

Unit 2—Lesson 2:

Functions

Functions

```
tieMyShoes()
```

```
makeBreakfast(food: "scrambled eggs", drink: "orange juice")
```

Functions

Defining a function

```
func functionName (parameters) -> ReturnType {  
    // Body of the function  
}
```

```
func displayPi() {  
    print("3.1415926535")  
}
```

```
displayPi()
```

3.1415926535

Parameters

```
func triple(value: Int) {  
    let result = value * 3  
    print("If you multiply \(value) by 3, you'll get \(result).")  
}
```

```
triple(value: 10)
```

If you multiply 10 by 3, you'll get 30.

Parameters

Multiple parameters

```
func multiply(firstNumber: Int, secondNumber: Int) {  
    let result = firstNumber * secondNumber  
    print("The result is \(result).")  
}
```

```
multiply(firstNumber: 10, secondNumber: 5)
```

The result is 50.

Return values

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    let result = firstNumber * secondNumber  
    return result  
}
```

Return values

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    let result = firstNumber * secondNumber  
    return result  
}
```

Return values

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    let result = firstNumber * secondNumber  
    return result  
}
```


Return values

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    return firstNumber * secondNumber  
}
```

Return values

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    return firstNumber * secondNumber  
}
```

```
let myResult = multiply(firstNumber: 10, secondNumber: 5)  
print("10 * 5 is \ (myResult)")
```

Return values

```
func multiply(firstNumber: Int, secondNumber: Int) -> Int {  
    return firstNumber * secondNumber  
}
```

```
let myResult = multiply(firstNumber: 10, secondNumber: 5)  
print("10 * 5 is \ (myResult)")
```

```
print("10 * 5 is \ (multiply(firstNumber: 10, secondNumber: 5))")
```

Argument labels

```
func sayHello(firstName: String) {  
    print("Hello, \(firstName)!")  
}
```

```
sayHello(firstName: "Amy")
```

Argument labels

```
func sayHello(to: String, and: String) {  
    print("Hello \$(to) and \$(and)")  
}
```

```
sayHello(to: "Luke", and: "Dave")
```

Argument labels

External names

```
func sayHello(to person: String, and anotherPerson: String) {  
    print("Hello \(person) and \(anotherPerson)")  
}
```

```
sayHello(to: "Luke", and: "Dave")
```

Argument labels

Omitting labels

```
print("Hello, world!")
```

```
func add(_ firstNumber: Int, to secondNumber: Int) -> Int {  
    return firstNumber + secondNumber  
}
```

```
let total = add(14, to: 6)
```

Default parameter values

```
func display(teamName: String, score: Int = 0) {  
    print("\(teamName): \(score)")  
}
```

```
display(teamName: "Wombats", score: 100)  
display(teamName: "Wombats")
```

Wombats: 100

Wombats: 0

Unit 2—Lesson 2

Lab: Functions



Open and complete the exercises in Lab – `Functions.playground`

