

# Uni Studies 3: Intro to Processing 3

Assoc. Professor Donald J. Patterson  
Uni Stu 3 Fall 2012



# A brief history of Android

- Originally an independent startup that “developed software for mobile phones”
- Business Week quote from founder Andy Rubin in 2003:
  - "Rubin said there was tremendous potential in developing smarter mobile devices that are more aware of its owner's location and preferences.



# What is Android?

- “Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.”

ANDROID



# A brief history of Android

- iPhone launched in 2007



# A brief history of Android

- Rumors of the “gPhone” started about the time the iPhone launched in 2007

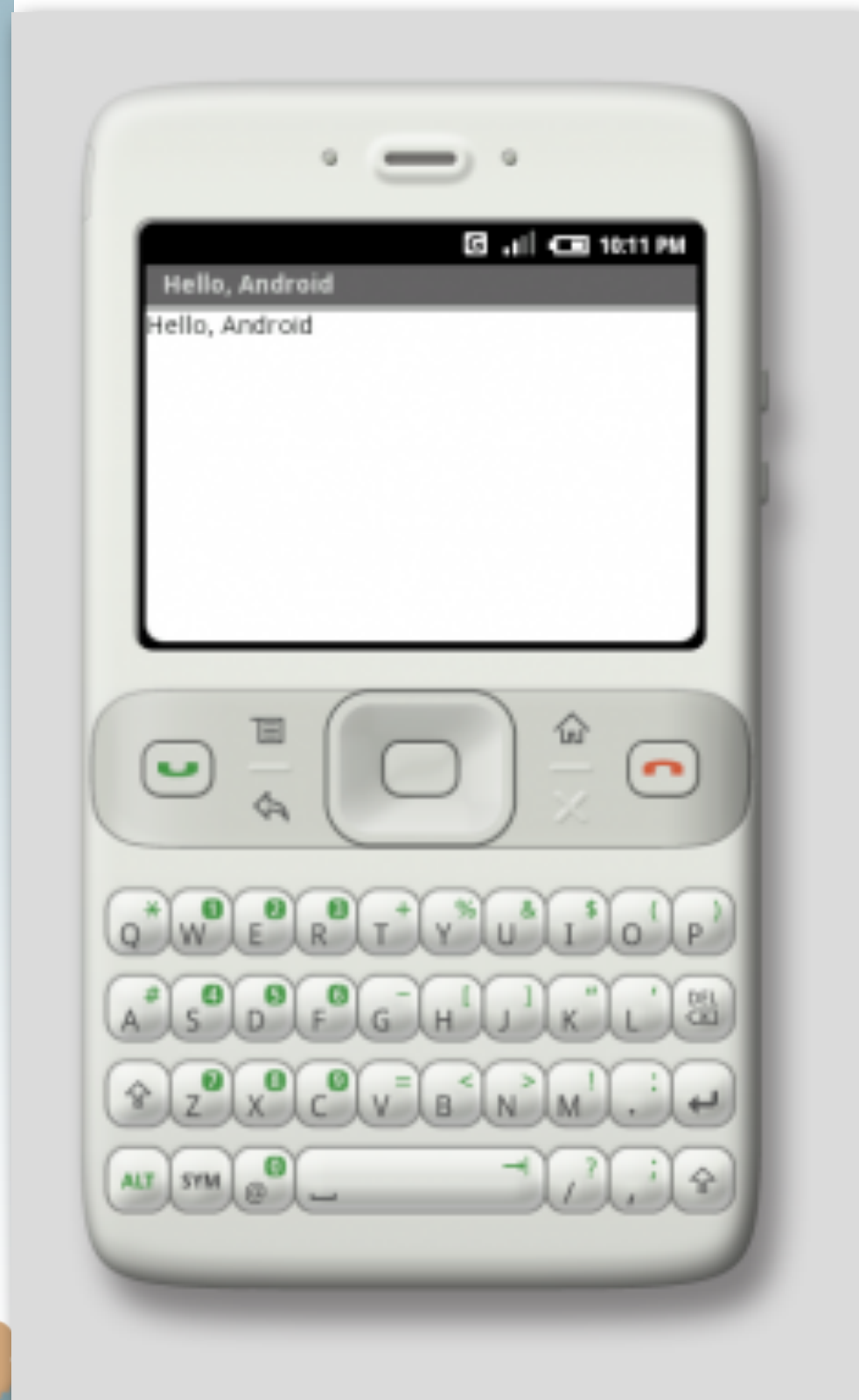


# gPhone concepts



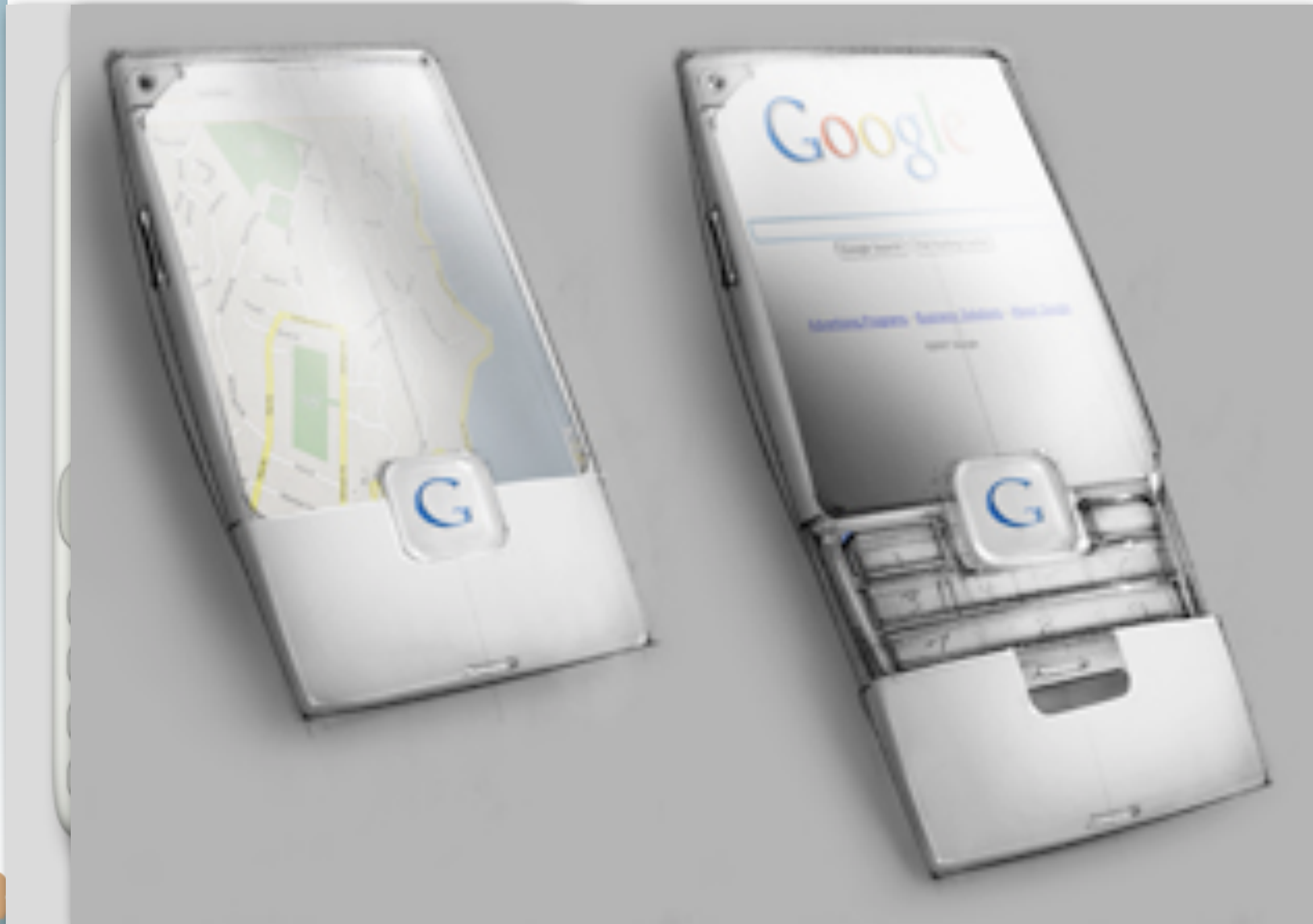
<http://www.google-phone.com/google-phone-or-gphone-concept-designs>

# gPhone concepts



<http://www.google-phone.com/google-phone-or-gphone-concept-designs>

# gPhone concepts



<http://www.google-phone.com/google-phone-or-gphone-concept-designs>



# gPhone concepts



<http://www.google-phone.com/google-phone-or-gphone-concept-designs>

# gPhone concepts



<http://www.google-phone.com/google-phone-or-gphone-concept-designs>

# gPhone concepts



Google



<http://www.google-phone.com/google-phone-or-gphone-concept-designs>

# gPhone concepts



<http://www.google-phone.com/google-phone-or-gphone-concept-designs>



# gPhone concepts



<http://www.google-phone.com/google-phone-or-gphone-concept-designs>

# A brief history of Android

- Rumors of the “gPhone” started about the time the iPhone launched
- Google dropped the bomb on 11/5/2007
  - It wasn't working on a handset
  - It was working on an operating system
    - to compete with Microsoft, Symbian, telephone companies. Who else ?



# A brief history of Android

- Why would Google do this?
  - It doesn't want to be locked out of mobile advertising
  - It's the same reason they support Firefox
  - It's the same reason they built Chrome
  - It's the same reason they bought YouTube
  - It's about maintaining access to advertising channels today, and having access to tomorrow's innovations
  - Competitors don't need to let Google advertise



## A brief history of Android

- Actually it wasn't just Google
- It was the Open Handset Alliance (OHA)
  - including HTC, LG, Samsung, T-Mobile and more
  - pushing
- Based on Linux
  - optimized for mobile devices



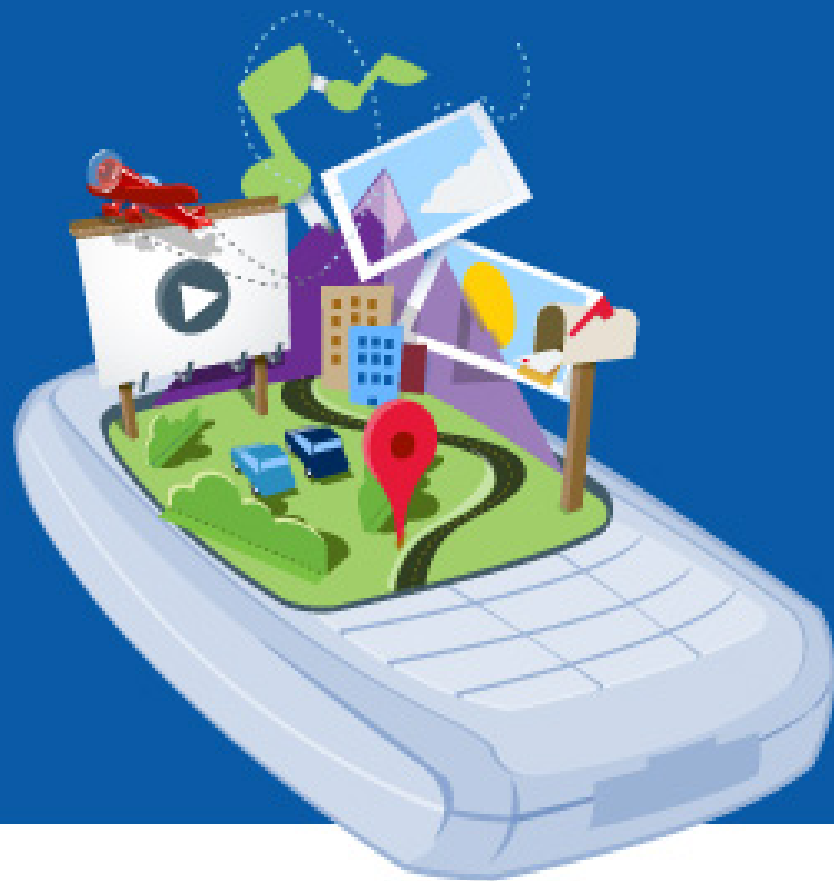


## A brief history of Android

- The architecture is highly modular
  - “Location” can come from many places
  - Text messaging handling can be done by any software component
- It is predominantly open-source
- It is predominantly Java-based



## A brief history of Android



### What would it take to build a better mobile phone?

A commitment to openness, a shared vision for the future, and concrete plans to make the vision a reality.

Welcome to the Open Handset Alliance™, a group of 47 technology and mobile companies who have come together to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience. Together we have developed Android™, the first complete, open, and free mobile platform.

We are committed to commercially deploy handsets and services using the Android Platform.

- **Develop Android applications:** [Get the SDK](#)
- **Contribute to the Android Open Source Project:** [Get the source code](#)

<http://www.openhandsetalliance.com>

- <http://www.youtube.com/watch?v=7Y4thikv-OM>



# A brief history of Android

- First SDK was released on 11/12/2007
- Main conceptual competitor is LiMO
  - Linux for Mobile
    - Verizon and Mozilla key initial players
    - Many partners in both projects
    - Outside Asia, no one cares



# A brief history of Android

- 6/24/2008 Nokia announces purchase of Symbian from Sony Ericsson in response
  - starts the Symbian Foundation to open-source their main platform, gets released in 2011 as “shared-source”
- 10/21/2008, Google puts all of Android into open source using the Apache License
- 12/9/2008 Sony Ericsson joins the OHA
  - Google denies rumors of the gPhone at the same time



<http://www.engadget.com/2008/06/24/nokia-buys-symbian/>

# A brief history of Android

- 8/12/2010 Oracle sues Google over intellectual property in Java
- 2/11/2011 Nokia announces it is abandoning Symbian for Windows 7



<http://www.engadget.com/2008/06/24/nokia-buys-symbian/>

# Overview of Android

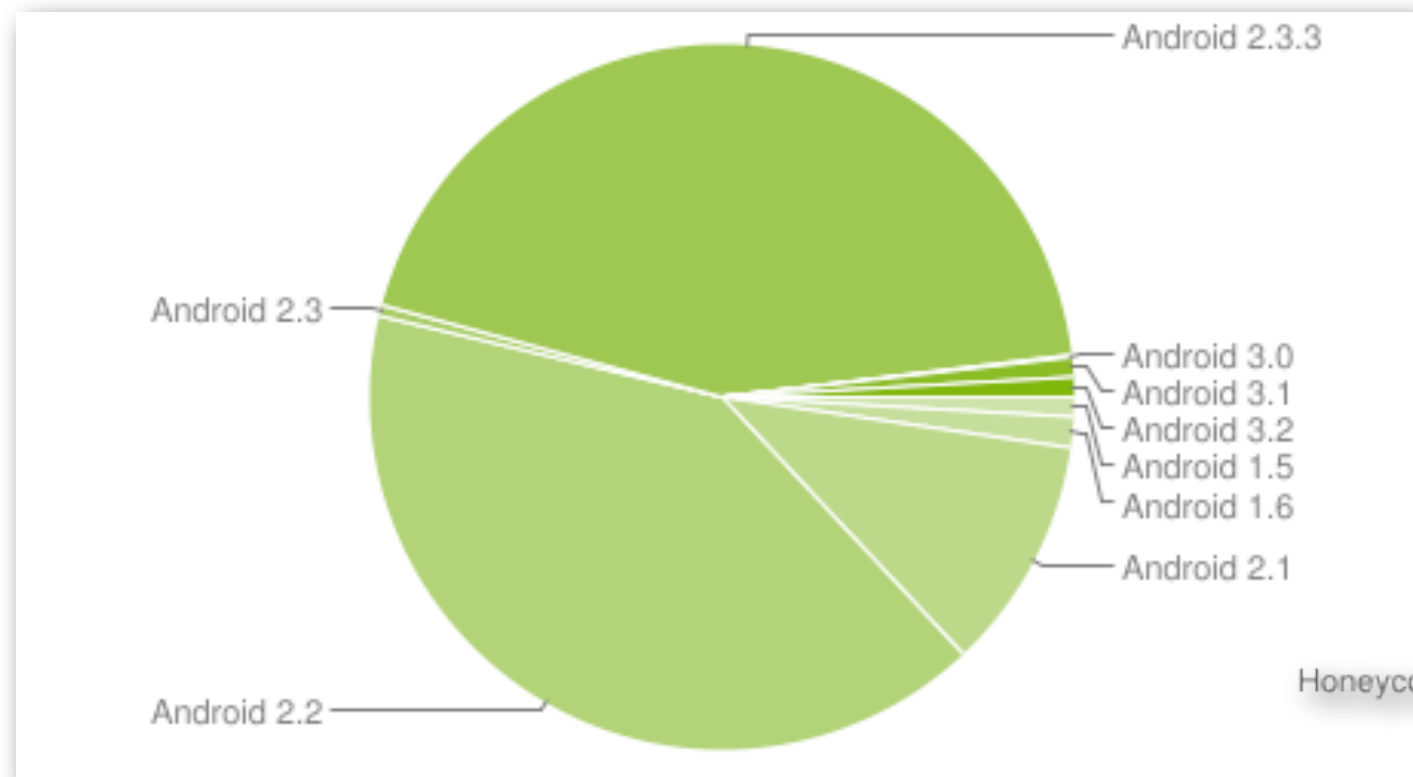
- Platforms running Android
  - in 2009
    - T-Mobile G1 phone
    - HTC developer phone
  - in 2011
    - Hundreds
  - in 2012
    - Silly



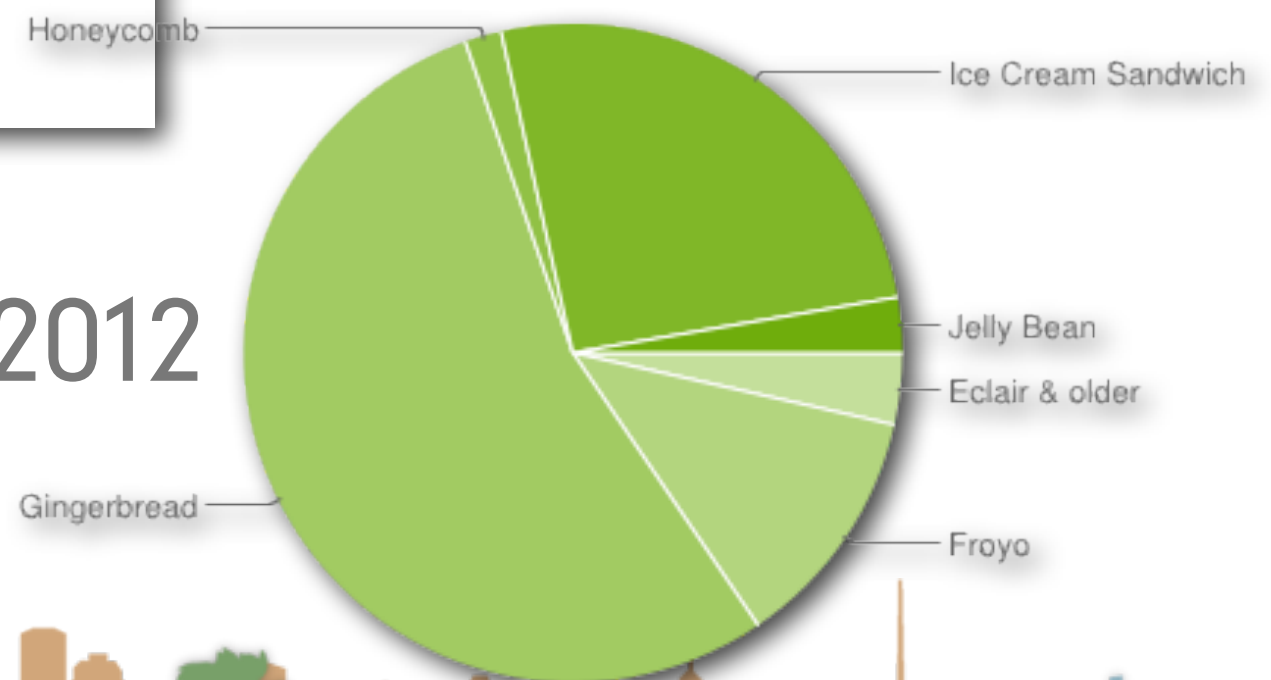
## Overview of Android

- Various code revisions are named after pastries (sort of)
  - Cupcake (released on 4/30/2009) v1.5
  - Donut (released on 9/15/2009) v1.6
  - Eclair (released on 10/26/2009) v2.1
  - Froyo (released on 5/20/2010) v2.2
  - Gingerbread (?) v2.3 (SIP, NFC)
  - Honeycomb (5/2011) v3.0 (tablet)
  - Ice Cream Sandwich (11/14/2011) v4.0 (face recognition)
  - Jelly Bean (11/13/2012) v4.2 (Google Now)

## Overview of Android 11/2011



## Overview of Android 11/2012





# Overview of Android

- Google play (née Android Marketplace)
  - Place to get 3rd-party android apps
  - Utilizes Google Checkout for payment
  - Not exclusive source for applications
  - Applications are not “blessed” at all by Google
    - It is merely a convenience for developers

A stylized illustration of a city skyline at the bottom of the slide. It features various building shapes in shades of orange, blue, and green, with some palm trees interspersed among the buildings.

<https://play.google.com/store?hl=en>


[Home](#) > [Apps](#) > [Lifestyle](#)

## Easy Tie

basesign



★★★★★ (98)

\$1.99 BUY

### More from developer



#### Easy Tie Lite

BASESIGN

★★★★★ (557)

Free



#### Lesari - Meine FilmDatenba

BASESIGN

★★★★★ (42)

Free



#### Calc n Con

BASESIGN

★★★★★ (2)

\$1.40

### Users who viewed this also viewed



#### Tie Deluxe

JQ SOFT

★★★★★ (31)

\$0.99

### Users who installed this also installed

### OVERVIEW

### USER REVIEWS

### WHAT'S NEW

### PERMISSIONS

## Description

Don't know how to tie a tie?

Learn it with Easy Tie!

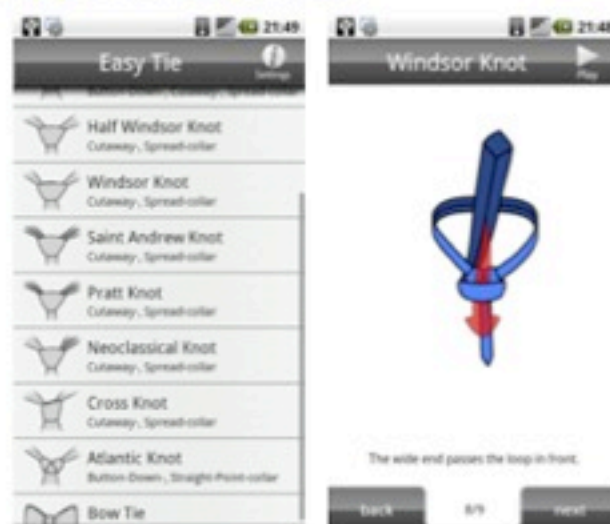
Easy Tie is an app that shows you step-by-step how to tie a necktie.

Knots included:

- Simple Knot
- Double K.
- Small K.
- Half Windsor K.
- Windsor K.

[Visit Developer's Website](#)

## App Screenshots



MORE

+1 10

[Tweet](#)

### ABOUT THIS APP

RATING:

★★★★★ (98)

UPDATED:

June 21, 2010

CURRENT VERSION:

1.6

REQUIRES ANDROID:

1.5 and up

CATEGORY:

Lifestyle

INSTALLS:

1,000 - 5,000



SIZE:

1.0M

PRICE:

\$1.99

CONTENT RATING:

Everyone

## User Reviews

5 star 71

4 star 10

Average rating:



## Easy Tie

basesign



★★★★★ (106)

\$1.99 BUY

⚠ You don't have any devices.

### More from developer



#### Easy Tie Lite

BASESIGN

★★★★★ (589)

Free



#### Lesari - Meine FilmDatenbank

BASESIGN

★★★★★ (69)

Free



#### Calc n Con

BASESIGN

★★★★★ (2)

\$1.40

See more

### Users who viewed this also viewed



#### How to knot a tie

MARC WEBER

★★★★★ (1,349)

### OVERVIEW

### USER REVIEWS

### WHAT'S NEW

### PERMISSIONS

## Description

Don't know how to tie a tie?

Learn it with Easy Tie!

Easy Tie is an app that shows you step-by-step how to tie a necktie.

Knots included:

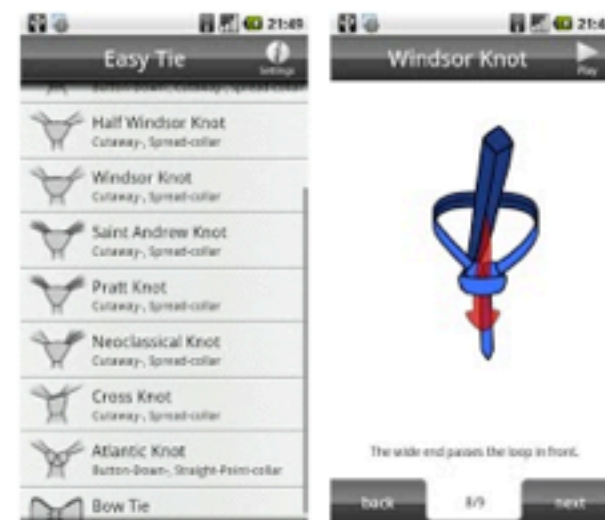
- Simple Knot
- Double K.
- Small K.
- Half Windsor K.
- Windsor K.
- Saint Andrew K.

[Visit Developer's Website](#)

[Email Developer](#)

MORE

## App Screenshots



## User Reviews

[Write a Review](#)

5 star

79

Average rating

+1 36

[Tweet](#)

### ABOUT THIS APP

RATING:

★★★★★ (106)

UPDATED:

June 21, 2010

CURRENT VERSION:

1.6

REQUIRES ANDROID:

1.5 and up

CATEGORY:

Lifestyle

INSTALLS:

5,000 - 10,000

SIZE:

1.0M

PRICE:

\$1.99

CONTENT RATING:

Everyone

# Android Features (hardware dependent)

- Rich development environment
  - a device emulator
  - tools for debugging
  - tools for memory profiling
  - tools for performance profiling
  - plugin for Eclipse IDE

A stylized illustration of a city skyline at the bottom of the slide. It features various buildings in shades of orange, brown, and blue, with green palm trees and foliage in the foreground. The URL is overlaid on this graphic.

<http://developer.android.com/guide/index.html>

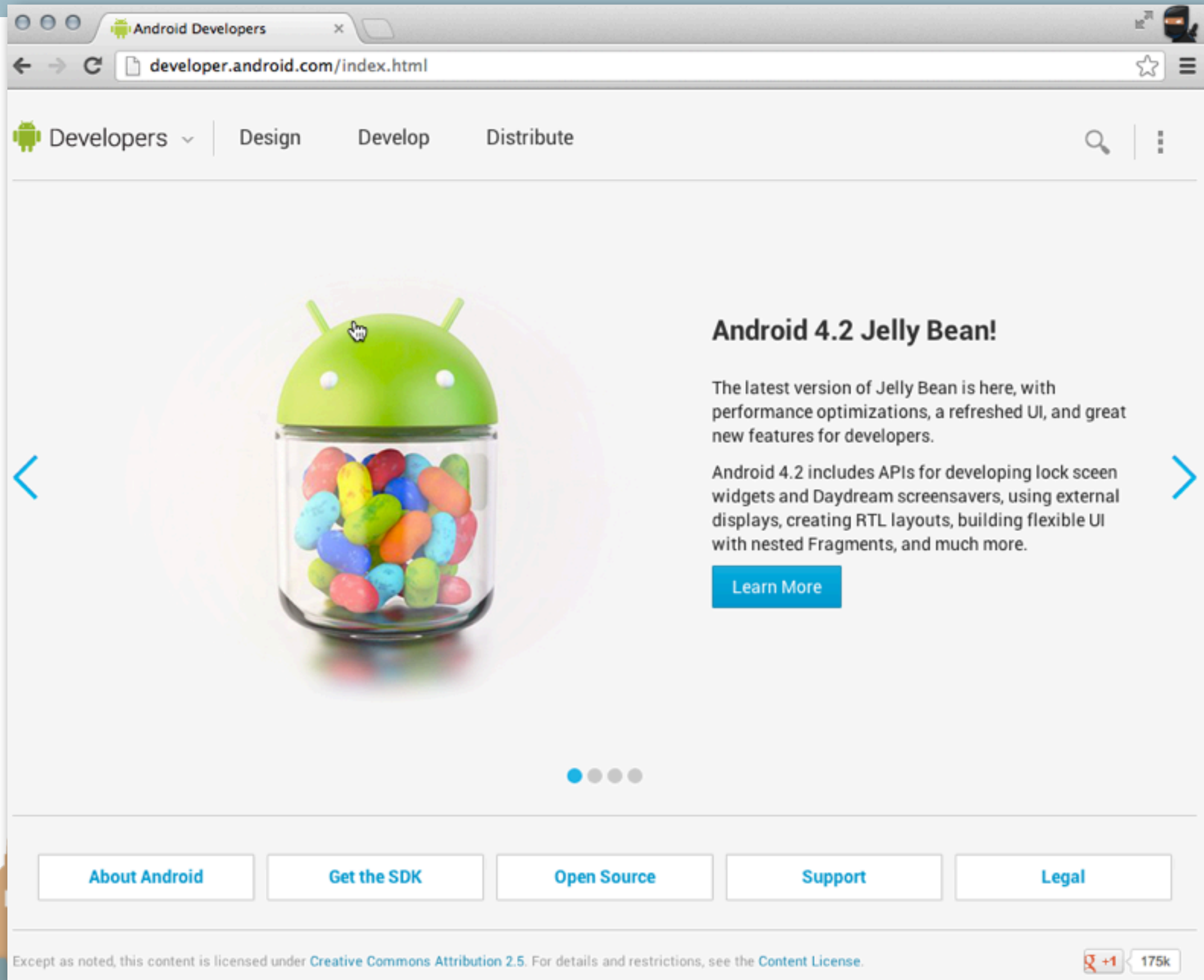
# Actually Developing for Android

- Download and install Eclipse (IDE)
- Download and install the Android SDK (external)
  - <http://developer.android.com/sdk/index.html>
- Download and install the ADT plugin
  - Use Eclipse's built-in installer
  - Set up SDK path
- Download and install the Android Platform (internal)
- Make a virtual device



<http://developer.android.com/guide/index.html>

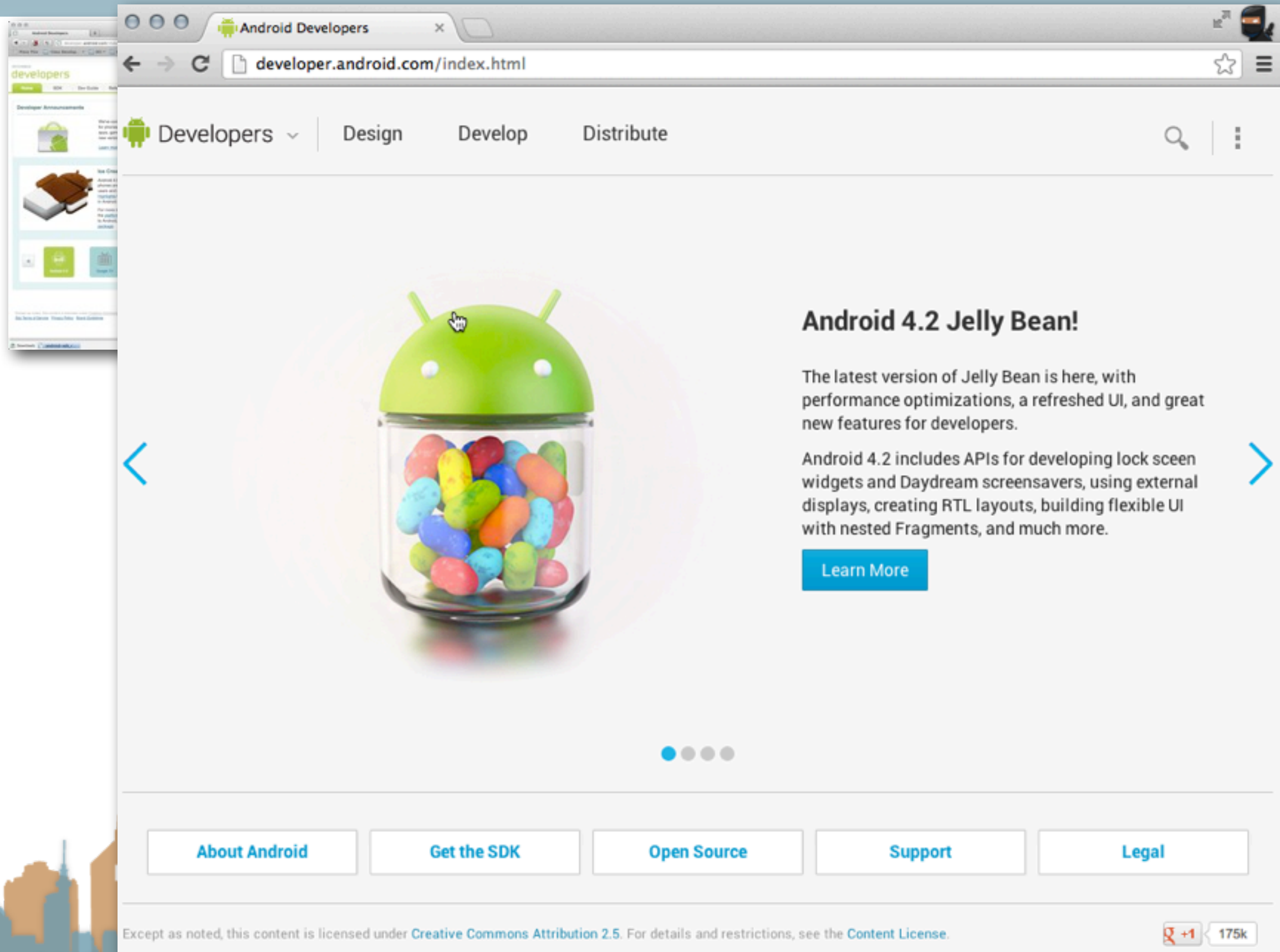
# Intro to Android



<http://developer.android.com/guide/index.html>



# Intro to Android



The screenshot shows a web browser window with the URL `developer.android.com/index.html`. The page features a navigation bar with links for **Developers**, **Design**, **Develop**, and **Distribute**. The main content area is titled **Android 4.2 Jelly Bean!** and includes a large image of the Android robot head on top of a glass jar filled with colorful jelly beans. To the right of the image, the text reads: "The latest version of Jelly Bean is here, with performance optimizations, a refreshed UI, and great new features for developers." Below this, it states: "Android 4.2 includes APIs for developing lock screen widgets and Daydream screensavers, using external displays, creating RTL layouts, building flexible UI with nested Fragments, and much more." A blue button labeled **Learn More** is positioned below the text. At the bottom of the page, there are five white buttons with blue text: **About Android**, **Get the SDK**, **Open Source**, **Support**, and **Legal**. A footer at the very bottom contains a Creative Commons license notice and a Google+ social share button showing 175k shares.

Android Developers

developer.android.com/index.html

Developers Design Develop Distribute

## Android 4.2 Jelly Bean!

The latest version of Jelly Bean is here, with performance optimizations, a refreshed UI, and great new features for developers.

Android 4.2 includes APIs for developing lock screen widgets and Daydream screensavers, using external displays, creating RTL layouts, building flexible UI with nested Fragments, and much more.

[Learn More](#)

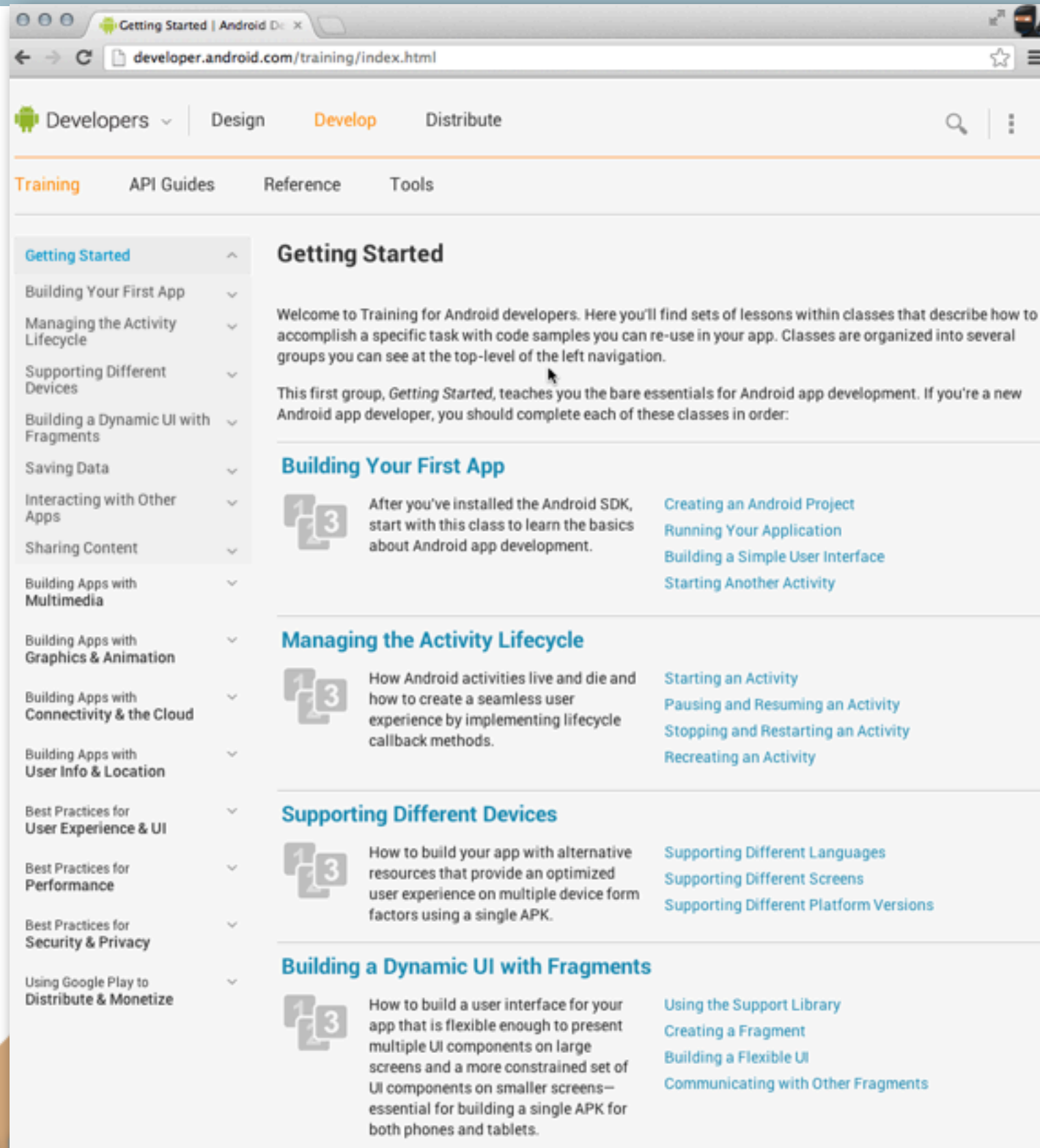
[About Android](#) [Get the SDK](#) [Open Source](#) [Support](#) [Legal](#)

Except as noted, this content is licensed under [Creative Commons Attribution 2.5](#). For details and restrictions, see the [Content License](#).

[+1](#) 175k

<http://developer.android.com/guide/index.html>

# Intro to Android

A screenshot of a web browser displaying the 'Getting Started' page of the Android Developer Training Index. The browser's address bar shows 'developer.android.com/training/index.html'. The page has a navigation bar with 'Developers', 'Design', 'Develop', and 'Distribute'. Below this is a sub-navigation bar with 'Training', 'API Guides', 'Reference', and 'Tools'. The main content area is titled 'Getting Started' and includes a welcome message, a list of recommended classes for new developers, and links to various training modules. The left sidebar lists all available training topics, each with a dropdown arrow.

Getting Started | Android De x

developer.android.com/training/index.html

Developers | Design | **Develop** | Distribute

Training | API Guides | Reference | Tools

## Getting Started

Welcome to Training for Android developers. Here you'll find sets of lessons within classes that describe how to accomplish a specific task with code samples you can re-use in your app. Classes are organized into several groups you can see at the top-level of the left navigation.

This first group, *Getting Started*, teaches you the bare essentials for Android app development. If you're a new Android app developer, you should complete each of these classes in order:

### Building Your First App

After you've installed the Android SDK, start with this class to learn the basics about Android app development.

- Creating an Android Project
- Running Your Application
- Building a Simple User Interface
- Starting Another Activity

### Managing the Activity Lifecycle

How Android activities live and die and how to create a seamless user experience by implementing lifecycle callback methods.

- Starting an Activity
- Pausing and Resuming an Activity
- Stopping and Restarting an Activity
- Recreating an Activity

### Supporting Different Devices

How to build your app with alternative resources that provide an optimized user experience on multiple device form factors using a single APK.

- Supporting Different Languages
- Supporting Different Screens
- Supporting Different Platform Versions

### Building a Dynamic UI with Fragments

How to build a user interface for your app that is flexible enough to present multiple UI components on large screens and a more constrained set of UI components on smaller screens—essential for building a single APK for both phones and tablets.

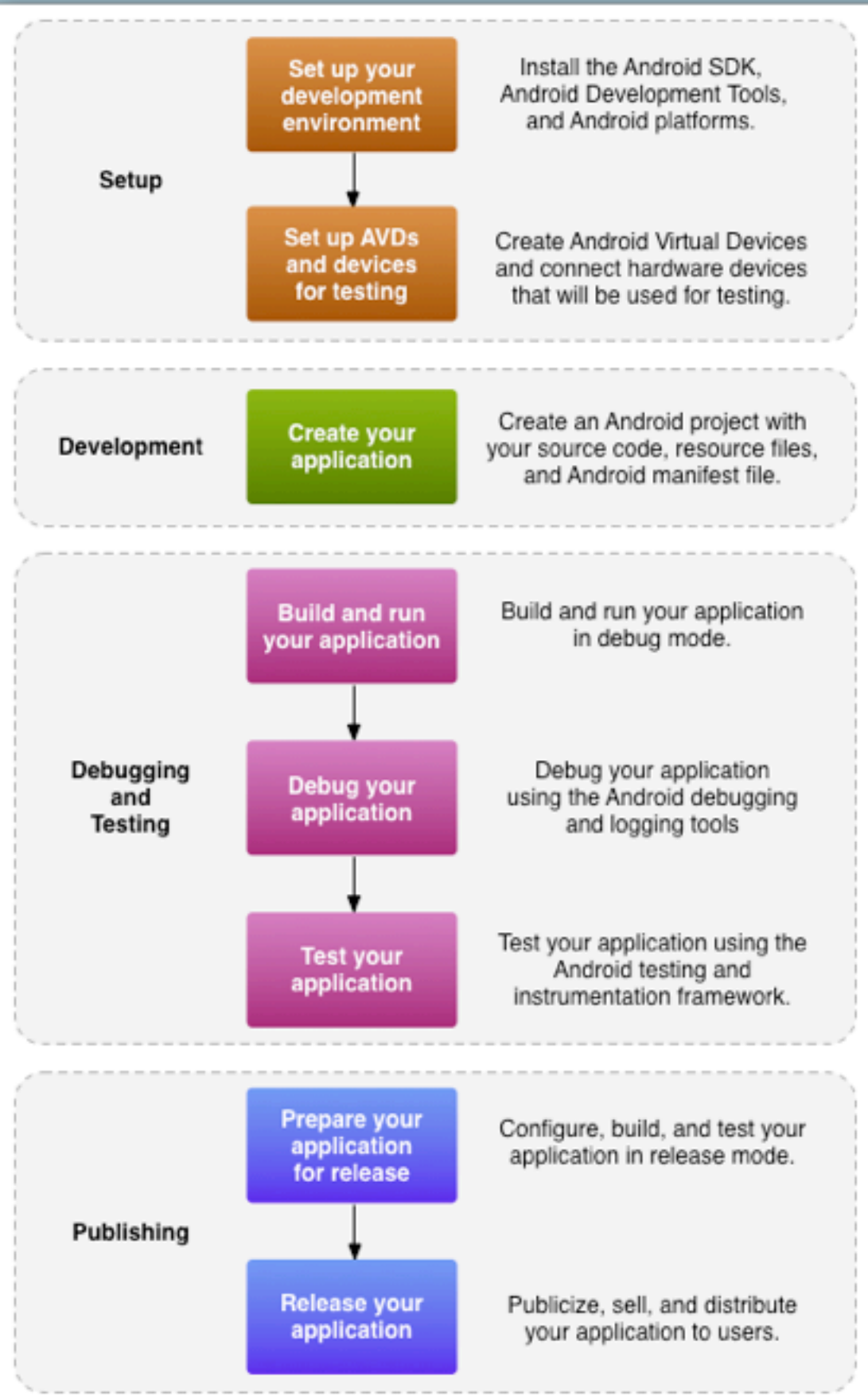
- Using the Support Library
- Creating a Fragment
- Building a Flexible UI
- Communicating with Other Fragments

**Left Navigation:**

- Getting Started
- Building Your First App
- Managing the Activity Lifecycle
- Supporting Different Devices
- Building a Dynamic UI with Fragments
- Saving Data
- Interacting with Other Apps
- Sharing Content
- Building Apps with Multimedia
- Building Apps with Graphics & Animation
- Building Apps with Connectivity & the Cloud
- Building Apps with User Info & Location
- Best Practices for User Experience & UI
- Best Practices for Performance
- Best Practices for Security & Privacy
- Using Google Play to Distribute & Monetize

<http://developer.android.com/training/index.html>





- SDK (Software development kit)
- AVD (Android Virtual Device)
- .apk (Android application package file)
- “platform”: Gingerbread v2.3.6 / v2.3.4 is our target
- ADT (Android Development Tools) Eclipse plug-ins plus command line tools

<http://developer.android.com/guide/index.html>

# Getting the Android DK

- Download and unpack the appropriate “Android SDK”
  - “Download for other platforms”
  - “SDK Tools Only”

## SDK Tools Only

Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android-sdk_r21-windows.zip</a>	99093893 bytes	7311452823470365f7975a545f8a2be4
	<a href="#">installer_r21-windows.exe</a> (Recommended)	77523031 bytes	29ca8cb8f0bc8db627fa2adc2139a3cc
Mac OS X	<a href="#">android-sdk_r21-macosx.zip</a>	65792626 bytes	67e46adca90dd18d7291443f6c15d6af
Linux	<a href="#">android-sdk_r21-linux.tgz</a>	91378351 bytes	7f8d73b629f808cdcfc9f9900bbd7580

<http://developer.android.com/sdk/index.html>

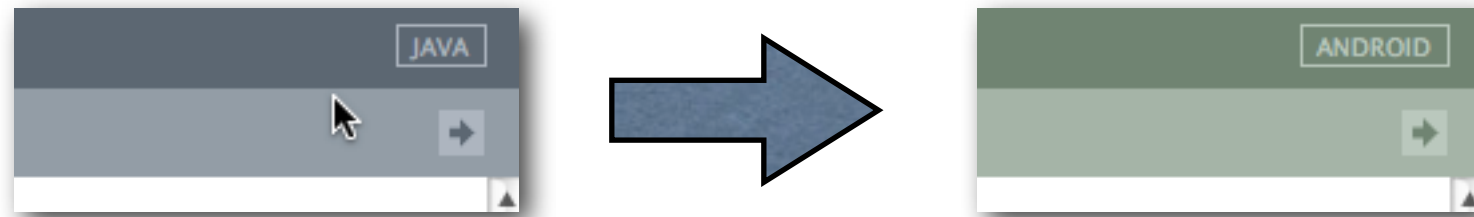
# Making the phone work

- Turn on developer mode
  - “home”->“menu”->“settings”->“applications”->“Development”
    - “USB debugging” on
    - “Stay awake” on
    - “Allow mock locations” on
  - Dial **\*##\*#CHECKIN#\*##\***
    - to update phone software



<http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930>

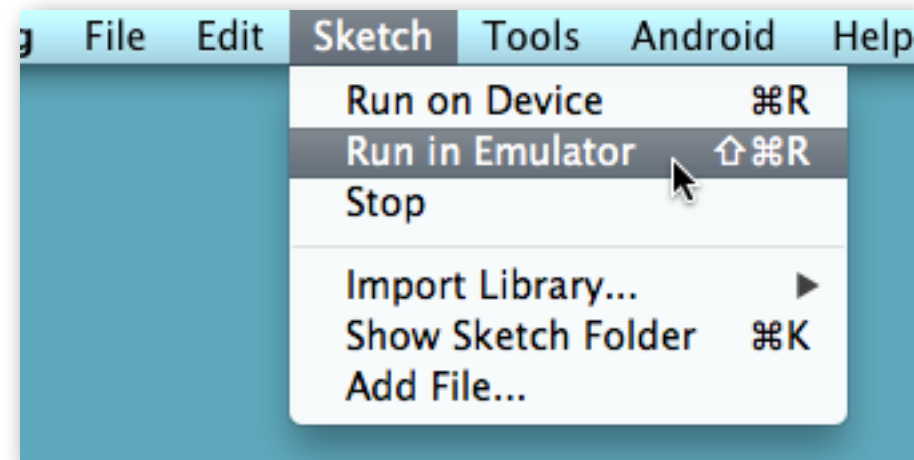
# Switch to Android Mode



- You will be asked where you put the SDK on your hard drive



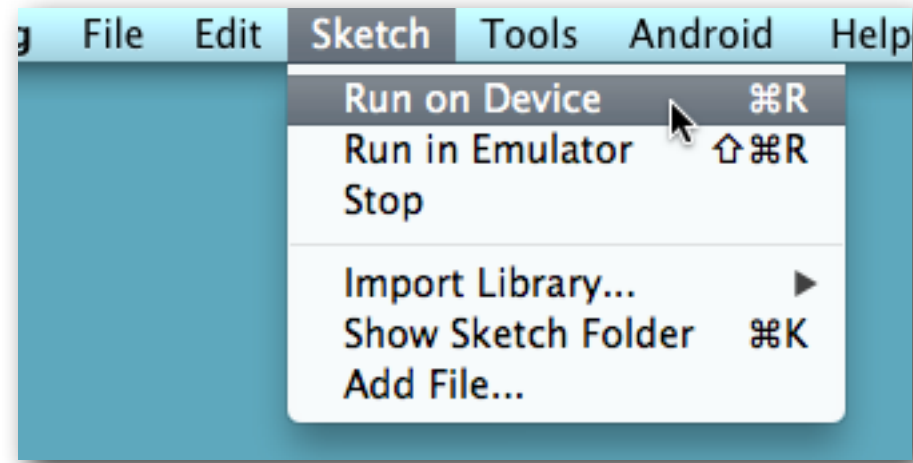
# Run in the emulator



- What is the emulator?
- Make sure you set your sketch to the right resolution
  - 480x800?
- Demo



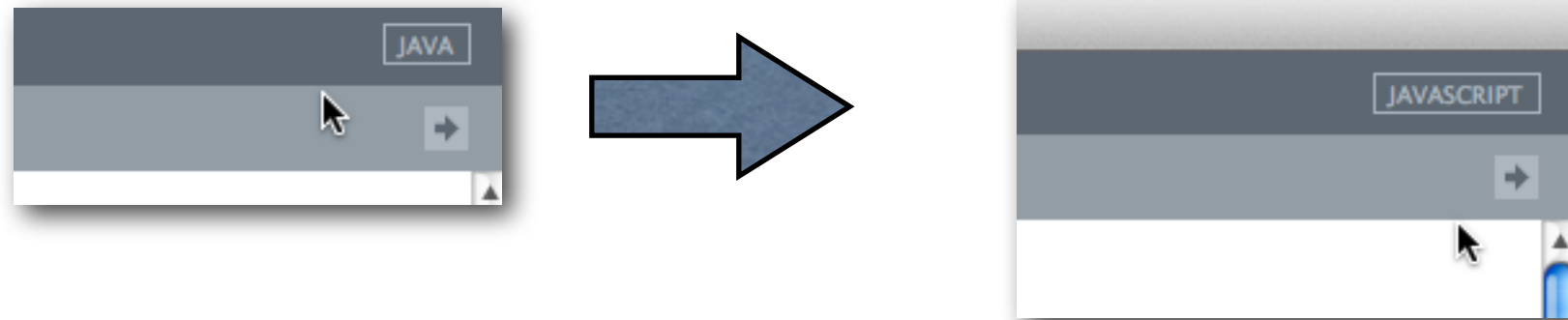
# Run on a real phone



- What is the emulator?
- Make sure you set your sketch to the right resolution
  - 480x800?
- Demo

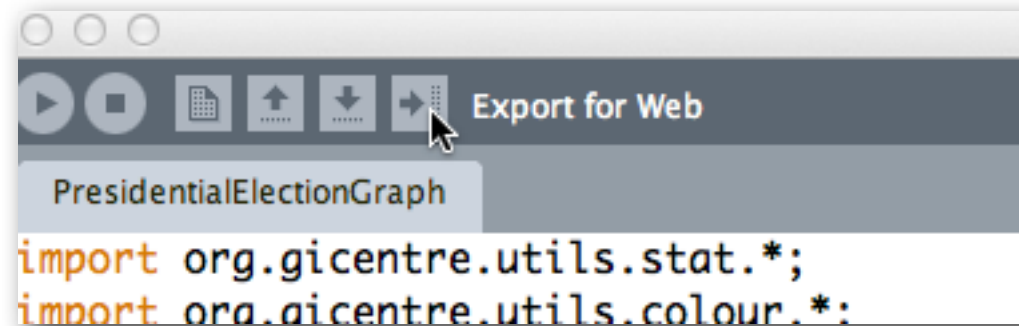


# Run on a website

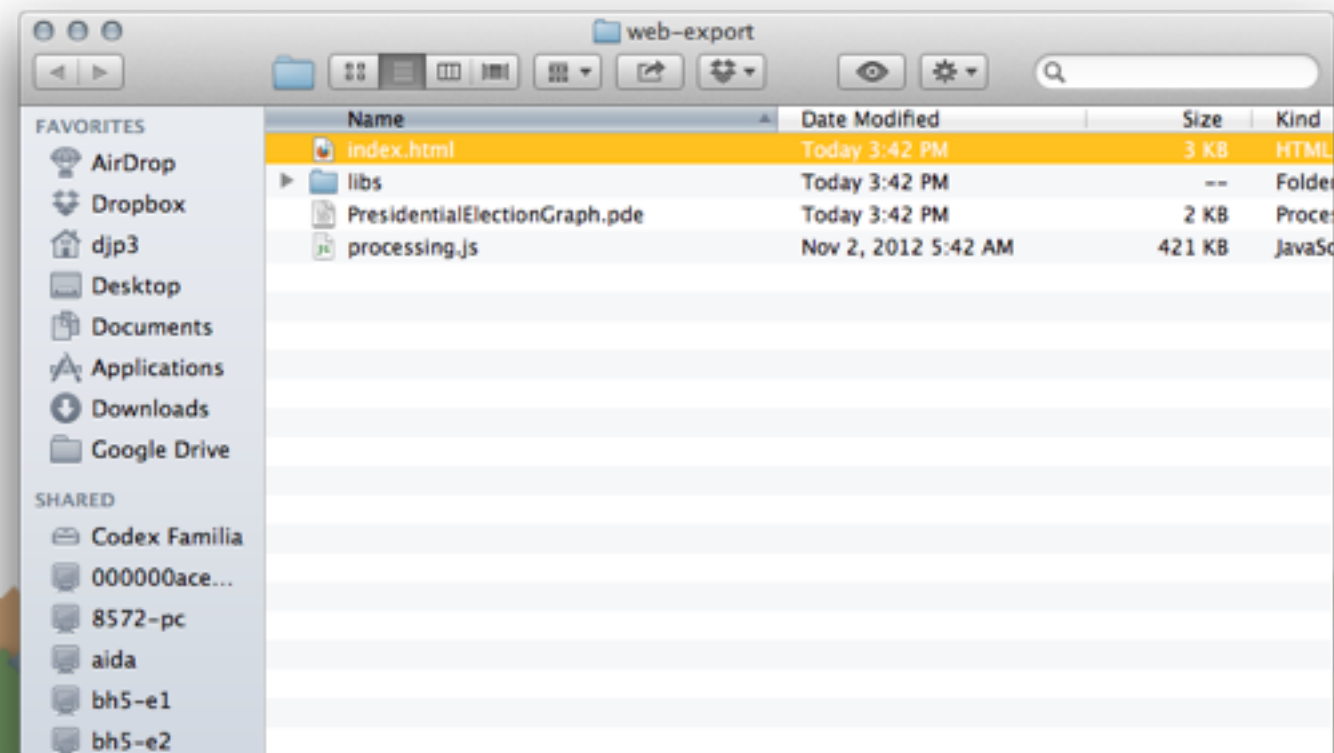




# Run on a web site

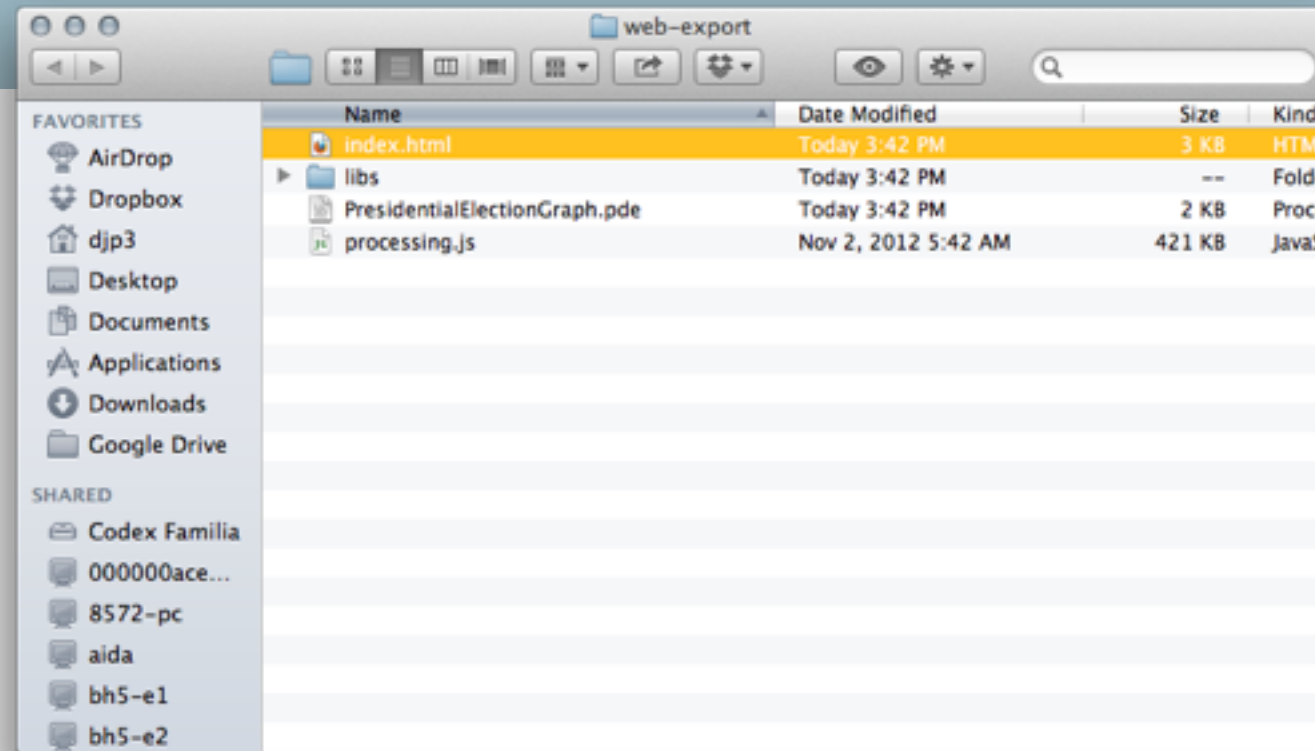


- “Export for Web”
- A file dialog will open showing you where the website is





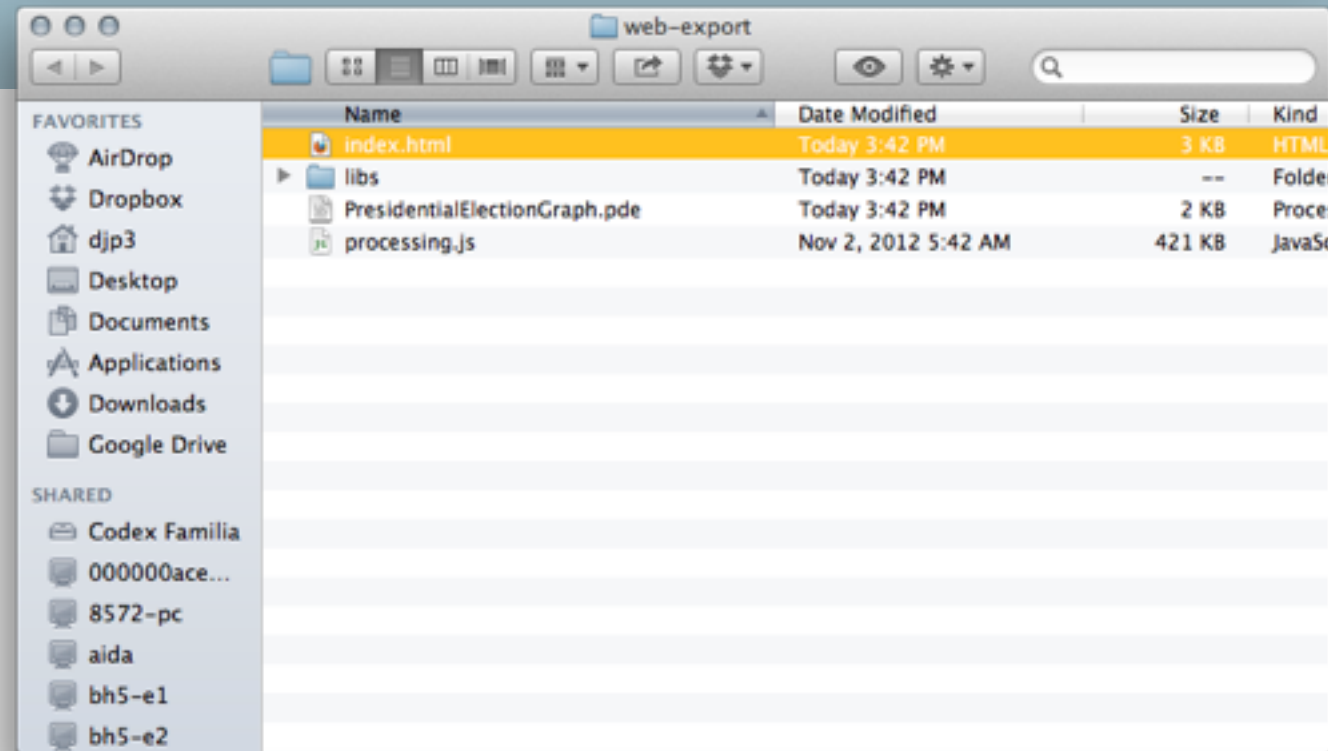
# Run on a web site



- You can open “index.html” in your browser from your hard drive.
- You might need to find the right browser because it uses Java (Chrome doesn't work for me, but Firefox does)



# Run on a web site



- To run it from a web site you need:
  - A server to put the files on
  - Move the files there
  - Then point your browser to the server
  - Demo



# To embed

- To put the sketch on your own website, just pull out the pieces you need from the exported version
- “View source” -> HTML (don't forget the external files)

```
</style>
<!--[if lt IE 9]>
  <script type="text/javascript">alert("Your browser does not support the canvas tag.");</script>
</endif-->
<script src="processing.js" type="text/javascript"></script>
<script type="text/javascript">
// convenience function to get the id attribute of generated sketch html element
function getProcessingSketchId () { return 'thanksgiving'; }
</script>

</head>
<body>
  <div id="content">
    <div>
      <canvas id="thanksgiving" data-processing-sources="thanksgiving.pde"
        width="750" height="500">
        <p>Your browser does not support the canvas tag.</p>
        <!-- Note: you can put any alternative content here. -->
      </canvas>
      <noscript>
        <p>JavaScript is required to view the contents of this page.</p>
      </noscript>
    </div>
    <h1>thanksgiving</h1>
    <p id="description"></p>
    <p id="sources">Source code: <a href="thanksgiving.pde">thanksgiving</a> </p>
    <p>
      Built with <a href="http://processing.org" title="Processing">Processing</a>
      and <a href="http://processingjs.org" title="Processing.js">Processing.js</a>
    </p>
  </div>
</body>
</html>
```

# To embed (alternate)

- Link the libraries and put the code in your HTML file

```
<script src="http://processing.org/javascript/MM_functions.js" type="text/javascript"></script>
<script src="http://processing.org/javascript/processing.js" type="text/javascript"></script>
<script src="http://processing.org/javascript/jquery-1.2.6.hardware.js" type="text/javascript"></script>
<script src="http://processing.org/javascript/slideshow.js" type="text/javascript"></script>
```

```
                                <div class="example"><script type="application/processing">
// The message to be displayed
String message = "How to Lie with Infographics";

PFont f;
// The radius of a circle
float r = 100;

void setup() {
  size(320, 320);
  f = createFont("Georgia",40,true);
  textFont(f);
  // The text must be centered!
  textAlign(CENTER);
  smooth();
}

void draw() {
  background(255);

  // Start in the center and draw the circle
  translate(width / 2, height / 2);
  noFill();
  stroke(0);

  // We must keep track of our position along the curve
  float arclength = 2*mouseX;

  // For every box
  for (int i = 0; i < message.length(); i++)
  {
```



To em

- Link

<script  
<script  
<script  
<script

```
centerOn(1);  
// The text must be centered!  
textAlign(CENTER);  
smooth();  
}  
  
void draw() {  
  background(255);  
  
  // Start in the center and draw the circle  
  translate(width / 2, height / 2);  
  noFill();  
  stroke(0);  
  
  // We must keep track of our position along the curve  
  float arclength = 2*mouseX;  
  
  // For every box  
  for (int i = 0; i < message.length(); i++)  
  {  
    // Instead of a constant width, we check the width of each character.  
    char currentChar = message.charAt(i);  
    float w = textWidth(currentChar);  
  
    // Each box is centered so we move half the width  
    arclength += w/2;  
    // Angle in radians is the arclength divided by the radius  
    // Starting on the left side of the circle by adding PI  
    float theta = PI + arclength / r;  
  
    pushMatrix();  
    // Polar to cartesian coordinate conversion  
    translate(r*cos(theta), r*sin(theta));  
    // Rotate the box  
    rotate(theta+PI/2); // rotation is offset by 90 degrees  
    // Display the character  
    fill(0);  
    text(currentChar,0,0);  
    popMatrix();  
    // Move halfway again  
    arclength += w/2;  
  }  
}
```

</script>

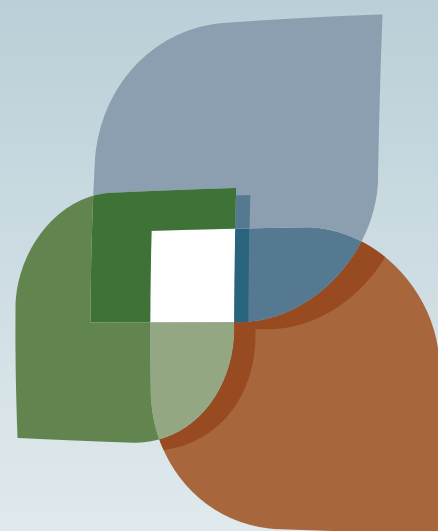
!important; "></canvas>

<canvas width="640" height="360" tabindex="0" id="\_\_processing

<p><strong>Inspiration for Exercise 2</strong></p>

</div>

pt>



L U C I

