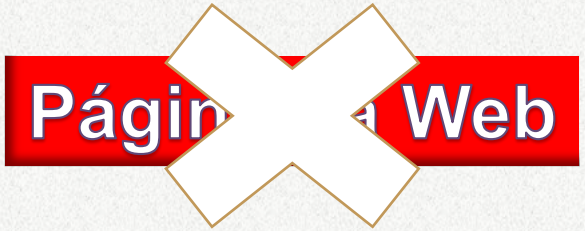
Descobrindo o Shiny: Como tornar sua visualização de dados mais atraente?

Definindo o Shiny...

Sistema para desenvolvimento de aplicações web

Server Side

User Side



Definindo o Shiny...

Onde fica o código de uma aplicação shiny?

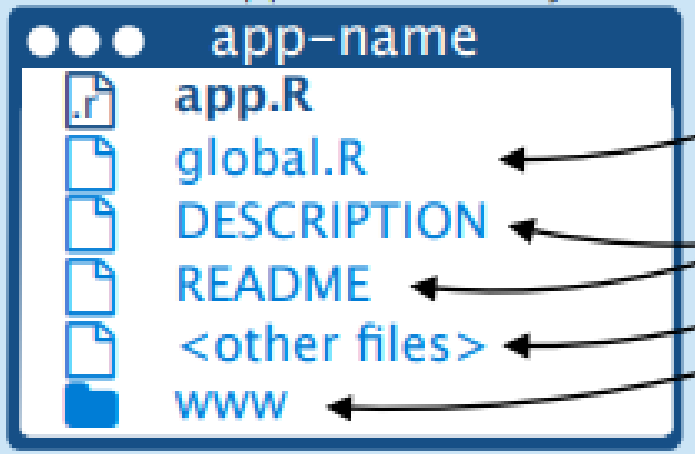
Recepção de informação + processamento de dados + visualização

Programação envolvida: HTML, JavaScript, CSS

Eu preciso saber tudo isso?

Estrutura de um arquivo Shiny...

Save each app as a directory that contains an **app.R** file (or a **server.R** file and a **ui.R** file) plus optional extra files.



The directory name is the name of the app

(optional) defines objects available to both ui.R and server.R

(optional) used in showcase mode

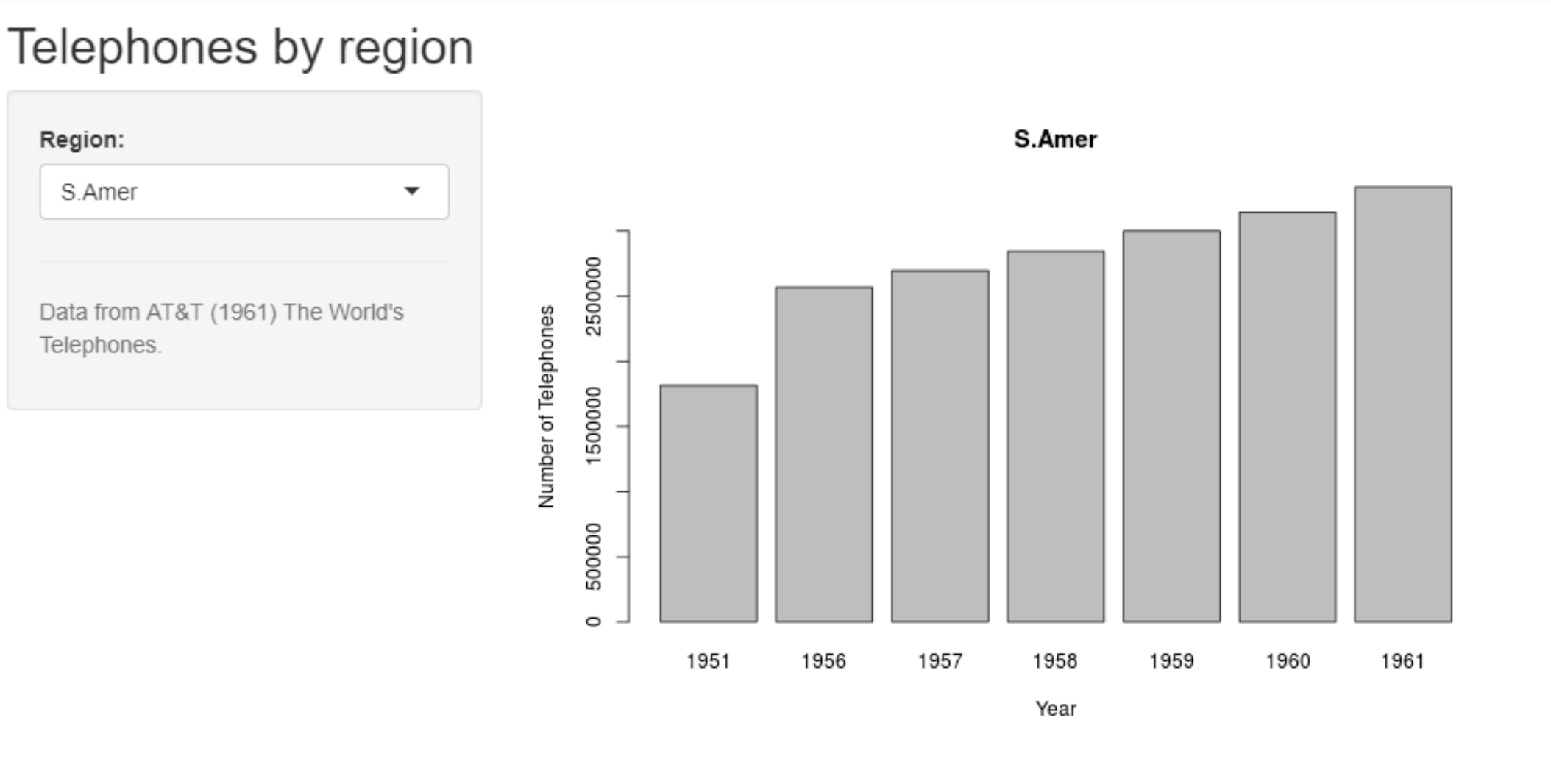
(optional) data, scripts, etc.

(optional) directory of files to share with web browsers (images, CSS, .js, etc.) Must be named "**www**"

Launch apps with `runApp(<path to directory>)`

Fonte: Cheat Sheet - RStudio

Conhecendo as estruturas Ui e Server...



Fonte: Shiny Gallery - RStudio

Conhecendo as estruturas Ui e Server...

server.R ui.R

```r
# Rely on the 'WorldPhones' dataset in the datasets
# package (which generally comes preloaded).
library(datasets)

# Define a server for the Shiny app
function(input, output) {

  # Fill in the spot we created for a plot
  output$phonePlot <- renderPlot({

    # Render a barplot
    barplot(WorldPhones[,input$region]*1000,
            main=input$region,
            ylab="Number of Telephones",
            xlab="Year")
  })
}
```

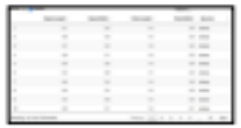
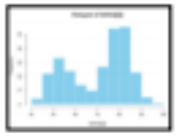
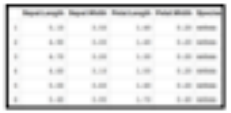
server.R ui.R

```r
# Rely on the 'WorldPhones' dataset in the datasets
# package (which generally comes preloaded).
library(datasets)

# Use a fluid Bootstrap layout
fluidPage(

  # Give the page a title
  titlePanel("Telephones by region"),

  # Generate a row with a sidebar
  sidebarLayout(

    # Define the sidebar with one input
    sidebarPanel(
      selectInput("region", "Region:",
                  choices=colnames(WorldPhones)),
      hr(),
      helpText("Data from AT&T (1961) The World's Telephones.")
    ),

    # Create a spot for the barplot
    mainPanel(
      plotOutput("phonePlot")
    )

  )
)
```

Fonte: Shiny Gallery - RStudio

Compreendendo Render e Outputs...

Outputs - render*() and *Output() functions work together to add R output to the UI

	DT::**renderDataTable**(expr, options, callback, escape, env, quoted)	works with	**dataTableOutput**(outputId, icon, …)
	renderImage(expr, env, quoted, deleteFile)		**imageOutput**(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
	**renderPlot**(expr, width, height, res, …, env, quoted, func)		**plotOutput**(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
	**renderPrint**(expr, env, quoted, func, width)		**verbatimTextOutput**(outputId)
	**renderTable**(expr,…, env, quoted, func)		**tableOutput**(outputId)
foo	**renderText**(expr, env, quoted, func)		**textOutput**(outputId, container, inline)
	**renderUI**(expr, env, quoted, func)		**uiOutput**(outputId, inline, container, …) & **htmlOutput**(outputId, inline, container, …)

Fonte: Cheat Sheet - RStudio

Itens de Performance para o Shiny

Button

Action

actionButton()

Single checkbox

Choice A

checkboxInput()

Checkbox group

Choice 1
Choice 2
Choice 3

checkboxGroupInput()

Date input

2014-01-01

dateInput()

Colour input

#52CC4E

colourpicker::colourInput()

Date range

2014-01-24 to 2014-01-24

dateRangeInput()

File input

Choose File No file chosen

fileInput()

Numeric input

1

numericInput()

Password Input

••••••••••

passwordInput()

Text area

Multiple lines
of text

textAreaInput()

Radio buttons

Choice 1
Choice 2
Choice 3

radioButtons()

Select box

Choice 1

selectInput()

Sliders

0 50 100
0 25 75 100

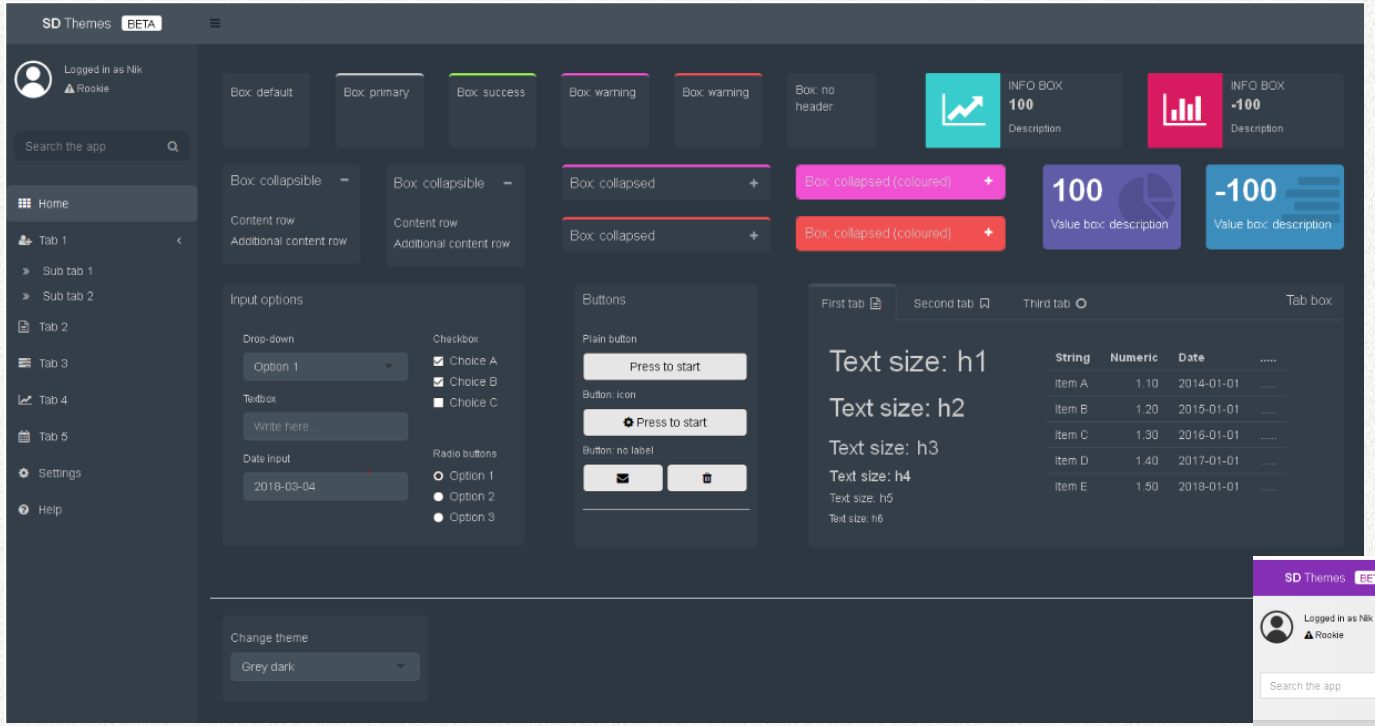
sliderInput()

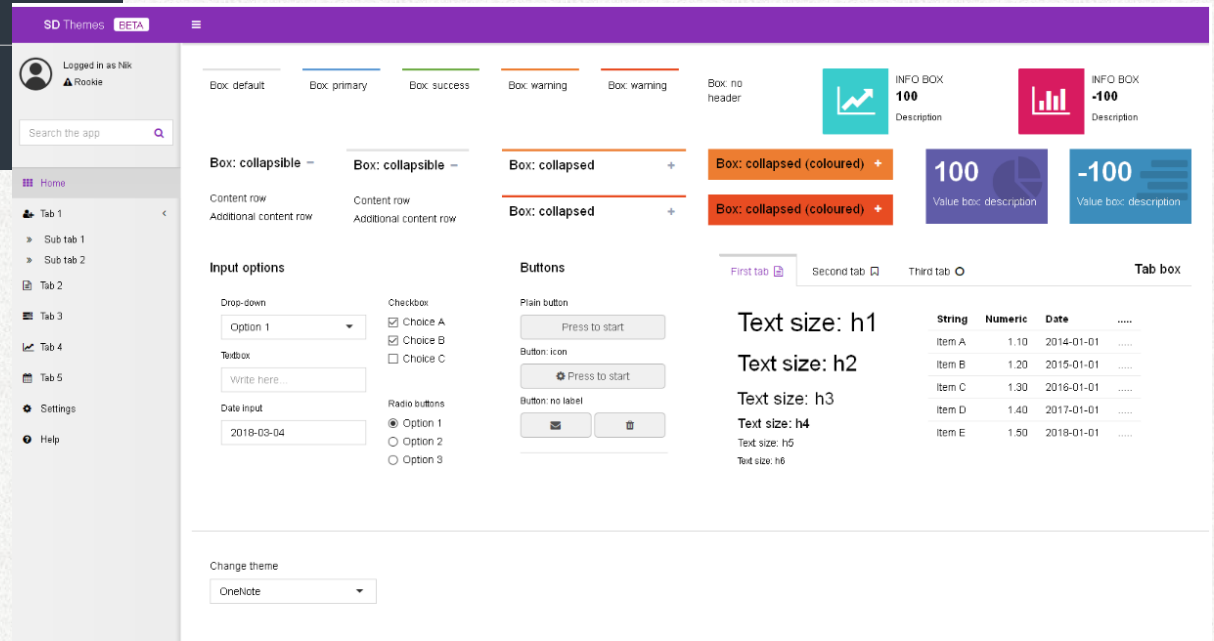
Text input

Enter text...

textInput()

Como dar vida ao seu dashboard ?





Programando um dashboard

SERVER.R

```
server <- function(input, output) {

  output$tabatletas = renderDT(
    atletas, options = list(lengthChange = FALSE)
  )

  output$aquatics <- renderValueBox({
    valueBox(
      paste0("Esportes Aquáticos","-",
             as.numeric(table(eventos$sport)[1])), "50m livre masculino, 50m livre feminino, etc.", icon = icon("trophy"),
      color = "blue"
    )
  })
```

```
output$ReceitaDin <- renderValueBox({
  valueBox(
    paste0("R$ ",formatC(sum(DADOS$DINHEIRO), format="d", big.mark="."))
  , "Dinheiro - Janeiro | 2019", icon = icon("money"),
    color = "blue"
  )
})
```

SERVER.R

```
output$plot <- renderPlotly({
  plot_ly(DADOS, x = ~DATA, y = ~DINHEIRO, name = 'Dinheiro', type = 'scatter', mode = 'lines') %>%
    add_trace(y = ~CREDITO, name = 'Crédito', mode = 'lines') %>%
    add_trace(y = ~DEBITO, name = 'Débito', mode = 'lines')
})
```

```
output$results <- renderTable({

  if(input$CutSelect=="Todos"){filtered <- produto} else {
  filtered <-
    produto %>%
    filter(MEIOPAGAMENTO == input$CutSelect)

  }

  filtered
  })
```

UI.R

```r
## CONSTRUINDO A INTERFACE GRAFICA ##
ui <- dashboardPage(
  |

  ## CABEÇALHO DA PAGINA ##
  dashboardHeader(title = "Dashboard", titleWidth = 300,

                  ## CRIA MENU DE NOTIFICACOES ##

                  dropdownMenu(type = "notifications",
                               notificationItem(
                                 text = "EER! no Youtube: 593 inscritos.",
                                 icon("users")
                               ),
                               notificationItem(
                                 text = "EER! no Facebook: 1.549 seguidores",
                                 icon("users")
                               )
                  ),

                  ## CRIA MENU DE MENSAGENS ##

                  dropdownMenu(type = "message",
                               messageItem(
                                 from = "Estatística é com R!",
                                 message = "IV SER: 21-23 de Maio de 2019.",
                                 icon = icon("calendar")
                  )
                  )


                  ),
```

UI.R

```r
## BARRA LATERAL DA PAGINA ##
dashboardSidebar(
  width = 300,

  ## CRIACAO DO MENU LATERAL ##
  sidebarMenu(

    #Item1
    menuItem("Página Inicial", tabName = "Home", icon = icon("home")),

    #Item2
    menuItem("Rio 2016", tabName = "Jogos", icon = icon("trophy"),
             #SubItem2.1
             menuSubItem("Atletas", tabName = "Jogos1"),
             #SubItem2.2
             menuSubItem("Eventos", tabName = "Jogos2")

             ),
```

UI.R

```
## PAGINA DO DASHBOARD (CORPO) ##
dashboardBody(

  ## DEFININDO O TEMA ##
  shinyDashboardThemes(theme = "boe_website"),

  #DEFININDO A LARGURA DAS NOTIFICACOES
  tags$head(tags$style(HTML('
.navbar-custom-menu>.navbar-nav>li>.dropdown-menu {
width:500px;
}
'))),

  ## ASSOCIANDO O CONTEUDO DA PAGINA COM OS ITENS DO MENU ##
  tabItems(

  # Item1
  tabItem("Home",
            fluidPage(column(12,h3("Descobrindo o Shiny: Como tornar sua visualização de dados mais atraente?"))
              ,column(12,tags$img(src='bannerlegenda.png',height="100%", width="100%")),
              column(12,h4("Palestrante: Camila Simões - camilasdsimoes@gmail.com")),
                          column(12,h4("Data: 05/02/2019")))),
```

UI.R

```
# SubItem2.1
tabItem("Jogos1", h2("Atletas | Jogos Olímpicos - Rio 2016"),
        fluidPage(DTOutput('tabatletas'))),

# SubItem2.2
tabItem("Jogos2", h2("Eventos | Jogos Olímpicos - Rio 2016"),

        ## CRIA BOX VALUE ##
        fluidRow(valueBoxOutput("aquatics",width = 5), valueBoxOutput("gymnastics",width = 5)),
        fluidRow(valueBoxOutput("basketball",width = 5), valueBoxOutput("hand",width = 5)),
        fluidRow(valueBoxOutput("canoagem",width = 5), valueBoxOutput("tdm",width = 5)),
        fluidRow(valueBoxOutput("ciclismo",width = 5), valueBoxOutput("tenis",width = 5)),
        fluidRow(valueBoxOutput("fut",width = 5)      , valueBoxOutput("volei",width = 5))),

# SubItem3.1
tabItem("Produto2",h2("Monitoramento de Indicadores do Produto"),

     fluidRow( valueBoxOutput("ReceitaDin",width = 4),valueBoxOutput("ReceitaCred",width = 4),valueBoxOutput("ReceitaDeb",width = 4) ),

     ## PLOTA GRÁFICO DO PLOTLY ##
     fluidRow( plotlyOutput("plot") )

        ),

# SubItem3.2
tabItem("Produto3",h2("Venda Detalhada do Produto"),

        fluidPage( column(3,
                          selectizeInput("CutSelect", "Meio de Pagamento", c("Todos",unique(produto$MEIOPAGAMENTO)), selected = NULL, multiple = F),
                          br() ), tableOutput("results"))

   ),
```

UI.R

```
    # Item 4
    tabItem(
      tabName = "HTML",

      tags$div(

      tags$p (tags$a(href="http://www.estatisticacomr.uff.br", "Estatística é com R! - Portal")),

      tags$p (tags$a(href="https://www.facebook.com/estatisticacomr/", "Estatística é com R! - Facebook")),

      tags$p(tags$a(href="https://www.youtube.com/channel/UCmbNWlpq8o3dpqY6c9HDGXg", "Estatística é com R! - Youtube"))

      )),

    # Item 5
    tabItem(tabName = "Pacotes", h2("Lista de Pacotes"),
            fluidPage(column(12,h3(" • shiny")),
                      column(12,h3(" • shinydashboard")),
                      column(12,h3(" • dashboardthemes")),
                      column(12,h3(" • DT")),
                      column(12,h3(" • dplyr")),
                      column(12,h3(" • plotly"))))

    )



  )
)
```

Resultado

Dashboard

127.0.0.1:4087

Dashboard

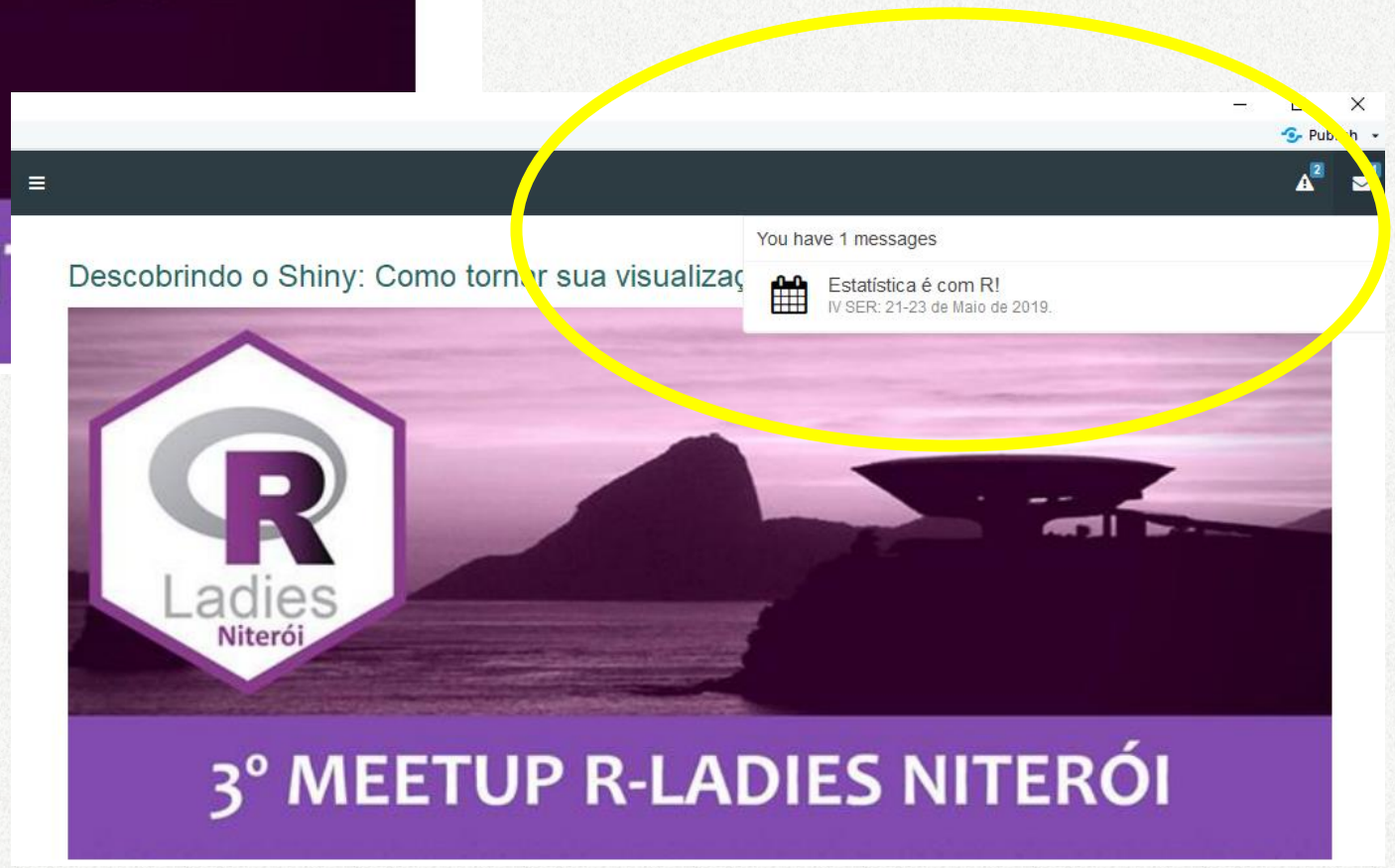
- Página Inicial
- Rio 2016
- Meu Produto
- Páginas - Estatística é com R!
- Pacotes Utilizados

Descobrindo o Shiny: Como tornar sua visualização de dados mais atraente?

Palestrante: Camila Simões - camilasdsimoes@gmail.com

Data: 05/02/2019


Publish
You have 2 notifications
EER! no Youtube: 593 inscritos.
EER! no Facebook: 1.549 seguidores
Descobrindo o Shiny: Como tornar sua visua
R
Ladies
Niterói
3º MEETUP R-LADIES NI


You have 1 messages
Estatística é com R!
IV SER: 21-23 de Maio de 2019.
Descobrindo o Shiny: Como tornar sua visualiza
R
Ladies
Niterói
3º MEETUP R-LADIES NITERÓI

~/Palesta R-Ladies/scriptunico.R - Shiny

http://127.0.0.1:4087 | Open in Browser | Publish

Dashboard

- Página Inicial
- Rio 2016
  - Atletas
  - Eventos
- Meu Produto
- Páginas - Estatística é com R!
- Pacotes Utilizados

Atletas | Jogos Olímpicos - Rio 2016

Search:

	id	name	nationality	sex	dob	height	weight	sport	gold	silver	bronze	medals
1	736041664	A Jesus Garcia	ESP	male	10/17/69	1.72	64	athletics	0	0	0	0
2	532037425	A Lam Shin	KOR	female	9/23/86	1.68	56	fencing	0	0	0	0
3	435962603	Aaron Brown	CAN	male	5/27/92	1.98	79	athletics	0	0	1	1
4	521041435	Aaron Cook	MDA	male	1/2/91	1.83	80	taekwondo	0	0	0	0
5	33922579	Aaron Gate	NZL	male	11/26/90	1.81	71	cycling	0	0	0	0
6	173071782	Aaron Royle	AUS	male	1/26/90	1.8	67	triathlon	0	0	0	0
7	266237702	Aaron Russell	USA	male	6/4/93	2.05	98	volleyball	0	0	1	1
8	382571888	Aaron Younger	AUS	male	9/25/91	1.93	100	aquatics	0	0	0	0
		Aauri										

~/Palesta R-Ladies/scriptunico.R - Shiny

http://127.0.0.1:4087 | Open in Browser | Publish

Dashboard

- Página Inicial
- Rio 2016
  - Atletas
  - Eventos
- Meu Produto
- Páginas - Estatística é com R!
- Pacotes Utilizados

Eventos | Jogos Olímpicos - Rio 2016

Esportes Aquáticos-4
50m livre masculino, 50m livre feminino, etc.

Ginástica-18
Individual, Time, Masculino, Feminino, etc.

Basquete-2
Basquete Masculino e Basquete Masculino

Handebol-2
Handebol Masculino e Feminino

Canoagem-16
Dupla, Individual, 200m, 1000m, etc.

Tênis de Mesa-4
Masculino, Feminino, Dupla e Individual

Ciclismo-18
Masculino, Feminino, Individual, Time, etc.

Tênis-5
Masculino, Feminino, Misto, Dupla e Individual

Futebol-2

Vôlei-4

~/Palesta R-Ladies/scriptunico.R - Shiny

http://127.0.0.1:4087 | Open in Browser | Publish

Dashboard

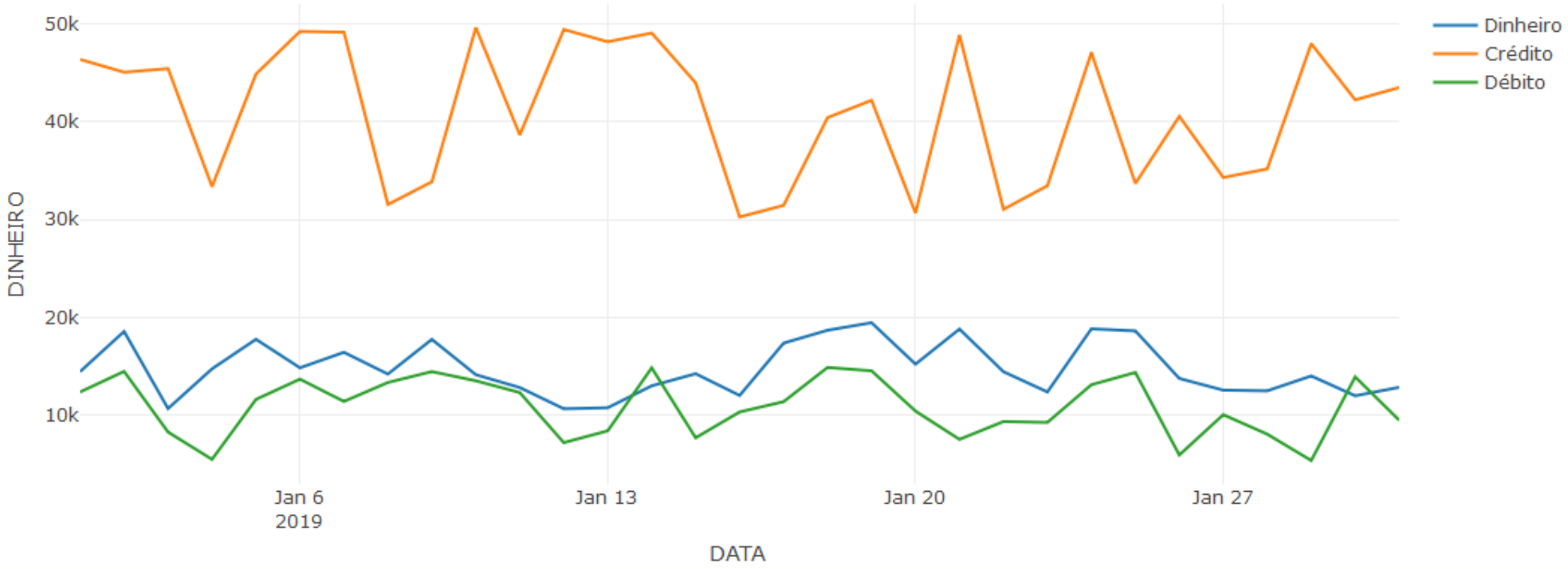
- Página Inicial
- Rio 2016
- Meu Produto
  - Fechamento de Vendas
  - Venda Diária
- Páginas - Estatística é com R!
- Pacotes Utilizados

Monitoramento de Indicadores do Produto

R$ 458.624
Dinheiro - Janeiro | 2019

R$ 1.271.365
Crédito - Janeiro | 2019

R$ 337.072
Débito - Janeiro | 2019



~/Palesta R-Ladies/scriptunico.R - Shiny

http://127.0.0.1:4087 | Open in Browser | Publish

Dashboard

- Página Inicial
- Rio 2016
- Meu Produto
  - Fechamento de Vendas
  - Venda Diária
- Páginas - Estatística é com R!
- Pacotes Utilizados

Venda Detalhada do Produto

Meio de Pagamento

Todos

- Todos
- Dinheiro
- Credito
- Debito

DATA	RECEITA	MEIOPAGAMENTO
2019-01-01	14459	Dinheiro
2019-01-02	18567	Dinheiro
2019-01-03	10677	Dinheiro
2019-01-04	14741	Dinheiro
2019-01-05	17765	Dinheiro
2019-01-06	14848	Dinheiro
2019-01-07	16433	Dinheiro
2019-01-08	14208	Dinheiro
2019-01-09	17756	Dinheiro
2019-01-10	14141	Dinheiro
2019-01-11	12829	Dinheiro
2019-01-12	10660	Dinheiro
2019-01-13	10760	Dinheiro
2019-01-14	13006	Dinheiro
2019-01-15	14243	Dinheiro
2019-01-16	12020	Dinheiro
2019-01-17	17374	Dinheiro

~/Palesta R-Ladies/scriptunico.R - Shiny

http://127.0.0.1:4087 | Open in Browser | Publish

Dashboard

- Página Inicial
- Rio 2016
- Meu Produto
  - Fechamento de Vendas
  - Venda Diária
- Páginas - Estatística é com R!
- Pacotes Utilizados

Estatística é com R! - Portal

Estatística é com R! - Facebook

Estatística é com R! - Youtube

~/Palesta R-Ladies/scriptunico.R - Shiny

http://127.0.0.1:4087 Open in Browser Publish

Dashboard

- Página Inicial
- Rio 2016
- Meu Produto
  - Fechamento de Vendas
  - Venda Diária
- Páginas - Estatística é com R!
- Pacotes Utilizados

Lista de Pacotes

- shiny
- shinydashboard
- dashboardthemes
- DT
- dplyr
- plotly

Conclusão

Compartilhando seu ShinyAPP

- Shiny Server Pro – Versão comercial do Shiny-Server
- shinyapps.io – Necessita do Linux ou de algum serviço de nuvem (AWS, por exemplo)
- Pacote RInno

Sugestão de Material – Resumo e Código

- RStudio Gallery
- Cheat Sheet - RStudio

Resumo

Interactive Web Apps with shiny Cheat Sheet

learn more at shiny.rstudio.com

RStudio

Basics

A **Shiny** app is a web page (**UI**) connected to a computer running a live R session (**Server**)

Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

App template

Begin writing a new app with this template. Preview the app by running the code at the R command line.

```
library(shiny)
ui <- fluidPage()
server <- function(input, output){}
shinyApp(ui = ui, server = server)
```

- **ui** - nested R functions that assemble an HTML user interface for your app
- **server** - a function with instructions on how to build and rebuild the R objects displayed in the UI
- **shinyApp** - combines `ui` and `server` into a functioning app. Wrap with **runApp()** if calling from a sourced script or inside a function.

Share your app

The easiest way to share your app is to host it on shinyapps.io, a cloud based service from RStudio

1. Create a free or professional account at **http://shinyapps.io**
2. Click the **Publish** icon in the RStudio IDE (>=0.99) or run: **rsconnect::deployApp("<path to directory>")**

Build or purchase your own Shiny Server at www.rstudio.com/products/shiny-server/

Building an App - Complete the template by adding arguments to fluidPage() and a body to the server function.

Add inputs to the UI with ***Input()** functions

Add outputs with ***Output()** functions

Tell server how to render outputs with R in the server function. To do this:

1. Refer to outputs with **output$<id>**
2. Refer to inputs with **input$<id>**
3. Wrap code in a **render*()** function before saving to output

```
library(shiny)

ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}

shinyApp(ui = ui, server = server)
```

Save your template as **app.R**. Alternatively, split your template into two files named **ui.R** and **server.R**.

```
library(shiny)

ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}

shinyApp(ui = ui, server = server)
```

```
# ui.R
fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)
```

ui.R contains everything you would save to ui.

```
# server.R
function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
```

server.R ends with the function you would save to server.

No need to call **shinyApp()**.

Save each app as a directory that contains an **app.R** file (or a **server.R** file and a **ui.R** file) plus optional extra files.

- app-name
  - app.R — The directory name is the name of the app
  - global.R — (optional) defines objects available to both ui.R and server.R
  - DESCRIPTION — (optional) used in showcase mode
  - README — (optional) data, scripts, etc.
  - <other files> — (optional) directory of files to share with web browsers (images, CSS, .js, etc.) Must be named "**www**"
  - www

Launch apps with `runApp(<path to directory>)`

Outputs - render*() and *Output() functions work together to add R output to the UI

		works with	
	DT::**renderDataTable**(expr, options, callback, escape, env, quoted)		**dataTableOutput**(outputId, icon, ...)
	renderImage(expr, env, quoted, deleteFile)		**imageOutput**(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
	renderPlot(expr, width, height, res, ..., env, quoted, func)		**plotOutput**(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
	renderPrint(expr, env, quoted, func, width)		**verbatimTextOutput**(outputId)
	renderTable(expr, ..., env, quoted, func)		**tableOutput**(outputId)
foo	**renderText**(expr, env, quoted, func)		**textOutput**(outputId, container, inline)
	renderUI(expr, env, quoted, func)		**uiOutput**(outputId, inline, container, ...) & **htmlOutput**(outputId, inline, container, ...)

Inputs - collect values from the user

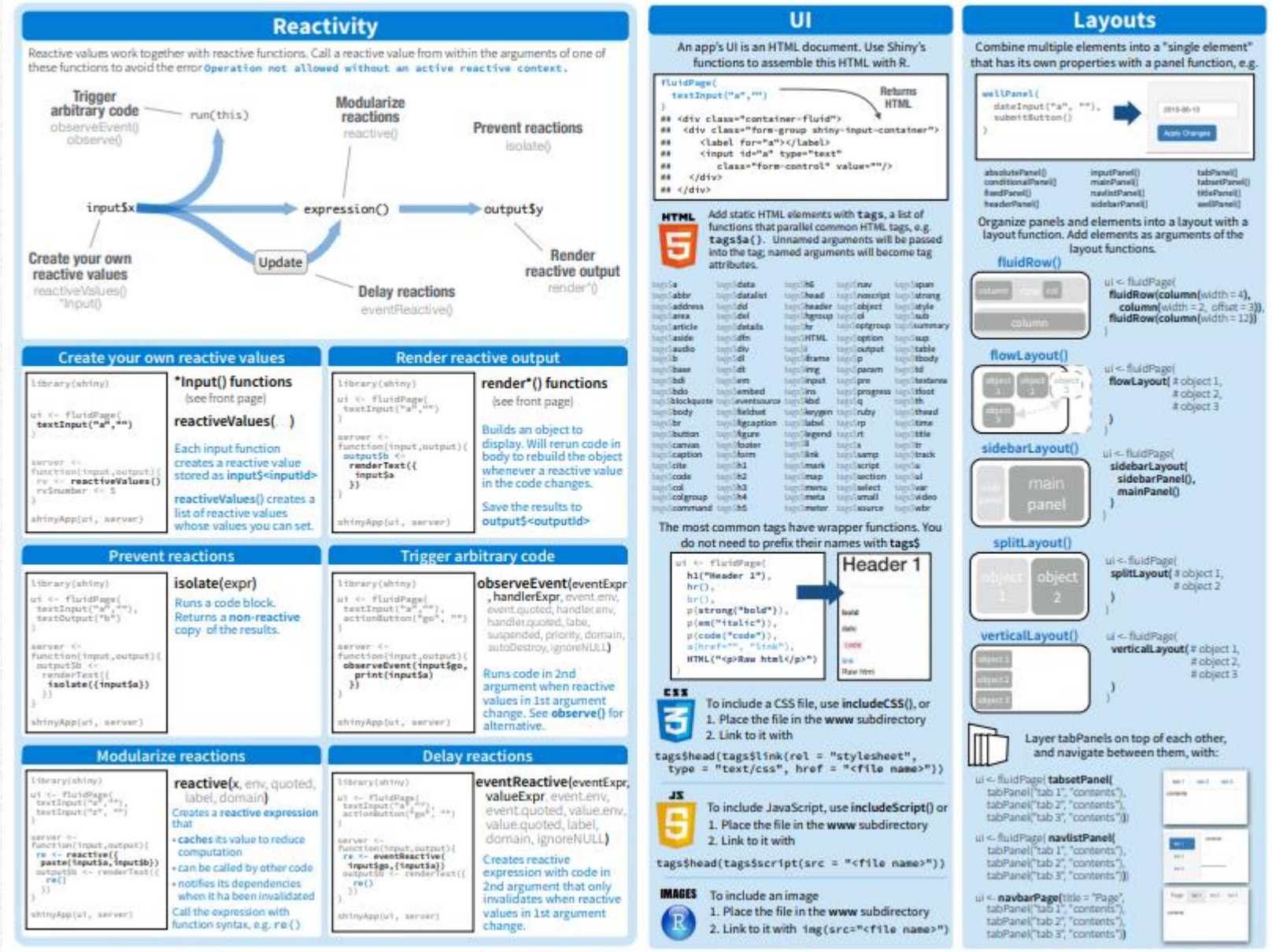
Access the current value of an input object with **input $<inputId>**. Input values are **reactive**.

Action	**actionButton**(inputId, label, icon, ...)
Link	**actionLink**(inputId, label, icon, ...)
Choice 1, Choice 2, Choice 3	**checkboxGroupInput**(inputId, label, choices, selected, inline)
Check me	**checkboxInput**(inputId, label, value)
	dateInput(inputId, label, value, min, max, format, startview, weekstart, language)
	dateRangeInput(inputId, label, start, end, min, max, format, startview, weekstart, language, separator)
Choose File	**fileInput**(inputId, label, multiple, accept)
1	**numericInput**(inputId, label, value, min, max, step)
•••••••••	**passwordInput**(inputId, label, value)
Choice A, Choice B, Choice C	**radioButtons**(inputId, label, choices, selected, inline)
Choice 1, Choice 1, Choice 2	**selectInput**(inputId, label, choices, selected, multiple, selectize, width, size) (also **selectizeInput()**)
0 0 10	**sliderInput**(inputId, label, min, max, value, step, round, format, locale, ticks, animate, width, sep, pre, post)
Apply Changes	**submitButton**(text, icon) (Prevents reactions across entire app)
Enter text	**textInput**(inputId, label, value)

More cheat sheets at http://www.rstudio.com/resources/cheatsheets/

Learn more at shiny.rstudio.com/tutorial • shiny 0.12.0 • Updated: 6/15

Resumo


Reactivity
Reactive values work together with reactive functions. Call a reactive value from within the arguments of one of these functions to avoid the error Operation not allowed without an active reactive context.
Trigger arbitrary code
observeEvent()
observe()
run(this)
Modularize reactions
reactive()
Prevent reactions
isolate()
input$x
expression()
output$y
Create your own reactive values
reactiveValues()
*Input()
Update
Delay reactions
eventReactive()
Render reactive output
render*()
Create your own reactive values
*Input() functions
(see front page)
reactiveValues(...)
Each input function creates a reactive value stored as input$<inputId>
reactiveValues() creates a list of reactive values whose values you can set.
Render reactive output
render*() functions
(see front page)
Builds an object to display. Will rerun code in body to rebuild the object whenever a reactive value in the code changes.
Save the results to output$<outputId>
Prevent reactions
isolate(expr)
Runs a code block. Returns a non-reactive copy of the results.
Trigger arbitrary code
observeEvent(eventExpr, handlerExpr, event.env, event.quoted, handler.env, handler.quoted, label, suspended, priority, domain, autoDestroy, ignoreNULL)
Runs code in 2nd argument when reactive values in 1st argument change. See observe() for alternative.
Modularize reactions
reactive(x, env, quoted, label, domain)
Creates a reactive expression that
• caches its value to reduce computation
• can be called by other code
• notifies its dependencies when it ha been invalidated
Call the expression with function syntax, e.g. re()
Delay reactions
eventReactive(eventExpr, valueExpr, event.env, event.quoted, value.env, value.quoted, label, domain, ignoreNULL)
Creates reactive expression with code in 2nd argument that only invalidates when reactive values in 1st argument change.
UI
An app's UI is an HTML document. Use Shiny's functions to assemble this HTML with R.
Returns HTML
HTML
Add static HTML elements with tags, a list of functions that parallel common HTML tags, e.g. tags$a(). Unnamed arguments will be passed into the tag; named arguments will become tag attributes.
The most common tags have wrapper functions. You do not need to prefix their names with tags$
Header 1
CSS
To include a CSS file, use includeCSS(), or
1. Place the file in the www subdirectory
2. Link to it with
tags$head(tags$link(rel = "stylesheet", type = "text/css", href = "<file name>"))
JS
To include JavaScript, use includeScript() or
1. Place the file in the www subdirectory
2. Link to it with
tags$head(tags$script(src = "<file name>"))
IMAGES
To include an image
1. Place the file in the www subdirectory
2. Link to it with img(src="<file name>")
Layouts
Combine multiple elements into a "single element" that has its own properties with a panel function, e.g.
Organize panels and elements into a layout with a layout function. Add elements as arguments of the layout functions.
fluidRow()
flowLayout()
sidebarLayout()
main panel
splitLayout()
verticalLayout()
Layer tabPanels on top of each other, and navigate between them, with:
tabsetPanel()
navlistPanel()
navbarPage()
RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com
More cheat sheets at http://www.rstudio.com/resources/cheatsheets/
Learn more at shiny.rstudio.com/tutorial • shiny 0.12.0 • Updated: 6/15

Obrigada pela atenção!