

# shinyjs

## Shiny Developer Conference

Dean Attali

January 31, 2016

<http://deanattali.com>

<https://github.com/daattali/shinyjs>

# Links

<a href="https://bit.ly/shinyjs">bit.ly/shinyjs</a>	shinyjs on GitHub
<a href="https://bit.ly/shinyjs-slides">bit.ly/shinyjs-slides</a>	Google Slides for this presentation
<a href="https://bit.ly/shinyjs-code">bit.ly/shinyjs-code</a>	All code chunks used in this presentation

# Dean Attali

- MSc student in Bioinformatics with Jenny Bryan
  - University of British Columbia
- Bachelor of Computer Science
  - University of Waterloo, Ontario
- Past life: Web developer in Silicon Valley
  - wish.com, tagged.com, makeupbee.com, glitr.com
- Even earlier: software engineer @ IBM, Google

# shinyjs

- Extension to Shiny
- Functions to enhance user interaction/user experience
- js = **J**ava**S**cript
- Absolutely 0 JavaScript knowledge required

# Happy 1st birthday shinyjs!



**initial (embarrassing) commit**  
daattali committed on Jan 25, 2015

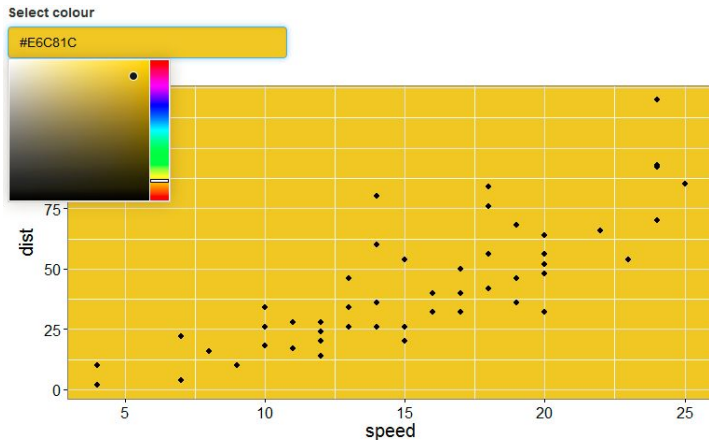


72ddc15



# colourInput(id, label, value, ...)

- Not very shinyjs-y
- Input to select a colour, works like any native Shiny input
  - `updateColourInput()`



# Using *shinyjs*

- Call ***useShinyjs()*** anywhere in the UI

```
library(shiny)
library(shinyjs)

ui <- fluidPage(
  useShinyjs(),
  actionButton("btn", "Hide"),
  textInput("text", "Text")
)

server <- function(input, output) {
  observeEvent(input$btn, {
    hide("text")
  })
}

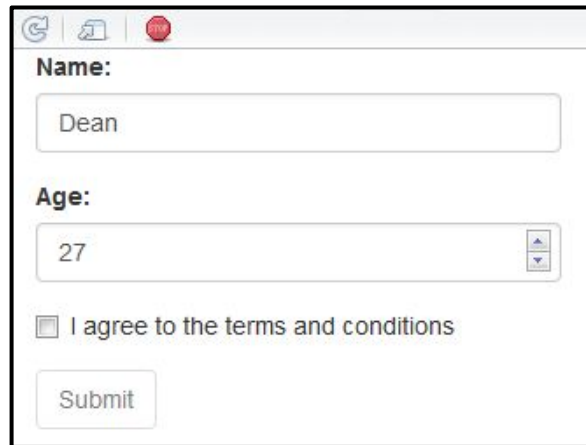
shinyApp(ui = ui, server = server)
```

# Common *shinyjs* functions

- *hide()* , *show()* , *toggle()*
  - *hidden()*
  - *disable()* , *enable()* , *toggleState()*
  - *disabled()*
  - *reset()*
- 
- Read documentation for full details on parameters



# Sample app using *shinyjs*



Name:  
Dean

Age:  
27

☐ I agree to the terms and conditions

Submit

- Only enable submit button when terms are accepted
- Reset form after submitting
- Show “Thank you” message after submitting

# *shinyjs* in interactive Rmd

- Works just the same, need to supply *rmd = TRUE* argument

```
```{r, echo=FALSE}
library(shinyjs)

useShinyjs(rmd = TRUE)
actionButton("button", "Click me")
div(id = "hello", "Hello!")

observeEvent(input$button, {
  toggle("hello")
})
```
```

# *shinyjs* in dashboards

## Common mistake:

```
ui <- dashboardPage(  
  useShinyjs(), ...  
)
```

## Place *useShinyjs()* inside *dashboardBody()*

```
ui <- dashboardPage(  
  dashboardHeader(),  
  dashboardSidebar(),  
  dashboardBody(  
    useShinyjs()  
  )  
)
```

# *shinyjs* in navbarLayout



Common mistake:

```
ui <- navbarPage(  
  useShinyjs(), "My title", ...  
)
```



Use a *tagList*

```
ui <- tagList(  
  useShinyjs(),  
  navbarPage(  
    "My title", ...  
  )  
)
```

# *extendShinyjs()*

## Calling your own JS functions

**somefile.js**

```
shinyjs.foo = function(params) {  
  alert(params);  
}
```

**ui.R**

```
fluidPage(  
  useShinyjs(),  
  extendShinyjs("somefile.js")  
)
```

**server.R**

```
function(input, output) {  
  js$foo("hello!")  
}
```

# *extendShinyjs()* goodies

- Use *shinyjs.getParams* to
  - Call your function with named/unnamed arguments
  - Use default values
- Want to run code on page load (eg. register event handlers)? Define *shinyjs.init* function

# *extendShinyjs()* requires V8 package

- V8 does not easily install on some OS + *extendShinyjs()* is advanced usage = V8 is not “required” but “suggested”
- Common issue: if deploying to shinyapps.io, add *library(V8)* somewhere in code
- If you cannot install V8, there is workaround - read docs

# FAQ solved with *extendShinyjs()*

- How do I show/hide the *shinydashboard* sidebar programmatically?
  - <http://stackoverflow.com/questions/31305930>
- How do I hide/disable a tab?
  - <http://stackoverflow.com/questions/31703241>
- How do I refresh the page?
  - <http://stackoverflow.com/questions/34142841>



# FAQ solved with *extendShinyjs()*

- How do I call a JavaScript function from a different JavaScript library?
  - Example can be found here <https://github.com/timelyportfolio/sweetalertR/issues/1#issuecomment-151685005>
- How do I change the values of a sliderInput?
  - <http://stackoverflow.com/questions/30502870>
- How do I call JavaScript code and use the return value?
  - <http://stackoverflow.com/questions/34717489>

# FAQ with no answers (yet)

- How do I reset the value of an *actionButton*?
- How do I reset a *fileInput*?
- How do I hide/disable *this specific* radio button/dropdown option?

Answer: exercise left for the reader

# Inside *shinyjs*:

## What does *useShinyjs()* do?

- Add required JavaScript functions to Shiny app
- Add required CSS for hiding elements
- Create all shinyjs message handlers
- Initialize variables used internally
- See for yourself: view the source of this app:

```
shinyApp(fluidPage(useShinyjs()), function(input, output) {})
```

# Inside *shinyjs*: What does *useShinyjs()* do?

view-source:127.0.0.1:6777

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5   <script type="application/shiny-singletons">d30a50537d83adac0f5262c0ca32d64199cc094</script>
6   <script type="application/html-dependencies">json2[2014.02.04];jquery[1.11.3];shiny[0.12.2.9006];bootstrap[3.3.5]</script>
7   <script src="shared/json2-min.js"></script>
8   <script src="shared/jquery.min.js"></script>
9   <link href="shared/shiny.css" rel="stylesheet" />
10  <script src="shared/shiny.min.js"></script>
11  <meta name="viewport" content="width=device-width, initial-scale=1" />
12  <link href="shared/bootstrap/css/bootstrap.min.css" rel="stylesheet" />
13  <script src="shared/bootstrap/js/bootstrap.min.js"></script>
14  <script src="shared/bootstrap/shim/html5shiv.min.js"></script>
15  <script src="shared/bootstrap/shim/respond.min.js"></script>
16  <style>.shinyjs-hide { display: none; }</style>
17  <script>Shiny.addCustomMessageHandler('show', function(params) { shinyjs.debugMessage('shinyjs: calling function "show" with parameters:'); shinyjs.debugMessage(params); shinyjs.show(params);});
18  Shiny.addCustomMessageHandler('hide', function(params) { shinyjs.debugMessage('shinyjs: calling function "hide" with parameters:'); shinyjs.debugMessage(params); shinyjs.hide(params);});
19  Shiny.addCustomMessageHandler('toggle', function(params) { shinyjs.debugMessage('shinyjs: calling function "toggle" with parameters:'); shinyjs.debugMessage(params); shinyjs.toggle(params);});
20  Shiny.addCustomMessageHandler('enable', function(params) { shinyjs.debugMessage('shinyjs: calling function "enable" with parameters:'); shinyjs.debugMessage(params); shinyjs.enable(params);});
21  Shiny.addCustomMessageHandler('disable', function(params) { shinyjs.debugMessage('shinyjs: calling function "disable" with parameters:'); shinyjs.debugMessage(params); shinyjs.disable(params);});
22  Shiny.addCustomMessageHandler('toggleState', function(params) { shinyjs.debugMessage('shinyjs: calling function "toggleState" with parameters:'); shinyjs.debugMessage(params); shinyjs.toggleState(params);});
23  Shiny.addCustomMessageHandler('addClass', function(params) { shinyjs.debugMessage('shinyjs: calling function "addClass" with parameters:'); shinyjs.debugMessage(params); shinyjs.addClass(params);});
24  Shiny.addCustomMessageHandler('removeClass', function(params) { shinyjs.debugMessage('shinyjs: calling function "removeClass" with parameters:'); shinyjs.debugMessage(params); shinyjs.removeClass(params);});
25  Shiny.addCustomMessageHandler('toggleClass', function(params) { shinyjs.debugMessage('shinyjs: calling function "toggleClass" with parameters:'); shinyjs.debugMessage(params); shinyjs.toggleClass(params);});
26  Shiny.addCustomMessageHandler('text', function(params) { shinyjs.debugMessage('shinyjs: calling function "text" with parameters:'); shinyjs.debugMessage(params); shinyjs.text(params);});
27  Shiny.addCustomMessageHandler('onevent', function(params) { shinyjs.debugMessage('shinyjs: calling function "onevent" with parameters:'); shinyjs.debugMessage(params); shinyjs.onevent(params);});
28  Shiny.addCustomMessageHandler('info', function(params) { shinyjs.debugMessage('shinyjs: calling function "info" with parameters:'); shinyjs.debugMessage(params); shinyjs.info(params);});
29  Shiny.addCustomMessageHandler('logjs', function(params) { shinyjs.debugMessage('shinyjs: calling function "logjs" with parameters:'); shinyjs.debugMessage(params); shinyjs.logjs(params);});
30  Shiny.addCustomMessageHandler('runjs', function(params) { shinyjs.debugMessage('shinyjs: calling function "runjs" with parameters:'); shinyjs.debugMessage(params); shinyjs.runjs(params);});
31  Shiny.addCustomMessageHandler('reset', function(params) { shinyjs.debugMessage('shinyjs: calling function "reset" with parameters:'); shinyjs.debugMessage(params); shinyjs.reset(params);});
32  Shiny.addCustomMessageHandler('delay', function(params) { shinyjs.debugMessage('shinyjs: calling function "delay" with parameters:'); shinyjs.debugMessage(params); shinyjs.delay(params);});</script>
33  <script src="shinyjs/shinyjs-default-funcs.js"></script>
34  <script>shinyjs.debug = false;</script>
35 </head>
36 <body>
37   <div class="container-fluid"></div>
38 </body>
39 </html>
```