Shiny 速查表

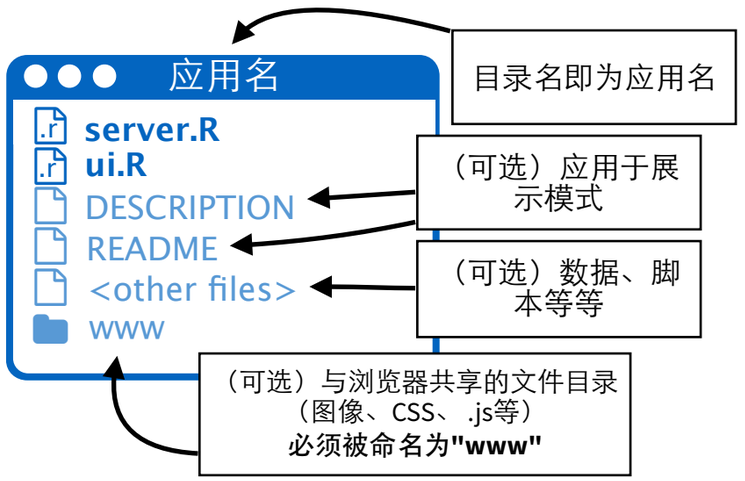
详情访问shiny.rstudio.com

Shiny 0.10.0 Updated: 6/14

RStudio

由supstat ANALYTICS 翻译

1. 架构 每个应用程序都相当于一个目录。这个目录包含一个server.R文件，一般还包含一个ui.R文件（以及可选的额外文件）



server.R

```
  # load libraries, scripts, data

A shinyServer(function(input, output) {B

  # make user specific variables

    output$text <- renderText({
      input$title
    })

C   output$plot <- renderPlot({D
      x <- mtcars[ , input$x]F
E     y <- mtcars[ , input$y]
      plot(x, y, pch = 16)
    })

})
```

render*函数

函数	预期对象	创建对象
renderDataTable	任何表格对象	DataTables.js table
renderImage	图像特征列表	HTML图像
renderPlot	图表	图表
renderPrint	任何打印输出	文本
renderTable	任何表格对象	无格式表格
renderText	字符串	文本
renderUI	Shiny标签对象或HTML	UI元素(HTML)

输入值都是有响应的。
它们必须受以下对象之一所包围：

render* - 创建Shiny UI组件
reactive - 创建响应表达式
observe - 创建响应观测器
isolate - 创建响应对象的非响应副本

2. server.R 应用程序的R组件的构建命令组。如何编写server.R：

A 为server.R提供最简短的必要代码，**shinyServer(function(input, output) {})**

B 在**function(input, output)**之后的大括号之间定义应用的R组件

C 以**output$<component name>**的方式在UI之中保存R组件

D 通过render*函数来创建每一个输出组件

E 把服务器所需的用于建立组件的R代码赋予到每一个render*函数中。服务器将会对每一个出现在R代码中的响应值做标记且在这些值发生改变的同时重建组件。

F 通过**input$<widget name>**引用部件值

3. 执行 把代码放置在运行次数最少的地方

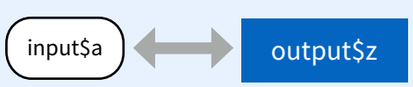
运行一次 - 当你首次启动应用程序时，在*shinyServer*函数之外的代码将只会运行一次。对于服务器只需要其一个副本的工具，利用该代码来建立此这些工具。

为每个用户运行一次 - 每当用户访问应用程序（或刷新浏览器页面）时，在*shinyServer*函数内的代码都将运行一次。对于服务器针对每一个用户均需要一个独特副本的工具，利用该代码来建立来建立这些工具。

多次运行 - 在render*、reactive或observe函数中的代码将会运行多次。只有在部件发生改变后服务器重建UI组件所需的代码会放在这个位置。

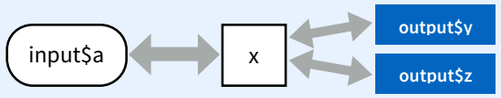
4. 响应 当输入改变时，服务器将重建依赖于此输入的每一个输出（即便是非直接的依赖关系）。你可通过改变依赖链的方式来控制该行为。

render* - 每当在其render*中的输入改变时，输出将会自动更新。




```
output$z <- renderText({
  input$a
})
```

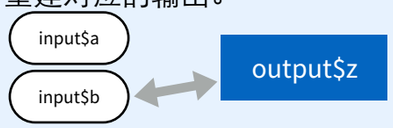
Reactive表达式 - 利用reactive表达式来创建将会用于多个输出的对象。




```
x <- reactive({
  input$a
})

output$y <- renderText({
  x()
})
output$z <- renderText({
  x()
})
```

isolate - 利用isolate来使用一个输入而无需依赖于该输入。当被isolate函数使用的输入改变时，Shiny将不会重建对应的输出。




```
output$z <- renderText({
  paste(
   isolate(input$a),
   input$b
  )
)
```

observe - 利用observe来创建代码使之在输入改变时运行，却不会生成输出对象。


```
observe({
  input$a
  # code to run
})
```


本介绍由SupStat Inc.翻译。
英文网址：supstat.com 中文网址：supstat.com.cn

ui.R

```
Ⓐ shinyUI(fluidPage(

    titlePanel("mtcars data"),
  Ⓑ sidebarLayout(
      sidebarPanel(

    Ⓒ   textInput("title", "Plot title:",
          value = "x v y"),

        selectInput("x", "Choose an x var:",
          choices = names(mtcars),
          selected = "disp"),

        selectInput("y", "Choose a y var:",
          choices = names(mtcars),
          selected = "mpg")
      ),

      mainPanel(
        h3(textOutput("text")),
        plotOutput("plot")
      )
    )
  ))
```

Ⓒ 在每一个面板和列中，放置...

5. ui.R 应用程序用户界面（UI）的描述，展示应用程序的网页。如何编写ui.R:

Ⓐ 在ui.R, shinyUI(fluidPage())中包含最简短的必要代码
* 注意: 若希望使用导航条来连接多页面应用程序，用navbarPage替代fluidPage

Ⓑ 为UI建立一个板面。在与sidebarPanel和mainPanel一同使用时，sidebarLayout将会提供一个默认板面。splitLayout、flowLayout和inputLayout把页面划分为多个等间隔排列的区域。fluidRow和column一同创建了一个基于网格的板面，可用于规划页面或面板。

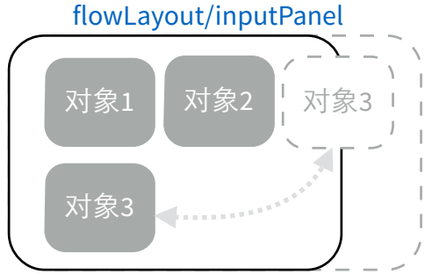
sidebarLayout


```
shinyUI(fluidPage(
  sidebarLayout(
    sidebarPanel(…),
    mainPanel(…)
  )
))
```

splitLayout

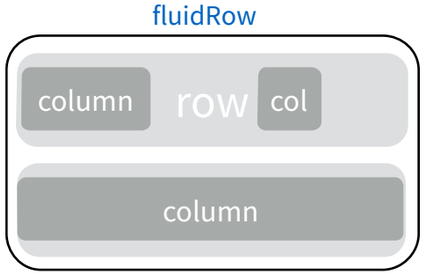

```
shinyUI(fluidPage(
  splitLayout(
    numericInput(…),
    selectInput(…)
  )
))
```

flowLayout/inputPanel




```
shinyUI(fluidPage(
  flowLayout(
    numericInput(…),
    selectInput(…),
    sliderInput(…)
  )
))
```

fluidRow




```
shinyUI(fluidPage(
  fluidRow(
    column(width = 4, …),
    column(width = 2,
      offset = 3, …),
  ),
  fluidRow(
    column(width = 12, …)
  )
))
```

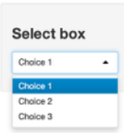
R组件 - 它们是在server.R种定义的输出对象。如何放置一个组件：

1. 选择*Output函数，使之建立你想安置在UI中的对象类型。
2. 把与在server.R中所指定的对象名相对应的字符串作为参数传递到*Output函数中，如

output$plot <- renderPlot({ ... }) ⟷ plotOutput("plot")

*Output函数

dataTableOutput	tableOutput
htmlOutput	textOutput
imageOutput	uiOutput
plotOutput	verbatimTextOutput



部件 - 每个部件函数的第一个参数都是部件的<名称>。你可以通过input$<名称>来访问在server.R中的部件当前值。

部件	函数	常用参数
动作按钮	actionButton	inputId, label
复选框	checkboxInput	inputId, label, value
复选框组	checkboxGroupInput	inputId, label, choices, selected
选择日期	dateInput	inputId, label, value, min, max, format
选择日期范围	dateRangeInput	inputId, label, start, end, min, max, format
文件上传	fileInput	inputId, label, multiple
数字字段	numericInput	inputId, label, value, min, max, step
单选按钮	radioButtons	inputId, label, choices, selected
选项框	selectInput	inputId, label, choices, selected, multiple
滑动条	sliderInput	inputId, label, min, max, value, step
提交按钮	submitButton	text
文本字段	textInput	inputId, label, value

HTML元素 - 通过Shiny函数添加HTML元素，它们相当于一般的HTML标签。

a	tags$col	tags$form	tags$input	tags$output	tags$sub
tags$abbr	tags$colgroup	h1	tags$ins	p	tags$summary
tags$address	tags$command	h2	tags$kbd	tags$param	tags$sup
tags$area	tags$data	h3	tags$keygen	pre	tags$table
tags$article	tags$datalist	h4	tags$label	tags$progress	tags$tbody
tags$aside	tags$dd	h5	tags$legend	tags$q	tags$td
tags$audio	tags$del	h6	tags$li	tags$ruby	tags$textarea
tags$b	tags$details	tags$head	tags$link	tags$rp	tags$tfoot
tags$base	tags$dfn	tags$header	tags$mark	tags$rt	tags$th
tags$bdi	div	tags$hgroup	tags$map	tags$s	tags$thead
tags$bdo	tags$dl	hr	tags$menu	tags$samp	tags$time
tags$blockquote	tags$dt	HTML	tags$meta	tags$script	tags$title
tags$body	em	tags$i	tags$meter	tags$section	tags$tr
br	tags$embed	tags$iframe	tags$nav	tags$select	tags$track
tags$button	tags$eventsource	img	tags$noscript	tags$small	tags$u
tags$canvas	tags$fieldset	includeCSS	tags$object	tags$source	tags$ul
tags$caption	tags$figcaption	includeMarkdo	tags$ol	span	tags$var
tags$cite	tags$figure	wn	tags$optgroup	strong	tags$video
code	tags$footer	includeScript	tags$option	tags$style	tags$wbr

6. 运行应用程序

runApp - 从本地文件中运行
runGitHub - 从www.GitHub.com管理的文件中运行
runGist - 从保存为gist的文件中运行 (gist.github.com)
runURL - 从保存在任意URL中的文件中运行

本介绍由SupStat Inc.翻译。 英文网址：supstat.com 中文网址：supstat.com.cn

7. 分享应用程序 将应用程序作为实时网页来启动，让用户在线访问

ShinyApps.io

在RStudio服务器种管理应用程序。免费和付费方案：
www.shinyapps.io

Shiny Server

构建个人的linux服务器来管理应用程序。免费且开源。
shiny.rstudio.com/deploy

Shiny Server Pro

构建商业服务器，享受身份验证、资源管理及更多服务。
shiny.rstudio.com/deploy