

Step 3: Generate code with `expandChain()`

```
> expandChain(output$plot)
```

```
downloads <-  
  cranlogs::cran_downloads(  
    "shiny",  
    from = "2019-11-11",  
    to = Sys.Date()  
  )
```

```
downloads_rolling <-  
  downloads %>%  
    mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))
```

```
ggplot(downloads_rolling, aes(date, count)) + geom_line()
```

This allows dynamic results
to be 'hard coded'



Step 3: Generate code with `expandChain()`

```
> expandChain(quote(library(tidyverse)), output$plot)
```

```
library(tidyverse)
```

```
downloads <-  
  cranlogs::cran_downloads(  
    "shiny",  
    from = "2019-11-11",  
    to = Sys.Date()  
  )
```

```
downloads_rolling <-  
  downloads %>%  
    mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))
```

```
ggplot(downloads_rolling, aes(date, count)) + geom_line()
```



Add quoted code to supply
'setup code'