Create a select list input control

Description

Create a select list that can be used to choose a single or multiple items from a list of values.

Usage

```
selectInput(inputId, label, choices, selected = NULL, multiple = FALSE,
  selectize = TRUE, width = NULL, size = NULL)

selectizeInput(inputId, ..., options = NULL, width = NULL)
```

Arguments

`inputId`	The `input` slot that will be used to access the value.
`label`	Display label for the control, or `NULL` for no label.
`choices`	List of values to select from. If elements of the list are named, then that name rather than the value is displayed to the user. This can also be a named list whose elements are (either named or unnamed) lists or vectors. If this is the case, the outermost names will be used as the "optgroup" label for the elements in the respective sublist. This allows you to group and label similar choices. See the example section for a small demo of this feature.
`selected`	The initially selected value (or multiple values if `multiple = TRUE`). If not specified then defaults to the first value for single-select lists and no values for multiple select lists.
`multiple`	Is selection of multiple items allowed?
`selectize`	Whether to use **selectize.js** or not.
`width`	The width of the input, e.g. `'400px'`, or `'100%'`; see `validateCssUnit`.
`size`	Number of items to show in the selection box; a larger number will result in a taller box. Not compatible with `selectize=TRUE`. Normally, when `multiple=FALSE`, a select input will be a drop-down list, but when `size` is set, it will be a box instead.
`...`	Arguments passed to `selectInput()`.
`options`	A list of options. See the documentation of **selectize.js** for possible options (character option values inside `I()` will be treated as literal JavaScript code; see `renderDataTable()` for details).

Details

By default, `selectInput()` and `selectizeInput()` use the JavaScript library **selectize.js** (https://github.com/selectize/selectize.js) to instead of the basic select input element. To use the standard HTML select input element, use `selectInput()` with `selectize=FALSE`.

In selectize mode, if the first element in `choices` has a value of `""`, its name will be treated as a placeholder prompt. For example: `selectInput("letter", "Letter", c("Choose one" = "", LETTERS))`

Value

A select list control that can be added to a UI definition.

Note

The selectize input created from `selectizeInput()` allows deletion of the selected option even in a single select input, which will return an empty string as its value. This is the default behavior of **selectize.js**. However, the selectize input created from `selectInput(..., selectize = TRUE)` will ignore the empty string value when it is a single choice input and the empty string is not in the `choices` argument. This is to keep compatibility with `selectInput(..., selectize = FALSE)`.

See Also

`updateSelectInput`

Other input elements: `actionButton`, `checkboxGroupInput`, `checkboxInput`, `dateInput`, `dateRangeInput`, `fileInput`, `numericInput`, `passwordInput`, `radioButtons`, `sliderInput`, `submitButton`, `textAreaInput`, `textInput`

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {

# basic example
shinyApp(
  ui = fluidPage(
    selectInput("variable", "Variable:",
                c("Cylinders" = "cyl",
                  "Transmission" = "am",
                  "Gears" = "gear")),
    tableOutput("data")
  ),
  server = function(input, output) {
    output$data <- renderTable({
      mtcars[, c("mpg", input$variable), drop = FALSE]
    }, rownames = TRUE)
  }
)

# demoing optgroup support in the `choices` arg
shinyApp(
  ui = fluidPage(
    selectInput("state", "Choose a state:",
      list(`East Coast` = c("NY", "NJ", "CT"),
           `West Coast` = c("WA", "OR", "CA"),
           `Midwest` = c("MN", "WI", "IA"))
    ),
    textOutput("result")
  ),
  server = function(input, output) {
    output$result <- renderText({
      paste("You chose", input$state)
    })
  }
)
}
```