Practical Kotlin
with a disco ball

Nick DiPatri

Practical Kotlin

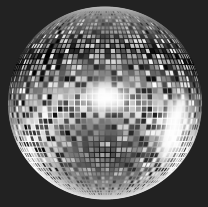
with a disco ball

Nick DiPatri

Overview

- What is Kotlin?
- Why should I use it?
- What are the risks?
- Syntactic Kotlin
- How to begin?
- Idiomatic Kotlin

What is Kotlin



- New language from JetBrains
- Syntax is very similar to Java
- Coexists with Java source
- Compiles down to Java bytecode

Why Kotlin?

- It's way better than Java!
- Fully supported by Google/Android
- Android leaders are using it

Why Kotlin?

What does this Java class do?


```
public class BeaconAttachment {

    private String description;
    private int distanceThreshold;
    private String placeId;

    public BeaconAttachment(EddystoneBeacon eddystoneBeacon) {
        this.description = eddystoneBeacon.getDescription();
        this.distanceThreshold = eddystoneBeacon.getDistanceThreshold();
        this.placeId = eddystoneBeacon.getPlaceId();
    }

    public String[] getAttachment() {
        return new String[] {description, String.valueOf(distanceThreshold), placeId};
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }
```

Why Kotlin?

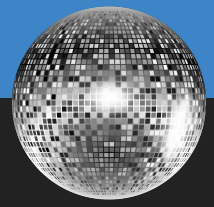
Answer: Very little.

With Kotlin, that fact is obvious

```
data class BeaconAttachment(var description: String = "a new beacon",
                            var distanceThreshold: Int = 5,
                            var placeId: String? = null) {

    val attachment: Array<String?>
        get() = arrayOf(description, distanceThreshold.toString(), placeId)

    constructor(eddystoneBeacon: EddystoneBeacon): this() {
        eddystoneBeacon.description?.let { this.description = it }
        this.distanceThreshold = eddystoneBeacon.distanceThreshold
        this.placeId = eddystoneBeacon.placeId
    }
}
```

What are the risks?

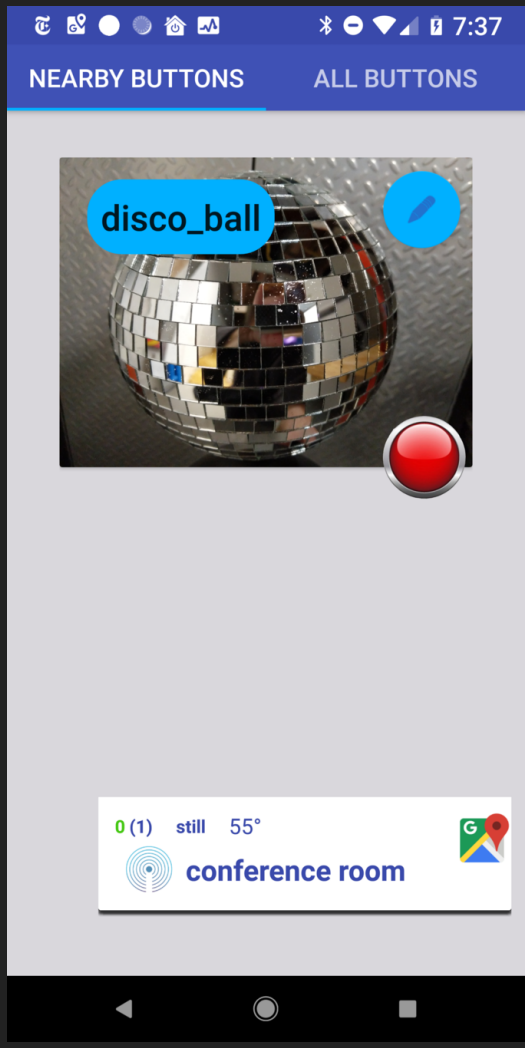
- Productivity hit from learning curve
- IDE auto-convert is imperfect
- You touch code, you break it

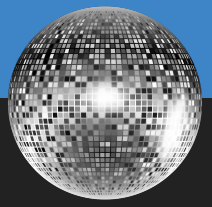


RoboButton Project

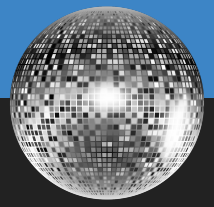
- Control nearby things with your phone

- Particle Photon Micro
- Bluetooth Beacons





Syntactic Kotlin



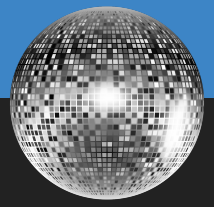
Mutable Property

```
var description = "a new beacon"  // inside BeaconAttachment.kt class
```

- Used elsewhere ...

```
...
beaconAttachment.getDescription();  // 'a new beacon'
beaconAttachment.setDescription("this room");
```


```
...
beaconAttachment.description  // 'a new beacon'
beaconAttachment.description = "this room"
```



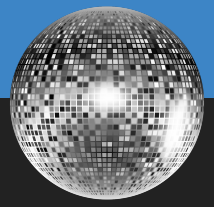
Read-Only Property

```
val description = "a new beacon"  // inside BeaconAttachment.kt class
```

- Used elsewhere ...

```
...
beaconAttachment.getDescription();  // 'a new beacon'
```

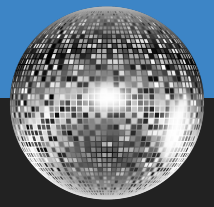

```
...
beaconAttachment.description  // 'a new beacon'
```



Type Inference


```
String description = "a new beacon";
```

```
var description = "a new beacon"
```

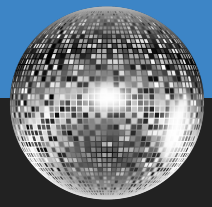


Compile-time NULL checking


```
String placeId = null;
```


```
Optional<String> placeId = Optional.empty();
```

```
var placeId: String? = null
```

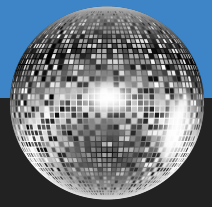


- We've declared our String 'nullable'

```
var placeId: String? = null
```

- Used elsewhere ...

```
...

placeId.length          // compile-time error

placeId!!.length        // "double bang" - compiles, but throws NPE

placeId?.length         // "safety" - may return null

placeId?.length ?: 0    // "elvis" - never returns null
```

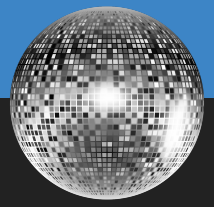


- What if we KNOW placeId is not null!

```
var placeId: String? = null

...

// do something here to define placeId

placeId!!.length
placeId!!.toLowerCase() // need to do !! every time
```

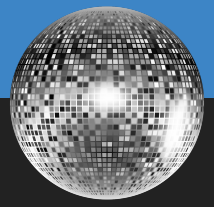
- Disable all null checking for a Property

```
var placeId: String  // compiler error, need to initialize

lateinit var placeId: String  // compiles!

...

placeId.length  // easier syntax, but may throw 'Not Initialized'
```



Object Construction & Initialization


```
public class BeaconAttachment {

    private String description;
    private int distanceThreshold;
    private String placeId;

    public BeaconAttachment() {
        this.description = "a new beacon";
        this.distanceThreshold = 5;
        this.placeId = null;
    }

    public BeaconAttachment(String description) {
        this.description = description;
        this.distanceThreshold = 5;
        this.placeId = null;
    }
...
```

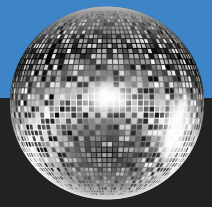


- Class declaration can include constructor
- Default and named arguments

```
class BeaconAttachment (var description: String = "a new beacon",
                        var distancedThresholed: int = 5,
                        var placeId: String? = null) {
```

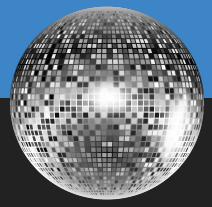
- Elsewhere in the code

```
var attachment = BeaconAttachment() // description = "a new beacon",
                                    // distanceThreshold = 5,
                                    // placeId = null

var attachment = BeaconAttachment("Office", 3) // placeId = null

// distanceThreshold = 5,  customerId = null
var attachment = BeaconAttachment(description = "Kitchen")
```



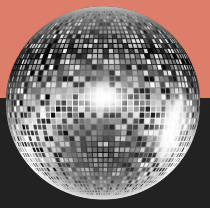
- Primary constructor cannot contain code
- init{} block is used for constructor logic
- Constructor args are also Properties

```
class BeaconAttachment (var description: String = "a new beacon",
                        var distancedThresholed: int = 5,
                        var placeId: String? = null) {

    init {
        initializeAttachment()
    }

    ...
}
```



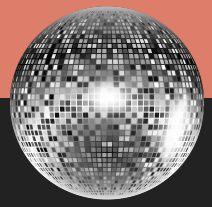
- Property vs Field
- Property without state of its own

```
class BeaconAttachment (var description: String = "a new beacon",
                        var distancedThresholed: int = 5,
                        var placeId: String? = null) {
...


val asAttachment: Array<String?>
    get() = arrayOf(description,
                    distanceThreshold.toString(),
                    placeId)

...

beaconAttachment.description
beaconAttachment.asAttachment
```



How to begin?

Let's get on the dance floor!



Great Documentation

https://kotlinlang.org/docs/reference/android-overview.html

Searchable Language Reference

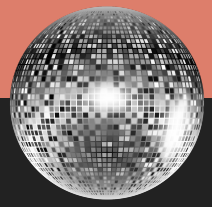
https://kotlinlang.org/docs/kotlin-docs.pdf

Style Guide

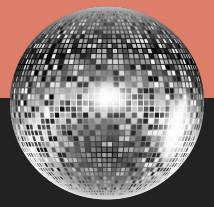
https://android.github.io/kotlin-guides/style.html

Online Sandbox and Tutorial

https://goo.gl/egTMsB

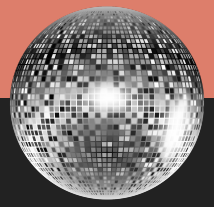


- Large Legacy Projects will be hybrid
- Legacy Java code can remain
- New features can be written in Kotlin
- Small projects can be completely converted



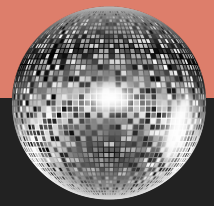
Install Kotlin Gradle Plugin

```
// file: RoboButton/build.gradle

buildscript {
    ext.kotlin_version = '1.2.21'
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.0.0'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}
```



Essential Kotlin support for Android

```
// file: RoboButton/App/build.gradle

...

apply plugin: 'kotlin-android'                // basic support
apply plugin: 'kotlin-android-extensions'  // view injection
apply plugin: 'kotlin-kapt'                   // Dagger

...

implementation "org.jetbrains.kotlin:kotlin-stdlib-jre7:$KOTLIN_VERSION"
```



Let's convert my RoboButton Project!

Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help — Sun 5:21 PM

roboButton - roboButton - [~/development/roboButton]

roboButton

app

Project

roboButton ~/development/roboButton

- .gradle
- .idea
- 3dEnclosures
- app
- arduino
- build
- captures
- gen
- gradle
- schematics
- .gitignore
- .pairs
- .travis.yml
- _config.yml
- build.gradle
- gradlew
- local.properties
- README.md
- roboButton.iml
- settings.gradle

External Libraries

roboButton

allprojects{} repositories{} flatDir{}

```
allprojects {
    repositories {
        mavenCentral()
        mavenLocal()
        maven {
            url 'https://oss.sonatype.org/content/repositories/snapshots/'
        }
        flatDir {
            dirs 'libs'
            dirs 'aars'
        }
        jcenter()
        maven { url
        maven {
            url "htt
        }
    }
    ext {
        currentGradleVersion = "2.14.1"
        currentBuildToolsVersion = "26.0.1"
        currentSupportLibVersion = "26.0.2"
        currentCompileSdkVersion = 26
        currentMinSdkVersion = 23
        currentTargetSdkVersion = 26
    }
}

subprojects {
    buildscript {
        repositories {
            mavenCentral()
            mavenLocal()
            maven {
                url 'https://oss.sonatype.org/content/repositories/snapshots/'
            }
            maven { url "https://jitpack.io" }
```

Convert Java to Kotlin

Converting Java to Kotlin (40 files) - pass...

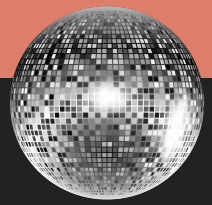
Cancel

.../fragments/ButtonDetailsDialogFragment.java

1: Project | Build Variants | 2: Favorites | 7: Structure | Captures | Gradle | Kotlin Bytecode | Device File Explorer

3: Find | 4: Run | TODO | Android Profiler | 6: Logcat | 9: Version Control | Terminal | 0: Messages | Event Log | Gradle Console

Gradle build finished in 13s 426ms (moments ago) — 10:23 LF UTF-8 Git: roboButtonBeforeKotlin Context: <no context>



- Java files are replaced with Kotlin
- Revision history is lost

```
On branch roboButtonBeforeKotlin
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    app/src/main/java/com/ndipatri/iot/roboButton/RBApplication.java
        new file:   app/src/main/java/com/ndipatri/iot/roboButton/RBApplication.kt
        deleted:    app/src/main/java/com/ndipatri/iot/roboButton/activities/MainActivity.java
        new file:   app/src/main/java/com/ndipatri/iot/roboButton/activities/MainActivity.kt
        deleted:    app/src/main/java/com/ndipatri/iot/roboButton/activities/RoboBaseActivity.java
        new file:   app/src/main/java/com/ndipatri/iot/roboButton/activities/RoboBaseActivity.kt
        deleted:    app/src/main/java/com/ndipatri/iot/roboButton/activities/ViewNearbyBeaconsActivity.java
```

- Most converted classes will need repairs

Idiomatic Kotlin for Android

Dependency Injection - Dagger

- Auto conversion fails

```
class ButtonDetailsDialogFragment : DialogFragment() {

    @Inject   // thanks to 'kotlin-kapt' plugin
    var particleCloudHelper: ParticleCloudHelper  // compiler error

    init {
        RBApplication.instance.graph.inject(this)
    }

...
```

Dependency Injection - Dagger

```
class ButtonDetailsDialogFragment : DialogFragment() {

    @Inject
    lateinit var particleCloudHelper: ParticleCloudHelper

    init {
        RBApplication.instance.graph.inject(this)
    }
...

   particleCloudHelper.updateButton(changedButton)  // no need for '?.' or '!!'
```

View Injection

- More Java ceremony


```
public class MainActivity extends RoboBaseActivity {
...
    @BindView(R.id.buttonViewPager)
    ViewPager buttonViewPager;
...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main)

        ButterKnife.bind(this);
    }

...

buttonViewPager.addOnPageChangeListener(...)
```

View Injection

- 'kotlin-android-extensions' plugin

```
package com.ndipatri.iot.roboButton.activities
...
import kotlinx.android.synthetic.main.activity_main.*  // activity_main.xml
...

class MainActivity : RoboBaseActivity() {

...

buttonViewPager.addOnPageChangeListener(...)
```

Decompilation

- Kotlin compiles down to Java bytecode
- bytecode can be decompiled to Java
- Kotlin can be decompiled to Java!

Decompilation

Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help — Sun 8:29 PM

MainActivity.decompiled.java - SolarMonitor - [~/development/androidEsoteric/SolarMonitor]

kotlinDecompiled › MainActivity.decompiled.java — app

MainActivity.kt | MainActivity.decompiled.java

MainActivity › updateStatusViews()

```
143
144 protected void handleConfigureFragmentDismiss() { this.resetToSteadyState(); }
147
148 private final void updateStatusViews() {
149    MainActivity.USER_STATE var10000 = this.userState;
150    if(this.userState == null) {
151       Intrinsics.throwUninitializedPropertyAccessException("userState");
152    }
153
154    TextView var1;
155    ProgressBar var2;
156    FloatingActionButton var3;
157    switch(MainActivity$WhenMappings.$EnumSwitchMapping$0[var10000.ordinal()]) {
158    case 1:
159       var1 = (TextView)this._$_findCachedViewById(id.mainTextView);
160       Intrinsics.checkExpressionValueIsNotNull(var1, "mainTextView");
161       ExtensionsKt.show(var1).setText((CharSequence)this.getString( resId: 2131165295));
162       var1 = (TextView)this._$_findCachedViewById(id.detailTextView);
163       Intrinsics.checkExpressionValueIsNotNull(var1, "detailTextView");
164       ExtensionsKt.hide((View)var1);
165       var2 = (ProgressBar)this._$_findCachedViewById(id.refreshProgressBar);
166       Intrinsics.checkExpressionValueIsNotNull(var2, "refreshProgressBar");
167       ExtensionsKt.gone((View)var2);
168       var3 = (FloatingActionButton)this._$_findCachedViewById(id.scanFAB);
169       Intrinsics.checkExpressionValueIsNotNull(var3, "scanFAB");
170       ExtensionsKt._show(var3);
171       var3 = (FloatingActionButton)this._$_findCachedViewById(id.loadFAB);
172       Intrinsics.checkExpressionValueIsNotNull(var3, "loadFAB");
173       ExtensionsKt._hide(var3);
174       var3 = (FloatingActionButton)this._$_findCachedViewById(id.configureFAB);
175       Intrinsics.checkExpressionValueIsNotNull(var3, "configureFAB");
176       ExtensionsKt._hide(var3);
177       break;
178    case 2:
179       var1 = (TextView)this._$_findCachedViewById(id.mainTextView);
180       Intrinsics.checkExpressionValueIsNotNull(var1, "mainTextView");
181       ExtensionsKt.show(var1).setText((CharSequence)this.getString( resId: 2131165305));
```

3: Find | 4: Run | TODO | 6: Logcat | 9: Version Control | Terminal | 0: Messages | Event Log | Gradle Console

Gradle build finished in 9s 768ms (12 minutes ago) — 151:68 LF UTF-8 Git: master Context: <no context>

Decompile to find magic

```kotlin
package com.ndipatri.iot.roboButton.activities
...
import kotlinx.android.synthetic.main.activity_main.*  // activity_main.xml
...

class MainActivity : RoboBaseActivity() {

...

buttonViewPager.addOnPageChangeListener(...)
```

```java
((ViewPager)this._$_findCachedViewById(id.buttonViewPager))
      .addOnPageChangeListener(...);
```

- Uses 'cached' findViewById()
- Can only use after 'setContentView()'

Extension Functions

- Make local changes to a 3rd party API


```
refreshProgressBar.setVisibility(INVISIBLE);
mainTextView.setVisibility(VISIBLE);
detailTextView.setVisibility(INVISIBLE);
```

```
refreshProgressBar.hide()
mainTextView.show()
detailTextView.hide()
```

Extension Functions


```
public class ViewHelper {
...
   public static final TextView show(TextView receiver) {
      receiver.setVisibility(View.VISIBLE);

      return receiver;
   }

...

ViewHelper.show(someTextView).setText("a beaon");
```

Extension Functions

```
// com.ndipatri.iot.roboButton.Extensions.kt

...

fun TextView.show(): TextView {

    visibility = View.VISIBLE // this.visibility = View.VISIBLE

    return this
}

...

 mainTextView.show().text = "a beacon"
```

- "Receiver" is a localized "this"

Pratical Kotlin with a Disco Ball

Questions?

https://github.com/ndipatri/roboButton

```
Copyright 2018 Nicholas DiPatri

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```