



Kotlin

The Programming Language For Android

Srikanth Reddy

Android Developer, Technical Trainer KMIT



Week 1 : Session 2

Kotlin Variables and Data Types



Agenda

- What is a variable?
- Mutable variable: var keyword
- Immutable variable: val keyword
- Kotlin Data Types
 - Numbers
 - Boolean
 - Characters
 - Arrays
 - Strings



What is a variable?

- Variable is a name, given to a location in memory that can hold data.

```
1 package com.example.kotlindemo
2
3 ▶ fun main(){
4     var myName = "Srikanth"
5     print("Hello "+myName)
6 }
```

- **var** is a keyword which is used for declaring a variable.
- **myName** is an identifier(name of the variable).
- **“Srikanth”** is the data(value of the variable) and the type of variable is String

```
1 package com.example.kotlindemo
2
3 ▶ fun main(){
4     var myName: String = "Srikanth"
5     print("Hello "+myName)
6 }
```

Mutable variable: var keyword

- Mutable means can be changed.
- Use **var** keyword to declare a mutable variable

```
1 package com.example.kotlindemo
2
3 fun main(){
4     var myName: String = "Srikanth"
5     myName = "Scott"
6     print("Hello "+myName)
7 }
```

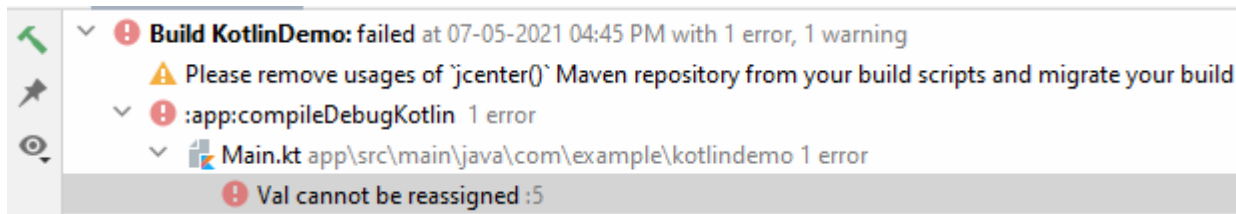
- We have reassigned a different value to **myName** variable.



Immutable variable: val keyword

- Immutable variable is one whose value cannot be changed.
- Also known as read-only variable.
- Immutable variable is declared using **val** keyword.

```
1 package com.example.kotlindemo
2
3 fun main(){
4     val myName: String = "Srikanth"
5     myName = "Scott"
6     print("Hello "+myName)
7 }
```





Kotlin Data Types

- Data Types are used to categorize a set of related values and define the operations that can be done on them.
- Basic data types used in Kotlin are:
 - **Numbers**
 - **Integers**
 - **Floating Point Numbers**
 - **Booleans** – true and false
 - **Characters** - 'a'
 - **Strings** – "abc"
 - **Arrays** – arrayOf()
- In Kotlin, everything (even the basic types like Int and Boolean) is an object. More specifically, everything behaves like an Object.

Numbers

- Numeric types in Kotlin are similar to any other programming language.
- They can be categorized into:
 - **Integers**
 - Byte - 8 bit
 - Short- 16 bit
 - Int - 32 bit
 - Long - 64 bit
 - **Floating Point Numbers**
 - Float - 32 bit single-precision floating point value.
 - Double - 64 bit double-precision floating point value.

```
1 package com.example.kotlindemo
2
3 fun main(){
4     // Kotlin Numeric Types Examples
5     val myByte: Byte = 10
6     val myShort: Short = 125
7
8     val myInt = 1000
9     val myLong = 1000L // The suffix 'L' is used to specify a long value
10
11     val myFloat = 126.78f // The suffix 'f' or 'F' represents a Float
12     val myDouble = 325.49
13
14     val hundredThousand = 100_000
15     val oneMillion = 1_000_000 // can also use underscore in numeric values
16
17     val myHexa = 0x0A0F // Hexadecimal values are prefixed with '0x' or '0X'
18     val myBinary = 0b1010 // Binary values are prefixed with '0b' or '0B'
19
20     print(myHexa)
21 }
```




Booleans

- The type Boolean is used to represent logical values
- It can have two possible values true and false.

```
1 package com.example.kotlindemo
2
3 ▶ fun main(){
4     // Kotlin Boolean Types Examples
5     val myBoolean: Boolean = true
6     val anotherBoolean = false
7
8     print(myBoolean)
9 }
```



Characters

- Characters are represented using the type Char.
- Char types cannot be treated as numbers they are objects.
- They are declared using single quotes

```
1 package com.example.kotlindemo
2
3 fun main(){
4     // Kotlin Character Types Examples
5     val letterChar: Char = 'A'
6     val digitChar = '9'
7
8     print(digitChar+"\n")
9     print(letterChar)
10 }
```

- Special characters in Kotlin are escaped using a backslash.
 - Escaped characters are - \n (newline), \t (tab), \r (carriage return), \b (backspace)

Strings

- Strings are represented using the String class.
- They are immutable, that means you cannot modify a String by changing some of its elements.

```
1 package com.example.kotlindemo
2
3 fun main(){
4     // Kotlin String Examples
5     var name = "Srikanth"
6     name = "Scott"
7     var firstCharInName = name[0] // 'S'
8     var lastCharInName = name[name.length - 1] // 't'
9
10    print("$firstCharInName\n$lastCharInName")
11 }
```

- Can access the character at a particular index in a String using `str[index]`...(index starts from zero).
- The `length` property is used to get the length of a String.



Arrays

- Arrays in Kotlin are represented using the Array class.
- Create an array in Kotlin either using
 - Library function `arrayOf()` and `Array()` constructor.

```
1 package com.example.kotlindemo
2
3 fun main(){
4     // Kotlin Arrays Examples
5     var numbers = arrayOf(1, 2, 3, 4, 5)
6     var animals = arrayOf("Cat", "Dog", "Lion", "Tiger")
7
8     var mixedArray = arrayOf(1, true, 3, "Hello", 'A') // Works and creates an array of Objects
9
10    //enforce a particular type
11    var numArray = arrayOf<Int>(1, 2, 3, 4)
12
13    var numArray1 = arrayOf<Int>(1, 2, 3, 4, "Hello") // Compiler Error
14
15    val firstElement = numbers[0]
16    val lastElement = numbers[numbers.size - 1]
17
18    print("$firstElement\n$lastElement")
19
20 }
```