



Reactivity

Read the [chapter](#) in Mastering Shiny to understand reactivity in general

Read the surveydown [reactivity page](#) for reactivity in surveydown

Reactivity

1. Reactivity basics
2. Reactive questions

Reactivity

1. **Reactivity basics**

2. Reactive questions

Reactivity is about changing
something based on respondent input

What's your name?

Hello J!

What's your name?

Hello Jo!

What's your name?

Hello Joe!

Same example in surveydown

In `survey.qmd`:

```
```{r}
sd_question(
 type = "text",
 id = "name",
 label = "What is your name?"
)
textOutput("greeting")
```
```

In `app.R`:

```
server <- function(input, output, session) {
  output$greeting <- renderText({
    paste0("Hello ", input$name, "!")
  })
  sd_server()
}
```

Simplified with `sd_output()` to display question values

In `survey.qmd`:

```
```{r}
sd_question(
 type = "text",
 id = "name",
 label = "What is your name?"
)
```
```

```
Hello, `r sd_output(id = "name", type = "value")`!
```

In `app.R`:

```
server <- function(input, output,
  session) {
  sd_server()
}
```


Understanding `sd_output()`

Question in `survey.qmd`:

```
```{r}
sd_question(
 type = "mc",
 id = "penguins",
 label = "What's your favorite penguin?",
 option = c(
 "Adélie" = "adelie",
 "Chinstrap" = "chinstrap",
 "Gentoo" = "gentoo"
)
)
```
```

Display a question *value*:

```
sd_output("penguins", type = "value")
```

Example: `adelie`

Display a question *label*:

```
sd_output("penguins", type =
  "label_option")
```

Example: `Adélie`

Understanding `sd_output()`

Question in `survey.qmd`:

```
```{r}
sd_question(
 type = "mc",
 id = "penguins",
 label = "What's your favorite penguin?",
 option = c(
 "Adélie" = "adelie",
 "Chinstrap" = "chinstrap",
 "Gentoo" = "gentoo"
)
)
```
```

Display a *question label*:

```
sd_output("penguins", type =
  "label_question")
```

Example:

What's your favorite penguin?

Your turn

07:00

In your survey, display a question *value* somewhere with:

```
`r sd_output("id", type = "value")`
```

In your survey, display a question *option label* somewhere with:

```
`r sd_output("id", type = "label_option")`
```

Reactivity

1. Reactivity basics

2. **Reactive questions**

Change the question text based on respondent choices

In `survey.qmd`:

```
```{r}
sd_question(
 id = "pet_type",
 type = "mc",
 label = "Which do you prefer, dogs or cats?",
 option = c(
 "Dogs" = "dog",
 "Cats" = "cat"
)
)
```
```

Follow up question:

If `dog`:

Are you a **dog** owner?

If `cat`:

Are you a **cat** owner?

Solution: Define the entire question in the server

```
library(glue)

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

  observe({
    # Trigger with any change to input$pet_type
    pet_type <- input$pet_type

    # Make the question
    sd_question(
      type    = "mc",
      id      = "pet_owner",
      label   = glue("Are you a {pet_type} owner?"),
      option  = c("Yes" = "yes", "No" = "no")
    )
  })

  # Only show the pet_owner question if pet_type is answered
  sd_show_if(
    sd_is_answered("pet_type") ~ "pet_owner"
  )

  sd_server()
}
```

Solution: Define the entire question in the server

```
library(glue)

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

  observe({
    # Trigger with any change to input$pet_type
    pet_type <- input$pet_type

    # Make the question
    sd_question(
      type = "mc",
      id = "pet_owner",
      label = glue("Are you a {pet_type} owner?"),
      option = c("Yes" = "yes", "No" = "no")
    )
  })

  # Only show the pet_owner question if pet_type is answered
  sd_show_if(
    sd_is_answered("pet_type") ~ "pet_owner"
  )

  sd_server()
}
```

Use **observe** to detect any change to **pet_type** question

Solution: Define the entire question in the server

```
library(glue)

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

  observe({
    # Trigger with any change to input$pet_type
    pet_type <- input$pet_type

    # Make the question
    sd_question(
      type    = "mc",
      id      = "pet_owner",
      label   = glue("Are you a {pet_type} owner?"),
      option  = c("Yes" = "yes", "No" = "no")
    )
  })

  # Only show the pet_owner question if pet_type is answered
  sd_show_if(
    sd_is_answered("pet_type") ~ "pet_owner"
  )

  sd_server()
}
```

Use `sd_question` to define the question

Solution: Define the entire question in the server

```
library(glue)

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

  observe({
    # Trigger with any change to input$pet_type
    pet_type <- input$pet_type

    # Make the question
    sd_question(
      type = "mc",
      id = "pet_owner",
      label = glue("Are you a {pet_type} owner?"),
      option = c("Yes" = "yes", "No" = "no")
    )
  })

  # Only show the pet_owner question if pet_type is answered
  sd_show_if(
    sd_is_answered("pet_type") ~ "pet_owner"
  )

  sd_server()
}
```

Hide the `pet_owner` question until the `pet_type` is answered

Use `sd_output()` to display the reactive question

In `survey.qmd`:

```
```{r}
sd_question(
 id = "pet_type",
 type = "mc",
 label = "Which do you prefer, dogs or cats?",
 option = c(
 "Dogs" = "dog",
 "Cats" = "cat"
)
)
sd_output("pet_owner", type = "question")
```
```

Follow up question:

If `dog`:

Are you a **dog** owner?

If `cat`:

Are you a **cat** owner?

Your turn

10:00

Define a question inside your server (**app.R** file):

```
server <- function(input, output, session) {  
  observe({  
    sd_question(  
      type = "mc",  
      id   = "id",  
      ...  
    )  
  })  
  sd_server()  
}
```

Display the question inside your survey (**survey.qmd** file):

```
sd_output("id", type = "question")
```