XML and Gradle

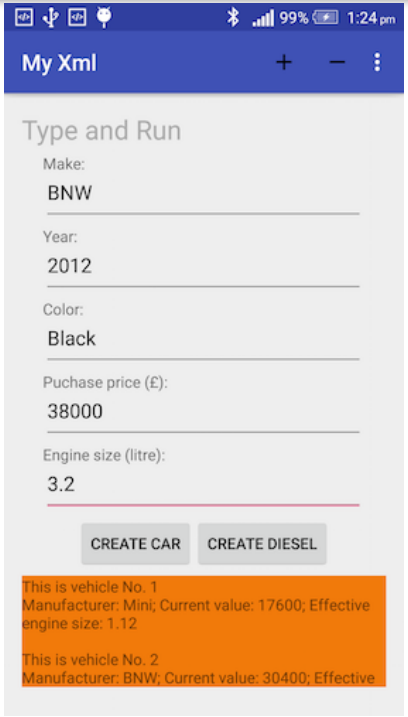
Last week?

- Any questions?
- Java keywords, onClick methods, interface?

LAB 1

- XML basics
- Android resources
- More Java
- IntelliJ features

The app i.e. outcome

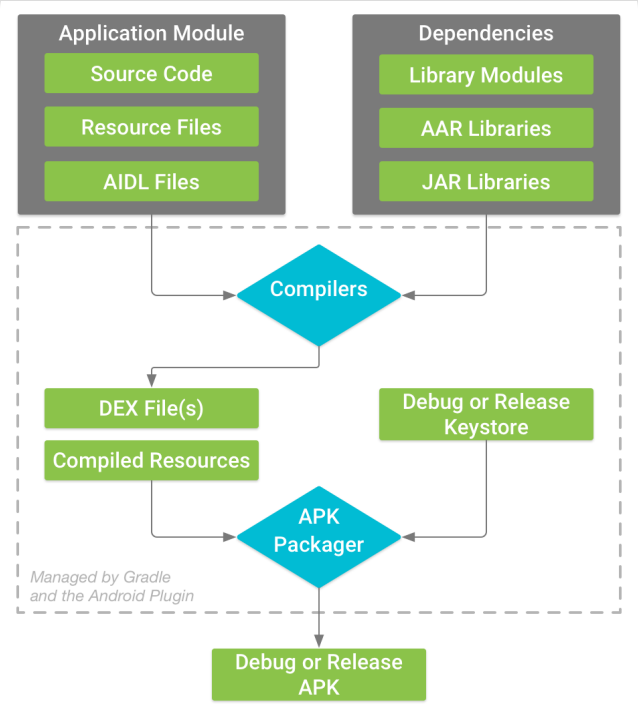


- app bar (old ‘action bar’)
- Scrollable
- Resources using XML:
  - Provide
  - Access
- ArrayList, StringBuilder, Wrapper class, autoboxing

Git

- Use Git to push and pull, not copy/paste
- Clone module's github
  - git clone https://github.com/covcom/300CEM.git
- For this current lab:
  - Download zip
  - svn checkout

Android build process



1. The compilers convert your source code into DEX (Dalvik Executable) files
2. The APK Packager combines the DEX files and compiled resources into a single APK
3. The APK Packager signs your APK using either the debug or release keystore
4. The packager uses the zipalign tool to optimize your app

https://developer.android.com/studio/build/index.html

Resource types

- Layout Resource
- String Resources
- Drawable Resources
- Menu Resource
- Style Resource
- Animation Resources
- Color State List Resource
- More Resource Types

https://developer.android.com/guide/topics/resources/providing-resources.html

Resource types

```
MyProject/
    src/
        MyActivity.java
    res/
        drawable/
            graphic.png
        layout/
            main.xml
            info.xml
        mipmap/
            icon.png
        values/
            strings.xml
```

```xml
<dimen name="margin_left">19dp</dimen>
<dimen name="margin_top">5dp</dimen>
<dimen name="margin_right">20dp</dimen>
```

XML vs HTML

- XML document carry data along with their description.
- Not have predefined tags.
- Must have closing tag.
- Case sensitive

- HTML document formats and displays web page data.
- Predefined tags (i.e. standards)
- May not have closing tag.
- Not case sensitive.

XML syntax

- Prolog
- Case sensitivity
- xmlns is URI not URL
- TAGS come in pairs
- Attribute values in quotes
- White spaces preserved

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.jianhuayang.myxml.MainActivity">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="80dp"
        android:background="?attr/colorAccent"
        android:elevation="4dp"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Type and Run"
        android:textColor="@android:color/darker_gray"
        android:textSize="24sp" />
```

http://www.w3schools.com/xml/xml_syntax.asp

Accessing resources

- In code:
  - Using a static integer from a sub-class of your R class, such as: R.string.hello
- In XML:
  - Using a special XML syntax that also corresponds to the resource ID defined in your R class, such as:@string/hello

https://developer.android.com/guide/topics/resources/accessing-resources.html

Dimensions

- dp - density-independent Pixels
- sp - scale-independent Pixels
- pt - points = 1/72 of an inch
- px – pixels
- mm - millimeters
- in - inches

https://developer.android.com/guide/topics/resources/more-resources.html#Dimension

Dimensions

XML file saved at `res/values/dimens.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="textview_height">25dp</dimen>
    <dimen name="textview_width">150dp</dimen>
    <dimen name="ball_radius">30dp</dimen>
    <dimen name="font_size">16sp</dimen>
</resources>
```

This application code retrieves a dimension:

```
Resources res = getResources();
float fontSize = res.getDimension(R.dimen.font_size);
```

This layout XML applies dimensions to attributes:

```
<TextView
    android:layout_height="@dimen/textview_height"
    android:layout_width="@dimen/textview_width"
    android:textSize="@dimen/font_size"/>
```

https://developer.android.com/guide/topics/resources/more-resources.html#Dimension

Strings

XML file saved at `res/values/strings.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello!</string>
</resources>
```

This layout XML applies a string to a View:

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello" />
```

This application code retrieves a string:

```
String string = getString(R.string.hello);
```

- getString()
- GetResources().getString()

https://developer.android.com/guide/topics/resources/string-resource.html#String

Images

- Similar to String resources
- Can be added in two ways:
  - Hard disks (lab demo)
  - Right-click in Android Studio
- **android-drawable-importer-intellij-plugin**
  - https://github.com/winterDroid/android-drawable-importer-intellij-plugin

App bar

- Instructions on GitHub may not be valid if you use your own example
- Depends on the template, this may be there already
- Make sure you use NoActionBar theme
- In layout XML, Add Toolbar form support library
- Add resource XML for menus
- In Activity Java code, set toolbar in onCreate() method
- In Activity Java code, inflate menu

Menus

- Customized xmlns
  - xmlns:app=http://schemas.android.com/apk/res-auto

```
<item
    android:id="@+id/menu_add"
    android:icon="@drawable/ic_add_black_24dp"
    android:orderInCategory="1"
    android:title="@string/menu_add"
    app:showAsAction="always|withText" />
```

```
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    switch (id) {
        case R.id.menu_add:
            addVehicle();
            return true;
        case R.id.menu_clear:
            return clearVehicleList();
        case R.id.action_settings:
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

More Java

- ArrayList
  - The diamond syntax
  - `private ArrayList<Vehicle> vehicleList = new ArrayList<>();`
- StringBuilder
  - `private StringBuilder outputs;`
- Wrapper class
  - Autoboxing
  - `private static Double depreciation;`

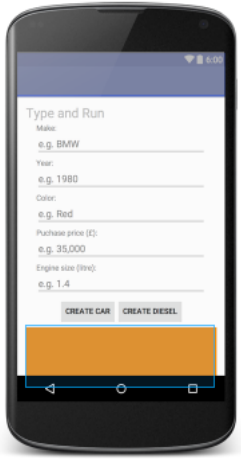
IntelliJ features

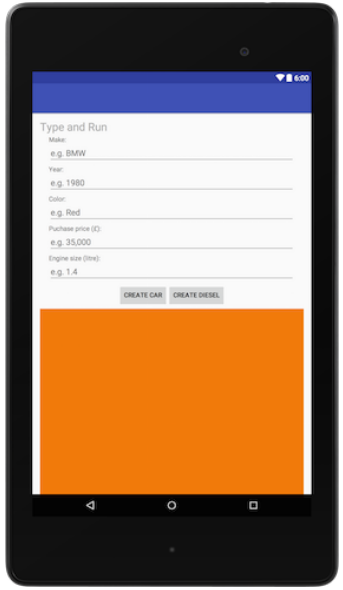
- Import projects
  - Open build.gradle
- Module settings 🡐 Manifest.xml
- Refactor
- Search everywhere
- Image import

LAB 2

- Style and themes
- Screen sizes
- Manifest file
- Gradle system
- More Java

The app i.e. outcome





- Styles/themes
- Alternative resources
- Manifest and build system

String Array

XML file saved at `res/values/strings.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
    </string-array>
</resources>
```

This application code retrieves a string array:

```
Resources res = getResources();
String[] planets = res.getStringArray(R.array.planets_array);
```

https://developer.android.com/guide/topics/resources/string-resource.html#String

Define styles

- Inheritance

```
<style name="CodeFont.Red">
    <item name="android:textColor">#FF0000</item>
</style>
```

```
<style name="CodeFont.Red.Big">
    <item name="android:textSize">30sp</item>
</style>
```

- Style properties

```
<style name="Numbers">
  <item name="android:inputType">number</item>
  ...
</style>
```

```
<EditText
    style="@style/Numbers"
    ... />
```

Access styles

Style Resource

```
<?xml version="1.0" encoding="utf-8"?>
<EditText
    style="@style/CustomText"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello, World!" />
```

https://developer.android.com/guide/topics/resources/style-resource.html

Referencing style attributes

```
<EditText id="text"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textColor="?android:textColorSecondary"
    android:text="@string/hello_world" />
```

?[<package_name>:][<resource_type>/]<resource_name>

https://developer.android.com/guide/topics/resources/accessing-resources.html

More examples

```
<activity android:theme="@android:style/Theme.Dialog">
```

```
@[<package_name>:]<resource_type>/<resource_name>
```

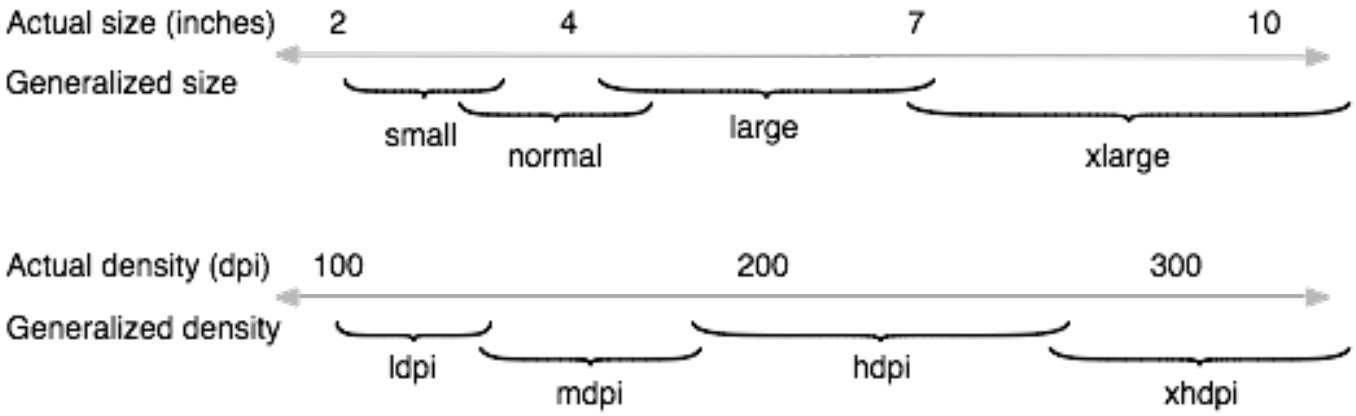
NOT to be confused with style attributes

```
android:layout_marginLeft="10dp"
android:layout_marginTop="32dp"
android:text="Small Text"
android:textAppearance="?android:attr/textAppearanceSmall"
```

```
?[<package_name>:][<resource_type>/]<resource_name>
```

http://developer.android.com/guide/topics/resources/accessing-resources.html

Screen sizes and orientations



- *xlarge* screens are at least 960dp x 720dp
- *large* screens are at least 640dp x 480dp
- *normal* screens are at least 470dp x 320dp
- *small* screens are at least 426dp x 320dp

https://developer.android.com/guide/practices/screens_support.html

Screen sizes and orientations

Type	Device	Platform	Screen dimensions in cm		Aspect Ratio	Width × Height dp	Width × Height px	Density
	Nexus 4	Android	4.7 in	3.2 × 4.0 in	5 : 3	384 × 640 dp	768 × 1280 px	2.0 xhdpi
	Nexus 5	Android	5.0 in	2.4 × 4.3 in	16 : 9	360 × 640 dp	1080 × 1920 px	3.0 xxhdpi
	Nexus 5X	Android	5.2 in	2.5 × 4.5 in	16 : 9	411 × 731 dp	1080 × 1920 px	2.6 xxhdpi
	Nexus 6	Android	6.0 in	2.9 × 5.2 in	16 : 9	411 × 731 dp	1440 × 2560 px	3.5 xxxhdpi
	Nexus 6P	Android	5.7 in	2.8 × 5.0 in	16 : 9	411 × 731 dp	1440 × 2560 px	3.5 xxxhdpi

https://design.google.com/devices/

Alternative resources

Create a new directory in res/ named in the form *<resources_name>-<config_qualifier>*.

```
res/
    drawable/
        icon.png
        background.png
    drawable-hdpi/
        icon.png
        background.png
```

Language and region	Examples:
	`en`
	`fr`
	`en-rUS`
	`fr-rFR`
	`fr-rCA`

smallestWidth	`sw<N>dp`
	
	Examples:
	`sw320dp`
	`sw600dp`
	`sw720dp`

https://developer.android.com/guide/topics/resources/providing-resources.html

Best practices

1. Use wrap_content, match_parent, or dp units when specifying dimensions in an XML layout file
2. Do not use hard coded pixel values in your application code
3. Do not use AbsoluteLayout (it's deprecated)
4. Supply alternative bitmap drawables for different screen densities

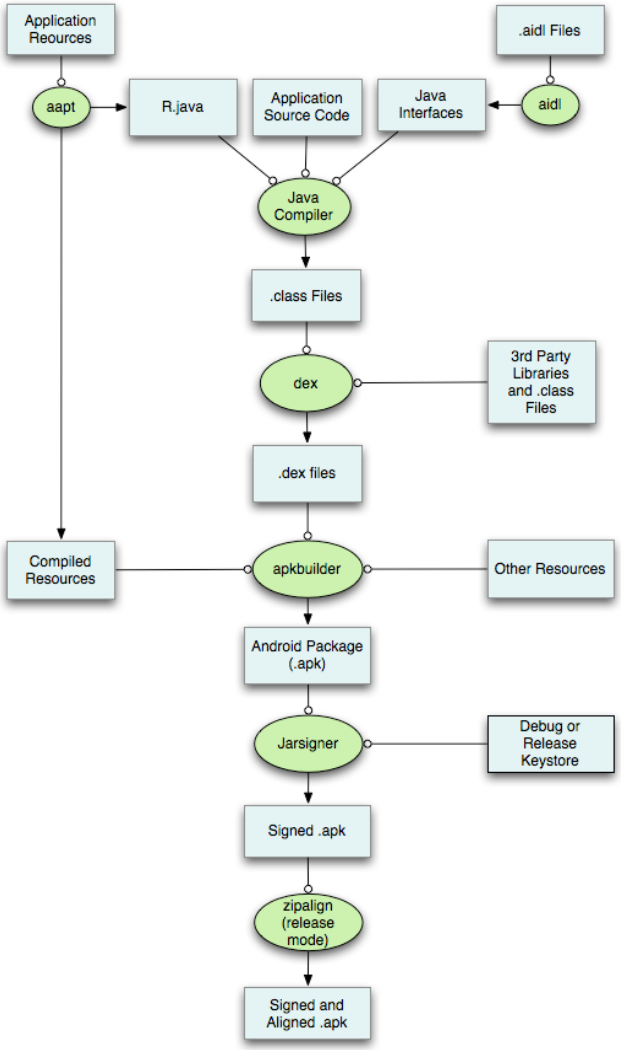
https://developer.android.com/guide/practices/screens_support.html#screen-independence

App manifest

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.jianhuayang.mycar">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- The file defines 'metadata' of the app.
- The application tag has several attributes i.e. icon, label, theme.
- There can be more than one activity tags in your app.
- In Activity name the leading dot '.' denotes the package attribute in 'manifest' tag.
- There can only be one launcher activity

Android build process



- Resource code generation
- Interface code generation
- Java compilation
- Byte code conversion
- Packaging
- Signing the package
- Package optimization

http://www.herongyang.com/Android/Project-Android-Application-Project-Build-Process.html

The gradle file

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 20
    buildToolsVersion "20.0.0"

    defaultConfig {
        applicationId "com.mycompany.myapplication"
        minSdkVersion 13
        targetSdkVersion 20
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
         debug {
            debuggable true
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:20.0.0'
    compile project(path: ':app2, configuration: 'android-endpoints')
}
```

- Task blocks
- Configuration block
- BuildTypes
- Proguard
- Local binary dependencies
- Remote binary dependencies (Maven coordinates group:name:version)

More Java

- HashMap

```
private Map<String, String> mapCarMaker = new HashMap<>();
mapCarMaker.put(manufacturers[i], descriptions[i]);
String vehicleDescription = mapCarMaker.get(v.getMake());
```

- Generic

```
private <T extends Number> Double depreciateAnything(T originalValue) {
    Double result;
    if (originalValue instanceof Double) {
        result = Math.round(originalValue.doubleValue() * 0.8 * 100) / 100.00;
    } else {
        result = originalValue.intValue() * 0.8;
    }
    return result;
}
```