

# Android for .NET Developers Series

## Adopting the Android Mindset

Working with Fragments

Jim Wilson

[hedgehogjim.wordpress.com](http://hedgehogjim.wordpress.com)



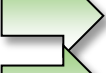
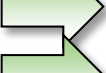
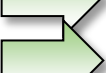


[@hedgehogjim](https://twitter.com/hedgehogjim)

[jimw@jwhh.com](mailto:jimw@jwhh.com)



**pluralsight**   
hardcore developer training

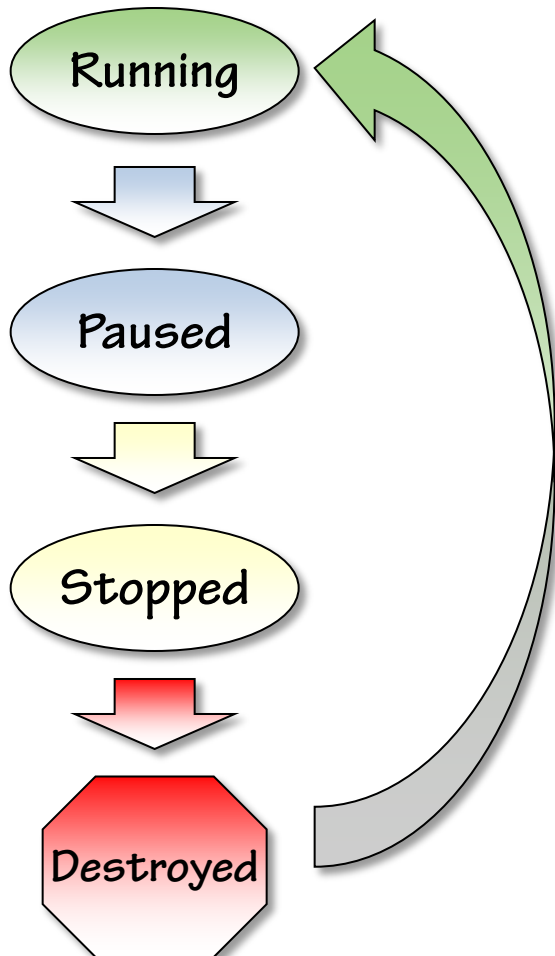
# Outline

-  Course prerequisites and expectations
-  What are Fragments?
-  Fragment availability
-  Creating Fragments
-  Activity/Fragment relationship
-  Coordinating Activities and Fragments
-  Swipe navigation

# Course prerequisites and expectations



**Android for .NET Developers:  
Building Apps With Android Studio**



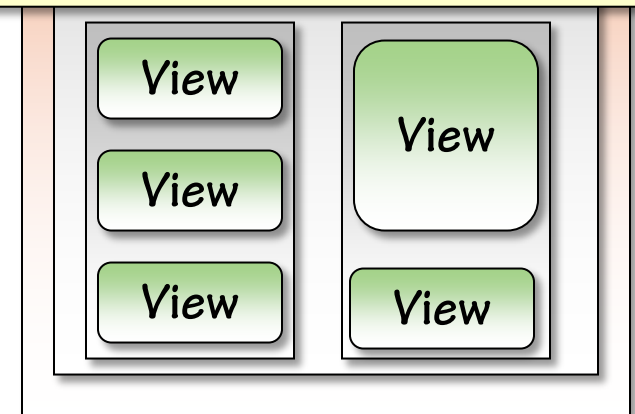
onActivityResult



```
<manifest . . . >
. . .
<application . . . >
  <activity
    android:name=".MainActivity" . . . />
  </application>
. . .
</manifest>
```



Android Studio





# Adopting the android mindset

*Partition the user interface*

# Fragments

- ➡ **Fragments enable dividing the UI into sections**
- ➡ Provide a group of user interface elements and their behavior
  - ❑ Can hold the same UI elements as an Activity
  - ❑ Can contain logic
- ➡ Create logical UI units
- ➡ Allow reorganizing UI for device differences
  - ❑ Can use static layout files
  - ❑ Can be assembled dynamically in code
- ➡ A key part of a many navigation behaviors
  - ❑ Page-based navigation
  - ❑ Tab navigation
  - ❑ List-based navigation

# Fragment Availability

- ➡ **Fragments supported by 99.9% of active Android devices**
- ➡ Native OS support for devices running Android 3.1 or newer
  - API level 12 and above
- ➡ Compatibility library support for devices running Android 1.6 through 2.3
  - API levels 4 through 10
  - Available from ...
    - <http://bit.ly/AndroidV4SupportLib>
  - Example use ...
    - <http://bit.ly/AndroidFragmentCompat>

*\*Although Android 3.0 (API Level 11) technically exists – no devices in the marketplace run it*

# Creating Fragments

## → Creating a Fragment requires a few simple steps

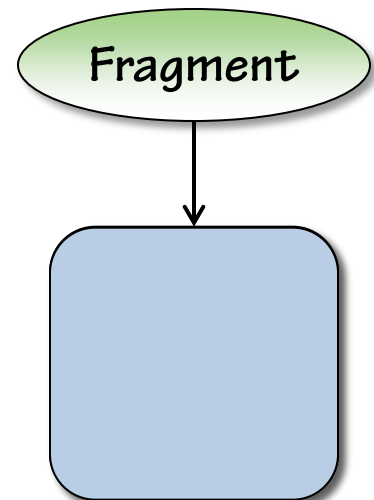
→ Declare a class that inherits from Fragment class

→ Provide the fragment's display contents

- Involves overriding onCreateView and/or onCreate
- Often uses an XML-based layout description much like an Activity
- Specialized Fragment-derived classes simplify special cases
  - More to come on this

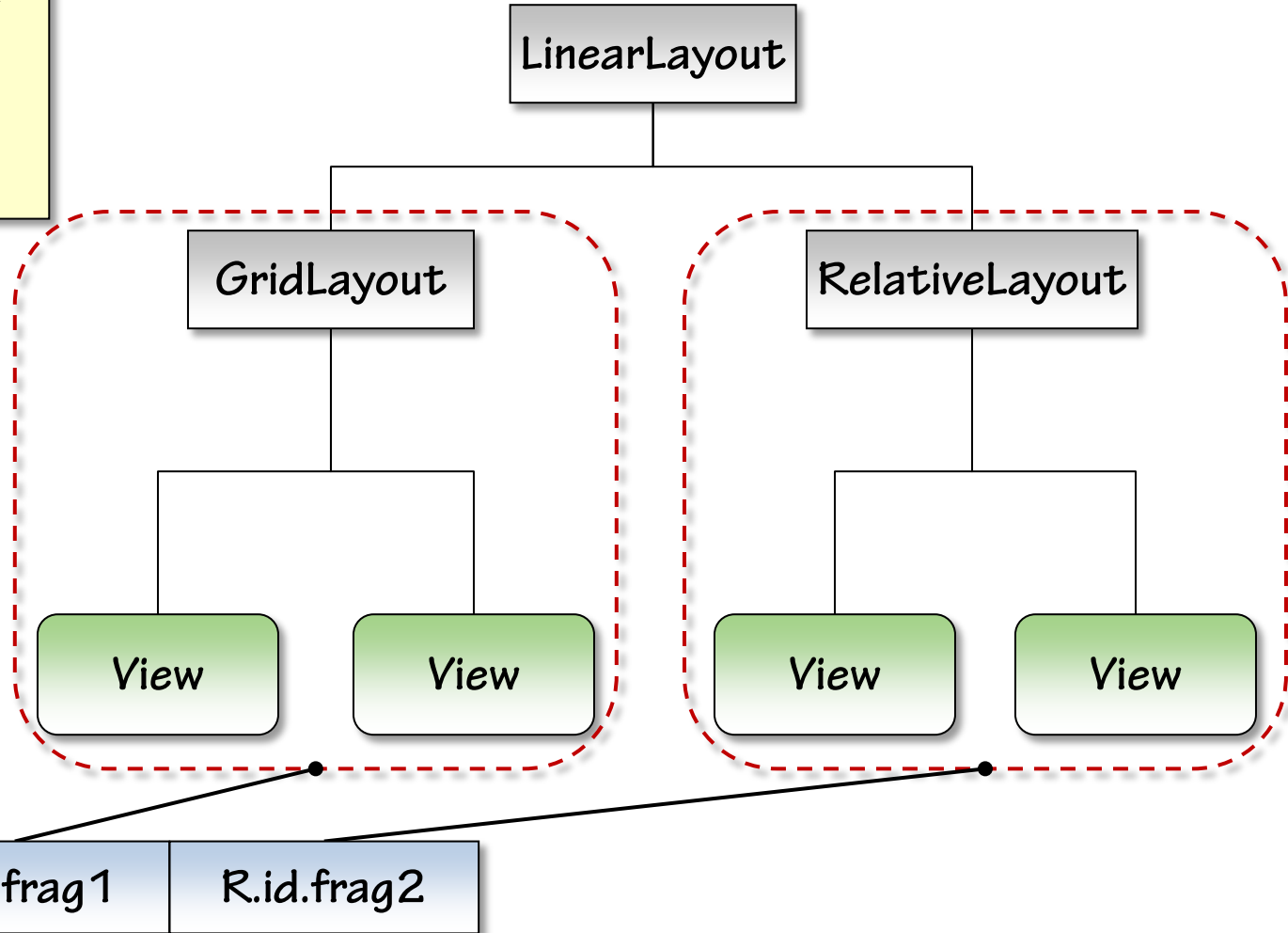
## → Attach your fragment to an Activity

- Use the <fragment> element in the Activity's XML layout
  - The "class" attribute identifies the fully qualified name of fragment class to create
- You can attach as many fragments to a view as you'd like

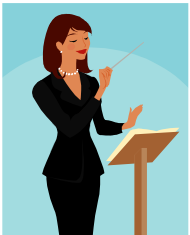


# Structure of an Activity containing Fragments

```
<LinearLayout ...>  
  <fragment  
    name="com.ps.frag1"  
    id="@+id/frag1" .../>  
  <fragment  
    name="com.ps.frag2"  
    id="@+id/frag2" .../>  
</LinearLayout>
```



FragmentManager





# Summary

- ➡ **Partition the user interface**
- ➡ **Fragments enable dividing UI into sections**
  - ❑ Self-contained UI elements and code
  - ❑ Allow reorganizing UI for device differences
  - ❑ Central to many navigation behaviors
- ➡ **Fragments available to virtually all devices**
  - ❑ Specific package depends on target platform
- ➡ **All Fragments inherit directly or indirectly from the Fragment class**
  - ❑ Populate contents by overriding appropriate callback method
- ➡ **FragmentManager coordinates access to an Activity's Fragments**