

# Solution Review: Anonymous Functions

In the following lesson, we will go over the solution of the challenge: Anonymous Functions.

## We'll cover the following

- Task
- Solution

## Task #

In this challenge, you had to create two functions.

1. `printAdd` - add two numbers and prints the result
2. `printSubtract` - subtracts two numbers and prints the result

## Solution #

A skeleton of the function was already provided for you. Let's look it over.

```
def arithmeticPrinter(f: (Int,Int) => Int, x: Int, y: Int) = {  
    print(f(x,y))  
}  
  
def printAdd(x: Int, y: Int) = {  
  
}  
  
def printSubtract(x: Int, y: Int) = {  
  
}
```

The `arithmeticPrinter` function you created in a previous challenge was provided for you. You had to use `arithmeticPrinter` and an anonymous function to write the function body of both `printAdd` and `printSubtract`. Remember that `arithmeticPrinter`'s first parameter is a function. The function to be passed in this challenge was an anonymous function.

- `printAdd` - To write the function body for `printAdd`, you needed to pass an anonymous function to `arithmetic` operator which takes two parameters and returns their sum.

```
arithmeticPrinter((x,y) => x+y, x, y)
```

- `printSubtract` - To write the function body for `printSubtract`, you needed to pass an anonymous function to `arithmetic` operator which takes two parameters and returns their difference.

```
arithmeticPrinter((x,y) => x-y, x, y)
```

You can find the complete solution below:

You were required to write the code on **line 6** and **line 10**.

This code requires the following environment variables to execute:

LANG C.UTF-8

```
def arithmeticPrinter(f: (Int,Int) => Int, x: Int, y: Int) = {  
  print(f(x,y))  
}  
  
def printAdd(x: Int, y: Int) = {  
  arithmeticPrinter((x,y) => x+y, x, y)  
}  
  
def printSubtract(x: Int, y: Int) = {  
  arithmeticPrinter((x,y) => x-y, x, y)  
}  
  
// Driver Code  
printAdd(75,10)
```



In the next lesson, we will learn how functions can return other functions.