Android Core Topics #1

“Learning” App

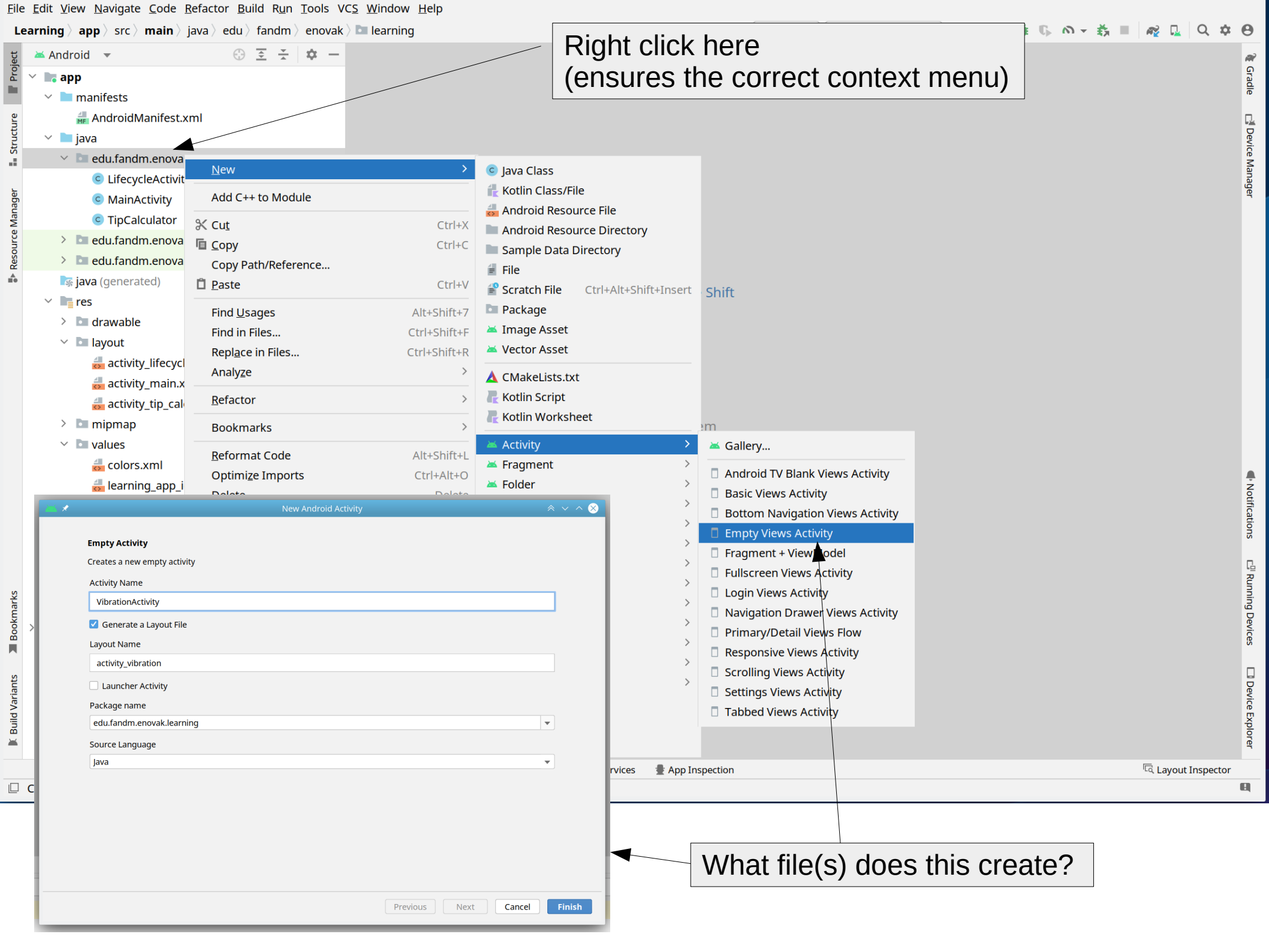
Wide Open

- Android Development is about learning APIs
- There are too many APIs!
  - https://developer.android.com/reference/packages
  - I will teach some
  - Some (most) you will teach yourself!
- “Learning app” as a workshop

To Do List

- Wrap-Up (*finish these first if we haven't already*)
  - Change Launcher icon
  - Launching another activity
  - Tip Calculator

- New Stuff
  - Launching another activity (passing data to it!)
  - ConstraintLayout Parent View color
  - Vibration API (easy)


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Learning app src main java edu fandm enovak learning
Right click here
(ensures the correct context menu)
Android
app
manifests
AndroidManifest.xml
java
edu.fandm.enova
LifecycleActivit
MainActivity
TipCalculator
edu.fandm.enova
edu.fandm.enova
java (generated)
res
drawable
layout
activity_lifecycl
activity_main.x
activity_tip_cal
mipmap
values
colors.xml
learning_app_i
New
Add C++ to Module
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+Shift+7
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Analyze
Refactor
Bookmarks
Reformat Code Alt+Shift+L
Optimize Imports Ctrl+Alt+O
Java Class
Kotlin Class/File
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Package
Image Asset
Vector Asset
CMakeLists.txt
Kotlin Script
Kotlin Worksheet
Activity
Fragment
Folder
Gallery...
Android TV Blank Views Activity
Basic Views Activity
Bottom Navigation Views Activity
Empty Views Activity
Fragment + ViewModel
Fullscreen Views Activity
Login Views Activity
Navigation Drawer Views Activity
Primary/Detail Views Flow
Responsive Views Activity
Scrolling Views Activity
Settings Views Activity
Tabbed Views Activity
New Android Activity
Empty Activity
Creates a new empty activity
Activity Name
VibrationActivity
Generate a Layout File
Layout Name
activity_vibration
Launcher Activity
Package name
edu.fandm.enovak.learning
Source Language
Java
Previous
Next
Cancel
Finish
What file(s) does this create?
App Inspection
Layout Inspector
Gradle
Device Manager
Notifications
Running Devices
Device Explorer
Project
Structure
Resource Manager
Bookmarks
Build Variants

XML File Changes

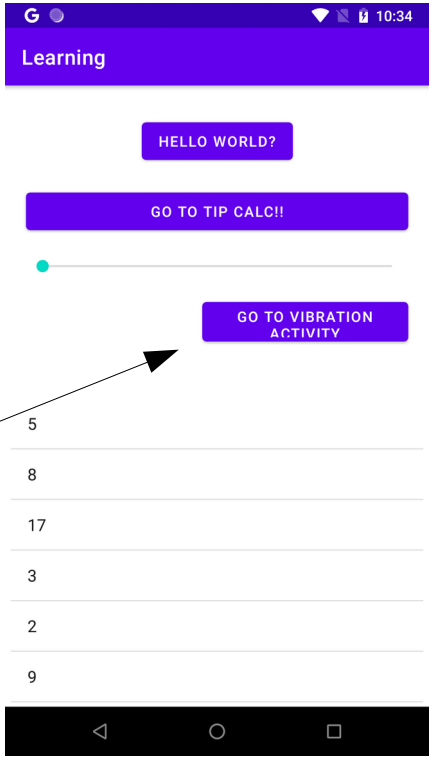
In order for the vibration API (java code on the next slides) to work at all the developer *must* put this "permission" line in their AndroidManifest.xml

This permission is shown to the user (a) when they are installing the app and (b) anytime via settings → Apps & notifications → X App

Unfortunately the permissions system isn't perfect.

AndroidManifest.xml

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

    <uses-permission android:name="android.permission.VIBRATE"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
```



activity_main.xml (below seekbar)

```
<Button
    android:id="@+id/vibrationBT"
    android:layout_width="200dp"
    android:layout_height="50dp"
    android:layout_margin="20dp"
    android:text="Go To Vibration Activity"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/randomSB" />
```

In your notes, draw a picture of this layout.

```xml
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/vibration_cl"
    tools:context=".VibrationActivity">

    <LinearLayout
        android:id="@+id/vibration_ll"
        app:layout_constraintTop_toTopOf="parent"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <!-- android:layout_margin will apply to
        the LL itself and not between the children -->

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView #1"
            android:textAlignment="center" />
            <!-- android:gravity="center" works here too -->

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView #2"
            android:textAlignment="center" />

    </LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView #3"
        android:background="@color/cardview_dark_background"
        android:textColor="@color/cardview_light_background"
        android:textSize="24sp"
        android:layout_margin="20dp"
        app:layout_constraintTop_toBottomOf="@id/vibration_ll"
        android:textAlignment="center" />

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

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

    Intent i = getIntent();
    if(i != null){
        int color = i.getIntExtra( name: "color",  defaultValue: 0);
        View cl = findViewById(R.id.vibration_cl);
        // gets the top-most parent view
        // View topMostView = cl.getRootView();

        cl.setBackgroundColor(color);
    }


    Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
    v.vibrate( milliseconds: 500);

    VibrationEffect ve = VibrationEffect.createOneShot( milliseconds: 750, VibrationEffect.DEFAULT_AMPLITUDE);
    ve.vibrate();
}
```

Call requires API level 26 (current min is 24): android.os.VibrationEffect#createOneShot

Surround with if (VERSION.SDK_INT >= VERSION_CODES.O) { ... } Alt+Shift+Enter More actions... Alt+Enter

```java
public static VibrationEffect createOneShot(
    long milliseconds,
    int amplitude
)
```

Create a one shot vibration.
One shot vibrations will vibrate constantly for the specified period of time at the specified amplitude, and then stop.

Params: milliseconds – The number of milliseconds to vibrate. This must be a positive number.
amplitude – The strength of the vibration. This must be a value between 1 and 255, or DEFAULT_AMPLITUDE.

Returns: The desired effect.

android.os.VibrationEffect

< Android API 33, extension level 3 Platform > (android.jar)

(30704 Verbose

Inside VibrationActivity.java

Inside Main Activity

Sends data to VibrationActivity

```java
public void launchOtherAct (View v) {
    Intent i = new Intent( packageContext: this, VibrationActivity.class);
    Random rnd = new Random();
    int color = Color.argb( alpha: 255, rnd.nextInt( bound: 255), rnd.nextInt( bound: 255), rnd.ne
    i.putExtra( name: "color", color);

    startActivity(i);
}
```

Emulator

VibrationActivity Fixed

In case onCreate is called from an app switch or some other way besides the button in MainActivity

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

    Intent i = getIntent();
    if(i != null){
        int color = i.getIntExtra( name: "color",  defaultValue: 0);
        View cl = findViewById(R.id.vibration_cl);
        // gets the top-most parent view
        // View topMostView = cl.getRootView();

        cl.setBackgroundColor(color);
    }


    Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
    if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
        VibrationEffect ve = VibrationEffect.createOneShot( milliseconds: 750, VibrationEffect.DEFAULT_AMPLITUDE);
        v.vibrate(ve);
    } else {
        v.vibrate( milliseconds: 500);
    }


}
```

Extract information sent from main activity

Android Nougat	New York Cheesecake	7.0	24	August 22, 2016	August 2019
		7.1 – 7.1.2	25	October 4, 2016	October 2019
Android Oreo	Oatmeal Cookie	8.0	26	August 21, 2017	January 2021
		8.1	27	December 5, 2017	October 2021
Android Pie	Pistachio Ice Cream[22]	9	28	August 6, 2018	January 2022
Android 10	Quince Tart[23]	10	29	September 3, 2019	February 2023
Android 11	Red Velvet Cake[23]	11	30	September 8, 2020	
Android 12	Snow Cone	12	31	October 4, 2021	
Android 12L	Snow Cone v2	12.1[a]	32	March 7, 2022	
Android 13	Tiramisu[25]	**13**	33	August 15, 2022	
Android 14	Upside Down Cake[26]	14	34	Q3 2023	—

Legend: Old version | Older version, still maintained | **Latest version** | Future release

Vibration API is different depending on which version of the Android OS the target device (phone) is running.

This is a very annoying, but very prevalent aspect of Android development.

Vibration Activity Layout

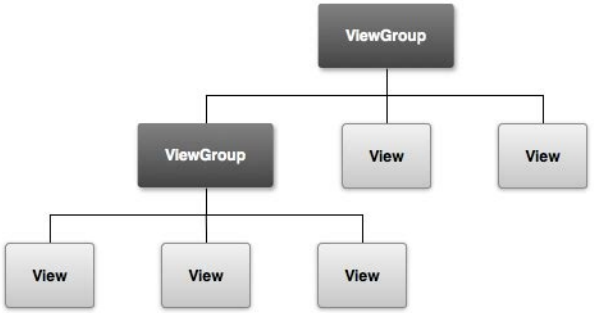
LinearLayout is a viewgroup containing textview #1 and textview #2

a findViewById() called via textview #1 cannot find any other views (besides it self) returning null

a findViewById() called via the linearlayout (the parent of the textviews) can only find textview #1 and textview #2 and the linearlayout itself

A findViewById() called via the ConstraintLayout can find anything here.

Use `anyview.getRootView()` on any view to go all the way up through the hierarchy to the top-most view / viewgroup



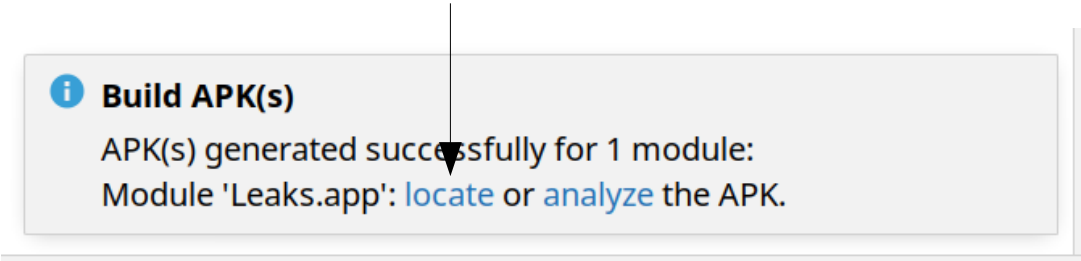

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/vibration_cl"
    tools:context=".VibrationActivity">

    <LinearLayout
        android:id="@+id/vibration_ll"
        app:layout_constraintTop_toTopOf="parent"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <!-- android:layout_margin will apply to
        the LL itself and not between the children -->

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView #1"
            android:textAlignment="center" />
            <!-- android:gravity="center" works here too -->

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView #2"
            android:textAlignment="center" />

    </LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView #3"
        android:background="@color/cardview_dark_background"
        android:textColor="@color/cardview_light_background"
        android:textSize="24sp"
        android:layout_margin="20dp"
        app:layout_constraintTop_toBottomOf="@id/vibration_ll"
        android:textAlignment="center" />

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

Arranges views in a vertical list

Creating an APK file

- An android app is packaged as a “.apk” file.
- Android Studio → Build → Build Bundle(s) / APK(s) → Build APK
- Click “locate” in the resulting pop window

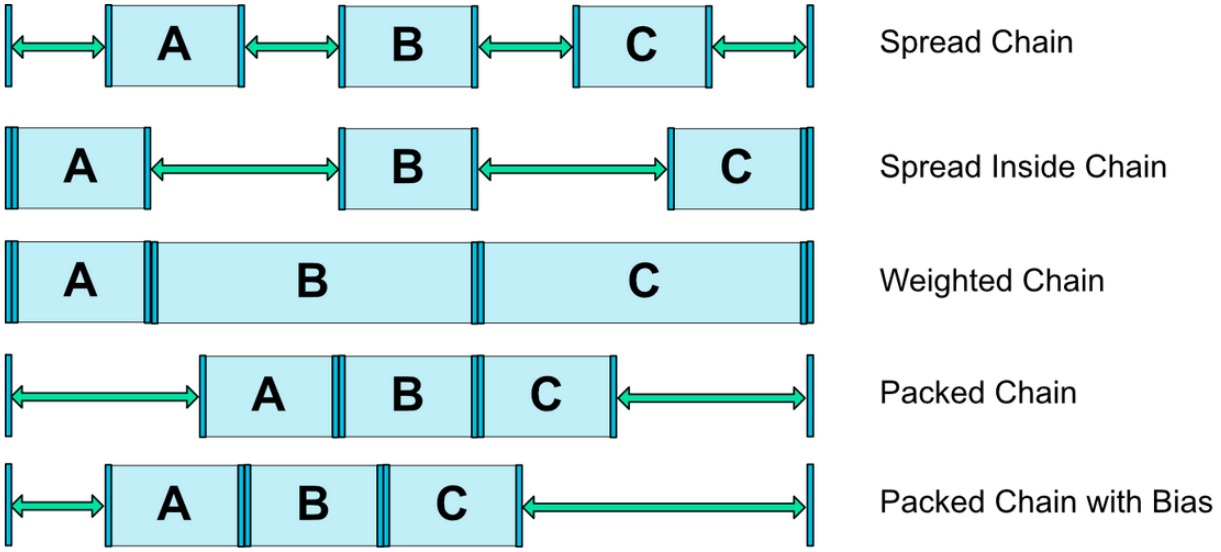
app/build/outputs/apk/debug/app-debug.apk



Spacing of Views in a ConstraintLayout

To remove space between two chained view, simply add the following line to the first item in the chain:

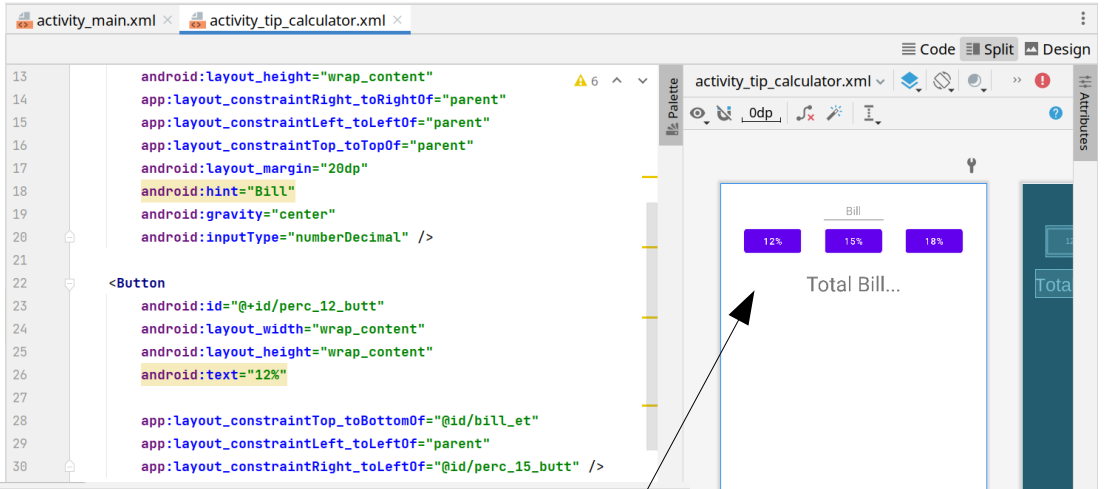
```
app:layout_constraintHorizontal_chainStyle="packed"
```



Another very useful layout control:
`android:layout_margin="20dp"`

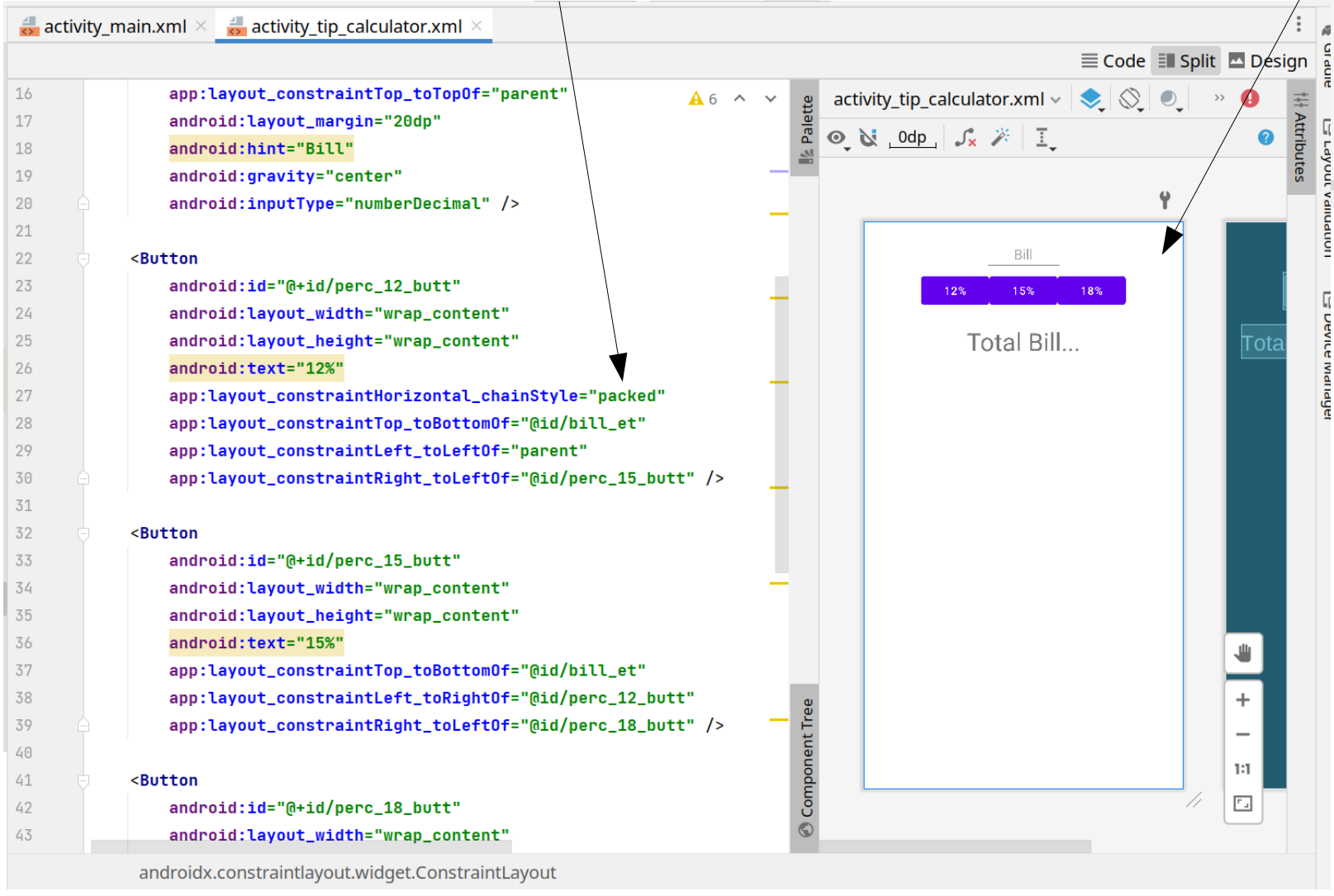
```
android:layout_marginTop="20dp"
android:layout_marginRight="20dp"
android:layout_marginBottom="20dp"
android:layout_marginLeft="20dp"
```

Before




```
        android:layout_height="wrap_content"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_margin="20dp"
        android:hint="Bill"
        android:gravity="center"
        android:inputType="numberDecimal" />

    <Button
        android:id="@+id/perc_12_butt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="12%"

        app:layout_constraintTop_toBottomOf="@id/bill_et"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toLeftOf="@id/perc_15_butt" />
```

After chain style is set to “packed”




```
        app:layout_constraintTop_toTopOf="parent"
        android:layout_margin="20dp"
        android:hint="Bill"
        android:gravity="center"
        android:inputType="numberDecimal" />

    <Button
        android:id="@+id/perc_12_butt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="12%"
        app:layout_constraintHorizontal_chainStyle="packed"
        app:layout_constraintTop_toBottomOf="@id/bill_et"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toLeftOf="@id/perc_15_butt" />

    <Button
        android:id="@+id/perc_15_butt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="15%"
        app:layout_constraintTop_toBottomOf="@id/bill_et"
        app:layout_constraintLeft_toRightOf="@id/perc_12_butt"
        app:layout_constraintRight_toLeftOf="@id/perc_18_butt" />

    <Button
        android:id="@+id/perc_18_butt"
        android:layout_width="wrap_content"
```

Displaying a Collection of Items

Declare some instance variables, these objects will be referenced from different functions in MainActivity.java

```
private Integer[] numbers;
private ArrayAdapter<Integer> aa;
```

Use a listview to display many items
This is in my main_activity.xml

```
<ListView
    android:id="@+id/randomNumberLV"
    android:layout_width="400dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    app:layout_constraintTop_toBottomOf="@id/vibrationBT"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"/>
```

In MainActivity.java:onCreate (at the end) define the list of numbers and define the ArrayAdapter and connect the listview to the arrayadapter.
Note: simple_list_item_1 is a very simple viewgroup provided by android (basically just a textview)

```
this.numbers = new Integer[]{5, 8, 17, 3, 2, 9, 20, 4, 7};
ListView lv = (ListView) findViewById(R.id.randomNumberLV);
this.aa = new ArrayAdapter<Integer>(getApplicationContext(), android.R.layout.simple_list_item_1, numbers);
lv.setAdapter(aa);
```

In the same button used to send the “Toast” sort the number array
And call notifyDataSetChanged()

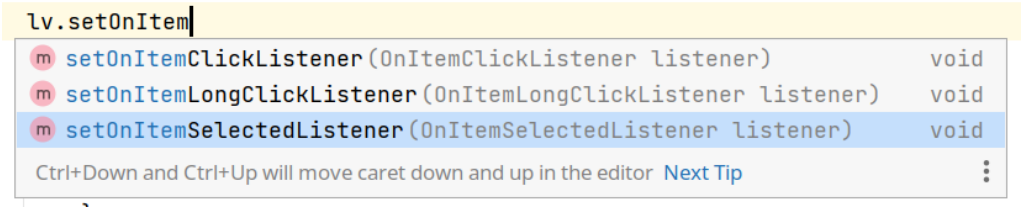
This function alerts the listview (via the arrayadapter) that the internal data has changed. The listview “resets” itself to display the new data

#import java.util.Arrays to get Arrays.sort()

The listview does not scroll and the content may not fit on the page!

```
Button b = findViewById(R.id.toastBT);
b.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        makeTheToastAgain(v);
        Arrays.sort(numbers);
        aa.notifyDataSetChanged();
    }
});
```

Make Items Clickable



Which one to use?

“itemSelected” is only for “Spinner” / a.k.a. drop down menus

Watch out for “onClickListener”

https://developer.android.com/reference/android/widget/ListView
ListView is a type of “AdapterView” which means we can lookup the various methods for the OnItemLongClickListener in that parent class
https://developer.android.com/reference/android/widget/AdapterView.OnItemLongClickListener

ListView Added in API level 1

Kotlin | **Java**

```
public class ListView
        extends AbsListView
```

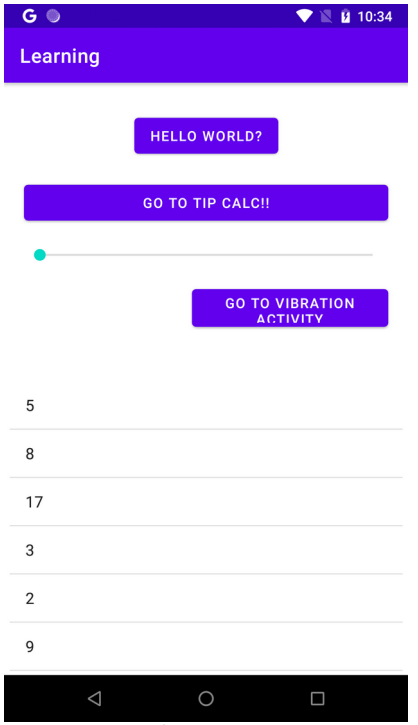
java.lang.Object
↳ android.view.View
↳ android.view.ViewGroup
↳ android.widget.AdapterView<android.widget.ListAdapter>
↳ android.widget.AbsListView
↳ android.widget.ListView

In our example, identical to the position number. But for other views and in other situations may differ.

The position of the view in the parent. In our example, “8” is in position 1

The view clicked. A “simple_list_item_1” https://github.com/aosp-mirror/platform_frameworks_base/blob/master/core/res/res/layout/simple_list_item_1.xml

The ListView in which the long clicked item is located




```
lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {



        return false;
    }
});
```

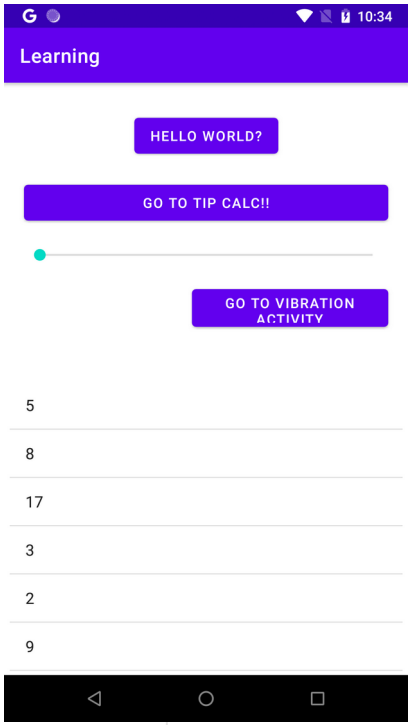
Return true to “consume” the click or return “false” to allow the click to also “go down/up” to the listview / parent

In our example, identical to the position number. But for other views and in other situations may differ.

The position of the view in the parent. In our example, “8” is in position 1

The view clicked. A “simple_list_item_1”
https://github.com/aosp-mirror/platform_frameworks_base/blob/master/core/res/res/layout/simple_list_item_1.xml

The ListView in which the long clicked item is located




```
lv.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
        int num = Integer.parseInt(((TextView)view).getText().toString());
        SeekBar sb = findViewById(R.id.randomSB);
        sb.setProgress(num);

        return false;
    }
});
```

Return true to “consume” the click or return “false” to allow the click to also “go down/up” to the listview / parent