

Kotlin Enum

Kotlin Enum Class

Kotlin Enum – Enum is a special data type that allows a variable to hold a value only from a set of predefined constants.

In this tutorial, we shall learn about Kotlin Enum Class : Syntax, Initialization, how enum classes are different from regular classes. An example program is provided to demonstrate the usage of Kotlin Enum.

Syntax

Following is the syntax to define an Enum class in Kotlin

```
enum class
<enum_class_name> {

enum class <enum_class_name> {
    constant1,
    constant2,
    constant3
}
```

Any number of constants could be specified inside the class separated by comma. Each Enum constant is implicitly an Object of type <enum_class_name>.

Following is an example of defining a Kotlin Enum class “MobileColor” with constants GOLD, SILVER, WHITE, BLACK and RED.

```
enum class MobileColor
{

enum class MobileColor {
    GOLD, SILVER, WHITE, BLACK, RED
}
```

Kotlin Enum Initialization

As Kotlin Enum is an object of an Enum class, these enum objects could be initialized. We shall use the same example of MobileColor enum class to demonstrate the initialization of enums.

```
enum class
MobileColor(val value:
```

```
enum class MobileColor(val value: Int) {  
    GOLD(0xffd323),  
    SILVER(0xaeaeea),  
    WHITE(0xffffffff),  
    BLACK(0x000000),  
    RED(0xFF0000)  
}
```

enum class **MobileColor**(**val value: Int**)

GOLD(0xffd323)

Please note that we have provided `val` in the definition, since the value has to be a constant.

Example – Kotlin Enum Class

Following Kotlin Application demonstrates the usage of enum class MobileColor.

We have initialized a [Data Class](#), Mobile, which has a value named color that can accept an Enum object of Enum class type MobileColor.

KotlinEnum.kt

```
/**  
 * Kotlin Example to
```

```

/**
 * Kotlin Example to Enum Classes in Kotlin
 */
fun main(args: Array<String>) {
    val mobile1: Mobile = Mobile("IPhone",MobileColor.GOLD)
    val mobile2: Mobile = Mobile("SONY",MobileColor.BLACK)
    // access enum variables
    println("The color of my "+mobile1.name+" is "+mobile1.color)
    println("The color of my "+mobile2.name+" is "+mobile2.color)
    // access the value of the variable in Enum Object
    println(mobile1.color.toString() + " value is "+mobile1.color.value)
    println(mobile2.color.toString() + " value is "+mobile2.color.value)
}
data class Mobile(val name:String, val color: MobileColor)
enum class MobileColor(val value: Int) {
    GOLD(0xffd323),
    SILVER(0xaeaeea),
    WHITE(0xffffff),
    BLACK(0x000000),
    RED(0xFF0000)
}

```

Console Output

```

The color of my iPhone
is GOLD

```

```

The color of my iPhone is GOLD
The color of my SONY is BLACK
GOLD value is 16765731
BLACK value is 0

```

Now let us try the same example, without having any value for the Enum constants.

KotlinEnum.kt

```

/**
 * Kotlin Example to

```

```

/**
 * Kotlin Example to Enum Classes in Kotlin
 */
fun main(args: Array<String>) {
    val mobile1 = Mobile("IPhone",MobileColor.GOLD)
    val mobile2 = Mobile("SONY",MobileColor.BLACK)

    // access enum variables
    println("The color of my "+mobile1.name+" is "+mobile1.color)
    println("The color of my "+mobile2.name+" is "+mobile2.color)
}

data class Mobile(val name:String, val color: MobileColor)

enum class MobileColor {
    GOLD,
    SILVER,
    WHITE,
    BLACK,
    RED
}

```

We have not given any value initialized for Enum constants.

Now, run this application and you shall see the following output in Console.

Console Output

```

The color of my iPhone
is GOLD

```

```

The color of my iPhone is GOLD
The color of my SONY is BLACK

Process finished with exit code 0

```

How enum class is different from a regular class

You cannot create new instances of an Enum class outside of its definition. In other words outside enum definition, you cannot create an object of type Enum Class by passing values to its primary constructor. You can do so for regular classes.

Conclusion

In this [Kotlin Tutorial – Kotlin Enum Classes](#), we have learnt the syntax and usage with help of an Example program.

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

