Motivation

- Android development is harder than it needs to be.
  - Many common tasks that should be simple aren't.

- Stanford (Marty) is creating a library to make it simpler:

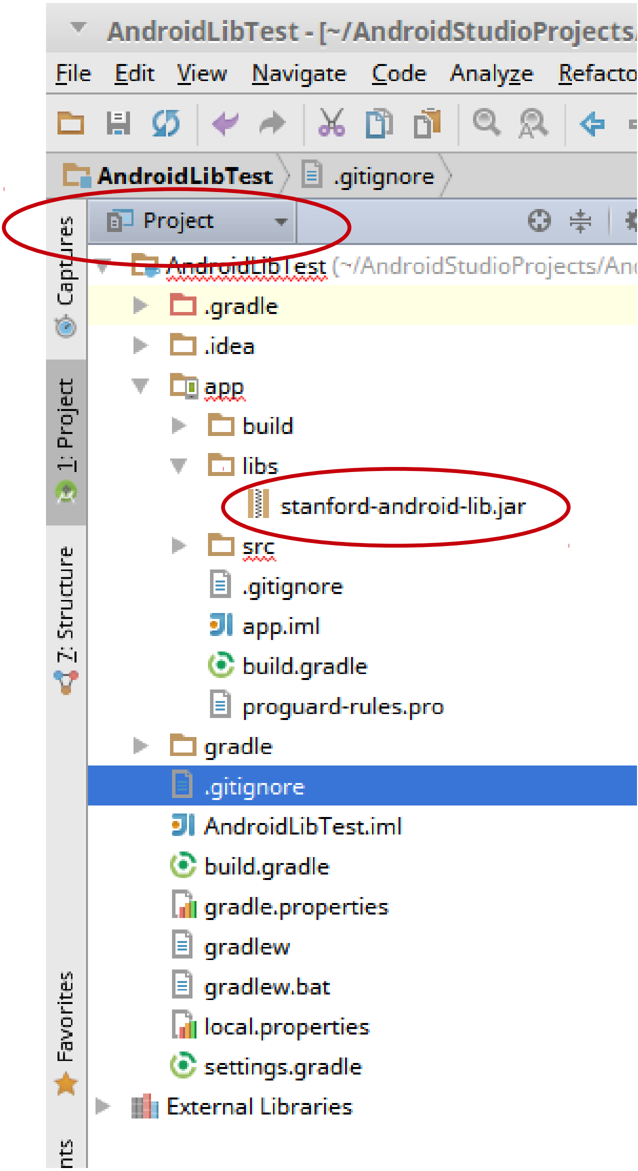
  ~~`public class MyActivity extends Activity {`~~
  `public class MyActivity extends SimpleActivity {`

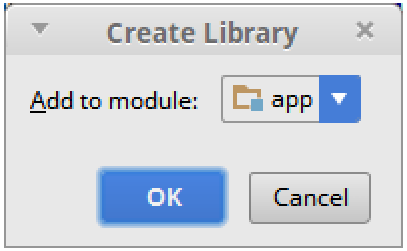
  - The `SimpleActivity` class provides lots of convenience methods and functionality for simplifying common Android tasks.

Using the library

- Download library JAR from class web site:
  - http://cs193a.stanford.edu/lib/

- Attach the .JAR file to your project:
  - Put the JAR in your project's app/libs/ folder.
  - In Android Studio:
    - make sure you are in "Project" view mode.
    - scroll down to app/libs/ folder.
    - right-click the JAR.
    - choose "Add as Library" near the bottom.
    - add the lib to your module named "app".





Add to module: app

Another way to add library

- Download library JAR from class web site:
  - http://cs193a.stanford.edu/lib/

- Attach the .JAR file to your project:
  - Put the JAR in your project's app/libs/ folder.
  - In Android Studio:
    - Open the **build.gradle** file for your app.
    - Find the section called 'dependencies'.
    - Add the following line inside that section.

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    ...
    compile files('libs/stanford-android-lib.jar')
}
```

Accessing widgets by IDs

`findButton(`***`id`***`)`	returns Button for given ID
`findCalendarView, findCheckBox, findDatePicker, findEditText, findFragment, findGridView, findImageButton, findImageView, findListView, findProgressBar, findRadioButton, findRadioGroup, findRatingBar, findScrollView, findSearchView, findSeekBar, findSpace, findSpinner, findStackView, findSwitch, findTextView, findTimePicker, findToggleButton, findToolbar, findZoomButton`	returns widget of given type that has the given ID

`find(`***`id`***`)`
`$(`***`id`***`)`
	alias for `findViewById` but using generics to avoid casts

`$B(`***`id`***`), $CB(`***`id`***`), $ET(`***`id`***`), $IB(`***`id`***`), $IV(`***`id`***`), $LV(`***`id`***`), $RB(`***`id`***`), $TV(`***`id`***`), ...`	alias for `find` but casts to Button, CheckBox, TextView, ...

```
// access widgets by ID without needing to cast
Button button = $B(R.id.mybutton);
ListView list = $LV(R.id.mylist);
TextView text = $(R.id.mytext);
$TV(R.id.mytext).setText("hello!");
...
```

Logging, printing, toasts

Method	Description

`log("`***message***`");`
`log(`***exception***`);`
`log("`***message***`", `***exception***`);`
	equivalent to `Log.d`

`println("`***message***`");`
`printf("`***formatStr***`", `***args***`);`
	equivalent to `Log.v`

`toast("`***message***`");`
`toast("`***message***`", `***time***`);`
	equivalent to `Toast.makeText`

```
// slightly easier printing of debug/toast messages
// (these methods are in SimpleActivity)
println("A message from SimpleActivity");
toast("A toast message");
```

The "with" pattern

```
// Many Android libraries use a pattern of
// ClassName.with(this)
//          .methodName();
//
// where 'this' is your Activity

ListView list = $(R.id.mylist);
SimpleList.with(this)
    .setItems(list, "Leo", "Mike", "Don", "Raph");
```

SimpleList

Method	Description
createAdapter(***items***)	create/return an ArrayAdapter
createAdapter(***item1***, ***item2***, ..., ***itemN***)	create/return an ArrayAdapter

getItems(***id***)
getItems(***listView***)
	return items as ArrayList

setItems(***id***, ***items***);
setItems(***listView***, ***items***);
	set items from ArrayList

setItems(***listView***, ***item1***, ***item2***, ..., ***itemN***);	set items in list view

```
// easy get/set of ListView items
SimpleList.with(this)
    .setItems(R.id.mylist, "Leo", "Mike", "Don", "Raph");
```

Standard list events

```
// normal crappy code to hear list item click events
ListView list = findListView(R.id.mylist);
list.setOnItemClickListener(
    new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent,
                View view, int index, long id) {
            // phew! event handler code goes here :-(
        }
    }
);
```

Easier list events

```
// SimpleActivity code to hear list item click events
ListView list = findListView(R.id.mylist);
list.setOnItemClickListener(this);
...

public void onItemClick(ListView list, int index) {
    // event handler code goes here :-)
}

// also available:
//      - onItemLongClick
//      - onItemSelected
//      - other similar events for other widget types
```

SimpleIO

Method	Description

openExternalFileBufferedReader("*filename*")
openExternalFileScanner("*filename*")
	read file in external storage

openExternalFilePrintStream(*filename*)	write file in external storage

openInternalFileBufferedReader(*id*)
openInternalFileScanner(*id*)
	read file in internal storage

readFileLines(*id*) // internal
readFileLines(*filename*) // external
	read file and return its lines as an ArrayList of strings

readFileText(*id*) // internal
readFileText(*filename*) // external
	read file and return its text as a String

writeFileLines(*filename*, *list*); // external
writeFileText(*filename*, *text*);
	write contents of a list or string to an external file

```
// more easily read and write files
Scanner scan = SimpleIO.with(this)
               .openInternalFileScanner(R.raw.myfile);
while (scan.hasNextLine()) { ... }
```

System directories

Method	Description
`getDocumentsDirectory()`	dir where docs are stored
`getDownloadsDirectory()`	dir where downloads are stored
`getMoviesDirectory()`	dir where movies are stored
`getMusicDirectory()`	dir where music/songs are stored
`getPhotosDirectory()`	dir where pictures are stored

```
// write to a file in the documents directory
File dir = SimpleIO.with(this).getDocumentsDirectory();
PrintStream out = SimpleIO.with(this)
    .openExternalPrintStream(dir, "myfile.txt");
out.println("this is a test");
out.close();
```

SimpleMedia

Method	Description
`play(id);`	play/unpause sound with given ID
`loop(id);`	repeatedly plays sound
`pause(id);`	pause sound if playing
`stop(id);`	stops the given sound if playing
`isPlaying(id)`	returns true if the sound is playing
`isLooping(id)`	returns true if the sound is looping
`getPosition(id)`	returns time index of playing clip in MS
`setPosition(id, ms)`	advances the clip to the given time

```
// convenience methods for playing sounds
SimpleMedia.with(this).play(R.id.cowabunga);
SimpleMedia.with(this).loop(R.id.tmnt_theme);
```