



Kotlin

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Kotlin is a programming language introduced by JetBrains, the official designer of the most intelligent Java IDE, named IntelliJ IDEA. This is a strongly statically typed language that runs on JVM. In 2017, Google announced Kotlin is an official language for android development.

Kotlin is an open source programming language that combines object-oriented programming and functional features into a unique platform. The content is divided into various chapters that contain related topics with simple and useful examples.

Audience

This tutorial has been prepared for the beginners to help them understand the basics of Kotlin programming language. After completing this tutorial, you will find yourself at a moderate level of expertise in Kotlin, from where you can take yourself to the next levels.

Prerequisites

Before proceeding with this tutorial you should have a basic understanding of Java programming language. Although it is a beginners' tutorial, we assume that the readers have a reasonable exposure to any programming environment and knowledge of basic concepts such as variables, commands, syntax, etc. We strongly recommend that you gain some basic knowledge of Java programming language before proceeding with Kotlin programming.

Copyright & Disclaimer

© Copyright 2019 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience	i
Prerequisites	i
Copyright & Disclaimer	i
Table of Contents	ii
 1. KOTLIN - OVERVIEW	 1
Advantages & Disadvantages	1
 2. KOTLIN - ENVIRONMENT SETUP	 2
 3. KOTLIN - ARCHITECTURE.....	 7
 4. KOTLIN - BASIC TYPES	 8
Numbers.....	8
Characters	9
Boolean	10
Strings	10
Arrays.....	11
Collections.....	12
Ranges.....	13
 5. KOTLIN - CONTROL FLOW	 15
If - Else.....	15
Use of When.....	16
For Loop	17
While Loop and Do-While Loop	18
Use of Return, Break, Continue	20

6.	KOTLIN - CLASS & OBJECT	23
	Nested Class	24
	Inner Class	25
	Anonymous Inner Class	26
	Type Aliases	27
7.	KOTLIN - CONSTRUCTORS	28
8.	KOTLIN - INHERITANCE	31
9.	KOTLIN - INTERFACE	34
10.	KOTLIN - VISIBILITY CONTROL	37
11.	KOTLIN - EXTENSION	39
	Function Extension	39
	Object Extension	40
12.	KOTLIN - DATA CLASSES	42
13.	KOTLIN - SEALED CLASS	43
14.	KOTLIN - GENERICS	44
15.	KOTLIN - DELEGATION	46
	Property Delegation	47
16.	KOTLIN - FUNCTIONS	50
	Lambda Function	51
	Inline Function	51
17.	KOTLIN - DESTRUCTURING DECLARATIONS	53
18.	KOTLIN - EXCEPTION HANDLING	54

1. Kotlin - Overview

Kotlin is a new open source programming language like Java, JavaScript, etc. It is a high level strongly statically typed language that combines functional and technical part in a same place. Currently, Kotlin targets Java and JavaScript. It runs on JVM.

Kotlin is influenced by other programming languages such as Java, Scala, Groovy, Gosu, etc. The syntax of Kotlin may not be exactly similar to JAVA, however, internally Kotlin is reliant on the existing Java Class library to produce wonderful results for the programmers. Kotlin provides interoperability, code safety, and clarity to the developers around the world.

Advantages & Disadvantages

Following are some of the advantages of using Kotlin for your application development.

Easy Language: Kotlin is a functional language and very easy to learn. The syntax is pretty much similar to Java, hence it is very easy to remember. Kotlin is more expressive, which makes your code more readable and understandable.

Concise: Kotlin is based on JVM and it is a functional language. Thus, it reduce lots of boiler plate code used in other programming languages.

Runtime and Performance: Better performance and small runtime.

Interoperability: Kotlin is mature enough to build an interoperable application in a less complex manner.

Brand New: Kotlin is a brand new language that gives developers a fresh start. It is not a replacement of Java, though it is developed over JVM. It is accepted as the first official language of android development. Kotlin can be defined as - Kotlin= JAVA +extra updated new features.

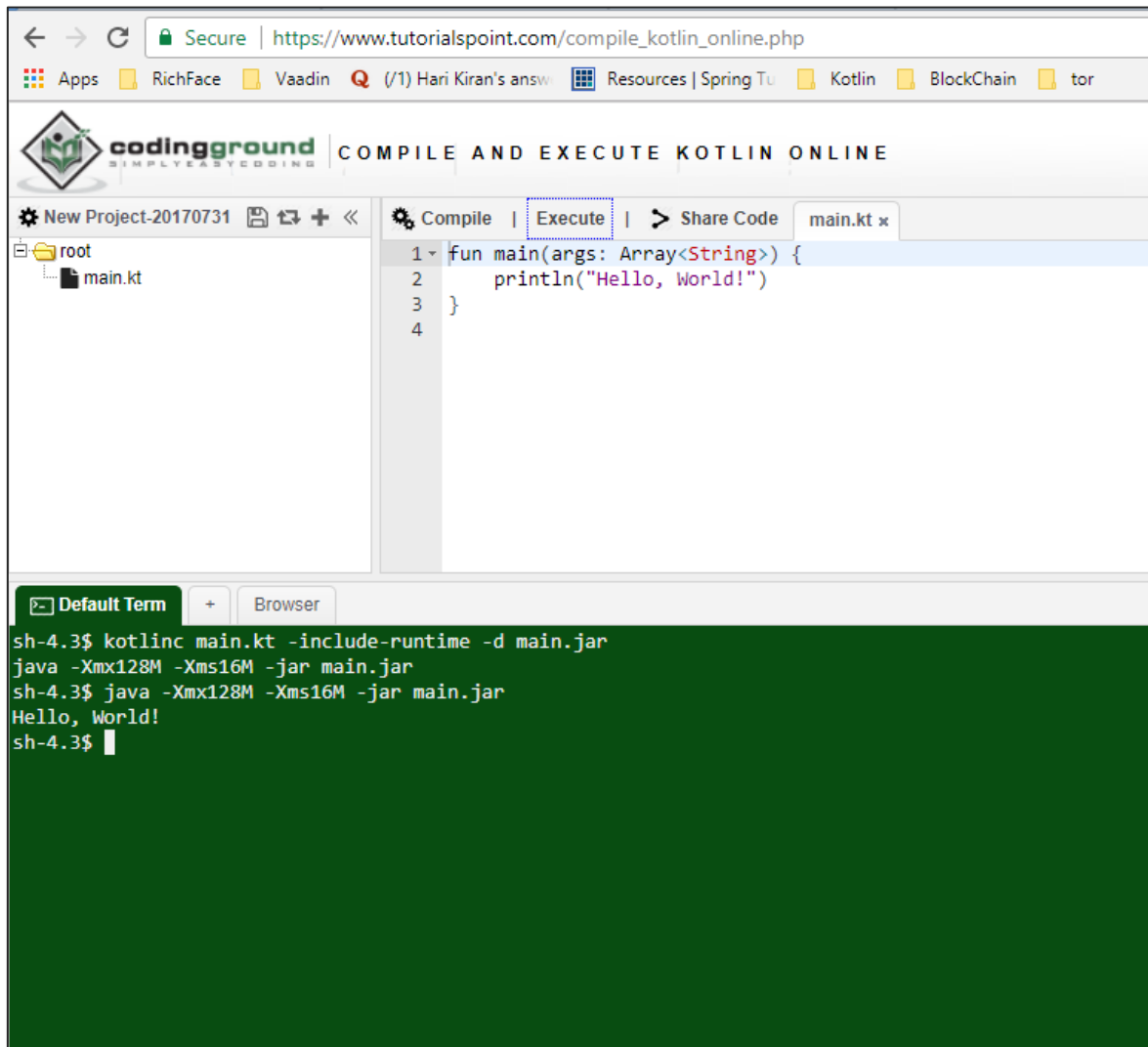
Following are some of the disadvantages of Kotlin.

Namespace declaration: Kotlin allows developers to declare the functions at the top level. However, whenever the same function is declared in many places of your application, then it is hard to understand which function is being called.

No Static Declaration: Kotlin does not have usual static handling modifier like Java, which can cause some problem to the conventional Java developer.

2. Kotlin - Environment Setup

In this chapter, we will set up our local environment to start with Kotlin programming. However, we already have set up Kotlin environment online (coding ground), so that compilation and execution of all available examples online can be easier. This gives confidence as you can check the result with different options. Feel free to modify any example and execute it online. Following is the screenshot of our online coding ground.



The screenshot shows a web browser window with the URL https://www.tutorialspoint.com/compile_kotlin_online.php. The page title is "COMPILE AND EXECUTE KOTLIN ONLINE". The interface includes a file explorer on the left showing a project named "New Project-20170731" with a file "main.kt". The main editor displays the following Kotlin code:

```
1 fun main(args: Array<String>) {  
2     println("Hello, World!")  
3 }  
4
```

Below the editor is a terminal window with the following commands and output:

```
sh-4.3$ kotlinc main.kt -include-runtime -d main.jar  
java -Xmx128M -Xms16M -jar main.jar  
sh-4.3$ java -Xmx128M -Xms16M -jar main.jar  
Hello, World!  
sh-4.3$
```

However, if you still want to use Kotlin offline in your local system, then you need to execute the following steps to configure your local workspace.

Step 1: Java 8 installation.

Kotlin runs on JVM, hence. it is really necessary to use JDK 8 for your local Kotlin development. Please refer to the official website of [oracle](https://www.oracle.com/in/java/technologies/javase-downloads.html) to download and install JDK 8 or an above version. You might have to set the environment variable for JAVA such that it can work properly. To verify your installation in Windows operating system, hit "java -version" in the command prompt and as an output it will show you the java version installed in your system.

Step 2: IDE installation.

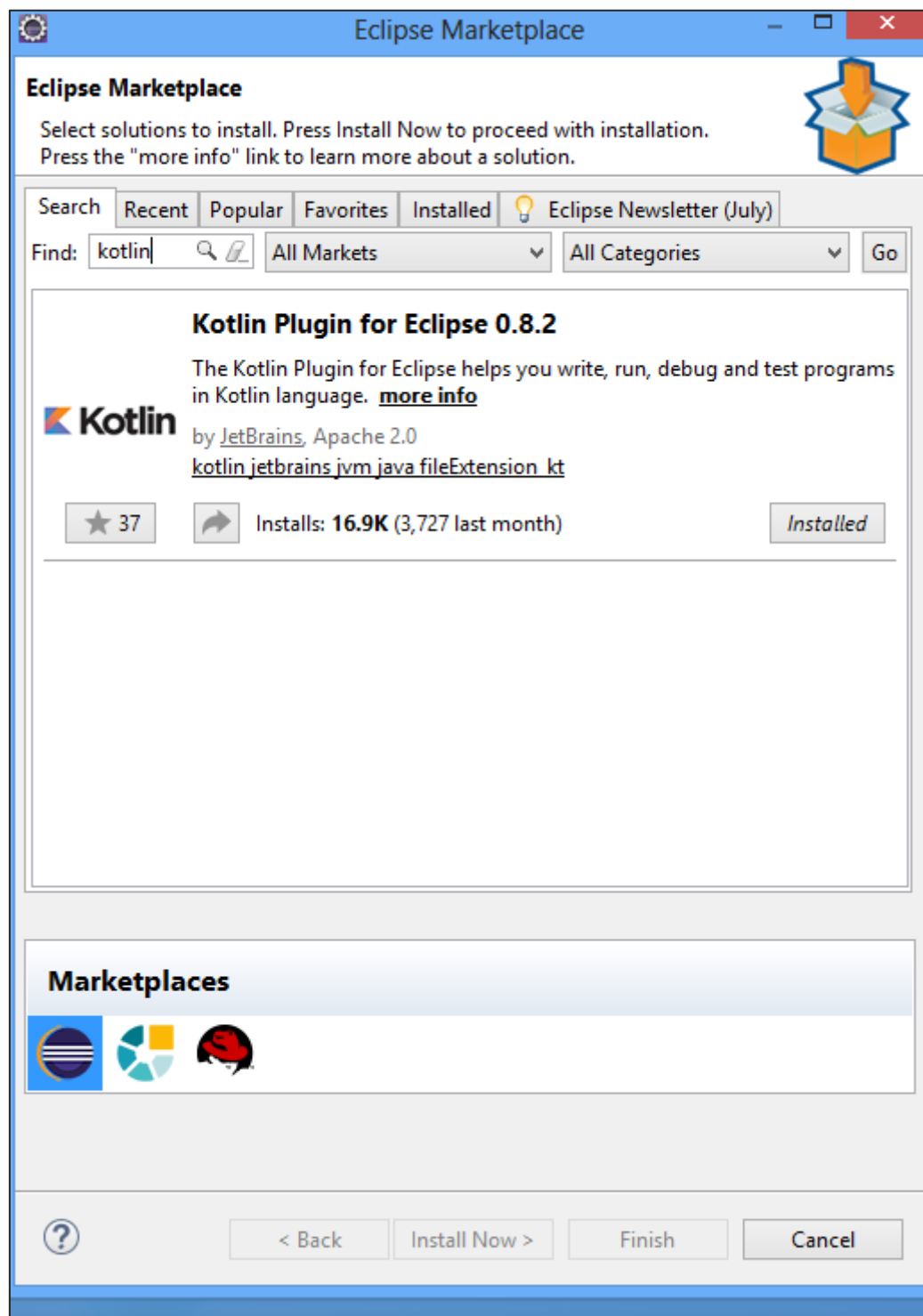
There are a number of IDE available over the internet. You can use any of your choice. You can find the download link of different IDE in the following table.

IDE Name	Installation Link
NetBeans	https://netbeans.org/downloads/
Eclipse	https://www.eclipse.org/downloads/
IntelliJ	https://www.jetbrains.com/idea/download/#section=windows

It is always recommended to use the recent software version to drag out maximum facility from it.

Step 3: Configuring Eclipse.

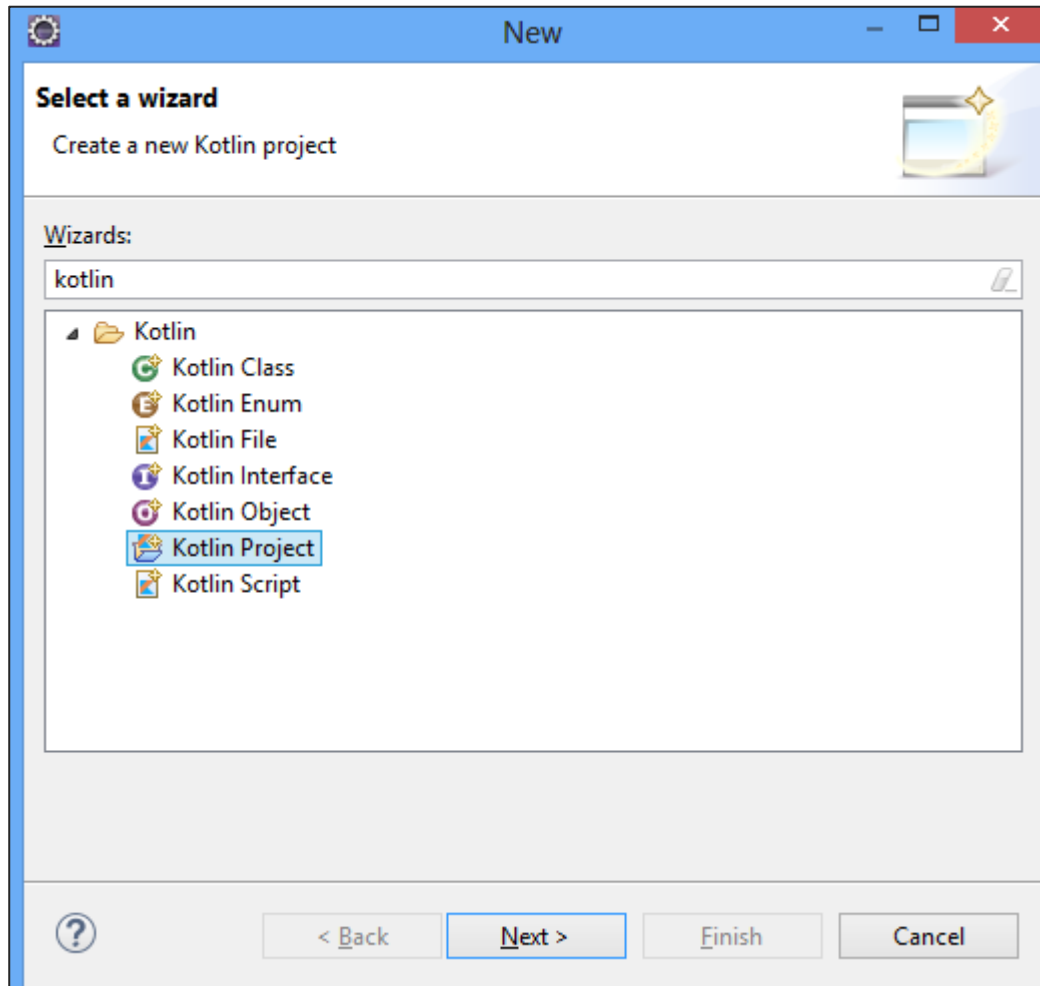
Open Eclipse and go to "Eclipse Market Place". You will find the following screen.



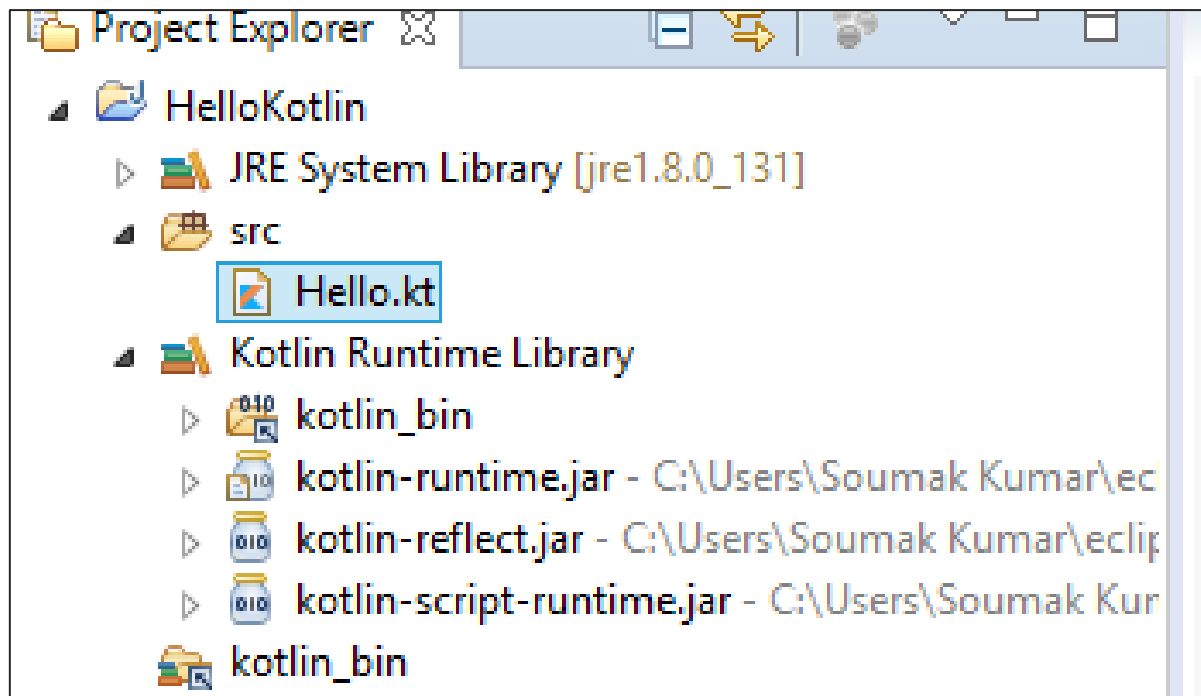
Search for Kotlin in the search box and install the same in your local system. It might take some time depending on the internet speed. You may have to restart your Eclipse, once it is successfully installed.

Step 4: Kotlin Project.

Once Eclipse is successfully restarted and Kotlin is installed, you will be able to create a Kotlin project on the fly. Go to **File** → **New** → **Others** and select "Kotlin project" from the list.



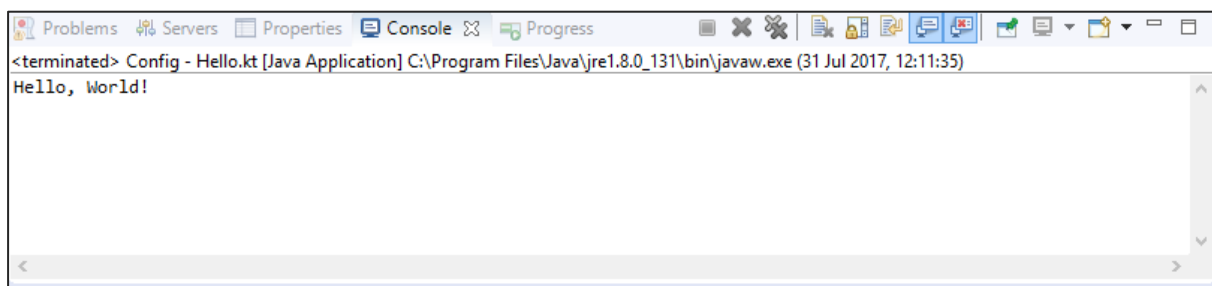
Once the project setup is done, you can create a Kotlin file under "SRC" folder. Left-click on the "Src" folder and hit "new". You will get an option for Kotlin file, otherwise you may have to search from the "others". Once the new file is created, your project directory will be look like the following.



Your development environment is ready now. Go ahead and add the following piece of code in the "Hello.kt" file .

```
fun main(args: Array<String>) {  
    println("Hello, World!")  
}
```

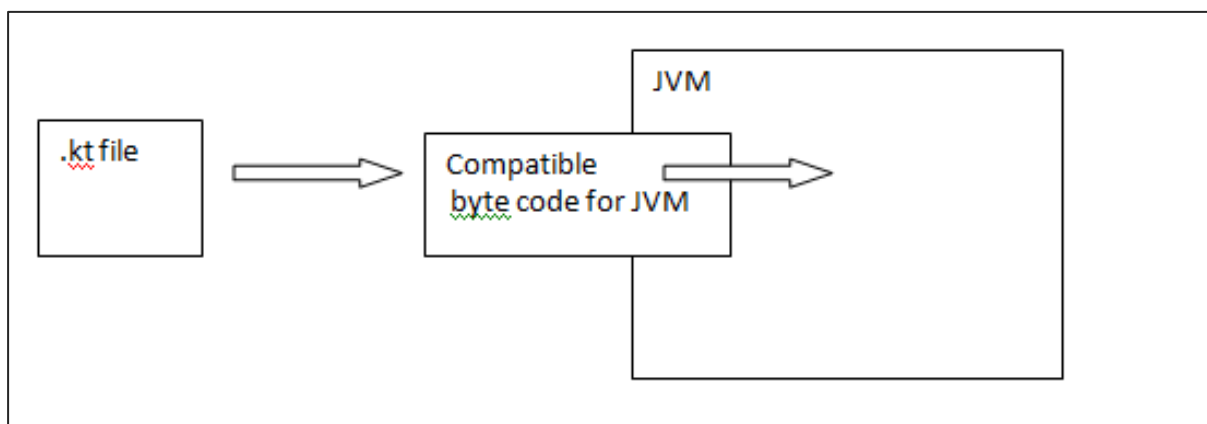
Run it as a Kotlin application and see the output in the console as shown in the following screenshot. For better understanding and availability, we will be using our coding ground tool.



3. Kotlin - Architecture

Kotlin is a programming language and has its own architecture to allocate memory and produce a quality output to the end user. Following are the different scenarios where Kotlin compiler will work differently, whenever it is targeting different other kind of languages such as Java and JavaScript.

Kotlin compiler creates a byte code and that byte code can run on the JVM, which is exactly equal to the byte code generated by the Java **.class** file. Whenever two byte coded file runs on the JVM, they can communicate with each other and this is how an interoperable feature is established in Kotlin for Java.



Whenever Kotlin targets JavaScript, the Kotlin compiler converts the **.kt** file into ES5.1 and generates a compatible code for JavaScript. Kotlin compiler is capable of creating platform basis compatible codes via LLVM.

End of ebook preview

If you liked what you saw...

Buy it from our store @ **<https://store.tutorialspoint.com>**