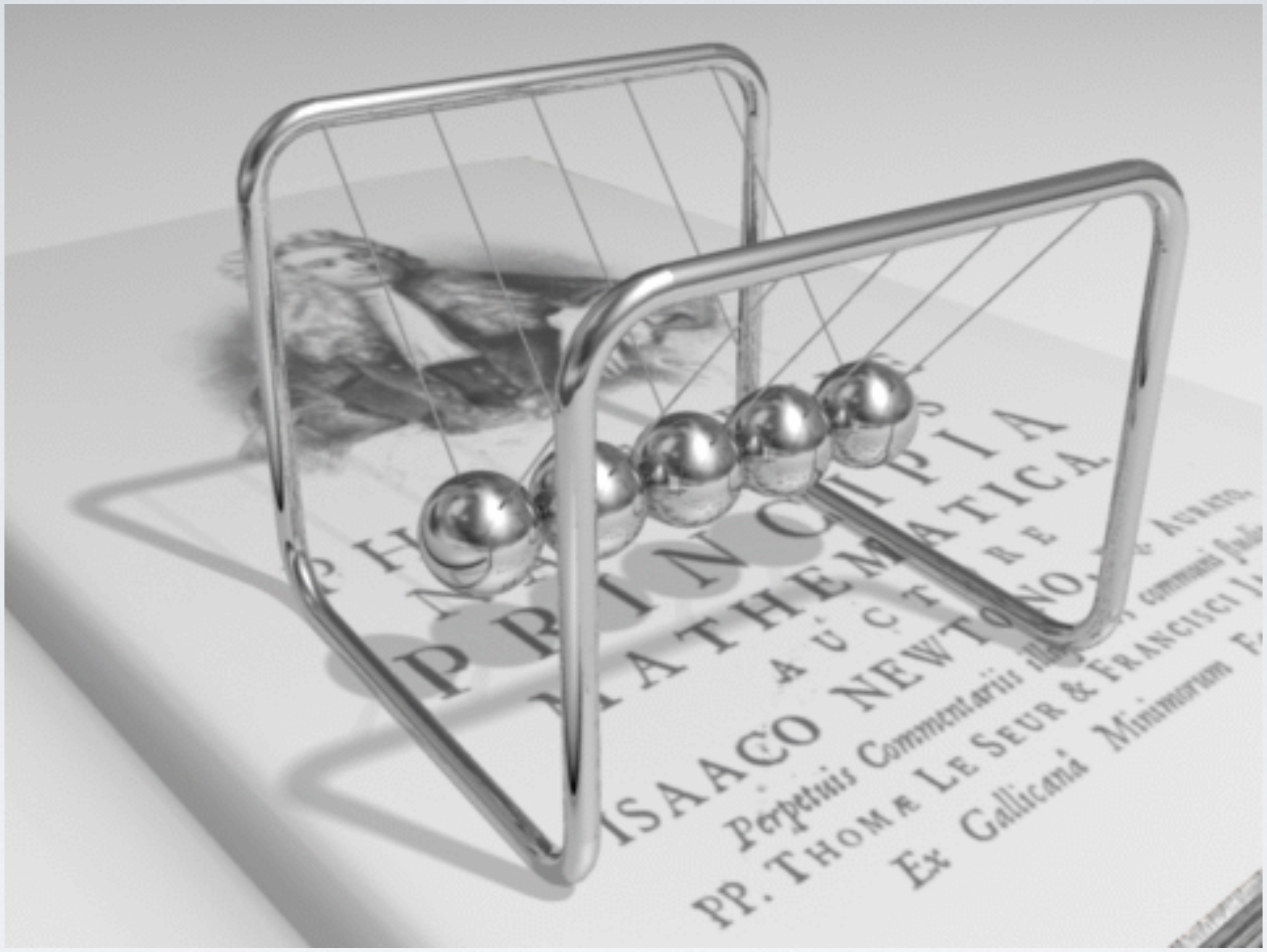


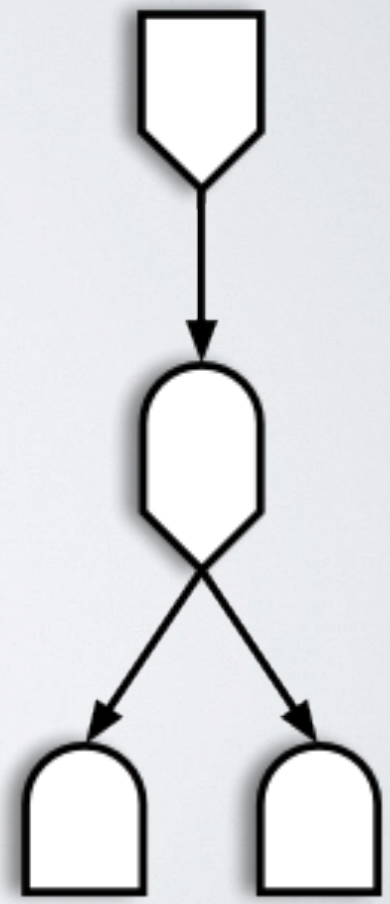
REACTIVITY

SISBID 2020

Heike Hofmann & Di Cook

ELEMENTS OF REACTIVITY

- Sources
  - Any input widget is a source
- Conductors
  - Use input and are being used further along
- Observers
  - Any output is an observer



TWO CONDUCTORS

- Reactive expressions and reactive events are two types of conductors
- Reactive expressions are the archetypical conductor: envelope functionality used in multiple places of an app, run evaluations only once and store current values.
- Reactive events are only triggered by specific events (such as a click on an action button)

REACTIVE EXPRESSIONS

```
rval <- reactive({
…
})
```

Called like a function as:

```
rval()
```

- reactive expressions are executed **lazily**, and their values are **cached**

- **Lazy:** evaluated only on demand, typically requested by a reactive endpoint.

- **Cached:** (re-)evaluated only when the value of a dependency changed.

REACTIVE EVENTS

```
rval <- eventReactive(actionbutton,{
…
})
```

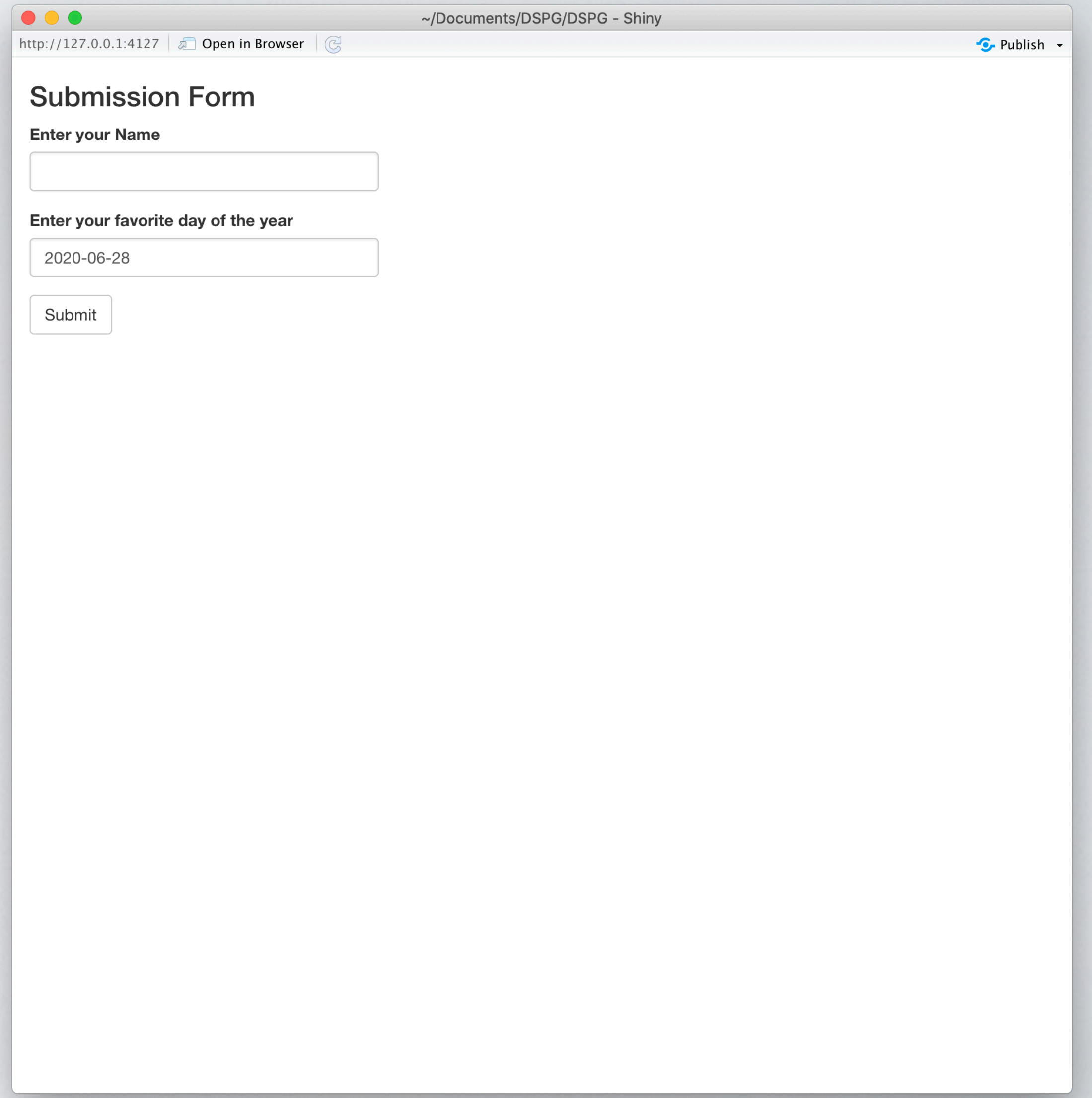
Called like a function as:

```
rval()
```

- reactive events are executed even more **lazily**: only on demand, typically requested by an action button

EXAMPLE: SUBMISSION FORM

- In RStudio, open file app.R in 03_submission
- Run the app (a couple of times)
- Turn on showcase mode runApp("03_submission/", display.mode = "showcase")



Submission Form

Enter your Name

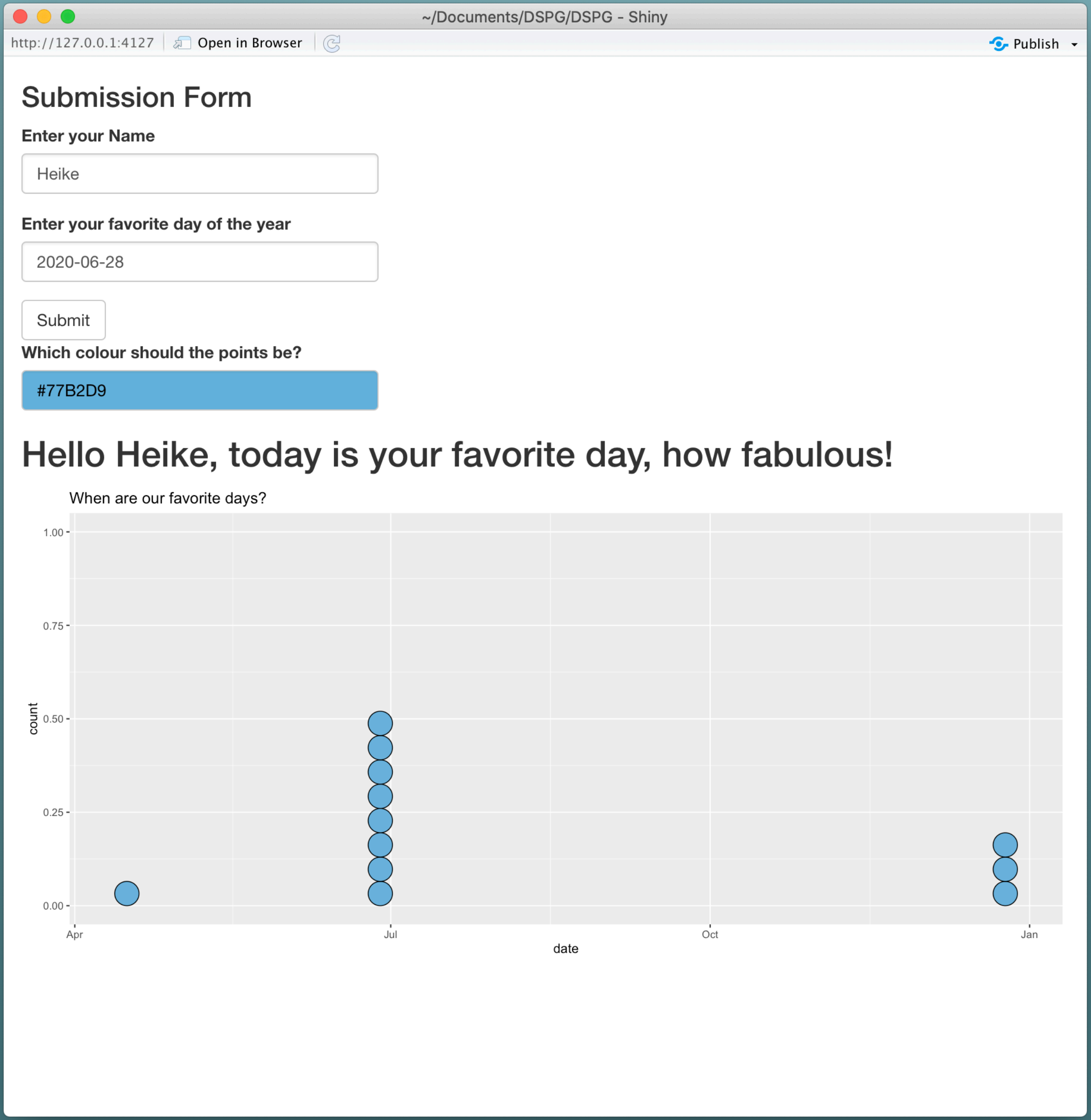
Enter your favorite day of the year

2020-06-28

Submit

YOUR TURN

- Open the file 03_submission.R
- The package colourpicker implements a color wheel as input widget
- Allow users to change the color of the dots in the dot plot
- What other interactive elements can you think of adding?
- The answer is in 03b_submission.R



Submission Form

Enter your Name

Heike

Enter your favorite day of the year

2020-06-28

Submit

Which colour should the points be?

#77B2D9

Hello Heike, today is your favorite day, how fabulous!

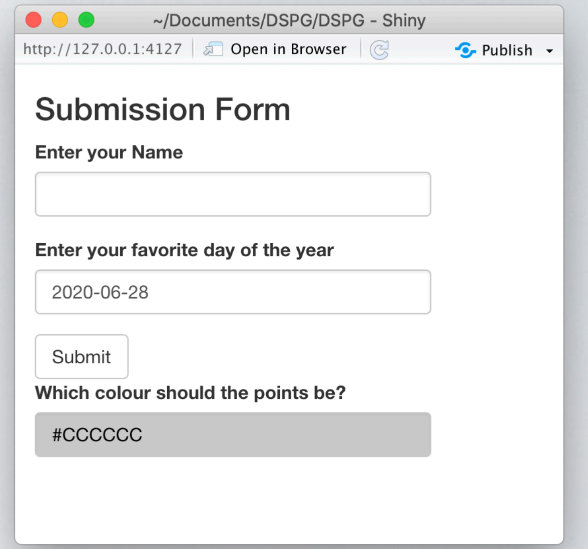
CONDITIONAL PANELS

- Showing a color picker before needed … might be confusing
  users of the app

- `conditionalPanel(condition, …)`
  allows us to encapsulate elements of the user interface and only show when 'condition' is fulfilled

- Here, a condition of
  `condition = "input.submit > 0"`
  is true when the submit button was pressed at least once

- This is implemented in 03c_submission.R



Submission Form

Enter your Name

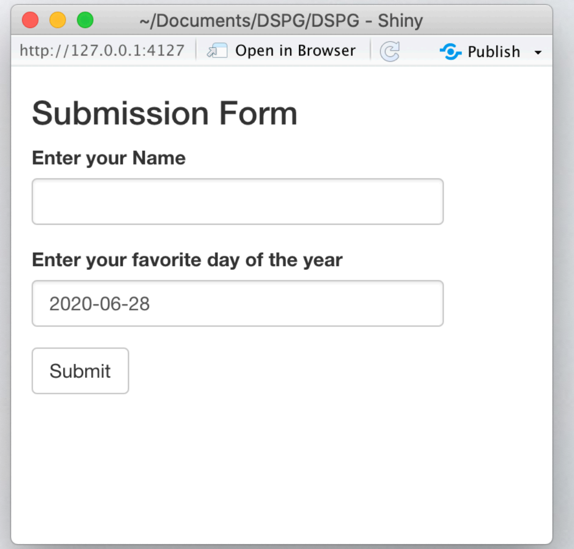
Enter your favorite day of the year

2020-06-28

Submit

Which colour should the points be?

#CCCCCC



Submission Form

Enter your Name

Enter your favorite day of the year

2020-06-28

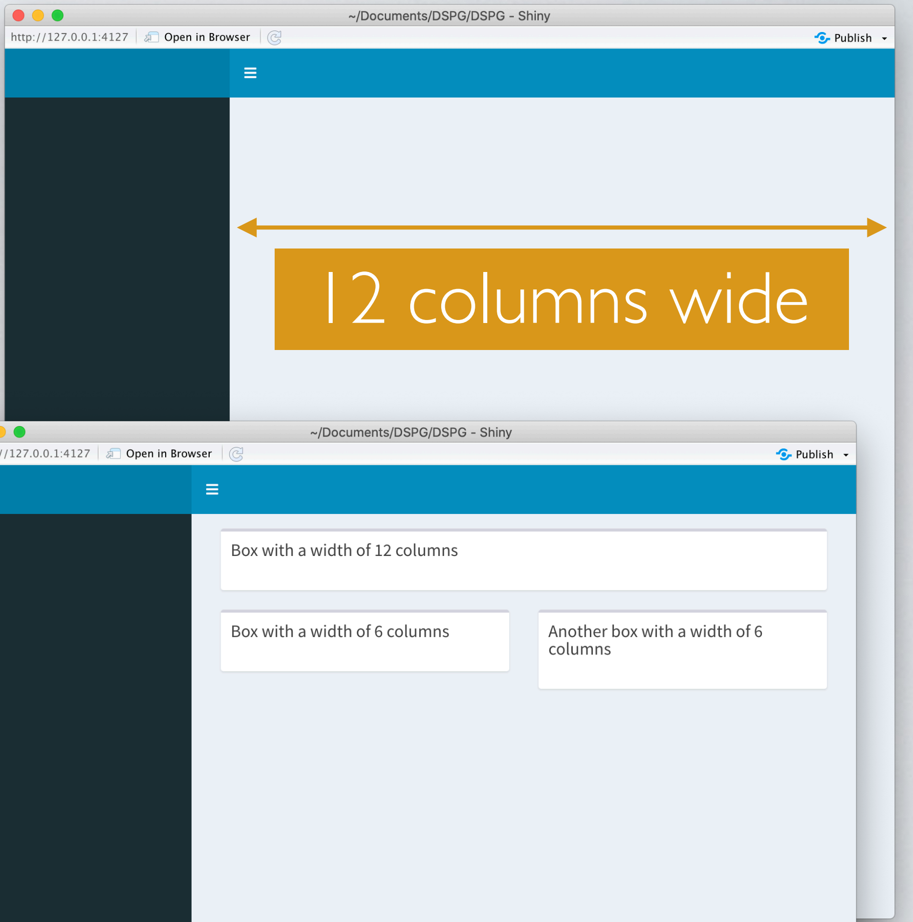
Submit

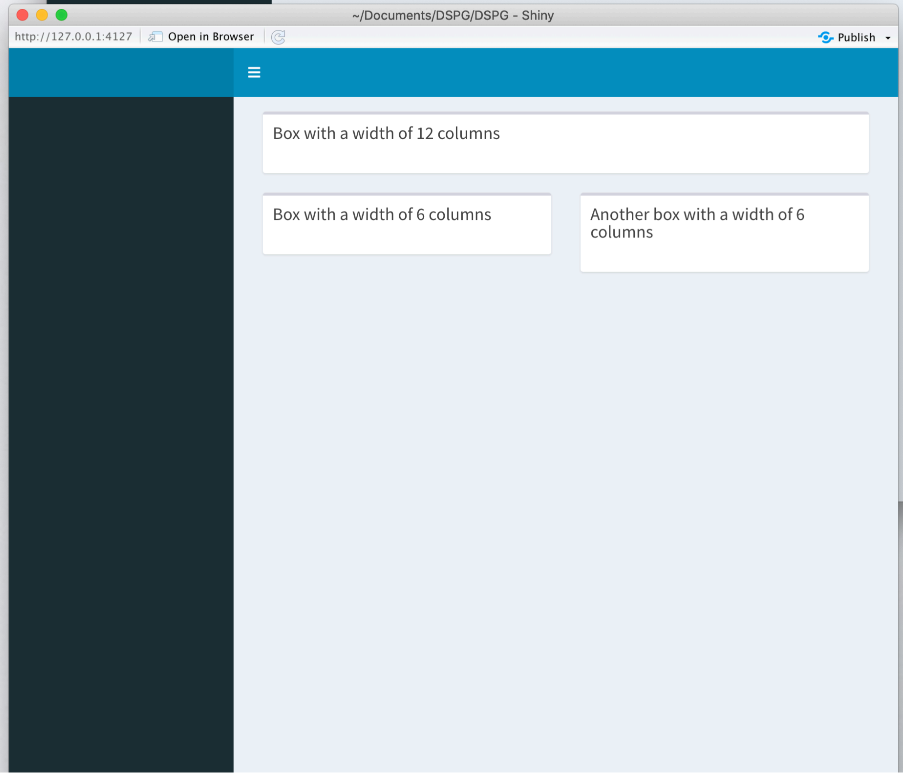
LAYOUT OF DASHBOARDS

- The body can be laid out in a grid - either row based or column based

- Structure is introduced by boxes:

```
box(..., title = NULL,
width = 6, height = NULL)
```





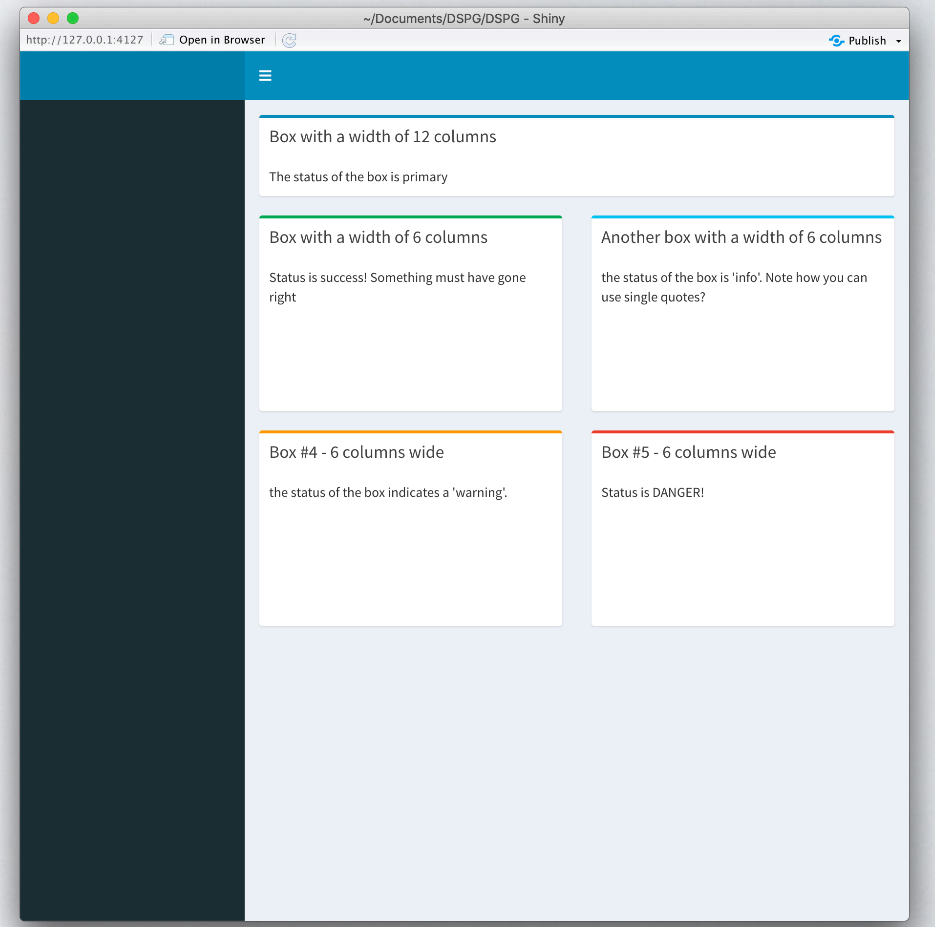
Box with a width of 12 columns

Box with a width of 6 columns

Another box with a width of 6 columns

BOXES

- Boxes help with structuring output
- Boxes also have a **`status`** parameter
- Status is shown as a colored bar along the top of a box
- **`?validStatuses`** are **`primary, success, info, warning, danger`**



Box with a width of 12 columns

The status of the box is primary

Box with a width of 6 columns

Status is success! Something must have gone right

Another box with a width of 6 columns

the status of the box is 'info'. Note how you can use single quotes?

Box #4 - 6 columns wide

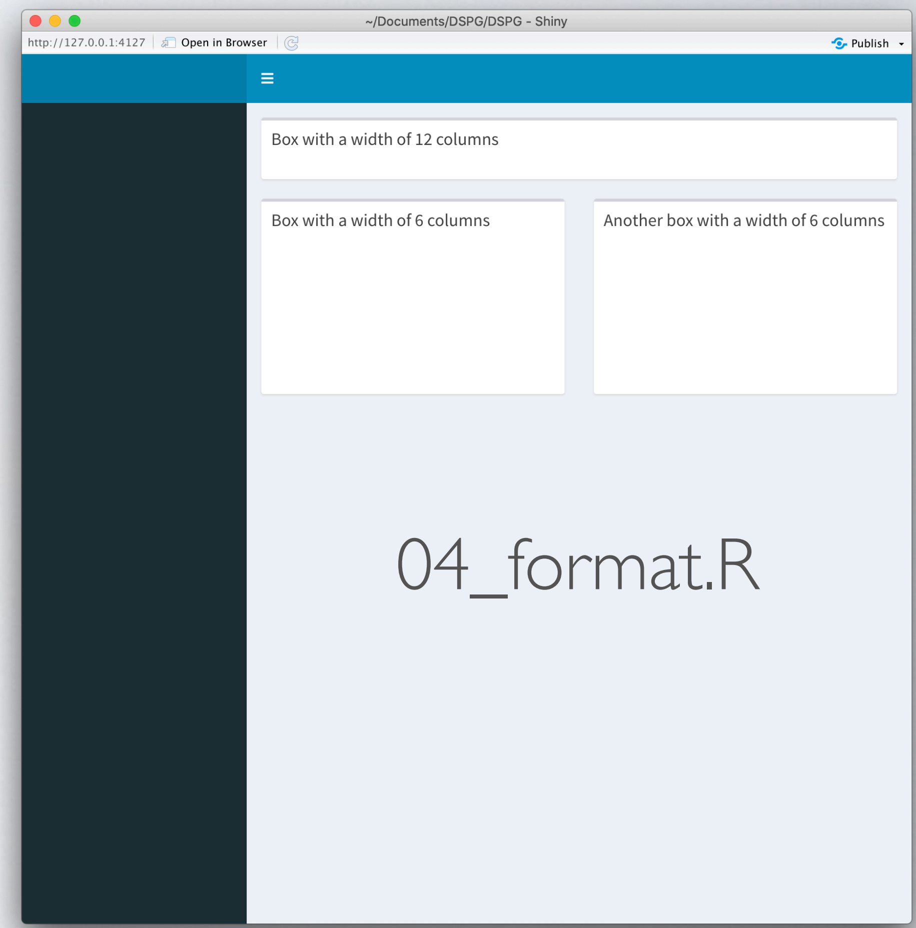
the status of the box indicates a 'warning'.

Box #5 - 6 columns wide

Status is DANGER!

ROW BASED LAYOUT

- Body is wrapped in a fluidRow function
- Tops of boxes are aligned
- Bottom of the boxes can be aligned by setting the height (in pixel)



Box with a width of 12 columns

Box with a width of 6 columns

Another box with a width of 6 columns

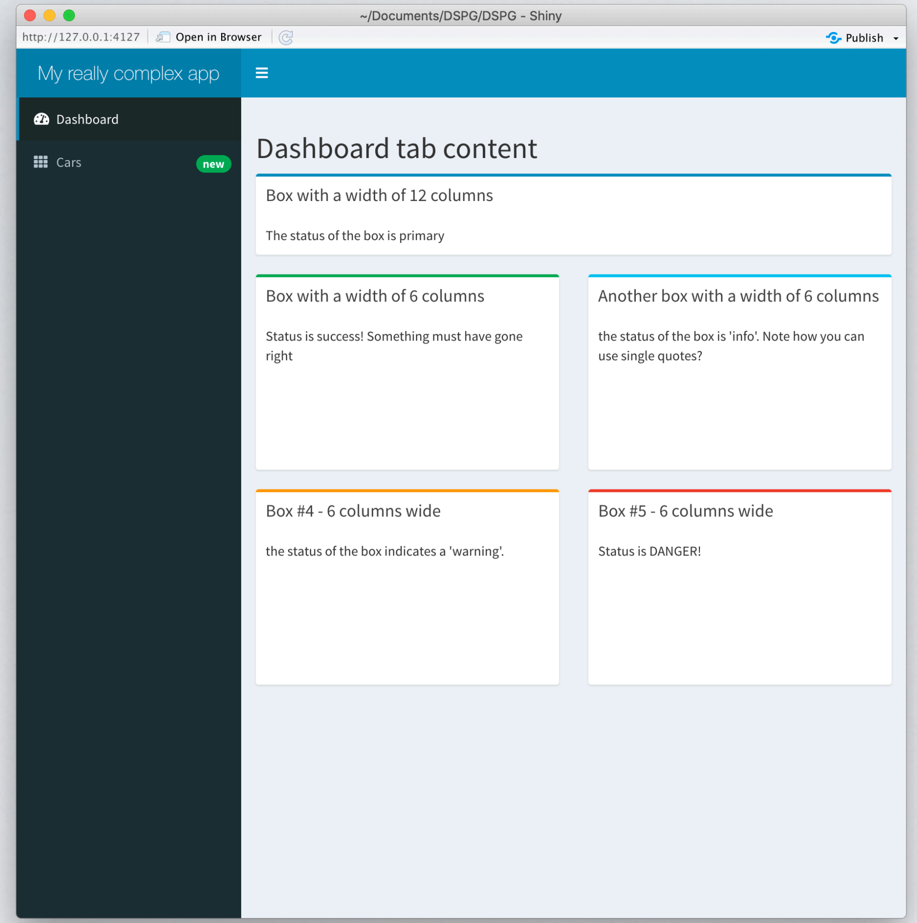
04_format.R

```
body <- dashboardBody(
  fluidRow(
  box(title = "Box with a width of 12 columns", width = 12),
  box(title = "Box with a width of 6 columns", width = 6, height = 200),
  box(title = "Another box with a width of 6 columns", width = 6, height = 200)
  )
)
```

OTHER LAYOUTS

- In **column** based layouts, the body is wrapped in a fluidRow function
  - Height of boxes are aligned,
    each column has to define a width, boxes are aligned in width.
- In mixed layouts fluidRow and column can be used sequentially

TABS IN DASHBOARDS



Dashboard tab content

Box with a width of 12 columns

The status of the box is primary

Box with a width of 6 columns

Status is success! Something must have gone right

Another box with a width of 6 columns

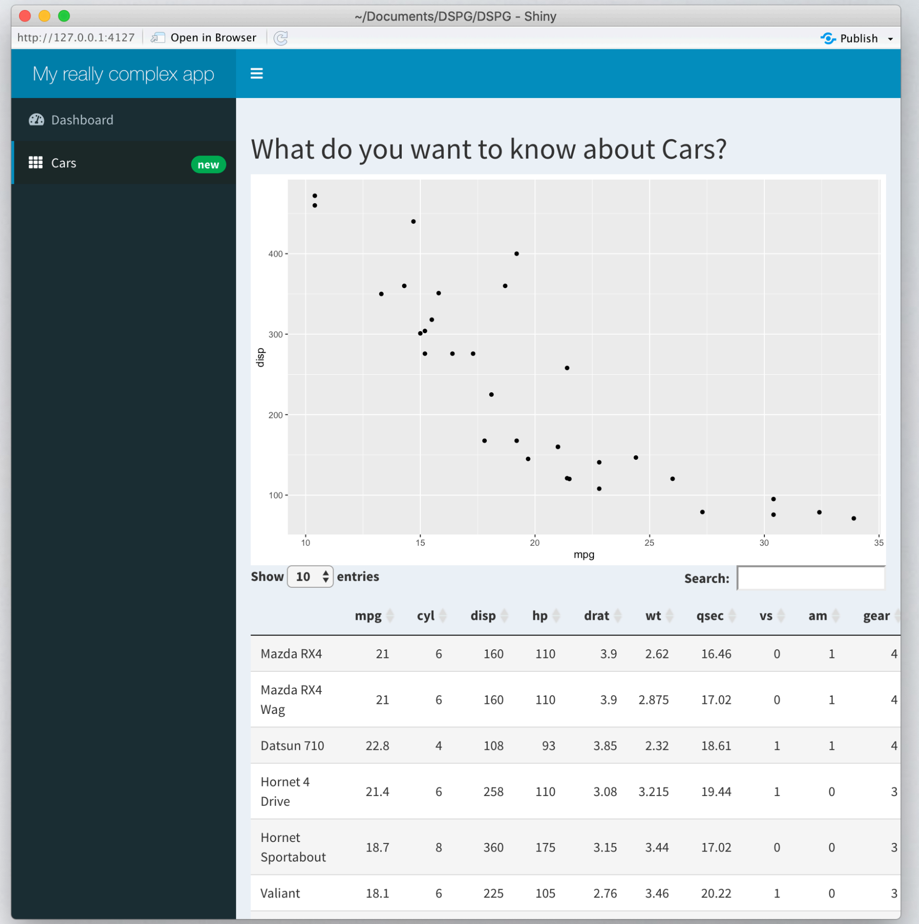
the status of the box is 'info'. Note how you can use single quotes?

Box #4 - 6 columns wide

the status of the box indicates a 'warning'.

Box #5 - 6 columns wide

Status is DANGER!



What do you want to know about Cars?

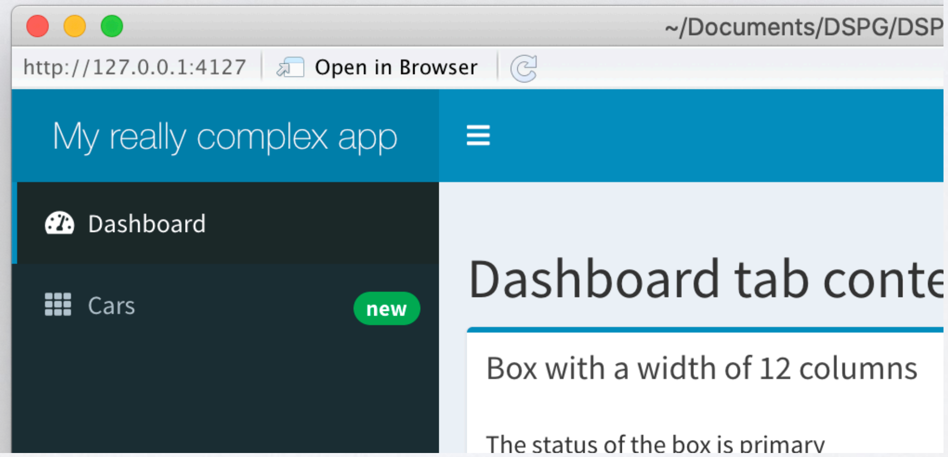
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3

- The sidebar menu can be used to introduce tabs for quick navigation

05_tabsets.R

TABS IN DASHBOARDS

```
sidebar <- dashboardSidebar(
  sidebarMenu(
    menuItem("Dashboard", tabName = "dashboard",
             icon = icon("dashboard")),
    menuItem("Cars", icon = icon("th"), tabName = "cars",
             badgeLabel = "new", badgeColor = "green")
  )
)
```



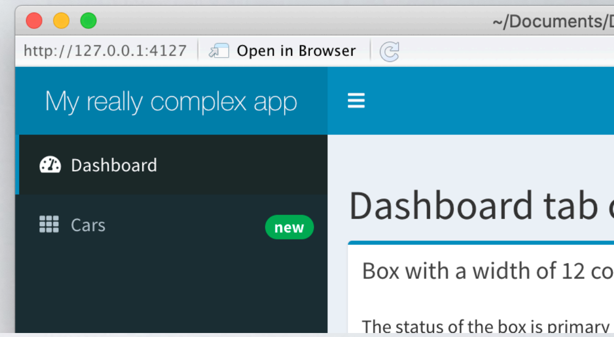
Dashboard tab conte

Box with a width of 12 columns

The status of the box is primary

05_tabsets.R

TABS IN DASHBOARDS


~/Documents/
http://127.0.0.1:4127 Open in Browser

Dashboard tab
Box with a width of 12 co
The status of the box is primary

```
sidebar <- dashboardSidebar(
  sidebarMenu(
    menuItem("Dashboard", tabName = "dashboard",
             icon = icon("dashboard")),
    menuItem("Cars", icon = icon("th"), tabName = "cars",
             badgeLabel = "new", badgeColor = "green")
  )
)
```

```
body <- dashboardBody(
  tabItems(
    tabItem(tabName = "dashboard",
            h2("Dashboard tab content"),
            fluidRow(
              box(title = "Box with a width of 12 columns", width = 12,
                  status = "primary", "The status of the box is primary"),
              …
              box(title = "Box #5 - 6 columns wide",
                  status = "danger", "Status is DANGER!",
                  width = 6, height = 200)
            )
    ),

    tabItem(tabName = "cars",
            h2("What do you want to know about Cars?"),
            plotOutput("myplot"),
            DTOutput("mytable")
    )
```

RESOURCES

- RStudio Tutorial: https://shiny.rstudio.com/articles/reactivity-overview.html
- shiny cheat sheet: https://github.com/rstudio/cheatsheets/raw/master/shiny.pdf
- gallery of shiny apps: https://shiny.rstudio.com/gallery/