

Introduction to Shiny



Agenda & Topics

► Agenda:

- What is Shiny?
- Why should I learn Shiny?
- What can I do with Shiny?

► Topics:

- Shiny app architecture
- User Interface and Server scripts
- Reactivity / Widgets / Functions
- How to share your app

What is Shiny?

Shiny is an R package that allows you to design and build interactive web applications using only .



In its simplest form, a Shiny application is an R script that looks like ...

```
library(shiny) # load the necessary package



ui <- fluidPage() # lay out the appearance of your app

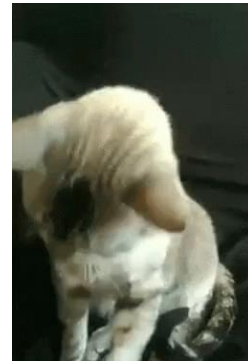
server <- function(input, output) {} # instructions needed to build the app

shinyApp(ui = ui, server = server) # combine ui and server to launch the app
```

Why should I learn Shiny?

In the past, creating an interactive web application was **not easy**, it required **different technologies** and you had to deal with **complex interaction** flows... good news ... not anymore!

1. You do not need to know how to code in HTML, CSS, or JavaScript
2. It is easy to write in  and develop the application using  Studio
3. It is a great way to share your work.
4. It makes you look cool!



What's under the hood?

The **{shiny}** package uses HTML to generate the web application with Bootstrap as the framework to handle HTML, CSS and JS.



```
fluidPage(  
  textInput(inputId = "name", label = "what is shiny?")  
)  
  
<div class="container-fluid">  
  <div class="form-group shiny-input-container">  
    <label class="control-label" for="name">What is Shiny?</label>  
    <input id="name" type="text" class="form-control" value="" />  
  </div>  
</div>
```



Remember you don't need to know HTML, CSS or JS but if you do, then you can customise your app!

Simple example - Demo

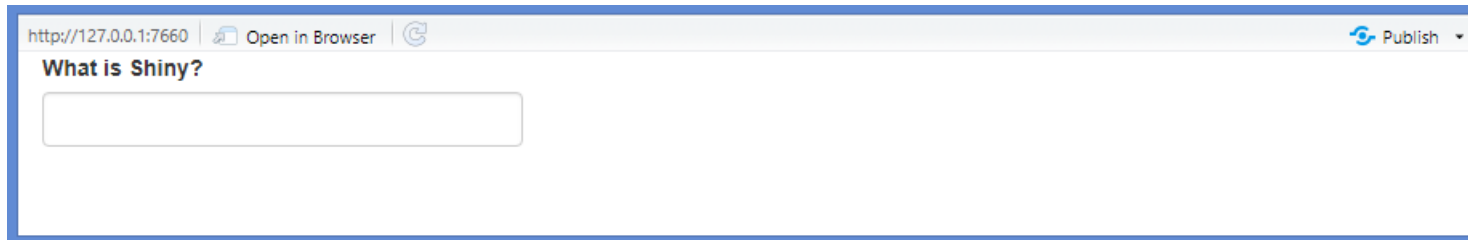
```
# Simple example

library(shiny) # load the necessary package

ui <- fluidPage( # lay out the appearance of your app
  textInput(inputId = "name", label = "what is shiny?")
)

server <- function(input, output) {} # instructions needed to build the app

shinyApp(ui = ui, server = server) # combine ui and server to launch the app
```



R (version 3.6.3)
shiny (version: 1.4.0.2)

What can I do with Shiny?

Bring your data to life with user friendly interaction!

There are so many exciting applications that you can build using the **{shiny}** package! Some examples are:

- Provide a self-service platform for data analysis and visualisations
- Build dashboards to track performance indicators and other metrics
- Interpret complex models to non-technical audience with “what-if” scenario analysis
- Create a web tool to support customer service department
- Track global pandemics
- ...

Lets see some examples! - Demo

```
runExample(example = "01_hello")
```

valid examples are:

- ✓ "01_hello"
- "02_text"
- ✓ "03_reactivity"
- "04_mpg"
- "05_sliders"
- ✓ "06_tabsets"
- "07_widgets"
- "08_html"
- "09_upload"
- "10_download"
- "11_timer"

Lets see some fancy examples! - Demo

- ▶ https://vac-lshtm.shinyapps.io/ncov_tracker/
- ▶ <https://gallery.shinyapps.io/nz-trade-dash/>
- ▶ <https://dreamrs.shinyapps.io/memory-hex/>
- ▶ more at <https://shiny.rstudio.com/gallery/>
- ▶ also check out public GitHub repos

Architecture of a shiny app

Some the “ingredients” we will need in order to create a shiny app are:

1. For the **User Interface** we need to supply:
 - a) Inputs
 - b) Outputs
 - c) Any other structures like navigation bars, tabs, boxes etc.
2. For the **Server** side we need to write the code that:
 - a) Handles the reactivity and the data
 - b) Creates the visuals (like plots, tables etc.)
 - c) Executes code from user actions



The User Interface script

ui.R

- ▶ Designs and structures the layout of the application
- ▶ Defines what the user sees and interacts with

The server script

server.R

- ▶ Defines the server-side logic of the application
- ▶ Contains the R code and other instructions to execute

Example: ui.R script

```
# Define UI for application
ui <- fluidPage(

  # Application title
  titlePanel("Hello Lucerne!"),

  # Sidebar with an input
  sidebarLayout(
    sidebarPanel(
      textInput("text_input", "Input text here:")
    ),

    # Main with output
    mainPanel(
      textOutput("text_output")
    )
  )
)
```

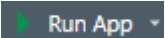
Example: server.R script

```
library(shiny)

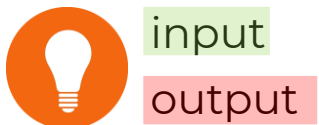
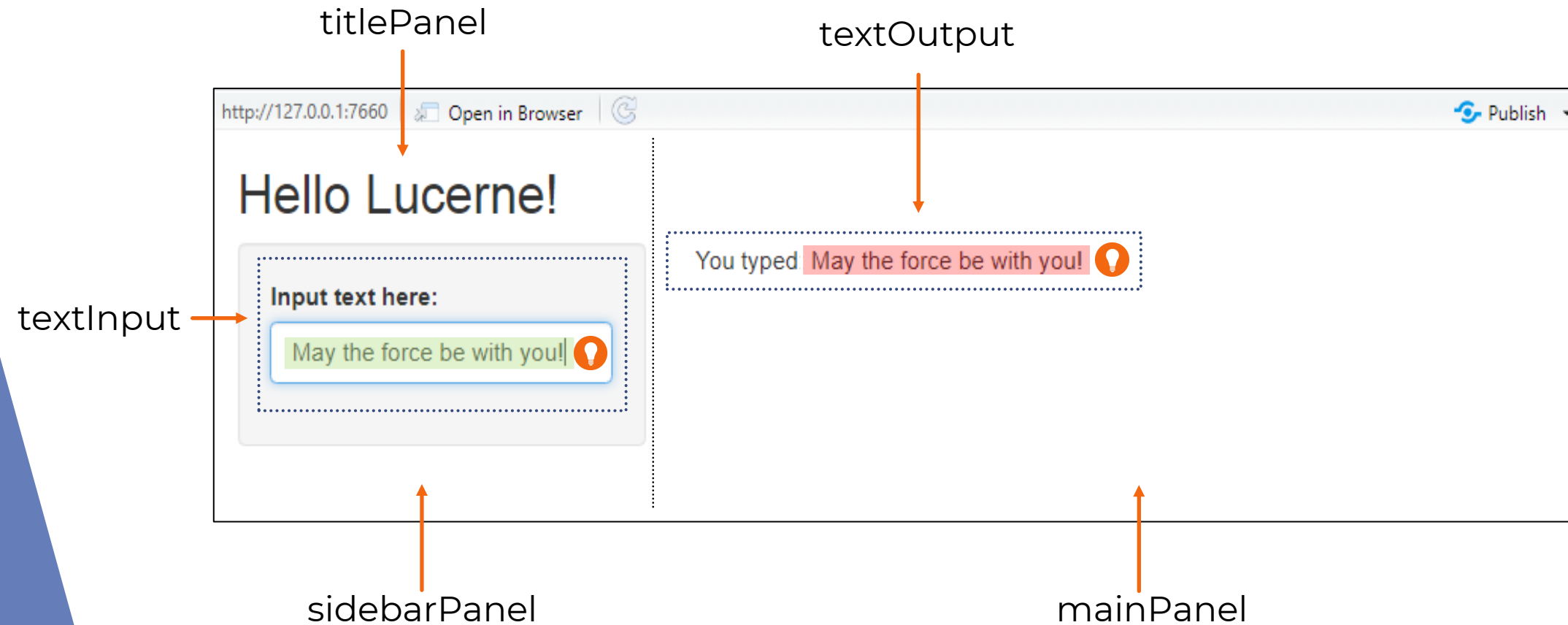
# Define server logic and R code
server <- function(input, output) {

  output$text_output <- renderText({
    # Display text input
    paste("You typed:", input$text_input)
  })
}
```



You can press the  button to start the app!

Architecture of a shiny app - Demo



Reactivity

This is what makes a shiny app responsive to user interactions.

- ▶ It happens automatically when the values of **inputs** are changed.
- ▶ The updated input values are passed on to the server.
- ▶ The server executes and runs any R code relating to the inputs.
- ▶ The updated **outputs** are rendered and returned to the app.

Widgets

function	widget
actionButton	Action Button
submitButton	A submit button
checkboxInput	A single check box
dateInput	A calendar to aid date selection
numericInput	A field to enter numbers
radioButtons	A set of radio buttons
selectInput	A box with choices to select from
sliderInput	A slider bar
textInput	A field to enter text

Buttons

Single checkbox
☒ Choice A

Checkbox group
☒ Choice 1
☐ Choice 2
☐ Choice 3

Date input

Date range
 to

File input

Help text

Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

Numeric input

Radio buttons
☒ Choice 1
☐ Choice 2
☐ Choice 3

Select box

Sliders

Text input



<https://shiny.rstudio.com/tutorial/written-tutorial/lesson3/>

Functions

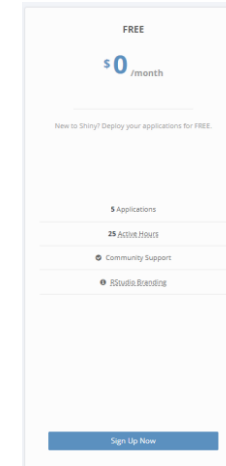
ui.R Output function	server.R render function	Creates
dataTableOutput	renderDataTable	DataTable
imageOutput	renderImage	image
plotOutput	renderPlot	plot
tableOutput	renderTable	table
textOutput	renderText	text
uiOutput	renderUI	raw HTML

 <https://shiny.rstudio.com/tutorial/written-tutorial/lesson4/>

How to share your app


- ▶ Shinyapps.io <https://www.shinyapps.io/>

An easy way to share you application that is secure and scalable utilising a server that is maintained by RStudio. There is a free tier available!



- ▶ Shiny Server Open Source <https://www.rstudio.com/products/shiny/shiny-server/>

You can have your own server to host your applications. This also allows you to customise each app to have its own URL.

 **DOWNLOAD SHINY
SERVER OPEN SOURCE**



You can follow the BarcelonaR workshop on: [Introduction to RStudio and Shiny servers](#) on how to setup your own server for free using Google Cloud. Also see:

https://github.com/nattalides/BarcelonaR_workshop_Introduction_to_RStudio_and_Shiny_servers

Other resources – Shiny cheat sheet

Interactive Web Apps with shiny Cheat Sheet

learn more at shiny.rstudio.com

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

- Create a free or professional account at <http://shinyapps.io>
- 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/

RStudio® is a trademark of RStudio, Inc. • www.rstudio.com • 844-448-1212 • <http://www.rstudio.com/resources/cheatsheets/>

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:

- Refer to outputs with `output$<id>`
- Refer to inputs with `input$<id>`
- Wrap code in a `render()` function before saving to output

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

ui.R contains everything you would save to ui.

```
library(shiny)
ui <- fluidPage(
  numericInput("input", "Sample size", value = 25),
  plotOutput("output")
)
```

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

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

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.

The directory name is the name of the app (optional) defines objects available to both ui.R and server.R

- DESCRIPTION** - optional used in showcase mode
- README** - optional directory of files to share with web browsers (images, CSS, JS, etc.) Must be named "www"
- www** - optional directory of files to share with web browsers (images, CSS, JS, etc.) Must be named "www"

Launch apps with `runApp("path to directory")`

Outputs - render()

DT::renderDataTable() (expr, options, callback, escape, env, quoted)

renderImage() (expr, env, quoted, deleteFile, click, hover, hoverDelay, hoverDelayType, brush, click, hoverId, inline)

renderPlot() (expr, width, height, res, ..., env, quoted, fun)

renderPrint() (expr, env, quoted, fun, width)

renderTable() (expr, ..., env, quoted, fun)

renderText() (expr, env, quoted, fun)

renderUI() (expr, env, quoted, fun)

dataTableOutput() (outputId, icon, ...)

imageOutput() (outputId, width, height, click, clickDelay, hover, hoverDelay, hoverDelayType, brush, click, hoverId, inline)

plotOutput() (outputId, width, height, click, clickDelay, hover, hoverDelay, hoverDelayType, brush, click, hoverId, inline)

verbatimTextOutput() (outputId)

tableOutput() (outputId)

textOutput() (outputId, container, inline)

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

Link

Choice 1

Choice 2

Choice 3

Choice 4

Choice 5

Choice 6

Choice 7

Choice 8

Choice 9

Choice 10

Choice 11

Choice 12

Choice 13

Choice 14

Choice 15

Choice 16

Choice 17

Choice 18

Choice 19

Choice 20

Choice 21

Choice 22

Choice 23

Choice 24

Choice 25

Choice 26

Choice 27

Choice 28

Choice 29

Choice 30

Choice 31

Choice 32

Choice 33

Choice 34

Choice 35

Choice 36

Choice 37

Choice 38

Choice 39

Choice 40

Choice 41

Choice 42

Choice 43

Choice 44

Choice 45

Choice 46

Choice 47

Choice 48

Choice 49

Choice 50

Choice 51

Choice 52

Choice 53

Choice 54

Choice 55

Choice 56

Choice 57

Choice 58

Choice 59

Choice 60

Choice 61

Choice 62

Choice 63

Choice 64

Choice 65

Choice 66

Choice 67

Choice 68

Choice 69

Choice 70

Choice 71

Choice 72

Choice 73

Choice 74

Choice 75

Choice 76

Choice 77

Choice 78

Choice 79

Choice 80

Choice 81

Choice 82

Choice 83

Choice 84

Choice 85

Choice 86

Choice 87

Choice 88

Choice 89

Choice 90

Choice 91

Choice 92

Choice 93

Choice 94

Choice 95

Choice 96

Choice 97

Choice 98

Choice 99

Choice 100

Choice 101

Choice 102

Choice 103

Choice 104

Choice 105

Choice 106

Choice 107

Choice 108

Choice 109

Choice 110

Choice 111

Choice 112

Choice 113

Choice 114

Choice 115

Choice 116

Choice 117

Choice 118

Choice 119

Choice 120

Choice 121

Choice 122

Choice 123

Choice 124

Choice 125

Choice 126

Choice 127

Choice 128

Choice 129

Choice 130

Choice 131

Choice 132

Choice 133

Choice 134

Choice 135

Choice 136

Choice 137

Choice 138

Choice 139

Choice 140

Choice 141

Choice 142

Choice 143

Choice 144

Choice 145

Choice 146

Choice 147

Choice 148

Choice 149

Choice 150

Choice 151

Choice 152

Choice 153

Choice 154

Choice 155

Choice 156

Choice 157

Choice 158

Choice 159

Choice 160

Choice 161

Choice 162

Choice 163

Choice 164

Choice 165

Choice 166

Choice 167

Choice 168

Choice 169

Choice 170

Choice 171

Choice 172

Choice 173

Choice 174

Choice 175

Choice 176

Choice 177

Choice 178

Choice 179

Choice 180

Choice 181

Choice 182

Choice 183

Choice 184

Choice 185

Choice 186

Choice 187

Choice 188

Choice 189

Choice 190

Choice 191

Choice 192

Choice 193

Choice 194

Choice 195

Choice 196

Choice 197

Choice 198

Choice 199

Choice 200

Choice 201

Choice 202

Choice 203

Choice 204

Choice 205

Choice 206

Choice 207

Choice 208

Choice 209

Choice 210

Choice 211

Choice 212

Choice 213

Choice 214

Choice 215

Choice 216

Choice 217

Choice 218

Choice 219

Choice 220

Choice 221

Choice 222

Choice 223

Choice 224

Choice 225

Choice 226

Choice 227

Choice 228

Choice 229

Choice 230

Choice 231

Choice 232

Choice 233

Choice 234

Choice 235

Choice 236

Choice 237

Choice 238

Choice 239

Choice 240

Choice 241

Choice 242

Choice 243

Choice 244

Choice 245

Choice 246

Choice 247

Choice 248

Choice 249

Choice 250

Choice 251

Choice 252

Choice 253

Choice 254

Choice 255

Choice 256

Choice 257

Choice 258

Choice 259

Choice 260

Choice 261

Choice 262

Choice 263

Choice 264

Choice 265

Choice 266

Choice 267

Choice 268

Choice 269

Choice 270

Choice 271

Choice 272

Choice 273

Choice 274

Choice 275

Choice 276

Choice 277

Choice 278

Choice 279

Choice 280

Choice 281

Choice 282

Choice 283

Choice 284

Choice 285

Choice 286

Choice 287

Choice 288

Choice 289

Choice 290

Choice 291

Choice 292

Choice 293

Choice 294

Choice 295

Choice 296

Choice 297

Choice 298

Choice 299

Choice 300

Choice 301

Choice 302

Choice 303

Choice 304

Choice 305

Choice 306

Choice 307

Choice 308

Choice 309

Choice 310

Choice 311

Choice 312

Choice 313

Choice 314

Choice 315

Choice 316

Choice 317

Choice 318

Choice 319

Choice 320

Choice 321

Choice 322

Choice 323

Choice 324

Choice 325

Choice 326

Choice 327

Choice 328

Choice 329

Choice 330

Choice 331

Choice 332

Choice 333

Choice 334

Choice 335

Choice 336

Choice 337

Choice 338

Choice 339

Choice 340

Choice 341

Choice 342

Choice 343

Choice 344

Choice 345

Choice 346

Choice 347

Choice 348

Choice 349

Choice 350

Choice 351

Choice 352

Choice 353

Choice 354

Choice 355

Choice 356

Choice 357

Choice 358

Choice 359

Choice 360

Choice 361

Choice 362

Choice 363

Choice 364

Choice 365

Choice 366

Choice 367

Choice 368

Choice 369

Choice 370

Choice 371

Choice 372

Choice 373

Choice 374

Choice 375

Choice 376

Choice 377

Choice 378

Choice 379

Choice 380

Choice 381

Choice 382

Choice 383

Choice 384

Choice 385

Choice 386

Choice 387

Choice 388

Choice 389

Choice 390

Choice 391

Choice 392

Choice 393

Choice 394

Choice 395

Choice 396

Choice 397

Choice 398

Choice 399

Choice 400

Choice 401

Choice 402

Choice 403

Choice 404

Choice 405

Choice 406

Choice 407

Choice 408

Choice 409

Choice 410

Choice 411

Choice 412

Choice 413

Choice 414

Choice 415

Choice 416

Choice 417

Choice 418

Choice 419

Choice 420

Choice 421

Choice 422

Choice 423

Choice 424

Choice 425

Choice 426

Choice 427

Choice 428

Choice 429

Choice 430

Choice 431

Choice 432

Choice 433

Choice 434

Choice 435

Choice 436

Choice 437

Choice 438

Choice 439

Choice 440

Choice 441

Choice 442

Choice 443

Choice 444

Choice 445

Choice 446

Choice 447

Choice 448

Choice 449

Choice 450

Choice 451

Choice 452

Choice 453

Choice 454

Choice 455

Choice 456

Choice 457

Choice 458

Choice 459

Choice 460

Choice 461

Choice 462

Choice 463

Choice 464

Choice 465

Choice 466

Choice 467

Choice 468

Choice 469

Choice 470

Choice 471

Choice 472

Choice 473

Choice 474

Choice 475

Choice 476

Choice 477

Choice 478

Choice 479

Choice 480

Choice 481

Choice 482

Choice 483

Choice 484

Choice 485

Choice 486

Choice 487

Choice 488

Choice 489

Choice 490

Choice 491

Choice 492

Choice 493

Choice 494

Choice 495

Choice 496

Choice 497

Choice 498

Choice 499

Choice 500

Choice 501

Choice 502

Choice 503

Choice 504

Choice 505

Choice 506

Choice 507

Choice 508

Choice 509

Choice 510

Choice 511

Choice 512

Choice 513

Choice 514

Choice 515

Choice 516

Choice 517

Choice 518

Choice 519

Choice 520

Choice 521

Choice 522

Choice 523

Choice 524

Choice 525

Choice 526

Choice 527

Choice 528

Choice 529

Choice 530

Choice 531

Choice 532

Choice 533

Choice 534

Choice 535

Choice 536

Choice 537

Choice 538

Choice 539

Choice 540

Choice 541

Choice 542

Choice 543

Choice 544

Choice 545

Choice 546

Choice 547

Choice 548

Choice 549

Choice 550

Choice 551

Choice 552

Choice 553

Choice 554

Choice 555

Choice 556

Choice 557

Choice 558

Choice 559

Choice 560

Choice 561

Choice 562

Choice 563

Choice 564

Choice 565

Choice 566

Choice 567

Choice 568

Choice 569

Choice 570

Choice 571

Choice 572

Choice 573

Choice 574

Choice 575

Choice 576

Choice 577

Choice 578

Choice 579

Choice 580

Choice 581

Choice 582

Choice 583

Choice 584

Choice 585

Choice 586

Choice 587

Choice 588

Choice 589

Choice 590

Choice 591

Choice 592

Choice 593

Choice 594

Choice 595

Choice 596

Choice 597

Choice 598

Choice 599

Choice 600

Choice 601

Choice 602

Choice 603

Choice 604

Choice 605

Choice 606

Choice 607

Choice 608

Choice 609

Choice 610

Choice 611

Choice 612

Choice 613

Choice 614

Choice 615

Choice 616

Choice 617

Choice 618

Choice 619

Choice 620

Choice 621

Choice 622

Choice 623

Choice 624

Choice 625

Choice 626

Choice 627

Choice 628

Choice 629

Choice 630

Choice 631

Choice 632

Choice 633

Choice 634

Choice 635

Choice 636

Choice 637

Choice 638

Choice 639

Choice 640

Choice 641

Choice 642

Choice 643

Choice 644

Choice 645

Choice 646

Choice 647

Choice 648

Choice 649

Choice 650

Choice 651

Choice 652

Choice 653

Choice 654

Choice 655

Choice 656

Choice 657

Choice 658

Choice 659

Choice 660

Choice 661

Choice 662

Choice 663

Choice 664

Choice 665

Choice 666

Choice 667

Choice 668

Choice 669

Choice 670

Choice 671

Choice 672

Choice 673

Choice 674

Choice 675

Choice 676

Choice 677

Choice 678

Choice 679

Choice 680

Choice 681

Choice 682

Choice 683

Choice 684

Choice 685

Choice 686

Choice 687

Choice 688

Choice 689

Choice 690

Choice 691

Choice 692

Choice 693

Choice 694

Choice 695

Choice 696

Choice 697

Choice 698

Choice 699

Choice 700

Choice 701

Choice 702

Choice 703

Choice 704

Choice 705

Choice 706

Choice 707

Choice 708

Choice 709

Choice 710

Choice 711

Choice 712

Choice 713

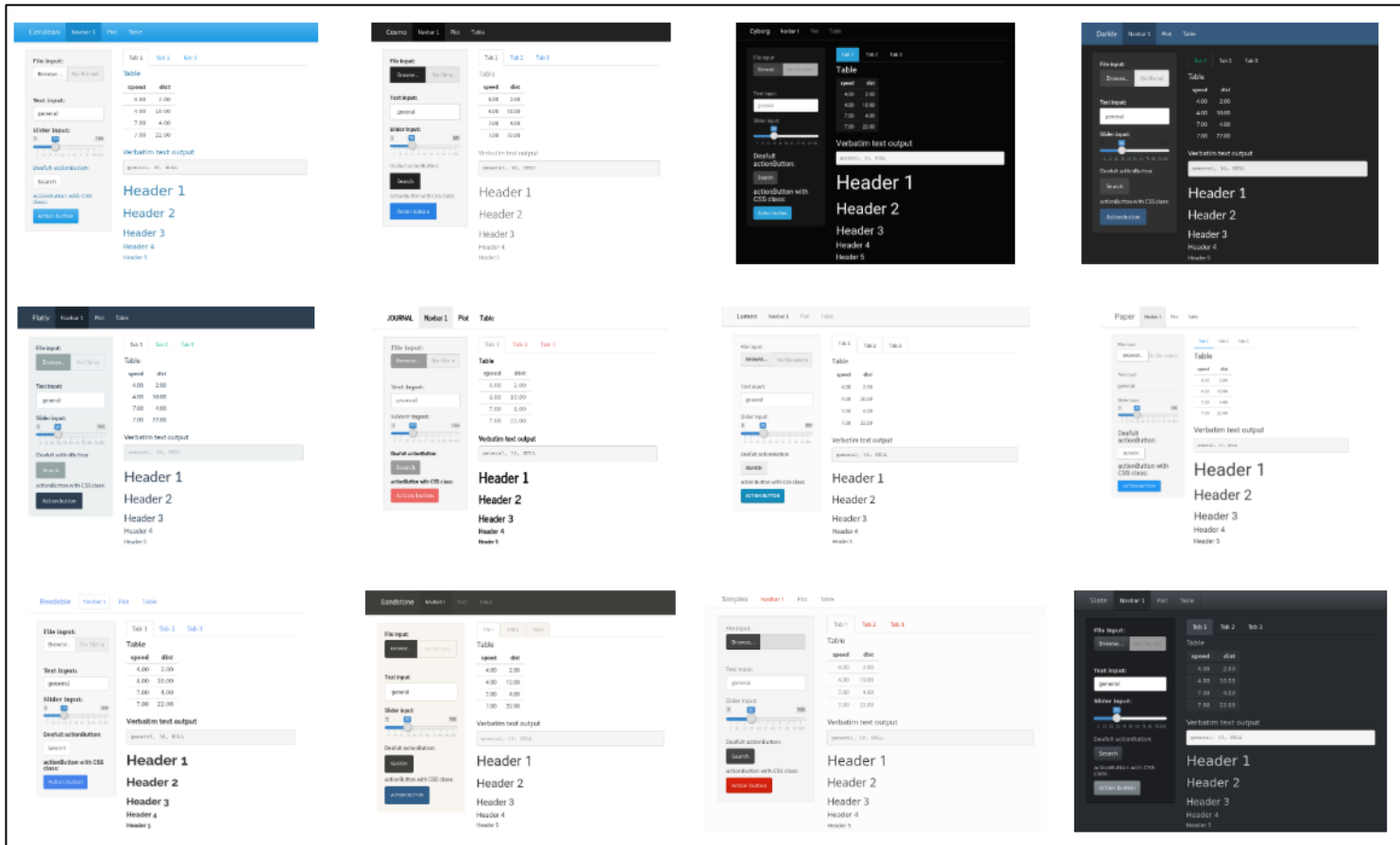
Choice 714

Choice 715

Choice 716

Choice

Other resources – Shiny Themes



 <https://rstudio.github.io/shinythemes/>

Thank you and happy coding!



Time for a short quiz ... get your phones ready!

<https://ahaslides.com/SHINY>