Package 'shinyds'

October 20, 2026

Title 'Shiny' Bindings for 'Designsystemet' Components

Version 0.7.2

Description Provides 'R' wrappers for the 'Designsystemet' component library <https://designsystemet.no>, enabling use of Norwegian government design system components in 'Shiny' applications. Includes web components and CSS-based HTML components with full 'Shiny' input binding support.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.4.0)

Imports htmltools (>= 0.5.0), shiny (>= 1.7.0)

Suggests testthat (>= 3.0.0), shinytest2, chromote, jsonlite, withr, bslib, quarto

VignetteBuilder quarto

Config/testthat/edition 3

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.0

URL https://github.com/novica/shinyds

BugReports https://github.com/novica/shinyds/issues

NeedsCompilation no

Author Novica Nakov [aut, cre, cph] (ORCID: <https://orcid.org/0009-0005-7773-7718>)

Maintainer Novica Nakov <nnovica@gmail.com>

Repository CRAN

Date/Publication 2026-10-20 17:20:02 UTC

Contents

ds_alert 3
ds_avatar 4

ds_avatar_stack 4
ds_badge 5
ds_badge_position 6
ds_breadcrumbs 7
ds_button 8
ds_card 9
ds_card_block 10
ds_checkbox 10
ds_chip 11
ds_chip_group 12
ds_combobox 12
ds_dependencies 13
ds_details 14
ds_dialog 14
ds_divider 15
ds_dropdown 16
ds_error_summary 16
ds_field 17
ds_fieldset 18
ds_heading 18
ds_input 19
ds_label 20
ds_link 20
ds_list 21
ds_list_item 22
ds_pagination 22
ds_paragraph 23
ds_popover 24
ds_radio 24
ds_radio_group 25
ds_search 26
ds_select 27
ds_skeleton 28
ds_skip_link 28
ds_spinner 29
ds_suggestion 29
ds_tab 30
ds_table 31
ds_tablist 32
ds_tabpanel 32
ds_tabs 33
ds_tag 34
ds_tbody 35
ds_td 35
ds_textarea 36
ds_th 37
ds_thead 37
ds_theme 38

ds_toggle_group 39
ds_tooltip 39
ds_tr 40
ds_validation_message 41
hide_ds_dialog 41
show_ds_dialog 42
update_ds_checkbox 43
update_ds_input 43
update_ds_pagination 44
update_ds_radio_group 45
update_ds_select 46
update_ds_suggestion 46
update_ds_tabs 47
use_designsystemet 48

Index **49**

`ds_alert` *Alert Component*

Description

Create an alert message using Designsystemet styles.

Usage

```
ds_alert(
  ...,
  variant = c("info", "warning", "danger", "success"),
  class = NULL
)
```

Arguments

`...`	Alert content
`variant`	Alert variant ("info", "warning", "danger", "success")
`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_alert("This is an informational message.", variant = "info")
ds_alert("Warning: Please review before continuing.", variant = "warning")
ds_alert("Error: Something went wrong.", variant = "danger")
ds_alert("Success! Your changes were saved.", variant = "success")
```

ds_avatar *Avatar Component*

Description

Create an avatar using Designsystemet styles.

Usage

```
ds_avatar(..., variant = NULL, size = NULL, class = NULL)
```

Arguments

...	Avatar content (typically an image or initials)
variant	Avatar variant
size	Size variant ("sm", "md", "lg")
class	Additional CSS classes

Details

Initials can be passed directly as text (e.g. "AB"); they scale automatically with size. The data-initials attribute is deprecated upstream and should not be used.

Value

A Shiny tag object

Examples

```
ds_avatar("AB", size = "md")
```

ds_avatar_stack *Avatar Stack Component*

Description

Create a stack of avatars using Designsystemet styles. The stack is rendered as a <ul> and each avatar is wrapped in an <li> automatically (children that are already <li> tags are left as is).

Usage

```
ds_avatar_stack(..., suffix = NULL, class = NULL)
```

Arguments

...	Avatar elements. Named arguments become attributes on the .
suffix	Optional text shown after the avatars, e.g. "+3" to indicate additional members. Rendered as a trailing .
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_avatar_stack(
  ds_avatar("AB"),
  ds_avatar("CD"),
  ds_avatar("EF")
)

ds_avatar_stack(
  ds_avatar("AB"),
  ds_avatar("CD"),
  suffix = "+2"
)
```

---

ds_badge *Badge Component*

Description

Create a count-indicator badge using Designsystemet styles.

Usage

```
ds_badge(
  count = NULL,
  color = NULL,
  variant = NULL,
  size = NULL,
  class = NULL,
  ...
)
```

Arguments

count	Numeric or character value shown inside the badge circle. Pass NULL for a plain dot.
color	Color context: one of "info", "success", "warning", "danger", "accent", "neutral", "brand1", "brand2". Maps to the DS data-color attribute.
variant	Structural variant: "tinted" for a muted background. Colour variants belong in color, not here.
size	Size variant ("sm", "md", "lg").
class	Additional CSS classes.
...	Additional attributes passed to the .

Details

The badge is a small circle that sits on top of another element (e.g. a button) to show a numeric count. Wrap the target element with a ds_badge_position() container and place ds_badge() inside it.

The circle text comes from count; ... is the accessible label for the element being badged (rendered as invisible span text).

Value

A Shiny tag object

Examples

```
# Standalone dot with no count
ds_badge(color = "success")

# Count badge on a button (typical use — see ds_badge_position())
## Not run:
ds_badge_position(
  ds_button("Inbox"),
  ds_badge(count = 4, color = "danger")
)

## End(Not run)
```

---

ds_badge_position *Badge Position Wrapper*

Description

Wraps an element so that a ds_badge() can be positioned on top of it.

Usage

```
ds_badge_position(..., placement = "top-right", overlap = NULL, class = NULL)
```

Arguments

...	The target element followed by a ds_badge().
placement	Badge placement: "top-right" (default), "top-left", "bottom-right", "bottom-left".
overlap	Set to "circle" when the target is a circular avatar.
class	Additional CSS classes.

Value

A Shiny tag object

Examples

```
ds_badge_position(
  ds_button("Inbox"),
  ds_badge(count = 3, color = "danger"),
  placement = "top-right"
)
```

---

ds_breadcrumbs *Breadcrumbs Component*

Description

Create a breadcrumb navigation trail.

Usage

```
ds_breadcrumbs(..., class = NULL)
```

Arguments

...	Child elements (typically list items with links)
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_breadcrumbs(
  htmltools::tags$ol(
    htmltools::tags$li(ds_link("Home", href = "/")),
    htmltools::tags$li(ds_link("Products", href = "/products")),
    htmltools::tags$li(htmltools::tags$span("Current Page"))
  )
)
```

ds_button *Button Component*

Description

Create a styled button using Designsystemet styles. `ds_action_button()` is a drop-in replacement for `shiny::actionButton()` with the same `(inputId, label, ...)` argument order.

Usage

```
ds_button(
  label,
  inputId = NULL,
  variant = c("primary", "secondary", "tertiary"),
  size = NULL,
  icon = FALSE,
  loading = FALSE,
  disabled = FALSE,
  fullwidth = FALSE,
  type = "button",
  ...
)

ds_action_button(inputId, label, ...)
```

Arguments

`label`	The button label.
`inputId`	Input ID (first argument, matching `shiny::actionButton`).
`variant`	Button variant: `"primary"`, `"secondary"`, or `"tertiary"`.
`size`	Size variant: `"sm"`, `"md"`, or `"lg"`.
`icon`	If `TRUE`, creates an icon-only button.
`loading`	If `TRUE`, shows loading state.
`disabled`	If `TRUE`, disables the button.
`fullwidth`	If `TRUE`, makes the button full width.
`type`	Button type: `"button"`, `"submit"`, or `"reset"`.
`...`	Additional attributes.

Value

A Shiny tag object.

Examples

```
ds_button("Click me", inputId = "btn")
ds_button("Submit", variant = "primary", type = "submit")
ds_button("Cancel", variant = "secondary")
ds_action_button("btn", "Click me")
```

`ds_card` *Card Component*

Description

Create a styled card container using Designsystemet styles.

Usage

```
ds_card(..., variant = NULL, class = NULL)
```

Arguments

`...`	Child elements
`variant`	Card variant ("default", "tinted")
`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_card(
  ds_card_block(
    ds_heading("Card Title", level = 3),
    ds_paragraph("Card content goes here.")
  )
)
```

ds_card_block *Card Block*

Description

A block section within a card with proper padding.

Usage

```
ds_card_block(..., class = NULL)
```

Arguments

...	Child elements
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_card_block(ds_paragraph("Card content."))
```

ds_checkbox *Checkbox Component*

Description

Create a styled checkbox using Designsystemet styles.

Usage

```
ds_checkbox(inputId, label, value = FALSE, size = NULL, disabled = FALSE, ...)
```

Arguments

inputId	The input slot for Shiny reactivity
label	The checkbox label
value	Initial checked state
size	Size variant ("sm", "md", "lg")
disabled	If TRUE, disables the checkbox
...	Additional attributes

Value

A Shiny tag object

Examples

```
ds_checkbox("agree", "I agree to the terms", value = FALSE)
```

---

ds_chip *Chip Component*

Description

Create an interactive chip using Designsystemet styles.

Usage

```
ds_chip(..., variant = NULL, size = NULL, selected = FALSE, class = NULL)
```

Arguments

`...`	Chip content
`variant`	Chip variant
`size`	Size variant ("sm", "md", "lg")
`selected`	If TRUE, shows selected state
`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_chip("React")
ds_chip("Vue", selected = TRUE)
```

ds_chip_group *Chip Group Component*

Description

Create a reactive group of toggle chips. Selected chip values are reported to Shiny as a character vector via `input[[inputId]]`.

Usage

```
ds_chip_group(inputId, ..., class = NULL)
```

Arguments

`inputId`	Shiny input ID
`...`	`ds_chip()` elements
`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
## Not run:
ds_chip_group(
  "languages",
  ds_chip("R",      value = "r",      selected = TRUE),
  ds_chip("Python", value = "python"),
  ds_chip("Julia",  value = "julia")
)

## End(Not run)
```

ds_combobox *Combobox Component*

Description

Create a combobox container using Designsystemet styles. Unlike `ds_suggestion`, this is a pure CSS component — keyboard navigation and dropdown behaviour must be implemented by the caller.

Usage

```
ds_combobox(inputId, ..., size = NULL, class = NULL)
```

Arguments

inputId	The input slot for Shiny reactivity
...	Child elements (input wrapper, options wrapper, etc.)
size	Size variant ("sm", "md", "lg")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_combobox("mycombo",
  htmltools::tags$div(class = "ds-combobox__input__wrapper",
    htmltools::tags$input(class = "ds-combobox__input", type = "text")
  ),
  htmltools::tags$div(class = "ds-combobox__options-wrapper")
)
```

---

ds_dependencies *Designsystemet Dependencies*

Description

Returns the HTML dependencies required for Designsystemet components.

Usage

```
ds_dependencies()
```

Value

A list of htmltools htmlDependency objects

Examples

```
ds_dependencies()
```

ds_details *Details/Accordion Component*

Description

Create an expandable details section using Designsystemet styles.

Usage

```
ds_details(summary, ..., open = FALSE, class = NULL)
```

Arguments

summary	The summary/title text shown when collapsed
...	Content shown when expanded
open	If TRUE, the details are initially open
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_details(
  summary = "Click to expand",
  ds_paragraph("Hidden content goes here.")
)
```

ds_dialog *Dialog Component*

Description

Create a modal dialog using Designsystemet styles. Place `ds_dialog()` in your UI and open/close it from the server with `show_ds_dialog()` and `hide_ds_dialog()`.

Usage

```
ds_dialog(..., id = NULL, class = NULL)
```

Arguments

...	Dialog content
id	Dialog ID for opening/closing via `show_ds_dialog()`
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_dialog(
  id = "my-dialog",
  ds_heading("Dialog Title", level = 2),
  ds_paragraph("Dialog content here."),
  ds_button("Close", onclick = "this.closest('dialog').close()")
)
```

`ds_divider` *Divider Component*

Description

Create a horizontal divider using Designsystemet styles.

Usage

```
ds_divider(class = NULL)
```

Arguments

`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_divider()
```

ds_dropdown *Dropdown Component*

Description

Create a dropdown menu using Designsystemet styles.

Usage

```
ds_dropdown(trigger, ..., id = NULL, class = NULL)
```

Arguments

trigger	The trigger element (usually a button)
...	Dropdown panel content
id	Optional ID for the dropdown panel. Auto-generated if omitted.
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_dropdown(
  trigger = ds_button("Options"),
  ds_list(ds_list_item("Edit"), ds_list_item("Delete"))
)
```

ds_error_summary *Error Summary Component*

Description

Display a summary of form validation errors.

Usage

```
ds_error_summary(..., heading = "Feil i skjema", class = NULL)
```

Arguments

...	Error list items
heading	The heading text for the error summary
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_error_summary(
  heading = "There were errors in the form",
  htmltools::tags$li(ds_link("Name is required", href = "#name")),
  htmltools::tags$li(ds_link("Email is invalid", href = "#email"))
)
```

---

ds_field *Field Container*

Description

A container for form field elements with proper spacing and layout.

Usage

```
ds_field(..., size = NULL, class = NULL)
```

Arguments

...	Child elements (label, input, validation message, etc.)
size	Size variant ("sm", "md", "lg")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_field(
  ds_label("Email", `for` = "email"),
  ds_input("email", type = "email", placeholder = "your@email.com"),
  ds_validation_message("Please enter a valid email")
)
```

ds_fieldset *Fieldset Component*

Description

Group related form fields with a legend using Designsystemet styles.

Usage

```
ds_fieldset(legend, ..., class = NULL)
```

Arguments

legend	The legend text for the fieldset group
...	Child elements (fields, inputs, etc.)
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_fieldset("Preferences",
  ds_checkbox("opt1", "Option 1"),
  ds_checkbox("opt2", "Option 2")
)
```

ds_heading *Heading Component*

Description

Create a styled heading using Designsystemet styles.

Usage

```
ds_heading(..., level = 2, size = NULL, class = NULL)
```

Arguments

...	Heading content
level	Heading level (1-6)
size	Size variant ("2xs", "xs", "sm", "md", "lg", "xl", "2xl")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_heading("Page Title", level = 1)
ds_heading("Section Title", level = 2, size = "lg")
```

`ds_input` *Text Input Component*

Description

Create a styled text input using Designsystemet styles.

Usage

```
ds_input(
  inputId,
  value = "",
  type = "text",
  placeholder = NULL,
  size = NULL,
  readonly = FALSE,
  disabled = FALSE,
  invalid = FALSE,
  ...
)
```

Arguments

`inputId`	The input slot for Shiny reactivity
`value`	Initial value
`type`	Input type ("text", "email", "password", "number", "tel", "url")
`placeholder`	Placeholder text
`size`	Size variant ("sm", "md", "lg")
`readonly`	If TRUE, makes the input read-only
`disabled`	If TRUE, disables the input
`invalid`	If TRUE, shows invalid state
`...`	Additional attributes

Value

A Shiny tag object

Examples

```
ds_input("name", placeholder = "Enter your name")
ds_input("email", type = "email", placeholder = "your@email.com")
```

---

ds_label *Label Component*

Description

Create a styled form label using Designsystemet styles.

Usage

```
ds_label(..., `for` = NULL, class = NULL)
```

Arguments

...	Label content
for	The ID of the input this label is for
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_label("Email address", `for` = "email")
```

---

ds_link *Link Component*

Description

Create a styled link using Designsystemet styles.

Usage

```
ds_link(..., href = "#", class = NULL)
```

Arguments

...	Link content
href	Link URL
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_link("Visit our website", href = "https://example.com")
```

`ds_list` *List Component*

Description

Create a styled list using Designsystemet styles.

Usage

```
ds_list(..., ordered = FALSE, class = NULL)
```

Arguments

`...`	List items
`ordered`	If TRUE, creates an ordered list
`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_list(
  ds_list_item("First item"),
  ds_list_item("Second item"),
  ds_list_item("Third item")
)
```

ds_list_item *List Item*

Description

Create a list item.

Usage

```
ds_list_item(...)
```

Arguments

...	Item content

Value

A Shiny tag object

Examples

```
ds_list_item("First item")
```

ds_pagination *Pagination Component*

Description

Create a pagination component for navigating through pages.

Usage

```
ds_pagination(
  inputId,
  current = 1,
  total,
  label = "Sidenavigasjon",
  href = NULL,
  ...
)
```

Arguments

inputId	The input slot for Shiny reactivity
current	The current page number (1-indexed)
total	The total number of pages
label	Accessibility label for the pagination nav
href	Optional URL template for page links (use sprintf style, e.g. "page?p=1")
...	Additional attributes

Value

A Shiny tag object

Examples

```
ds_pagination("mypage", current = 1, total = 10)
```

---

ds_paragraph *Paragraph Component*

Description

Create a styled paragraph using Designsystemet styles.

Usage

```
ds_paragraph(..., size = NULL, class = NULL)
```

Arguments

...	Paragraph content
size	Size variant ("sm", "md", "lg")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_paragraph("This is a paragraph of text.")
```

ds_popover *Popover Component*

Description

Create a popover using Designsystemet styles.

Usage

```
ds_popover(..., variant = NULL, class = NULL)
```

Arguments

...	Popover content
variant	Popover variant ("default", "tinted")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_popover("Helpful information about this field.")
ds_popover("Highlighted note", variant = "tinted")
```

ds_radio *Radio Button Component*

Description

Create a single styled radio button using Designsystemet styles. This is the building block used by ds_radio_group() — a standalone ds_radio() is not reactive on its own. To get a Shiny input value out of a group of radio buttons, use ds_radio_group() instead.

Usage

```
ds_radio(
  inputId,
  label,
  value,
  name,
  checked = FALSE,
  size = NULL,
  disabled = FALSE,
  ...
)
```

Arguments

inputId	The id/for pair linking this radio to its label
label	The radio button label
value	The value for this radio button
name	The name grouping radio buttons together
checked	If TRUE, this radio is initially selected
size	Size variant ("sm", "md", "lg")
disabled	If TRUE, disables the radio button
...	Additional attributes

Value

A Shiny tag object

Examples

```
ds_radio("opt_a", label = "Option A", value = "a", name = "opts")
```

---

ds_radio_group *Radio Group Component*

Description

Create a reactive group of radio buttons. The value of whichever ds_radio() is checked is reported to Shiny as a single value via input[[inputId]] — mirroring how Shiny's own radioButtons() works.

Usage

```
ds_radio_group(
  inputId,
  label,
  choices,
  selected = NULL,
  size = NULL,
  disabled = FALSE,
  class = NULL
)
```

Arguments

inputId	Shiny input ID
label	The legend text for the group
choices	Named vector of choices (names are labels, values are values)
selected	Initially selected value. Defaults to the first choice.
size	Size variant ("sm", "md", "lg")
disabled	If TRUE, disables every radio button in the group
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_radio_group("opts",
  label = "Choose one",
  choices = c("Choice A" = "a", "Choice B" = "b", "Choice C" = "c"),
  selected = "a"
)
```

---

ds_search *Search Component*

Description

Create a search input using Designsystemet styles.

Usage

```
ds_search(
  inputId,
  value = "",
  placeholder = NULL,
  size = NULL,
  class = NULL,
  ...
)
```

Arguments

inputId	The input slot for Shiny reactivity
value	Initial value
placeholder	Placeholder text
size	Size variant ("sm", "md", "lg")
class	Additional CSS classes
...	Additional attributes

Value

A Shiny tag object

Examples

```
ds_search("search", placeholder = "Search. . . ")
```

`ds_select` *Select Component*

Description

Create a styled select dropdown using Designsystemet styles.

Usage

```
ds_select(
  inputId,
  choices,
  selected = NULL,
  size = NULL,
  disabled = FALSE,
  ...
)
```

Arguments

`inputId`	The input slot for Shiny reactivity
`choices`	Named vector of choices (names are labels, values are values)
`selected`	Initially selected value
`size`	Size variant ("sm", "md", "lg")
`disabled`	If TRUE, disables the select
`...`	Additional attributes

Value

A Shiny tag object

Examples

```
ds_select("country",
  choices = c("Norway" = "no", "Sweden" = "se", "Denmark" = "dk"),
  selected = "no"
)
```

ds_skeleton *Skeleton Component*

Description

Create a loading skeleton placeholder using Designsystemet styles.

Usage

```
ds_skeleton(variant = "text", width = NULL, height = NULL, class = NULL)
```

Arguments

variant	Skeleton variant ("text", "circle", "rectangle")
width	Width of the skeleton
height	Height of the skeleton
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_skeleton(variant = "text", width = "200px")
ds_skeleton(variant = "circle", width = "50px", height = "50px")
```

ds_skip_link *Skip Link Component*

Description

Create an accessibility skip link using Designsystemet styles.

Usage

```
ds_skip_link(..., href = "#main", class = NULL)
```

Arguments

...	Link content
href	Target anchor
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_skip_link("Skip to main content", href = "#main")
```

---

ds_spinner *Spinner Component*

Description

Create a loading spinner using Designsystemet styles.

Usage

```
ds_spinner(title = "Laster...", size = NULL, class = NULL)
```

Arguments

title	Accessibility title for the spinner
size	Size variant ("sm", "md", "lg")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_spinner(title = "Loading...")
```

---

ds_suggestion *Suggestion/Autocomplete Component*

Description

Create an autocomplete input with suggestions.

Usage

```
ds_suggestion(
  inputId,
  choices = NULL,
  value = "",
  placeholder = NULL,
  size = NULL,
  ...
)
```

Arguments

inputId	The input slot for Shiny reactivity
choices	Character vector of suggestion choices
value	Initial value
placeholder	Placeholder text
size	Size variant ("sm", "md", "lg")
...	Additional attributes

Value

A Shiny tag object

Examples

```
ds_suggestion("city",
  choices = c("Oslo", "Bergen", "Trondheim", "Stavanger"),
  placeholder = "Search for a city..."
)
```

---

ds_tab *Tab Button*

Description

A single tab button within ds_tablist.

Usage

```
ds_tab(label, value, selected = FALSE, ...)
```

Arguments

label	The tab label text
value	The value to return when this tab is selected
selected	Whether this tab is initially selected
...	Additional attributes

Value

A Shiny tag object

Examples

```
ds_tab("Overview", value = "overview", selected = TRUE)
```

---

ds_table *Table Component*

Description

Create a styled table using Designsystemet styles.

Usage

```
ds_table(..., size = NULL, class = NULL)
```

Arguments

`...`	Table content (thead, tbody, etc.)
`size`	Size variant ("sm", "md", "lg")
`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_table(
  ds_thead(
    ds_tr(
      ds_th("Name"),
      ds_th("Email")
    )
  ),
  ds_tbody(
    ds_tr(
      ds_td("John Doe"),
      ds_td("john@example.com")
    )
  )
)
```

ds_tablist *Tab List Container*

Description

Container for tab buttons within ds_tabs.

Usage

```
ds_tablist(..., class = NULL)
```

Arguments

...	Child elements (ds_tab elements)
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_tablist(
  ds_tab("First", value = "tab1", selected = TRUE),
  ds_tab("Second", value = "tab2")
)
```

ds_tabpanel *Tab Panel*

Description

Content panel for a tab.

Usage

```
ds_tabpanel(..., value, class = NULL)
```

Arguments

...	Panel content
value	The value that matches the corresponding ds_tab
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_tabpanel(value = "overview", ds_paragraph("Content here."))
```

`ds_tabs` *Tabs Component*

Description

Create a tabbed interface using Designsystemet tabs.

Usage

```
ds_tabs(inputId, ..., class = NULL)
```

Arguments

`inputId`	The input slot for Shiny reactivity
`...`	Child elements (typically ds_tablist and ds_tabpanel elements)
`class`	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_tabs("mytabs",
  ds_tablist(
    ds_tab("First", value = "tab1", selected = TRUE),
    ds_tab("Second", value = "tab2")
  ),
  ds_tabpanel(value = "tab1", "Content 1"),
  ds_tabpanel(value = "tab2", "Content 2")
)
```

ds_tag *Tag Component*

Description

Create a text label tag using Designsystemet styles.

Usage

```
ds_tag(..., color = NULL, variant = NULL, size = NULL, class = NULL)
```

Arguments

`...`	Tag content (text or child elements).
`color`	Color context: one of `"info"`, `"success"`, `"warning"`, `"danger"`, `"accent"`, `"neutral"`, `"brand1"`, `"brand2"`. Maps to the DS `data-color` attribute.
`variant`	Structural variant: `"outline"` adds a subtle border.
`size`	Size variant (`"sm"`, `"md"`, `"lg"`).
`class`	Additional CSS classes.

Details

Tags are the right component for standalone text status labels such as "New", "Beta" or "Active". For numeric count indicators, use ds_badge.

Value

A Shiny tag object

Examples

```
ds_tag("New")
ds_tag("Active",  color = "success")
ds_tag("Warning", color = "warning", variant = "outline")
```

ds_tbody *Table Body*

Description

Create a table body section.

Usage

```
ds_tbody(...)
```

Arguments

...	Body rows

Value

A Shiny tag object

Examples

```
ds_tbody(ds_tr(ds_td("John"), ds_td("john@example.com")))
```

ds_td *Table Data Cell*

Description

Create a table data cell.

Usage

```
ds_td(...)
```

Arguments

...	Cell content

Value

A Shiny tag object

Examples

```
ds_td("Cell content")
```

ds_textarea *Textarea Component*

Description

Create a styled textarea using Designsystemet styles.

Usage

```
ds_textarea(
  inputId,
  value = "",
  placeholder = NULL,
  rows = 3,
  size = NULL,
  readonly = FALSE,
  disabled = FALSE,
  invalid = FALSE,
  ...
)
```

Arguments

inputId	The input slot for Shiny reactivity
value	Initial value
placeholder	Placeholder text
rows	Number of visible rows
size	Size variant ("sm", "md", "lg")
readonly	If TRUE, makes the textarea read-only
disabled	If TRUE, disables the textarea
invalid	If TRUE, shows invalid state
...	Additional attributes

Value

A Shiny tag object

Examples

```
ds_textarea("msg", placeholder = "Type here...", rows = 4)
```

ds_th *Table Header Cell*

Description

Create a table header cell.

Usage

```
ds_th(..., scope = "col")
```

Arguments

...	Cell content
scope	Cell scope ("col", "row")

Value

A Shiny tag object

Examples

```
ds_th("Column header")
```

ds_thead *Table Header*

Description

Create a table header section.

Usage

```
ds_thead(...)
```

Arguments

...	Header rows

Value

A Shiny tag object

Examples

```
ds_thead(ds_tr(ds_th("Name"), ds_th("Email")))
```

ds_theme *Designsystemet Theme*

Description

Override Designsystemet CSS tokens for color, typography, and spacing. Call inside your UI alongside `use_designsystemet()`.

Usage

```
ds_theme(
  color = NULL,
  font_size = NULL,
  border_radius = NULL,
  button_padding = NULL,
  ...
)
```

Arguments

`color`	Global color context. One of `"default"`, `"brand1"`, `"brand2"`, `"neutral"`, `"success"`, `"warning"`, `"danger"`, `"info"`. Sets `data-color` on `<html>`.
`font_size`	Base font size as a CSS value (e.g. `"16px"`, `"1rem"`). Overrides `--ds-font-size-base` on `:root`.
`border_radius`	Border radius as a CSS value (e.g. `"4px"`, `"0"`). Overrides `--ds-border-radius-md` on `:root`.
`button_padding`	Padding for buttons as a CSS shorthand (e.g. `"0.4rem 0.9rem"`). Overrides `--dsc-button-padding` on `:root`.
`...`	Named CSS custom property overrides in the form `"--ds-token-name" = "value"`. Applied to `:root`.

Details

Color palette tokens follow the Designsystemet naming convention. Use `data-color="brand2"` on a container to switch its color context, or set `color` here to change the global default.

Value

A `<style>` tag applied to `:root` and optionally a JS snippet to set `data-color` on `<html>`.

Examples

```
ui <- bslib::page_fluid(
  use_designsystemet(),
  ds_theme(color = "brand1", border_radius = "2px", button_padding = "0.3rem 0.8rem"),
  ds_button("Click me", inputId = "btn")
)
```

ds_toggle_group *Toggle Group Component*

Description

Create a toggle button group using Designsystemet styles.

Usage

```
ds_toggle_group(inputId, ..., size = NULL, class = NULL)
```

Arguments

inputId	The input slot for Shiny reactivity
...	Toggle buttons
size	Size variant ("sm", "md", "lg")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
## Not run:
ds_toggle_group(
  "view_mode",
  tags$button(class = "ds-button", `data-variant` = "secondary",
              `aria-pressed` = "true",  value = "list", "List"),
  tags$button(class = "ds-button", `data-variant` = "secondary",
              `aria-pressed` = "false", value = "grid", "Grid")
)

## End(Not run)
```

ds_tooltip *Tooltip*

Description

Add a tooltip to an element by attaching `data-tooltip` attributes.

Usage

```
ds_tooltip(tag, text, placement = "top")
```

Arguments

tag	The element to attach the tooltip to.
text	The tooltip text.
placement	Tooltip placement: "top", "bottom", "left", or "right".

Value

The element with tooltip attributes added.

Examples

```
ds_tooltip(
  ds_button("Hover me"),
  text = "This is helpful information"
)
```

---

ds_tr *Table Row*

Description

Create a table row.

Usage

```
ds_tr(...)
```

Arguments

...	Row cells

Value

A Shiny tag object

Examples

```
ds_tr(ds_td("Cell 1"), ds_td("Cell 2"))
```

ds_validation_message *Validation Message Component*

Description

Create a validation message using Designsystemet styles.

Usage

```
ds_validation_message(..., variant = "error", class = NULL)
```

Arguments

...	Message content
variant	Message variant ("error", "warning")
class	Additional CSS classes

Value

A Shiny tag object

Examples

```
ds_validation_message("This field is required")
```

hide_ds_dialog *Close a dialog from the server*

Description

Sends a message to the browser to call `close()` on a `ds_dialog()` element.

Usage

```
hide_ds_dialog(id, session = shiny::getDefaultReactiveDomain())
```

Arguments

id	The `id` of the `ds_dialog()` to close.
session	Shiny session object.

Value

Called for side effects.

Examples

```
## Not run:
observeEvent(input$cancel_btn, {
  hide_ds_dialog("confirm-dialog")
})

## End(Not run)
```

`show_ds_dialog` *Open a dialog from the server*

Description

Sends a message to the browser to call `showModal()` on a `ds_dialog()` element already present in the UI.

Usage

```
show_ds_dialog(id, session = shiny::getDefaultReactiveDomain())
```

Arguments

`id`	The `id` of the `ds_dialog()` to open.
`session`	Shiny session object.

Value

Called for side effects.

Examples

```
## Not run:
observeEvent(input$open_btn, {
  show_ds_dialog("confirm-dialog")
})

## End(Not run)
```

update_ds_checkbox *Update Checkbox*

Description

Change the checked state from the server.

Usage

```
update_ds_checkbox(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  value = NULL
)
```

Arguments

session	The Shiny session object
inputId	The input ID
value	The new checked state (TRUE/FALSE)

Value

Called for its side effect. Returns NULL invisibly.

Examples

```
## Not run:
update_ds_checkbox(session, "agree", value = FALSE)

## End(Not run)
```

update_ds_input *Update Text Input*

Description

Change the value from the server.

Usage

```
update_ds_input(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  value = NULL
)
```

Arguments

session	The Shiny session object
inputId	The input ID
value	The new value

Value

Called for its side effect. Returns NULL invisibly.

Examples

```
## Not run:
update_ds_input(session, "name", value = "")

## End(Not run)
```

---

update_ds_pagination *Update Pagination Input*

Description

Change the current page or total pages from the server.

Usage

```
update_ds_pagination(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  current = NULL,
  total = NULL
)
```

Arguments

session	The Shiny session object
inputId	The input ID of the pagination component
current	The new current page number
total	The new total number of pages

Value

Called for its side effect. Returns NULL invisibly.

Examples

```
## Not run:
update_ds_pagination(session, "pager", current = 1)

## End(Not run)
```

`update_ds_radio_group` *Update Radio Group*

Description

Change the selected value of a `ds_radio_group()` from the server.

Usage

```
update_ds_radio_group(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  value = NULL
)
```

Arguments

`session`	The Shiny session object
`inputId`	The input ID
`value`	The new selected value

Value

Called for its side effect. Returns `NULL` invisibly.

Examples

```
## Not run:
update_ds_radio_group(session, "opts", value = "b")

## End(Not run)
```

update_ds_select *Update Select*

Description

Change the selected value or choices from the server.

Usage

```
update_ds_select(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  value = NULL,
  choices = NULL
)
```

Arguments

session	The Shiny session object
inputId	The input ID
value	The new selected value
choices	New named vector of choices

Value

Called for its side effect. Returns NULL invisibly.

Examples

```
## Not run:
update_ds_select(session, "country", value = "no")

## End(Not run)
```

update_ds_suggestion *Update Suggestion Input*

Description

Change the value or choices from the server.

Usage

```
update_ds_suggestion(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  value = NULL,
  choices = NULL
)
```

Arguments

session	The Shiny session object
inputId	The input ID of the suggestion component
value	The new value
choices	New character vector of choices

Value

Called for its side effect. Returns NULL invisibly.

Examples

```
## Not run:
update_ds_suggestion(session, "fruit", value = "Apple")

## End(Not run)
```

---

update_ds_tabs *Update Tabs Input*

Description

Change the selected tab from the server.

Usage

```
update_ds_tabs(
  session = shiny::getDefaultReactiveDomain(),
  inputId,
  selected = NULL
)
```

Arguments

session	The Shiny session object
inputId	The input ID of the tabs component
selected	The value of the tab to select

Value

Called for its side effect. Returns NULL invisibly.

Examples

```
## Not run:
update_ds_tabs(session, "my_tabs", selected = "overview")

## End(Not run)
```

---

use_designsystemet *Use Designsystemet in a Shiny App*

Description

Include this function in your UI to load all necessary Designsystemet CSS and JavaScript dependencies.

Usage

```
use_designsystemet(color_scheme = "light")
```

Arguments

color_scheme	Color scheme to apply to the page. One of "light" or "dark". Sets the data-color-scheme attribute on <html>, which activates Designsystemet's CSS custom property tokens.

Value

A tagList containing the dependencies

Examples

```
ui <- bslib::page_fluid(
  use_designsystemet(),
  ds_button("Click me", inputId = "btn")
)
```

Index

ds_action_button *(ds_button)*, 8
ds_alert, 3
ds_avatar, 4
ds_avatar_stack, 4
ds_badge, 5, *34*
ds_badge_position, 6
ds_breadcrumbs, 7
ds_button, 8
ds_card, 9
ds_card_block, 10
ds_checkbox, 10
ds_chip, 11
ds_chip_group, 12
ds_combobox, 12
ds_dependencies, 13
ds_details, 14
ds_dialog, 14
ds_divider, 15
ds_dropdown, 16
ds_error_summary, 16
ds_field, 17
ds_fieldset, 18
ds_heading, 18
ds_input, 19
ds_label, 20
ds_link, 20
ds_list, 21
ds_list_item, 22
ds_pagination, 22
ds_paragraph, 23
ds_popover, 24
ds_radio, 24
ds_radio(), *25*
ds_radio_group, 25
ds_radio_group(), *24*, *45*
ds_search, 26
ds_select, 27
ds_skeleton, 28
ds_skip_link, 28
ds_spinner, 29
ds_suggestion, 29
ds_tab, 30
ds_table, 31
ds_tablist, 32
ds_tabpanel, 32
ds_tabs, 33
ds_tag, 34
ds_tbody, 35
ds_td, 35
ds_textarea, 36
ds_th, 37
ds_thead, 37
ds_theme, 38
ds_toggle_group, 39
ds_tooltip, 39
ds_tr, 40
ds_validation_message, 41

hide_ds_dialog, 41
hide_ds_dialog(), *14*

show_ds_dialog, 42
show_ds_dialog(), *14*

update_ds_checkbox, 43
update_ds_input, 43
update_ds_pagination, 44
update_ds_radio_group, 45
update_ds_select, 46
update_ds_suggestion, 46
update_ds_tabs, 47
use_designsystemet, 48
use_designsystemet(), *38*