R – Functions

Abhishek Kumar
ItsAbhishekKumar.com
@MeAbhishekKumar

pluralsight
hardcore dev and IT training

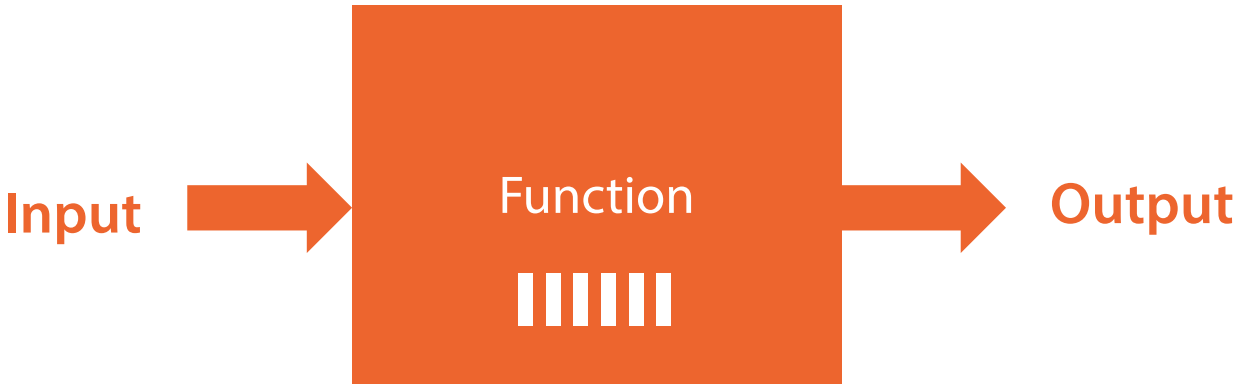
Outline

Overview:
components &
guidelines

Arguments :
matching &
evaluation

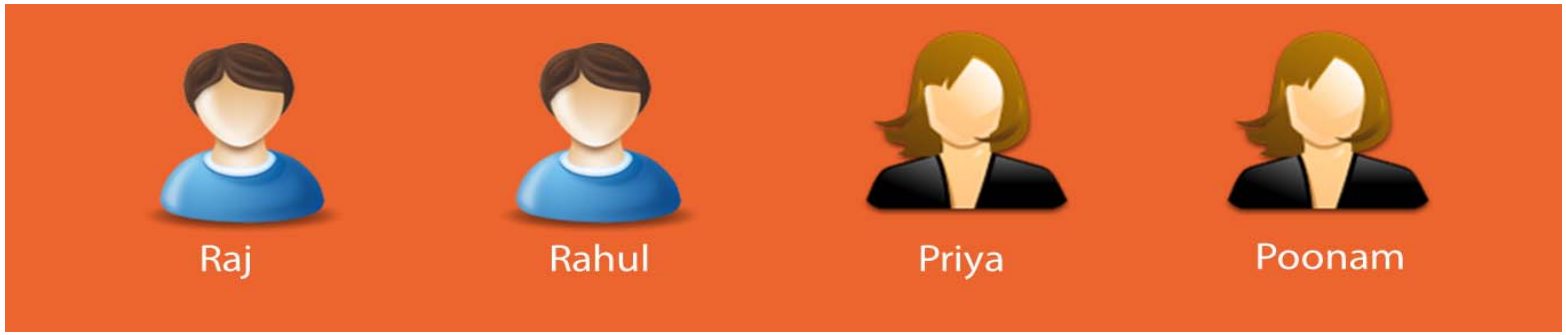
Functions as objects:
benefits & usage

Function



Function

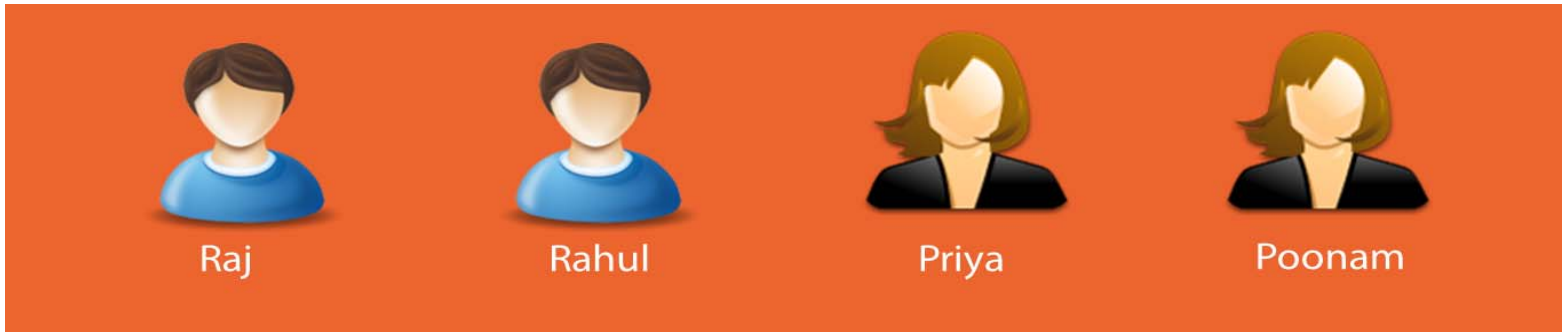
- Why do we need functions ?




```
student.physics.quiz.marks <- c( 70L , 75L , 80L,  85L)

student.physics.viva.marks <- c( 7L , 5L , 8L,  6L)

student.physics.total.marks <- student.physics.quiz.marks +
student.physics.viva.marks

student.physics.total.marks
```

Function

- Why do we need functions ?

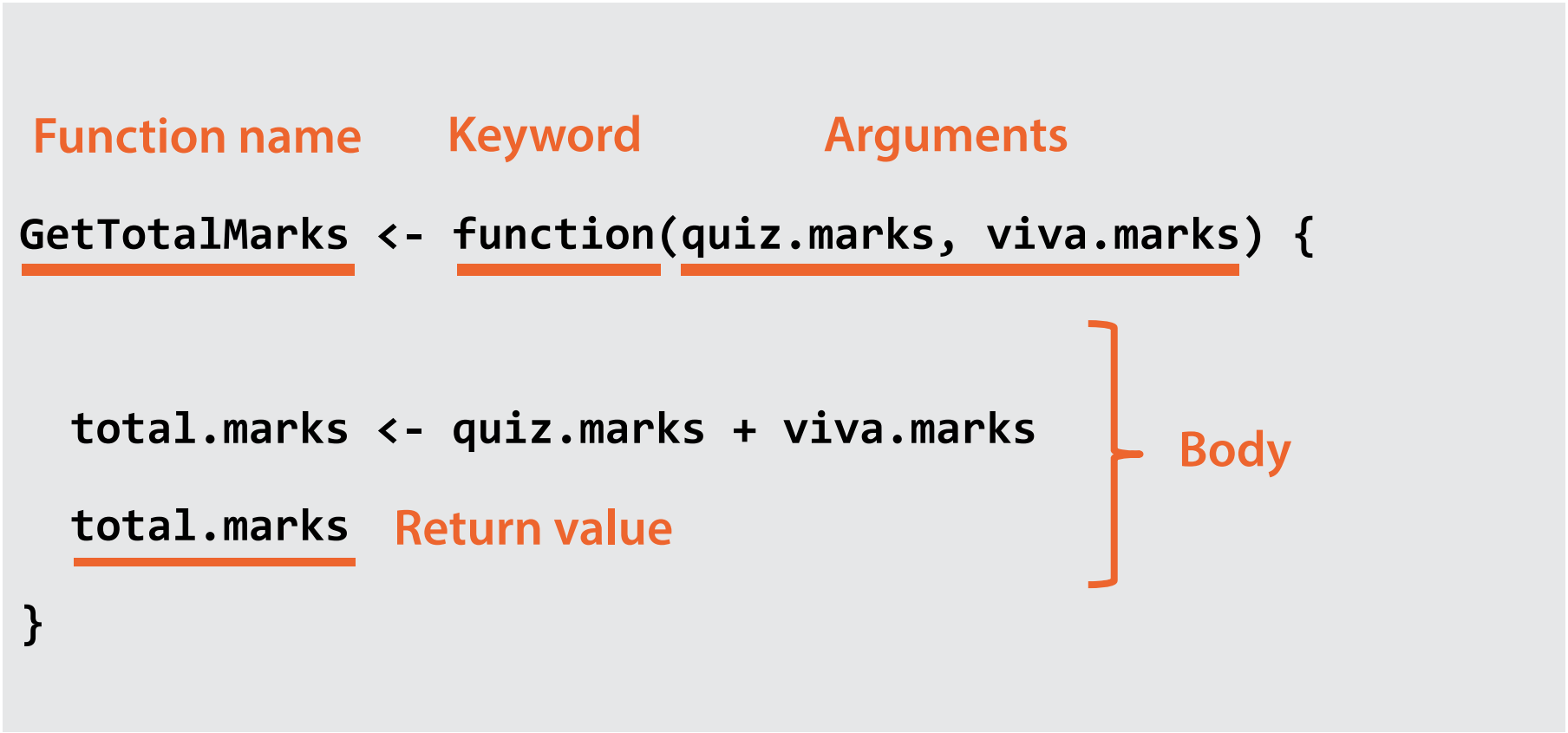



```
student.chemistry.quiz.marks <- c( 70L , 75L , 80L,  85L)

student.chemistry.viva.marks <- c( 7L , 5L , 8L,  6L)

student.chemistry.total.marks <- student.chemistry.quiz.marks
+ student.chemistry.viva.marks

student.chemistry.total.marks
```

Function

- Demo

Function Components



Function Naming Guidelines

- **Google R style guide**
  - Initial capital letters
  - With no dots, underscore etc.

AddExtraMarks 

addExtraMarks 

Add.Extra.Marks 

Add_Extra_Marks 

Argument Matching

```
MyFunction(v1, v2)
```

```
MyFunction <- function(x, y) {

  #body

}
```

Argument Matching

- Match by position

```
GetTotalMarks <- function(quiz.marks, viva.marks) {
  total.marks <- quiz.marks + viva.marks
  total.marks
}
GetTotalMarks(c(70L, 75L, 80L, 85L), c(7L, 5L, 8L, 6L))
```

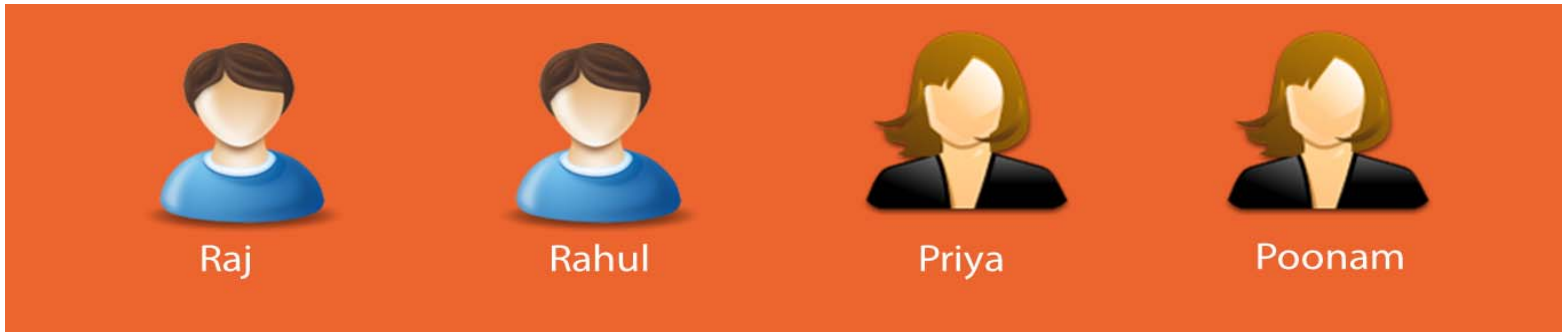
Argument Matching

- Match by name

```
GetTotalMarks <- function(quiz.marks, viva.marks) {

  total.marks <- quiz.marks + viva.marks

  total.marks

}

GetTotalMarks(quiz.marks = c(70L, 75L, 80L, 85L), viva.marks
= c(7L, 5L, 8L, 6L))

GetTotalMarks(viva.marks = c(7L, 5L, 8L, 6L), quiz.marks =
c(70L, 75L, 80L, 85L)) # reorder arguments
```

Arguments With Default Values

- Also known as default argument
- An argument which is not required to specify



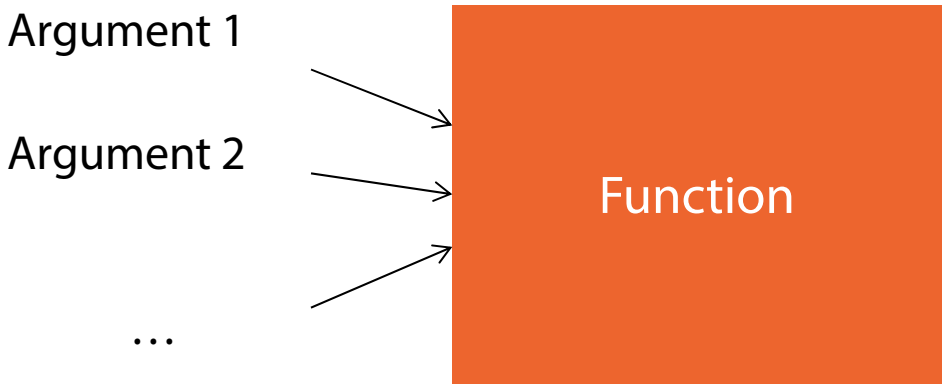
Assignment marks	5	5	5	5
	2	1	3	4

Arguments With Default Values

- Demo

Additional Arguments Using Ellipsis

- Represented by three dots (…)
- A special argument
- Means "anything else"



Additional Arguments Using Ellipsis

- Demo

Lazy Evaluation

- Evaluation of an expression is deferred until it is first used
- Common in functional programming languages

Lazy Evaluation

- Demo

Multiple Return Values

- Use list with a return statement

Multiple Return Values

- Demo

Functions as Objects

- **Functions as first class objects**
  - Look into them
  - Assign them
  - Pass them as arguments to other functions

Functions as Objects

- Demo

Anonymous Function

- Function without any name
- Use it if you have to create small functions

Anonymous Function

- Demo

Summary

Functions

Overview
Components
Naming guidelines

Argument matching
Default arguments
Using ellipsis
Lazy evaluation
Multiple return values

Functions as objects
Anonymous functions