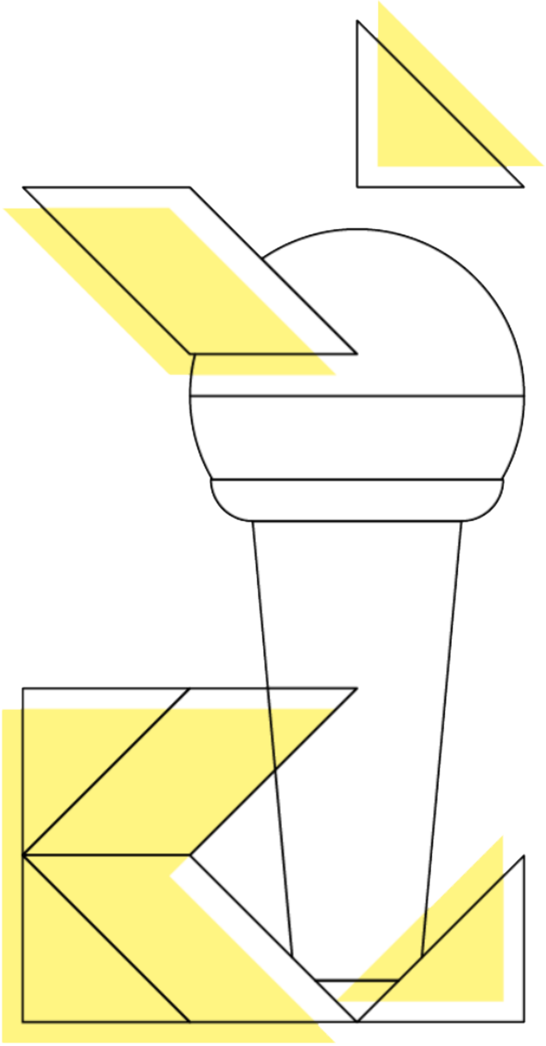
KOTLIN / Everywhere
Turkey KUG

WELCOME

Introduction to Kotlin

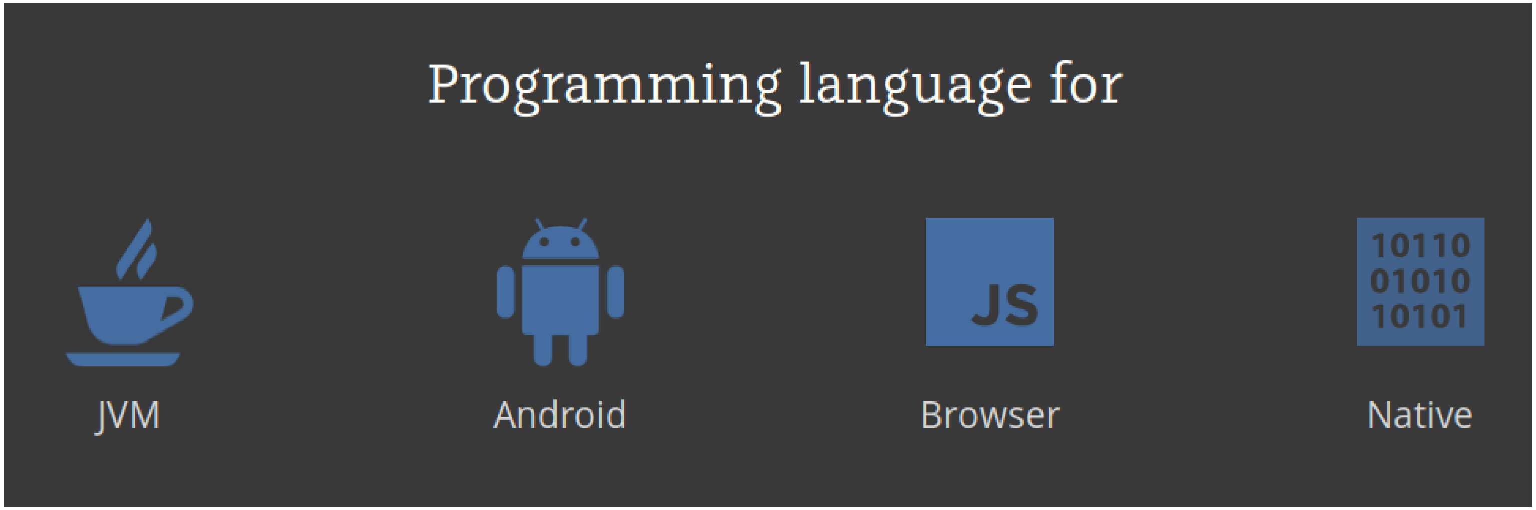
SPEAKER NAME
Mehmet Ali SICAK
CS Teacher, MEB



What is Kotlin?

- General-purpose
- Object Oriented Programming (OOP)
- Static Typing
- Open source (Apache 2.0)
- Developed by JetBrains

What is the purpose of Kotlin?



Programming language for

JVM

Android

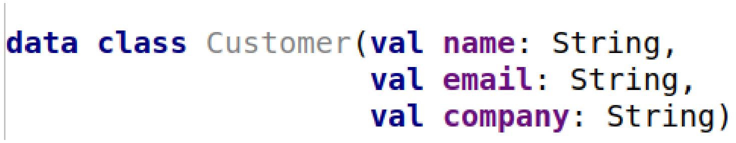
Browser

Native

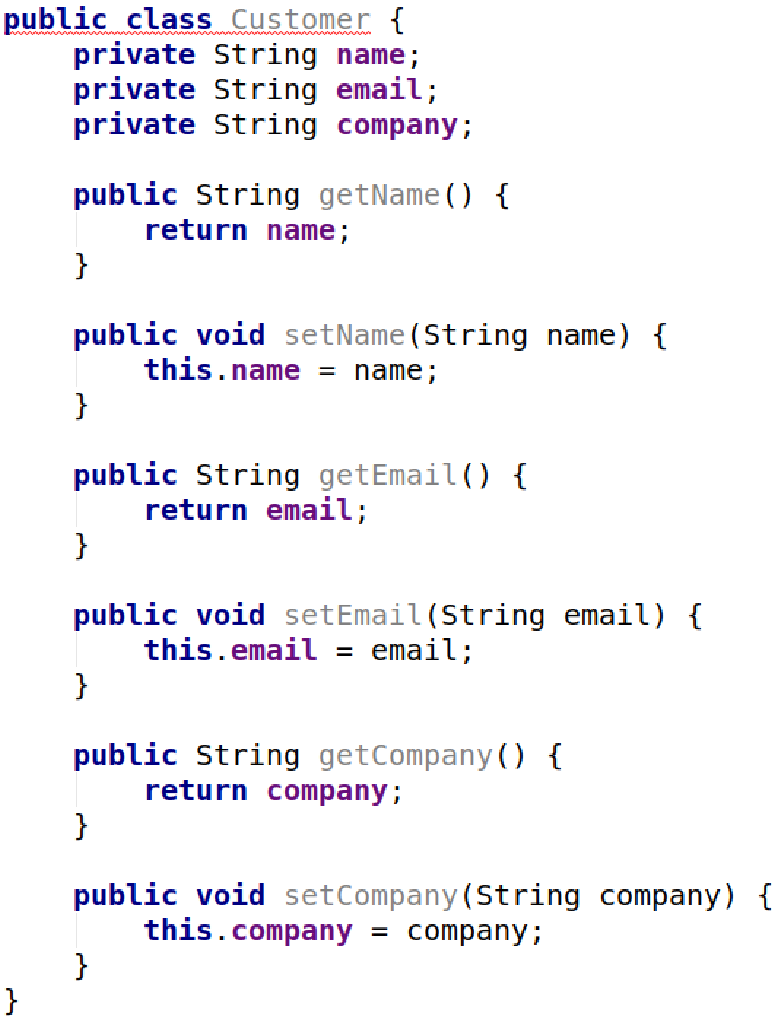
Why Kotlin?

- Concise
- Safe
- Interoperable
- Tool-friendly

Concise



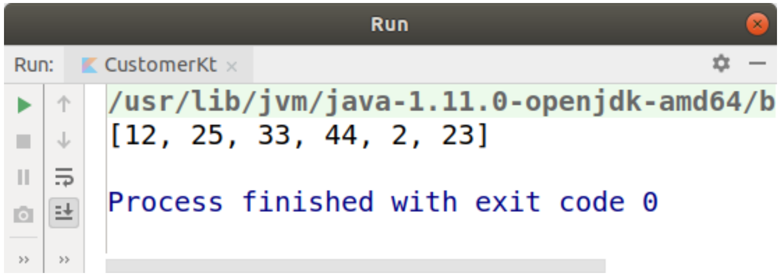
```
data class Customer(val name: String,
                    val email: String,
                    val company: String)
```



```
public class Customer {
    private String name;
    private String email;
    private String company;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getCompany() {
        return company;
    }

    public void setCompany(String company) {
        this.company = company;
    }
}
```

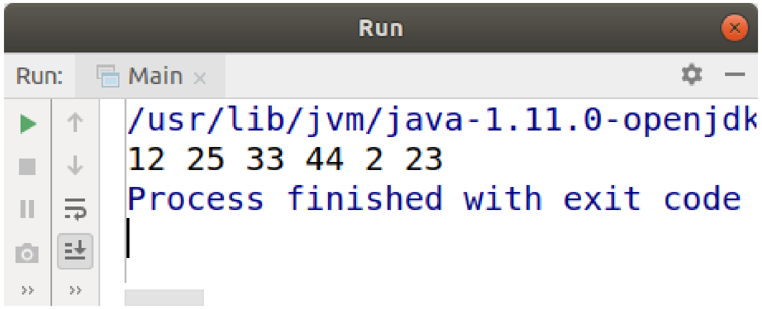
Concise

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

    var list = ArrayList<Int>()
    list.addAll(listOf(12,-9,25,-56,33,44,2,23,-1))

    val positiveNumbers = list.filter { it > 0 }

    println(positiveNumbers)

}
```



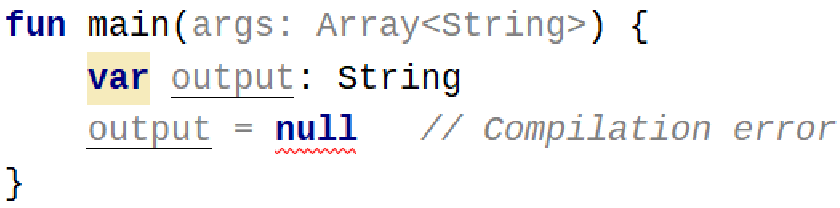
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/b
[12, 25, 33, 44, 2, 23]

Process finished with exit code 0
```

```
import java.util.ArrayList;
import java.util.Arrays;

public class Main {

    public static void main(String[] args) {
        ArrayList list = new ArrayList(Arrays.asList(12, -9, 25, -56, 33, 44, 2, 23, -1));

        for (int i = 0; i < list.size(); i++) {
            int j=Integer.parseInt(list.get(i).toString());
            if (j > 0) {
                System.out.print(j+" ");
            }
        }
    }
}
```

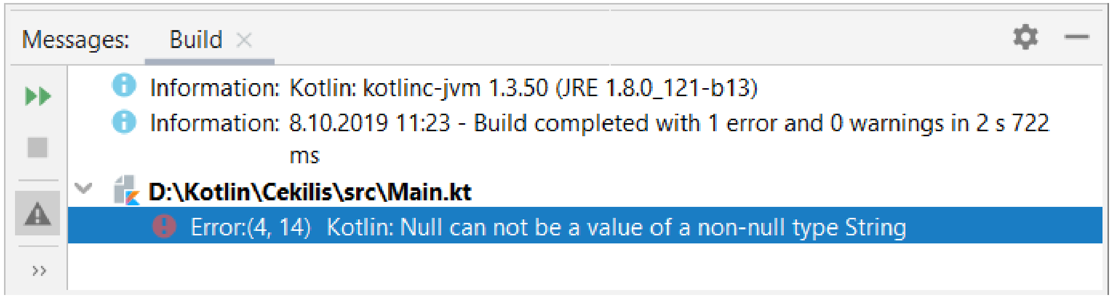


```
/usr/lib/jvm/java-1.11.0-openjdk
12 25 33 44 2 23
Process finished with exit code
```

Safe



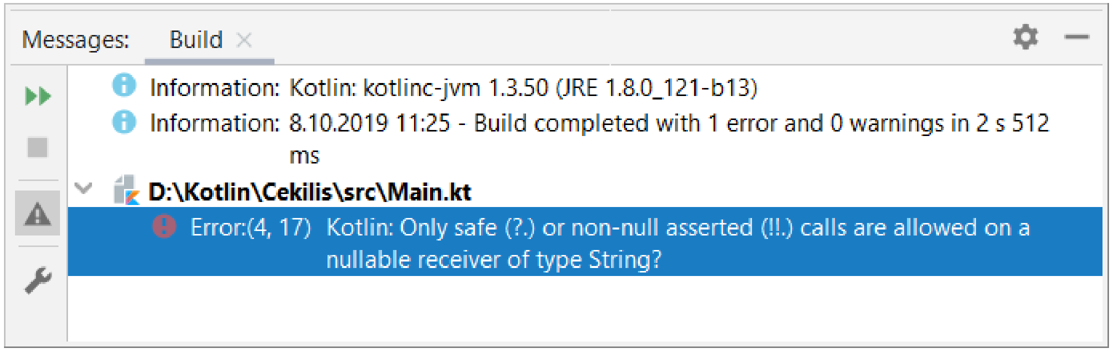
```kotlin
fun main(args: Array<String>) {
    var output: String
    output = null    // Compilation error
}
```



```
Information: Kotlin: kotlinc-jvm 1.3.50 (JRE 1.8.0_121-b13)
Information: 8.10.2019 11:23 - Build completed with 1 error and 0 warnings in 2 s 722
             ms
D:\Kotlin\Cekilis\src\Main.kt
    Error:(4, 14)  Kotlin: Null can not be a value of a non-null type String
```

Safe

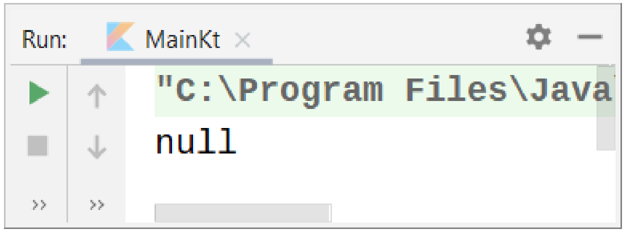
```
fun main(args: Array<String>) {
    val name: String? = null    // Nullable type
    println(name.length)      // Compilation error
}
```



```
Information: Kotlin: kotlinc-jvm 1.3.50 (JRE 1.8.0_121-b13)
Information: 8.10.2019 11:25 - Build completed with 1 error and 0 warnings in 2 s 512
             ms
D:\Kotlin\Cekilis\src\Main.kt
    Error:(4, 17)  Kotlin: Only safe (?.) or non-null asserted (!!.) calls are allowed on a
                   nullable receiver of type String?
```

Safe

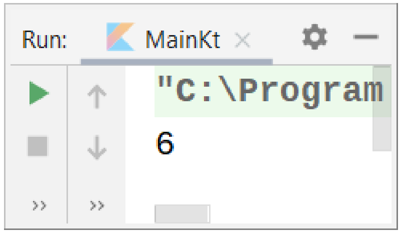
```
fun main(args: Array<String>) {
    val name: String? = null    // Nullable type
    println(name?.length)      // Safe calls
}
```



```
"C:\Program Files\Java
null
```

Safe

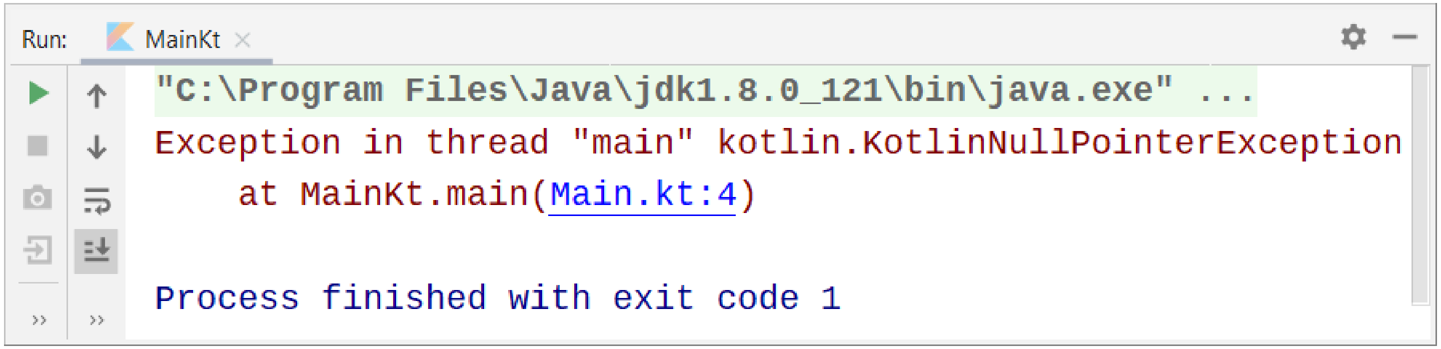
```
fun main(args: Array<String>) {
    val name: String? = "Kotlin"    // Nullable type
    println(name?.length)       // Safe calls
}
```



```
"C:\Program
6
```

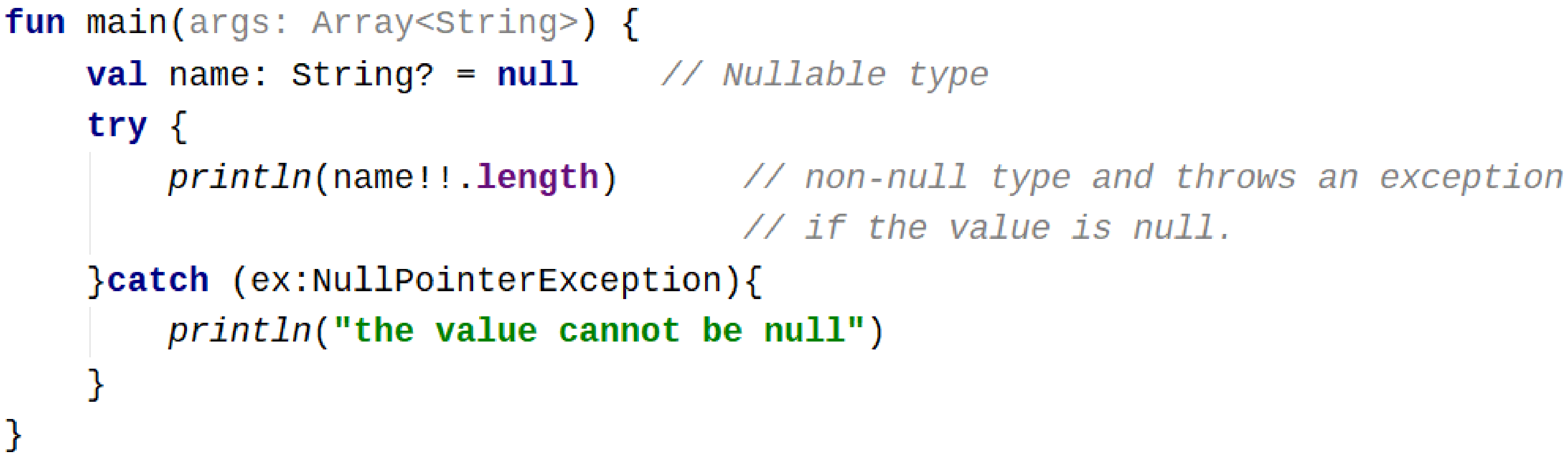
Safe

```
fun main(args: Array<String>) {
    val name: String? = null    // Nullable type
    println(name!!.length)      // non-null type and throws an exception
                                // if the value is null.
}
```

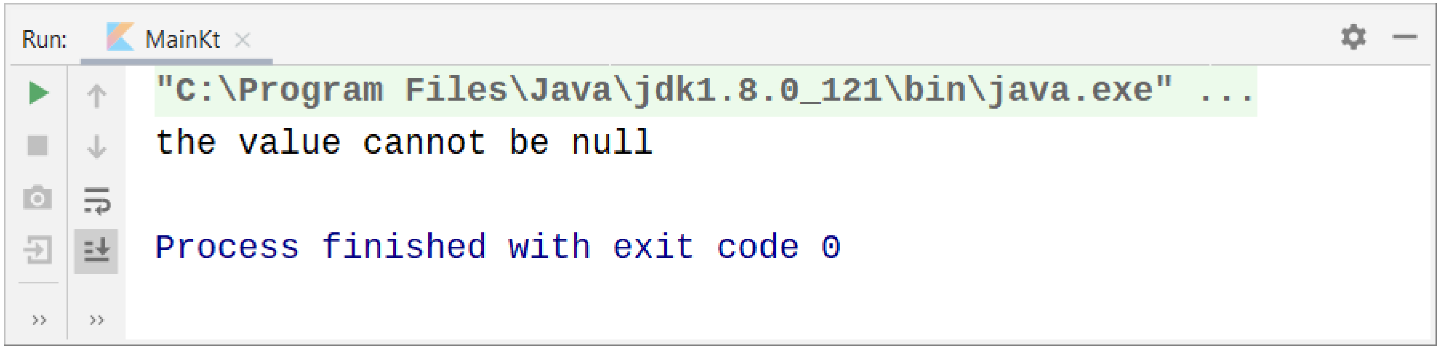


```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ...
Exception in thread "main" kotlin.KotlinNullPointerException
	at MainKt.main(Main.kt:4)

Process finished with exit code 1
```

Safe

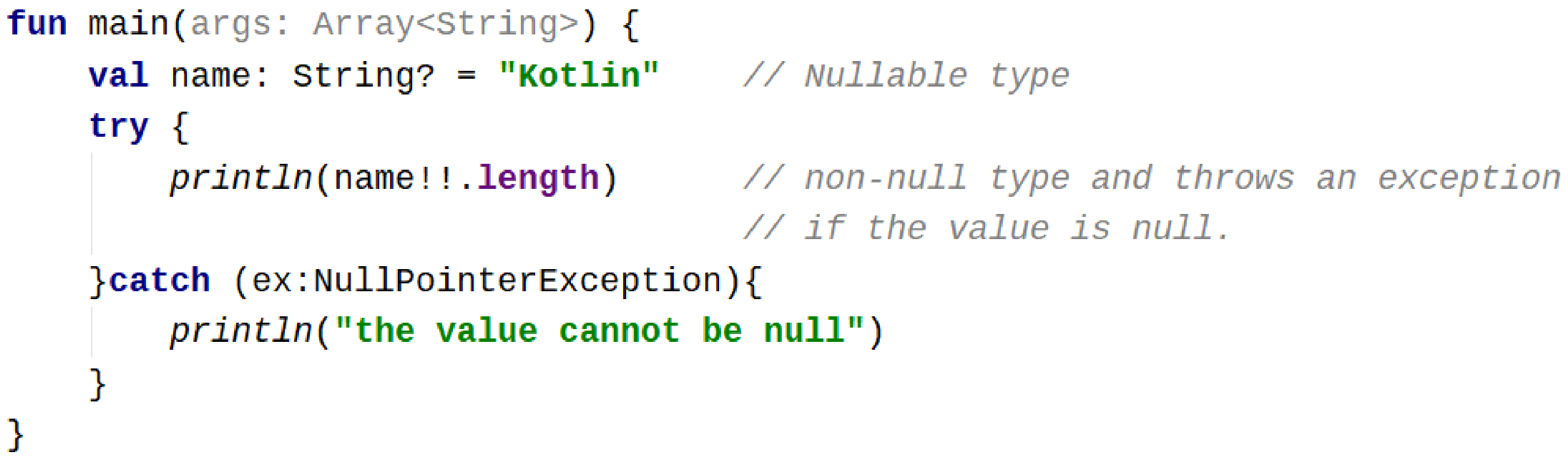


```kotlin
fun main(args: Array<String>) {
    val name: String? = null    // Nullable type
    try {
        println(name!!.length)      // non-null type and throws an exception
                                    // if the value is null.
    }catch (ex:NullPointerException){
        println("the value cannot be null")
    }
}
```

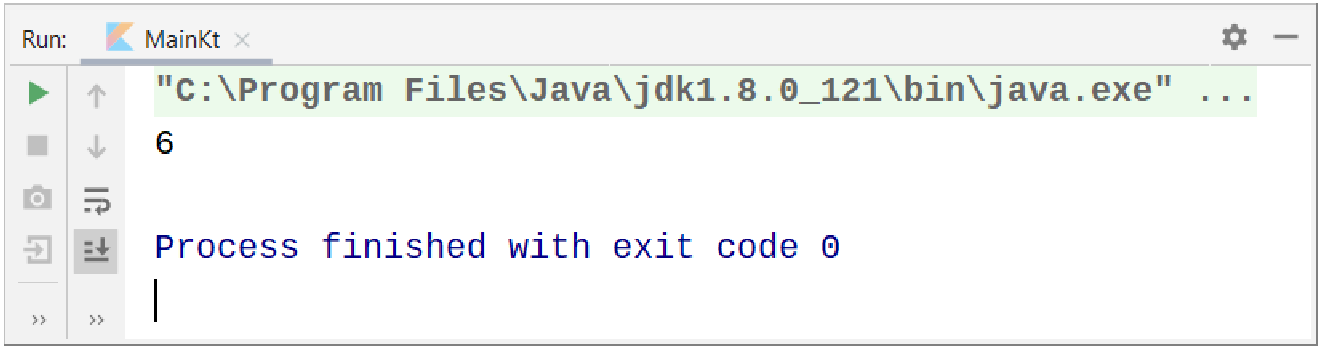


```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ...
the value cannot be null

Process finished with exit code 0
```

Safe



```
fun main(args: Array<String>) {
    val name: String? = "Kotlin"    // Nullable type
    try {
        println(name!!.length)      // non-null type and throws an exception
                                    // if the value is null.
    }catch (ex:NullPointerException){
        println("the value cannot be null")
    }
}
```



```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ...
6

Process finished with exit code 0
```

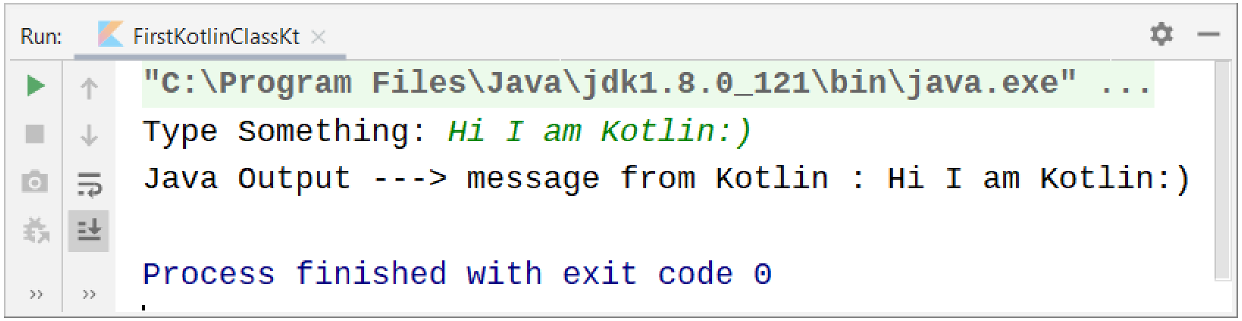
Interoperable

- Kotlin is designed with Java Interoperability in mind.
- Existing Java code can be called from Kotlin in a natural way,
- and Kotlin code can be used from Java rather smoothly as well.

JS
JVM
Android
Browser

Interoperable with Java

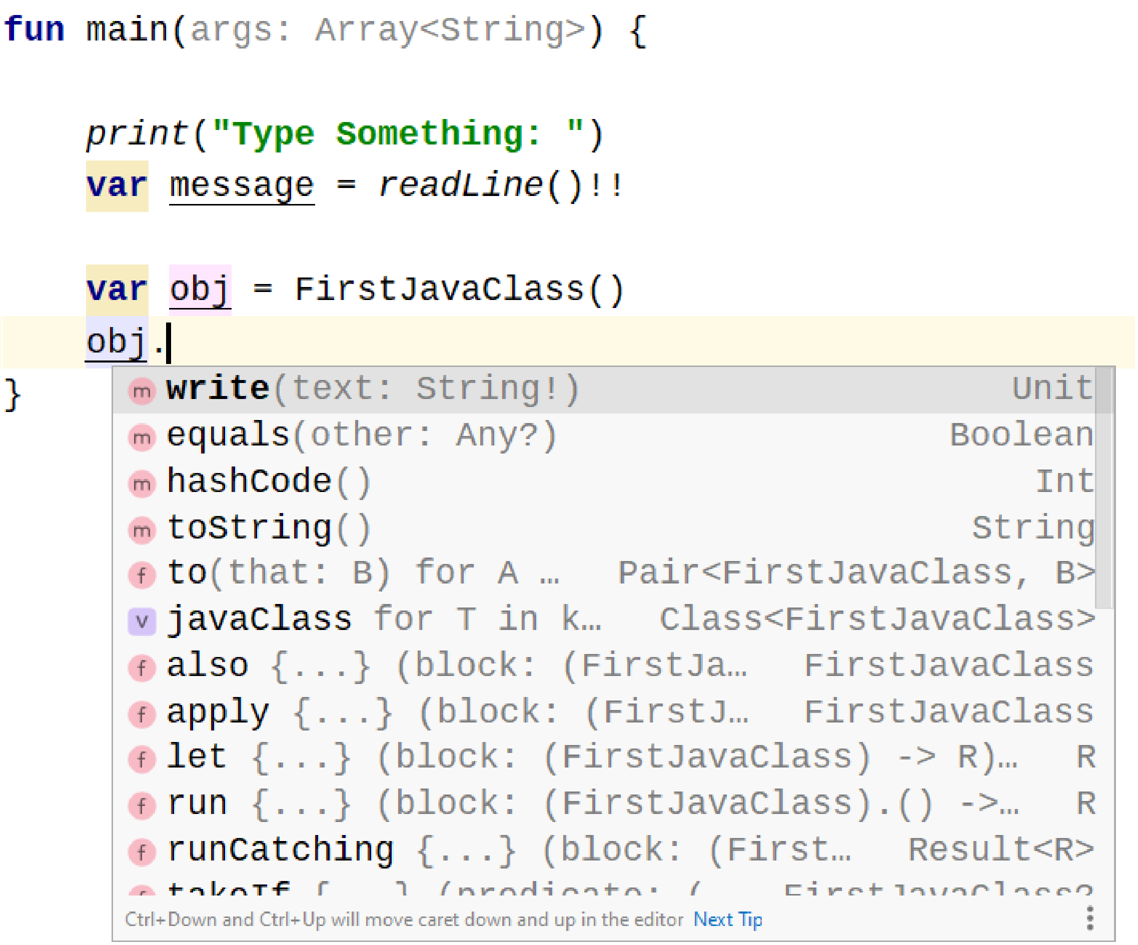
```kotlin
fun main(args: Array<String>) {
    print("Type Something")
    var message = readLine()!!

    var obj = FirstJavaClass()
    obj.write(message)
}
```



```
"C:\Program Files\Java\jdk1.8.0_121\bin\java.exe" ...
Type Something: Hi I am Kotlin:)
Java Output ---> message from Kotlin : Hi I am Kotlin:)

Process finished with exit code 0
```

```java
public class FirstJavaClass {
    public void write(String text) {
        System.out.println("Java Output ---> message from Kotlin : " + text);
    }
}
```

Tool-friendly



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

    print("Type Something: ")
    var message = readLine()!!

    var obj = FirstJavaClass()
    obj.
}
```

```
m write(text: String!)                              Unit
m equals(other: Any?)                            Boolean
m hashCode()                                         Int
m toString()                                      String
f to(that: B) for A …        Pair<FirstJavaClass, B>
v javaClass for T in k…       Class<FirstJavaClass>
f also {...} (block: (FirstJa…        FirstJavaClass
f apply {...} (block: (FirstJ…        FirstJavaClass
f let {...} (block: (FirstJavaClass) -> R)…        R
f run {...} (block: (FirstJavaClass).() ->…        R
f runCatching {...} (block: (First…        Result<R>
f takeIf {...} (predicate: (…      FirstJavaClass?
Ctrl+Down and Ctrl+Up will move caret down and up in the editor  Next Tip
```

Useful Links

- https://kotlinlang.org
- http://kotlinveandroid.com/
- https://try.kotlinlang.org
- https://try.kotlinlang.org/koans
- https://memetalisicak.com/category/kotlin

HAPPY HOUR

THANK YOU