

# Intro to Android

pt. 2

# ListView

- Extends ViewGroup
- Extends AbsListView
- Attributes for adding dividers
- Methods for adding Headers and Footers
- Handles selection
- Depends on an Adapter supplying list items.

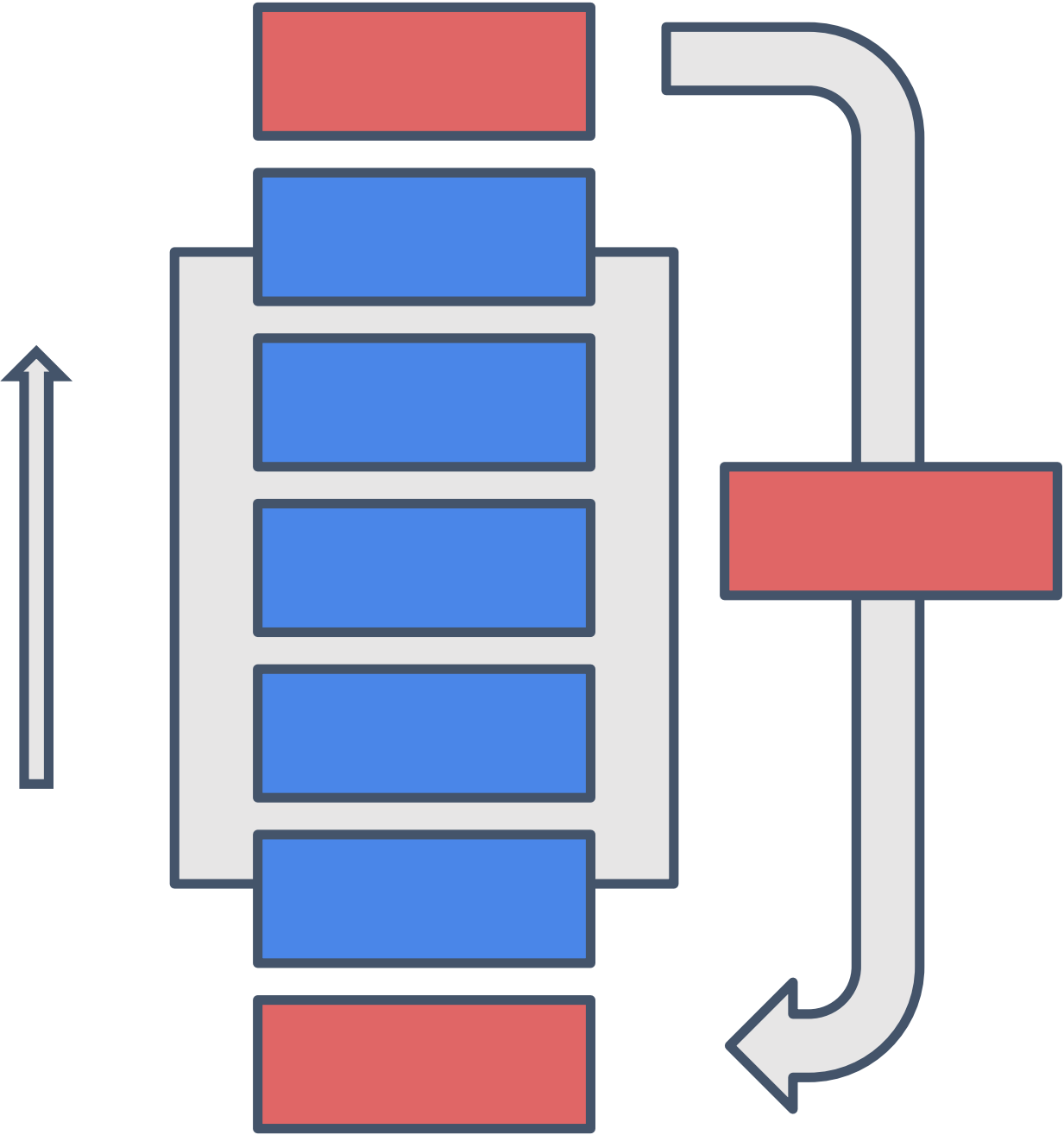
# GridView

- Extends ViewGroup
- Extends AbsListView
- ListView for grids
- Attributes for:
  - cell gravity
  - columnWidth
  - numColumns
  - spacing
  - stretchMode

# support.v7.RecyclerView

- New standard
- Forces the ViewHolder pattern
- More flexible than ListView
- Uses LayoutManager to layout list items
- Uses RecyclerView.Adapter
- Better animation support
- No built in ItemClickListener

Recycling



# support.v7.RecyclerView

```
RecyclerView recyclerView = (RecyclerView) findViewById(R.id.recyclerView);  
recyclerView.setHasFixedSize(true);
```

```
GridLayoutManager layoutManager = new GridLayoutManager(this, 2);  
layoutManager.setOrientation(GridLayoutManager.HORIZONTAL);  
recyclerView.setLayoutManager(layoutManager);
```

```
recyclerView.setAdapter(new RecyclerViewAdapter(DummyContent.TITLES));
```

# RecyclerView.Adapter

```
@Override public ExampleViewHolder onCreateViewHolder(ViewGroup parent,
                                                    int viewType) {

    View view = LayoutInflater
        .from(parent.getContext())
        .inflate(R.layout.listitem_example, parent, false);

    return new ExampleViewHolder(view);
}
```

# RecyclerView.ViewHolder

```
public static class ExampleViewHolder extends RecyclerView.ViewHolder {  
  
    TextView textView;  
  
    public ExampleViewHolder(View itemView) {  
        super(itemView);  
        textView = (TextView) itemView.findViewById(R.id.text);  
    }  
}
```



# RecyclerView.Adapter

```
@Override public void onBindViewHolder(ExampleViewHolder holder, int position) {  
    holder.textView.setText(mNames[position]);  
}  
  
@Override public int getItemCount() {  
    return mNames.length;  
}
```

# Manipulating data

**public void** notifyItemInserted (int position)

**public void** notifyItemChanged (int position)

**public void** notifyItemMoved (int fromPosition, int toPosition)

**public void** notifyItemRemoved (int fromPosition, int toPosition)