

Function reference version 0.13.0

UI Layout

Functions for laying out the user interface for your application.

<code>absolutePanel (absolutePanel.html)</code> <code>(fixedPanel)</code>	Panel with absolute positioning
<code>bootstrapPage (bootstrapPage.html)</code> <code>(basicPage)</code>	Create a Bootstrap page
<code>column (column.html)</code>	Create a column within a UI definition
<code>conditionalPanel</code> <code>(conditionalPanel.html)</code>	Conditional Panel
<code>fillPage (fillPage.html)</code>	Create a page that fills the window
<code>fillRow (fillRow.html) (fillCol)</code>	Flex Box-based row/column layouts
<code>fixedPage (fixedPage.html) (fixedRow)</code>	Create a page with a fixed layout
<code>fluidPage (fluidPage.html) (fluidRow)</code>	Create a page with fluid layout
<code>headerPanel (headerPanel.html)</code>	Create a header panel
<code>helpText (helpText.html)</code>	Create a help text element
<code>icon (icon.html)</code>	Create an icon
<code>mainPanel (mainPanel.html)</code>	Create a main panel
<code>navbarPage (navbarPage.html)</code> <code>(navbarMenu)</code>	Create a page with a top level navigation bar
<code>navlistPanel (navlistPanel.html)</code>	Create a navigation list panel
<code>pageWithSidebar</code> <code>(pageWithSidebar.html)</code>	Create a page with a sidebar
<code>sidebarLayout (sidebarLayout.html)</code>	Layout a sidebar and main area
<code>sidebarPanel (sidebarPanel.html)</code>	Create a sidebar panel
<code>tabPanel (tabPanel.html)</code>	Create a tab panel
<code>tabsetPanel (tabsetPanel.html)</code>	Create a tabset panel
<code>titlePanel (titlePanel.html)</code>	Create a panel containing an application title.
<code>inputPanel (inputPanel.html)</code>	Input panel
<code>flowLayout (flowLayout.html)</code>	Flow layout
<code>splitLayout (splitLayout.html)</code>	Split layout
<code>verticalLayout (verticalLayout.html)</code>	Lay out UI elements vertically
<code>wellPanel (wellPanel.html)</code>	Create a well panel

UI Inputs

Functions for creating user interface elements that prompt the user for input values or interaction.

<code>actionButton (actionButton.html)</code> <code>(actionLink)</code>	Action button/link
<code>checkboxGroupInput</code> <code>(checkboxGroupInput.html)</code>	Checkbox Group Input Control
<code>checkboxInput (checkboxInput.html)</code>	Checkbox Input Control
<code>dateInput (dateInput.html)</code>	Create date input
<code>dateRangeInput (dateRangeInput.html)</code>	Create date range input
<code>fileInput (fileInput.html)</code>	File Upload Control
<code>numericInput (numericInput.html)</code>	Create a numeric input control
<code>radioButtons (radioButtons.html)</code>	Create radio buttons
<code>selectInput (selectInput.html)</code> <code>(selectizeInput)</code>	Create a select list input control
<code>sliderInput (sliderInput.html)</code> <code>(animationOptions)</code>	Slider Input Widget
<code>submitButton (submitButton.html)</code>	Create a submit button
<code>textInput (textInput.html)</code>	Create a text input control
<code>passwordInput (passwordInput.html)</code>	Create a password input control
<code>updateCheckboxGroupInput</code> <code>(updateCheckboxGroupInput.html)</code>	Change the value of a checkbox group input on the client
<code>updateCheckboxInput</code> <code>(updateCheckboxInput.html)</code>	Change the value of a checkbox input on the client
<code>updateDateInput</code> <code>(updateDateInput.html)</code>	Change the value of a date input on the client
<code>updateDateRangeInput</code> <code>(updateDateRangeInput.html)</code>	Change the start and end values of a date range input on the client
<code>updateNumericInput</code> <code>(updateNumericInput.html)</code>	Change the value of a number input on the client
<code>updateRadioButtons</code> <code>(updateRadioButtons.html)</code>	Change the value of a radio input on the client
<code>updateSelectInput</code> <code>(updateSelectInput.html)</code> <code>(updateSelectizeInput)</code>	Change the value of a select input on the client
<code>updateSliderInput</code> <code>(updateSliderInput.html)</code>	Change the value of a slider input on the client

(updateTextInput.html)

updateTabsetPanel
(updateTabsetPanel.html)
(updateNavbarPage,
updateNavlistPanel)

Change the selected tab on the client

updateTextInput
(updateTextInput.html)

Change the value of a text input on the client

UI Outputs

Functions for creating user interface elements that, in conjunction with rendering functions, display different kinds of output from your application.

htmlOutput (htmlOutput.html)
(uiOutput)

Create an HTML output element

plotOutput (plotOutput.html)
(imageOutput)

Create an plot or image output element

outputOptions (outputOptions.html)

Set options for an output object.

tableOutput (tableOutput.html)
(dataTableOutput)

Create a table output element

textOutput (textOutput.html)

Create a text output element

verbatimTextOutput
(verbatimTextOutput.html)

Create a verbatim text output element

downloadButton (downloadButton.html)
(downloadLink)

Create a download button or link

Progress (Progress.html)

Reporting progress (object-oriented API)

withProgress (withProgress.html)
(incProgress, setProgress)

Reporting progress (functional API)

Interface builder functions

A sub-library for writing HTML using R functions. These functions form the foundation on which the higher level user interface functions are built, and can also be used in your Shiny UI to provide custom HTML, CSS, and JavaScript.

builder (builder.html) (a, br, code,
div, em, h1, h2, h3, h4, h5, h6, hr,
img, p, pre, span, strong, tags)

HTML Builder Functions

HTML (HTML.html)

Mark Characters as HTML

include (include.html) (includeCSS,
includeHTML, includeMarkdown,
includeScript, includeText)

Include Content From a File

singleton (singleton.html)
(includeOnce)

Include content only once

<code>(is.singleton)</code>	
<code>tag (tag.html) (tagAppendAttributes, tagAppendChild, tagAppendChildren, tagList, tagSetChildren)</code>	HTML Tag Object
<code>validateCssUnit (validateCssUnit.html)</code>	Validate proper CSS formatting of a unit
<code>withTags (withTags.html)</code>	Evaluate an expression using
<code>htmlTemplate (htmlTemplate.html)</code>	Process an HTML template
<code>bootstrapLib (bootstrapLib.html)</code>	Bootstrap libraries
<code>suppressDependencies (suppressDependencies.html)</code>	Suppress web dependencies

Rendering functions

Functions that you use in your application's server side code, assigning them to outputs that appear in your user interface.

<code>renderPlot (renderPlot.html)</code>	Plot Output
<code>renderText (renderText.html)</code>	Text Output
<code>renderPrint (renderPrint.html)</code>	Printable Output
<code>renderDataTable (renderDataTable.html)</code>	Table output with the JavaScript library DataTables
<code>renderImage (renderImage.html)</code>	Image file output
<code>renderTable (renderTable.html)</code>	Table Output
<code>renderUI (renderUI.html)</code>	UI Output
<code>downloadHandler (downloadHandler.html)</code>	File Downloads
<code>reactivePlot (reactivePlot.html)</code>	Plot output (deprecated)
<code>reactivePrint (reactivePrint.html)</code>	Print output (deprecated)
<code>reactiveTable (reactiveTable.html)</code>	Table output (deprecated)
<code>reactiveText (reactiveText.html)</code>	Text output (deprecated)
<code>reactiveUI (reactiveUI.html)</code>	UI output (deprecated)

Reactive constructs

A sub-library that provides reactive programming facilities for R.

<code>invalidateLater (invalidateLater.html)</code>	Scheduled Invalidation
---	------------------------

<code>(invalidateLater.html)</code>	
<code>is.reactivevalues</code> <code>(is.reactivevalues.html)</code>	Checks whether an object is a reactivevalues object
<code>isolate</code> (<code>isolate.html</code>)	Create a non-reactive scope for an expression
<code>makeReactiveBinding</code> <code>(makeReactiveBinding.html)</code>	Make a reactive variable
<code>observe</code> (<code>observe.html</code>)	Create a reactive observer
<code>observeEvent</code> (<code>observeEvent.html</code>) <code>(eventReactive)</code>	Event handler
<code>reactive</code> (<code>reactive.html</code>) <code>(is.reactive)</code>	Create a reactive expression
<code>reactiveFileReader</code> <code>(reactiveFileReader.html)</code>	Reactive file reader
<code>reactivePoll</code> (<code>reactivePoll.html</code>)	Reactive polling
<code>reactiveTimer</code> (<code>reactiveTimer.html</code>)	Timer
<code>reactiveValues</code> (<code>reactiveValues.html</code>)	Create an object for storing reactive values
<code>reactiveValuesToList</code> <code>(reactiveValuesToList.html)</code>	Convert a reactivevalues object to a list
<code>domains</code> (<code>domains.html</code>) <code>(getDefaultReactiveDomain,</code> <code>onReactiveDomainEnded,</code> <code>withReactiveDomain)</code>	Reactive domains
<code>showReactLog</code> (<code>showReactLog.html</code>)	Reactive Log Visualizer

Boilerplate

Functions that are required boilerplate in `ui.R` and `server.R`.

<code>shinyUI</code> (<code>shinyUI.html</code>)	Create a Shiny UI handler
<code>shinyServer</code> (<code>shinyServer.html</code>)	Define Server Functionality

Running

Functions that are used to run or stop Shiny applications.

<code>runApp</code> (<code>runApp.html</code>)	Run Shiny Application
<code>runGadget</code> (<code>runGadget.html</code>)	Run a gadget
<code>runExample</code> (<code>runExample.html</code>)	Run Shiny Example Applications
<code>runGadget</code> (<code>runGadget.html</code>)	Run a gadget
<code>runUrl</code> (<code>runUrl.html</code>) (<code>runGist,</code> <code>runGitHub)</code>	Run a Shiny application from a URL

<code>stopApp</code> (<code>stopApp.html</code>)	Stop the currently running Shiny app
<code>viewer</code> (<code>viewer.html</code>) (<code>browserViewer</code> , <code>dialogViewer</code> , <code>paneViewer</code>)	Viewer options

Extending Shiny

Functions that are intended to be called by third-party packages that extend Shiny.

<code>createWebDependency</code> (<code>createWebDependency.html</code>)	Create a web dependency
<code>addResourcePath</code> (<code>addResourcePath.html</code>)	Resource Publishing
<code>registerInputHandler</code> (<code>registerInputHandler.html</code>)	Register an Input Handler
<code>removeInputHandler</code> (<code>removeInputHandler.html</code>)	Deregister an Input Handler
<code>markRenderFunction</code> (<code>markRenderFunction.html</code>)	Mark a function as a render function

Utility functions

Miscellaneous utilities that may be useful to advanced users or when extending Shiny.

<code>req</code> (<code>req.html</code>)	Check for required values
<code>validate</code> (<code>validate.html</code>) (<code>need</code>)	Validate input values and other conditions
<code>session</code> (<code>session.html</code>)	Session object
<code>exprToFunction</code> (<code>exprToFunction.html</code>)	Convert an expression to a function
<code>installExprFunction</code> (<code>installExprFunction.html</code>)	Install an expression as a function
<code>parseQueryString</code> (<code>parseQueryString.html</code>)	Parse a GET query string from a URL
<code>plotPNG</code> (<code>plotPNG.html</code>)	Run a plotting function and save the output as a PNG
<code>repeatable</code> (<code>repeatable.html</code>)	Make a random number generator repeatable
<code>shinyDeprecated</code> (<code>shinyDeprecated.html</code>)	Print message for deprecated functions in Shiny
<code>serverInfo</code> (<code>serverInfo.html</code>)	Collect information about the Shiny Server environment
<code>shiny-options</code> (<code>shiny-options.html</code>)	Global options for Shiny

Plot interaction

Functions related to interactive plots

<code>brushedPoints</code> (<code>brushedPoints.html</code>)	Find rows of data that are selected by a brush
<code>brushOpts</code> (<code>brushOpts.html</code>)	Create an object representing brushing options
<code>clickOpts</code> (<code>clickOpts.html</code>)	Create an object representing click options
<code>dblclickOpts</code> (<code>dblclickOpts.html</code>)	Create an object representing double-click options
<code>hoverOpts</code> (<code>hoverOpts.html</code>)	Create an object representing hover options
<code>nearPoints</code> (<code>nearPoints.html</code>)	Find rows of data that are near a click/hover/double-click

Modules

Functions for modularizing Shiny apps

<code>NS</code> (<code>NS.html</code>) (<code>ns.sep</code>)	Namespaced IDs for inputs/outputs
<code>callModule</code> (<code>callModule.html</code>)	Invoke a Shiny module

Embedding

Functions that are intended for third-party packages that embed Shiny applications.

<code>shinyApp</code> (<code>shinyApp.html</code>) (<code>as.shiny.appobj</code> , <code>as.shiny.appobj.character</code> , <code>as.shiny.appobj.list</code> , <code>as.shiny.appobj.shiny.appobj</code> , <code>as.tags.shiny.appobj</code> , <code>is.shiny.appobj</code> , <code>print.shiny.appobj</code> , <code>shinyAppDir</code> , <code>shinyAppFile</code>)	Create a Shiny app object
<code>maskReactiveContext</code> (<code>maskReactiveContext.html</code>)	Evaluate an expression without a reactive context