

Swift Nested Functions

In this tutorial, we will learn about nested functions in Swift with the help of examples.

In Swift, a function can exist inside the body of another function. This is called a nested function.

Before you learn about nested functions make sure to know about [Swift functions \(/swift-programming/functions\)](/swift-programming/functions).

Syntax of Nested Function

Here's how we create nested functions in swift.

```
// outer function
func function1() {
    // code

    // inner function
    func function2() {
        // code
    }
}
```

Here, the function `function2()` is nested inside the outer function `function1()`



Search tutorials and examples

www.domain-name.com

```
// inner function
func displayName() {
    print("Good Morning Abraham!")
}

// calling inner function
displayName()

// calling outer function
greetMessage()
```

Output

Good Morning Abraham!

In the above example, we have created two functions:

- `greetMessage()` - a regular function
- `displayName()` - an inner function nested inside `greetMessage()`

Here, we are calling the inner function `displayName()` from the outer function.

Note: If we try to call the inner function from outside of the outer function, we'll get the error message: `use of unresolved identifier`.

Example 2: Nested Function with Parameters



(/)

Search tutorials and examples

www.domain-name.com

```
func display(num1: Int, num2: Int) {  
    print("5 + 10 =", num1 + num2)  
}  
  
// calling inner function with two values  
display(num1: 5, num2: 10)  
}  
  
// calling outer function  
addNumbers()
```

Output

```
Addition  
5 + 10 = 15
```

In the above example, the function `display()` is nested inside the function `addNumbers()`. Notice the inner function,

```
func display(num1: Int, num2: Int) {  
    ...  
}
```



Example 3: Nested Function with Return Values

```
func operate(symbol: String) -> (Int, Int) -> Int {  
  
    // inner function to add two numbers  
    func add(num1:Int, num2:Int) -> Int {  
        return num1 + num2  
    }  
  
    // inner function to subtract two numbers  
    func subtract(num1:Int, num2:Int) -> Int {  
        return num1 - num2  
    }  
  
    let operation = (symbol == "+") ? add : subtract  
    return operation  
}  
  
let operation = operate(symbol: "+")  
let result = operation(8, 3)  
print("Result:", result)
```

Output

Result: 11

In the above example, the functions `add()` and `subtract()` are nested inside the `operate()` function . Notice the declaration of the outer function

```
func operate(symbol: String) -> (Int, Int) -> Int {  
    ...  
}
```

Here, the return type `(Int, Int) -> Int` specifies that the outer function returns a function with



Search tutorials and examples

www.domain-name.com

This is why we are able to call the inner function from outside of the outer function as well using


```
let result = operation(8, 3)
```


Here, `operation(8, 3)` is replaced by either `add(8, 3)` or `subtract(8, 3)` based on the value of `symbol` passed to the `operate()` function.

Next Tutorial: [\(/swift-programming/recursion\)](/swift-programming/recursion)
Swift Recursion

Previous Tutorial: [\(/swift-programming/function-parameter-return-values\)](/swift-programming/function-parameter-return-values)
[Swift Function Parameters](#)

Share on:

 (<https://www.facebook.com/sharer/sharer.php?u=https://www.programiz.com/swift-programming/nested-functions>)

 (<https://twitter.com/intent/tweet?text=Check%20this%20amazing%20programming/nested-functions>)

Did you find this article helpful?





Search tutorials and examples

www.domain-name.com

Related Tutorials

[Swift Tutorial](#)

[Swift Functions](#)

[\(/swift-programming/functions\)](#)

[Swift Tutorial](#)

[Swift Function Parameters and Return Values](#)

[\(/swift-programming/function-parameter-return-values\)](#)

[Swift Tutorial](#)

[Nested Loops in Swift](#)

[\(/swift-programming/nested-loops\)](#)

[Swift Tutorial](#)

[Swift Generics](#)

[\(/swift-programming/generics\)](#)



Search tutorials and examples

www.domain-name.com