

Android Development

In this lesson, you will learn about the development environment that you can set up to begin creating applications in Android.

WE'LL COVER THE FOLLOWING ^

- Android Studio IDE
- Programming languages

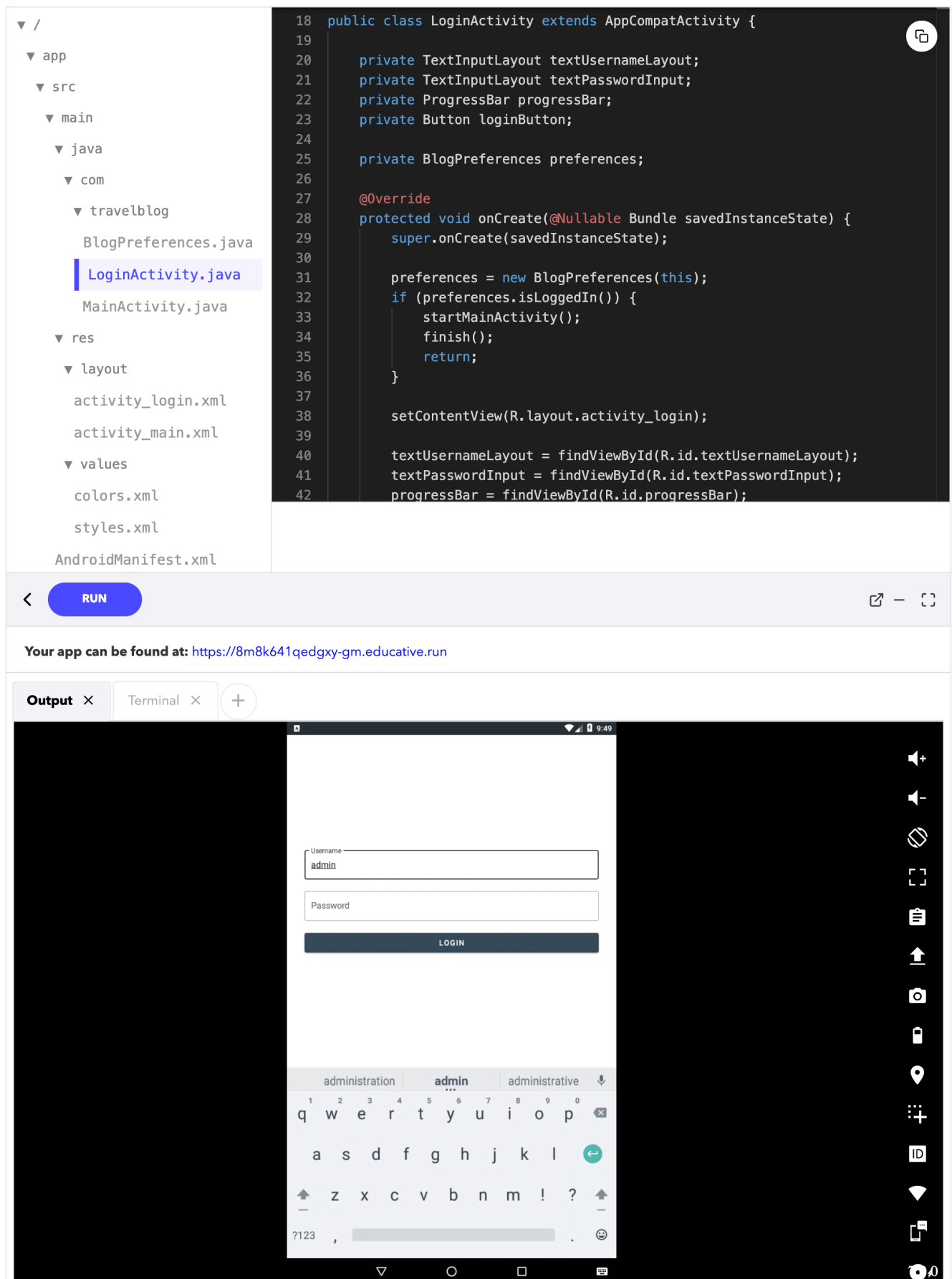
Android Studio IDE

Google provides great tooling to build Android applications, and the main one is [Android Studio IDE](#). It's based on IntelliJ IDEA and provides tons of features, like:

- [Visual layout editor](#)
- [Fast emulator](#)
- [Intelligent code editor](#)
- [Realtime profilers](#)
- [Flexible build system](#)

With Android Studio, we can test applications either on [emulator or directly on the device](#).

Through this course, we will use [educative.io](#) Android widget which provides a pre-configured environment to build and launch Android applications in your browser, so you don't need to do any setup.



Screenshot of educative Android widget

However, if you want, you can [install Android Studio](#), create the project and copy the code provided in this course into your project.

Programming languages

There are 3 programming languages and 1 markup language which are used in Android development:

- Java/Kotlin - source code
- Groovy - build scripts
- XML - layout structure and resources

Java is the official language for Android development and is supported by Android Studio. Java classes are compiled into a proprietary bytecode format and run on Android Runtime (ART) or older Dalvik, a specialized virtual machine (VM) designed for Android.

Android supports all Java 7 language features and a subset of Java 8 language features that vary by platform version.

Here is an example of Android `MainActivity.java` file. We will discuss it in more details in the following lessons:

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        TextView mainTextView = findViewById(R.id.mainTextView);  
        mainTextView.setText("Hello educative.io");  
    }  
}
```

MainActivity.java

Kotlin has been a second official language for Android Development since 2017. Kotlin is an Android-compatible language that is concise, expressive, and designed to be type- and null-safe.

Through this course, we will use Java language, since it's more widely known.

Here is an example of Android `MainActivity.kt` file so you can compare it to `MainActivity.java` file.

```
public class MainActivity : AppCompatActivity {
```

```

override fun onCreate(savedInstanceState: Bundle) {
    super.onCreate(savedInstanceState)

    setContentView(R.layout.activity_main)

    val mainTextView = findViewById<TextView>(R.id.mainTextView)
    mainTextView.text = "Hello educative.io"
}
}

```

MainActivity.kt

Android build system is powered by [Gradle](#). It is an open-source, build-automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based, domain-specific language (DSL).

Here is an example of a Gradle build file. We will discuss Gradle in more details in the following lessons:

```

apply plugin: 'com.android.application'
android {
    defaultConfig {
        applicationId "com.travelblog"
        versionCode 1
        versionName "1.0"
    }
}
dependencies {
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.android.material:material:1.1.0-alpha10'
}

```

build.gradle

XML is heavily used in Android to declare a layout for the user interface, strings, dimensions, and some other resource files.

Here is an example of a layout that shows text in the middle of the screen. We will discuss it in more details in the following lessons:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```
android:layout_height="match_parent"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_main.xml

In the next chapter, we will cover some basic Android concepts and create our first “Hello World” screen.