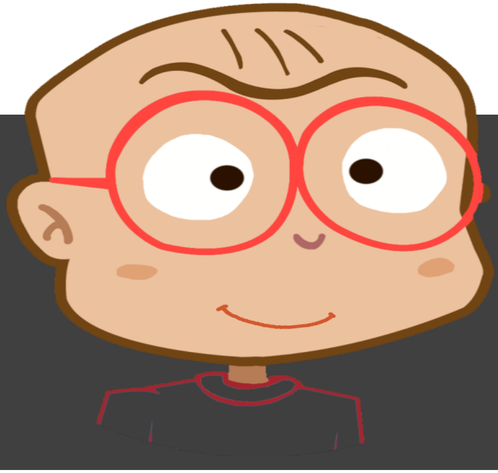
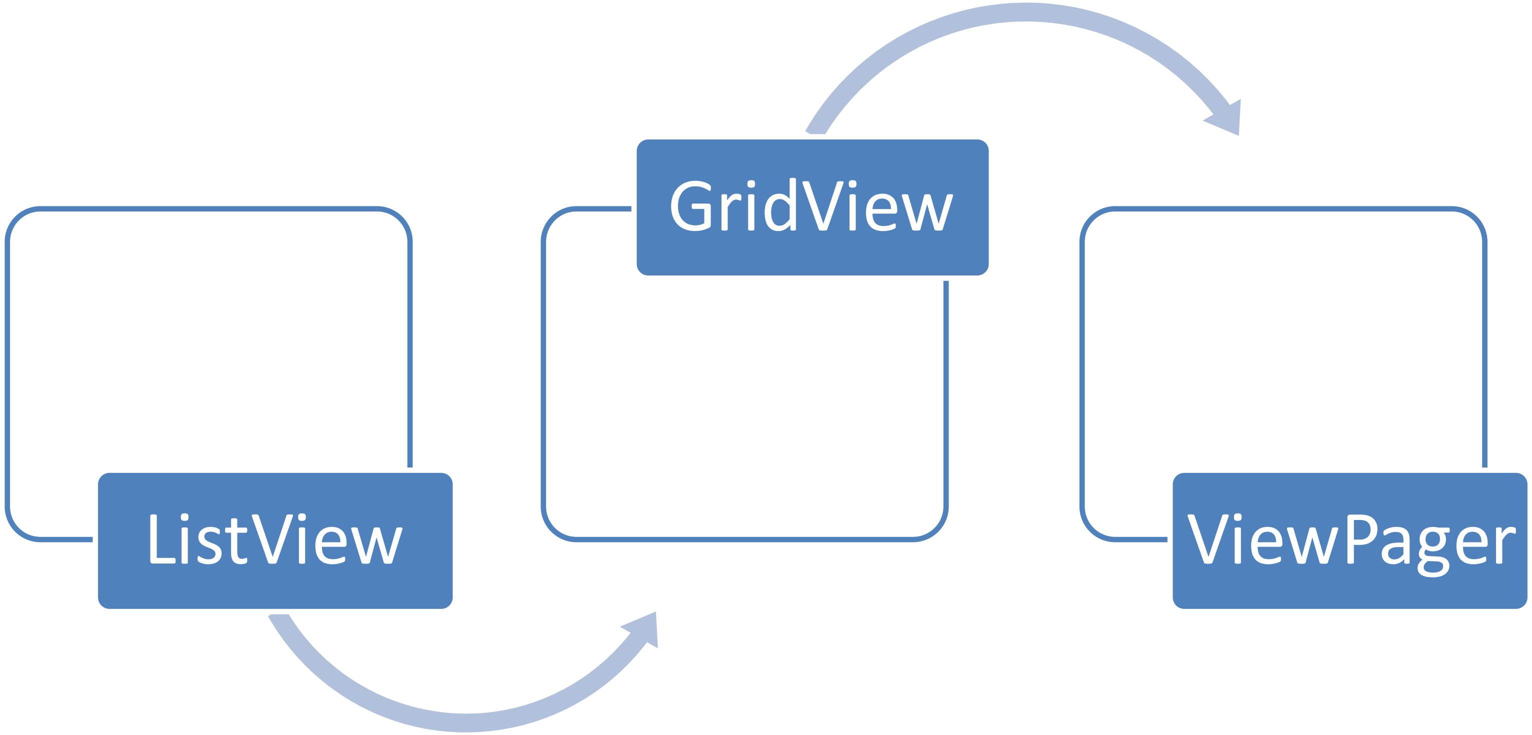
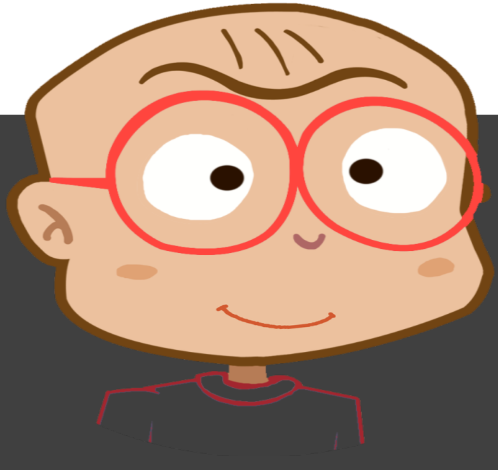


# ANDROID USER INTERFACE

START »

# *Table of contents*

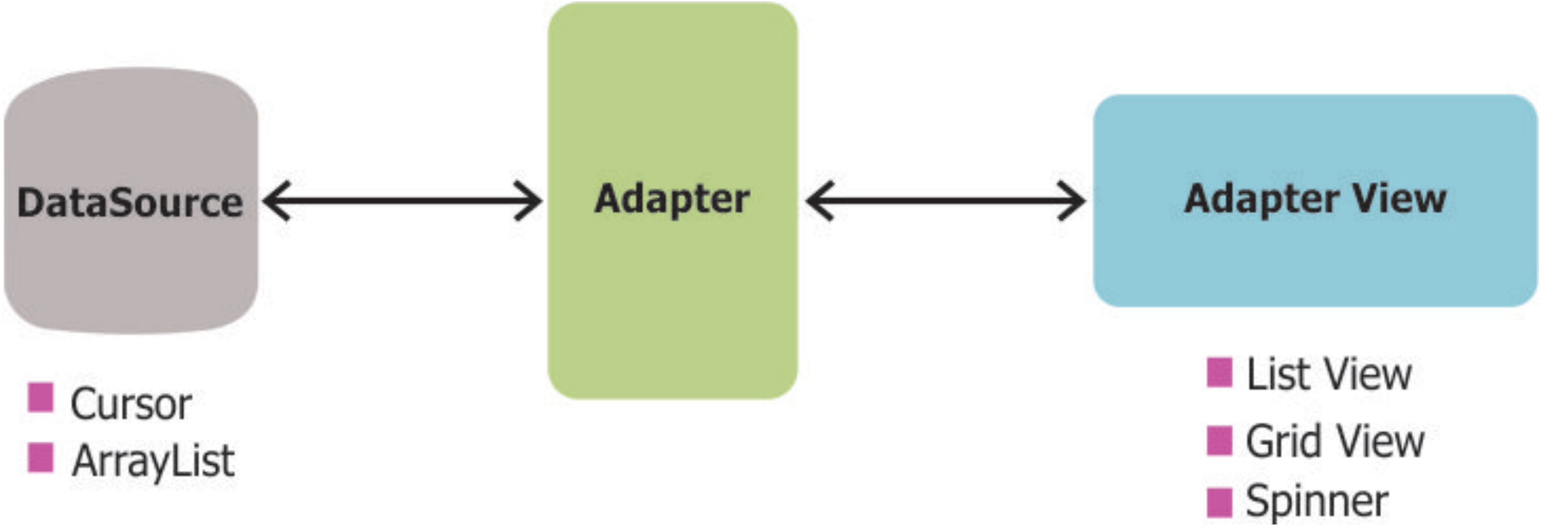




# Adapter

# Adapter

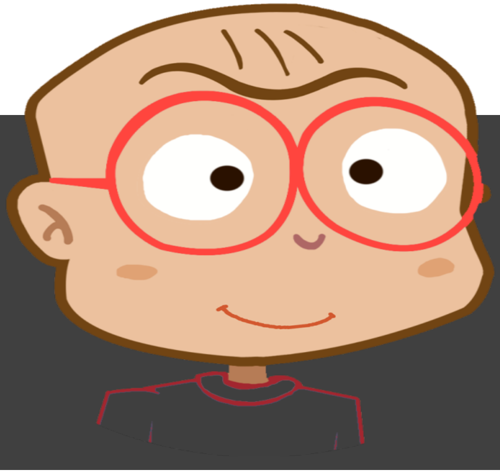
An adapter actually bridges between UI components and the data source that fill data into UI Component. Adapter can be used to supply the data to like spinner, list view, grid view etc.



# Adapter

Android provides several subclasses of Adapter that are useful for retrieving different kinds of data and building views for an AdapterView ( ie. ListView or GridView)

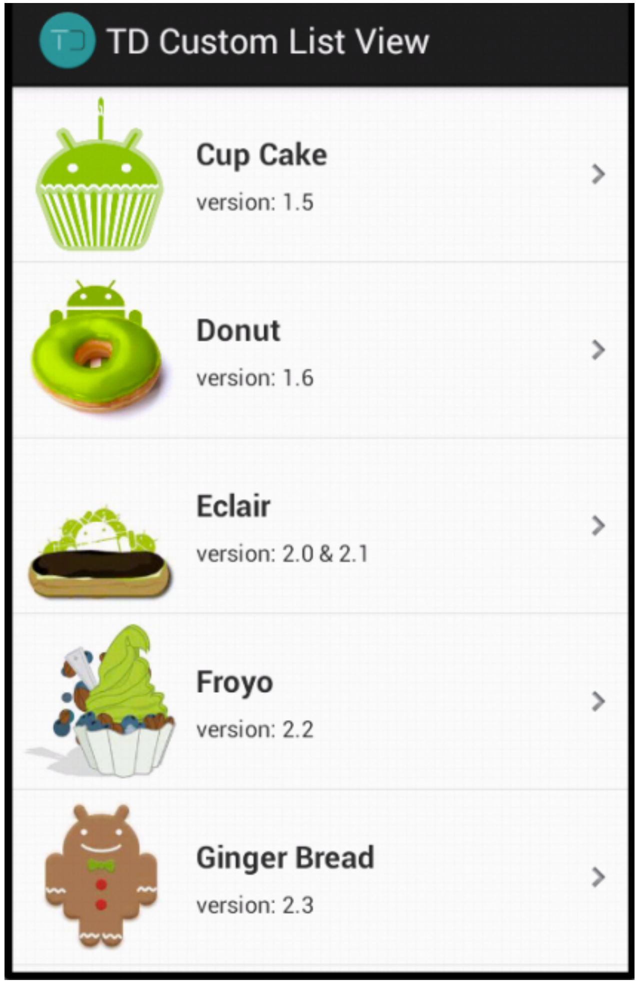
- ArrayAdapter
- SimpleCursorAdapter
- CursorAdapter
- BaseAdapter
- PagerAdapter
- FragmentPagerAdapter
- FragmentStatePagerAdapter
- etc..



# List View

# ListView

- Android **ListView** is a view which groups several items and display them in vertical scrollable list. The list items are automatically inserted to the list using an **Adapter** that pulls content from a source such as an array or database.
- The **ListView** and **GridView** are subclasses of **AdapterView** and they can be populated by binding them to an **Adapter**, which retrieves data from an external source and creates a View that represents each data entry.



TD Custom List View

Cup Cake
version: 1.5

Donut
version: 1.6

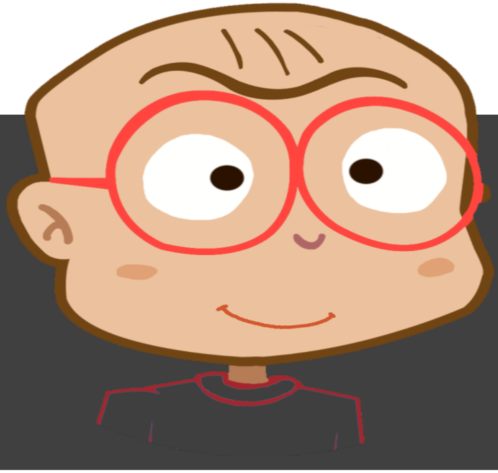
Eclair
version: 2.0 & 2.1

Froyo
version: 2.2

Ginger Bread
version: 2.3

# ListView - Attribute

| Attribute | Description |
|---|---|
| android:id | This is the ID which uniquely identifies the layout. |
| android:divider | This is drawable or color to draw between list items. . |
| android:dividerHeight | This specifies height of the divider. This could be in px, dp, sp, in, or mm. |
| android:footerDividersEnabled | When set to false, the ListView will not draw the divider before each footer view. The default value is true. |
| android:headerDividersEnabled | When set to false, the ListView will not draw the divider after each header view. The default value is true. |

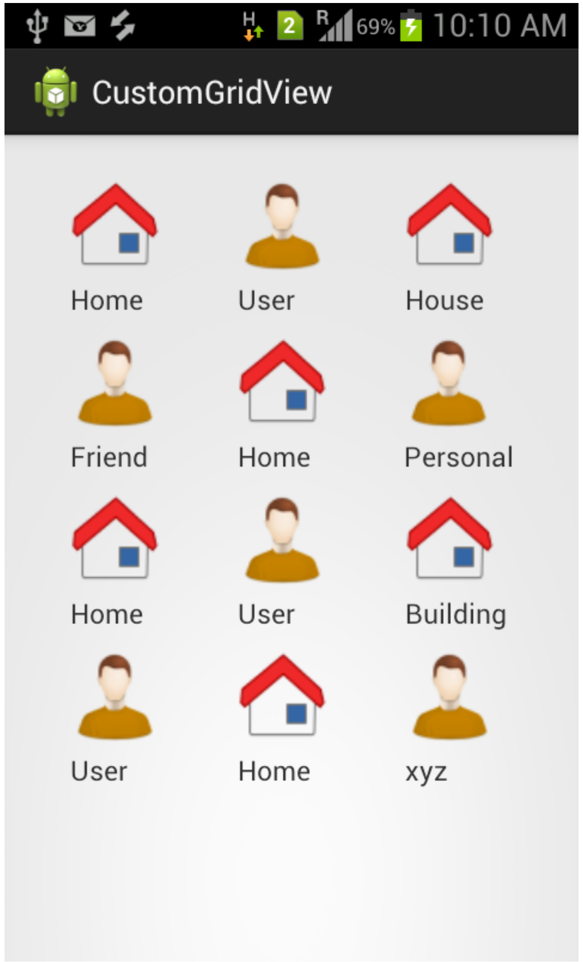


# Grid View

# GridView

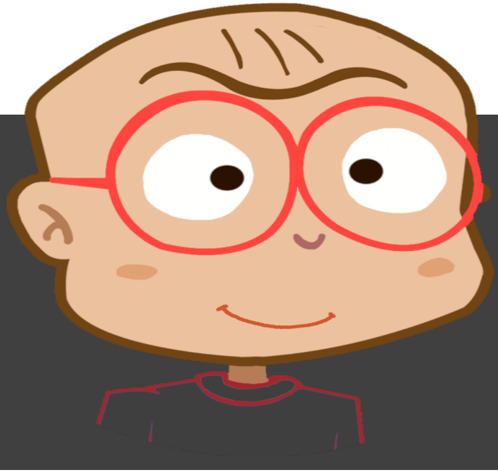
Android **GridView** shows items in two-dimensional scrolling grid (rows & columns) and the grid items are not necessarily predetermined but they automatically inserted to the layout using a **Adapter**

The **GridView** is subclasses of **AdapterView** and they can be populated by binding them to an **Adapter**, which retrieves data from an external source and creates a View that represents each data entry.



# GridView - Attributes

| Attribute | Description |
|---|---|
| android:id | This is the ID which uniquely identifies the layout. |
| android:columnWidth | This specifies the fixed width for each column. This could be in px, dp, sp, in, or mm. |
| android:gravity | Specifies the gravity within each cell. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc. |
| android:horizontalSpacing | Defines the default horizontal spacing between columns. This could be in px, dp, sp, in, or mm. |
| android:numColumns | Defines how many columns to show. May be an integer value, such as "100" or auto_fit which means display as many columns as possible to fill the available space. |
| android:verticalSpacing | Defines the default vertical spacing between rows. This could be in px, dp, sp, in, or mm. |



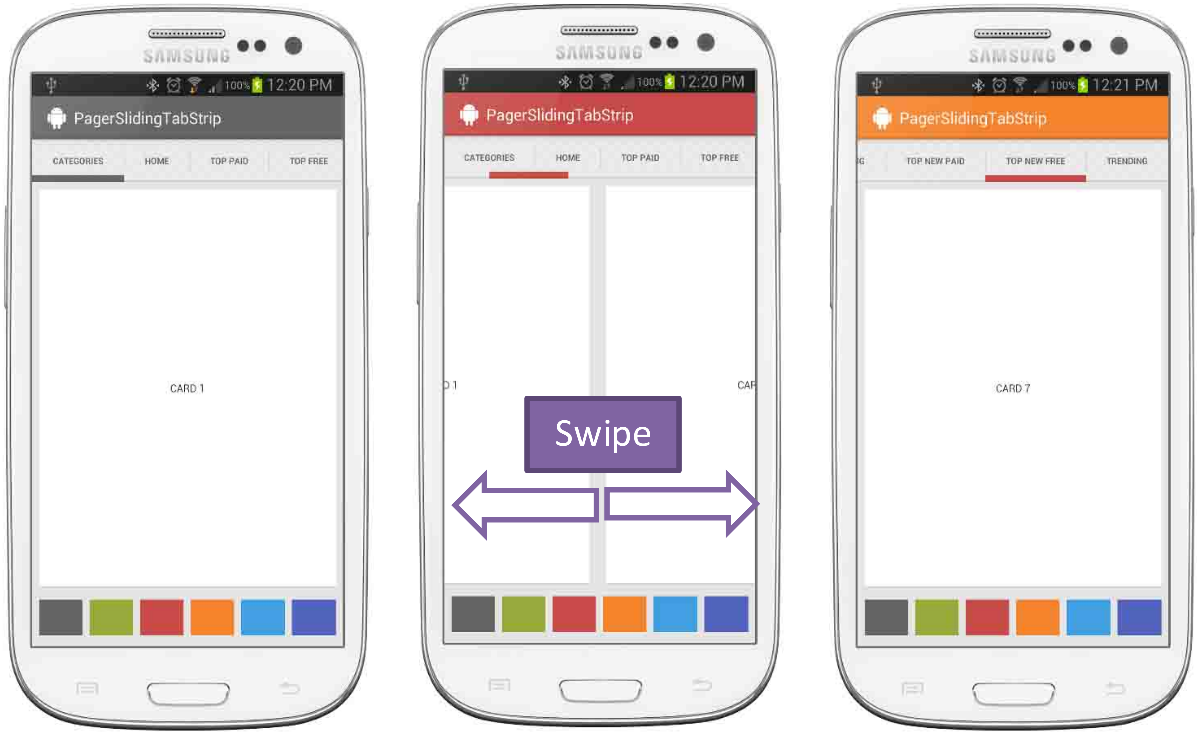
# ViewPager

# ViewPager

The **ViewPager** is the widget that allows the user to swipe left or right to see an entirely new screen. In a sense, it’s just a nicer way to show the user multiple tabs

To generate the pages that the view shows. You must implement:

- PagerAdapter
- FragmentPagerAdapter
- FragmentStatePagerAdapter



# *Exit Course*

THANK YOU