

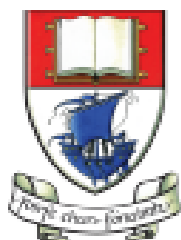


# An Introduction to Kotlin

---

Produced  
by:

Dr. Siobhán Drohan ([sdrohan@wit.ie](mailto:sdrohan@wit.ie))



Waterford Institute *of* Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics  
<http://www.wit.ie/>



“ Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less. At the core of the Java ecosystem is the JVM. ”

**James Gosling,**

Creator of the Java Programming Language(2011, TheServerSide)



# JVM Language History





Created in 2011 by [JetBrains](https://www.jetbrains.com/),  
the creators of IntelliJ.

On Object Oriented Language with  
Functional Constructs (e.g. lambdas).



# The next JVM language to learn

2014



10 Million Lines of code

Kotlin 1.1

2.1 Million Lines of code

Kotlin 1.0

2012

2013

2014

2015

2016

2017

*Many companies like Netflix, Uber, Trello, Pinterest, Corda etc. are using Kotlin (along with other programming languages) to create applications.*



April 21<sup>st</sup> 2017



*“Java is the primary and official language for Android development, but that doesn’t mean it is the best or the only choice.”*

*“To give Java credit, it is a robust language that has been around for a while, but it comes with a specific set of challenges that can be a deterrent for developers.”*







*“A new breed of modern JVM languages is slowly gaining traction within the Android community, and **Kotlin** is leading the pack.”*



**May 17<sup>th</sup> 2017**



GOOGLE DEVELOPERS

Hot from the [#io17](#) stage: Android is now officially supporting the Kotlin programming language, in addition to the Java language and C++.

<https://t.co/rsNtNZUTmK>



Twitter / May 17 - 9:10 pm IST



*“For Android developers, Kotlin support is a chance to use a modern and powerful language, helping solve common headaches such as runtime exceptions and source code verbosity.*

*Kotlin is easy to get started with and can be gradually introduced into existing projects, which means that your existing skills and technology investments are preserved.”*



# Kotlin Overview

---



**Kotlin**





# Kotlin

---

- runs on Java Virtual Machine.
- is an evolution of the Java syntax but is more concise and has cleaner syntax.
- is not syntax compatible with Java; but is interoperable with Java.
- relies on some Java Class Libraries e.g. Collections framework.
- is a statically-typed programming language.
- offers null safety.



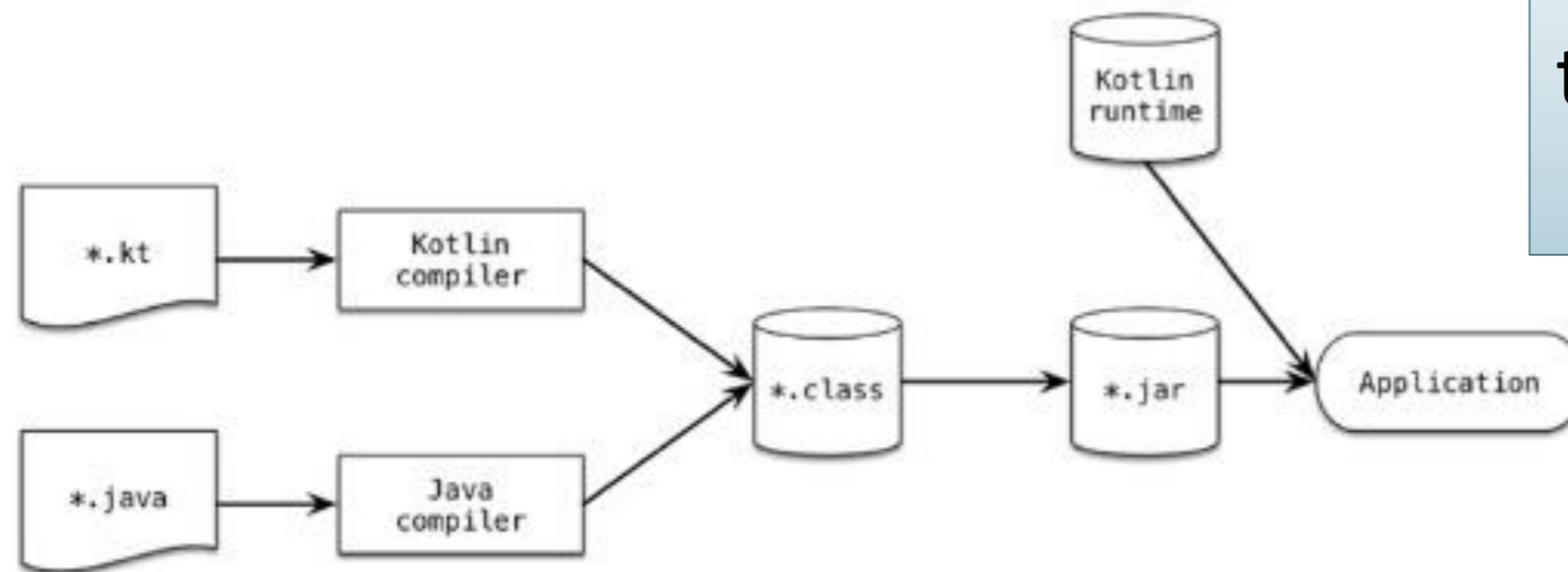
# Kotlin

---

- runs on Java Virtual Machine.
- is an evolution of the Java syntax but is more concise and has cleaner syntax.
- is not syntax compatible with Java; but is interoperable with Java.
- relies on some Java Class Libraries e.g. Collections framework.
- is a statically-typed programming language.
- offers null safety.

# Runs on Java Virtual Machine

## Kotlin and the JVM



\* Interoperable 100%

Kotlin compiles to JVM ByteCode (like Java)

*Note: Kotlin also compiles to JavaScript*

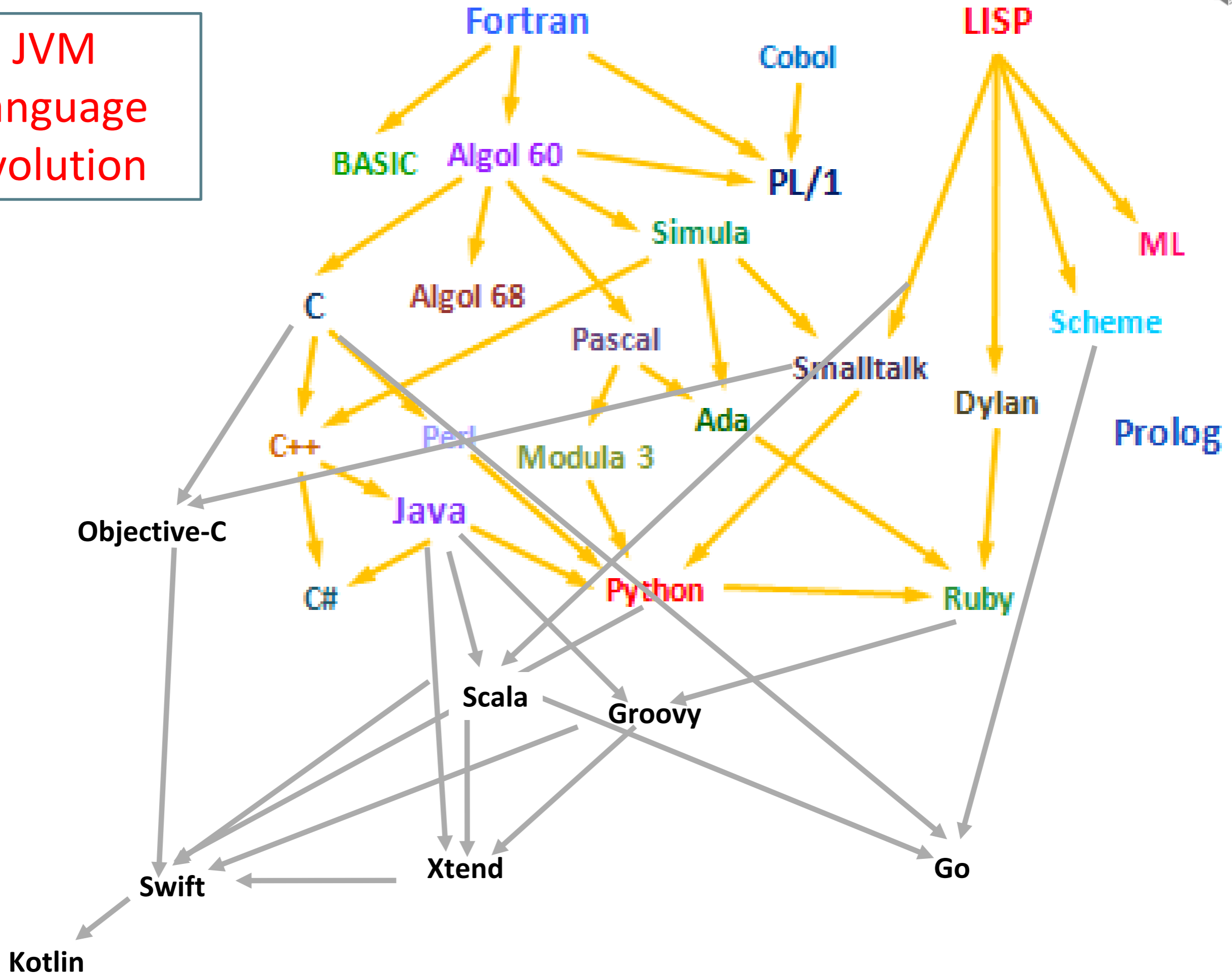




- runs on Java Virtual Machine.
- is an evolution of the Java syntax but is more concise and has cleaner syntax.
- is not syntax compatible with Java; but is interoperable with Java.
- relies on some Java Class Libraries e.g. Collections framework.
- is a statically-typed programming language.
- offers null safety.

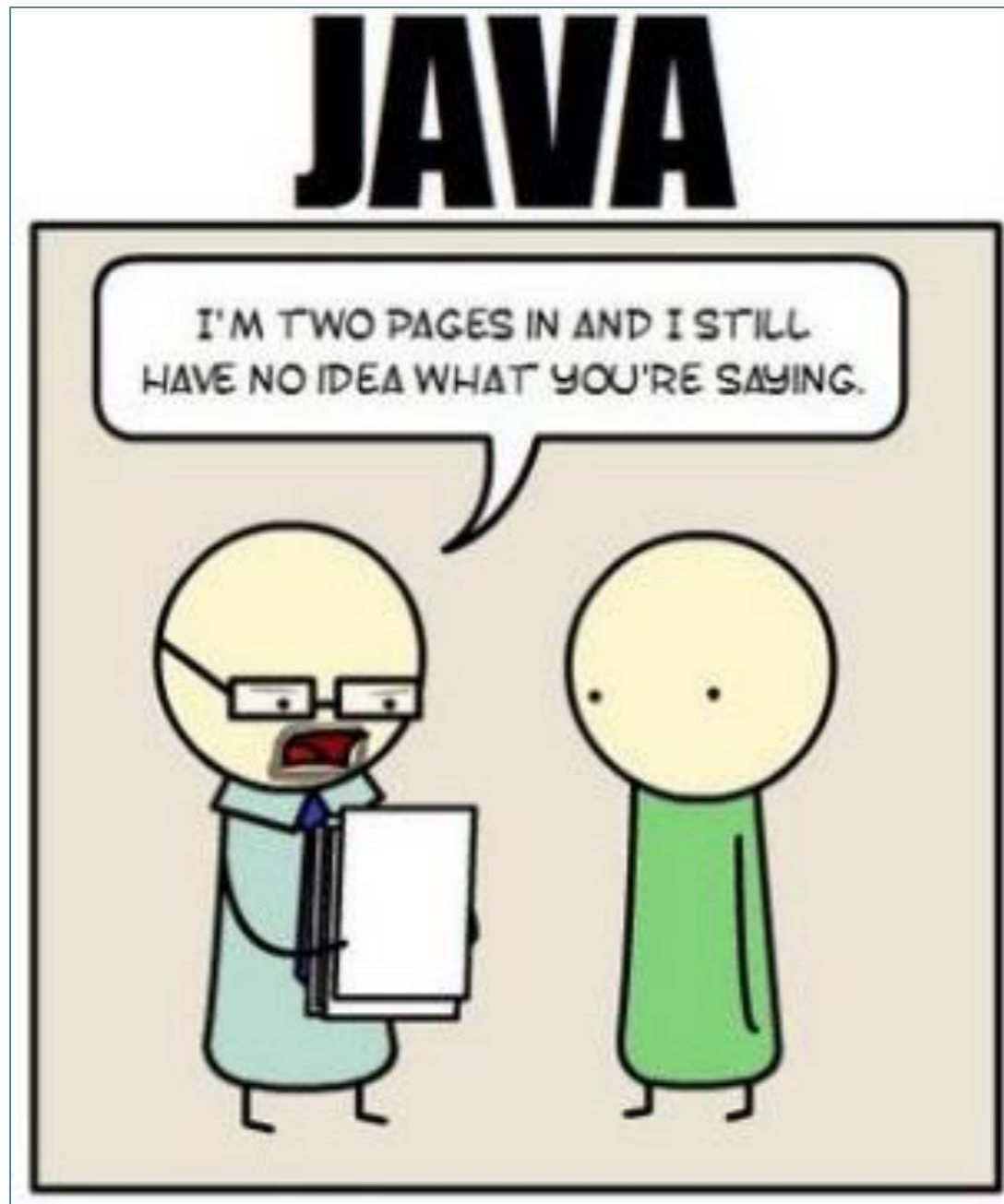


JVM  
Language  
Evolution



# Kotlin (Concise) Vs Java (Overly Verbose)

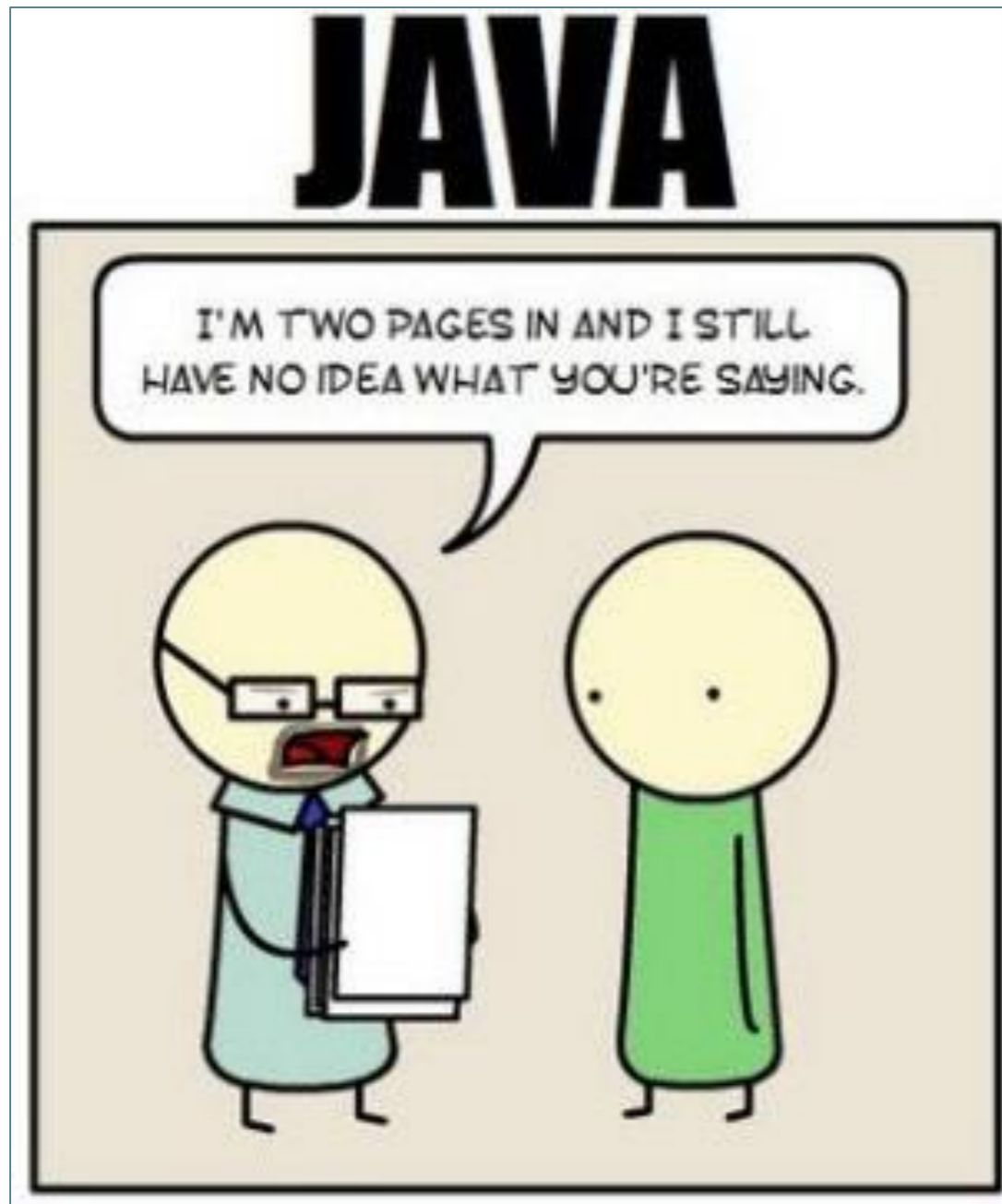
---



*“Java is extremely verbose and ceremonious. Programmers need to write reams of code to get a simple task done. There's a great deal of ‘ceremony’ in Java APIs, and Android aggravates this by forcing developers to go through many steps, in a specific order, to get things done.”*

# Kotlin (Concise) Vs Java (Overly Verbose)

---



*“Java code is verbose...”*

*“Kotlin provides a well thought-out syntax and extensive standard library that removes many of the pain points that exist in Java.”*



# Kotlin (Concise) Vs Java (Overly Verbose)

---

*Rough estimates  
indicate approximately  
a 40% cut in the  
number of lines of code.*



# Kotlin (Concise) Vs Java (Overly Verbose)

toptal

```
public class User {  
    private String firstName;  
  
    private String lastName;  
  
    public User() {  
        return this;  
    }  
  
    public String  
    firstName  
    this  
    me;  
    }  
  
    public  
    ret  
    }  
  
    public void setLastName(String  
    lastName) {  
        this.lastName = lastName;  
    }  
}
```

```
class User {  
    var firstName: String? = null  
  
    var lastName: String? = null  
}
```

Kotlin drastically reduces the amount of boilerplate code you have to write.

The less code you write, the fewer mistakes you make, the less to test, the better the execution.



# Easy(ish) learning curve for Java Developers

---

*“Kotlin is approachable and can be acquired in a few hours by simply reading the language reference. It has a lean and intuitive syntax.”*

*“Kotlin is also designed to have a gentle learning path for Java developers. Java programmers will find that most of the Kotlin syntax feels familiar.”*



# Kotlin

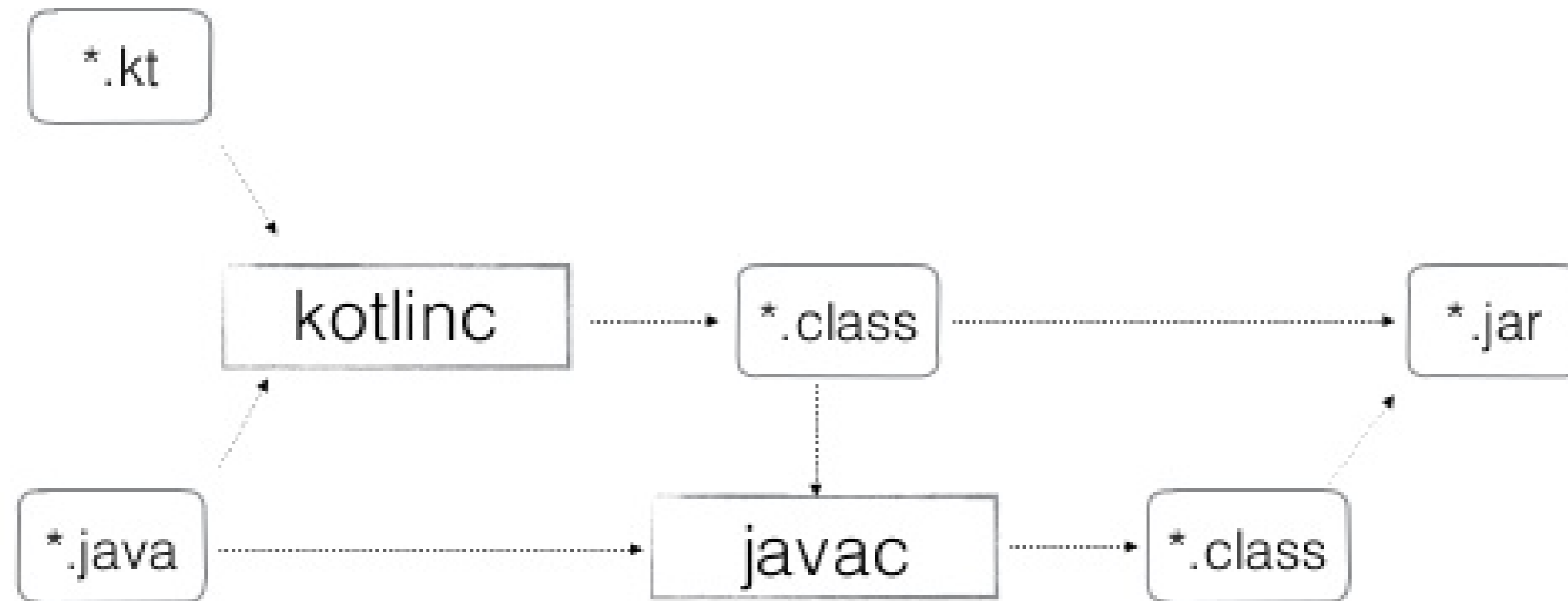
---

- runs on Java Virtual Machine.
- is an evolution of the Java syntax but is more concise and has cleaner syntax.
- is not syntax compatible with Java; but is interoperable with Java.
- relies on some Java Class Libraries e.g. Collections framework.
- is a statically-typed programming language.
- offers null safety.



# Kotlin / Java Interoperability

## Compilation of a mixed project





# Kotlin / Java Interoperability

---

- Kotlin and Java are 100% interoperable; Java and Kotlin code can co-exist very well in the same project and compile perfectly.
- Kotlin can be called from Java and Java from Kotlin.
- Both .java and .kt files are compiled to .class bytecode.
- When a project containing both Java and Kotlin is compiled, it would be difficult to tell which parts were created in Java and which in Kotlin.
- You can start using Kotlin in an existing Java project, without having to convert the project to Kotlin.



# Kotlin

---

- runs on Java Virtual Machine.
- is an evolution of the Java syntax but is more concise and has cleaner syntax.
- is not syntax compatible with Java; but is interoperable with Java.
- **relies on some Java Class Libraries e.g. Collections framework.**
- is a statically-typed programming language.
- offers null safety.



# Kotlin and the Java Class Libraries

---

## Interoperability advantages:

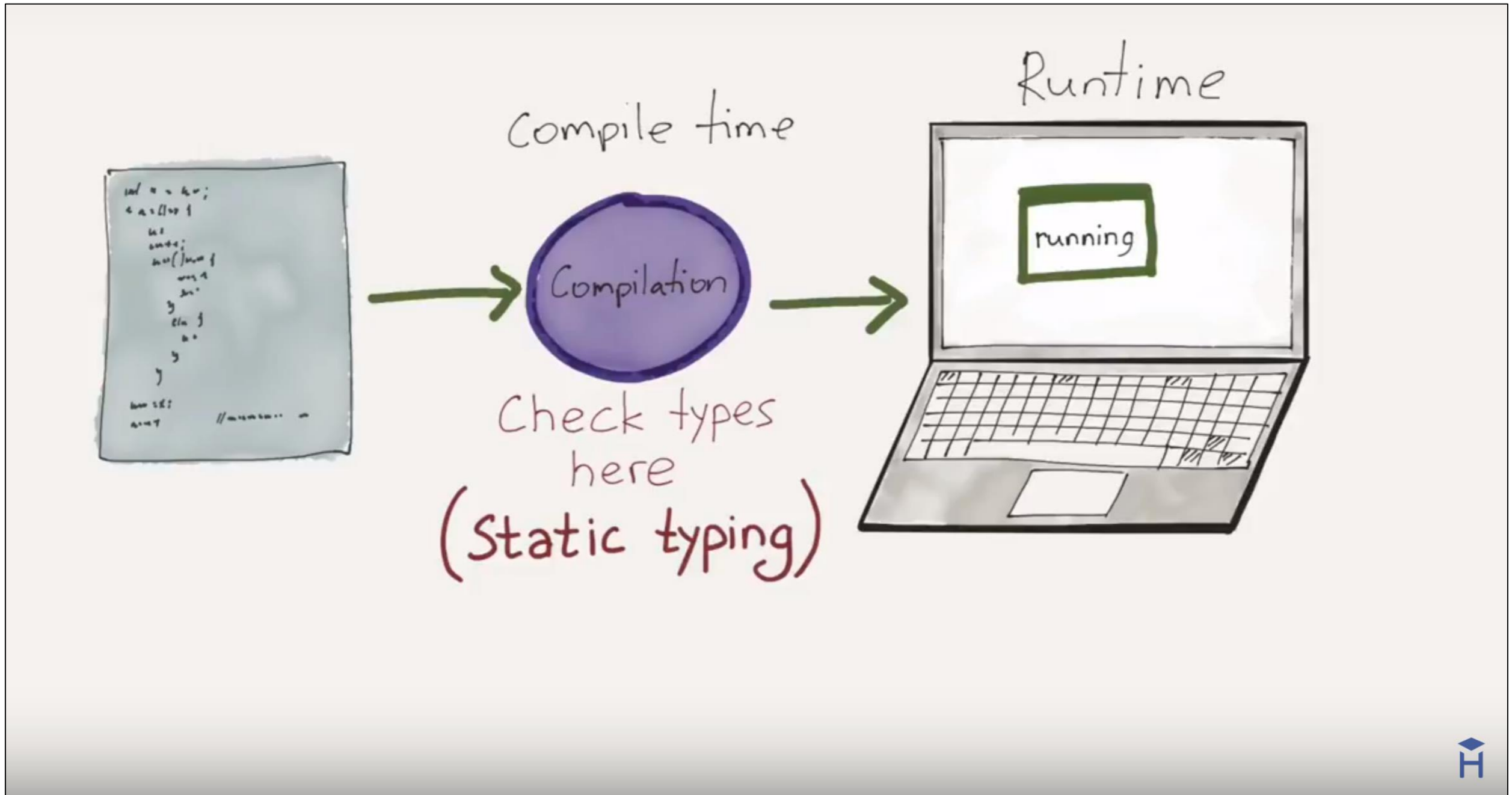
- you can use any of the vast number of Java Libraries and Frameworks in a Kotlin project.
- Kotlin can also easily integrate with Maven, Gradle and other build systems.



# Kotlin

---

- runs on Java Virtual Machine.
- is an evolution of the Java syntax but is more concise and has cleaner syntax.
- is not syntax compatible with Java; but is interoperable with Java.
- relies on some Java Class Libraries e.g. Collections framework.
- **is a statically-typed programming language.**
- offers null safety.



## STATIC TYPING

*"Variable declarations are mandatory before usage, else results in a compile-time error"*

# Static Typing – Example



```
String greeting = "Hello!";  
int someRandomInteger = 100;  
double aDoubleVariable = 2.2;
```

A type is  
assigned to  
each variable.

In Java, if we don't assign a  
type, we get a compiler error  
→ Java is statically typed.

Types determine the  
operations we can perform on  
the variables.



## Static Typing – Example



**Kotlin**

*In Kotlin, you don't have to specify the type of each variable explicitly, even though Kotlin is statically-typed.*

*Here, Kotlin determines the type from the initialisation.*

```
fun main(args : Array<String>)  
{  
    var someRandomInteger = 100  
    var aDoubleVariable = 2.2  
    println (someRandomInteger)  
    println (aDoubleVariable)  
}
```





## Static Typing – Example



Kotlin

*However, you can choose to explicitly define a data type.*

```
fun main(args : Array<String>)  
{  
    var someRandomInteger : Int = 100  
    var aDoubleVariable : Double = 2.2  
    println (someRandomInteger)  
    println (aDoubleVariable)  
}
```



## Static Typing – Example



Kotlin

*With Kotlin, you have to either define a type or initialise the variable (kotlin then determines the type!).*

```
fun main(args : Array<String>)  
{  
    var someRandomInteger //compile error  
    var aDoubleVariable : Double = 2.2  
    println (someRandomInteger)  
    println (aDoubleVariable)  
}
```



# Static Typing – Example



Kotlin

```
fun main(args : Array<String>)  
{  
    var someRandomInteger : Int = 100  
    var aDoubleVariable : Double = 2.2  
  
    someRandomInteger = 2.65           //compile error  
    aDoubleVariable = 233             //compile error  
  
    println (someRandomInteger)  
    println (aDoubleVariable)  
}
```



# Kotlin

---

- runs on Java Virtual Machine.
- is an evolution of the Java syntax but is more concise and has cleaner syntax.
- is not syntax compatible with Java; but is interoperable with Java.
- relies on some Java Class Libraries e.g. Collections framework.
- is a statically-typed programming language.
- **offers null safety.**

# Null – Billion Dollar Mistake

---



I call it my billion-dollar mistake. It was the invention of the null reference in 1965.

— *Tony Hoare* —

AZ QUOTES



# Kotlin and Null Safety

---

- Kotlin eliminates most sources of null references by making all types non-nullable by default — meaning that the compiler won't let you use a non-initialized, non-nullable variable.
- If you need a variable to hold a null value, you have to declare the type as nullable, adding a question mark after the type (more on this in later lectures).

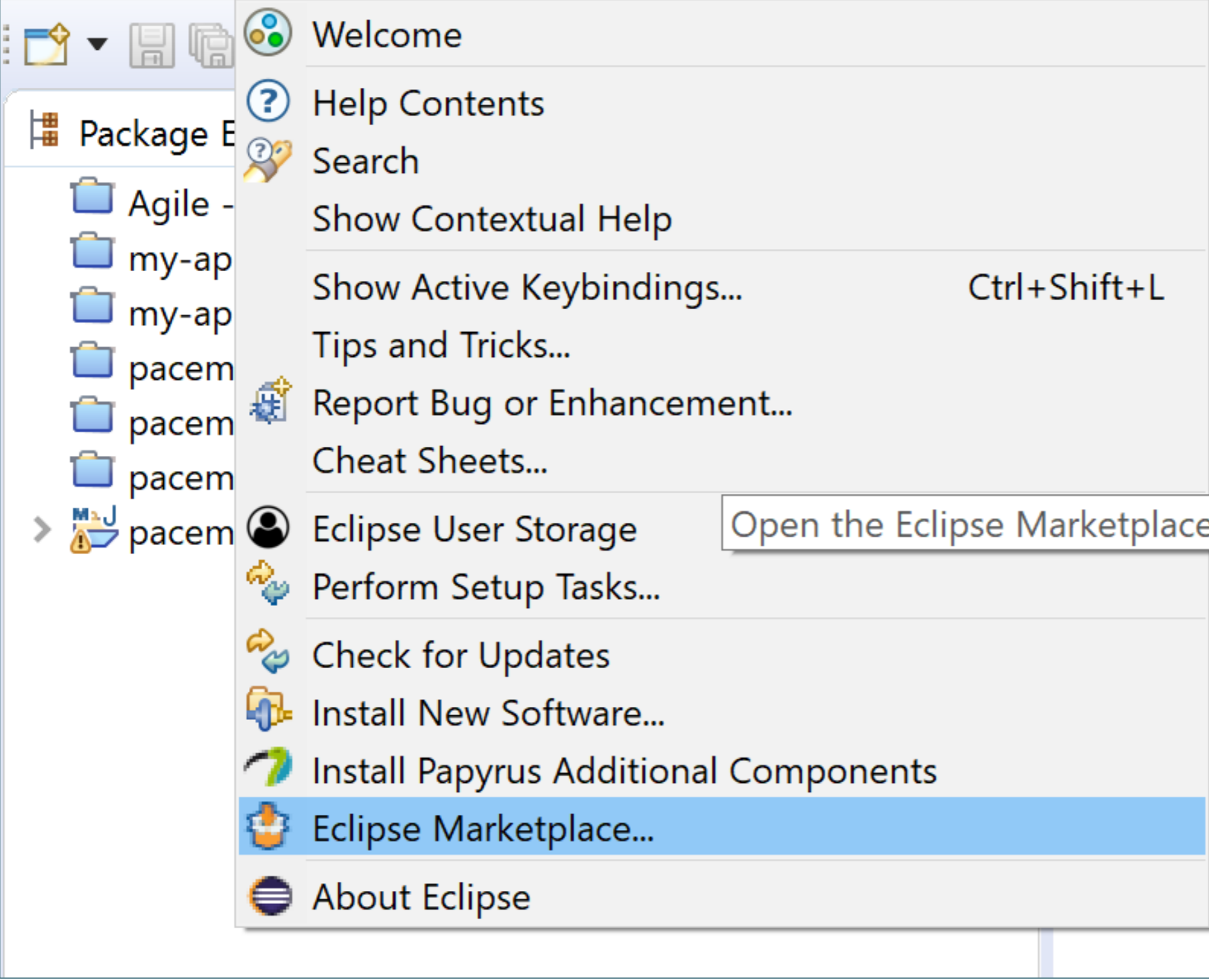
```
1 var nonNullable: String = "My string" // needs to be initialized
2 var nullable: String?
```



# Kotlin

---





The screenshot shows the Eclipse IDE interface with the Help menu open. The menu items are: Welcome, Help Contents, Search, Show Contextual Help, Show Active Keybindings... (Ctrl+Shift+L), Tips and Tricks..., Report Bug or Enhancement..., Cheat Sheets..., Eclipse User Storage, Perform Setup Tasks..., Check for Updates, Install New Software..., Install Papyrus Additional Components, Eclipse Marketplace..., and About Eclipse. The Eclipse Marketplace... option is highlighted in blue. A tooltip box is positioned over the Eclipse User Storage option, containing the text "Open the Eclipse Marketplace wizard".

Open the Eclipse Marketplace wizard

**Install Kotlin  
plugin into  
Eclipse**









Eclipse Marketplace

## Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.  
Press the "more info" link to learn more about a solution.




Search Recent Popular Favorites Installed  Eclipse Newsletter (Oct 2017)



Find:    All Markets All Categories

### Kotlin Plugin for Eclipse 0.8.2

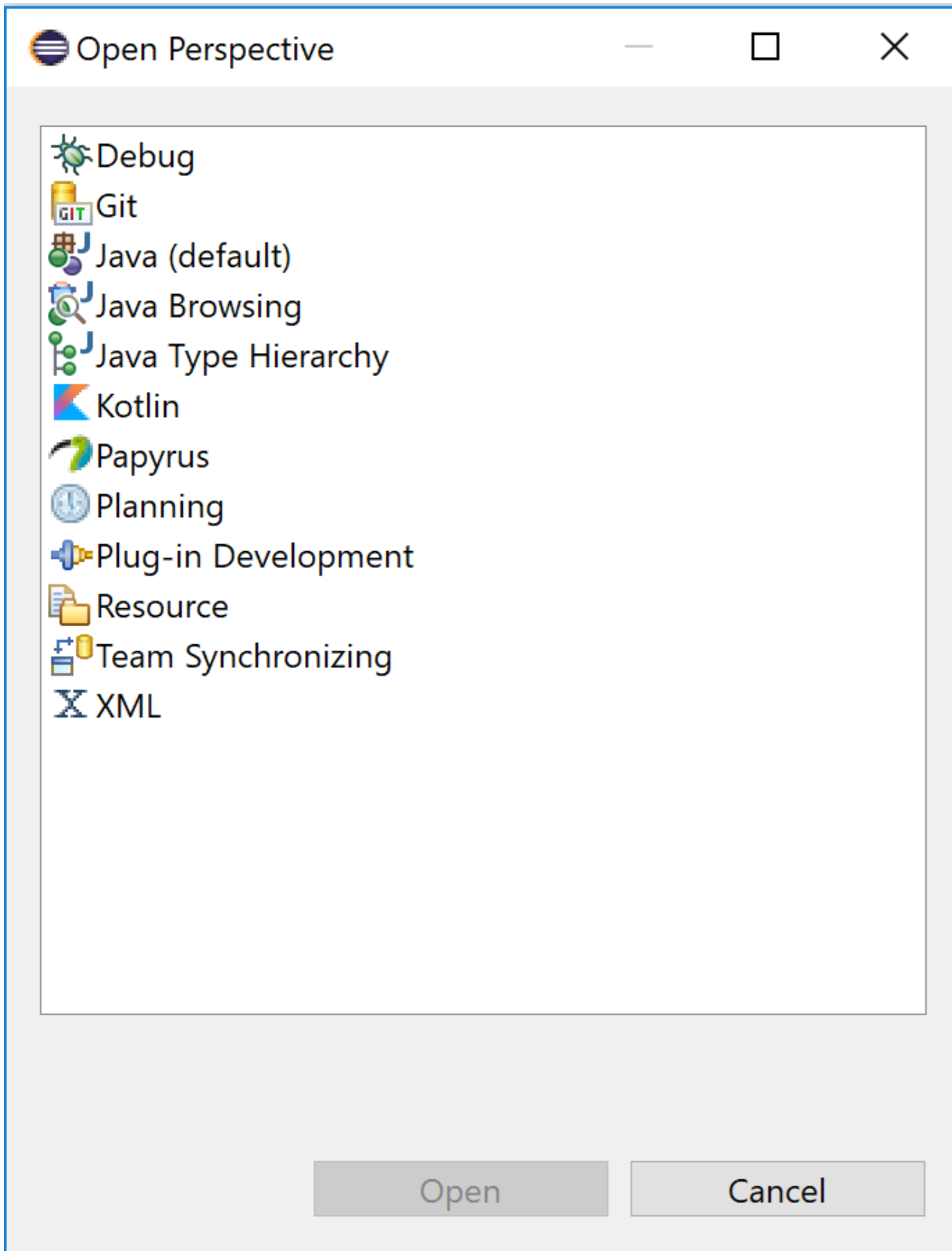
The Kotlin Plugin for Eclipse helps you write, run, debug and test programs in Kotlin language. [more info](#)



by [JetBrains](#), Apache 2.0  
[kotlin](#) [jetbrains](#) [jvm](#) [java](#) [fileExtension](#) [kt](#)

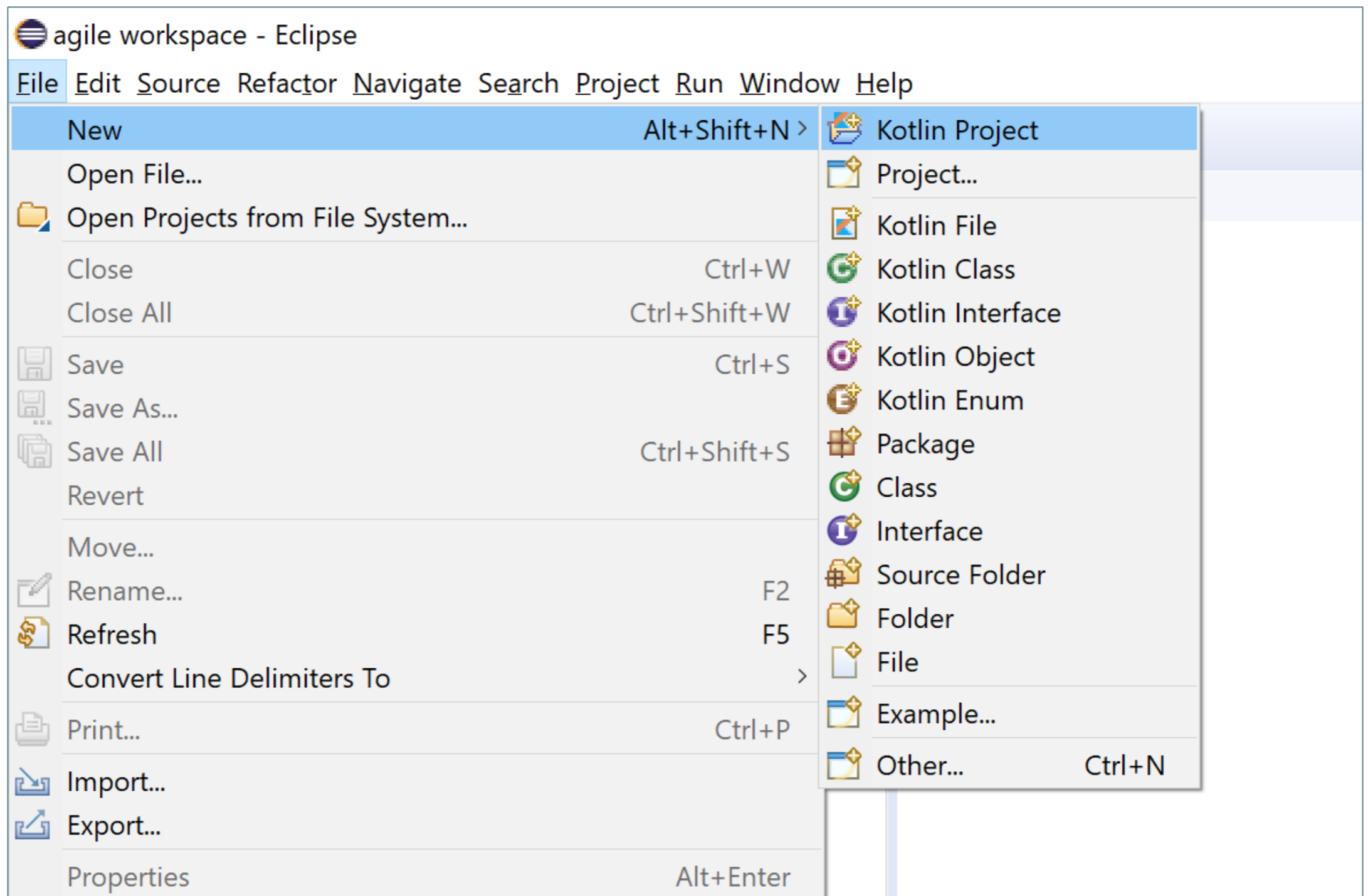
 69  Installs: **25.4K** (2,687 last month)

Install Kotlin  
plugin into  
Eclipse

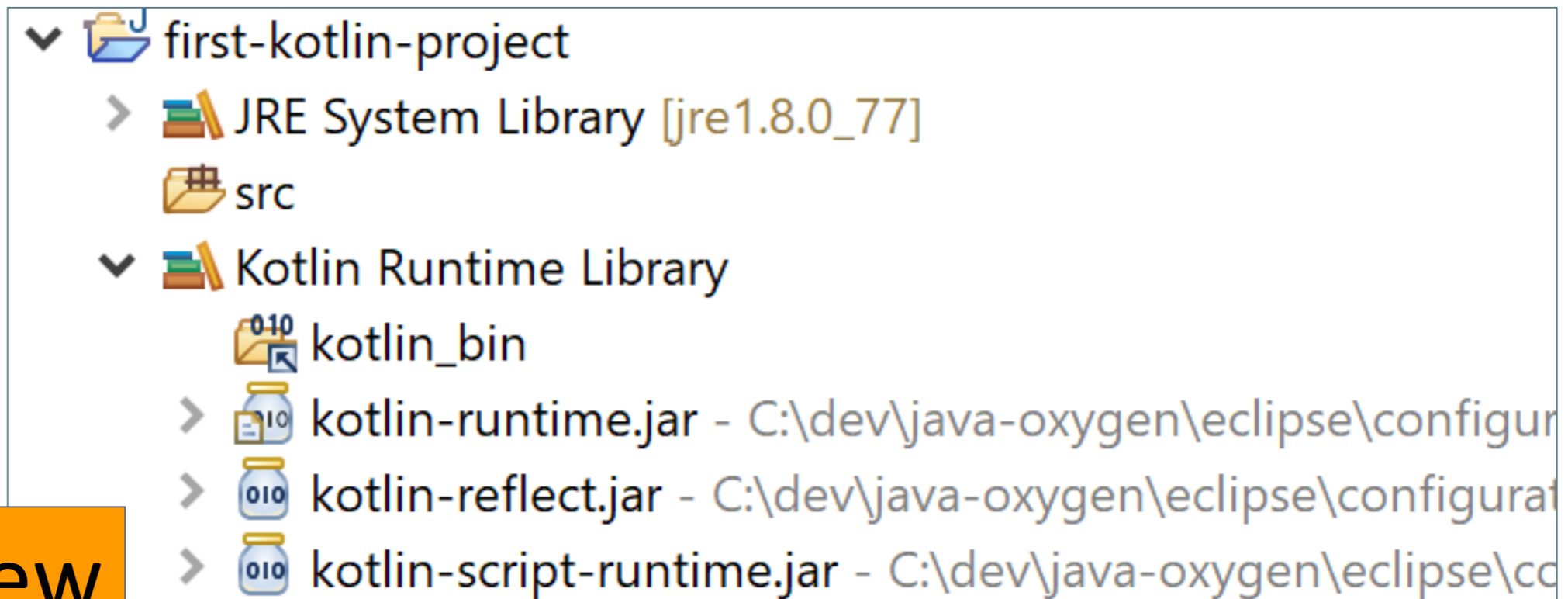
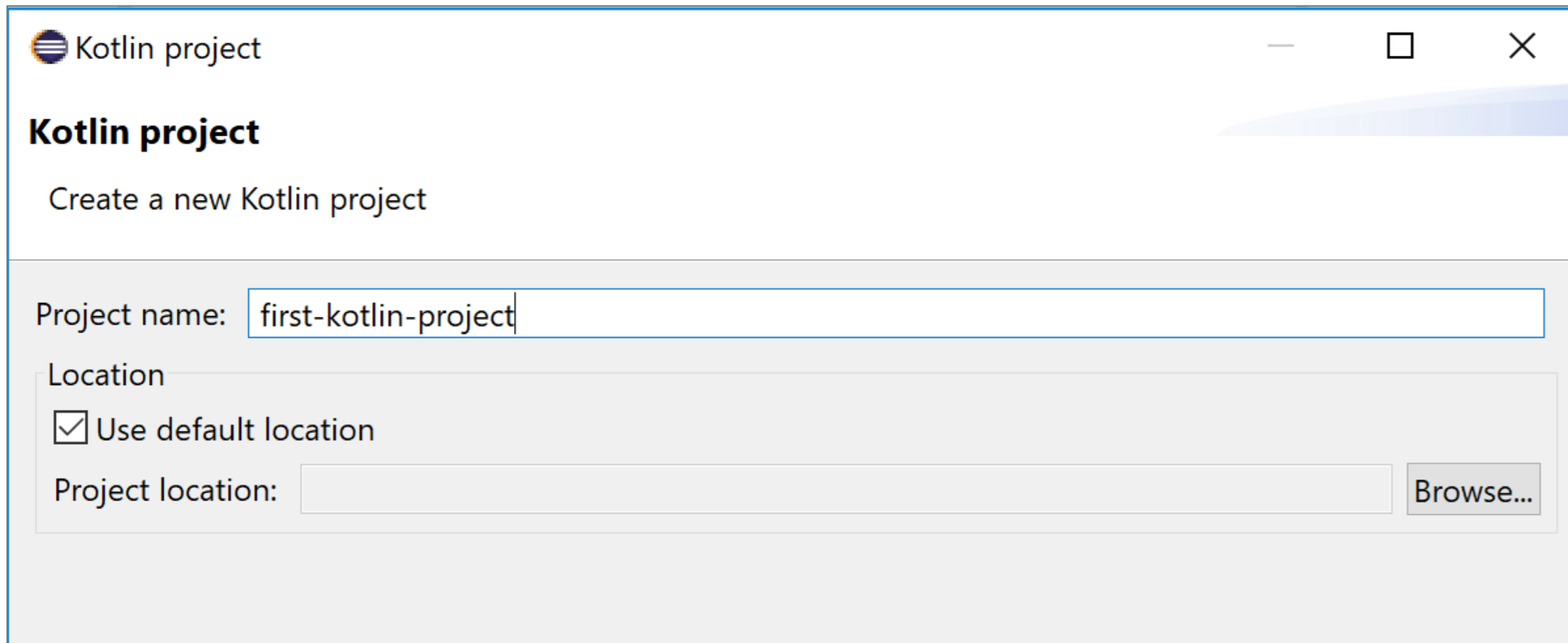


Change  
perspective  
to Kotlin.

Install Kotlin  
plugin into  
Eclipse



Create a new  
Kotlin project

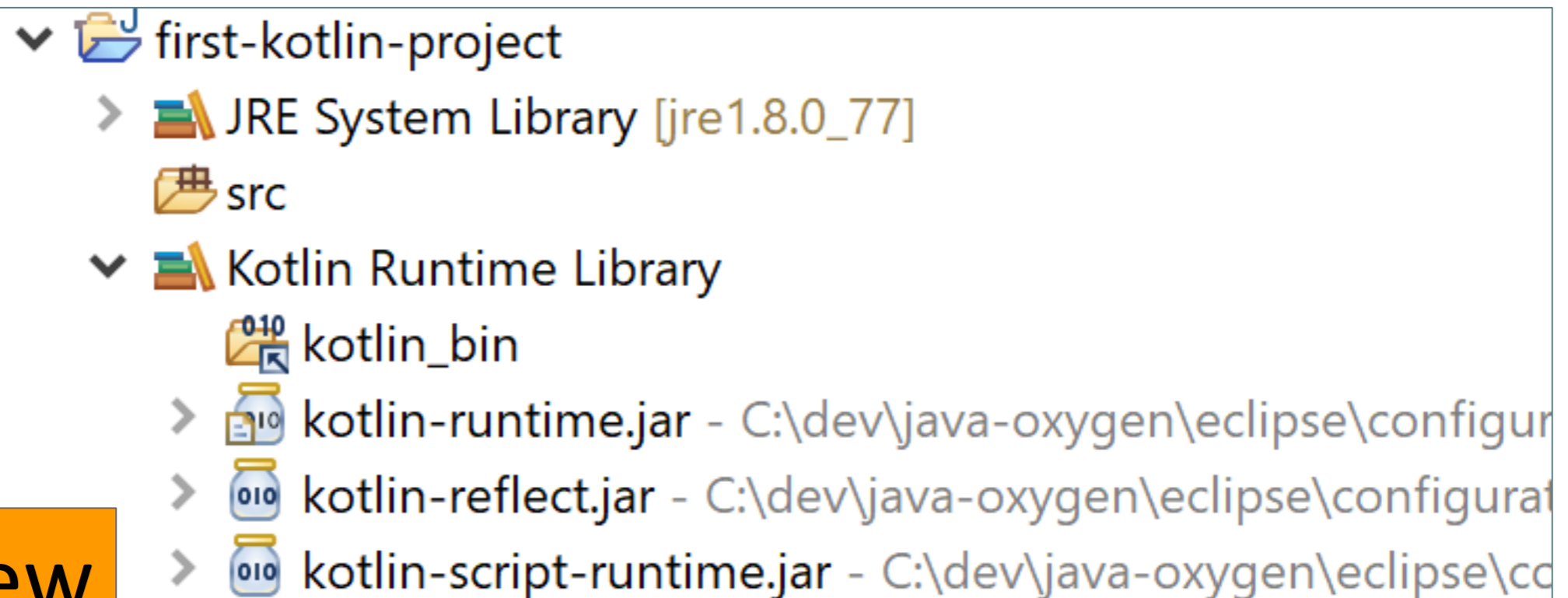


Create a new  
Kotlin project

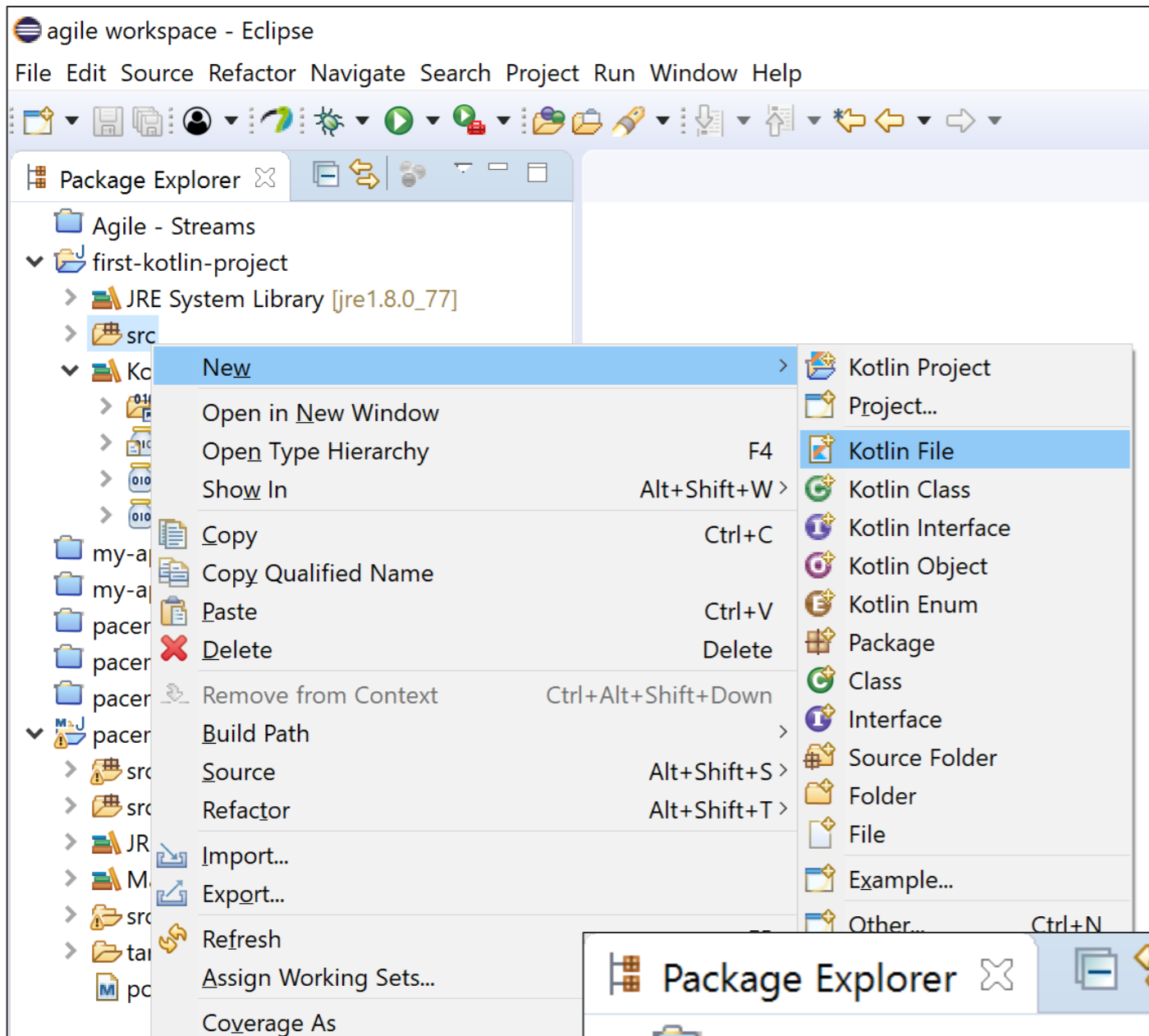
Eclipse project is also a Java project with a:

- Kotlin Builder and
- Kotlin Runtime Library.

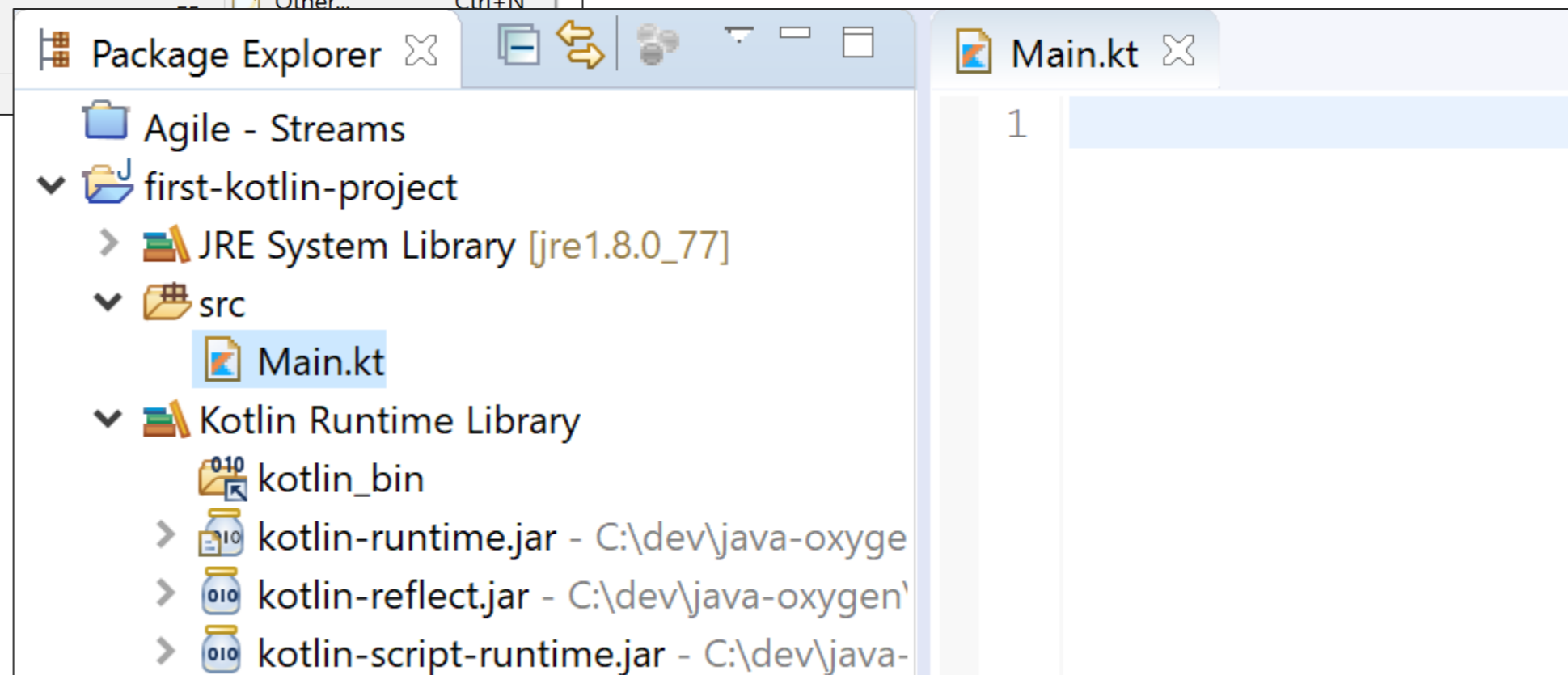
→ Can add Java classes to the project, mixing and matching Kotlin and Java code where required.

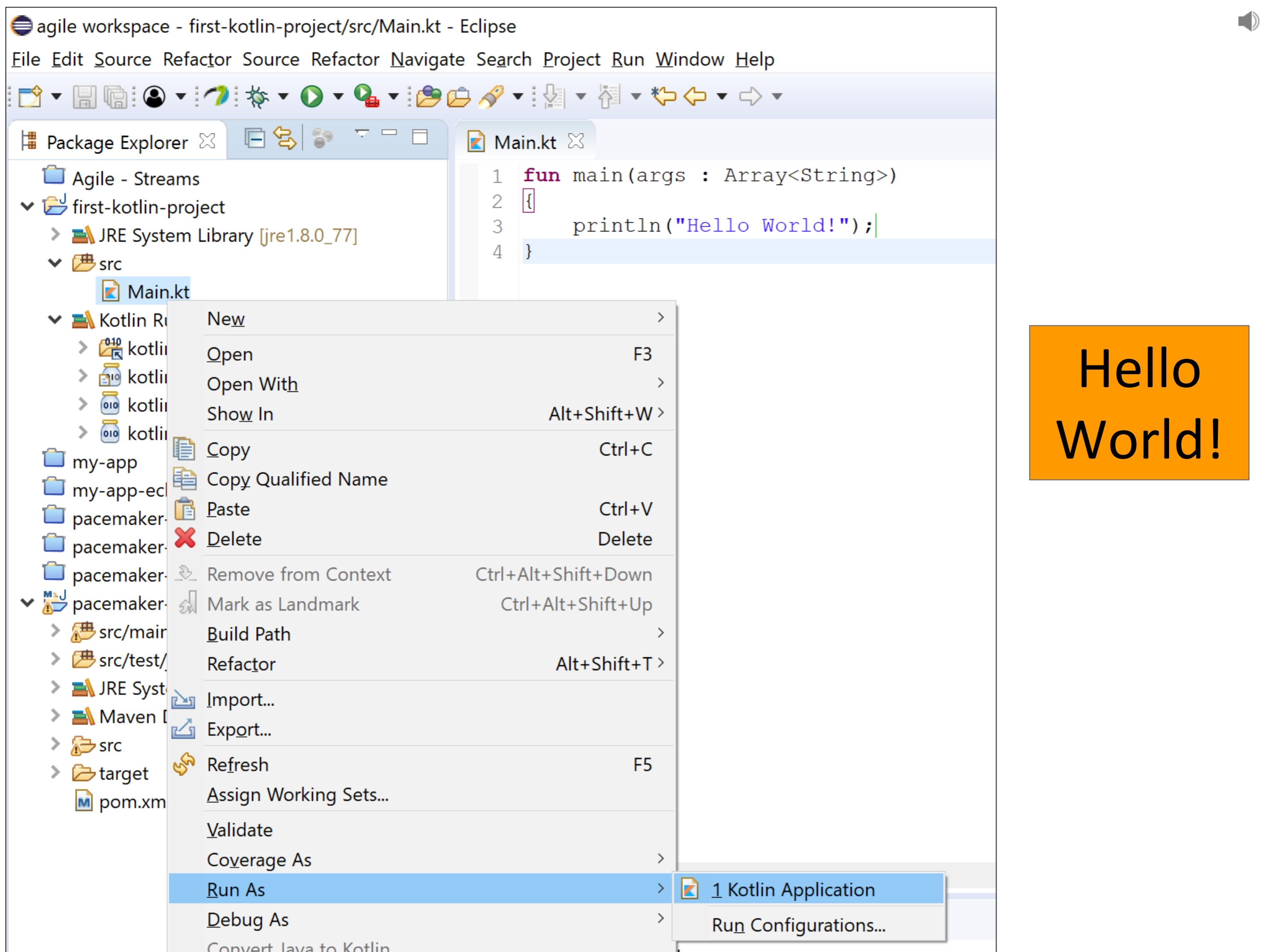


Create a new  
Kotlin project



Create a new  
Kotlin file





Package Explorer

- Agile - Streams
- first-kotlin-project
  - JRE System Library [jre1.8.0\_77]
  - src
    - Main.kt

```
1 fun main(args : Array<String>)  
2 {  
3     println("Hello World!");  
4 }
```

- New
- Open F3
- Open With
- Show In Alt+Shift+W
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Mark as Landmark Ctrl+Alt+Shift+Up
- Build Path
- Refactor Alt+Shift+T
- Import...
- Export...
- Refresh F5
- Assign Working Sets...
- Validate
- Coverage As
- Run As
- Debug As
- Convert Java to Kotlin

- 1 Kotlin Application
- Run Configurations...

Hello World!



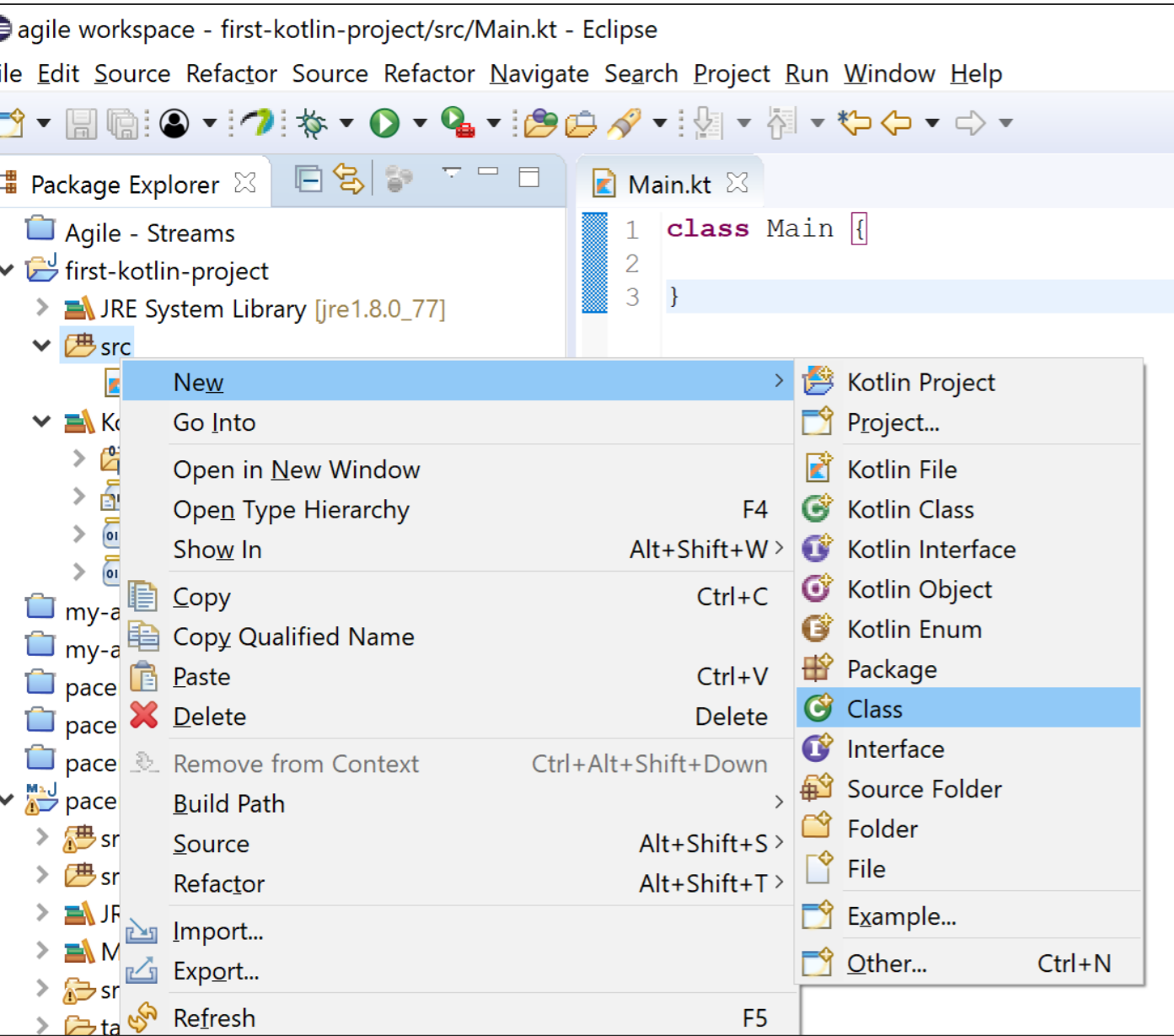
```
Main.kt X
1 fun main(args : Array<String>)
2 {
3     println("Hello World!");
4 }
5
```

Hello  
World!



```
Problems Console X
<terminated> Config - Main.kt [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\javaw.exe (24 Oct 2017, 20:47:52)
Hello World!
```





# Interoperability: Create a new **Java** class

```
ain.kt Customer.java  
  
public class Customer {  
  
    private String name;  
  
    public Customer(String name) {  
        super();  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    @Override  
    public String toString() {  
        return "Customer [name=" + name + "];"  
    }  
}
```

# Interoperability: Create a new **Java** class

```
Main.kt x Customer.java
1 fun main(args : Array<String>)
2 {
3     val customer = Customer("Kotlin Customer")
4     println(customer.getName())
5 }
6
```



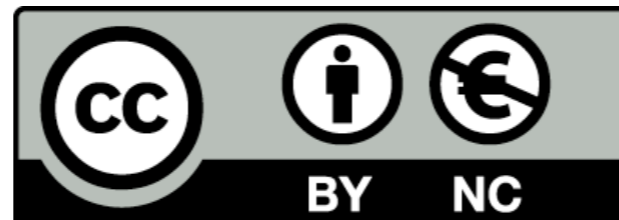
```
Problems Console x
<terminated> Config - Main.kt [Java Application] C:\P
Kotlin Customer
```

# Kotlin Vs Java



```
Customer.kt ✕  
1 class Customer(name: String?) {  
2     var name: String? = null  
3  
4     init {  
5         this.name = name  
6     }  
7  
8     override fun toString(): String {  
9         return "Customer [name=" + name + "]"  
10    }  
11 }
```

```
ain.kt Customer.java ✕  
  
public class Customer {  
  
    private String name;  
  
    public Customer(String name) {  
        super();  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    @Override  
    public String toString() {  
        return "Customer [name=" + name + "];"  
    }  
  
}
```



Except where otherwise noted, this content is licensed under a [Creative Commons Attribution-NonCommercial 3.0 License](http://creativecommons.org/licenses/by-nc/3.0/).

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>

