Android App Development Basics

Mike Swierenga

Goals

- Get familiar with Android Studio
- Learn how to write an Android app with the Kotlin programming language
- Build a list app where you can save custom items

If you haven't already, and you want to follow along:
Download Android Studio 3 at https://developer.android.com/studio/

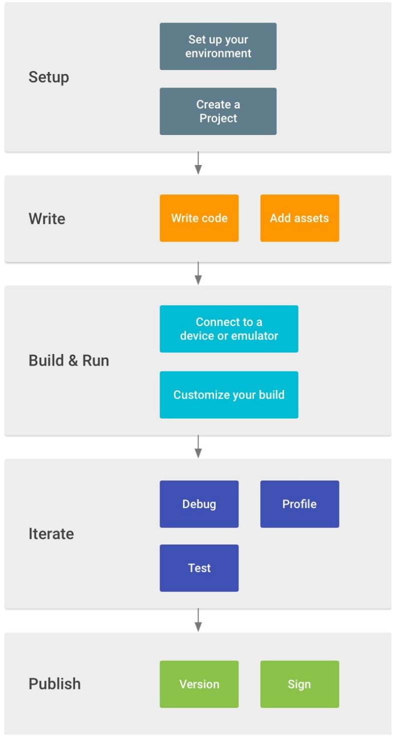
The code and slides for this workshop are available to view or download on GitHub at https://github.com/mierenga/nhacks-BasicListApp

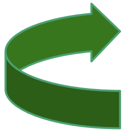
Android Development Background

- Why **Kotlin**?
    - It is the newest Android language with a lot of advantages over the traditional **Java**
    - It simplifies code in ways that were not invented yet when Java was first written
    - It is an official language of Android, and many developers are switching their apps to it
    - Downside is many online Android examples will be written in Java
- Why **Android Studio**?
    - It is the official IDE* for Android Development
    - Its adds a lot of tools to help you when writing code and drawing screen layouts
- Other options
    - Java, JavaScript, C+#, C++ and other languages can all be used to write Android apps
        - Check out the chart and links in the appendix if you want to learn more

*Integrated Development Environment, where you develop software like Android or iOS apps

Developer Workflow

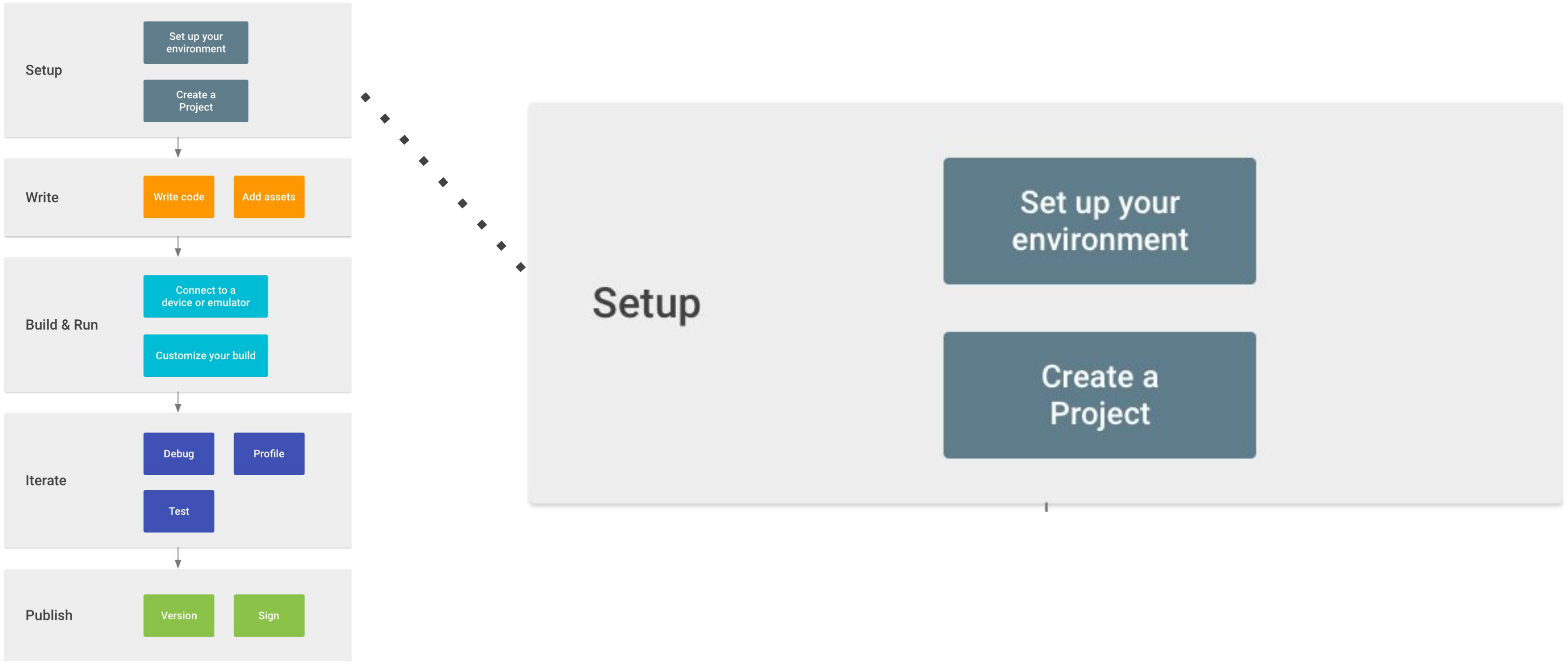




- Setup
- Write
- Build & Run

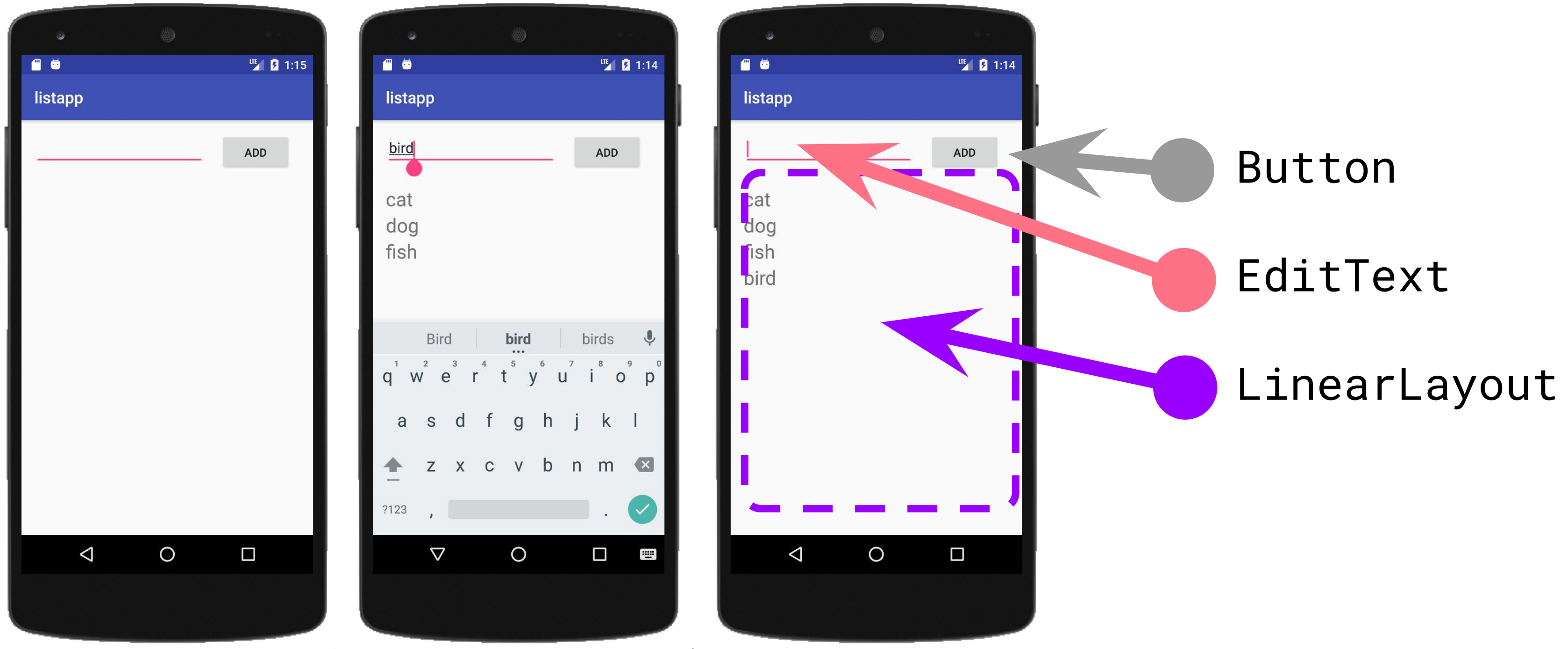
https://developer.android.com/studio/workflow

Android Developer Workflow: **Setup**



https://developer.android.com/studio/workflow

BasicListApp

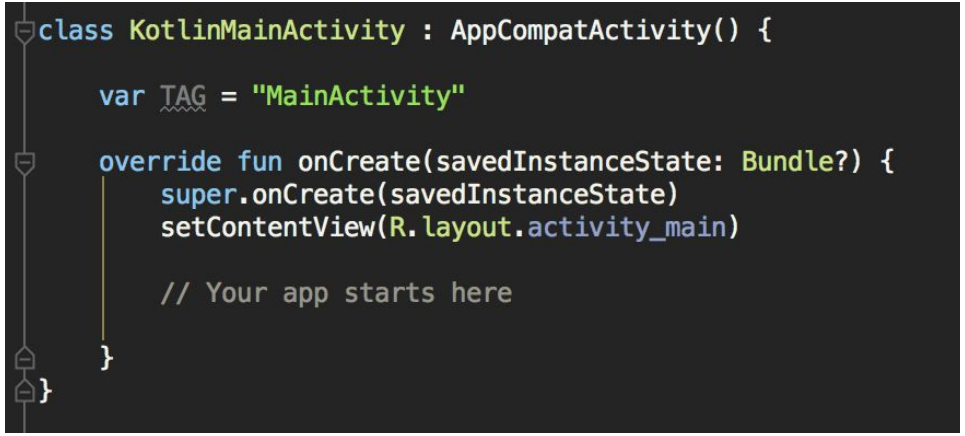


Activity Components

Activity Code

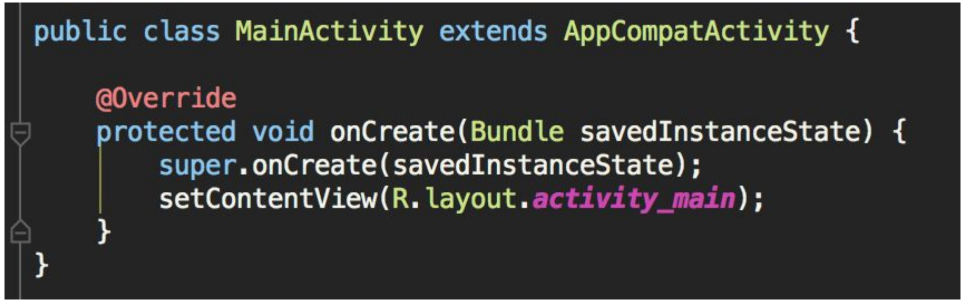
- Kotlin or Java code
- `app/java` folder
- Defines behavior of the app

`MainActivity.kt`



```kotlin
class KotlinMainActivity : AppCompatActivity() {

    var TAG = "MainActivity"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Your app starts here

    }
}
```

`MainActivity.java`



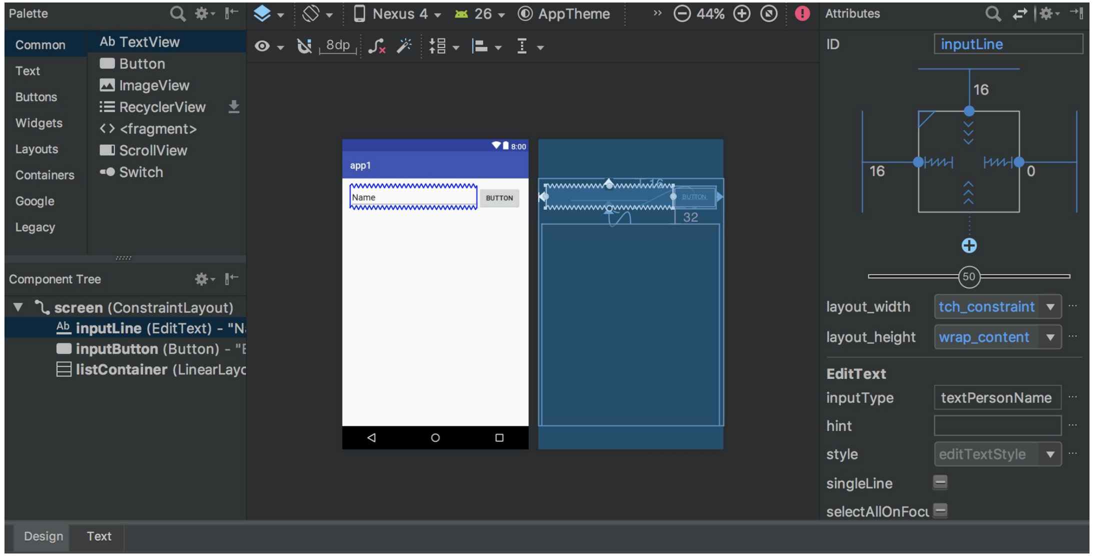
```java
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Activity Components

Activity Layout

- Visual designer
- `res/layout` folder

e.g., **`activity_main.xml`**



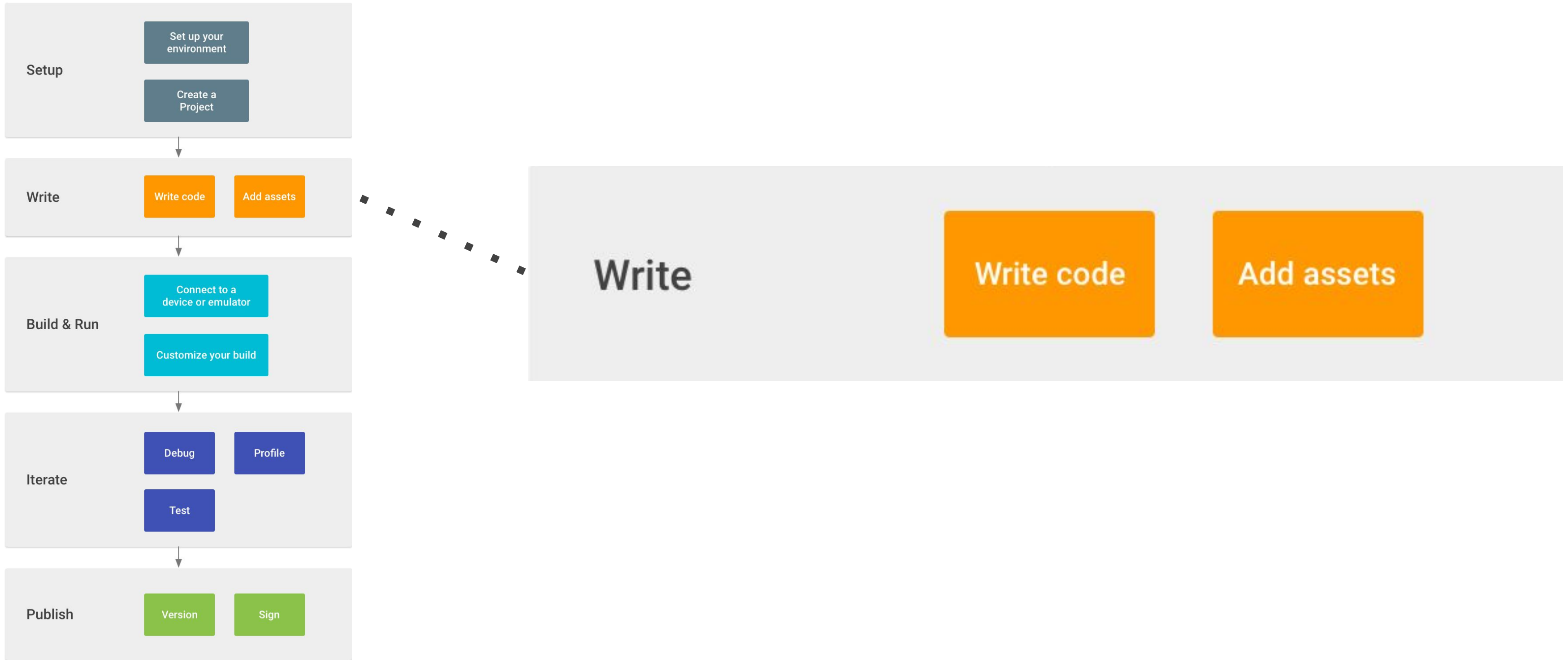
Component Tree
- screen (ConstraintLayout)
  - inputLine (EditText) - "N
  - inputButton (Button) - "E
  - listContainer (LinearLayo

ID	inputLine
layout_width	tch_constraint
layout_height	wrap_content

EditText

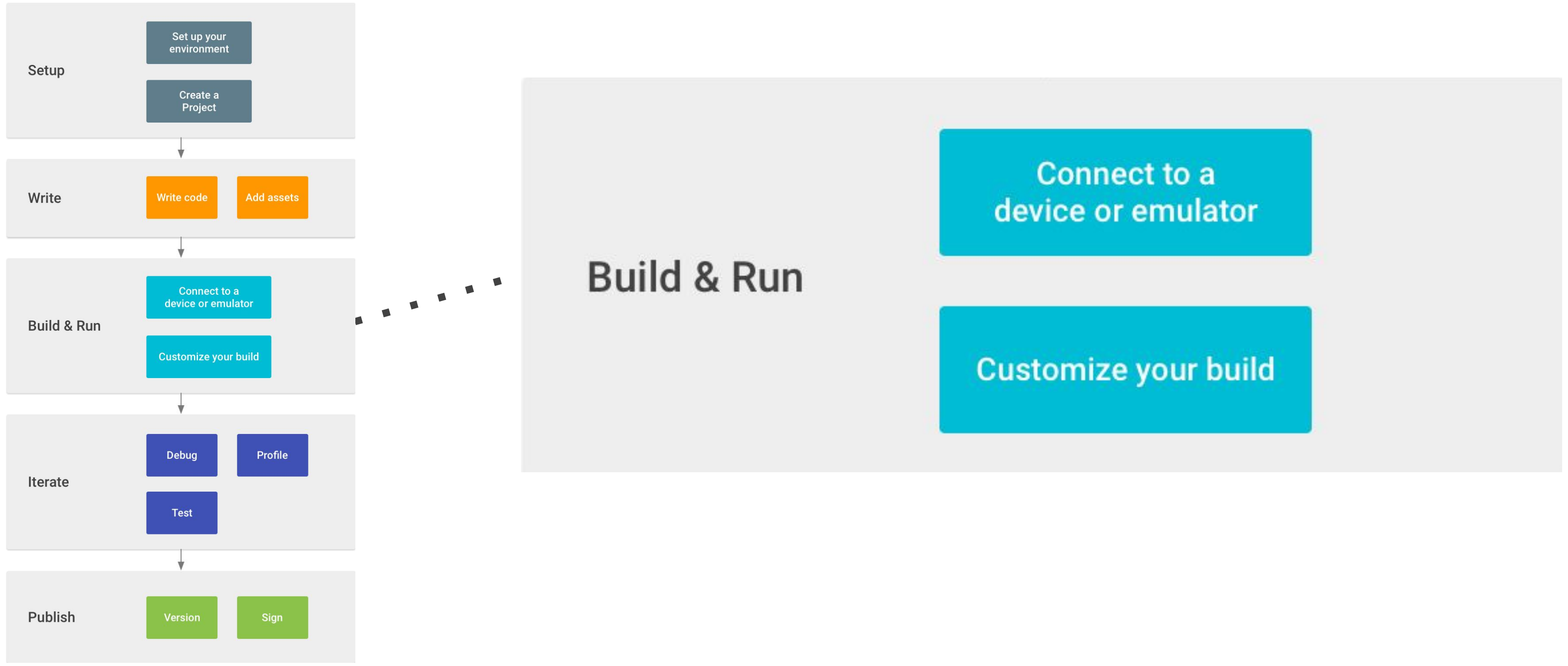
inputType	textPersonName
hint	
style	editTextStyle
singleLine	
selectAllOnFocu	

Android Developer Workflow: **Write**



https://developer.android.com/studio/workflow

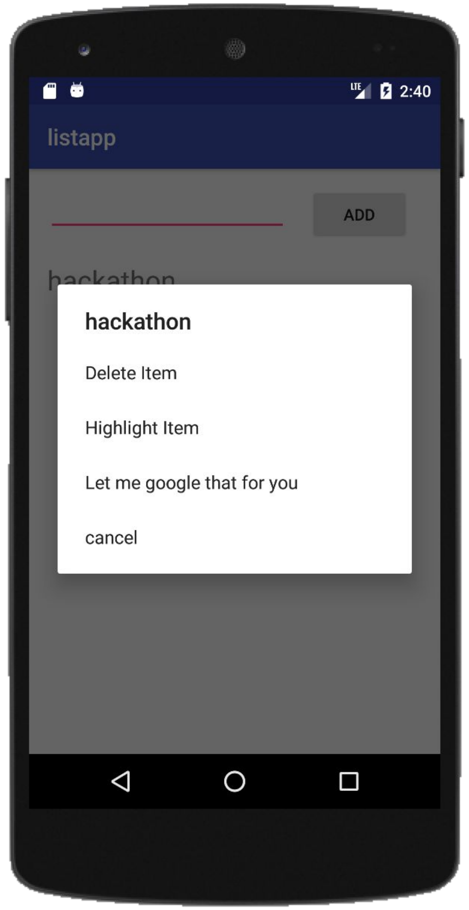
Android Developer Workflow: **Build & Run**



https://developer.android.com/studio/workflow

AlertDialog Menu

- Delete item
- Highlight item
- Search for the item on google



Appendix

Android Documentation and Guides

- Android Studio: https://developer.android.com/studio/
- Official guides: https://developer.android.com/guide/index.html
- App fundamentals: https://developer.android.com/guide/components/fundamentals
- Build your first app: https://developer.android.com/training/basics/firstapp/
- App samples: https://developer.android.com/samples/

Samples with more advanced features

Kotlin

- Using the camera
- Using the video camera
- Using a CardView
- Using a RecyclerView
- Scheduling a background task
- Display a PDF
- Bluetooth between devices
- Accelerometer

Java

- App Architecture Overview
- Basic Notifications
- Using the camera
- Using the video camera
- Use the network to fetch HTML
- Bluetooth between devices
- Bluetooth advertisements
- Bluetooth chat between devices
- Accelerometer
- Touch gesture detection

Android Programming Languages

	Native	Compiles to Native	Share Code with iOS app	Developer Documentation
Java	✓			Android Official
Kotlin	✓			Kotlin for Android
C# (Xamarin)		✓	✓	
JavaScript with HTML/CSS		✓ React Native ⊖ PhoneGap	✓	• React Native • PhoneGap

Android Device Developer Mode

Developer mode is required to run your apps on your device.

You can enable it by following this guide:
https://developer.android.com/studio/debug/dev-options

Java Basics

- Always remember to end each statement with the required **semicolon**;
- Always pay attention to **capitalization**, it is very important
- **//** Text after two slashes is **not code**, it is just a comment for humans
- **/*** Text between these symbols
  is also a **comment */**
- Java code uses structures that are called **objects**

Java Objects

- **Objects** have a format for how they are declared and assigned

```
<type> <name>;                      // empty value
<type> <name> = <value>;            // initialized to value
```

e.g.,

```
EditText editText;                  // empty value
String TAG = “MainActivity”;        // initialized to “MainActivity”
```

Object class <type> examples : **`String`**, **`MainActivity`**, **`AppCompatActivity`**, **`EditText`**, **`Button`**, **`LinearLayout`** `and hundreds more…`

Primitive <type> examples : **`int`**, **`boolean`**, **`float`**, **`double`**, **`char`**

Java Objects

- **Objects** have special abilities that are unlocked with a **period.**
- These abilities are called **methods**

```
String text = “  some thing  ”;    // text is a String object
text = text.trim();                // trim() is one of its abilities
```

```
“  some thing  “ becomes “some thing”

“ “ becomes “”
```

Java Methods

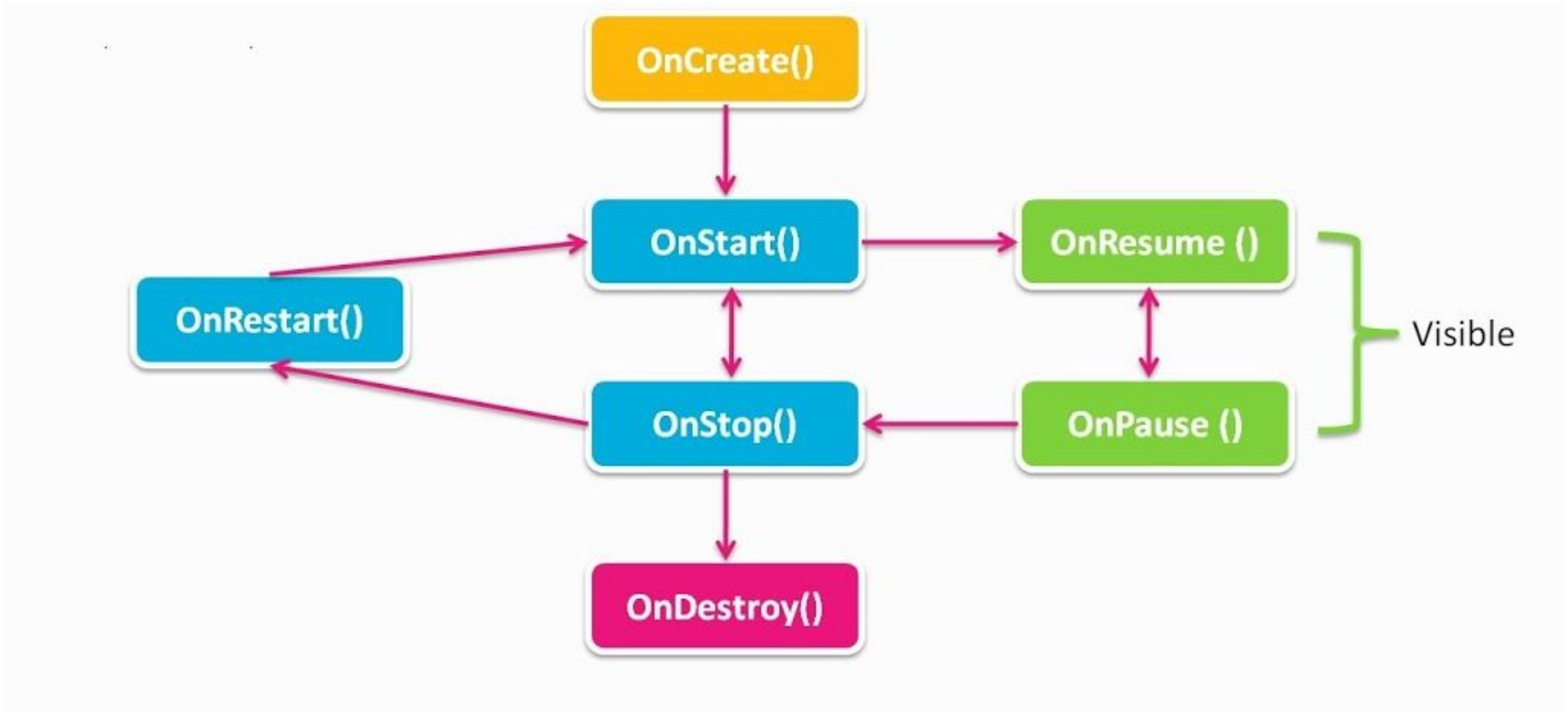
```
<return> <name> (<argtype> <argname>, … ) {
    <body>
}
```

```
e.g.,
```

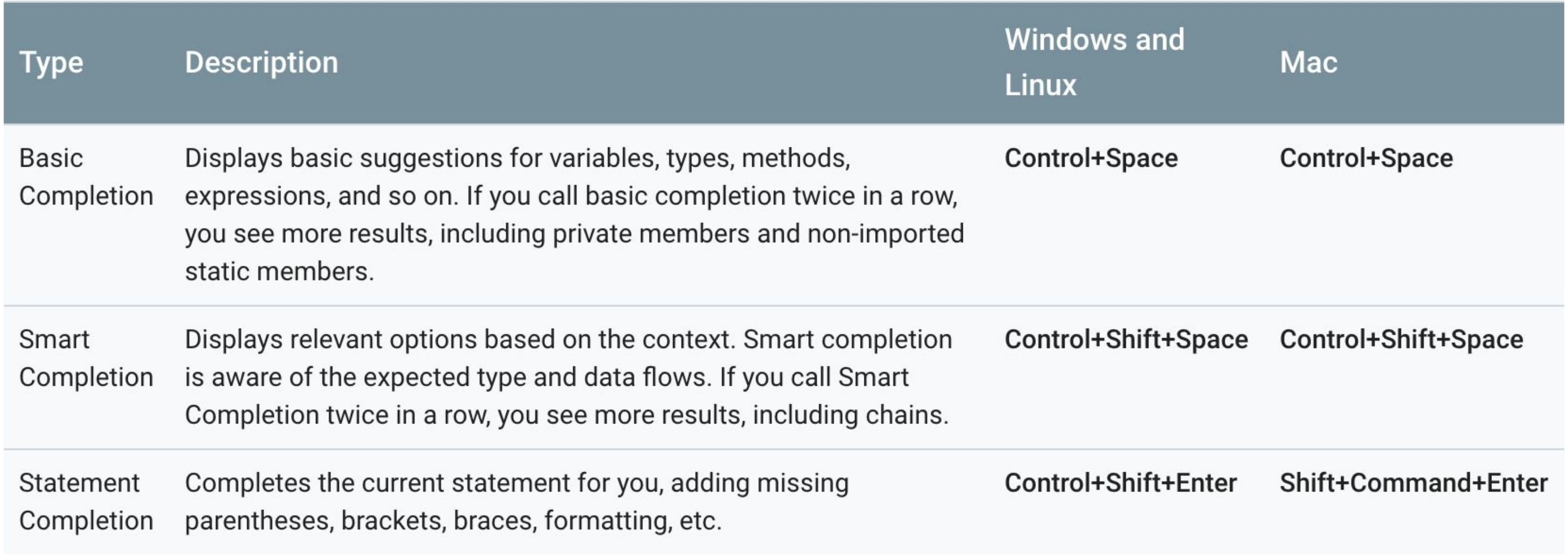
```
void addNewItem(String text) {
    TextView item = new TextView(this);
    item.setText(text);
    inputList.addView(item);
}
```

Return type `void` means it is empty (it doesn't return anything).
Many methods in Java have `void` as the return type.

Activity Lifecycle Methods



Android Studio Code Completion



Type	Description	Windows and Linux	Mac
Basic Completion	Displays basic suggestions for variables, types, methods, expressions, and so on. If you call basic completion twice in a row, you see more results, including private members and non-imported static members.	**Control+Space**	**Control+Space**
Smart Completion	Displays relevant options based on the context. Smart completion is aware of the expected type and data flows. If you call Smart Completion twice in a row, you see more results, including chains.	**Control+Shift+Space**	**Control+Shift+Space**
Statement Completion	Completes the current statement for you, adding missing parentheses, brackets, braces, formatting, etc.	**Control+Shift+Enter**	**Shift+Command+Enter**