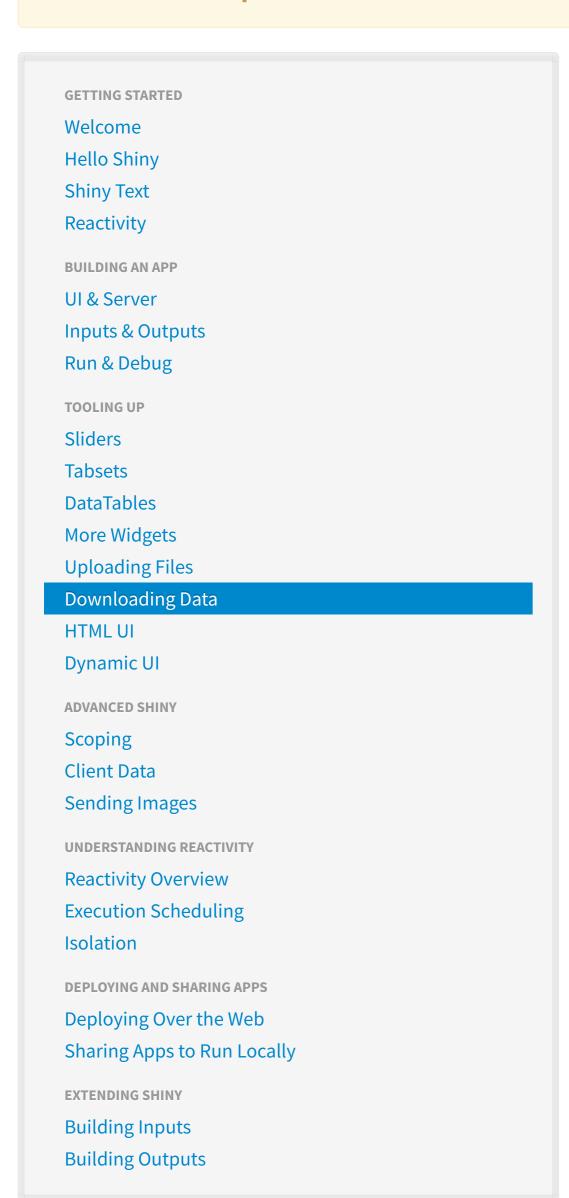
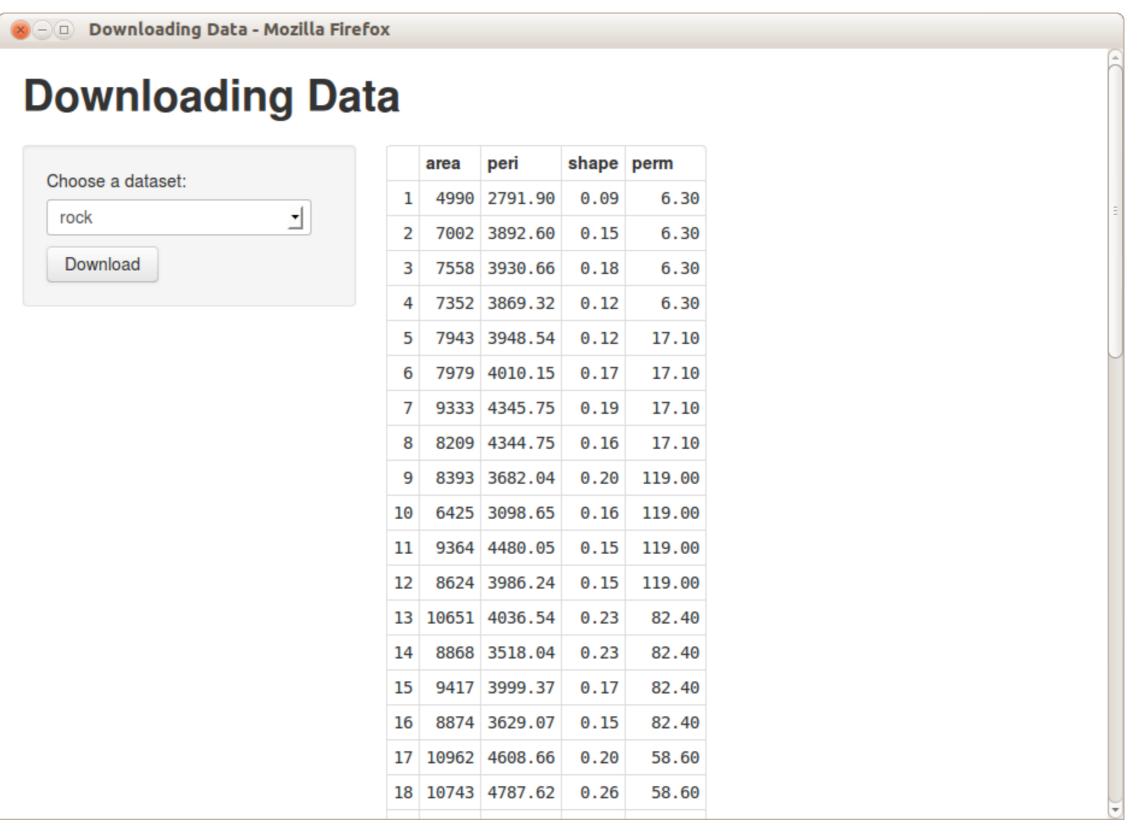
This tutorial is deprecated. Learn more about Shiny at our new location, shiny.rstudio.com.





The examples so far have demonstrated outputs that appear directly in the page, such as plots, tables, and text boxes. Shiny also has the ability to offer file downloads that are calculated on the fly, which makes it easy to build data exporting features.

To run the example below, type:

```
> library(shiny)
> runExample("10_download")
```

You define a download using the downloadHandler function on the server side, and either downloadButton or downloadLink in the UI:

```
shinyUI(pageWithSidebar(
  headerPanel('Download Example'),
  sidebarPanel(
```

server.R

ui.R

As you can see, downloadHandler takes a filename argument, which tells the web browser what filename to default to when saving. This argument can either be a simple string, or it can be a function that returns a string (as is the case here).

The content argument must be a function that takes a single argument, the file name of a non-existent temp file. The content function is responsible for writing the contents of the file download into that temp file.

Both the filename and content arguments can use reactive values and expressions (although in the case of filename, be sure your argument is an actual function; filename = paste(input\$dataset, '.csv') is not going to work the way you want it to, since it is evaluated only once, when the download handler is being defined).

Generally, those are the only two arguments you'll need. There is an optional contentType argument; if it is NA or NULL, Shiny will attempt to guess the appropriate value based on the filename. Provide your own content type string (e.g. "text/plain") if you want to override this behavior.

