



Reactive Programming

# • Reactive Programming -

Paradigm concerned about the  
propagation of change

- Three types of reactive elements



#Squirrels

input \$KEY\_A



#conditors

valid ~~read~~ ~~write~~ ( ) { }

input \$KEY\_A + 1



#Endpoints

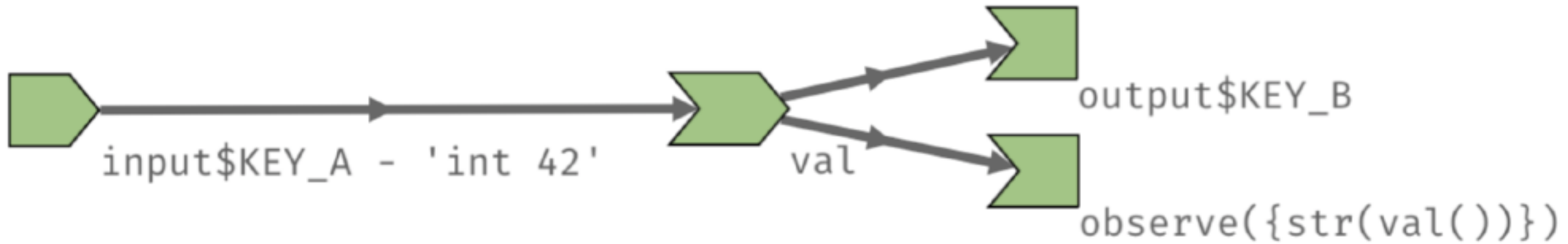
output\$KEY\_B <- renderPrint({

val()

oldseirve()

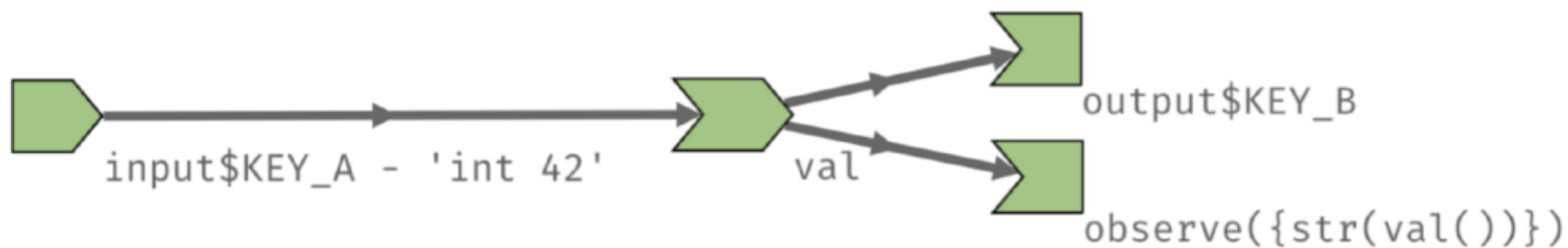


str(val))



# Reactive Programming

- Reactive Programming -  
Paradigm concerned about the propagation of change
- Three types of reactive elements



```
# Sources
input$KEY_A

# Conductors
val <- reactive({
  input$KEY_A + 1
})

# Endpoints
output$KEY_B <- renderPrint({
  val()
})

observe({
  str(val())
})
```

What is `reactlog`?