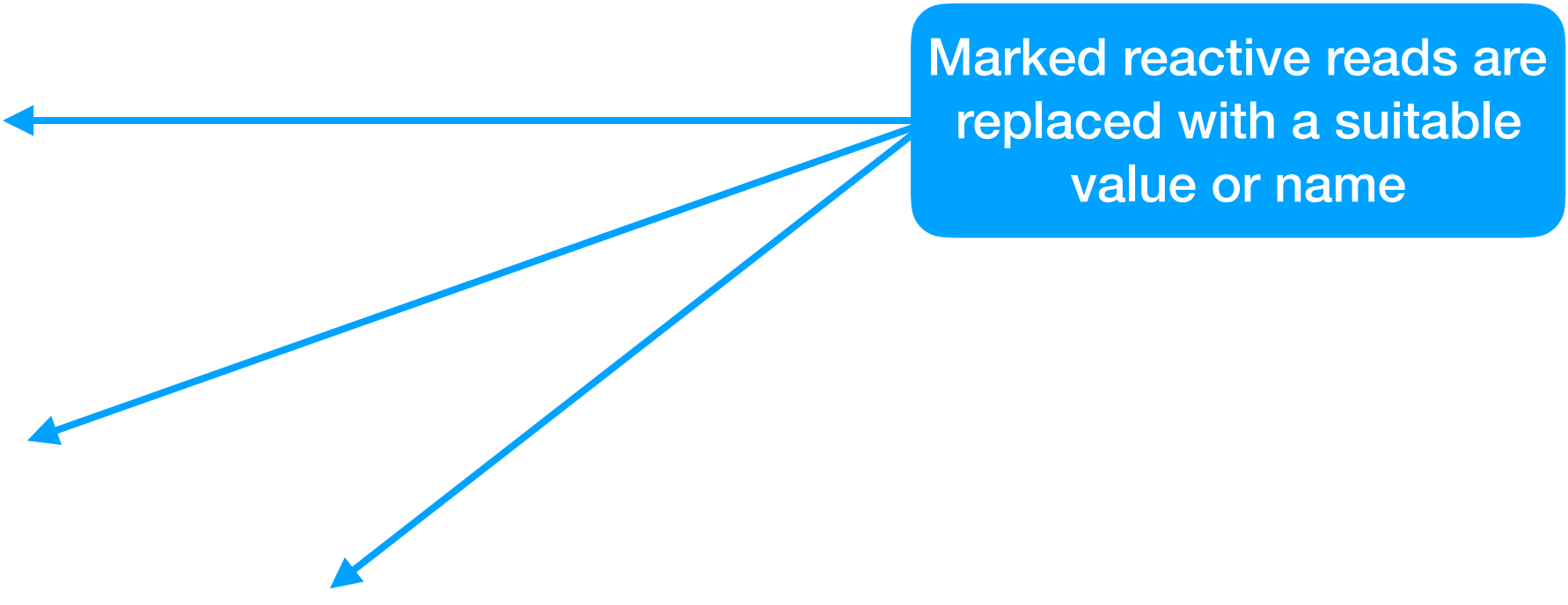
```
> expandChain(output$plot)

downloads <-
  cranlogs::cran_downloads(
    "shiny",
    from = ..(format(Sys.Date() - 365)),
    to = Sys.Date()
  )

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

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

Step 3: Generate code with `expandChain()`


Marked reactive reads are replaced with a suitable value or name

Step 3: Generate code with `expandChain()`

```
> expandChain(output$plot)

downloads <-
  cranlogs::cran_downloads(
    "shiny",
    from = ..(format(Sys.Date() - 365)),
    to = Sys.Date()
  )

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

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

Marked reactive reads are replaced with a suitable value or name

Step 3: Generate code with `expandChain()`

```
> expandChain(output$plot)

downloads <-
  cranlogs::cran_downloads(
    "shiny",
    from = ..(format(Sys.Date() - 365)),
    to = Sys.Date()
  )

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

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

Other code wrapped in ..() is evaluated (i.e. unquoted)