

[Home](#) [About](#) [Contact](#) [Privacy Policy](#)

Code Carbon

All Things Flutter and Fuchsia

[Articles](#)[News](#)[Flutter](#)[Kotlin](#)[Fuchsia](#)[Dart](#)[JavaScript](#)[Tutorials](#)

ON JANUARY 5, 2021

ARTICLES

POPULAR TOPICS

Most Useful Kotlin Tips and Tricks for Beginners

ANDROID

ANDROID DEVELOPMENT

API

CHROME OS

CODE EDITORS

CODING

CROSS-PLATFORM DEVELOPMENT

DART

ECOMMERCE

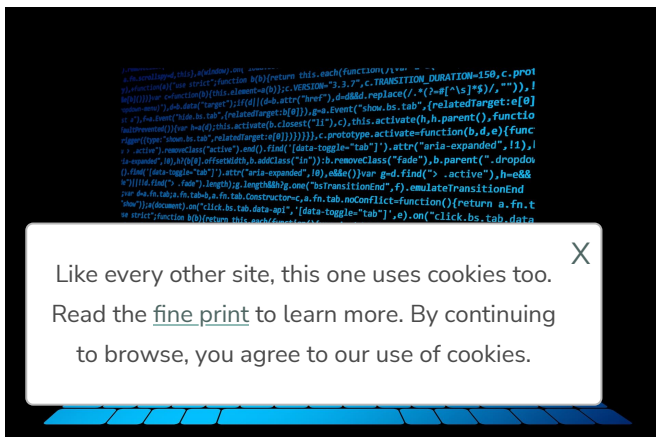
ECOMMERCE TOOLS

FLUTTER

FLUTTER WIDGETS

FREE PLUGINS

GOOGLE



Kotlin was launched to make Android development a whole lot easier as well as faster. After being named as an official language for Android, it has gradually taken over Java and is the first choice for Android developers in the current scenario.

Considering their similarities, the adaptations could not have been easier for Java developers switching to Kotlin. However, it's really easy to take notice of the unlikeness or rather conversational aspects of Kotlin on the same time. This can help to write code as same as Java.

Savvy Apps began to employ Kotlin for its new Android projects late in 2016, that is, when Kotlin 1.0.4 was launched. At first, Kotlin was tried out on a smaller scale project. It was tried and seen to ease the use of this programming language. It showed the potential to easily separate the components from business logic with the help of various extensions. Typically, the time taken for development was saved for us. And eventually, the developers decided that it would be the language of choice moving forward. And following this, the developers made use of Kotlin to

Like every other site, this one uses cookies too. Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.

Kotlin libraries too.

IOS

JAVA

JAVASCRIPT

KOTLIN

LINUX

MARKETING

MICROSOFT

MISTAKES TO AVOID

MOBILE APP DEVELOPMENT

OPTIMIZATION AND PERFORMANCE

PLUGINS

PLUGINS FOR WORDPRESS

PROGRAMMING LANGUAGES

REVIEW

SMALL BUSINESS

SOFTWARE

SOFTWARE DEVELOPMENT

SWIFT

TOOLS

VSCODE

WEB APP DEVELOPMENT

WEB BROWSER

WEB DESIGN

WEB DESIGN SOFTWARE

WEB DEVELOPMENT

WEBSITE

WHITE LABEL

WINDOWS

In order to establish their experience with Kotlin, they decided to assemble their most useful and noteworthy tips. Before we dive into these tips, you should know that these tips specifically focus on using Kotlin within the context of Android development. You might as well have experience with the Android SDK. You should also be familiar with the concept of **Kotlin plugin**. Using Kotlin with Android Studio is provided by **Jet Brains**, that is, the creators of Kotlin.

WOOCOMMERCE

WORDPRESS

WORDPRESS PLUGIN

Initial Kotlin Tips for Android

LAZY LOADING

There are various different advantages which come along with the lazy loading. Lazy loading can lead to rapid start-up time. Since loading is delayed when the variable is retrieved, this is particularly useful in using Kotlin for an Android app. For Android apps, the developers want to take the edge off the app start-up time with the intent that the user sees the app content faster, rather than sitting

a

L

memory structured, as the developers

Like every other site, this one uses cookies too.

Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.

X

only burden the resource into memory if it is addressed. Memory usage is a very well-planned on mobile platforms such as Android as the phones have limited, shared resources.

CUSTOM GETTERS/SETTERS

Kotlin's custom getters and setters exercise the structure of a model. However, you will have to state the custom behavior to get and set the domain. Let's say, you are using custom models for specific frameworks, just like the **Parse SDK**. In doing so, you are attracting values which are not really local variables in the class. However, they are stored and retrieved in some custom fashioned way, for instance from JSON.

LAMBIDAS

Lambdas diminish the lines of code in general in a source file. It also enables you for the purpose of functional programming. While lambdas are at present possible with Android, Kotlin takes them a step ahead by making sure that you don't have to deal with **Retro**

Like every other site, this one uses cookies too. Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.

The Android SDK composes a huge variety of cases where you are setting a listener or executing a single method. Therefore, Lambdas work efficiently, in those circumstances.

DATA CLASSES

Data classes disentangles the classes, adding `equals()`, `hashCode()`, `copy()`, and `toString()` formula instinctively. They also bring certain clarification in the intention of the model and what should go in it by splitting pure data from business logic.

If you are exercising data classes with something like Gson or another JSON parsing library, you can build the default constructor using the default values.

COLLECTION FILTERING

Collections are handled repeatedly when working with an API. On a very general basis, you would want to filter or bring changes to the contents of that collection. So, by employing Kotlin's collection filtering, you will successfully be able to add certain amount of clarity

to
y
te

Like every other site, this one uses cookies too. X

Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.

r to

comprise of using the collection filtering .

Straining a collection using the built-in Kotlin technique is very similar to other functional programming languages too, such as Java 8 streams or Swift collection types. Seeping through the collections in a unified way helps. It come to a great help when talking with team members about what operations need to be made to get a list down to the right functionalities to present.

OBJECT EXPRESSIONS

Object expressions permits for strict singleton definition so there is no misapprehending it for a class that can be incorporated. They also guarantee that you do not have to box those singletons somewhere like in the Application class or as a static class variable.

COMPANION OBJECT

While taking a glimpse, Kotlin seems to be missing static variables and techniques. In a manner, it does not have

th
o
o

Like every other site, this one uses cookies too. Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.

X dea
on

class. They comprise of the techniques and variables you might want to access in a static fashion. A companion object gives you the permission to define constants and methods, similar to static in Java. With it, you can follow the **newInstance** pattern of pairing.

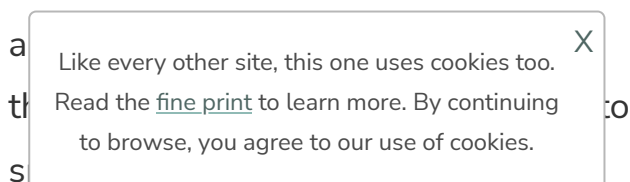
GLOBAL CONSTANTS

Kotlin permits you to define constants that span over an entire app in one place, if applicable. Customarily, constants should have their extent brought down as much as possible. However, when scope needs to be global this is an efficient way to do so without having to undergo a constants class.

OPTIONAL PARAMETERS

Alternative parameters make technique calls more malleable without having to pass null or a default value. This is specifically important when defining animations.

For instance, if you want to have a technique defined for reducing the views



Intermediate Kotlin Tips for Android

EXTENSIONS

Extensions are essential as they enable you to add to the elements of a class without having to adopt from it. With the help of the extensions, you can easily discard the need to have utility classes or techniques. It can also help you improve the readability of your code.

LATEINIT

A main component of Kotlin is its commitment to null safety. Lateinit offers a simple method to both have null safety and divide a variable the way Android needs you to. This is a great language feature. Even so, it takes some time to get used to after doing extensive Java development.

These components of the language can be disappointing when dealing with Android layouts. This is because we know that the views are going to exist within the Activity or Fragment, but we

in `onCreate` or `onCreateView` after the

Like every other site, this one uses cookies too.
Read the [fine print](#) to learn more. By continuing
to browse, you agree to our use of cookies.

X

layout is distended. You could also handle this by maintaining checks in each place you touch the view throughout the Activity. However, this would be disappointing to handle and avoidable from a null check point of view. Rather, you can use the lateinit editor.

SAFE TYPECASTING

Some Android conventions need a safe typecasting due to its normal typecasting that would cause an exception. For instance, an usual method of forming a Fragment in an Activity is to first check and see if it subsists already, with the help of the FragmentManager.

LEVERAGING LET

Leveraging let allows you to implement a block if the value of the object is not null. This further enables you to keep away from null checks and makes code more readable.

ISNULLOREEMPTY | ISNULLORBLANK

We need to certify several times in the

o

If

Like every other site, this one uses cookies too.

Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.

X pp.

making the use of Kotlin, you may have found the TextUtils class in Android

Advanced Kotlin Tips for Android

AVOIDING SINGLE ABSTRACT METHODS FOR KOTLIN CLASSES

This tip enables you to use lambdas, which enables cleaner, more compact code. For instance, when working in Java, it is very normal to have a simple listener class. One of the great features of Kotlin is that it performs **SAM (Single Abstract Method)** conversions for Java classes.

COROUTINES INSTEAD OF ASYNCTASK

Because AsyncTask is bulky and leads to leaks. The developers prefer to use coroutines as they enhance readability and don't leak memory. It's important to notice that since coroutines are

Like every other site, this one uses cookies too. Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.

most **async** purposes.

Conclusion

Thus, we have gathered these tips from what is considered to be the most efficient methods and techniques which one should learn before you get started with developing in Kotlin. These tips will for sure help you understand its key features. It will also provide you with a head start in using Kotlin for your Android projects.

References

- [16 Kotlin Tips for Android Development](#) – SAVVY
- [Kotlin: Tips and Tricks Part -1](#) – Medium
- [7 Quick Kotlin Tips for Android Developers](#) – Medium

Click [here](#) to know more learn the difference between Java and Kotlin.

ANDROID CODING JAVA KOTLIN

MOBILE APP DEVELOPMENT

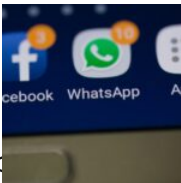
PROGRAMMING LANGUAGES

Like every other site, this one uses cookies too.
Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.





← NEXT
PREVIOUS ARTICLE
ARTICLE →



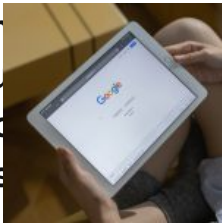
WhatsApp App
of Privacy
Project Policy
Structuring Gets
in Users
Programming
Moving
to
Signal
Private
Messenger

Recommended Articles

ON MARCH 10, 2022

ARTICLES

Why You Should Respond to Negative Reviews on Google



UPDATED ON
DECEMBER 14, 2021

ARTICLES

10 Core Marketing Tools You Can't Live Without:



Like every other site, this one uses cookies too.
Read the [fine print](#) to learn more. By continuing to browse, you agree to our use of cookies.



Marketing Game

UPDATED ON
JANUARY 27, 2022

ARTICLES

The
6
Best
Free
WordPress
Forum
Discussion
Plugins:
Allow
Visitors
to
Engage
More
With
Your
Website



© 2022 Code Carbon. All rights reserved. Managed by WebFactory Ltd Vilva | Developed By Blossom Themes. Powered by WordPress. [Privacy Policy](#)

Like every other site, this one uses cookies too.
Read the [fine print](#) to learn more. By continuing
to browse, you agree to our use of cookies.

