



## IOS interview question

1. Protocols | How will you create custom protocols ?
  2. NSNotification vs Apple push notification
  5. Atomic vs Non atomic
  6. Copy vs retain
  7. Frame vs bounds
  8. Application states
  9. Objective c vs swift key differences and what is new
  11. Get vs post requests
  12. How to get data from the server when the application is in the background mode.
  - 15 Swift - How do you return multiple data types from a method.
  16. iOS layers
  17. Where should we store application secure information.
  18. What is NSSet
  19. Categories
  20. KVC vs KVO
  21. Blocks or Closures
  22. How will you handle app crashes when the app is already in the app store.
  23. NSJSONSerialization
  33. Use of guard keyword in swift?
  34. Use of will set and did set?
  35. Describe an interesting problem and how you solved it?
  36. Features added in iOS 11
  52. APNS maximum size of data that can be sent
- UI :
1. Application delegate methods
  2. View controller delegate methods
  3. Table view datasource and delegate methods
  4. What is a size class
  5. Can we have multiple UIWindow in an iOS application.
  6. How to override a UIView and draw to new contents to it.
  7. How to redraw a UIView?
  8. What is the use of load view?
  9. Can we retain a delegate method ?

### Parsers :

1. Sax vs Dom parsers
2. NSXML parser delegate methods
3. Incase if we need to parse a large xml document which parser would be efficient and why (DOM or SAX)

**DB :**

1. Core data layers
2. Sqlite vs core data
3. How to encrypt a core data database and Sqlite DB
4. How will you handle a database corruption.

**URL Requests :**

1. How will you initiate a post request?
2. NSURL connection delegates
3. How to cancel an url request?
4. How to make your server requests secure?

**Project Management :**

1. What are the design patterns you worked on?
2. Project management tools used ?
3. Defect management tools used ?
4. What is Agile ?
5. Crash reporting tools used ?
6. Encryption tools used ?
7. oops concepts ? Abstraction, inheritance, encapsulation, polymorphism