
TRY
ANDROID

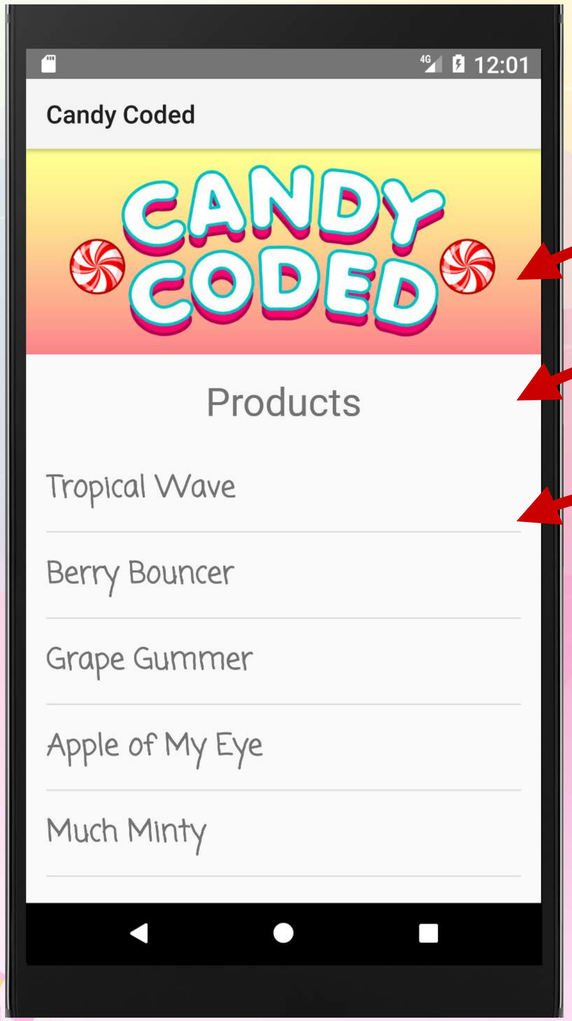
Level 3

ListView & Adapters

Using Adapters to Put Data Into a ListView

Adding a ListView

Now that we've added a TextView **and** ImageView**, we want to add a** ListView **to list our store's available candy.**



An `ImageView` with our Candy Coded logo

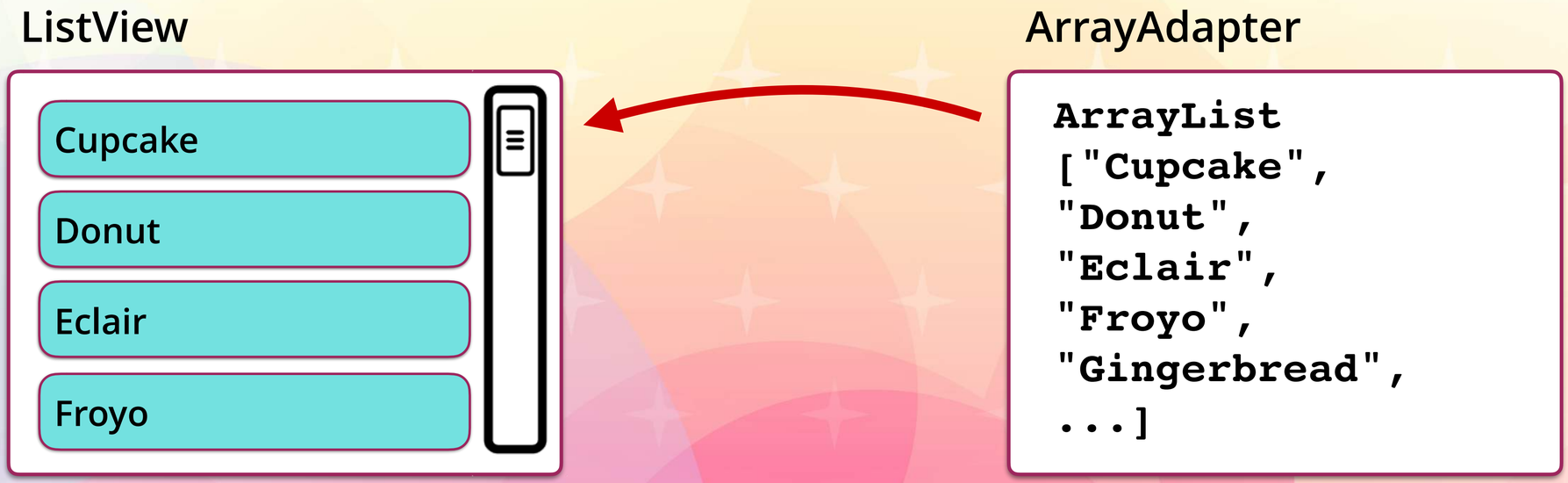
A `TextView` with text "Products"

A `ListView` listing all of our candy products

TRY ANDROID

ListView & ArrayAdapter

A ListView **get its data from a data source via an** Adapter**.**



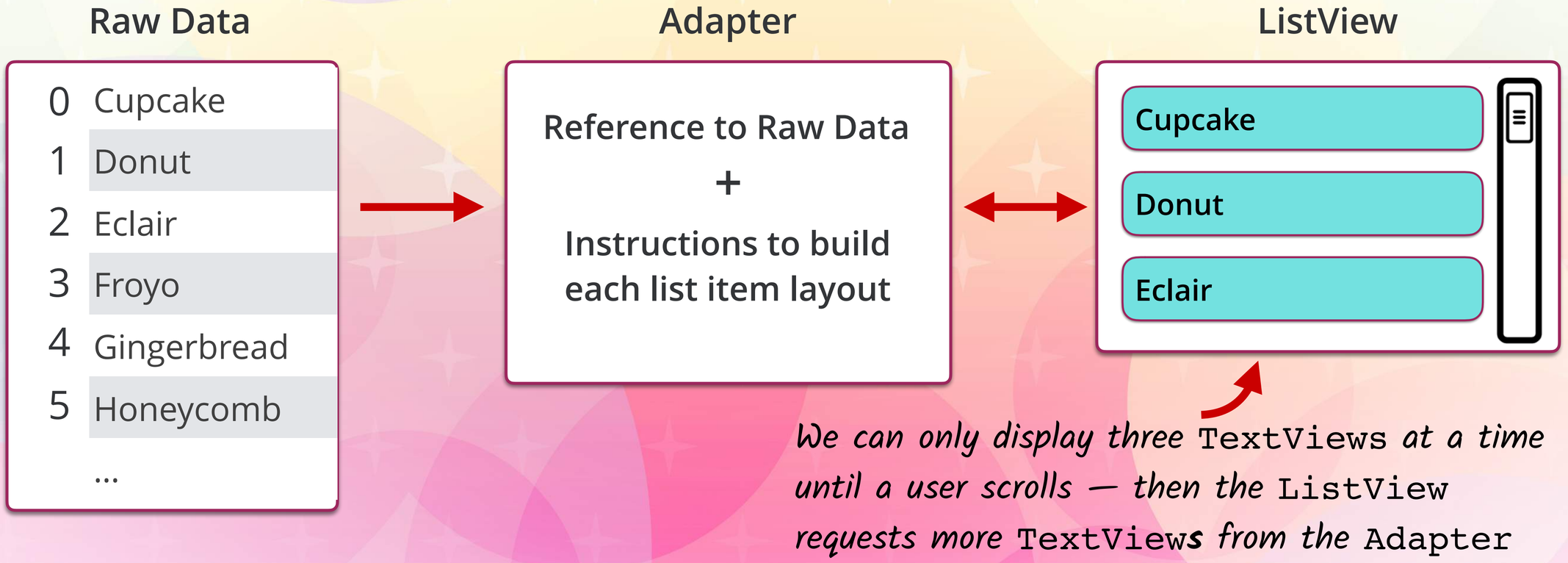
An `ArrayAdapter` *can populate a* `ListView` *with an* `ArrayList`

But why do we need an `Adapter`*? Can't we just send a* `ListView` *a list?*

TRY ANDROID

How Adapters Work

The ArrayAdapter **allows us to only create the items requested by the** ListView**.**



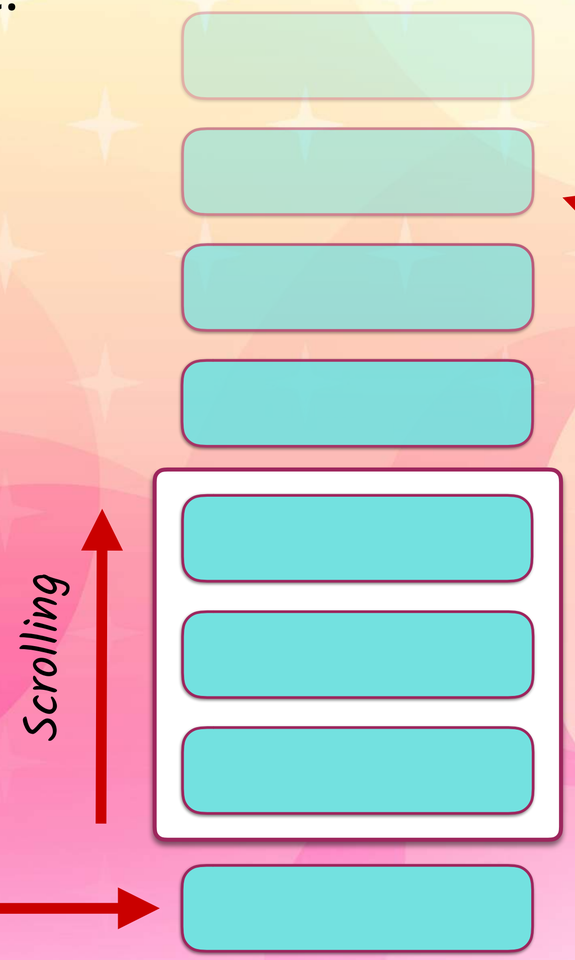
TRY ANDROID

A ListView Without Recycling Views

If we don't recycle the TextViews that make up our ListView cell as they scroll offscreen, we would quickly affect memory and performance.

A `ListView` with its visible `TextViews` and one in either direction:

If we create a new `TextView` as the user scrolls...

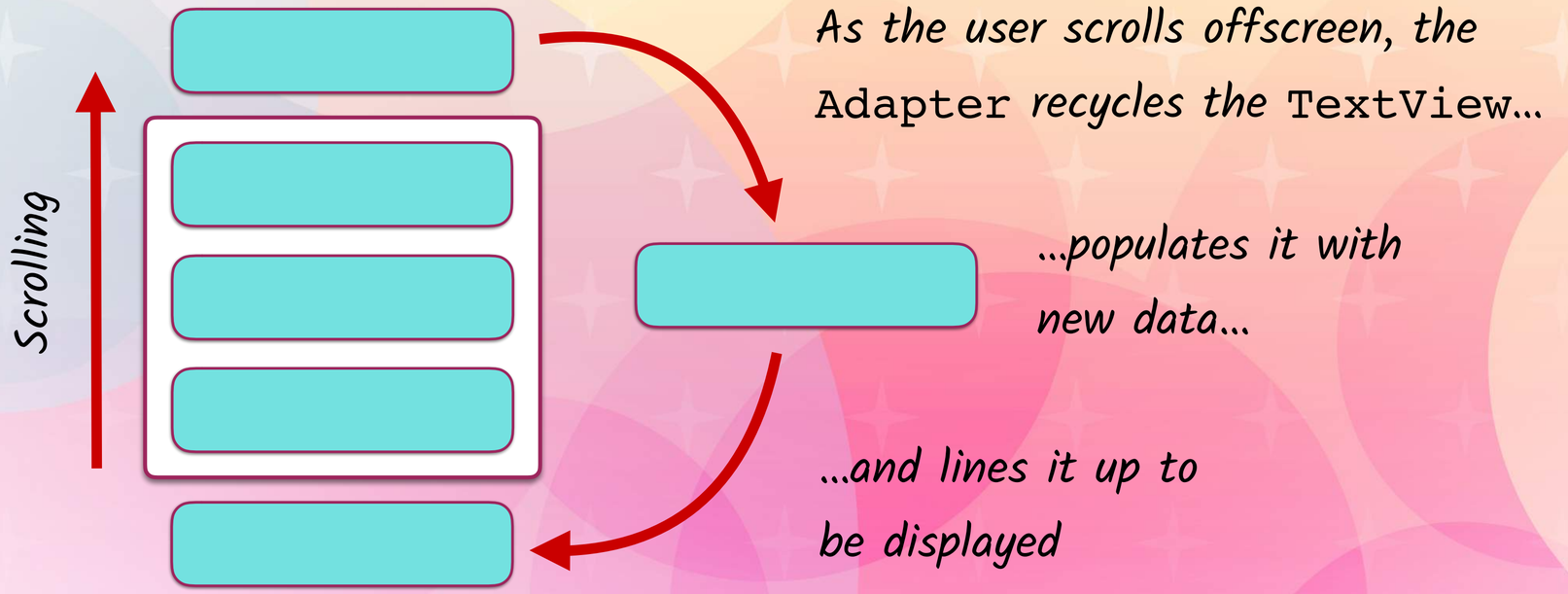


We would quickly affect memory and performance by creating all these cells for a list with thousands of records

TRY ANDROID

How ListView Recycling Works

ListView cells are recycled as they scroll offscreen, and any new cells are just recycled cells filled with new data.



As the user scrolls offscreen, the `Adapter` *recycles the* `TextView`*...*

...populates it with new data...

...and lines it up to be displayed

TRY ANDROID

The Steps to Create the ListView and ArrayAdapter

In order to get our ListView **populated with data from the** ArrayAdapter**, we need to do the following steps:**

Steps to set up a `ListView` *and* `ArrayAdapter`*:*

Layout steps

1) Create a **`ListView`** in the **`main_activity.xml`** layout
2) Create a separate layout to describe what each list item will look like

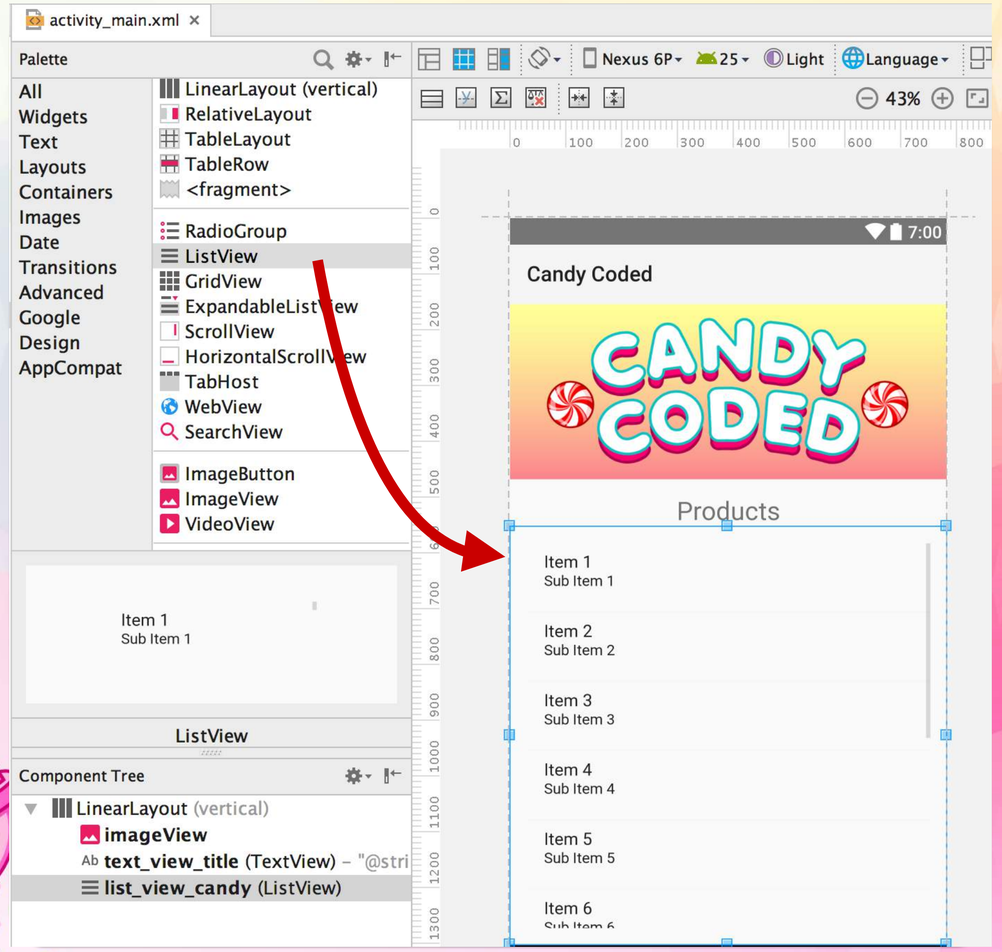
Java steps

3) Get the raw data we want to display into an **`ArrayList`**
4) Create the **`ArrayAdapter`**
5) Connect the **`ArrayAdapter`** to the **`ListView`**

TRY ANDROID

1 Creating a ListView in the MainActivity Layout

To add a ListView **to our** activity_main **layout, we can drag it over in the design view, which will generate the following** ListView **XML.**



activity_main.xml

```
<?xml ...>
<LinearLayout ...>
    <ImageView .../>
    <TextView .../>

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/list_view_candy"/>

</LinearLayout>
```

Our ListView's Properties

The ListView**'s** layout_width**,** layout_height**, and** id **were set to default values for us.**

activity_main.xml

```
<?xml ...>
<LinearLayout ...>
    <ImageView .../>
    <TextView .../>

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/list_view_candy"/>

</LinearLayout>
```

`match_parent` *means the view will try to be as big as its parent by expanding to take up the remaining space in the layout*

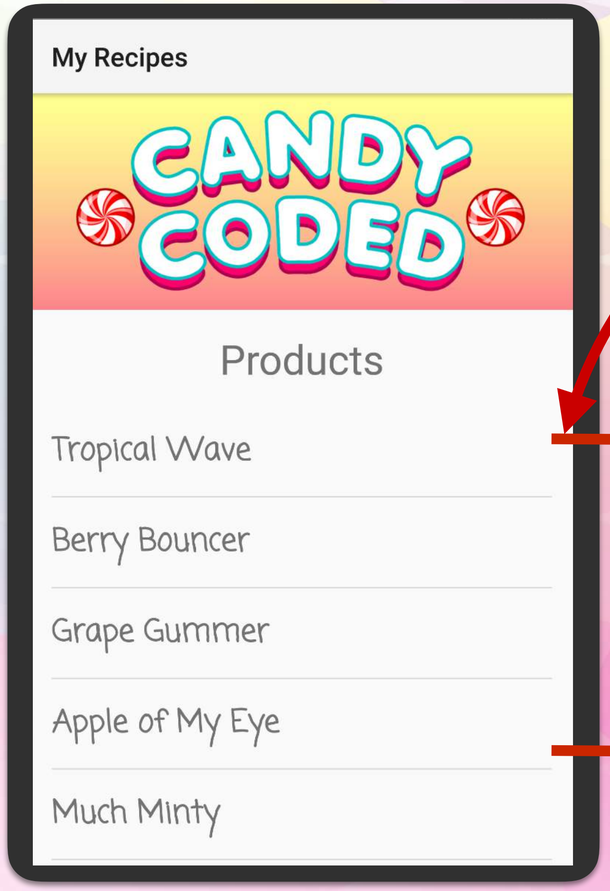
`@+id` *means this id will be created as a resource at runtime*

`list_view_candy` *is this* `ListView`*'s id we picked, but we could name it anything*

TRY ANDROID

What Does match_parent Mean?

The ListView**'s** layout_width **and** layout_height **are** match_parent**.**



`match_parent` *as the* `layout_height` *means the list will take up the available height left in its parent layout*

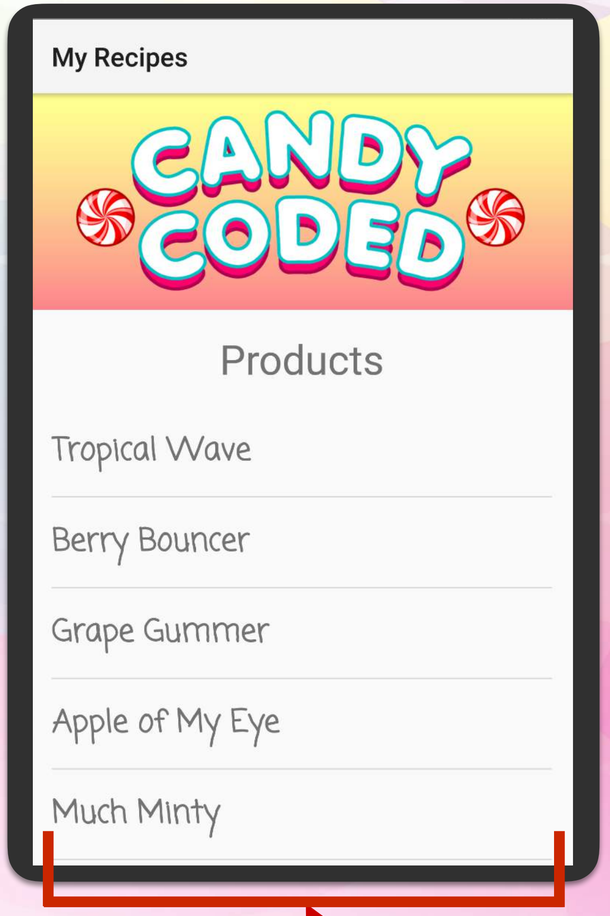
So it will be the height from the bottom of the `ImageView` *to the bottom of the screen*

If our list content is longer than the screen, it will automatically scroll

TRY ANDROID

What Does match_parent Mean?

The ListView**'s** layout_width **and** layout_height **are** match_parent**.**

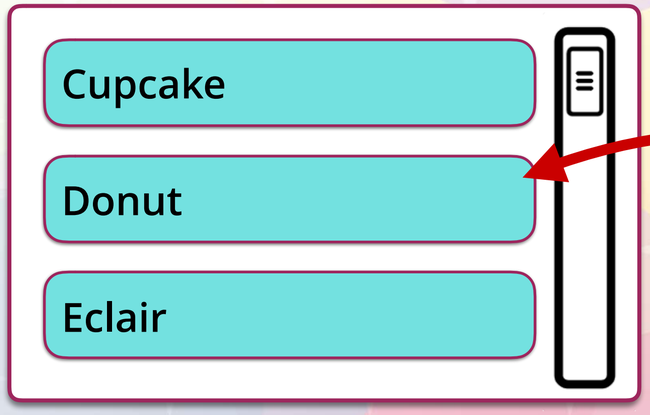


`match_parent` *as the* `layout_width` *means the view list will be the entire width of the screen*

TRY ANDROID

2 Creating a Layout for Each Item in Our ListView

We need a layout to tell the ArrayAdapter **how to build each item for the** ListView**.**



We'll use a single `TextView` *for each item in our* `ListView`

list_item_candy.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/text_view_candy">
</TextView>
```

`match_parent` *since we want each list item to fill the* `ListView`

We need an id to reference this later

Screencast: The Layout for the Items in Our ListView

TRY ANDROID

Where to Add Code in MainActivity.java

We'll add our code to the bottom of the onCreate() **method, after we set the** TextView**'s text.**

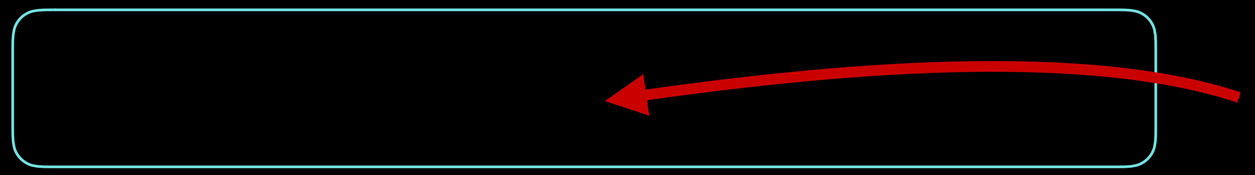
MainActivity.java **Java**

```
...
public class MainActivity extends AppCompatActivity {

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

        TextView textView = (TextView)this.findViewById(
                R.id.text_view_title);
        textView.setText(R.string.welcome_msg);

    }
}
```



We'll add all the code to create our `ListView` *here*

3 Creating the ArrayList of Raw Data

We need to create the list of raw data to display. We'll use an ArrayList for that.

MainActivity.java Java

```
import java.util.ArrayList;

...
        ArrayList<String> candyList = new ArrayList<String>();

        candyList.add("Tropical Wave");
        candyList.add("Berry Bouncer");
        candyList.add("Grape Gummer");
        candyList.add("Apple of My Eye");
        ...
        candyList.add("ROYGBIV Spinner");
...
```

Creating an empty `ArrayList` *of* `Strings`

Adding `Strings` *to the list*

4 Creating the ArrayAdapter From an ArrayList

We need a few things in the ArrayAdapter **constructor: the** Activity**, the** Layout **for each item, the** TextView **for each item, and the** ArrayList **we just created.**

MainActivity.java **Java**

```
import java.util.ArrayList;

...
        ArrayList<String> candyList = new ArrayList<String>();
        ...

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
                this,
                R.layout.list_item_candy,
                R.id.text_view_candy,
                candyList  );

...
```

We need these things in our ArrayAdapter *constructor to create our* ArrayAdapter

4 Creating the ArrayAdapter From an ArrayList

We need a few things in the ArrayAdapter **constructor: the** Activity**, the** Layout **for each item, the** TextView **for each item, and the** ArrayList **we just created.**

MainActivity.java **Java**

```
import java.util.ArrayList;

...
        ArrayList<String> candyList = new ArrayList<String>();
        ...

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
                this,
                R.layout.list_item_candy,
                R.id.text_view_candy,
                candyList  );

...
```

Where this `ArrayAdapter` *will be used, or the context, is* `this` `Activity`

4 Creating the ArrayAdapter From an ArrayList

We need a few things in the ArrayAdapter **constructor: the** Activity**, the** Layout **for each item, the** TextView **for each item, and the** ArrayList **we just created.**

MainActivity.java **Java**

```
import java.util.ArrayList;

...
        ArrayList<String> candyList = new ArrayList<String>();
        ...

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
                this,
                R.layout.list_item_candy,
                R.id.text_view_candy,
                candyList  );

...
```

The `Layout` *file we created to build each item in the* `ListView`

4 Creating the ArrayAdapter From an ArrayList

We need a few things in the ArrayAdapter **constructor: the** Activity**, the** Layout **for each item, the** TextView **for each item, and the** ArrayList **we just created.**

MainActivity.java **Java**

```
import java.util.ArrayList;

...
        ArrayList<String> candyList = new ArrayList<String>();
        ...

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
                this,
                R.layout.list_item_candy,
                R.id.text_view_candy,
                candyList  );

...
```

We need the specific `TextView` *inside the* `Layout` *file. Remember we created an id for this previously...*

4 Creating the ArrayAdapter From an ArrayList

We need a few things in the ArrayAdapter **constructor: the** Activity**, the** Layout **for each item, the** TextView **for each item, and the** ArrayList **we just created.**

MainActivity.java **Java**

```java
import java.util.ArrayList;

...
        ArrayList<String> candyList = new ArrayList<String>();
        ...

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
                this,
                R.layout.list_item_candy,
                R.id.text_view_candy,
                candyList );

...
```

Finally, the `ArrayList` *of our candies*

5 Connecting the ArrayAdapter to the ListView

To connect the ArrayAdapter **to the** ListView**, we first need to find the** ListView **so we have a reference to it. We can find** Views **with the** findViewById() **method.**

MainActivity.java **Java**

```
import java.util.ArrayList;

...
        ArrayList<String> candyList = new ArrayList<String>();
        ...

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
                ... );

        ListView listView = (ListView)this.findViewById(
                R.id._____________);

...
```

Remember we created an id for the ListView *in the* MainActivity*'s Layout*

The id for the ListView in main_activity.xml

We need the id **for the** ListView **so we can find it and then attach our** ArrayAdapter **to it.**

activity_main.xml XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout ...>
    <TextView …/> <ImageView …/>
    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/list_view_candy"/>
</LinearLayout>
```

MainActivity.java

```
...
    ListView listView = (ListView)this.findViewById(
            R.id.list_view_candy);
...
```

5 Connecting the ArrayAdapter to the ListView

Finally! We can connect the ListView **to the** ArrayAdapter **and see our** ListView **in action.**

MainActivity.java **Java**

```
...
        ArrayList<String> candyList = new ArrayList<String>();
        ...

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
                ... );

        ListView listView = (ListView)this.findViewById(
                R.id.list_view_candy);

        listView.setAdapter(adapter);
...
```

Now that we found the ListView *by its* id, *we can set its* Adapter

Screencast: Demo All the Code in MainActivity.java


TRY
ANDROID