

Best 7 Programming Languages for Mobile App

List of Figures :

<i>Figure 1 : Abstract & introduction</i>	<i>P.2</i>
<i>Figure 2 : JavaScript React Native</i>	<i>P.3,4</i>
<i>Figure 3 : Java Android</i>	<i>P.4,5</i>
<i>Figure 4 : Kotlin Android</i>	<i>P.5,6</i>
<i>Figure 5 : Swift iOS</i>	<i>P.6,7</i>
<i>Figure 6 : Objective-C</i>	<i>P.7</i>
<i>Figure 7 : C#</i>	<i>P.8</i>
<i>Figure 8 : C/C++</i>	<i>P.9</i>
<i>Figure 9 : Research Aims and objectives</i>	<i>P.10</i>
<i>Figure 10 : Conclusion</i>	<i>P.10</i>
<i>Figure 11 : References</i>	<i>P.11</i>

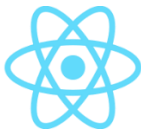
Abstract :

With the development of operating systems, new and powerful programming languages appear to support the creation and development of applications for developers. With the emergence of modern systems, some programming languages have appeared that support a dedicated platform or several platforms. In this research, the best seven programming languages for mobile applications will be mentioned.

Introduction :

With the development of technologies and devices, some modern systems have recently appeared in smart phones. With the development of smart phone systems, some programming languages have appeared that help developers create and develop their applications on dedicated platforms or multiple platforms, for example, the smartphone system iOS There are platforms for developing applications such as Swift The same is the case in Android and other OS's

1.



JavaScript React Native :

" React Native, an open-source JavaScript-based framework created by Facebook is used to create mobile apps for iOS and Android. JavaScript, a programming language, is preferred for cross-platform solutions. It allows you to reuse code to create iOS and Android apps. [1] "

JavaScript language allows to create one app that can run on different platforms instead of creating separate apps for each OS.

Why is React Native the best language for mobile app development 2022? because a few reasons :

- *Speeds up the mobile application development process up to 30%*
- *Can share the codebase for iOS, Android, and the Web (with ReactXP and React Native for Web)*
- *Requires a smaller team*
- *Lowers cost of mobile application development*
- *Ensures stable growth of the app*
- *Thanks to a wide range of libraries and tools, it has a rich ecosystem that facilitates developers' work and accelerates software development.*
- *Can integrate with the native application*

Pros of JavaScript (React Native):

- | |
|--|
| <ul style="list-style-type: none">• An open-source framework.• This helps in the creation of faster, more affordable apps.• This saves time and reduces the cost of coding overheads.• Code reuse.• It is easy to learn.• It can be easily integrated with other languages.• It is accessible to a large number of developers. |
|--|

Table1.1

Cons of JavaScript (React Native):

- Compromises data security.
- Some functions won't work with older browsers.
- Mangel of debugging facilities
- Only one inheritance is supported
- Slow bitwise function
- Even a single code error can stop rendering.

Table1.2

2.



Java (Android) :

"It is a highly required programming language in the world of mobile application development. It is considered the most widely used language in programming Android applications that uses Java JVM , which works on all systems

Today it has become very popular. In fact according to VersionEye, which tracks open source software libraries, Java has the most complete open source projects by developers.

Java is preferred by most organizations as it provides flexibility for application developers to develop mobile applications quickly and error-free."[2]

Pros of Java (Android):

- Has efficient open-source development tools.
- It is easy to understand and learn.
- It works in all systems.
- Supports the vibrant and large user community.

- | |
|--|
| <ul style="list-style-type: none">• Flexibility, versatility, and portability are all important. |
|--|

Table2.1

Cons of Java (Android):

- | |
|--|
| <ul style="list-style-type: none">• Memory management can be expensive.• Performance issues• Higher prices.• No support for low-level programming.• The lack of templates can make it difficult to develop high-quality data structures. |
|--|

Table2.2

3.



Kotlin (Android)

“ Kotlin is a statically-typed programming language. It was created to overcome the shortcomings of Java when developing Android apps. Kotlin can be used to create more efficient and faster apps. Developers can now focus on their ideas and not write mediocre code.

Kotlin, although it is still relatively young, is rapidly evolving. In 2018, 1.4 million Kotlin developers were estimated to exist around the globe.”[1]

Pros of Kotlin (Android):

- | |
|---|
| <ul style="list-style-type: none">• Simple syntax and clean code.• Simple code means easy maintenance• Eliminates all null references |
|---|

- Generates a smaller and cleaner code than Java.
- Java code is cleaner and has less code.

Table3.1

Cons of Kotlin (Android):

- Fluctuating compilation speeds
- There are not enough learning resources.
- It is still different than Java.

Table3.2

4.



(iOS) :

“Apple introduced Swift in 2014, a powerful programming language that allows for the development of iOS apps. It is not only compatible with iOS apps but also works for macOS and tvOS. Swift is an excellent choice to build high-performance iOS applications. Swift language is a great choice for developers because it allows them to write cleaner code, and provides expressive syntax.

Swift, a promising iOS programming language that is still young, is growing faster than ever and will be the future of iOS app development .”[1,4]

Pros of Swift (iOS):

- An open-source language.
- Offers interactive code, expressive syntax, and safe design.
- Modern, fast, and powerful.
- Easy to read codes.
- The code written in Swift runs lightning-fast.
- Offers multiple features to add seamless design to apps.

Table4.1

Cons of Swift (iOS):

- Its regular updates make it unstable.
- Poor interoperability with third-party tools and IDEs.

- Lacks support for earlier iOS versions.

Table4.2

5.



Objective-C

“Prior to Apple’s Swift launch, Objective-C dominated iOS app development. Objective-C, the core language for iOS mobile apps is object-oriented and general-purpose programming. It offers object-oriented performance and active runtime. Additionally, it acquires syntax and flows control statements.

The best part about Object-C is that developers can use C++ and C++ simultaneously, which is not possible with Swift. Although Swift is now more popular than Objective-C, many still prefer Objective-C.” []

Pros of Objective-C (iOS):

- A well-tested, basic language.
- Provides object-oriented performance and an active runtime.
- Objective-C is more stable than Swift.
- Easier usage of private APIs.
- Objective-C adds language-level support.
- Compatible with C and C++ language.

Table5.1

Cons of Objective-C (iOS):

- Has a steep learning curve.
- Comes with limited functionality.
- Carries more security issues.

Table5.2

6.



C-sharp

“C# (pronounced C-sharp) is a higher-level object-oriented language that uses the C programming language. It is widely used in mobile app development. C#’s cross-platform capabilities make it suitable for app development. It is primarily used for creating native apps for Microsoft platforms. However, it can also help to create native iOS and Android applications.”
[1,3]

Pros of C# Language:

- | |
|---|
| <ul style="list-style-type: none">• Its similar syntax gives developers easy learning.• Integration with the .NET library.• Pointer types are not permitted.• Integrates well with Windows with no special configurations. |
|---|

Table6.1

Cons of C# Language:

- | |
|---|
| <ul style="list-style-type: none">• Allows pointers in unsafe blocks.• Code compilation issues.• Microsoft stopped supporting. NET. |
|---|

Table6.2

7.

C/C++

“C/C++ is considered a good choice for Android app development. C++ is a good choice because of its syntactical analogy to C# and C. It is one of the most popular programming languages and has created some of the most powerful tools such as Photoshop, World of Warcraft, and Google Chrome. ” [1,2]

Pros of C/C++:

- Exceptionally fast and efficient native code.
- Gives an easy way to access the hardware.
- A vast and rich ecosystem of libraries and tools.
- Variety of data types and powerful operators.
- A vast and knowledgeable community of users.

Table 7.1

Cons of C/C++:

- No runtime checking in language.
- Crude and primitive language that lacks a safety net.
- C++ is horrendously large, complex, and complicated.

Table 7.2

Aims and objectives :

The research was written to see the best and latest languages for smartphone applications, and it was discussed in the research, mentioning the pros and cons of the mentioned languages. So that the reader can see the details, pros and cons of each phone programming language, and they differ from them to dedicated platforms and from them to several platforms.

Conclusion :

Programming languages, some of them remain at the top, and others disappear because they depend on some systems and developers and their popularity and target platforms

Does it have a sufficient audience

Programming languages may differ according to the desires and goals of the developers, perhaps targeting Android devices, so what is the point of learning Swift, or perhaps the opposite, as it depends entirely on the goals of the language and not the language itself.

References :

[1] The next tech “ <https://www.the-next-tech.com/mobile-apps/top-7-popular-programming-languages-for-mobile-app-development/> ”

[2] Devcount “ <https://devcount.com/mobile-app-languages/> ”

[3] Wikipedia

[4] Designveloper “ <https://www.designveloper.com/blog/android-app-development-programming-languages/> ”