# Swift Advance

## 1. What is an extension?

Functionality of an existing class, structure or enumeration type can be added with the help of extensions. Type functionality can be added with extensions but overriding the functionality is not possible with extensions. Swift Extension is a useful feature that helps in adding more functionality to an existing Class, Structure, Enumeration or Protocol type. This includes adding functionalities for types where you don’t have the original source code too (extensions for Int, Bool, String etc. types).

Swift Extension Functionalities −

- Adding computed properties and computed type properties
- Defining instance and type methods.
- Providing new initializers.
- Defining subscripts
- Defining and using new nested types
- Making an existing type conform to a protocol

Extensions are declared with the keyword 'extension'

Syntax:

```
extension SomeType {
  // new functionality can be added here
```

## 5. What is Generic?

Swift 4 language provides 'Generic' features to write flexible and reusable functions and types. Generics are used to avoid duplication and to provide abstraction. Swift 4 standard libraries are built with generics code. Swift 4s 'Arrays' and 'Dictionary' types belong to generic collections. With the help of arrays and dictionaries the arrays are defined to hold 'Int' values and 'String' values or any other types. Generic code enables you to write flexible, reusable functions and types that can work with any type, subject to requirements that you define. You can write code that avoids duplication and expresses its intent in a clear, abstracted manner. more clearly, we can make such kinds of functions and types , which can work with any type like (int, string, struct, etc....). With Generic we can write a function once and use it on different types.