

BUILD X : ANDROID

Lecture Three: Recap and ListViews



Recap



Recap – What are we doing?

We're going to build a basic todo list app, **KCL Tech Todo**.

If you want a preview, it is **available on Google Play**.

- Search for "**KCL Tech Todo**" on Google or on the Play Store.



Recap – Layouts, Views and View Groups

- **Layout:** a specific type of application resource
 - These define the structure and appearance of parts of your app
- **View:** an individual component of a layout
 - A button, input field, text field, image, etc.
- **View Group:** a special type of View that can contain other views
 - You can't see the view group, but you can see the views inside of it
 - LinearLayout, ScrollView, RelativeLayout, etc.
- **Other Resources:** used to supplement a layout, amongst other things
 - Strings, styles, dimensions, etc.

Recap – Events, Intents and the Activity Lifecycle

- **Events:** your app can “listen” to all types of events on all types of views
 - Once you have a reference to your view in Java, it’s easy to set an action listener with a few lines of code.
- **Intents:** these tell Android that your app *intends* to do something
 - Usually these start a new activity, but they can do all sorts of other things.
- **Activity Lifecycle:** your activity will move through many different states
 - A **callback** method is called when each of your activities moves to a new state, such as `onPause()`.

How would you scroll through them?

What happens if the data changes?

How would you display a list of items?

What happens when items are not on the screen?

Where would you get the data from?

ListView



ListView

A ListView displays a list of items.

They are the **most common type of complex view**.

ListView – What's the point?

When it comes to memory, **views are expensive**, and making them is even more expensive!

A View usually consumes a couple of KB of RAM, and with complicated layouts this can really add up.

Imagine having 100 contacts in the phone book, each having a name, number, picture and probably more views/view groups - that's already a few MB of memory before any computation.

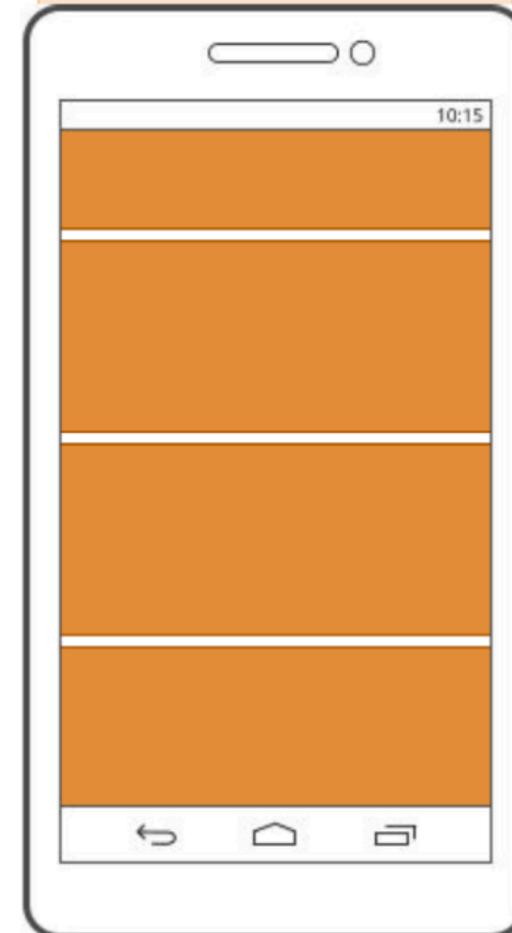
List Views – What's the point?

A ListView will **recycle** views when they aren't needed, which is **much more efficient**.

ListView - Recycling

The **Wrong** Way:

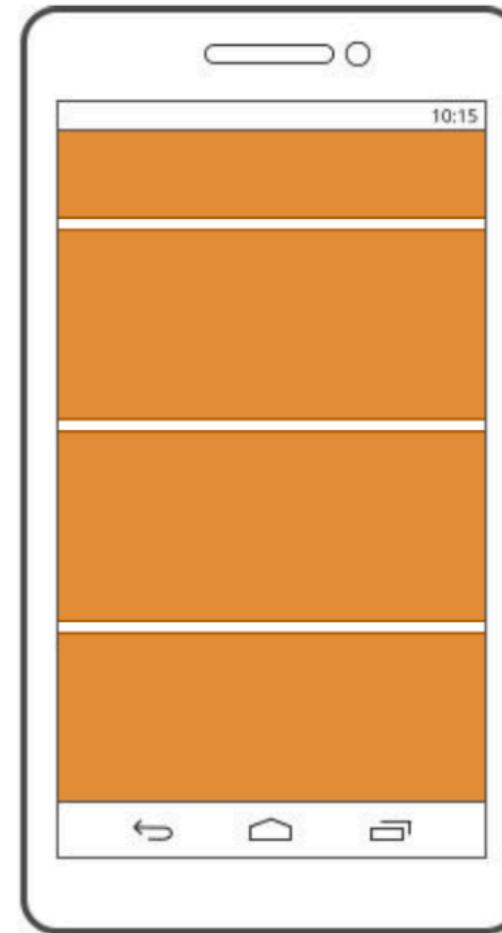
Don't recycle them at all. Create a new view for each item, taking up **a ton of memory** and slowing things down.



ListView - Recycling

The **Right** Way:

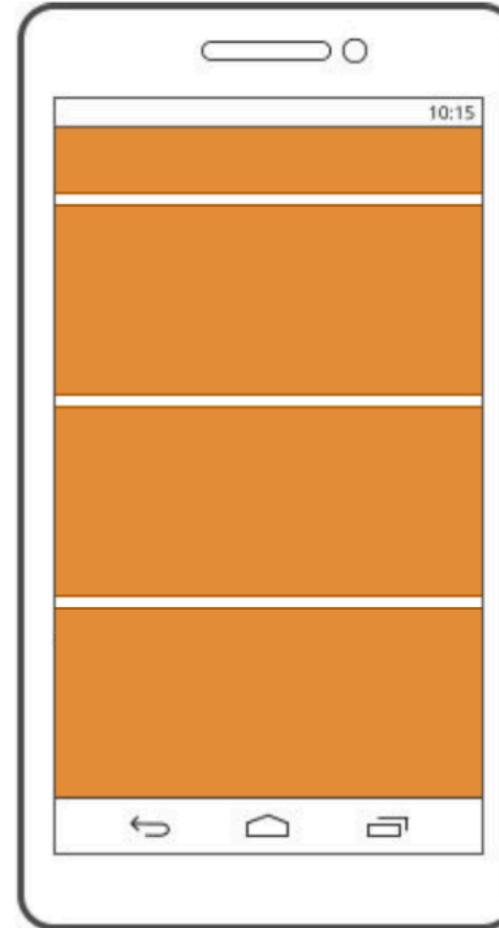
When the user scrolls down, a view
"disappears" off the top.



ListView - Recycling

The **Right** Way:

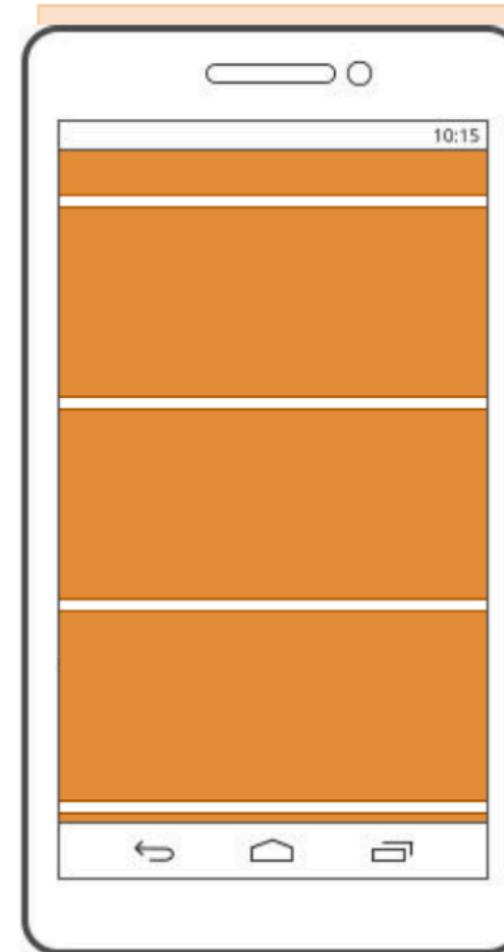
When the user scrolls down, a view
"disappears" off the top.



ListView - Recycling

The **Right** Way:

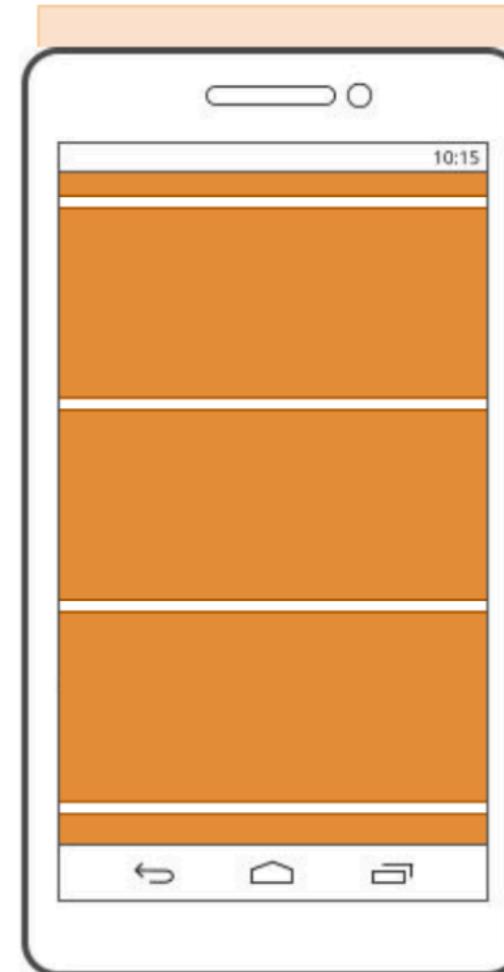
When the user scrolls down, a view
"disappears" off the top.



ListView - Recycling

The **Right** Way:

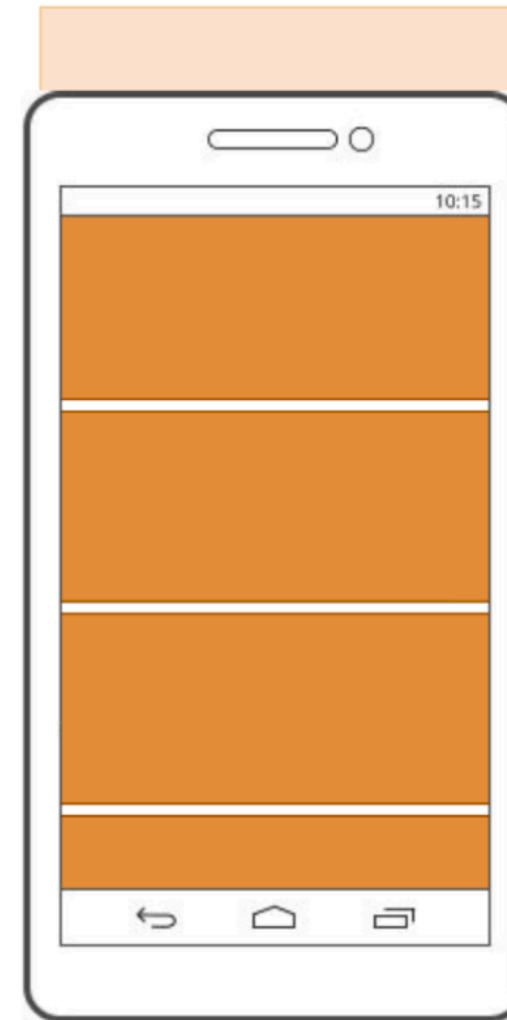
When the user scrolls down, a view
"disappears" off the top.



ListView - Recycling

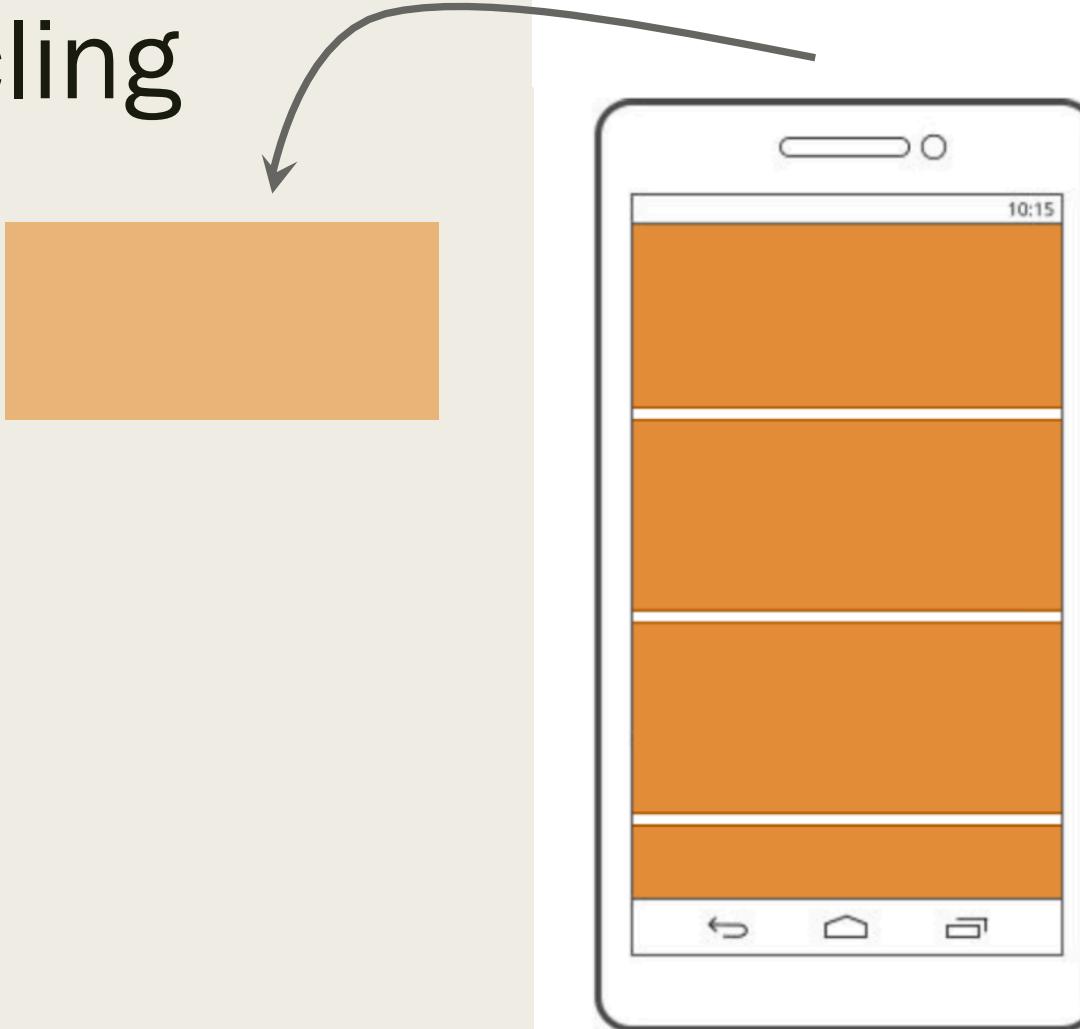
The **Right** Way:

When the user scrolls down, a view
"disappears" off the top.



ListView - Recycling

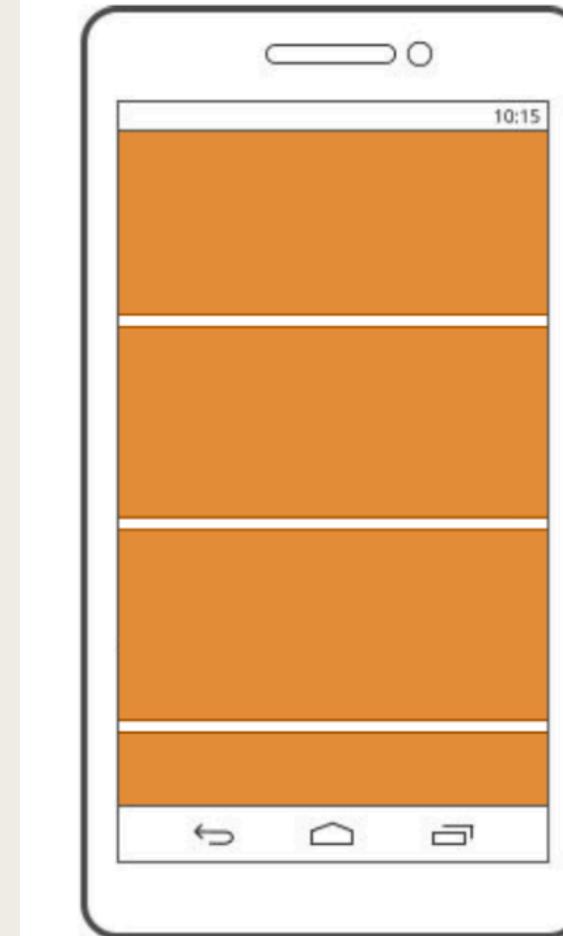
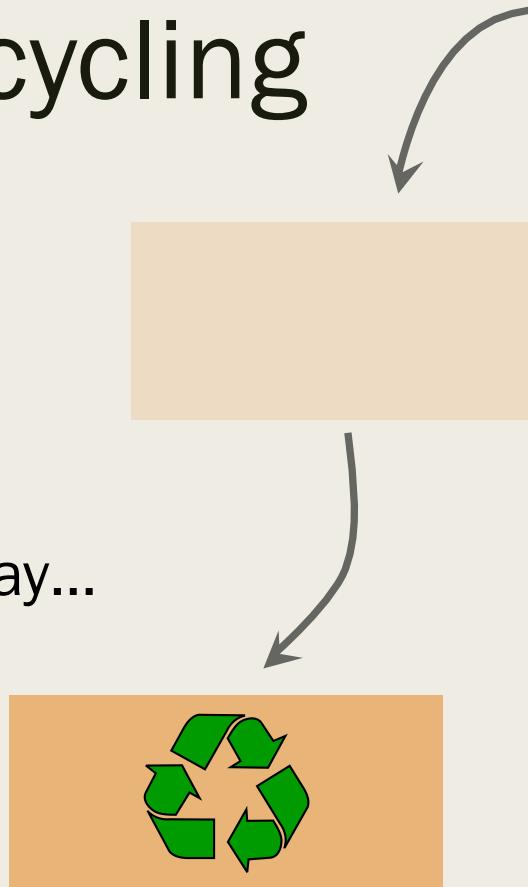
The **Right** Way:
The view is then taken away...



ListView - Recycling

The **Right** Way:

The view is then taken away...
...recycled...



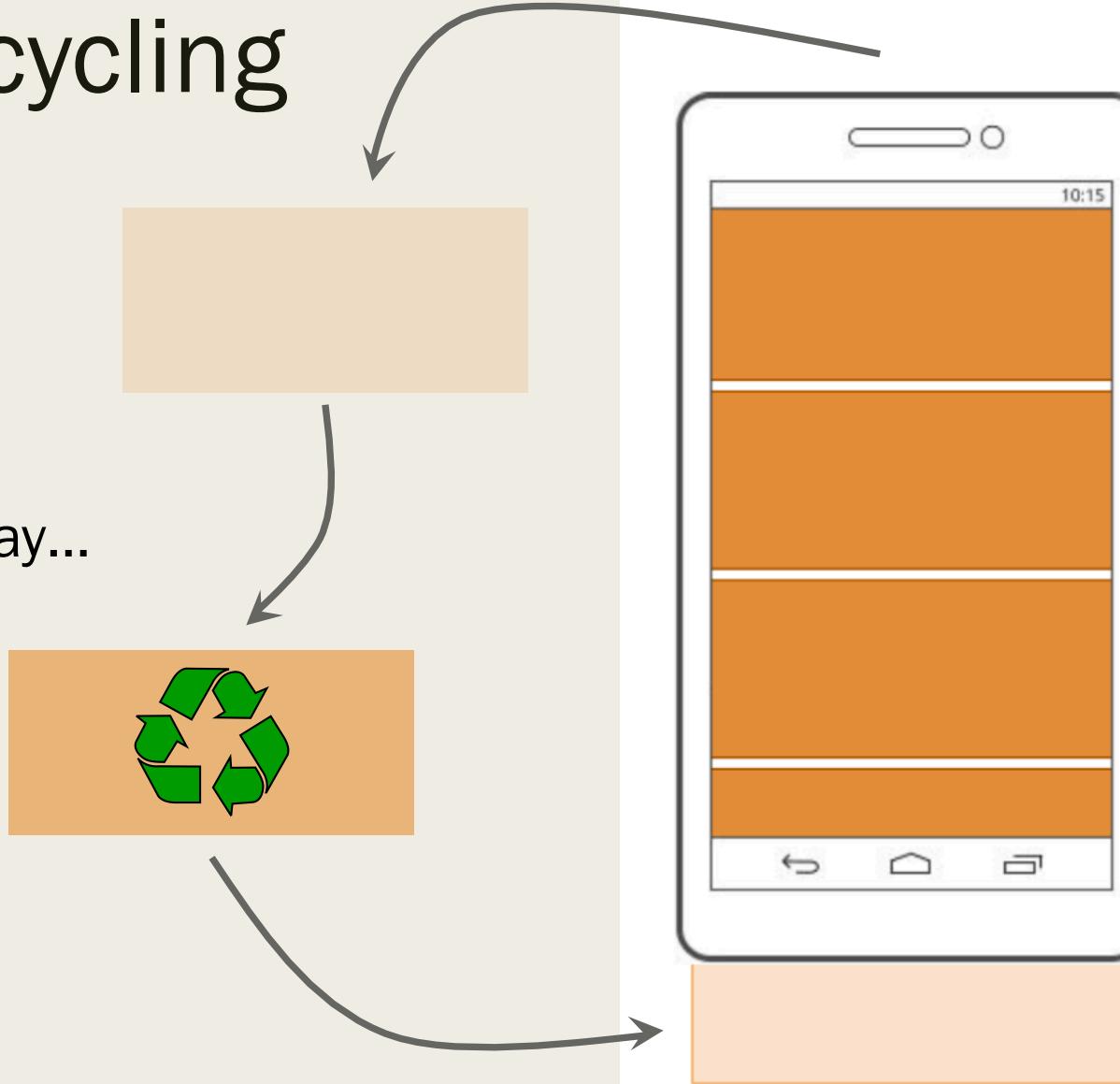
ListView - Recycling

The **Right** Way:

The view is then taken away...

...recycled...

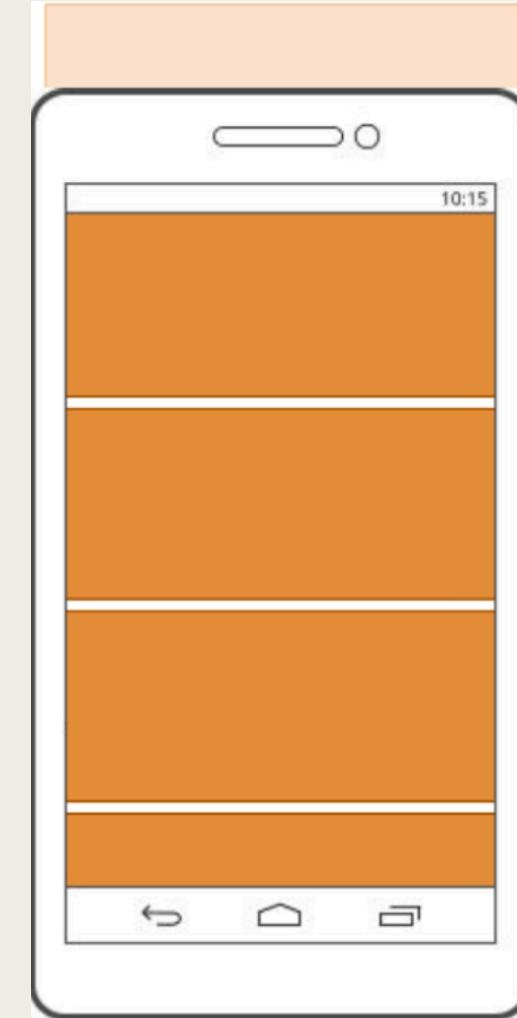
...and replaced back at
the bottom, ready to
appear.



ListView – Joining the Dots



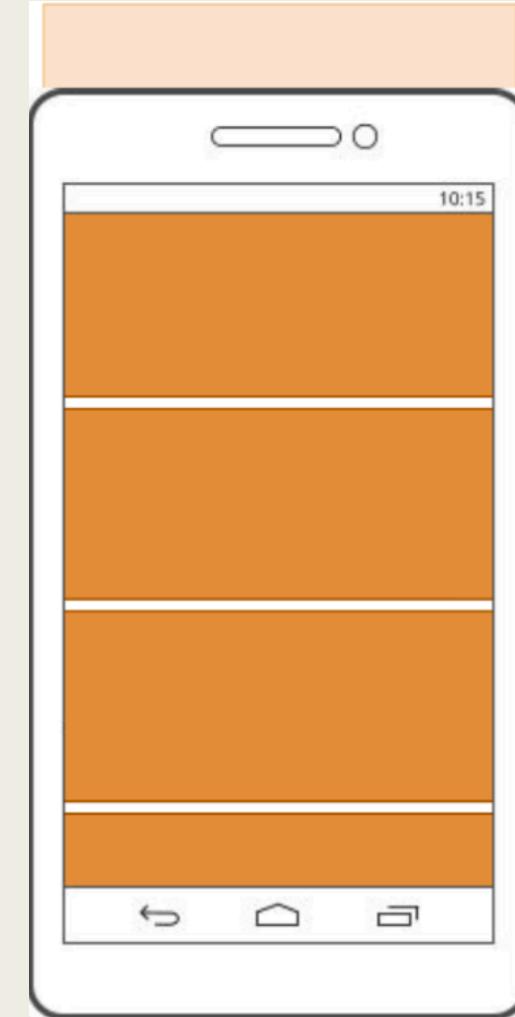
Magic?



ListView – Joining the Dots



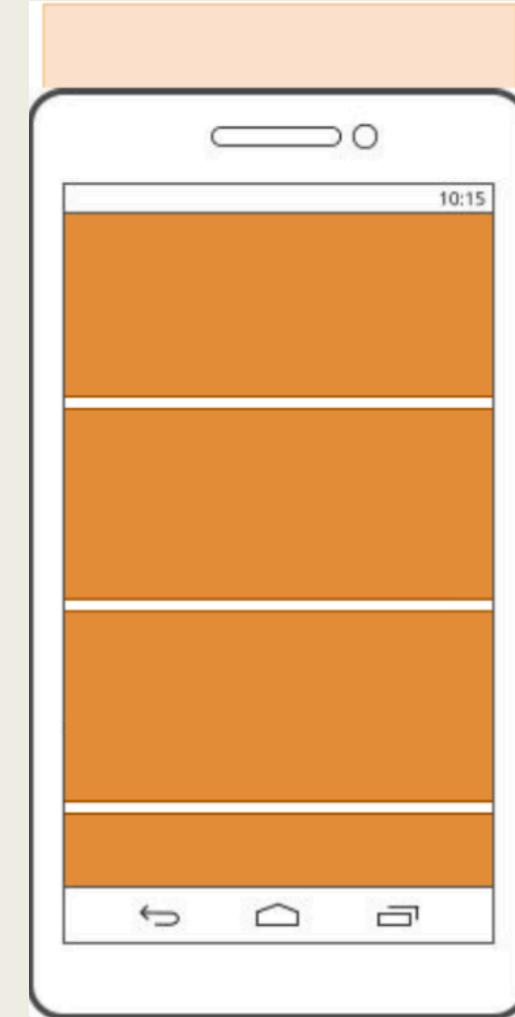
→ **Nope.** →



ListView – Joining the Dots



An
Adapter



ListView Adapters

An **adapter** maps your **ListView** to a **data source** – a list of contacts, the tasks in our todo app, etc.

An **adapter** is just a Java class that as certain methods, used by a **ListView**.

Adapters help maintain the **separation of concerns** in your software.

Your data **does not know, and does not care**, that it is being displayed in a **ListView**.

The **ListView does not know, and does not care**, what your data represents.

ListView Adapters

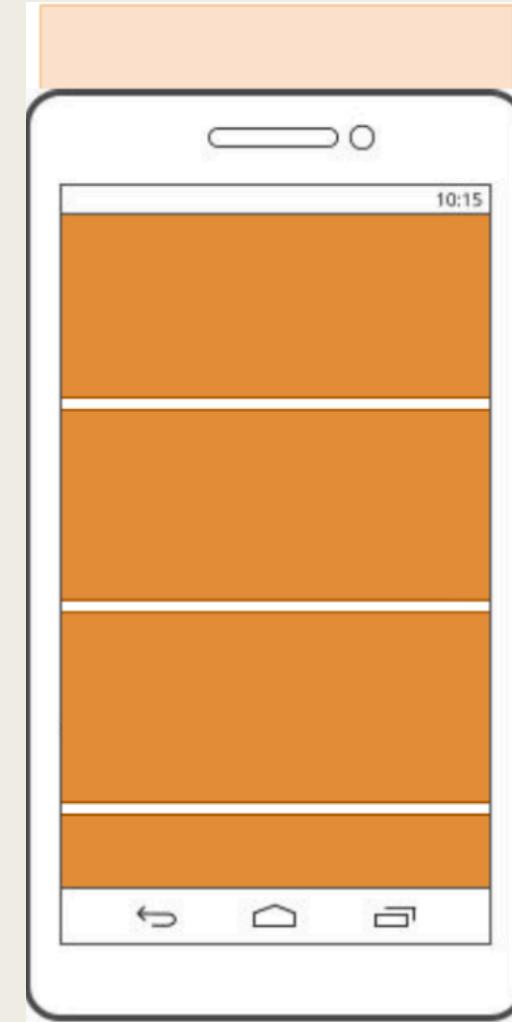
The process is simple: you give your data to the adapter, and the ListView uses the adapter to retrieve your data (and information about it).

This allows you to build a **custom adapter** to accept any kind of data you have, and expose a **standard pre-defined interface** that a ListView can understand.

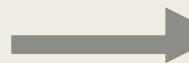
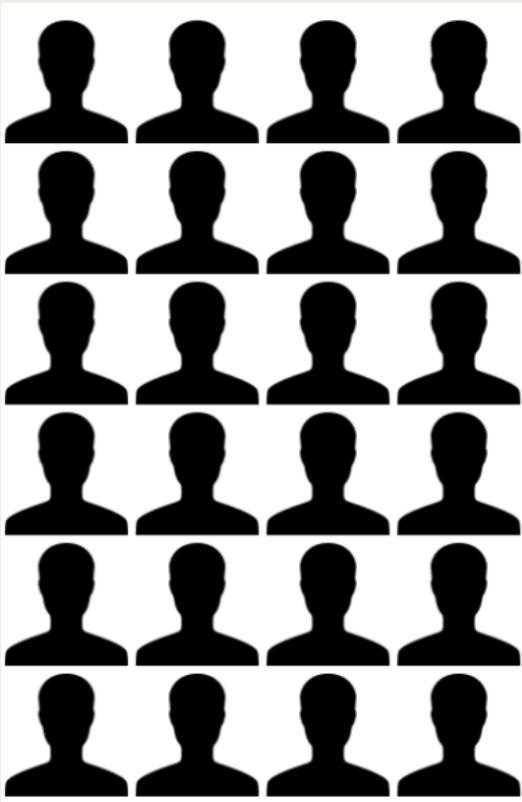
ListView Adapters



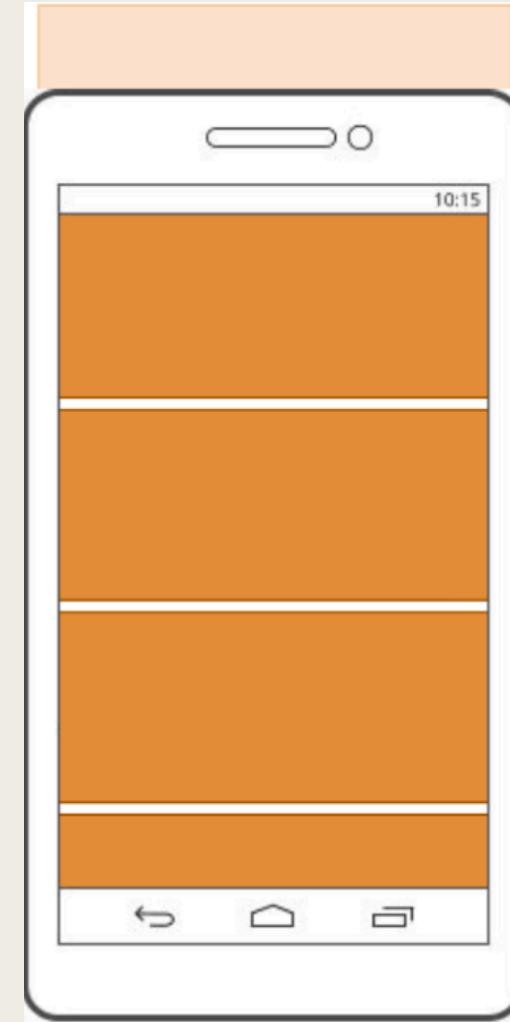
An
Adapter



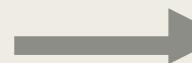
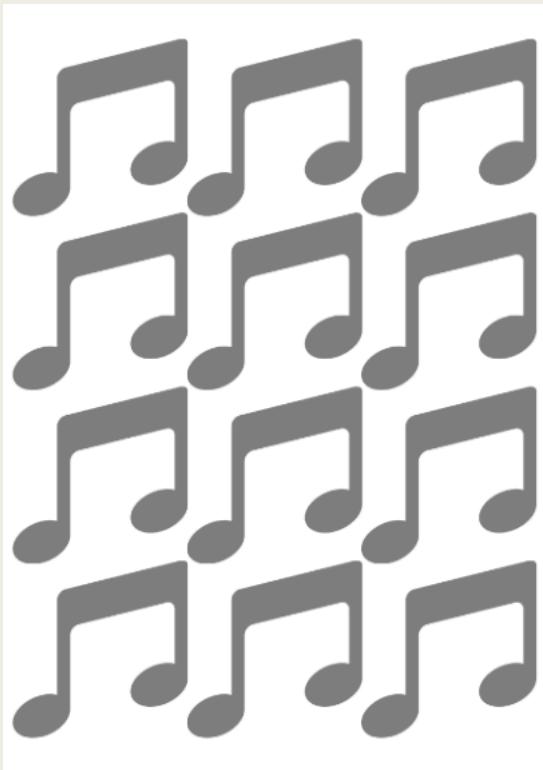
ListView Adapters



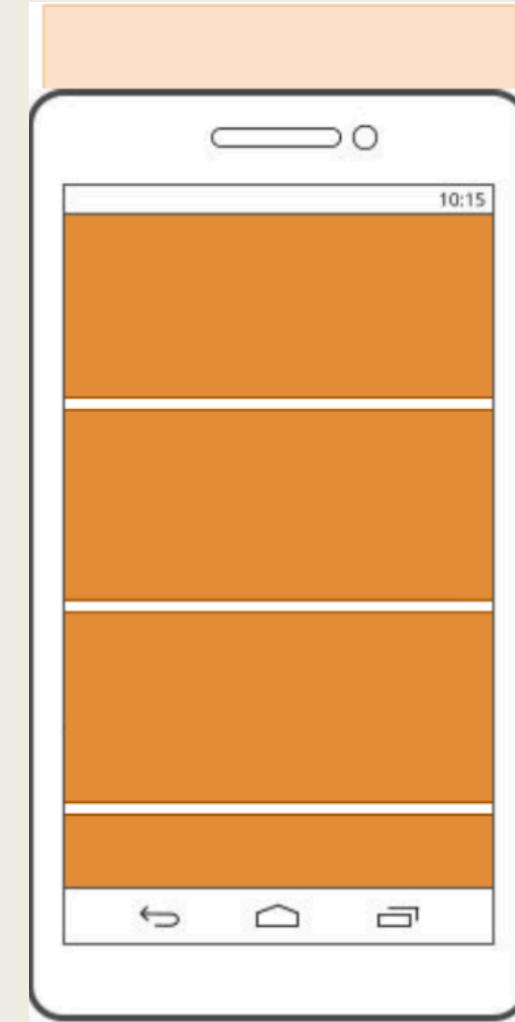
An
Adapter



ListView Adapters



An
Adapter





Dude, just chill out and relax.

Break! (10 mins)

ListView

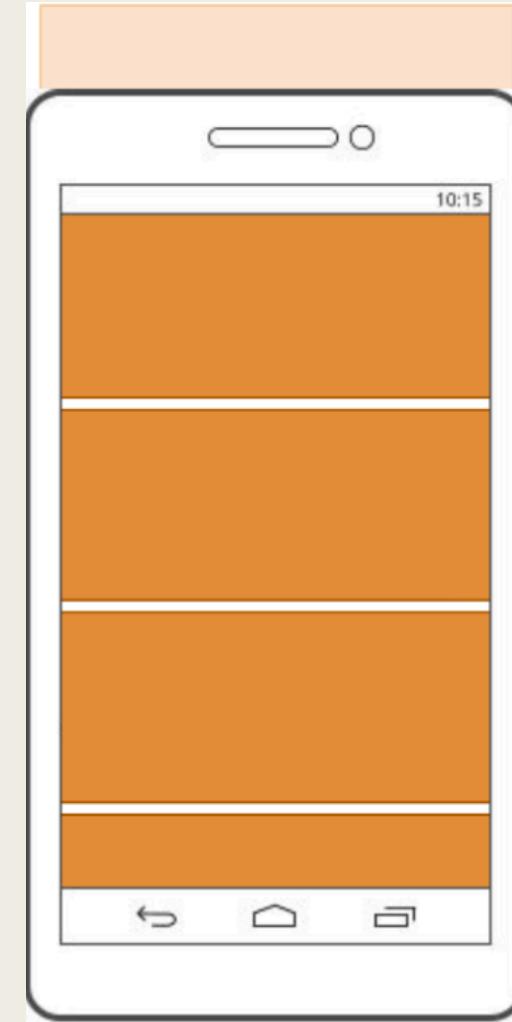
A ListView displays a list of items.

They are the **most common type of complex view**.

ListView Adapters



An
Adapter





Coding time!

Next Week: Databases

Next week we're going to be studying the use of **databases** in Android apps.

We're not doing anything exotic with them, but if you haven't used databases in a while, **do a little reading** to brush up on them.

If you know roughly what these words and terms mean, you're all good.

database table column
record row query SQL



Break! (10 mins)