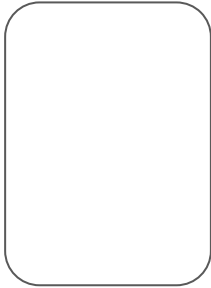


Writing your own functions in R

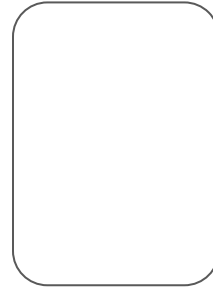
FN.5 - FN.11

# Calling a function

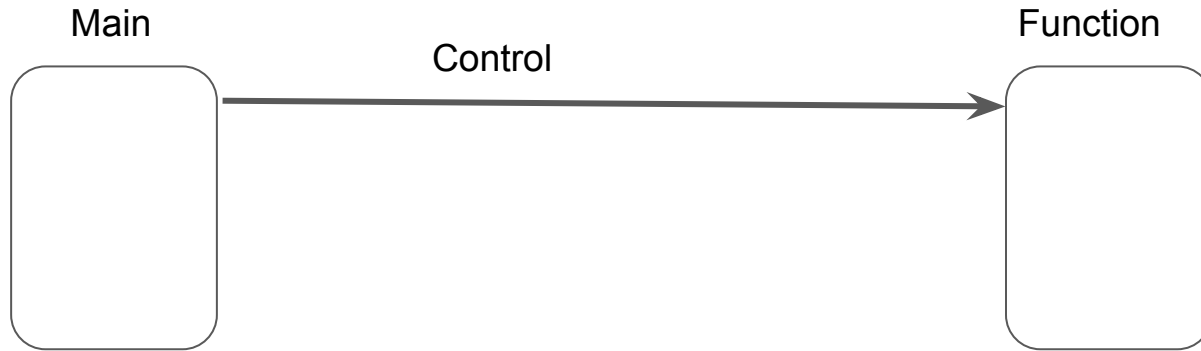
Main



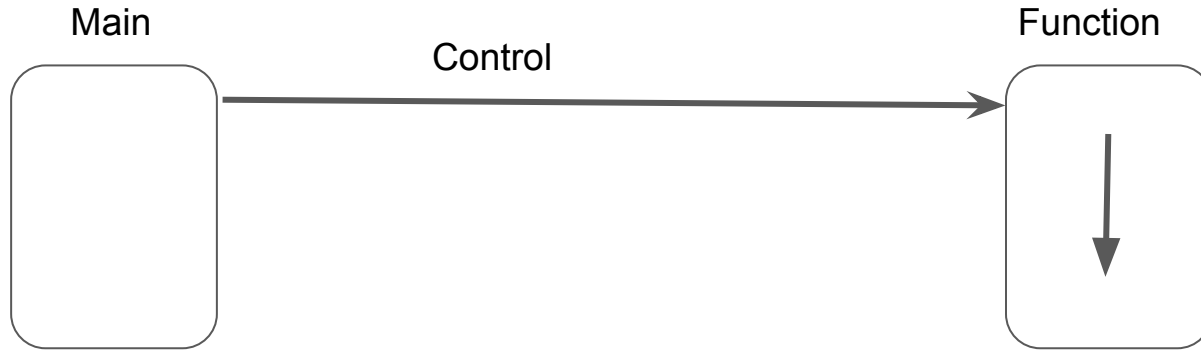
Function



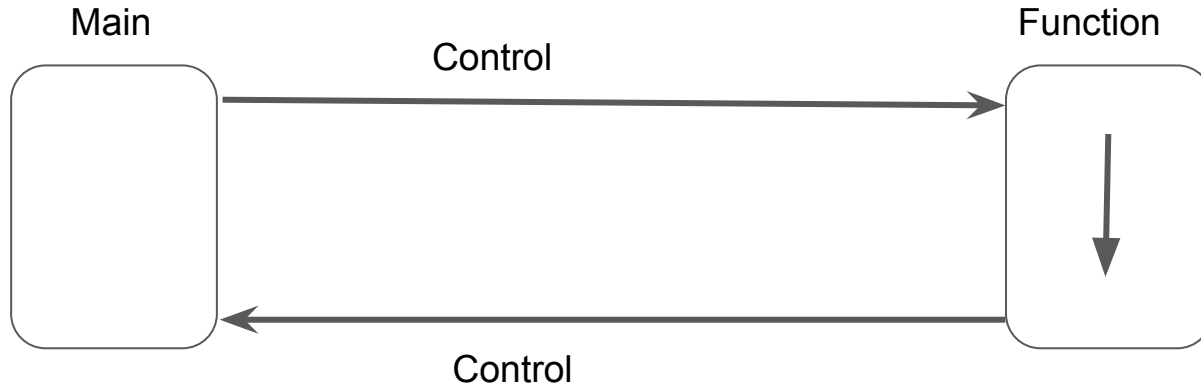
# Calling a function



# Calling a function

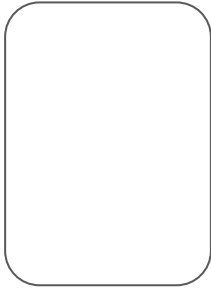


# Calling a function



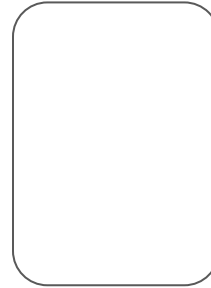
# Calling a function

Main



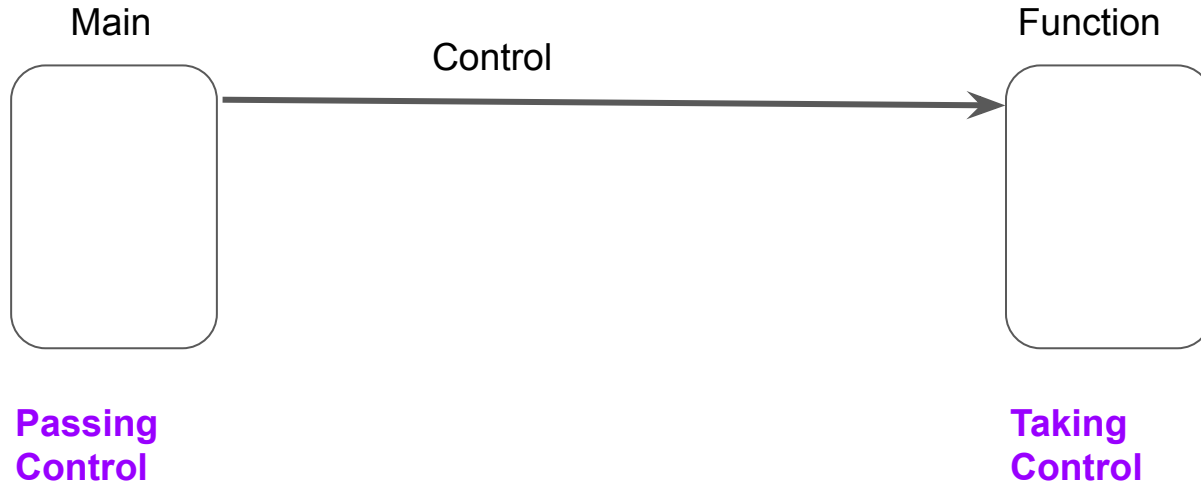
**Executing**

Function

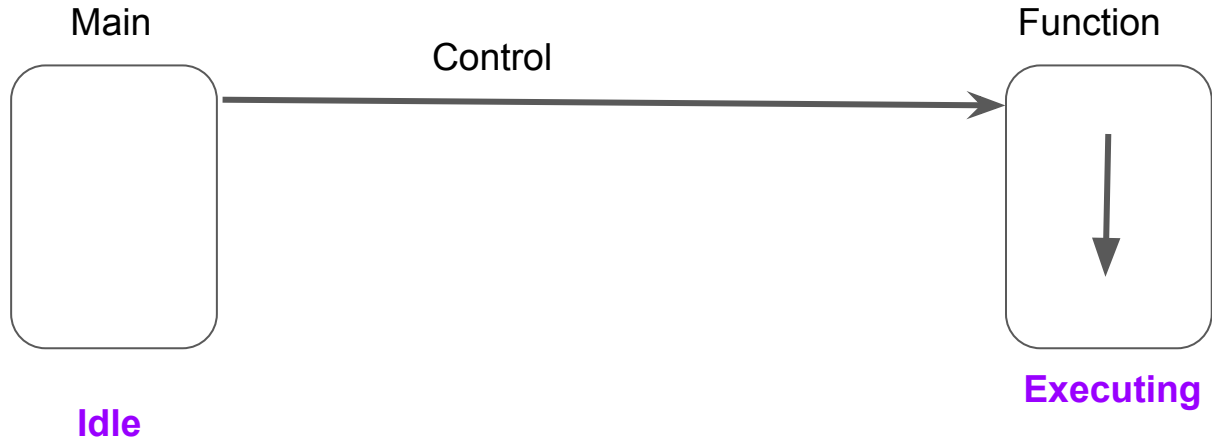


**Idle**

# Calling a function

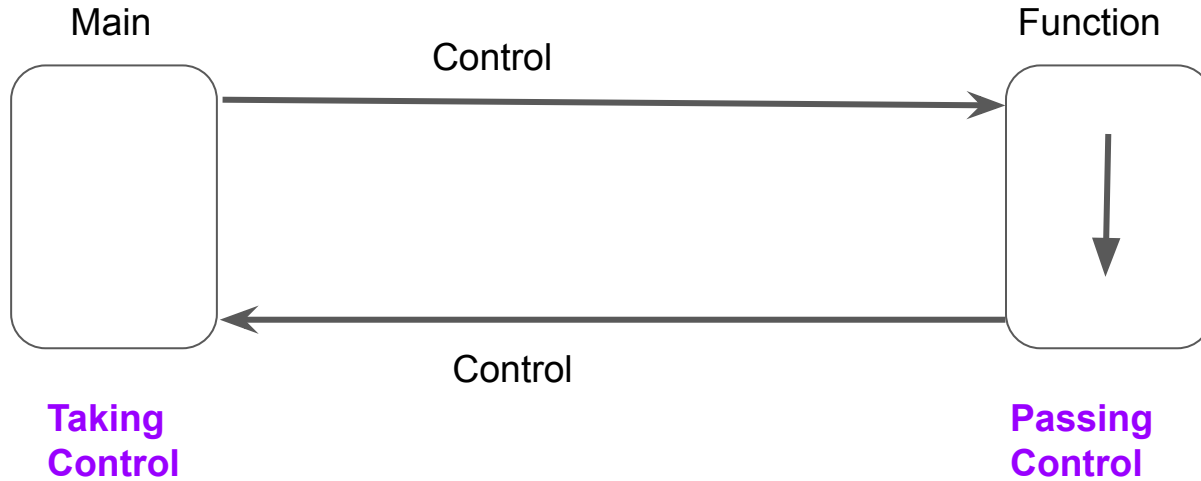


# Calling a function



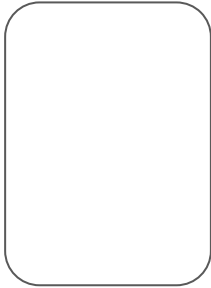


# Calling a function



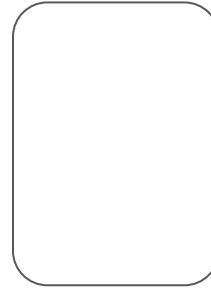
# Calling a function

Main



**Executing**

Function



**Idle**

# Calling a function

Control is given to function, control is returned after function execution.

# Calling a function

Control is given to function, control is returned after function execution.

But what about data being passed back and forth?

# Calling a function

Control is given to function, control is returned after function execution.

But what about data being passed back and forth?

Depending on data movement, there are 4 different ways to structure a function.

No data in, no data returned

# No data in, no data returned

```
menu <- function() {  
  print("Here are the options for your chosen action:")  
  print("-- a -- to add a new item to the inventory")  
  print("-- u -- to update the inventory of a item")  
  print("-- r -- to remove inventory of an item as a result  
    of a sale")  
  print("-- v -- to compute the value of all the inventory  
    in the store")  
  print("-- o -- to print out all items that need  
    reordering")  
  print("-- q -- to quit the program\n")  
}
```

# No data in, no data returned

```
menu()  
print("OK")
```

```
menu <- function() {  
  print("Here are the options for your chosen action:")  
  print("-- a -- to add a new item to the inventory")  
  print("-- u -- to update the inventory of a item")  
  print("-- r -- to remove inventory of an item as a result  
    of a sale")  
  print("-- v -- to compute the value of all the inventory  
    in the store")  
  print("-- o -- to print out all items that need  
    reordering")  
  print("-- q -- to quit the program\n")  
}
```



# Data in, no data returned

```
prettyPrint <- function(item, quantity, price){  
  line1 <- paste("In inventory we have", quantity, "of ",  
                item)  
  line2 <- paste("They sell for $", price, "each for a total  
                value of $", quantity * price)  
  print(line1)  
  print(line2)  
}  
  
prettyPrint("no-name laptop", 25, 750)
```

# No data in, data returned

```
userOption <- function() {  
  print("Here are the options for your chosen action:")  
  print("-- a -- to add a new item to the inventory")  
  print("-- u -- to update the inventory of a item")  
  print("-- r -- to remove inventory of an item as a result of a  
    sale")  
  print("-- v -- to compute the value of all the inventory in the  
    store")  
  print("-- o -- to print out all items that need reordering")  
  print("-- q -- to quit the program\n")  
  choice <- readline(prompt = "Please enter your action: ")  
  return(choice)  
}
```

# No data in, data returned

```
action <- userOption()  
print(paste("you chose ",  
           action))
```

```
userOption <- function() {  
  print("Here are the options for your chosen action:")  
  print("-- a -- to add a new item to the inventory")  
  print("-- u -- to update the inventory of a item")  
  print("-- r -- to remove inventory of an item as a result of a  
         sale")  
  print("-- v -- to compute the value of all the inventory in the  
         store")  
  print("-- o -- to print out all items that need reordering")  
  print("-- q -- to quit the program\n")  
  choice <- readline(prompt = "Please enter your action: ")  
  return(choice)  
}
```

# Data in, data returned

```
value <- function(quantity, price){  
  return(quantity * price)  
}  
  
quantity <- 25  
item <- "no name laptop"  
price <- 750  
line1 <- paste("In inventory we have", quantity, "of", item)  
line2 <- paste("They sell for $", price, "each for a total value of $",  
value(quantity, price))  
print(line1)  
print(line2)
```

# Summary

When writing your own functions, consider the following:

1. what part of the program's "job" can be compartmentalized into a function?
2. does the function need to receive any data in order to do its job?
  - a. if so, exactly what data does it need?
3. does the calling program need to receive any results from the function?
  - a. if so, how will those results be used by the calling program?

Answer these questions and think about the control and data flow ***before*** you start to code!!!