

# Swift Interview Questions

## What is IOS Swift?

**IOS Swift** is the latest programming language released by APPLE and is meant to be replacing the Objective-C and trying to corner the industry with this prominent programming language.

So, many people have started transitioning to Swift programming language from Objective-C or just trying to learn the swift with a Java background in order to cope up with the market requirement.

For those who are preparing for a junior developer interview, here are some questions that can be helpful for you to crack the interview and to carry on your future career in the programming industry.

### Quick Questions About Swift

<b>Swift</b> is written In	C++
<b>Swift</b> is a	General-purpose, multi-paradigm, compiled programming language for iOS, macOS, watchOS, tvOS, Linux, and z/OS
<b>Swift</b> is developed By	Apple Inc
<b>Swift</b> supports	Darwin, Linux, FreeBSD, Windows, z/OS
<b>Swift</b> is designed by	Chris Lattner, Doug Gregor, John McCall, Ted Kremenek, Joe Groff, and Apple Inc
<b>Swift</b> Licence	Apache License 2.0
<b>Swift</b> Current Stable release	5.1.2

### Q1. What is IOS Swift?

**IOS swift** is an intuitive and stalwart language for iOS. It is joyful to deal with the writing stuff in Swift. It is very advanced and its syntax is also concise and expressive too. It is very good by design and is able to produce software that runs very quick. It encompasses diverse modern hallmarks.

It is one of the programming languages and also a system creating an application for iOS and OS X. For Cocoa and Cocoa touch, this programming language is considered to be an innovative one.

### Q2. What is Dictionary in Swift?

It enables you to store the key-value pairs and access the value by providing the key. It is similar to that of the hash tables in other programming languages.

### Q3. Explain Enum in Swift?

An **enum in swift** is used to defines a common type for a group of related values. It enables you to work with those values in a type-safe way within your code.

Basically, it is a type that contains a group of various related values under the same umbrella.

#### **Q4. How can you make a property optional in Swift?**

An optional in Swift is a type that can hold either a value or no value. Optionals are written by appending a ? to any type:

Example

```
var name: String? = "Online Interview Questions"
```

Declaring a question mark "?" in the code can make a property optional.

If a property doesn't hold a value, then this symbol "?" helps in avoiding the runtime errors.

#### **Q5. Explain Functions in Swift Programming?**

**Functions** let you group a series code altogether to perform a specific task. A function is written once and called any number of time in the code.

In swift programming, **func** keyword is used to create a function

```
func greet(person: String) -> String {  
    let greeting = "Hello, " + person + "!"  
    return greeting  
}
```

#### **Q6. List the features of Swift Programming?**

##### **Major features of Swift Programming**

- Closures unified with function pointers.
- Supports Generics
- Fast and concise iteration over a range or collection.
- Objective-C Interoperability
- Variables are always initialized before use.
- Memory is managed automatically.
- Arrays and integers are checked for overflow.
- It eliminates the classes that are in unsafe mode.

## Q7. How can you write a multiple line comment Swift?

The multiple line comment is written in between the symbols (/\*) at the start and (\*/) at the end.

## Q8. Can you explain Regular expression and Responder chain?

**Regular Expression** – These are the special string patterns that describe how a search is performed through a string.

**Responder Chain** – It is a hierarchy of objects that obtain the opportunity to respond to the events.

## Q9. Explain some common execution states in iOS?

The states of the common execution can be as follows:

- **Not running** – This state means that there is no code that is being executed and the application is completely switched off.
- **Inactive** – This state means that the application is running in the background and is not receiving any events.
- **Active** – This state means that the applications are running in the background and is receiving the events.
- **Background** – This state means that the application is executing the code in the background.
- **Suspended** – This state means that the application is in the background and is not executing.

## Q10. What are the important data types found in Objective-C?

There are four data types that the developers mostly use in Objective – C.

- **BOOL** – It represents a Boolean value that is true or false.
- **NSInteger** – It represents an Integer.
- **NSString** – It represents a string.
- **Cgfloat** – It represents a floating-point value.

## Q11. Explain the Adapter and Memento Pattern?

**Adapter** – It lets the classes with incompatible interfaces to work together and it wraps itself around the object to expose a standard interface to interact with that object.

**Memento** – This pattern in iOS is used as a part of state restoration. That is this externalized state can be restored without violating any encapsulation. This pattern is especially used for the archiving in Apple.

## **Q12. Why do we use swift? Mention some advantages of Swift?**

A few of the many advantages of using swift are:

- Optional types
- Closures
- Built-in error handling
- Much faster than other languages
- Supports pattern matching
- Type-safe language

## **Q13. How can you define a base class in swift?**

The classes are not inherited directly from the base class in the Swift programming language. The classes defined by the developer without specifying the superclass becomes the base class automatically.

## **Q14. What are the control transfer statements in swift?**

The control transfer statements in swift are:

- **Continue**
- Break
- Fallthrough
- Return

## **Q15. What do you mean by the term “defer”?**

The term “defer” is a keyword that provides a block of code that executes when the execution is leaving the current scope.

## **Q16. Can you explain any three-shift pattern matching techniques?**

- Typecasting Patterns – This pattern allows you to match or cast the types.
- Wildcard Patterns – This pattern matches as well as ignores any kind and type of value.
- Optional Patterns – This pattern is used to match the optional values.

## **Q17. Can you explain completion handler?**

When our application is making an API call and we are supposed to update the UI to show the data from the API call, then Completion Handler becomes handy and is super convenient.

## **Q18. Distinguish between @synthesize and @dynamic in Objective –C?**

- @synthesize – It generates the getter and setter methods for the property.
- @dynamic – It notifies the compiler that the getter and setter are implemented at some other place.

**Q19. What do you do when you realize that your App is prone to crashing?**

- Firstly, determine the iOS version or model of the device.
- Collect enough information to reproduce the issue.
- If possible, acquire device logs.
- Acquire tooling or create a unit test and begin debugging once you get the idea of the nature of the issue.

**Q20. How can you prioritize the usability of the demand Process?**

- Firstly, think like a user while designing the UX.
- Consider all the situations that prove to be useful while promoting an App.
- Continue to work on the utility of the App, even after the launch.

**Q21. How to post an HTTP request with a JSON body in Swift?**

**Q22. Explain the difference between let and var in Swift Programming?**

Both the terms 'let' and 'var' are used for declaring the functions in JavaScript but the main difference between these two terms is that the 'let' is block scoped while a 'var' is the functional scope. It is not wrong to say that a variable represented with var is defined throughout the program in relation to let

Example of var:

```
Input:
console.log(x);
var x=5;
console.log(x);
Output: undefined 5
```

Example of let:

```
Input:
console.log(x);
let x=5; console.l
og(x);
Output: Error
```

### **Q23. What is difference between single and double ? in Swift?**

**Difference between single and double ? in Swift:**

**Single ?** in Swift is used to denote an optional value either contains a value or contains nil to indicate that the value is missing.

**double ??** in swift is called nil coalescing operator that finds if optional has no value then it uses a default value.

### **Q24. List some control transfer statements used in Swift?**

Some control statements are:

- Control flow statements as the name suggests are efficient to check the course of execution in a program. There are several kinds of control transfer statements are present in Swift such as loop branch statement, loop statement, and control transfer statement
- Loop branch permit a block of code to be executed time and again
- Branch statements permit a particular block of code to be executed only if some circumstances are satisfied,
- Control transfer statements present a method to change the system in which code is executed.

### **Q25. Explain Deinitializer in Swift programming Language?**

In Swift, a deinitializer is used to free up the resources by deallocates your instance class when they remain no longer useful for you. a dinit keyword is used to represent it. It is only present on class types.

### **Q26. What is the difference between nil and None in Swift?**

### **Q27. What are generics? How to make a method or variable generics in Swift?**

Generic is efficient to write sensitive and reusable functions and types. It is provided in Swift 4. Most of the people use this feature to evade the risk of duplication. 'Arrays' and 'Dictionary' are also referred to generic collection. The libraries of Swift 4 are comprised of generic code.

You can add generics to a method (function) but can't add to a variable. Understand it with an example:

```
func Vector3D(x: [T], y: [T], z: [T])
```

```
{
    self.xCord = x
    self.yCord = y
    self.zCord = z}
```

In this example, the generic created here is in a function. You can change the T with anything you want. This is how you can add generic in function.

## **Q28. [How to call Objective-C code from Swift](#)**

Follow the following steps:

**Step 1: Add Objective-C Implementation --.m:** First of all, add a .m file to the class and then name it CustomObject.m.

**Step 2: Add Bridging Header:** At the time of adding your .m file, you receive a prompt with three options of YES, NO and cancel. You need to select YES

**Step 3: Add Objective-C Header --.h:** After that Add one more .h file and name it CustomObject.h.

### **Step 4: Build your Objective-C Class**

#### **In CustomObject.h**

```
#import <Foundation/Foundation.h>
@interface CustomObject : NSObject
@property (strong, nonatomic) id someProperty;
- (void) someMethod;
@endIn CustomObject.m
#import "CustomObject.h"
@implementation CustomObject
- (void) someMethod {
    NSLog(@"SomeMethod Ran");
}
@end
```

Step 5: The next step is to add Class to Bridging-Header

#### **In YourProject-Bridging-Header.h**

```
#import "CustomObject.h"
```

### **Step 6: At last, use your Object**

#### **In SomeSwiftFile.swift:**

```
var instanceOfCustomObject: CustomObject = CustomObject()
instanceOfCustomObject.someProperty = "Hello World"
println(instanceOfCustomObject.someProperty)
instanceOfCustomObject.someMethod()
```

In the bridging header, there is no need to import explicitly.

## **Q29. How to create a constant in Swift Programming?**

You have to use the **let keyword** to declare a constant in Swift Programming. Here is the example in which we show you how to use declare street as a constant in lieu of a variable. But for this, you need to use let keyword.

1. let street: String = "5th Avenue"
2. var number: Int
3. street = "Main Street"
4. number = 10

Swift does not allow you to change the value of constant. If you want to update the first line then XCode will show an error for a naive reason. Comment out or remove the line in which you want to change the value to the street so that it won't create any type of error.

1. let street: String = "5th Avenue"
2. var number: Int
3. // street = "Main Street"
4. number = 10

## **Q30. In Swift What is use of backticks ` while declaring a variable?**

Backticks (`) are used as a name and to surround the keyword in the case if you want to give the same name as a reserved Swift keyword to a constant or a variable. But it is strongly advised to ignore the keyword unless you have no choice available.

## **Q31. What is nil-coalescing operator?**

## **Q32. How to declare an empty dictionary in Swift?**

## **Q33. What are the differences between a struct and a class?**



### **Q34. In Swift, what type of object are basic data types?**

Swift utilizes the standard set of basic data types for different purposes such as numbers, Boolean values, and strings.

**1.Int:** Int is used to store the integer value. On 32-bit devices, it is always 32 bits deep and on 64-bit devices, it has 64 bits deep. You are able to access the maximum and minimum values the type enable store with the help of the min and max static properties:

```
println("\(Int.min)")
//output on 32-bit device: -2147483648
//output on 64-bit device: -9223372036854775808
println("\(Int.max)")
//output on 32-bit device: 2147483647
//output on a 64-bit device: 9223372036854775807
```

**2. Double and Float:** The use of Float and Double in Swift is considered when you are used to working with the decimal number. Double will be always 64-bit whereas float is always a 32-bit value. It does not depend upon the architecture of the device. The compiler utilizes the Double type instead of a Float in case of using decimal literal values. So, if you don't need the value of a 64-bit value, then it is definitely that you declare

Float variable.

```
var distance = 0.0 //distance is a Double var seconds: Float = 0.0 // second is a Float
```

**3. Bool:** The Bool basic type is used to store the Boolean value. The main difference between the Objective-C and Swift is it uses true and false instead of YES and NO as in Objective-C

It is unable to compare the value to 0 or nil in Swift. The expression that can definitely retort a Bool can come into used to describe a boot value. Let's take an example

```
var someInteger = 0
var hasSome:Bool = (someInteger != 0)
//hasSome: false
someInteger = 100
hasSome = (someInteger != 0)
//hasSome = true
```

**4. Strings:** String literals are the text that is enclosed by double quotes in Swift. var greetingString = "I am Rich"

A string is the factory of the characters and the character pictures a Unicode character, has symbols, one of more than 100 scripts and 110,000 characters. There are a large number of character encoding method to implement the characters like UTF-8 or UTF-16. These encoding methods store every character with the help of a variable number of bytes in memory.

**5. Arrays:** In Swift, arrays are the collection types. It is an arranged list of the items of the equal type. At the time of declaring an array in Swift, you need to specify what type it will contain. When it is done, it is able to contain only that type. This whole process is to help you in ensuring the expected guaranteed type when you pull an item out of the array. You enclosed a list of elements with square brackets to create an array literal. For example `var dogs = ["Harlow", "Cliff", "Rusty", "Mia", "Bailey"]`

You need to assure that all the should be of equal types or else you will get a compile-time error.

It consists of two ways to represent that is long form and short form. Both the ways are comparable can be used mutually

Long form: `Array` Short form: `[ValueType]`

**6. Dictionaries:** A dictionary is an unordered factory of items of a particular type and every type is connected with a unique key. Like the arrays, it also contains the two ways to represent the type: the short form and the long form. Both of these types are comparable and can be used mutually.

Long form: `Dictionary<KeyType, ValueType>`

Short form: `[KeyType: ValueType]`

Here is an example of syntax to declare and initialize a dictionary with the use  
`var people: [String:SomePersonClass] = [:] //explicit type`  
`//or, alternately`  
`var people = [String:SomePersonClass]() //implicit type`

### **Q35. What are the two main Integer types in Swift?**

Int (signed) and UInt (unsigned) are the two main Integer types in Swift Programming.

### **Q36. What are initializer in Swift?**

**Swift initializer** is a special type of method that can be called to create a new instance of a particular type.

Swift provides a default initializer for any structure or class that provides default values for all of its non-optional properties and does not provide the initializer itself.

Swift initializer Example

```
class User {
    var first_name: String
    var last_name: String

    init(first_name: String, last_name: String) {
        self.first_name = first_name
        self.last_name = last_name
    }
}
let p = Person(first_name:"Santosh", last_name:"Botre")
```

### **Q37. What is forced unwrapping? Why is it potentially unsafe?**

**Force unwrapping in swift** is a process of converting an optional type to a non-optional type.

Force unwrapping is potentially unsafe because if the optional has no value and you try to unwrap it, this will trigger a runtime error and cause the app to crash

### **Q38. What is an in-out parameter in Swift?**

In-out parameter lets you change the value of a function parameter from within the body of that function.

### **Q39. What is an attribute in Swift?**

In Swift, an attribute is a construct which provides some additional information of declaration or its type. It is represented with @ symbol and then succeeded by the attribute name and arguments. Arguments are distinct only to some attributes. For example,

```
@available(<attribute_name>)
```

```
@available(<attribute_name>)(<arguments>)
```

```
@available
```

### **Q40. Explain What is half-open range operator?**

**Half-Open Range Operator:** It is an operator ( $a < b$ ) that is used to defines a range that runs from a to b but it doesn't contain b. The reason for its name that is half-open is that it does not contains its last value, just have its first value. The value of 'a' compulsory is not greater than b in the closed ranged operator. The resulting range will be empty in the case of a is equal to b.

It is useful when you are working with zero-based lists like arrays where it is useful to calculate the length of the list. For example:

```
let names = ["Alley", "Anna", "Jack", "Brian"]
let count = names.count
for i in 0..
```

### **Q41. What is benifit of using higher order functions?**

Higher order functions are those in which you can pass a function as an argument and returns a function as a result. There are many benefits to using them:

- They provide flexibility but compromises with complexity.
- They are useful in asynchronous communications in which simple functions can't be used.
- They improve the readability of code and decreases the length of the code.

#### **Q42. What is tuple? How to create a tuple in Swift ?**

A tuple is used to arrange multiple values in a single compound Value. In tuple, it is not compulsory for value to be of the same type. It can be of any type.

For an instance, ("interview", 123) is a tuple having two values: one is string Type, and another one is an integer type.

**To create a tuple:** To create it, you should have tuple literals which are a list of values listed in tuple separated by the comma. For example, the point is a tuple created by tuple literals written in between a pair of parentheses. To change a tuple literal, use dot notation with tuple's name and assign the desired value. Check example:

```
var point = (0, 0)
point.0 = 10
point.1 = 15point // (10, 15)
```

#### **Q43. What is Swift Messages?**

**Swift Messages** is a library in iOS which used for displaying brief messages in the form of a status bar across the top or bottom of the screen.