa_divider, 31
wa_drawer, 32
wa_dropdown, 33
wa_dropdown_item, 35
wa_format_bytes, 36
wa_format_date, 37
wa_format_number, 39
wa_icon, 40
wa_include, 42
wa_input, 43
wa_intersection_observer, 46
wa_js, 48
wa_markdown, 50
wa_mutation_observer, 51
wa_number_input, 52
wa_option, 55
wa_page, 56
wa_popover, 58
wa_popup, 60
wa_progress_bar, 63
wa_progress_ring, 64
wa_qr_code, 65
wa_radio, 66
wa_radio_group, 67
wa_rating, 69
wa_relative_time, 70
wa_resize_observer, 72
wa_scroller, 73
wa_select, 74
wa_set_property, 76
wa_skeleton, 78
wa_slider, 79
wa_spinner, 81
wa_split_panel, 82
wa_switch, 84
wa_tab, 85
wa_tab_group, 86
wa_tab_panel, 87
wa_tag, 88
wa_textarea, 90
wa_tooltip, 93
wa_tree, 94
wa_tree_item, 95
wa_version, 96
wa_zoomable_frame, 97
webawesomePage, 98