Using shinymeta

1. You (the app author) **identify the domain logic in your app code** so we can separate it from the reactive structure
2. Within that domain logic, you **identify references to reactive values and reactive expressions** that need to be replaced with static values and static code, respectively
3. At runtime, **choose which pieces** of domain logic to export, and in what order
4. **Present the code** to the user (in a window, as a downloadable script or report, etc.)

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