

IOS Development Interview Question Part 1

1) What is meant by App Delegate:-

application(_:didFinishLaunchingWithOptions:): Called when your app is first launched.

applicationWillEnterForeground(_:): Triggered when the app is about to become active after being in the background.

applicationDidBecomeActive(_:): Indicates that the app is now active and in the foreground.

applicationWillResignActive(_:): Called when the app is about to enter the background.

applicationDidEnterBackground(_:): Occurs when the app has entered the background.

applicationWillTerminate(_:): Triggered just before the app terminates.

2) What is meant by Delegate:-

Marathi Explanation:-

डेलिगेट हा एक डिझाईन पॅटर्न आहे ज्यामध्ये एक ऑब्जेक्ट (उदा., बटन, टेबलव्यू) दुसऱ्या ऑब्जेक्टला (प्रतिनिधीला) त्याच्या काही क्रिया किंवा निर्णयांची जबाबदारी सोपवतो. उदाहरणार्थ, जेव्हा तुम्ही टेबलव्यूमधील एखादी सेल निवडता, तेव्हा टेबलव्यू स्वतः हा इव्हेंट हाताळत नाही तर तो त्याच्या डेलिगेट (सहसा ViewController) ला सांगतो, "हे इव्हेंट झाले आहे, तू यावर काही कृती कर".

डेलिगेटची कामे:

- इव्हेंट हाताळणे: उदा., बटन क्लिक, टेबल सेल निवड.
- डेटा अपडेट करणे: उदा., टेक्स्टफील्डमधील टेक्स्ट बदलल्यावर डेलिगेटला सूचित करणे.
- कम्युनिकेशन: एका ऑब्जेक्टची माहिती दुसऱ्या ऑब्जेक्टपर्यंत पोहोचवणे.

उदाहरण:

समजा, तुमच्याकडे एक UITableView आहे. त्याला कोणती सेल दाखवायची, सेलवर क्लिक झाल्यावर काय करायचे, हे सगळे तो टेबलव्यू स्वतः ठरवत नाही. त्याऐवजी तो ViewController ला (त्याच्या डेलिगेटला) विचारतो:

- "मला किती सेल दाखवायचे आहेत?" (numberOfRowsInSection)
- "सेलवर क्लिक झाला तर काय करू?" (didSelectRowAt)

या पद्धतीमुळे कोड मोड्युलर, सुस्पष्ट आणि पुनर्वापरायोग्य बनतो.

मराठीत टिप:

- डेलिगेट म्हणजे "प्रतिनिधी" – एखाद्या कामासाठी जबाबदार असलेली व्यक्ती.
- प्रोटोकॉल (Protocol) हे डेलिगेटसाठी नियम (करार) असतात, जे म्हणतात, "जर तू माझा डेलिगेट बनणार असेल, तर तुला हे हे मेथड्स लिहायचे असतील."

3) What are the Table View DataSource Methods ???

In iOS app development, the **UITableViewDataSource** protocol is used to provide data to a **UITableView**.

func numberOfSections(in tableView: UITableView) -> Int

func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell

4) Geometry Reader in SwiftUI

Geometry Reader चा उपयोग **responsive UI** तयार करण्यासाठी केला जातो, म्हणजेच वेगवेगळ्या स्क्रीन आकारांवर **layout** योग्यप्रकारे जुळवून घेण्यासाठी.

Geometry Reader चा उपयोग कसा करायचा?

जेव्हा तुम्ही **Geometry Reader** वापरता, तेव्हा त्याला **Geometry Proxy** नावाचा एक **parameter** मिळतो. याच्या मदतीने तुम्ही खालील गोष्टी मिळवू शकता:

- **size** → ह्या **view** ची रुंदी आणि उंची
- **frame** → ह्या **view** ची जागा (position)
- **coordinate Space** → वेगवेगळ्या **coordinate spaces** मध्ये स्थान

Geometry Reader कुठे वापरावा?

1. **Dynamic layouts** → वेगवेगळ्या स्क्रीन साइजसाठी **responsive UI** बनवण्यासाठी
2. **Custom positioning** → **child views** साठी **adaptive positioning**
3. **Scroll effects** → **Parallax scrolling**, **Sticky headers** यांसाठी

निष्कर्ष:

GeometryReader चा उपयोग करून तुम्ही UI ला डायनामिक आणि **flexible** बनवू शकता, ज्यामुळे तो **responsive design** मध्ये खूप उपयुक्त ठरतो

5) What is the difference between a Text and a Label in SwiftUI?

A Text is a low-level text view that can be customized with various text styles and attributes, while a Label is a higher-level text view that uses a standard font and text color by default.

6) What is meant by Swift Data in SwiftUI ???

SwiftUI मध्ये SwiftData म्हणजे डेटा व्यवस्थापन आणि कायमस्वरूपी (Persistent) स्टोरेज करण्यासाठी Apple ने सादर केलेली

नवीन फ्रेमवर्क आहे. ही Core Data चा एक प्रगत पर्याय असून डेटा स्टोअर करणे, व्यवस्थापित करणे आणि अपडेट करणे अधिक सोपे बनवते.

7) How can you use animations in SwiftUI?

SwiftUI provides several animation functions such as `animation()`, `withAnimation()`, `transition()` and `matchedGeometryEffect()` that can be used to animate views. You can also use the `animatableData` property to create animatable properties in your custom views.

8) Nested Functions in Swift (iOS)

In Swift, functions can be nested inside other functions. Nested functions are useful when you want to encapsulate helper logic that is only relevant within a specific function. These inner functions have access to variables from their outer functions

Nested Function म्हणजे काय?

iOS Swift मध्ये, **Nested Function** म्हणजे एका फंक्शनच्या आत दुसरे फंक्शन (sub-function) डिफाइन करणे. हे फंक्शन बाहेरून थेट ऍक्सेस करता येत नाही, फक्त मुख्य फंक्शनच्या आत वापरता येते.

Nested Function चे फायदे:

1. कोड मॉड्युलर आणि स्वच्छ दिसतो – मोठ्या फंक्शनमधील विशिष्ट भाग वेगळा करता येतो.
2. डेटा एन्कप्सुलेशन (**Encapsulation**) – बाहेरच्या कोडला अॅक्सेस देत नाही.
3. कोडची पुनरावृत्ती (**Reusability**) कमी होते – nested function परत परत वापरता येते.

```
func makeIncrementer(incrementBy value: Int) -> (Int) -> Int {
```

```
    func incrementer(number: Int) -> Int {
```

```
        return number + value
```

```
    }
```

```
    return incrementer
```

```
}
```

```
let addFive = makeIncrementer(incrementBy: 5)
```

```
print(addFive(10)) // Output: 15
```

9) What is the difference between app id and bundle identifier in IOS SWIFT ???

1.Bundle Identifier (बंडल आयडेंटिफायर)

- बंडल आयडेंटिफायर हा प्रत्येक iOS अॅपसाठी अनोखा असतो.
- हा Xcode मध्ये सेट केला जातो आणि प्रोजेक्टच्या **Info.plist** फाईलमध्ये स्टोअर केला जातो.
- बंडल आयडेंटिफायर अॅपला iOS सिस्टीममध्ये वेगळ्या ओळखीने नोंदवतो.
- हे एका उलट डोमेन स्वरूपात असते (Reverse Domain Name Notation), जसे की:

2. App ID (अॅप आयडी)

- **App ID** हा अॅपचा एक विशिष्ट ओळख क्रमांक असतो, जो **Apple Developer Account** मध्ये तयार केला जातो.
- App ID दोन भागांत विभागलेला असतो:
 1. **Team ID (टीम आयडी)** – हा Apple Developer Account वर आधारित असतो आणि आपोआप निर्माण होतो.
 2. **Bundle ID (बंडल आयडी किंवा App-Specific ID)** – हा तुमच्या अॅपसाठी वेगळ्या ओळखीने सेट केला जातो.
- App ID काही वेळा **Explicit App ID** (निश्चित बंडल आयडीसह) किंवा **Wildcard App ID** (**com.companyname.*** यासारख्या) स्वरूपात सेट केला जातो.

वैशिष्ट्य	Bundle Identifier	App ID
कोठे सेट केला जातो?	Xcode मध्ये (Info.plist मध्ये)	Apple Developer Portal मध्ये
अनोखा असतो का?	हो, प्रत्येक अॅपसाठी वेगळा असतो	हो, प्रत्येक अॅपसाठी एक App ID असतो
कार्यप्रणाली	अॅप सिस्टीममध्ये वेगळ्या ओळखीने नोंदतो	अॅप डेव्हलपमेंट आणि डिस्ट्रीब्युशनसाठी आवश्यक

वैशिष्ट्य	Bundle Identifier	App ID
उदाहरण	com.abc.myapp	ABCDE12345.com.abc.myapp (Team ID + Bundle ID)

10) What are the delegates in IOS Swift ???

Delegate Method	Description
<code>tableView(_:heightForRowAt:)</code>	Set the height for each row.
<code>tableView(_:didSelectRowAt:)</code>	Called when a row is selected.
<code>tableView(_:willSelectRowAt:)</code>	Asks the delegate if a row should be selected.
<code>tableView(_:didDeselectRowAt:)</code>	Called when a row is deselected.
<code>tableView(_:heightForHeaderInSection:)</code>	Sets the height for the header in a section.
<code>tableView(_:viewForHeaderInSection:)</code>	Returns a custom view for the header in a section.
<code>tableView(_:heightForFooterInSection:)</code>	Sets the height for the footer in a section.
<code>tableView(_:viewForFooterInSection:)</code>	Returns a custom view for the footer in a section.

tableView(_:willDisplay:forRowAt:)	Called just before a cell is displayed.
tableView(_:willDisplayHeaderView:forSection:)	Called just before a header view is displayed.
tableView(_:willDisplayFooterView:forSection:)	Called just before a footer view is displayed.
tableView(_:accessoryButtonTappedForRowWith:)	Handles the tapping of an accessory button in a row.
tableView(_:editingStyleForRowAt:)	Defines the editing style (delete, insert, etc.) for a row.
tableView(_:commit:forRowAt:)	Commits insertions or deletions in a table row.
View(_:canEditRowAt:)	Determines if a row can be edited.
tableView(_:canMoveRowAt:)	Determines if a row can be moved.
tableView(_:moveRowAt:to:)	Handles reordering of table rows.
tableView(_:shouldIndentWhileEditingRowAt:)	Determines whether a row should be indented while editing.
tableView(_:leadingSwipeActionsConfigurationForRowAt:)	Defines actions when the user swipes left to right.
tableView(_:trailingSwipeActionsConfigurationForRowAt:)	Defines actions when the user swipes right to left.

11) Difference between frame and width in IOS Swift

frame:- represent a view's location(x,y) and size (width, height) relative to its parent view's(superView) coordinate system.

bounds:- represent a view's location (x,y)and size (width, height) relative to its own coordinate system

12) What does Generics mean in IOS Swift ???

Generics in Swift allow you to write flexible and reusable functions, classes, structures, and enumerations that can work with any type. By using generics, you can avoid duplication and provide type safety in your code.

```
func swapTwoValues<T>(_ a: inout T, _ b: inout T) {  
  
    let temp = a  
  
    a = b  
  
    b = temp  
  
}
```

```
var int1 = 10  
  
var int2 = 20  
  
swapTwoValues(&int1, &int2)  
  
print(int1, int2) // Output: 20 10
```

```
var str1 = "Hello"  
  
var str2 = "World"  
  
swapTwoValues(&str1, &str2)  
  
print(str1, str2)
```

13) Asynchronous and Synchronous in ios swift

1. सिंक्रोनस (Synchronous) म्हणजे काय?

सिंक्रोनस कोडमध्ये प्रत्येक ओळ मागील ओळ पूर्ण झाल्याशिवाय पुढे जात नाही. म्हणजेच, एखादी प्रक्रिया पूर्ण होईपर्यंत मुख्य थ्रेड थांबतो.

```
print("Step 1")
```

```
sleep(2) // 2 सेकंद थांबतो
```

```
print("Step 2")
```

2. असिंक्रोनस (Asynchronous) म्हणजे काय?

असिंक्रोनस कोडमध्ये प्रक्रिया बॅकग्राउंडमध्ये सुरू राहते आणि मुख्य थ्रेड थांबत नाही. म्हणजेच, इतर कार्ये सुरूच राहतात आणि नंतर प्रोसेस पूर्ण झाल्यावर रिझल्ट मिळतो.

```
print("Step 1")
```

```
DispatchQueue.global().async {  
    sleep(2) // 2 सेकंद प्रतीक्षा (पण मुख्य थ्रेड थांबत नाही)  
    print("Step 2 (Background Task Completed)")  
}
```

```
print("Step 3")
```

14) What is the flow of Push Notification in IOS Swift ?

iOS में Push Notification का फ्लो इस प्रकार होता है:

App Register करता है:

UNUserNotificationCenter से परमिशन मांगता है।

UIApplication.shared.registerForRemoteNotifications() कॉल करता है।

APNs (Apple Push Notification Service):

Apple का सर्वर device token जनरेट करता है और ऐप को देता है।

Backend Server:

यह device token स्टोर करता है और जब जरूरी हो, APNs को नोटिफिकेशन भेजता है।

APNs Notification Forward करता है:

APNs नोटिफिकेशन को iOS डिवाइस तक डिलीवर करता है।

App में Notification Handle होता है:

Foreground में `userNotificationCenter(_:willPresent:)`

Background/Closed में `didReceiveRemoteNotification` या System Notification Center में दिखता है।