

# Package: shinyds (via r-universe)

June 4, 2026

**Title** 'Shiny' Bindings for Designsystemet Components

**Version** 0.2.2

**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

**RoxygenNote** 8.0.0

**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** 8.0.0

**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](mailto:nnovica@gmail.com)>

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-04 14:12:23 UTC

**RemoteUrl** <https://github.com/cran/shinyds>

**RemoteRef** HEAD

**RemoteSha** 61e8d60363c094882bbe8bb8500a21f697e4242b

## Contents

ds_alert	3
ds_avatar	4
ds_avatar_stack	4
ds_badge	5
ds_badge_position	6
ds_breadcrumbs	7
ds_button	7
ds_card	8
ds_card_block	9
ds_checkbox	10
ds_chip	10
ds_combobox	11
ds_dependencies	12
ds_details	12
ds_dialog	13
ds_divider	13
ds_dropdown	14
ds_error_summary	15
ds_field	15
ds_fieldset	16
ds_heading	17
ds_input	17
ds_label	18
ds_link	19
ds_list	19
ds_list_item	20
ds_pagination	20
ds_paragraph	21
ds_popover	22
ds_radio	22
ds_search	23
ds_select	24
ds_skeleton	25
ds_skip_link	25
ds_spinner	26
ds_suggestion	26
ds_tab	27
ds_table	28
ds_tablist	29
ds_tabpanel	29
ds_tabs	30
ds_tag	31
ds_tbody	32
ds_td	32
ds_textarea	33
ds_th	34

ds_thead . . . . .	34
ds_toggle_group . . . . .	35
ds_tooltip . . . . .	35
ds_tr . . . . .	36
ds_validation_message . . . . .	37
update_ds_checkbox . . . . .	37
update_ds_input . . . . .	38
update_ds_pagination . . . . .	39
update_ds_select . . . . .	39
update_ds_suggestion . . . . .	40
update_ds_tabs . . . . .	41
use_designsystemmet . . . . .	42

<b>Index</b>	<b>43</b>
--------------	-----------

---

ds_alert	<i>Alert Component</i>
----------	------------------------

---

## Description

Create an alert message using Designsystemmet 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	<i>Avatar Component</i>
-----------	-------------------------

---

**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

**Value**

A Shiny tag object

**Examples**

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

---

ds_avatar_stack	<i>Avatar Stack Component</i>
-----------------	-------------------------------

---

**Description**

Create a stack of avatars using Designsystemet styles.

**Usage**

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

**Arguments**

...	Avatar elements
class	Additional CSS classes

**Value**

A Shiny tag object

**Examples**

```

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

```

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 <span>.

**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**

```
# Count badge on a button (typical use)
htmltools::tags$span(
  class = "ds-badge--position",
  ds_button("Inbox"),
  ds_badge(count = 4, color = "danger")
)

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

---

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	<i>Breadcrumbs Component</i>
----------------	------------------------------

---

**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	<i>Button Component</i>
-----------	-------------------------

---

**Description**

Create a styled button using Designsystemet styles.

**Usage**

```
ds_button(  
  label,  
  inputId = NULL,  
  variant = c("primary", "secondary", "tertiary"),  
  size = NULL,  
  icon = FALSE,  
  loading = FALSE,
```

```

    disabled = FALSE,
    fullwidth = FALSE,
    type = "button",
    ...
  )

```

### Arguments

label	The button label
inputId	Optional input ID for Shiny reactivity (creates action button)
variant	Button variant ("primary", "secondary", "tertiary")
size	Size variant ("sm", "md", "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", "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\_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	<i>Checkbox Component</i>
-------------	---------------------------

---

**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	<i>Chip Component</i>
---------	-----------------------

---

**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\_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	<i>Designsystemet Dependencies</i>
-----------------	------------------------------------

---

**Description**

Returns the HTML dependencies required for Designsystemet components.

**Usage**

```
ds_dependencies()
```

**Value**

A list of htmltools htmlDependency objects

**Examples**

```
ds_dependencies()
```

---

ds_details	<i>Details/Accordion Component</i>
------------	------------------------------------

---

**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	<i>Dialog Component</i>
-----------	-------------------------

---

**Description**

Create a modal dialog using Designsystemet styles.

**Usage**

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

**Arguments**

...	Dialog content
id	Dialog ID for opening/closing
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	<i>Divider Component</i>
------------	--------------------------

---

**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, ..., class = NULL)
```

**Arguments**

trigger	The trigger element (usually a button)
...	Dropdown content
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	<i>Heading Component</i>
------------	--------------------------

---

**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	<i>Text Input Component</i>
----------	-----------------------------

---

**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	<i>Link Component</i>
---------	-----------------------

---

**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	<i>List Component</i>
---------	-----------------------

---

**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	<i>List Item</i>
--------------	------------------

---

**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	<i>Pagination Component</i>
---------------	-----------------------------

---

**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 <code>sprintf</code> 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 styled radio button using Designsystemet styles.

**Usage**

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

**Arguments**

inputId	The input slot for Shiny reactivity
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\_search

*Search Component*


---

**Description**

Create a search input using Designsystemet styles.

**Usage**

```
ds_search(
  inputId,
  value = "",
  placeholder = "Search...",
  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	<i>Skeleton Component</i>
-------------	---------------------------

---

**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	<i>Skip Link Component</i>
--------------	----------------------------

---

**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	<i>Tab List Container</i>
------------	---------------------------

---

**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	<i>Tab Panel</i>
-------------	------------------

---

**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	<i>Tag Component</i>
--------	----------------------

---

**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	<i>Textarea Component</i>
-------------	---------------------------

---

**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_toggle_group	<i>Toggle Group Component</i>
-----------------	-------------------------------

---

**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	<i>Tooltip</i>
------------	----------------

---

**Description**

Add a tooltip to an element using the data-tooltip attribute.

**Usage**

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

**Arguments**

element	The element to attach the tooltip to
text	The tooltip text
placement	Tooltip placement ("top", "bottom", "left", "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")
```

---

**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	<i>Update Text Input</i>
-----------------	--------------------------

---

**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\_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	<i>Update Tabs Input</i>
----------------	--------------------------

---

**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\\_alert](#), 3  
[ds\\_avatar](#), 4  
[ds\\_avatar\\_stack](#), 4  
[ds\\_badge](#), 5, 31  
[ds\\_badge\\_position](#), 6  
[ds\\_breadcrumbs](#), 7  
[ds\\_button](#), 7  
[ds\\_card](#), 8  
[ds\\_card\\_block](#), 9  
[ds\\_checkbox](#), 10  
[ds\\_chip](#), 10  
[ds\\_combobox](#), 11  
[ds\\_dependencies](#), 12  
[ds\\_details](#), 12  
[ds\\_dialog](#), 13  
[ds\\_divider](#), 13  
[ds\\_dropdown](#), 14  
[ds\\_error\\_summary](#), 15  
[ds\\_field](#), 15  
[ds\\_fieldset](#), 16  
[ds\\_heading](#), 17  
[ds\\_input](#), 17  
[ds\\_label](#), 18  
[ds\\_link](#), 19  
[ds\\_list](#), 19  
[ds\\_list\\_item](#), 20  
[ds\\_pagination](#), 20  
[ds\\_paragraph](#), 21  
[ds\\_popover](#), 22  
[ds\\_radio](#), 22  
[ds\\_search](#), 23  
[ds\\_select](#), 24  
[ds\\_skeleton](#), 25  
[ds\\_skip\\_link](#), 25  
[ds\\_spinner](#), 26  
[ds\\_suggestion](#), 26  
[ds\\_tab](#), 27  
[ds\\_table](#), 28  
[ds\\_tablist](#), 29  
[ds\\_tabpanel](#), 29  
[ds\\_tabs](#), 30  
[ds\\_tag](#), 31  
[ds\\_tbody](#), 32  
[ds\\_td](#), 32  
[ds\\_textarea](#), 33  
[ds\\_th](#), 34  
[ds\\_thead](#), 34  
[ds\\_toggle\\_group](#), 35  
[ds\\_tooltip](#), 35  
[ds\\_tr](#), 36  
[ds\\_validation\\_message](#), 37  
[update\\_ds\\_checkbox](#), 37  
[update\\_ds\\_input](#), 38  
[update\\_ds\\_pagination](#), 39  
[update\\_ds\\_select](#), 39  
[update\\_ds\\_suggestion](#), 40  
[update\\_ds\\_tabs](#), 41  
[use\\_designsystemet](#), 42