

Kotlin Tutorial

Welcome to Kotlin Tutorial! Kotlin has become popular with Android. If you would like to learn Kotlin programming language, this is a good start.

This Kotlin Tutorial provides a detailed introduction to Kotlin programming language, its basics, and examples to understand basic functional programming statements, object-oriented concepts, file operations, string operations, etc.

What is Kotlin

Basically, Kotlin is a programming language. It can be used to develop modern multi-platform applications for Desktop, Android, Web, etc.

Kotlin runs on Java Virtual Machine like other programming languages: Scala, Groovy, Kawa, etc.

Why Kotlin has become popular

If the word Kotlin has been buzzing around for quite some time, all the credit goes to Google Android, because Google had brought this language to lime light. Once Google Android declared that Kotlin will be one of the officially supported language for Android App development, Kotlin's popularity had surged. And Kotlin has become popular with the existing Android community and also the emerging.

Kotlin Tutorial

Kotlin programming language can be used to develop Desktop Applications and Android Applications with ease. These tutorials have been designed to present you the basics and new idiomatic approach.



A little introduction to Kotlin's history

Kotlin is relatively a new statically-typed language by 2017, developed by JetBrains. Kotlin is targeted to run on the Java Virtual Machine (JVM). Kotlin addresses most of the redundancies present in Java programming language, and also new features have been added that could make application development faster and easier.

There are other new languages with new features that could run on JVM, but Kotlin excels in the reduced

compile time (comparable to the time taken by Java Compiler) when compared to other languages.

Also, Kotlin is concise and expressive while maintaining a good compatibility with existing [Java](#) stack. And Kotlin could be written alongside Java or we could [convert existing Java classes to Kotlin files](#) or classes using IntelliJ IDEA. All these features make it easy for a Java developer to get started with Kotlin quickly.

That being said, the first stable version of Kotlin, Kotlin 1.0 has been released on 15th, Feb 2016. JetBrains would provide backward compatibility for Kotlin 1.0 for a long time. Also, Google added Kotlin as an officially supported language for Android Application development, which is available from Android Studio 3.0.

Prerequisites for this Kotlin Tutorial

Understanding of [Java](#) Basics would help a lot in learning Kotlin quickly. However this Kotlin Tutorial is designed to address both beginners and advanced developers.

IDE for Kotlin Development

Applications using Kotlin Language could be developed on IntelliJ IDEA and is recommended. Eclipse or any editor could also be used along with a compiler to compile from the command line.

Android Studio

Android Applications could be developed using Kotlin from Android Studio version 3.0.

Refer [Kotlin Android Tutorial](#) to develop Android Applications using Kotlin programming language.

Kotlin Environment Setup

First and foremost thing to do is, setting up your environment to work with Kotlin.

- [Setting up IDE for Kotlin](#)
- [HelloWorld Kotlin Program](#)

Having set up our environment and a peek of Kotlin program, let us understand the basics of Kotlin.

Kotlin Basics

Kotlin Basics cover the topics to get you started with Kotlin – like Kotlin Classes; how primary and secondary constructors work; Data Classes; looping statements; decision making statements; how you could extend the functionality of a class; enum classes and some new concepts like Null Safety.

- [Kotlin main function](#) – Kotlin Tutorial about the entry point to a Kotlin Program for execution.
- [Kotlin Classes and Constructors](#) – Kotlin Tutorial about the syntax of a Class in Kotlin, and two types of constructors: primary and secondary.
- [Kotlin Sealed Classes](#) – Kotlin Sealed Class represent restricted class hierarchies, where a value can have a type from a restricted set.
- [Kotlin Data Classes](#) – Kotlin Class that stores only properties.
- [Kotlin Inheritance](#) – The mechanism in which a class can extend the behavior of another class.
- [Kotlin Abstraction](#) – Object Oriented Programming concept that explains how abstraction works in Kotlin.

- [Kotlin Abstract Class](#) – Kotlin Tutorial about the syntax and usage of an Abstract Class in Kotlin.
- [Kotlin Interfaces](#) – Interfaces are those that are similar to normal Kotlin classes, but in addition, can contain uninitialized variables, abstract and non-abstract methods.
- [Kotlin Loops](#) – A basic introduction to Loops in Kotlin.
 - [Kotlin For Loop](#) – For loop to execute a block of statements for each element in the iterable.
 - [Kotlin Repeat Loop](#) – Repeat statement is used to execute a set of statements for a definite number of times.
 - [Kotlin While Loops](#) – Kotlin Tutorial with while and do-while loops explained.
- [Kotlin When expression](#) – Kotlin when is kind of a switch statement, that selects a case based on the value of expression evaluation.
- [Kotlin Extension Functions](#) – Kotlin extension functions actually help in extending the functionality of a class without actually inheriting the class.
- [Kotlin Null Safety](#) – Null Safety can be used to eliminate the risk of NullPointerException in real time.
- [Kotlin Enum Classes](#) – Kotlin Tutorial about how Enum class allows a variable to hold a value only from a set of predefined constants.
- [Kotlin use function](#) – In Kotlin, use is a keyword, used to execute a given block function on this resource.
- [Kotlin Ranges](#) – Kotlin Tutorial on how to express a series of numbers defined as a range and different scenarios where ranges can be used.

Kotlin Exception Handling

Exception Handling is a must known tool to every developer for any programming language.

- [Kotlin Try Catch](#)
- [Kotlin Throw Exception](#)
- [Kotlin Custom Exception](#)

Kotlin String Operations

It is very necessary to know about String Operations in programming applications. Following topics take you through some of the String Operations that are commonly used.

- [Compare Strings in Kotlin](#)
- [Split String to Lines in Kotlin](#)

Working with Files in Kotlin

If your application requires working with files which need operations like reading, writing, modifying and appending content in a file or moving files from one location to another and such, following are the tutorials that help you with file operations.

- [Reading the content of File – Kotlin Examples](#)
- [Read the content of a File as List of lines – Kotlin Examples](#)
- [Writing content to File – Kotlin Examples](#)
- [Append text to File – Kotlin Examples](#)
- [Copy a file to other location – Kotlin Examples](#)
- [Iterate through all files in Directory](#)

Kotlin Interfacing to Database

If your Kotlin Application needs interfacing to a database, following topics explain how to connect to different databases.

- [Connect to MySQL Database from Kotlin using JDBC](#)

Kotlin Error Handling

This section of tutorials focuses on how to handle most frequently occurring errors during programming in Kotlin.

- [Kotlin Error: Primary Constructor call expected](#)
- [Kotlin Error: Variable must be initialized](#)
- [Kotlin Error: Null cannot be a value of a non-null type](#)
- [Kotlin Error: Cannot create an instance of abstract class](#)

Interview Questions & Answers

Find the mostly asked Interview Questions and respective answers for Kotlin programming.

- [Kotlin Interview Questions](#)

Conclusion

By following these Kotlin Tutorials, you would definitely learn Kotlin basics and tools required to work with Desktop applications and Android Applications.

Kotlin Java

- ↳ [Kotlin Tutorial](#)

Getting Started

- ↳ [Setup Kotlin\(Java\) Project](#)
- ↳ [Kotlin Example Program](#)
- ↳ [Convert Java to Kotlin](#)
- ↳ [Kotlin Main Function](#)
- ↳ [Kotlin Loops](#)
- ↳ [Kotlin For Loop](#)
- ↳ [Kotlin While, Do While Loops](#)
- ↳ [Kotlin Repeat](#)
- ↳ [Kotlin Ranges](#)
- ↳ [Kotlin When](#)

Object Oriented Concepts

Classes

- ↳ [Kotlin - Class, Primary and Secondary Constructors](#)

↳ Kotlin Sealed Class

↳ Kotlin Data Class

↳ Kotlin Enum

↳ Kotlin - Extension Functions

Inheritance

↳ Kotlin Inheritance

↳ Kotlin Override Method of Super Class

Abstraction

↳ Kotlin Abstraction

↳ Kotlin Abstract Class

↳ Kotlin - Interfaces

↳ Kotlin Null Safety

Exception Handling

↳ Kotlin Try Catch

↳ Kotlin Throw Exception

↳ Kotlin Custom Exception

Fix Compilation Errors

↳ Kotlin - Variable must be initialized

↳ Kotlin - Primary Constructor call expected

↳ Kotlin - Null can not be a value of a non-null type String

↳ Kotlin - Cannot create an instance of an abstract class

Kotlin - String Operations

↳ Kotlin - Compare Strings

↳ Kotlin - Replace String

↳ Kotlin - Split String

↳ Kotlin - Split String to Lines

↳ Kotlin - String Capitalize

Kotlin - Functions

↳ Kotlin Function - Default Arguments

↳ Kotlin - Use Function

Kotlin Collections

Kotlin List

↳ Kotlin List

↳ [Kotlin List forEach](#)

Kotlin File Operations

↳ [Kotlin - Create File](#)

↳ [Kotlin - Read File](#)

↳ [Kotlin - Read File as List of Lines](#)

↳ [Kotlin - Write to File](#)

↳ [Kotlin - Append Text to File](#)

↳ [Kotlin - Check if File Exists](#)

↳ [Kotlin - Copy a File to Other](#)

↳ [Kotlin - Iterate through all files in a directory](#)

↳ [Kotlin - Delete Recursively](#)

↳ [Kotlin - Get File Extension](#)

Kotlin Interview Q/A

↳ [Kotlin Interview Questions](#)

Kotlin Android

↳ [Kotlin Android Tutorial](#)

Useful Resources

↳ [How to Learn Programming](#)