

This tutorial is deprecated. Learn more about Shiny at our new location, shiny.rstudio.com.

GETTING STARTED

Welcome
Hello Shiny
Shiny Text
Reactivity

BUILDING AN APP

UI & Server
Inputs & Outputs
Run & Debug

TOOLING UP

Sliders
Tabsets
DataTables

More Widgets

Uploading Files
Downloading Data
HTML UI
Dynamic UI

ADVANCED SHINY

Scoping
Client Data
Sending Images

UNDERSTANDING REACTIVITY

Reactivity Overview
Execution Scheduling
Isolation

DEPLOYING AND SHARING APPS

Deploying Over the Web
Sharing Apps to Run Locally

EXTENDING SHINY

Building Inputs
Building Outputs

More Widgets - Mozilla Firefox

More Widgets

Choose a dataset:

rock

Number of observations to view:

10

Note: while the data view will show only the specified number of observations, the summary will still be based on the full dataset.

Update View

Summary

	area		peri		shape		perm
Min.	: 1016	Min.	: 308.6	Min.	:0.09033	Min.	: 6.30
1st Qu.	: 5305	1st Qu.	:1414.9	1st Qu.	:0.16226	1st Qu.	: 76.45
Median	: 7487	Median	:2536.2	Median	:0.19886	Median	: 130.50
Mean	: 7188	Mean	:2682.2	Mean	:0.21811	Mean	: 415.45
3rd Qu.	: 8870	3rd Qu.	:3989.5	3rd Qu.	:0.26267	3rd Qu.	: 777.50
Max.	:12212	Max.	:4864.2	Max.	:0.46413	Max.	:1300.00

Observations

	area	peri	shape	perm
1	4990	2791.90	0.09	6.30
2	7002	3892.60	0.15	6.30
3	7558	3930.66	0.18	6.30
4	7352	3869.32	0.12	6.30
5	7943	3948.54	0.12	17.10
6	7979	4010.15	0.17	17.10
7	9333	4345.75	0.19	17.10
8	8209	4344.75	0.16	17.10
9	8393	3682.04	0.20	119.00
10	6425	3098.65	0.16	119.00

The More Widgets application demonstrates the help text and submit button widgets as well as the use of embedded HTML elements to customize formatting. To run the example type:

```
> library(shiny)
> runExample("07_widgets")
```

UI Enhancements

In this example we update the Shiny Text application with some additional controls and formatting, specifically:

- We added a `helpText` control to provide additional clarifying text alongside our input controls.
- We added a `submitButton` control to indicate that we don't want a live connection between inputs and outputs, but rather to wait until the user clicks that button to update the output. This is especially useful if computing output is computationally expensive.
- We added `h4` elements (heading level 4) into the output pane. Shiny offers a variety of functions for including HTML elements directly in pages including headings, paragraphs, links, and more.

Here is the updated source code for the user-interface:

```
ui.R

library(shiny)

# Define UI for dataset viewer application
shinyUI(pageWithSidebar(

  # Application title.
  headerPanel("More Widgets"),

  # Sidebar with controls to select a dataset and specify the number
  # of observations to view. The helpText function is also used to
  # include clarifying text. Most notably, the inclusion of a
  # submitButton defers the rendering of output until the user
  # explicitly clicks the button (rather than doing it immediately
  # when inputs change). This is useful if the computations required
  # to render output are inordinately time-consuming.
  sidebarPanel(
    selectInput("dataset", "Choose a dataset:",
               choices = c("rock", "pressure", "cars")),

    numericInput("obs", "Number of observations to view:", 10),

    helpText("Note: while the data view will show only the specified",
             "number of observations, the summary will still be based",
             "on the full dataset."),

    submitButton("Update View")
  ),

  # Show a summary of the dataset and an HTML table with the requested
  # number of observations. Note the use of the h4 function to provide
  # an additional header above each output section.
  mainPanel(
    h4("Summary"),
    verbatimTextOutput("summary"),

    h4("Observations"),
    tableOutput("view")
  )
))
```

Server Script

All of the changes from the original Shiny Text application were to the user-interface, the server script remains the same:

```
server.R

library(shiny)
library(datasets)

# Define server logic required to summarize and view the selected dataset
shinyServer(function(input, output) {

  # Return the requested dataset
  datasetInput <- reactive({
    switch(input$dataset,
           "rock" = rock,
           "pressure" = pressure,
           "cars" = cars)
  })

  # Generate a summary of the dataset
  output$summary <- renderPrint({
    dataset <- datasetInput()
    summary(dataset)
  })

  # Show the first "n" observations
  output$view <- renderTable({
    head(datasetInput(), n = input$obs)
  })
})
```

← Previous

Next →

© 2012-2013 RStudio, Inc. All rights reserved.
Code samples in this tutorial are released under the [Creative Commons Zero 1.0](#) license (CC0).