Swift Function

Functions in Swift are self-contained chunks of code that perform a specific task. They can be defined once and called multiple times throughout a program. Here, I'll explain the key concepts related to functions in Swift and provide detailed examples for each.

1. Basic Function

A basic function in Swift is defined using the **func** keyword, followed by the function name, a pair of parentheses, and a set of curly braces **{}** containing the code to be executed.

```
func sayHello() {
    print("Hello, world!")
}

sayHello() // Output: Hello, world!
```

2. Function with Parameters

Functions can accept parameters to pass information into them.

```
func greet(person: String) {
    print("Hello, \(person)!")
}

greet(person: "Alice") // Output: Hello, Alice!
```

3. Function with Return Value

Functions can return a value after execution. The return type is specified after the **->** symbol.

```
func add(a: Int, b: Int) -> Int {
    return a + b
}

let sum = add(a: 3, b: 5)
print(sum) // Output: 8
```

5. Parameter Labels

Swift functions can have parameter labels that provide a descriptive name for the argument when the function is called.

```
func divide(numerator: Int, by denominator: Int) -> Int {
    return numerator / denominator
}

let result = divide(numerator: 10, by: 2)
print(result) // Output: 5
```

6. Omitting Parameter Labels

You can use an underscore **_** to omit the parameter label.

```
func subtract(_ a: Int, from b: Int) -> Int {
    return b - a
}

let difference = subtract(3, from: 10)
print(difference) // Output: 7
```

7. Default Parameter Values

Functions can have parameters with default values.

```
func greet(person: String, withGreeting greeting: String = "Hello") {
    print("\(greeting), \(person)!")
}

greet(person: "Bob") // Output: Hello, Bob!
greet(person: "Alice", withGreeting: "Hi") // Output: Hi, Alice!
```

In-Out Parameters

In-out parameters allow a function to modify the value of an argument.

```
func swap(_ a: inout Int, _ b: inout Int) {
    let temp = a
    a = b
    b = temp
}

var x = 10
var y = 20
swap(&x, &y)
print("x: \(x), y: \(y)") // Output: x: 20, y: 10
```