

Android Development Environment

What is Development Environment?

- The essential tools and setup that are necessary for performing development work are collectively called the ***Development Environment***.
- A development environment is the location where there exists tools that allow you to write, compile, test, and debug your projects before uploading it to the staging environment and then to the production environment.

Necessary tools for android development environment

Tools that are necessary for setting up an Android development environment are

- Java SE Development Kit (JDK)
- Android Software Development Kit (SDK)
- Integrated Development Environment (IDE)

JDK (Java Development Kit)

- set of Java libraries for building applications, applets, and components using the Java programming language
- includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform
- Includes Essential tools like :
 - Java Runtime Environment (JRE) that executes the byte code
 - an interpreter/loader (java) that converts Bytecode into the specific OS-compatible machine code.,
 - a java compiler (javac) that compiles your .java files into byte code,
 - an archiver (jar) that helps build jar files,
 - a documentation generator (javadoc) and
 - other tools needed in Java development.
- The natural language of Android programming is Java, the JDK is a necessary tool to have in your Android development environment.

Android Software Development Kit (SDK)

The **Android SDK** is a set of API libraries and development tools necessary to build, test, and debug applications for Android. Among other things.

It consists of the following main components:

- Android-specific libraries for building Android applications.
- A command-line tool for managing Android projects in window terminal.
- Dalvik Virtual Machine for compiling Android applications.
- Android Emulator for test running Android applications without a physical device.
- Dalvik Debug Monitor Server (DDMS) for debugging Android applications.

Required libraries, Debugger, An emulator, Relevant documentation for the Android application program interfaces (APIs), Sample source code, Tutorials for the Android OS.

Integrated Development Environment (IDE)

- Why need IDE?

To write a computer program, at the very least, you will need another computer program to act as a text editor where you can type the code in plain text. Example: *Notepad* for PC, *TextEdit* for Mac, or *vim* for Linux. Using a text editor is the most simplistic way to write your computer program.

Once the code is written it can be compiled and run by using some command-line tool in a terminal window.

However, if your program refuses to work or does not work correctly (which happens most of the time), you will have to open it back in the text editor to debug it, and then re-compile and run it again using the command-line tools in the terminal window.

The drudgery of the debugging process and the hassle of switching back and forth between the text editor and the terminal window would quickly drown out the enthusiasm of many would-be programmers.

IDE to the rescue!

Integrated Development Environment (IDE)

An IDE is a software application that facilitates application development.

In general, an IDE offers a composing and editing environment that consists of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder.

It provides an integrated environment whereby a developer can quickly execute different aspects of application development all-under-one-roof without the need to switch to other applications.

A good IDE makes you more productive and efficient. Projects can get off the ground faster with the help of scaffolding, coding can move faster with the help of intelligent autocompletes and refactoring tools, applications become more maintainable with the help of integrated unit testing tools.

Integrated Development Environment (IDE)

- purpose of Android development, you can choose between two IDEs - ***Android Studio*** or ***Eclipse***.
- **Android Studio**
Android Studio is a new open source Android development environment
 - a. developed by Google based on the open source community edition of [IntelliJ IDEA](#).
 - b. It was announced on 16 May 2013 and is currently in beta stage.
 - c. While Eclipse was built to be a general all-purpose IDE that can be used with any platforms and any languages, Android Studio is designed specifically for Android development only.
 - d. It is designed to provide new features and improvements over Eclipse ADT and will be the official Android IDE when it is ready.

Integrated Development Environment (IDE)

- Eclipse with Android Development Kit (ADT)

Eclipse is an open source IDE famous for Java development traditionally. It is written primarily in Java. It is highly extensible and versatile.

By means of appropriate plugins, Eclipse can also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Perl, PHP, Python, R, Ruby, and many more.

ADK:

to develop Android using Eclipse, a plugin called **Android Development Kit (ADK)** has to be added to Eclipse. The ADK extends the capabilities of Eclipse so that a developer can quickly set up new Android projects, create an application UI, add packages based on the Android Framework API, debug applications using the Android SDK tools, and export .apk files for distribution all within a single development environment.

Some Important Terms

- **Backwards Compatibility**

You should always declare the [minimum SDK version](#) that your application would support in the Android manifest file. This ensures that your application is backwards compatible to run on older versions (down to the minimum SDK version). At the same time, the Android system will prevent users from installing the application if their system's API Level is lower than the minimum SDK version of the application.

On the question of which minimum SDK version your application should support, you should consult [Android Dashboards](#) for the frequent update on distributions of platform versions, screen sizes and densities of Android devices.

- **Forwards Compatibility**

You should always build your Android application against the latest platform version in order to use new features when running on devices with the latest version of Android. In addition, you should also test your application on an AVD that uses a higher API Level as target than that required by your application. In other words, you should make your application forwards-compatible. Forward-compatibility ensures that when users who have used your application receive a system update, your application will continue to function properly.

Some Important Terms

- **Responsive Design**

In Android, you can use Fragments (modular Activities) in your Android applications to achieve responsive design. By using Fragment, for example, your application can show a two-column table on a tablet, with the left column showing list of items and a detailed view of the selected item on the right; the same application when runs on a smartphone will show the list of items on a single page and the selected item on another page.