1. A new family of reactive objects

What was wrong with Shiny’s existing reactive objects?

```r
downloads <- reactive({
  cranlogs::cran_downloads(input$package,
    from = Sys.Date() - 365, to = Sys.Date())
})
```

- Call `downloads()` to retrieve the current dataset

- Automatically caches the result until `input$package` changes

- Works well for regular Shiny apps, BUT there’s no easy way for us to get the code out

1. A new family of reactive objects

What was wrong with Shiny’s existing reactive objects?

```
downloads <- reactive({
  cranlogs::cran_downloads(input$package,
    from = Sys.Date() - 365, to = Sys.Date())
})
```

- Call `downloads()` to retrieve the current dataset
- Automatically caches the result until `input$package` changes
- Works well for regular Shiny apps, BUT there’s no easy way for us to get the code out

1. A new family of reactive objects

With shinymeta:

```
downloads <- metaReactive({
  cranlogs::cran_downloads(input$package,
    from = Sys.Date() - 365, to = Sys.Date())
})
```

- A `metaReactive` does everything a regular `reactive` does, plus, can give you its own source code at runtime