Kotlin - Constructors

In this chapter, we will learn about constructors in Kotlin. Kotlin has two types of constructor - one is the **primary constructor** and the other is the **secondary constructor**. One Kotlin class can have one primary constructor, and one or more secondary constructor. Java constructor initializes the member variables, however, in Kotlin the primary constructor initializes the class, whereas the secondary constructor helps to include some extra logic while initializing the same. The primary constructor can be declared at class header level as shown in the following example.

```
class Person(val firstName: String, var age: Int) {
   // class body
}
```

In the above example, we have declared the primary constructor inside the parenthesis. Among the two fields, first name is read-only as it is declared as “val”, while the field age can be edited. In the following example, we will use the primary constructor.

Live Demo

```
fun main(args: Array<String>) {
   val person1 = Person("TutorialsPoint.com", 15)
   println("First Name = ${person1.firstName}")
   println("Age = ${person1.age}")
}
class Person(val firstName: String, var age: Int) {
}
```

The above piece of code will automatically initialize the two variables and provide the following output in the browser.

```
First Name = TutorialsPoint.com
Age = 15
```

As mentioned earlier, Kotlin allows to create one or more secondary constructors for your class. This secondary constructor is created using the “constructor” keyword. It is required whenever you want to create more than one constructor in Kotlin or whenever you want to include more logic in the primary constructor and you cannot do that because the primary constructor may be called by some other class. Take a look at the following example, where we have created a secondary constructor and are using the above example to implement the same.

Live Demo

```
fun main(args: Array<String>) {
```

```
   val HUman = HUman("TutorialsPoint.com", 25)
   print("${HUman.message}"+"${HUman.firstName}"+
      "Welcome to the example of Secondary  constructor, Your Age is-${HUman.age}")
}
class HUman(val firstName: String, var age: Int) {
   val message:String  = "Hey!!!"
        constructor(name : String , age :Int ,message :String):this(name,age) {
   }
}
```

Note − Any number of secondary constructors can be created, however, all of those constructors should call the primary constructor directly or indirectly.

The above piece of code will yield the following output in the browser.

```
Hey!!! TutorialsPoint.comWelcome to the example of Secondary  constructor, Your Age
```