VICTORIA GONDA

VICTORIAGONDA.COM

LET KOTLIN COME TO YOU

WHO AM I?

- Android Engineer at Buffer
- Author at Ray Wenderlich
- Chicago
- Dancer




Kotlin

“STATICALLY TYPED PROGRAMMING LANGUAGE FOR MODERN MULTIPLATFORM APPLICATIONS”

kotlinlang.org

KOTLIN

- Developed by JetBrains
- Available for JVM, Android, JavaScript, and Native
- Concise, safe interoperable, tool-friendly
- Built in IDE support (with JetBrains products)

KOTLIN

- Strong, static, inferred, null safe typing
- Immutability by default
- Smart casting
- Higher order functions
- Boilerplate reduction

TYPING

STATICALLY TYPED




```java
final String name = "Victoria";
```

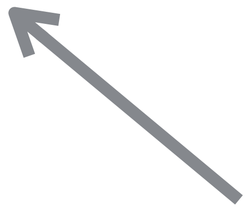


```kotlin
val name = "Victoria"
```

NULL TYPING

```
val icecream: IceCream?
```

```
icecream.flavor
```



```
if (icecream != null) {
  return icecream.flavor
} else {
  return null
}
```

SAFE CALL OPERATOR

```
return icecream?.flavor
```

!! OPERATOR

```
return icecream!!.flavor
```

ELVIS OPERATOR

```
return icecream?.flavor ?: "Chocolate Chip"
```

NULL SAFE SCOPING WITH HIGHER ORDER FUNCTIONS

```
return icecream?.let({ nonNullIceCream ->
    nonNullIceCream.flavor })

return icecream?.let({ it.flavor })

return icecream?.let { it.flavor }

return icecream?.let(IceCream::flavor)
```

IMMUTABILITY

```
val conference = "GOTO Chicago"

conference = "GOTO Chicago 2018”
```

```
var conference = "GOTO Chicago"

conference = "GOTO Chicago 2018”
```

```
val readOnlylist = listOf(1, 2, 3)

val mutableList = mutableListOf("a", "b", “c")
```

SMART CASTING

SMART CASTING

```
class Conference(val year: String): Event()

fun smartCasting(event: Event) {
    if (event is Conference) {
        event.year
    }
}
```

BOILERPLATE REDUCTION

CLASSES

```
class Cupcake(
  val flavor: String,
  var icing: String?
)
```

```java
public final class Cupcake {
  @NotNull
  private final String flavor;
  @Nullable
  private String icing;

  @NotNull
  public final String getFlavor() {}

  @Nullable
  public final String getIcing() {}

  public final void setIcing(@Nullable String var1) {}

  public Cupcake(@NotNull String flavor, @Nullable String icing) {}
}
```

```
data class Cupcake(
  val flavor: String,
  var icing: String?
)
```

```
public final class Cupcake {
  // …

  @NotNull
  public final String component1() {}

  @Nullable
  public final String component2() {}

  @NotNull
  public final Cupcake copy(@NotNull String flavor, @Nullable String icing) {}

  public String toString() {}

  public int hashCode() {}

  public boolean equals(Object var1) {}
}
```

SEALED CLASSES

ENUMS

```
enum class GetConferenceState {
    SUCCESS, ERROR, PROGRESS
}
```

SEALED CLASSES

```
sealed class GetConferenceState {

    class Success(val result: Conference): GetConferenceState()

    data class Error(val errorMessage: String): GetConferenceState()

    object InProgress: GetConferenceState()
}
```

SEALED CLASSES

```
when(state) {

  is GetConferenceState.Success -> showConference(state.conference)

  is GetConferenceState.Error -> showError(state.errorMessage)

  is GetConferenceState.InProgress -> showLoading()
}
```

AND MORE!


GOTO 2017 • Kotlin - Ready for Production • Hadi Hariri
goto;
conference
Kotlin

Kotlin: Uncovered

Victoria Gonda

@TTGonda

www.victoriagonda.com

WHERE CAN KOTLIN BE USED?

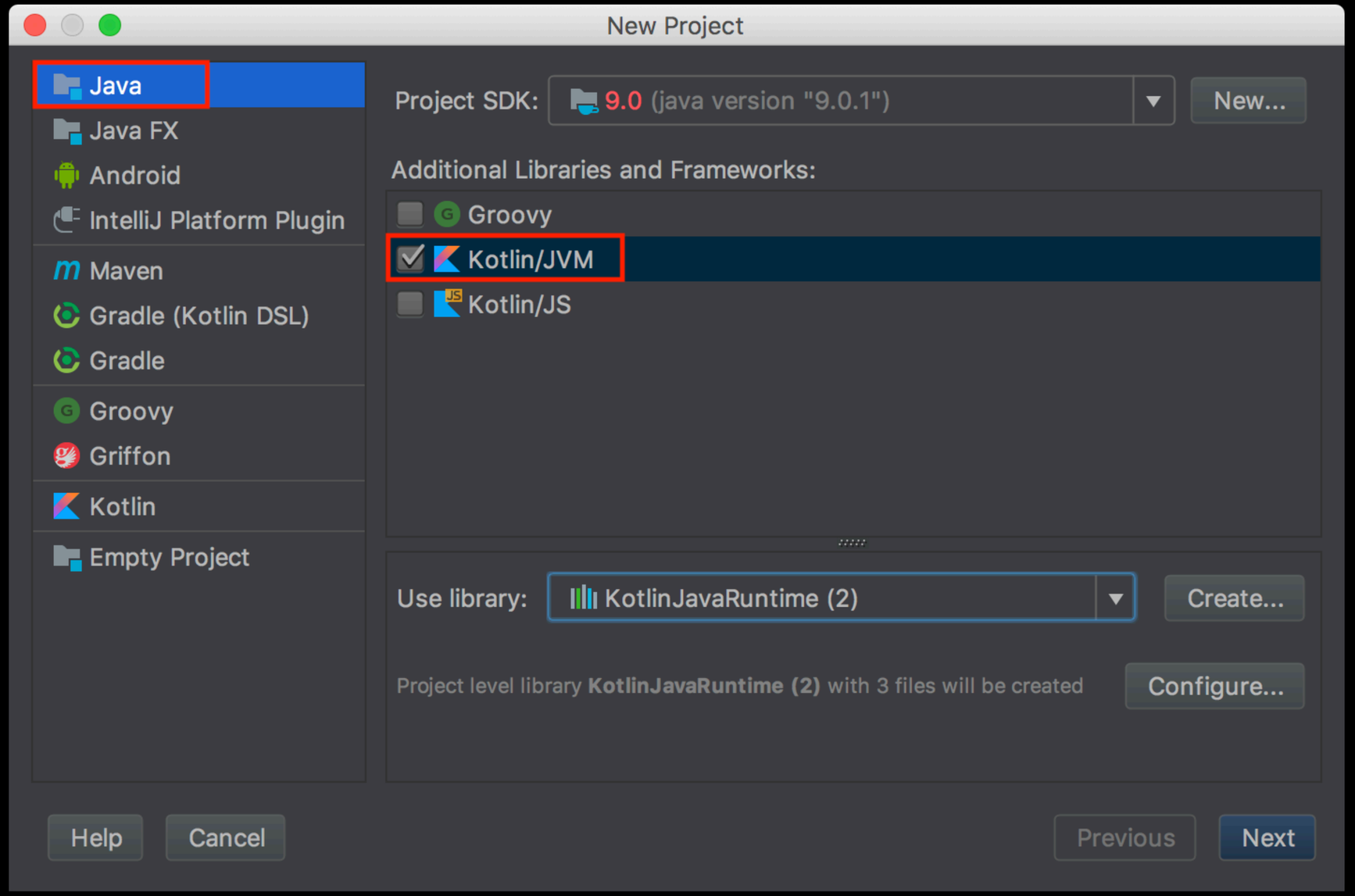
WHERE CAN KOTLIN BE USED?

- JVM, Android, JavaScript, Native
- Mobile, Server, Browser, Views
- IntelliJ, Android Studio, Eclipse, Standalone Compilers
- Kotlin all the things!
- Try it out in the browser (try.kotlinlang.org)

JAVA VIRTUAL MACHINE

- Use on the server, Android, Java applications
- Spring Boot, for example
- Anywhere the JVM lives
- Interoperable with Java
- More flexible and concise than Java
- Less feature bloat than Scala

JVM - INTELLIJ



buffer

JVM - INTELLIJ

- New Kotlin file in src directory

```
fun main(args: Array<String>) {
    println("Hello GOTO Chicago")
}
```

- Run! ▶️

JVM - GRADLE

- Set up gradle-kotlin plugin and apply it. Add dependencies.

```
repositories {
    mavenCentral()
}

dependencies {
    compile "org.jetbrains.kotlin:kotlin-stdlib"
}
```

- Alternatively in IntelliJ, **Tools > Kotlin > Configure Kotlin in Project**

JVM – ECLIPSE

- Download plugin from Marketplace
- ***File > New > Kotlin Project***

JVM - ECLIPSE

- New Kotlin file in src directory

```
fun main(args: Array<String>) {
    println("Hello GOTO Chicago")
}
```

- *Run As > Kotlin Application* ▶️

JVM - COMMAND LINE

- Install Kotlin

```
// hello.kt
fun main(args: Array<String>) {
    println("Hello GOTO Chicago")
}
```

- Run! ▶️

```
$ kotlinc hello.kt -include-runtime -d hello.jar
$ java -jar hello.jar
```

JAVA EQUIVALENT

```
public final class HelloKt {
   public static final void main(@NotNull String[] args) {
      Intrinsics.checkParameterIsNotNull(args, "args");
      String var1 = "Hello GOTO Chicago";
      System.out.println(var1);
   }
}
```

CONVERTING JAVA TO KOTLIN

- Android Studio and IntelliJ have built in conversion
- Three ways to get there
  - ***Code > Convert Java File to Kotlin File***
  - ⎇⇧⌘K
  - Search for action “Convert File to Kotlin File

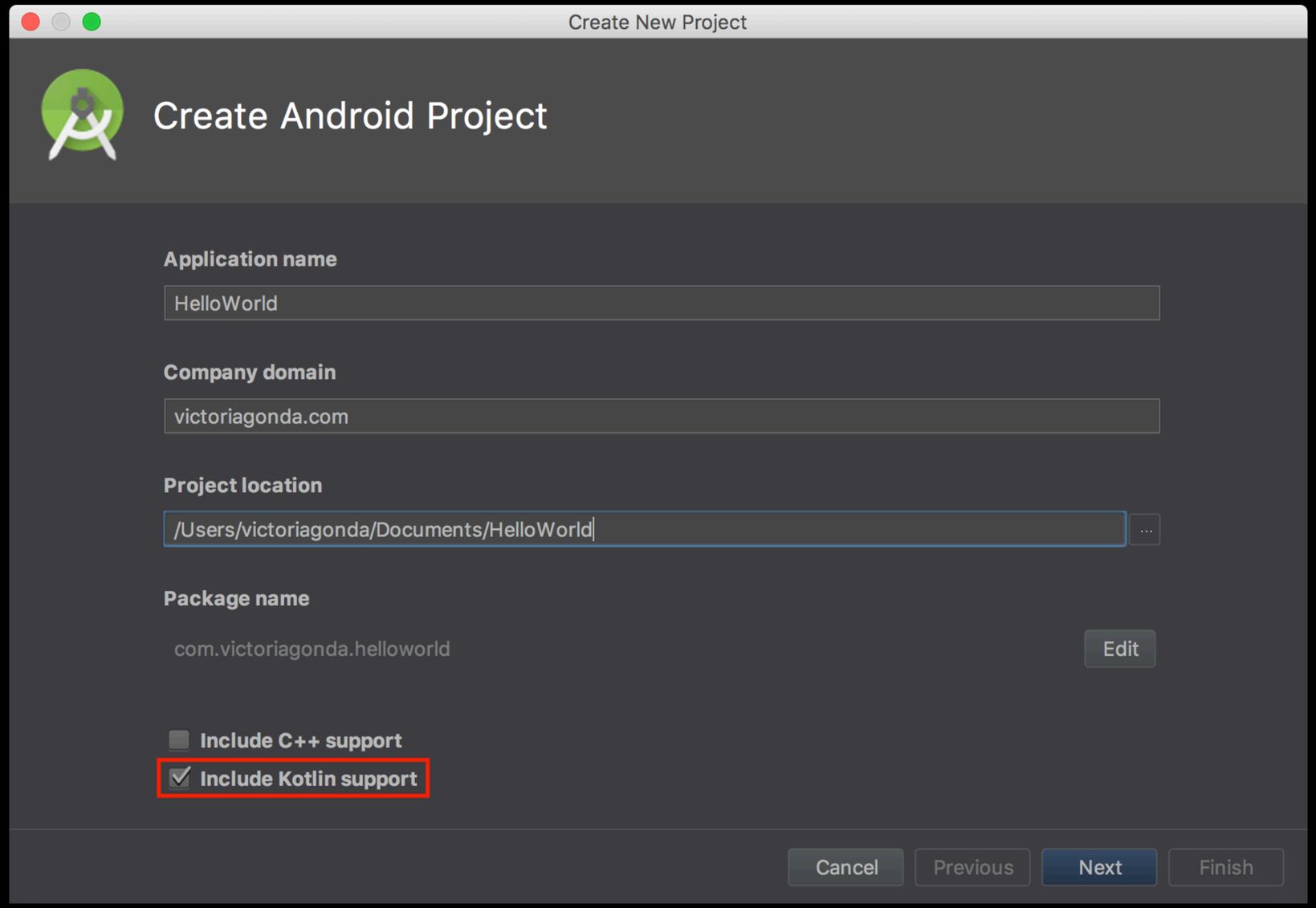
JVM - OTHER OPTIONS

- Ant
- Maven
- Griffon

ANDROID

- More flexible and concise than Java
- Interoperable with Java
- Can use all the same libraries
- Supported in Android Studio

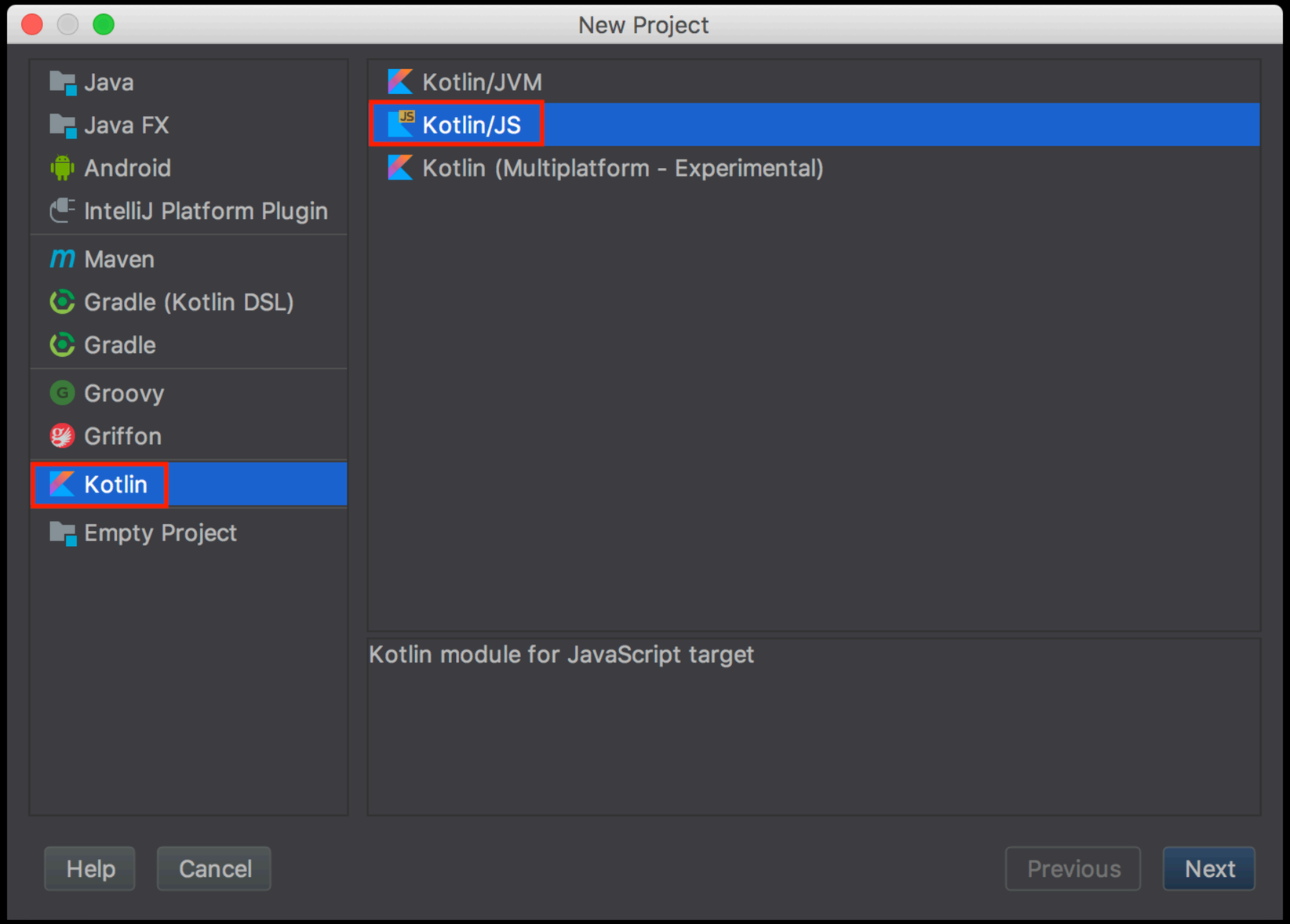
ANDROID STUDIO



JAVASCRIPT

- Use in browser, and any JS framework
- ReactJS, for example
- Type safe
- Interop with JavaScript

JS - INTELLIJ


New Project
Java
Java FX
Android
IntelliJ Platform Plugin
Maven
Gradle (Kotlin DSL)
Gradle
Groovy
Griffon
Kotlin
Empty Project
Kotlin/JVM
Kotlin/JS
Kotlin (Multiplatform - Experimental)
Kotlin module for JavaScript target
Help
Cancel
Previous
Next

JS - INTELLIJ

- New Kotlin file in src directory

```
fun main(args: Array<String>) {
    println("Hello GOTO Chicago")
}
```

JS – INTELLIJ

- Create index.html that runs script

```
<script type="text/javascript" src=“out/production/sampleapp/lib/kotlin.js">
</script>
<script type="text/javascript" src=“out/production/sampleapp/sampleapp.js">
</script>
```

- ***Build > Build Application***
- Open index.html in browser ▶️

JS - COMMAND LINE

- Library code is in *library.kt*

```
kotlinc-js -output sample-library.js -meta-info library.kt
```

- Output:

```
sample-library.js
sample-library.meta.js
```

JS - OTHER OPTIONS

- Gradle
- Maven

VIEWS

- Use kotlinx.html library to interact with, and build HTML views
- Use Anko to build views for Android
- You can use Kotlin for the logic of what to display in your views!

VIEWS

- kotlinx.html

```
val myDiv = document.create.div {
    p { +"text inside" }
}
```

NATIVE

- Available for macOS, Linux and Windows
- LLVM
- Interop with C libraries

NATIVE - COMMAND LINE

- Download compiler for your OS
- Create file *hello.kt*

```
fun main(args: Array<String>) {
    println("Hello GOTO Chicago")
}
```

- Compile

```
konanc hello.kt
```

- Run ▶️

```
/hello.kexe
```

NATIVE - OTHER OPTIONS

- CLion IDE
- Gradle

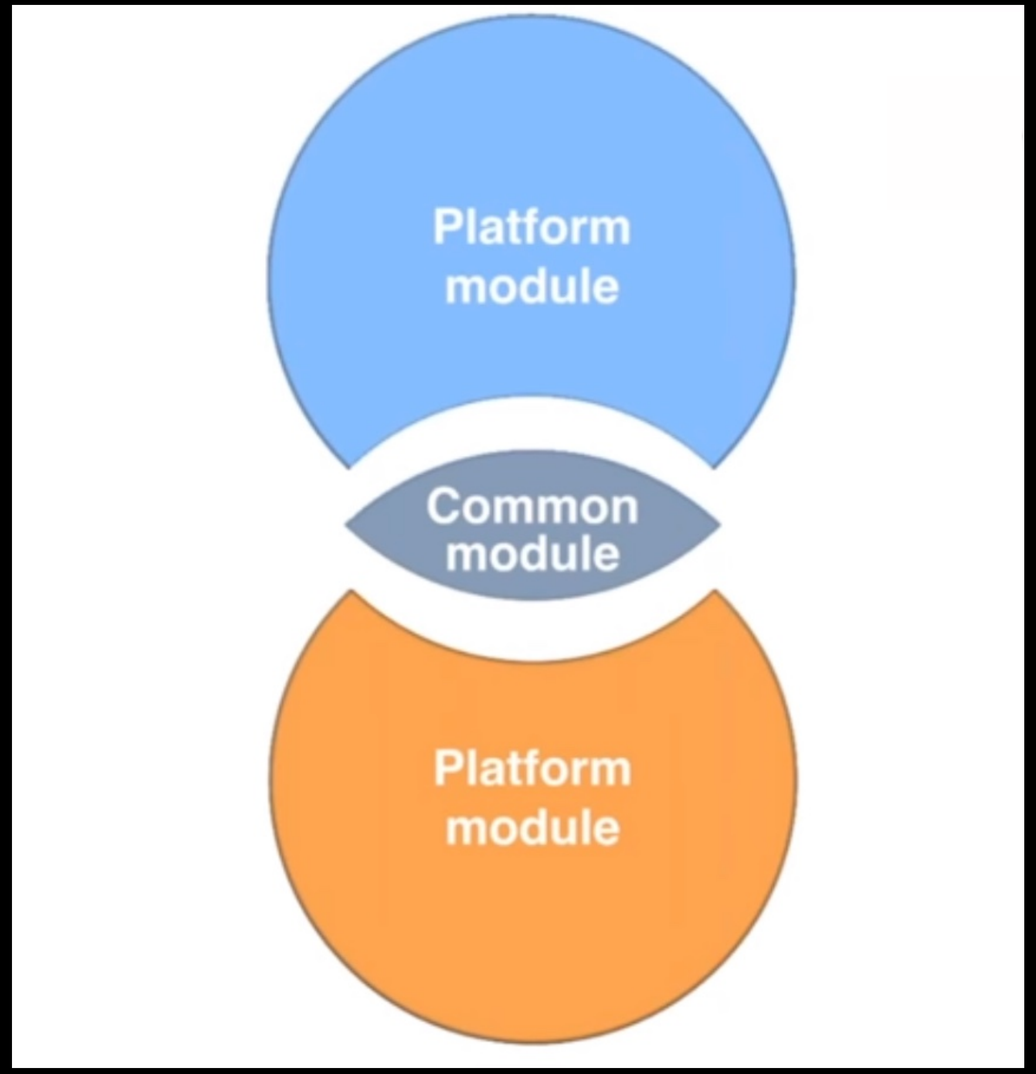
MULTI PLATFORM

MULTI PLATFORM

- Share logic across platforms
- Write once, use everywhere
- Keep native for any framework components
- Note: experimental

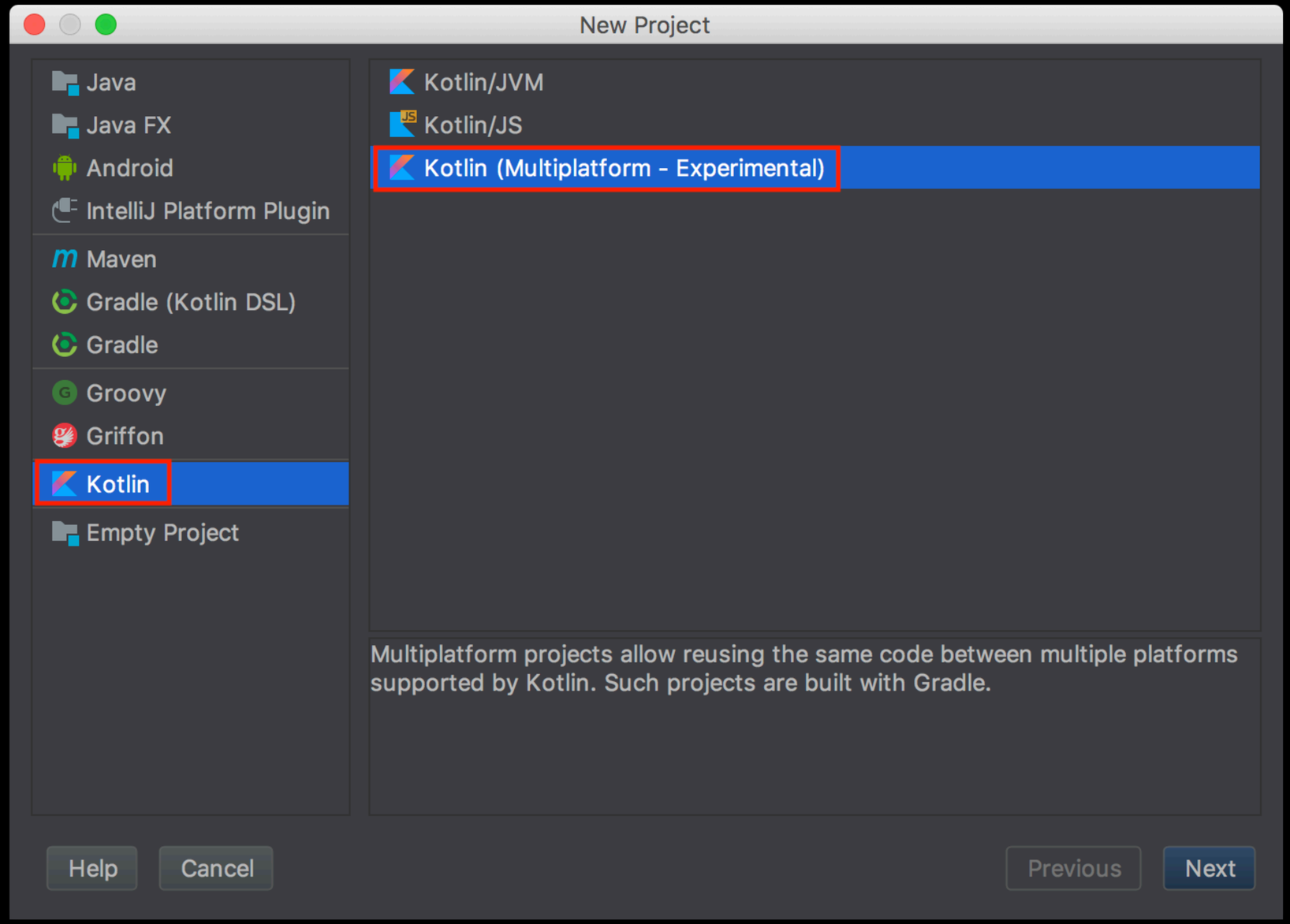
MULTI PLATFORM

- Common module with code shared across platforms
- Platform specific module
- “Interface” for the common code to call platform specific implementations



http://developine.com/kotlin-native-ios-development-multiplatform-project/

MULTI PLATFORM - INTELLIJ



buffer

JVM
Kotlin

common
Kotlin

JS
Kotlin



***.class**

***.js**

```
// In common module
expect class Dog(bar: String) {
    fun bark()
}

fun main(args: Array<String>) {
    Dog("Spot").bark()
}


// In platform module
actual class Dog actual constructor(val name: String) {
    actual fun bark() {
        println("$name says ruff")
    }
}
```

GET STARTED!

- kotlinlang.org 💯 docs
- try.kotlinlang.org to try Kotlin in the browser
- kotlin.link to find anything Kotlin you might want
- github.com/vgonda/Talk-Resources my resources from this talk (and others)
- Kotlin Slack
- Find me online!

VICTORIA GONDA

VICTORIAGONDA.COM

THANKS!