

CS194A

Android Programming Workshop

Lecture 2: Sep 29, 2021
Rahul Pandey

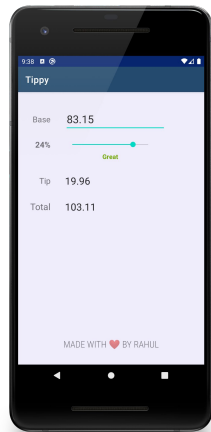
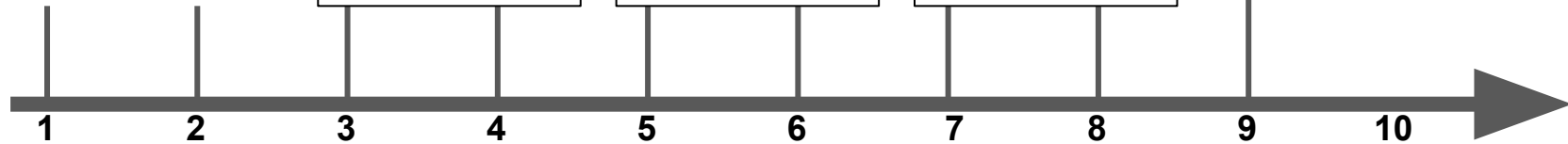


Outline

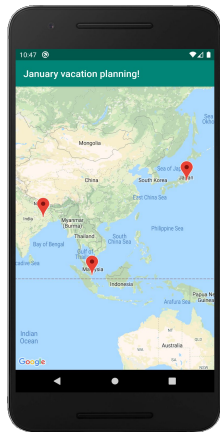
- Logistics
- Bigger Number app
- Kotlin overview

Outline

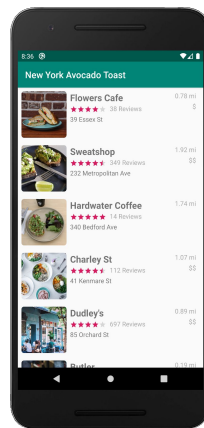
- **Logistics**
- Bigger Number app
- Kotlin overview



Assn 1: Tip Calculator



Assn 2: Google Maps



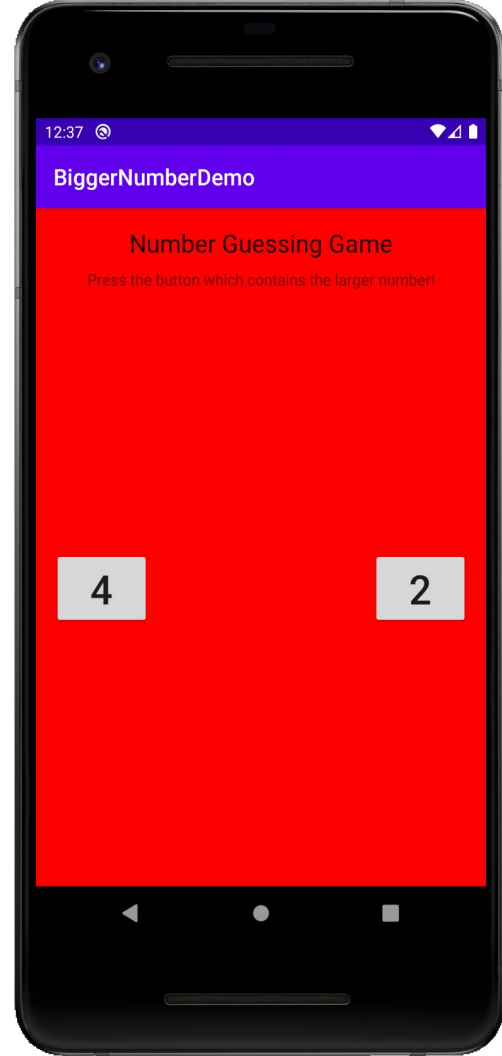
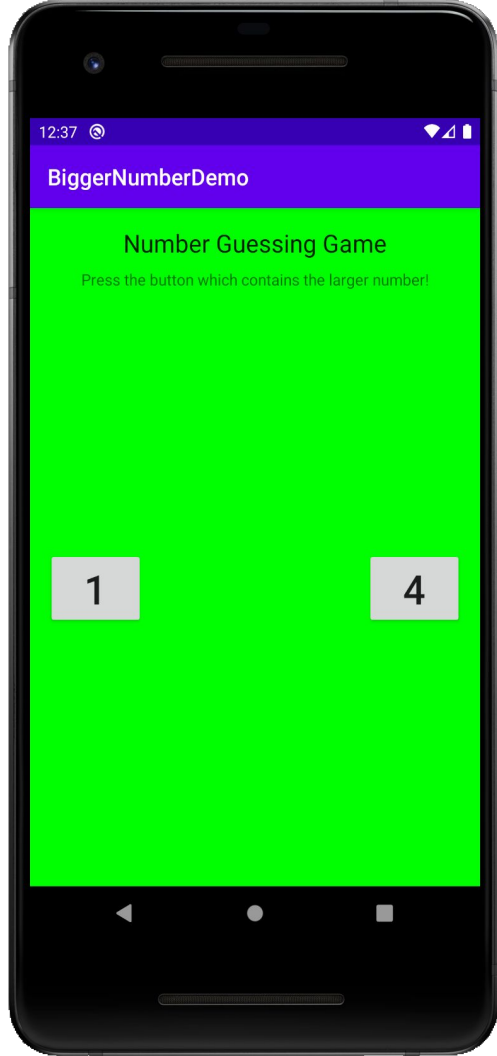
Assn 3: Yelp Clone

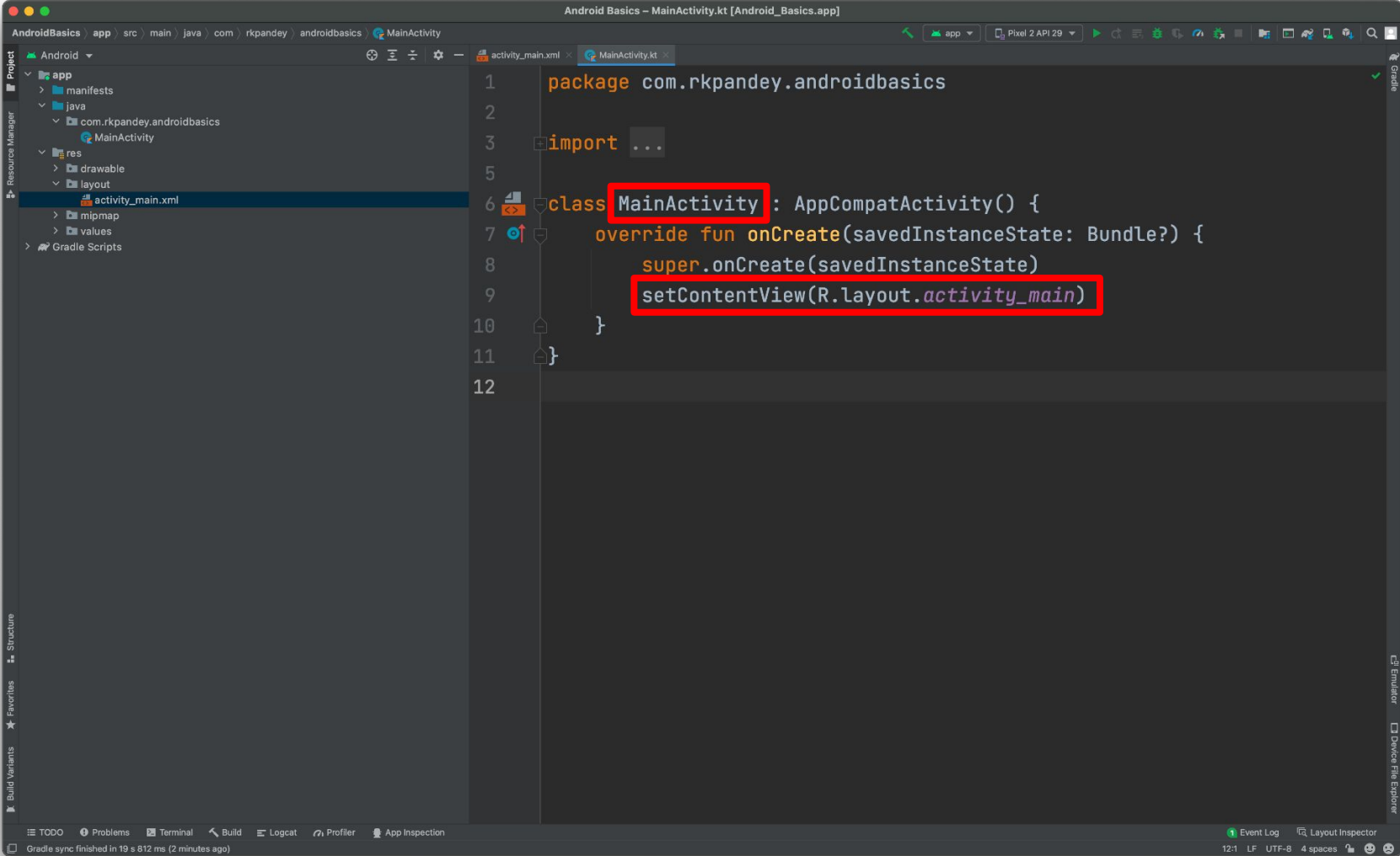
Industry panel discussion

Week #

Outline

- Logistics
- **Bigger Number app**
- Kotlin overview





1 package com.rkpandey.androidbasics

3 import ...

6 class MainActivity : AppCompatActivity() {

7 override fun onCreate(savedInstanceState: Bundle?) {

8 super.onCreate(savedInstanceState)

9 setContentView(R.layout.activity_main)

10 }

11 }

12

Activity \longleftrightarrow Layout Communication

```
activity_main.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6
7     <TextView
8         android:id="@+id/tvHelloWorld"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Hello World!" />
12
13 </LinearLayout>
```

```
MainActivity.kt
1 package com.rkpandey.androidbasics
2
3 import ...
4
5
6
7 class MainActivity : AppCompatActivity() {
8     private lateinit var tvHelloWorld: TextView
9
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14         tvHelloWorld = findViewById(R.id.tvHelloWorld)
15         tvHelloWorld.text = "Updated!"
16     }
17 }
```


Outline

- Logistics
- Bigger Number app
- **Kotlin overview**

What is Kotlin?

- Programming language developed by JetBrains, [1.0 released in 2016](#)
 - In contrast, Java was released in 1995 by Sun Microsystems
- Statically typed, object oriented
- Incorporated lessons from the widespread use of Java
- Top priority for Kotlin is **pragmatism**
 - Inter-operates with Java by running on the JVM

Why Kotlin- “follow the community”

- Google I/O 2017: Kotlin support for Android
 - Google I/O 2019: Android will be “Kotlin-first”
- More and more companies are adopting Kotlin
- Potential growth outside the world of Android as well
- Kotlin will make you a better Java developer

Why Kotlin- “follow the tech”

- Concise
- Safe
- Interoperable
- Tool-friendly
- (Android) Kotlin extensions

Get last element of a list

```
val myList = listOf(42, 91, 51)
```

```
myList.get(myList.size - 1)
```

```
myList[myList.size - 1]
```

```
myList[myList.lastIndex]
```

```
myList.last()
```

Why Kotlin [[repo](#)]

- Type inference
- Immutability
- Null safety

Type inference

Java

```
String first = "Jane";  
String last = "Smith";  
int age = 35;
```

```
List<String> friends =  
Arrays.asList("April",  
"John");
```

Kotlin

```
val first: String = "Jane"  
val last = "Smith"  
val age = 35
```

```
val friends = listOf("April", "John")
```

Immutability

Java

```
int age = 35;  
age += 1;
```

Java

```
List<String> friends =  
Arrays.asList("April",  
"John");  
  
friends = Arrays.asList("May",  
"Mark");  
  
friends[0] = "June"
```

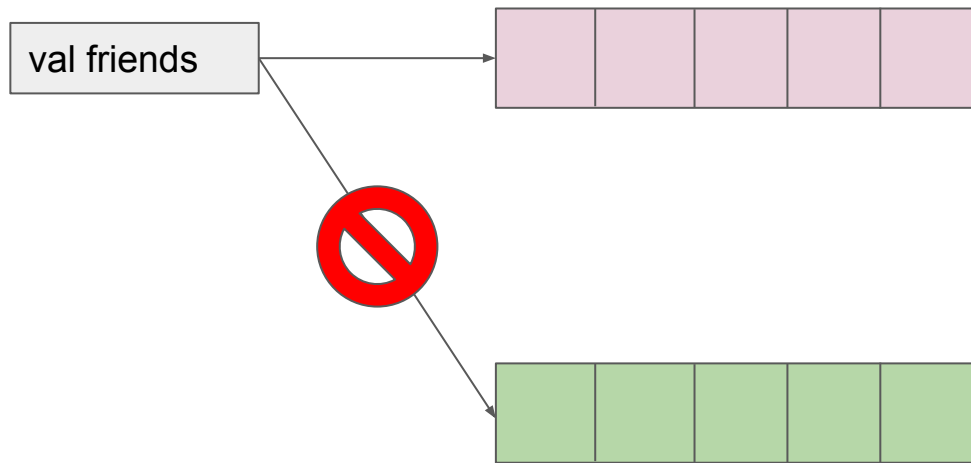
Kotlin

```
var age = 35  
age += 1
```

Kotlin

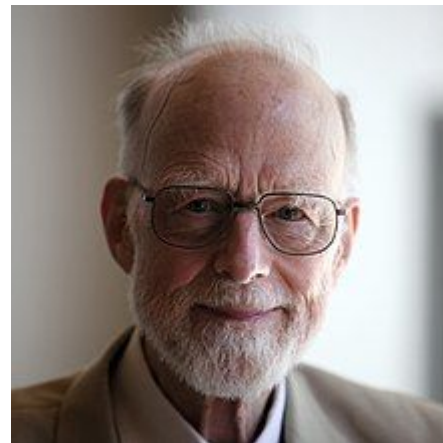
```
val friends = mutableListOf("April",  
"John")  
  
friends[0] = "May"
```


val vs var



Nullability

- “I call it my billion-dollar mistake”
 - Tony Hoare
- Kotlin adds null checks at compile-time rather than run-time



Nullability in Kotlin

Java

```
String name = null;
int length = name.length();    // runtime crash
if (name != null) {
    int length = name.length(); // ok
}
```

Kotlin

```
val bad: String = null        // compiler error!
val name: String? = null     // ok
val lengthBad = name.length() // compiler error!
val length1 = name?.length() ?: 0
if (name != null) {
    val length2 = name.length()
}
```

Practice: [gist](#)

```
data class ExamResult(val name: String, val score: Int)
```

Extension functions

Extend a class with new functionality without having to inherit from the class

```
fun <T> List<T>.secondToLast(): T? {  
    if (this.size < 2) {  
        return null  
    }  
    return this[this.size - 2]  
}
```

Other nice features

- Data classes
- Single expression functions
- Automatic setters and getters
- Wrapping Java primitives into objects:
 - Byte, Short, Int, Long, Float, Double, Char, Boolean
 - Still uses primitive under the hood for performance

Prep for next week

- Code LeetCode problems with Kotlin
 - [Two Sum](#)
 - [Water container](#)

- Android Studio [shortcuts video](#)