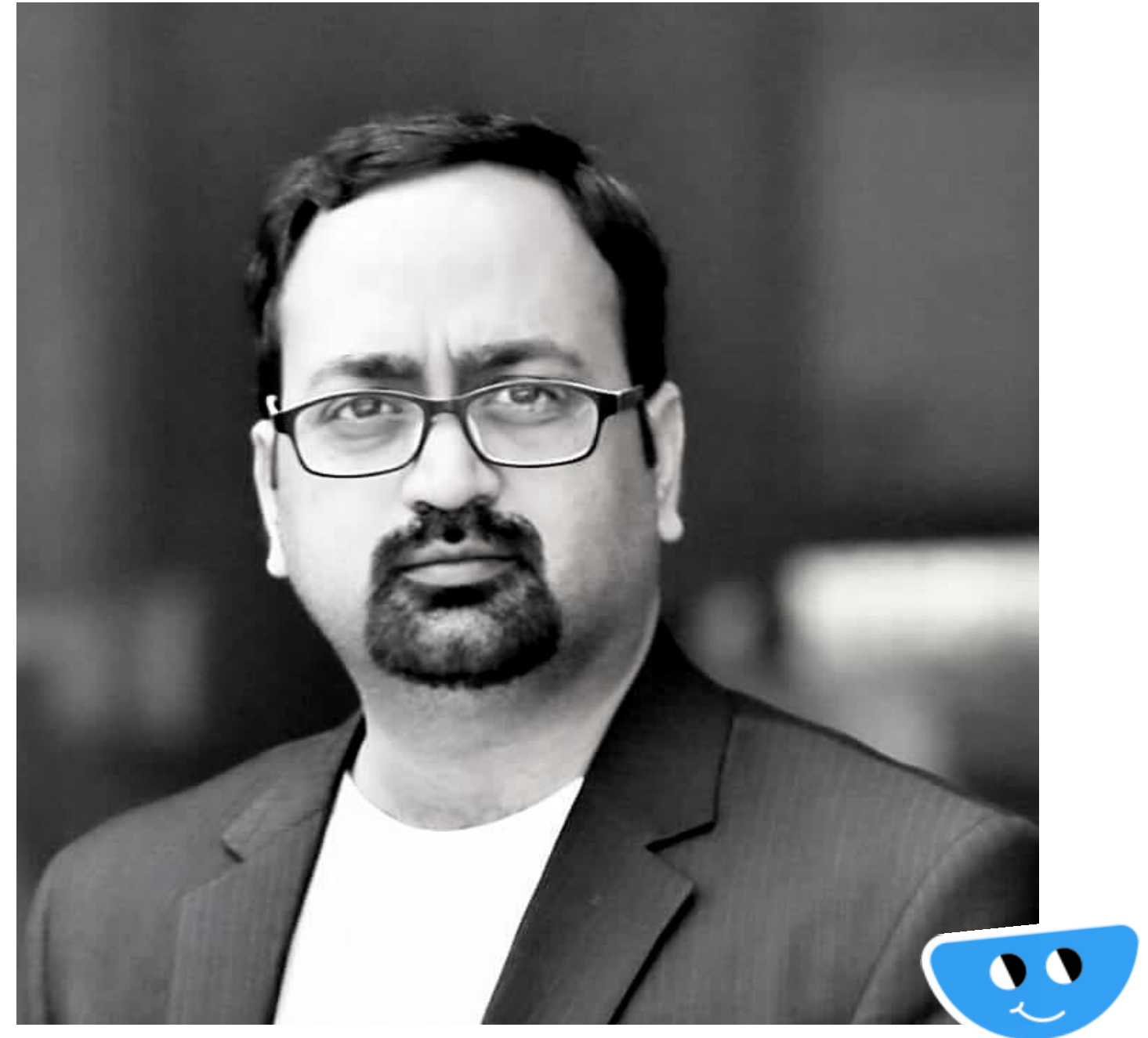# KOTLIN

A modern programming language that makes developers happier.

About Me

International Trainer and Author

- Member of NetBeans Dream Team
- Member of Google Train The Trainer Team
- 12 Awards till date
- 15 Years of Experience
- Developed Android Certification Module for UK based E-Learning company

Aatul Palandurkar

Today's Agenda

1 Introduction to the Session

2 History

3 Features

4 Architecture and Some Basic Syntax

5 Kotlin Vs Java

6 Conclusion

Let's Begin!

Are you ready?

It is JVM targeted language

20% faster compiler

Multi platform projects support

Kotlin is a statically-typed programming language

Introduction
Kotlin

It support object oriented programing and functional programing feature

Kotlin is an Open-Source Language.

It can also can be compiled to JavaScript source code

Kotlin also supports procedural programming with the use of functions

Its primary development is from a team of JetBrains

Kotlin is the Official Language of Android.

History

JetBrains hopes that the new language will drive IntelliJ IDEA sales.

1 In July 2011 JetBrains Start Project Kotlin, a new language for the JVM, which had been under development for a year.

2 Kotlin V 1.0 was released on February 15, 2016.

3 In May 2017, Google I/O Declared as the First class language for Android Application development

Features

Here are some key features of Kotlin programming language

Aatul Palandurkar | Kotlin for Beginners

Use Kotlin for any types of Enterprise Java EE development

100% compatible with all JVM frameworks

Null Safety in Kotlin. Kotlin avoids the null pointer exception

Kotlin compiles to JVM bytecode or JS to run on Node.JS or browser

Kotlin wants you to write less code

Higher order function

Features

Here are some some benefits of Kotlin programming language

Aatul Palandurkar | Kotlin for Beginners

Familiar Syntax

Kotlin programs can use all existing Java Frameworks and Libraries

Automatic conversion of Java to Kotlin

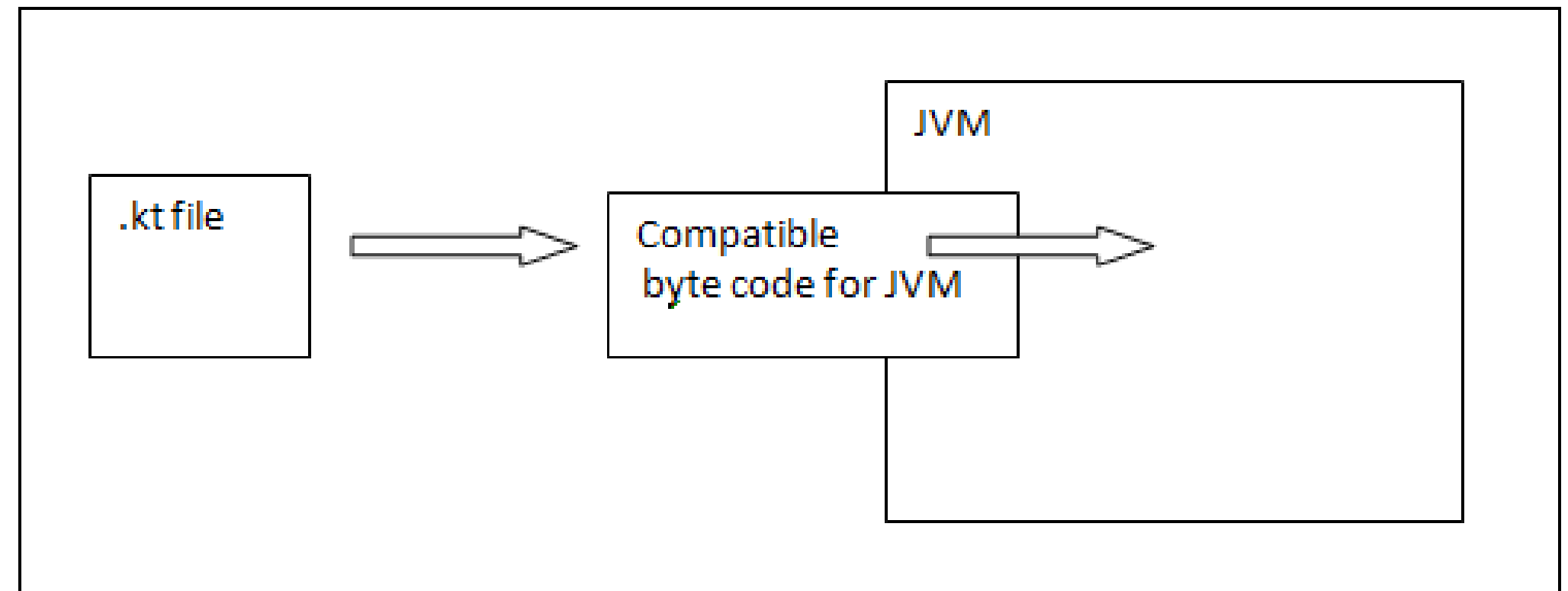Kotlin can be learned easily

Code reviews are not a problem

Kotlin has a strong community

Architecture

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.

① 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

② 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.

③ Whenever Kotlin targets JavaScript, the Kotlin compiler converts the .kt file into ES5.1 and generates a compatible code for JavaScript.

④ Kotlin compiler is capable of creating platform basis compatible codes via LLVM.

Kotlin

.kt file → Compatible byte code for JVM → JVM

Kotlin is a programming language and has its own architecture to allocate memory and produce a quality output to the end user.

Basic Syntax

Are you ready?

Variable initialization

```
/* Java Code */
static int num1, num2; //explicit declaration
num1 = 20; //use the variables anywhere
num2 = 30;


/* Kotlin Code */
val a: Int
val b: Int
a=5
b=10
```

Hello World in Kotlin

/* Kotlin Code*/

/* Simple **Hello Word Example**/

```kotlin
//optional package header
package hello

//Entry Level or main function
fun main() {
val scope = "world"
println("Hello, $scope!")
//semicolons are optional, have you noticed that? :)
}
```

No "new" keyword

```java
/* Java Code*/
Demo d = new Demo();
```

```kotlin
/* Kotlin Code*/
val d = Demo()
```

Null Pointer Exceptions

/* Kotlin Code*/

/* Simple **Hello Word Example*/**

```
//optional package header
package hello

//Entry Level or main function
fun main() {
val scope = "world"
println("Hello, $scope!")
//semicolons are optional, have you noticed that? :)
}
```

No Checked Exception

```
/* Java Code*/
try
{
Thread.sleep(1000);
// This statement throws an checked exception
}
catch(Exception e)
{
// some code here
}


/* Kotlin has no checked exception */
```

Type Interface

```
/* Java Code*/
int a = 10;
String name = "Aatul"
Demo d;

/* Kotlin Code */
var a = 1
var b = "Aatul"
var d: Demo
```

Lines of Code / Code Size

/* Java Code*/
```
class Person
{
String firstName;
Person(String firstName)
{
this.firstName = firstName;
}
}
```

/* Kotlin Code */
```
class Person(firstName: String)
```

Data Classes / Getter-Setter

1. To hold the data we must make simple classes with getter and setter method in data

2. In Kotlin, they provide the concept of data classes to hold the data

3. To define an data class in Kotlin we use the keyword data before the class keyword

```java
/* Java */
public class Address {
    private String city;
    private Country country;
    public Address(String city, Country country) {
        this.city = city;
        this.country = country;
    }
    public String getCity() {
        return city;
    }
    public void setCity(String city) {
        this.city = city;
    }
    public Country getCountry() {
        return country;
    }
    public void setCountry(Country country) {
        this.country = country;
    }
}
```

Data Classes / Getter–Setter

1 To hold the data we must make simple classes with getter and setter method in data

2 In Kotlin, they provide the concept of data classes to hold the data

3 To define an data class in Kotlin we use the keyword data before the class keyword

/* Kotlin Code */

```
data class Address(var city:String, var country:Country)
```

Null Safety

```kotlin
/* Kotlin Code */
val name: String? = null     //assigned null and it will compile also.

fun getName() : String? = null     //returned null and it  will compile too.
/* won't compile */
 val name: String? = null
 val len = name.length

/* correct way */
 val name: String? = null
 val len = name?.length

 // "?" Safe Call Operator
```

Smart Casts

```
/* Kotlin Code */
fun demo(x:Any){
if(x is String){
print(x.length)  // x is automatically cast to string
}
}
```

Conclusion

1 Kotlin is more expressive than Java and it we need to write less code

2 In future Kotlin may come as Java server side and mobile application with features like more declarative, less code, mixed language database and a lot.

Thank you!

Have a great day ahead.