PhoneGap, Processing

Phonegap

4. Oct 2016

INF5261

Mobile app development

- Native code
- Android
  - Java
- iPhone
  - Objective-C
  - Swift

- Hybrid

  (HTML & JavaScript)

  +

  (Native code)
- Native code wrapped around a webview
- HTML & JavaScript for app logic and UI
- Can compile to app

- HTML & JavaScript
  - Browser based
    - Normal webpage

Hybrid, “webview” based app development

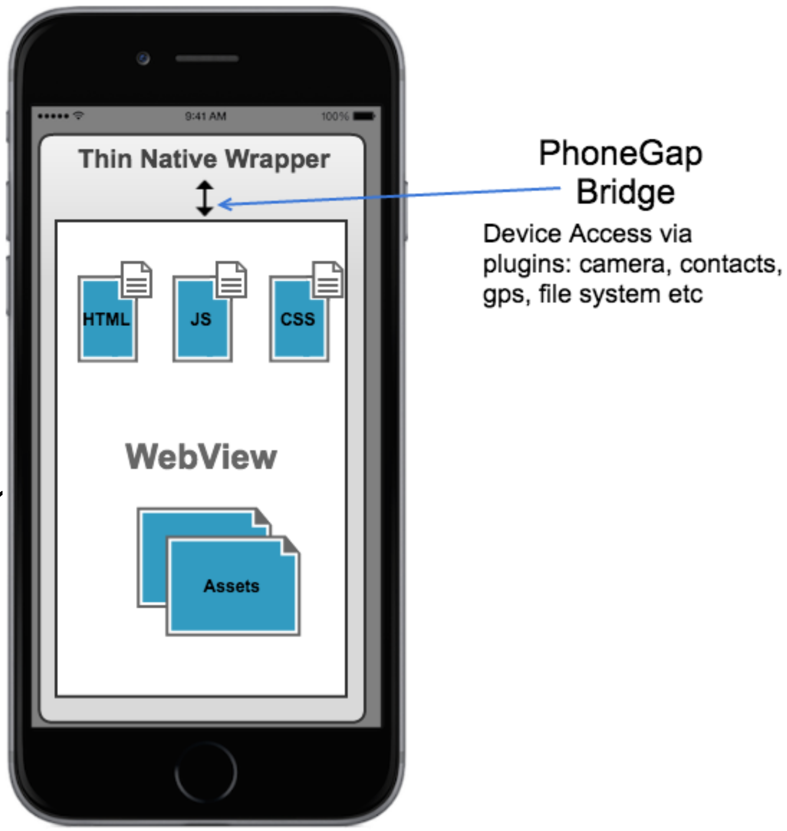
- Apps (can) use a lot of readymade code in the form of components:
  - Buttons, lists, media players, webview
- Webview is a browser that can be embedded inside an app to show web content.
  - Handles HTML and JavaScript, provides only a rendered view, no menus.
  - Can give JavaScript access to APIs that only native code normally can use, if you write your own JavaScript interface.
- But some native code must be written

Hybrid, the good news

- You don't have to do the native part

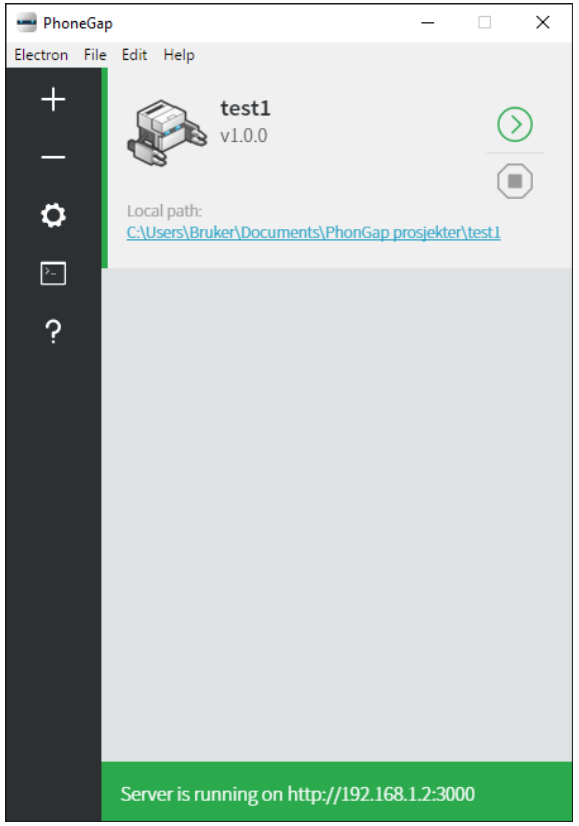
- Use PhoneGap/Cordova and concentrate on the web code
- Exposes a lot of native APIs to JavaScript
- 1500+ plugins to provide extra functionality
  - RFID/NFC, Barcode, Camera, Bluetooth, payment, maps

- http://phonegap.com/
- https://cordova.apache.org/
- Adobe owns PhoneGap.
- Cordova is the open source foundation for PhoneGap and other similar tools
- PhoneGap/Cordova is well established
  - PhoneGap gives approximately 8 120 000 hits on Google



PhoneGap

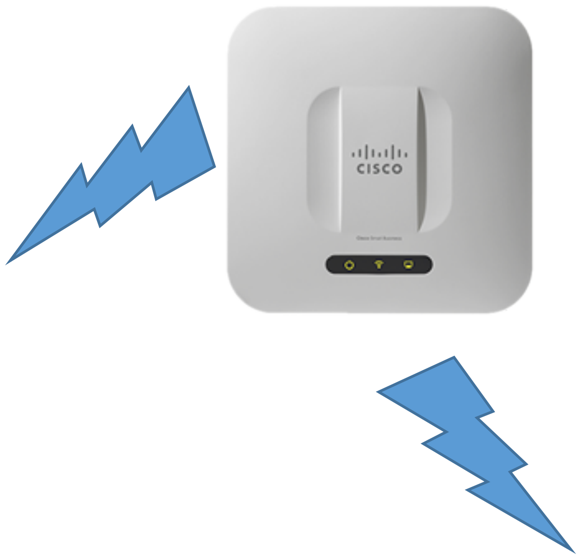
- Desktop app
  - Generate project
  - Server for mobile app



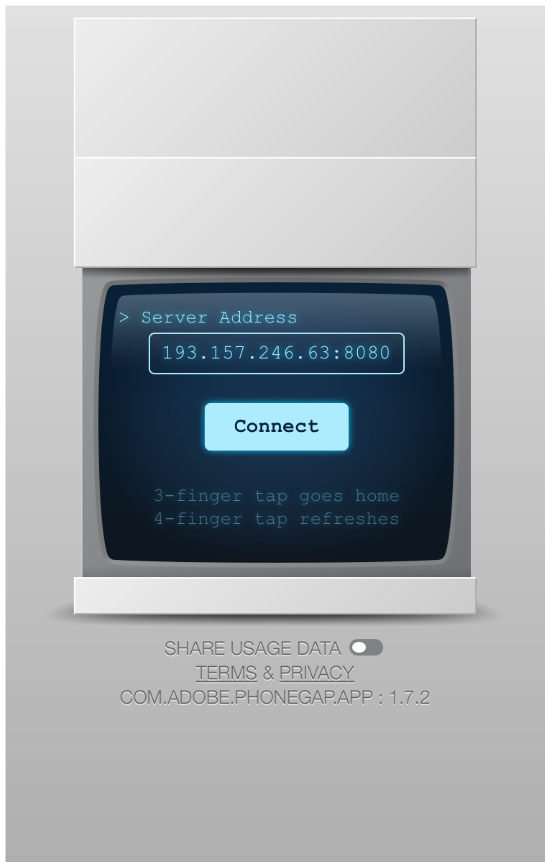
test1
v1.0.0

Local path:
C:\Users\Bruker\Documents\PhonGap prosjekter\test1

Server is running on http://192.168.1.2:3000



- Mobile app
  - Hosting mobile app for testing during development
  - Connecting to desktop server to download app



> Server Address
193.157.246.63:8080

3-finger tap goes home
4-finger tap refreshes

SHARE USAGE DATA
TERMS & PRIVACY
COM.ADOBE.PHONEGAP.APP : 1.7.2

- Desktop and mobile must be on same local network

Download & install

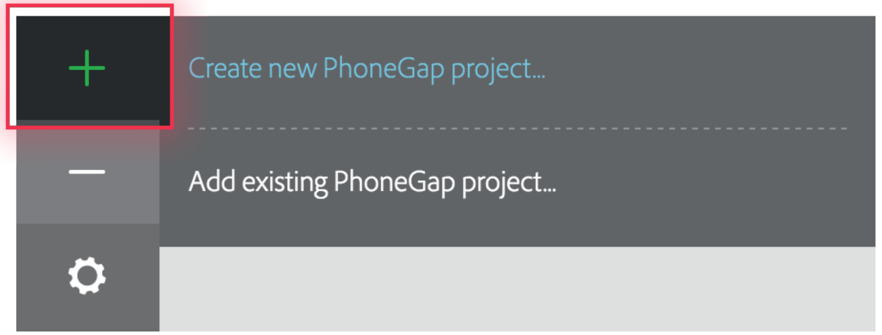
- Desktop
  - http://phonegap.com/products/
  - Scroll down and download desktop app for Mac og Windows
- Mobile
  - Go to App Store or Google Play
  - Install PhoneGap Developer by Adobe

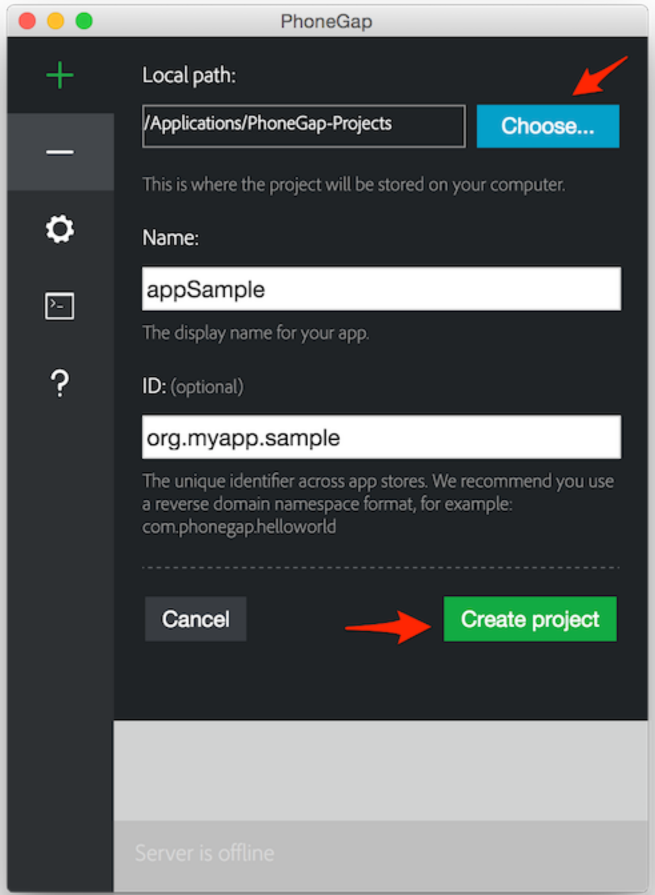
Create your new app

- Open the PhoneGap Desktop app
  - create a new project.

- Select a folder
- Write a project name
- Click green button

- http://phonegap.com/getstarted/


Create new PhoneGap project...
Add existing PhoneGap project...

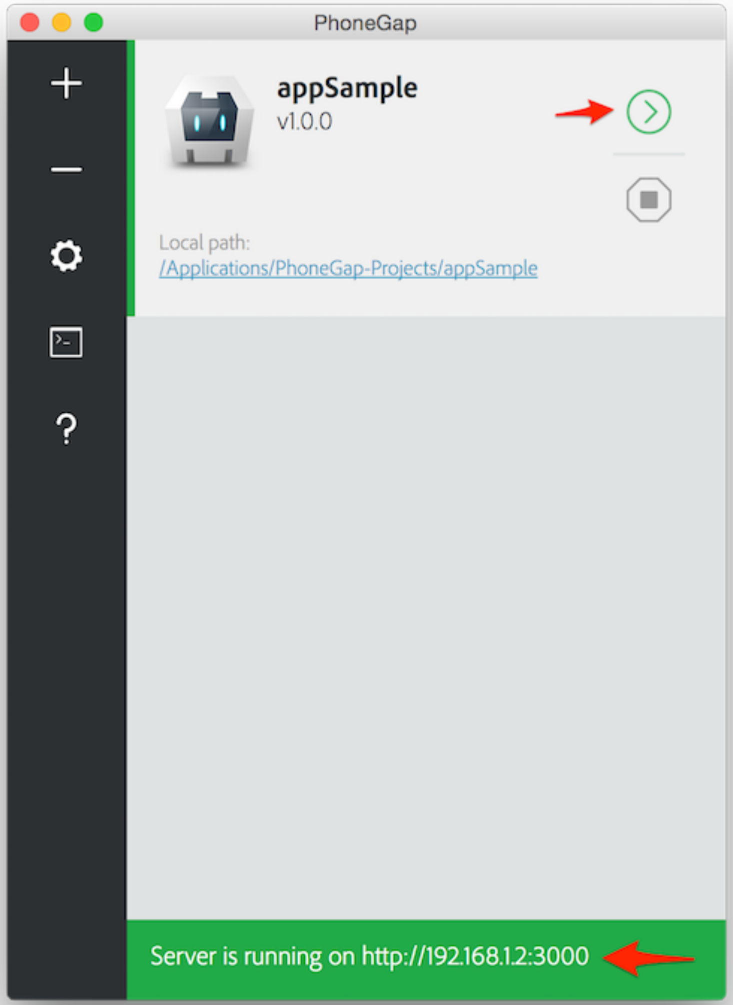


Local path:
/Applications/PhoneGap-Projects
This is where the project will be stored on your computer.
Name:
appSample
The display name for your app.
ID: (optional)
org.myapp.sample
The unique identifier across app stores. We recommend you use a reverse domain namespace format, for example: com.phonegap.helloworld

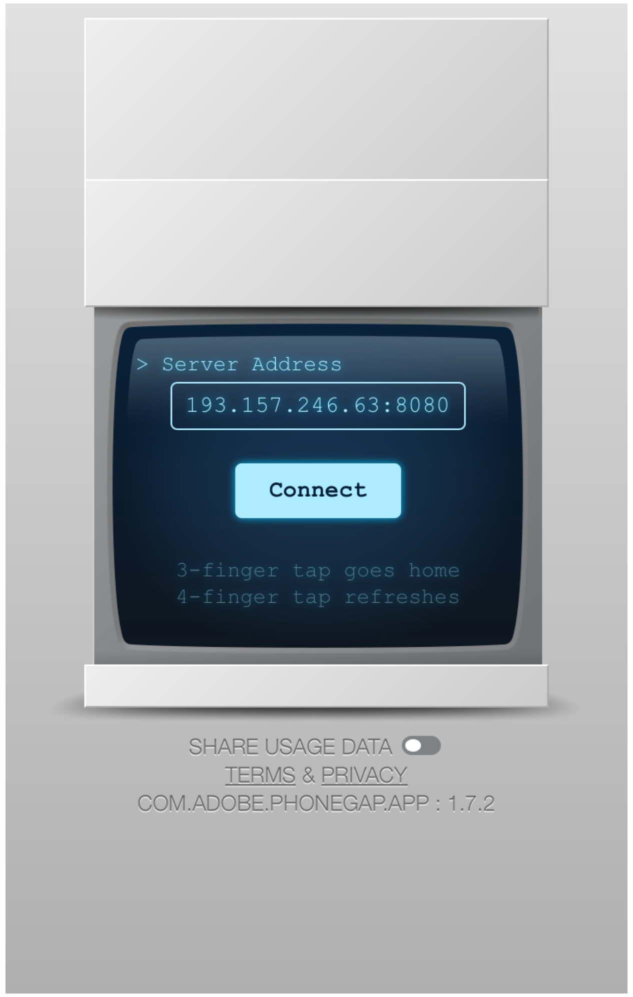
Server is offline

Preview your first app

- Start the server
- On the mobile:
  - Input the IP address and port number shown in the green field on the desktop server.

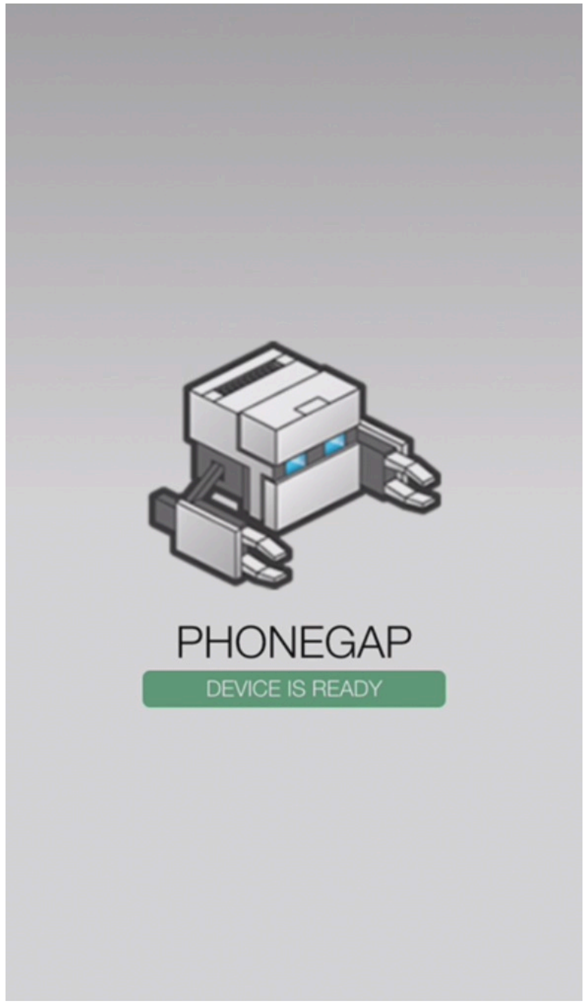


Server is running on http://192.168.1.2:3000



Preview your first app 2

- What it should look like



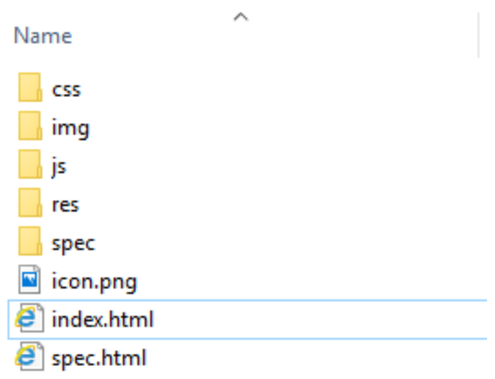
Use Chrome or Safari for preview

- Safari Responsive Design Mode
- Chrome DevTools' Device Mode

- They will not emulate all APIs
- But good for UI and logic

Where is the code?

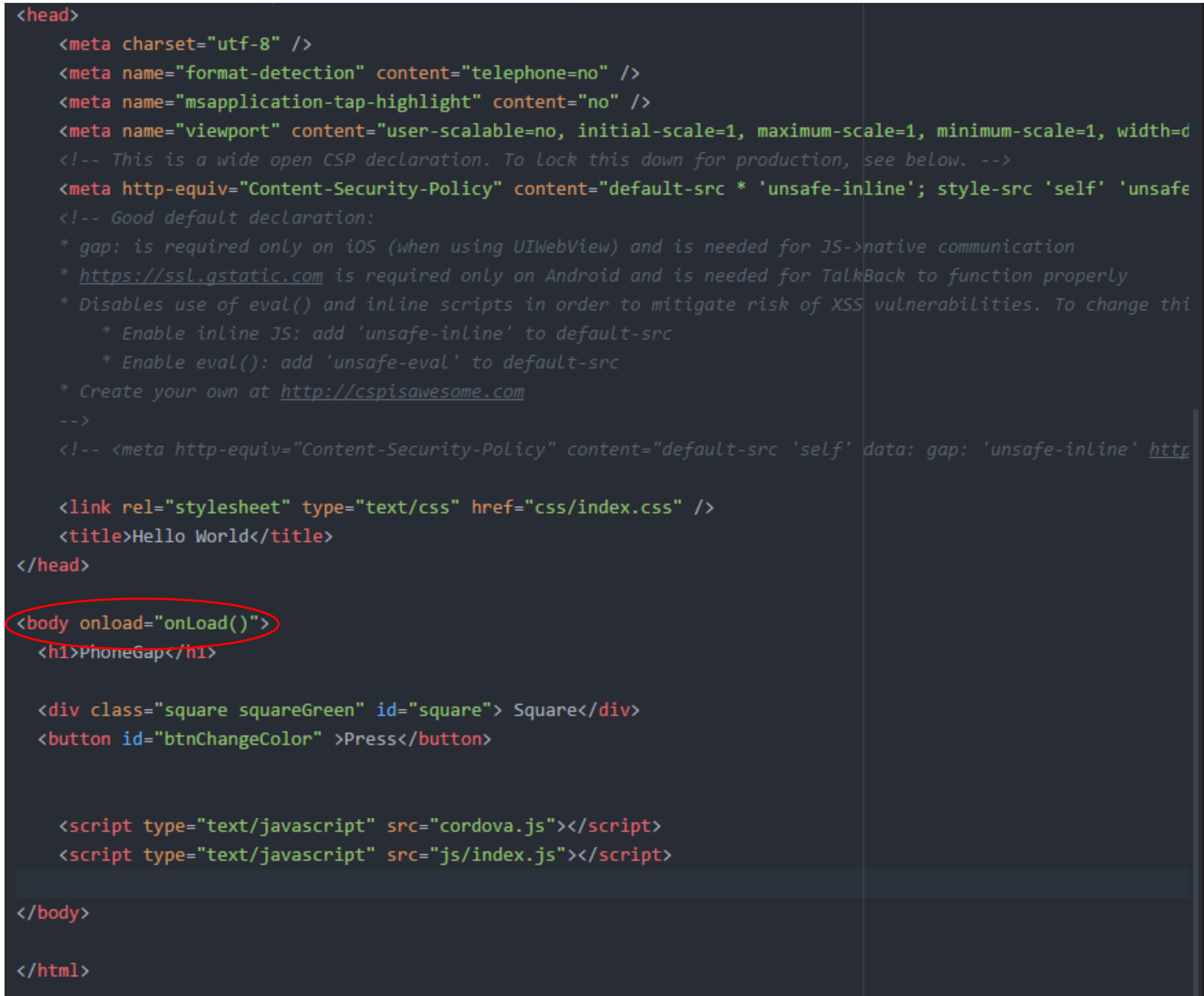
- In the project folder


Name
css
img
js
res
spec
icon.png
index.html
spec.html

Som code to start the app

```
function onLoad() {
   document.addEventListener("deviceready", onDeviceReady,
false);
}
function onDeviceReady() {
   // Now safe to use the Cordova API
}
```

- Deviceready event fires when cordova/phongap is loaded
  - Is “safe” to call cordova api from javascript



```
<head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="msapplication-tap-highlight" content="no" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=d
    <!-- This is a wide open CSP declaration. To lock this down for production, see below. -->
    <meta http-equiv="Content-Security-Policy" content="default-src * 'unsafe-inline'; style-src 'self' 'unsafe
    <!-- Good default declaration:
    * gap: is required only on iOS (when using UIWebView) and is needed for JS->native communication
    * https://ssl.gstatic.com is required only on Android and is needed for TalkBack to function properly
    * Disables use of eval() and inline scripts in order to mitigate risk of XSS vulnerabilities. To change thi
        * Enable inline JS: add 'unsafe-inline' to default-src
        * Enable eval(): add 'unsafe-eval' to default-src
    * Create your own at http://cspisawesome.com
    -->
    <!-- <meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap: 'unsafe-inline' http

    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <title>Hello World</title>
</head>

<body onload="onLoad()">
  <h1>PhoneGap</h1>

  <div class="square squareGreen" id="square"> Square</div>
  <button id="btnChangeColor" >Press</button>


    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript" src="js/index.js"></script>

</body>

</html>
```

Adding some tags and code

- Text
- Button
- CSS
- Javascript
  - jQuery
  - Angular
  - ….
- Plugin
  - Camera
  - Barcode??

Processing

- Developed for creative coding
- Lots of libraries
- Can compile to Android

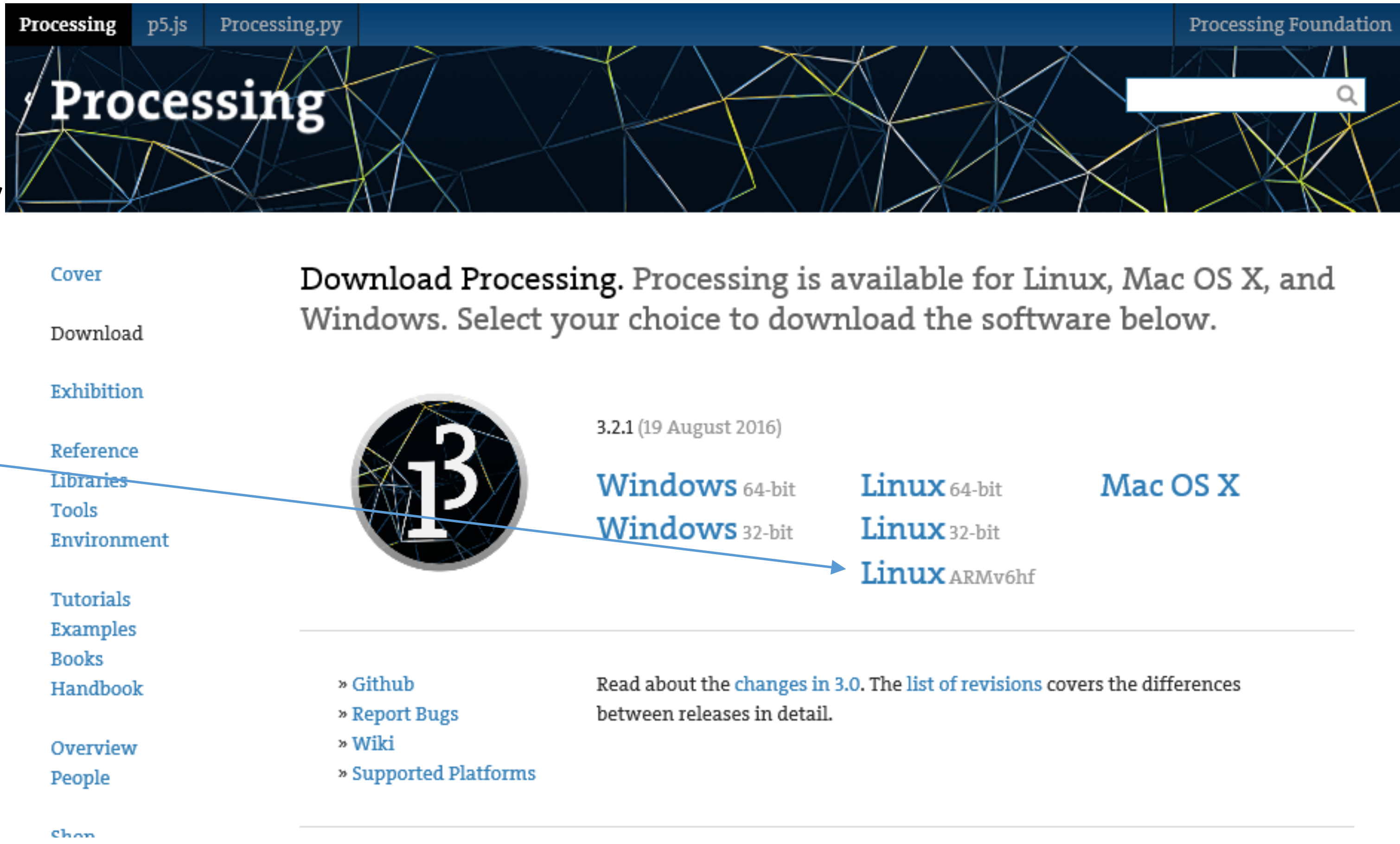
- Simple Java syntax
- Runs on:
  - Mac
  - Windows
  - Linux
  - ARM(RaspberryPi)
  - Android

Processing Visualization Design example:
https://vimeo.com/173760057

Processing

- Processing.org
- processing.org/download/

- Raspberry Pi



Processing | p5.js | Processing.py | Processing Foundation

Processing

Cover

Download

Exhibition

Reference
Libraries
Tools
Environment

Tutorials
Examples
Books
Handbook

Overview
People

Shop

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.

3.2.1 (19 August 2016)

Windows 64-bit | Linux 64-bit | Mac OS X
Windows 32-bit | Linux 32-bit
Linux ARMv6hf

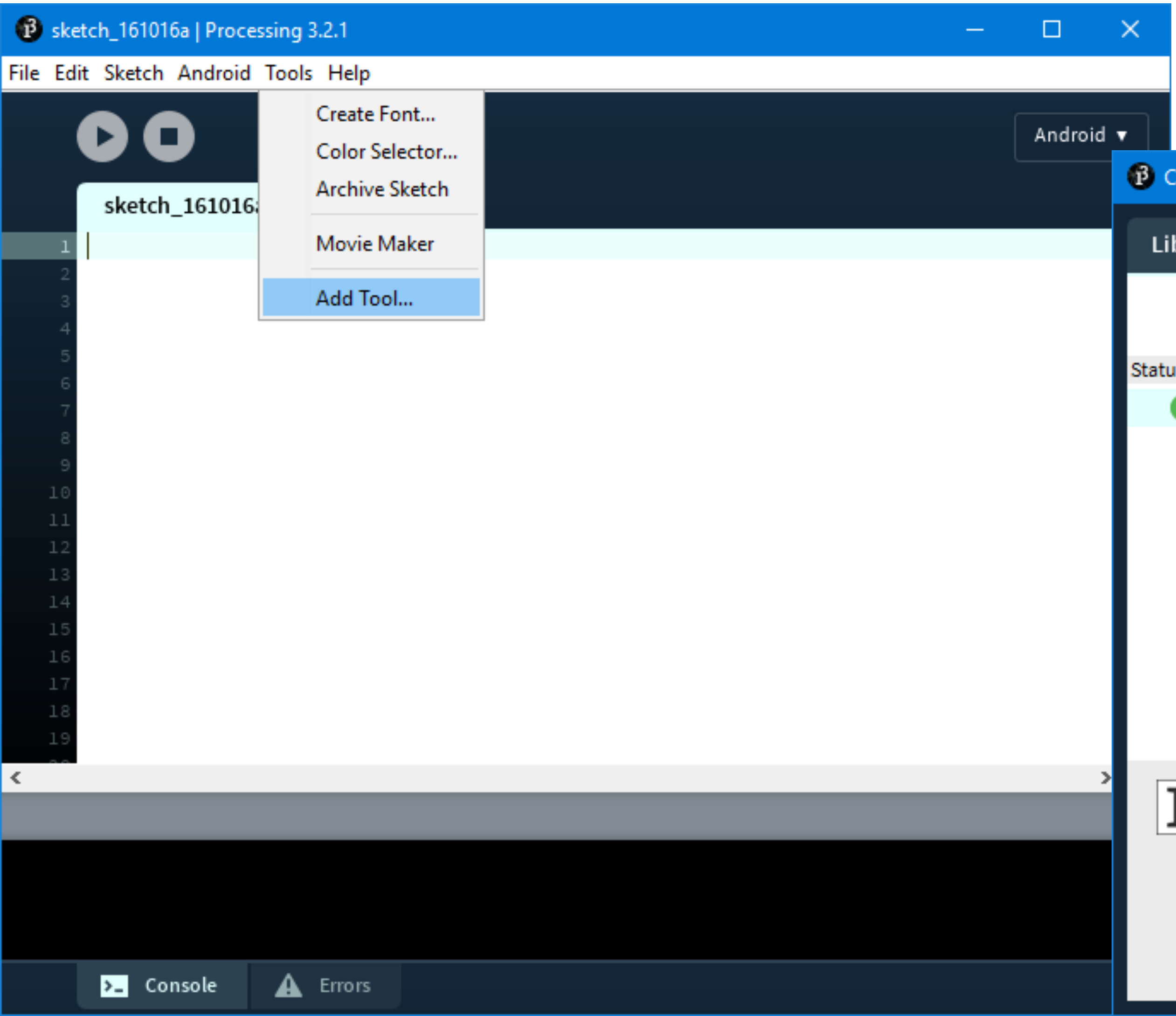
» Github
» Report Bugs
» Wiki
» Supported Platforms

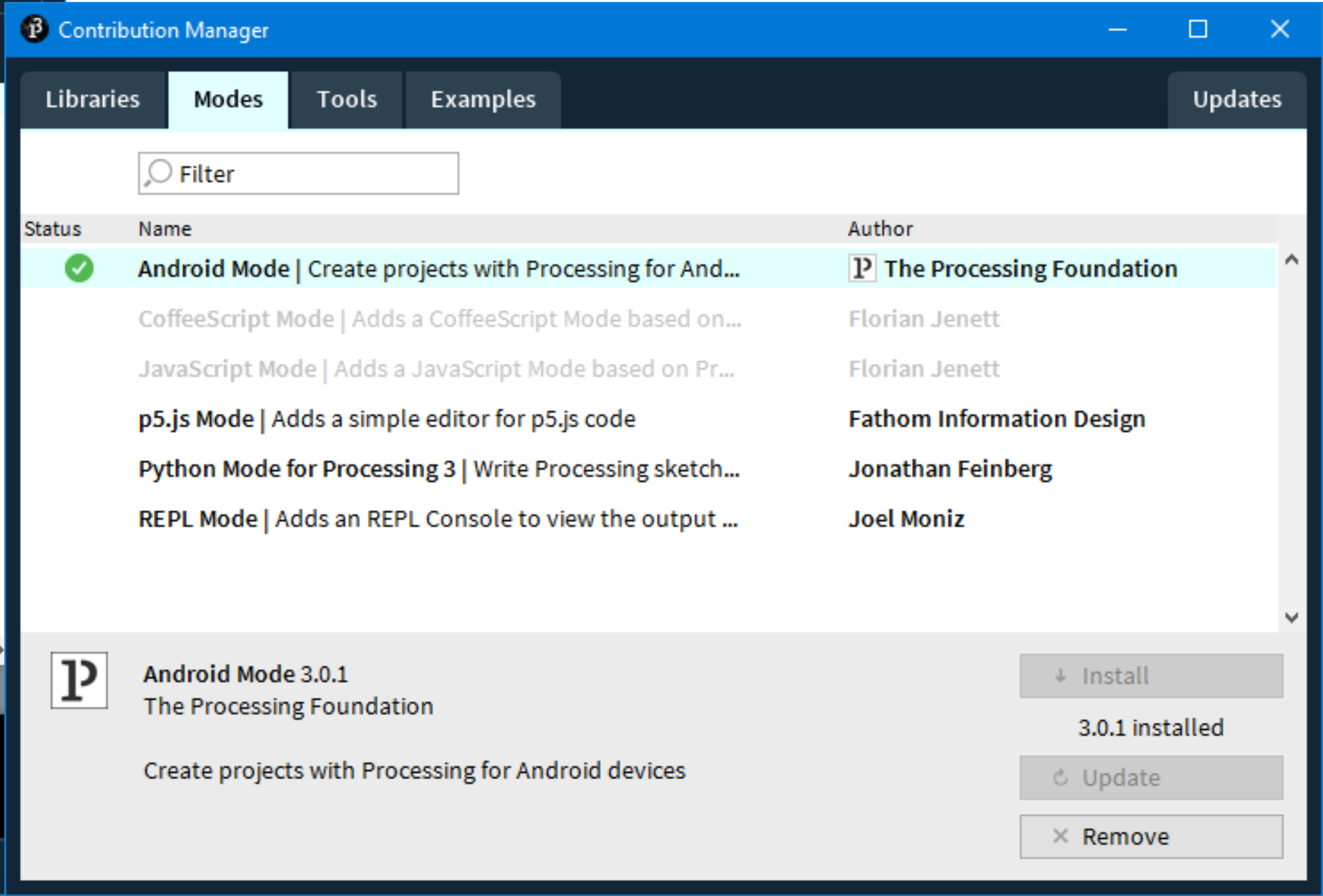
Read about the changes in 3.0. The list of revisions covers the differences between releases in detail.

Processing: installation

- Download
- Unzip
- Place folder where you like
- Make shortcut to desktop

Processing: editor and android mode





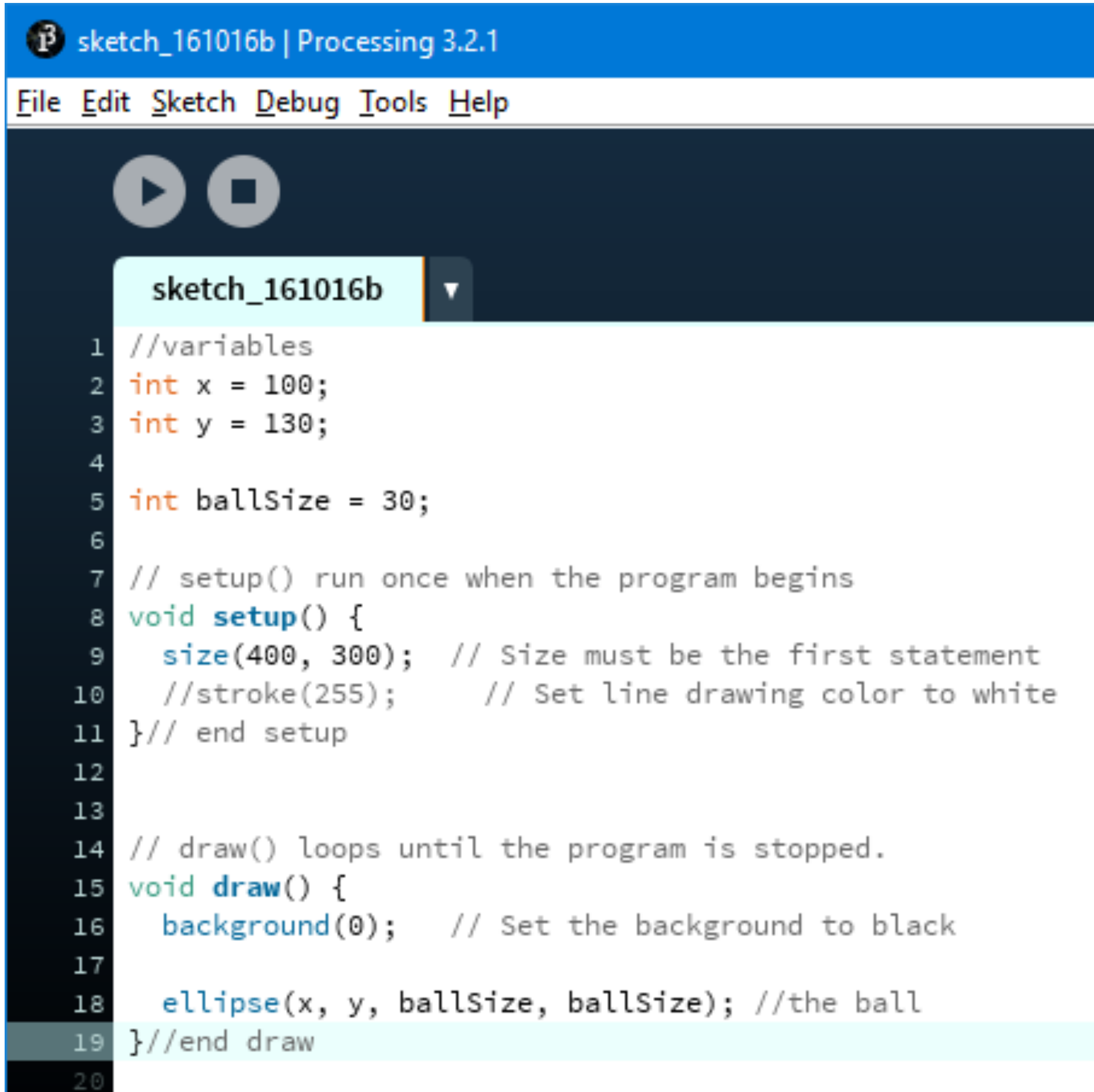
Status	Name	Author
✓	Android Mode | Create projects with Processing for And...	The Processing Foundation
	CoffeeScript Mode | Adds a CoffeeScript Mode based on...	Florian Jenett
	JavaScript Mode | Adds a JavaScript Mode based on Pr...	Florian Jenett
	p5.js Mode | Adds a simple editor for p5.js code	Fathom Information Design
	Python Mode for Processing 3 | Write Processing sketch...	Jonathan Feinberg
	REPL Mode | Adds an REPL Console to view the output ...	Joel Moniz

Android Mode 3.0.1
The Processing Foundation

Create projects with Processing for Android devices

Processing: first code

- Try this code
- Setup and draw

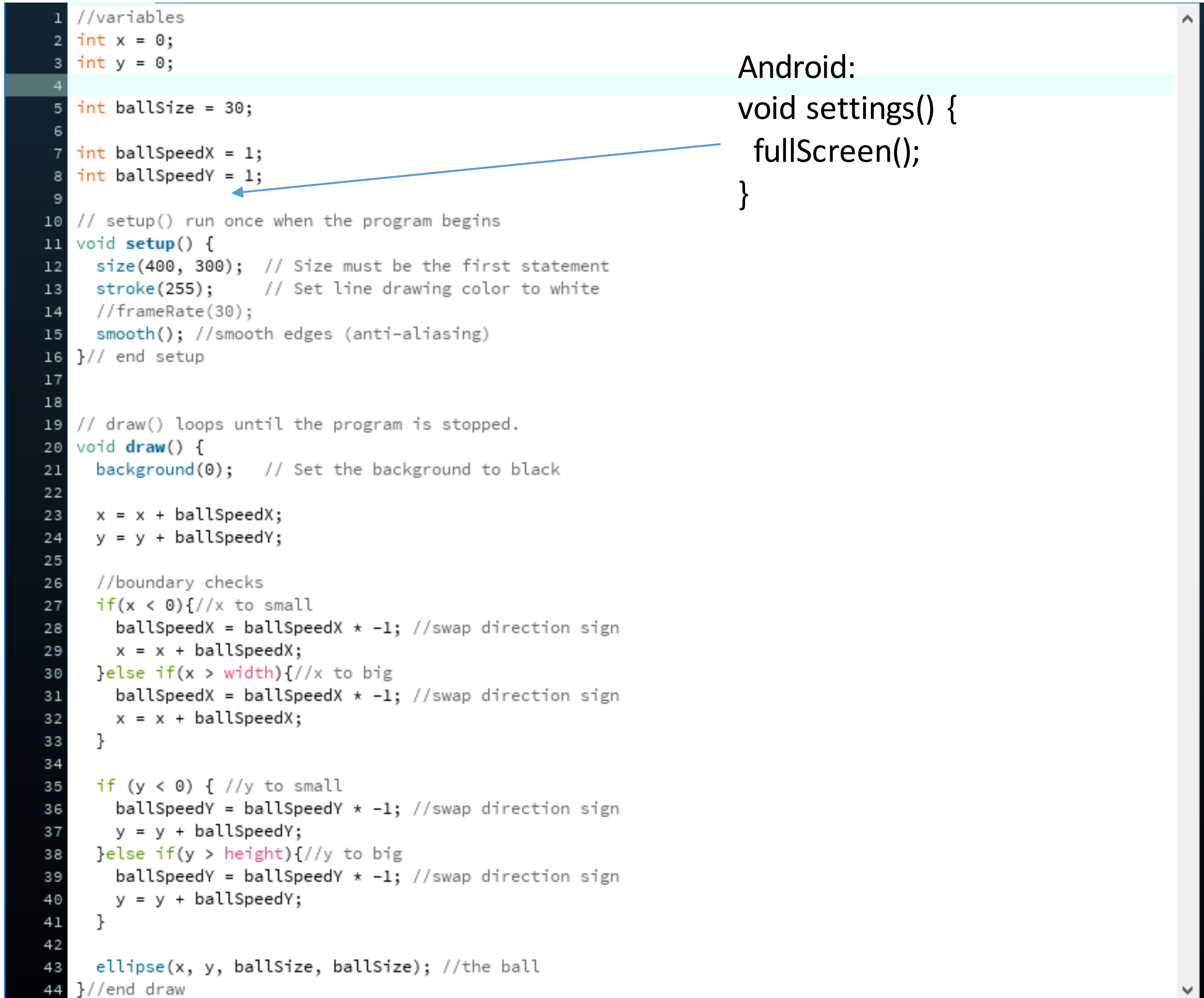

sketch_161016b | Processing 3.2.1
File Edit Sketch Debug Tools Help
sketch_161016b

```
//variables
int x = 100;
int y = 130;

int ballSize = 30;

// setup() run once when the program begins
void setup() {
  size(400, 300);  // Size must be the first statement
  //stroke(255);     // Set line drawing color to white
}// end setup


// draw() loops until the program is stopped.
void draw() {
  background(0);   // Set the background to black

  ellipse(x, y, ballSize, ballSize); //the ball
}//end draw
```

Processing:

- More code
- What is it?
- Bouncing ball

- Can you improve the code?
- Try it on your phone
  - fullScreen() insted of size() for Android

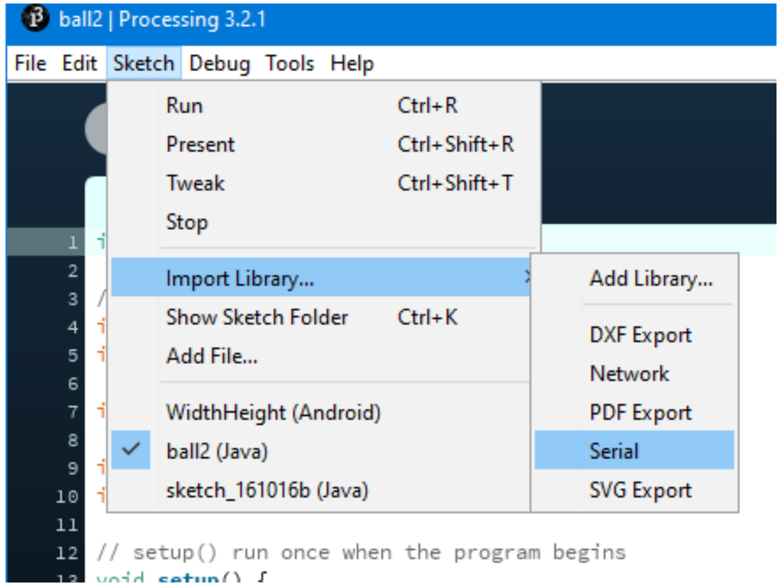


```
//variables
int x = 0;
int y = 0;

int ballSize = 30;

int ballSpeedX = 1;
int ballSpeedY = 1;

// setup() run once when the program begins
void setup() {
  size(400, 300);  // Size must be the first statement
  stroke(255);     // Set line drawing color to white
  //frameRate(30);
  smooth(); //smooth edges (anti-aliasing)
}// end setup


// draw() loops until the program is stopped.
void draw() {
  background(0);   // Set the background to black

  x = x + ballSpeedX;
  y = y + ballSpeedY;

  //boundary checks
  if(x < 0){//x to small
    ballSpeedX = ballSpeedX * -1; //swap direction sign
    x = x + ballSpeedX;
  }else if(x > width){//x to big
    ballSpeedX = ballSpeedX * -1; //swap direction sign
    x = x + ballSpeedX;
  }

  if (y < 0) { //y to small
    ballSpeedY = ballSpeedY * -1; //swap direction sign
    y = y + ballSpeedY;
  }else if(y > height){//y to big
    ballSpeedY = ballSpeedY * -1; //swap direction sign
    y = y + ballSpeedY;
  }

  ellipse(x, y, ballSize, ballSize); //the ball
}//end draw
```

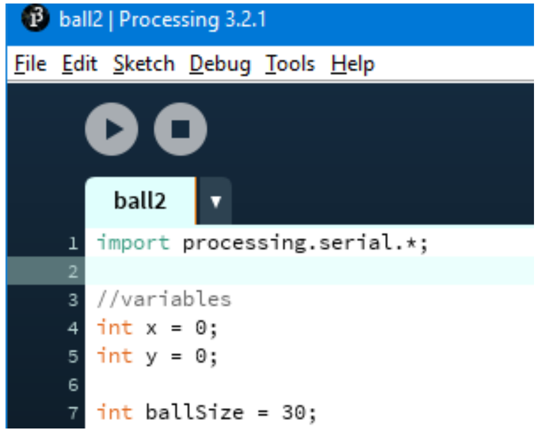
Android:
```
void settings() {
  fullScreen();
}
```

Processing: serial communication

- Can receive and handle data from Arduino
- Import serial library



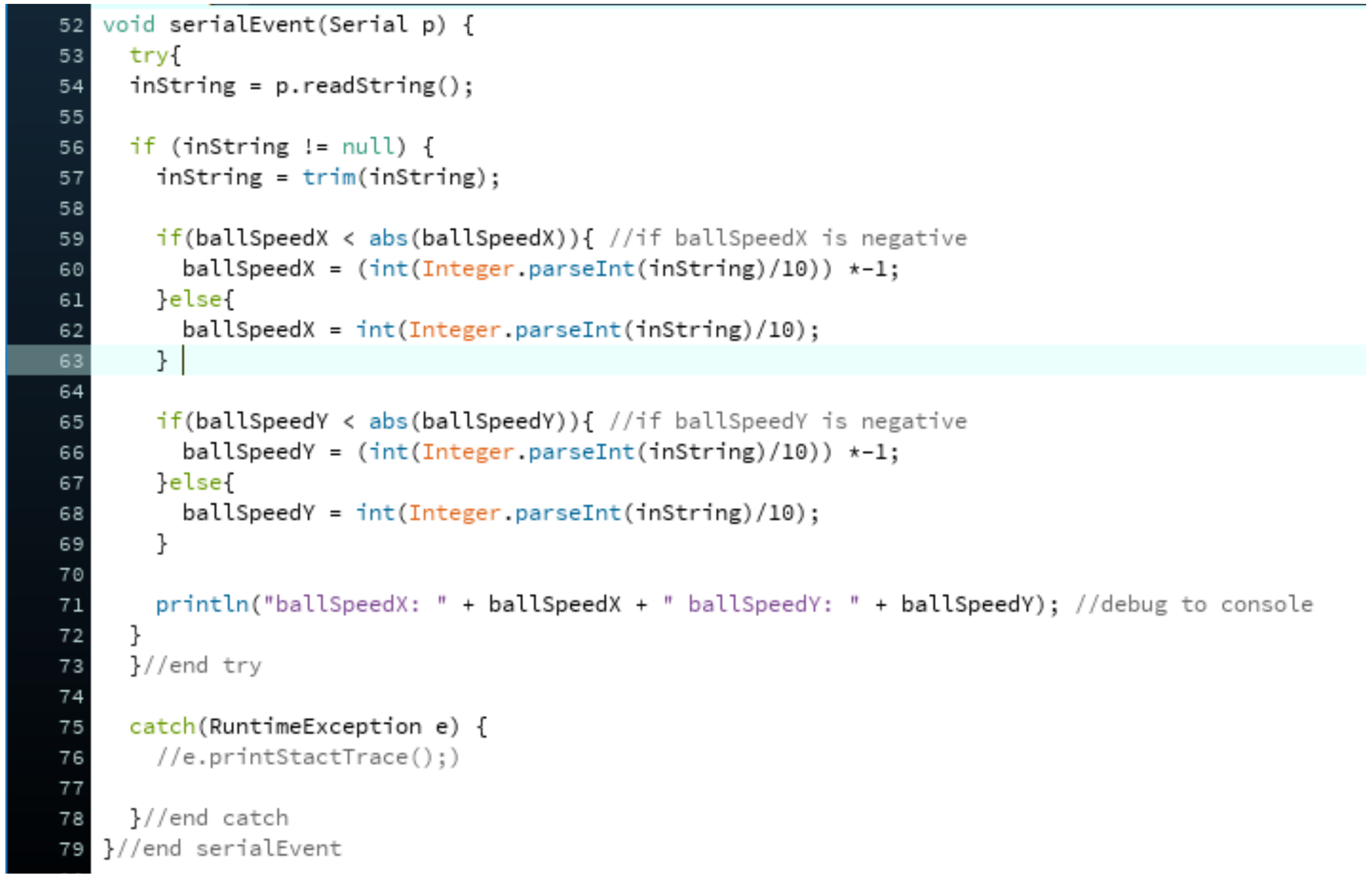
```
// setup() run once when the program begins
```



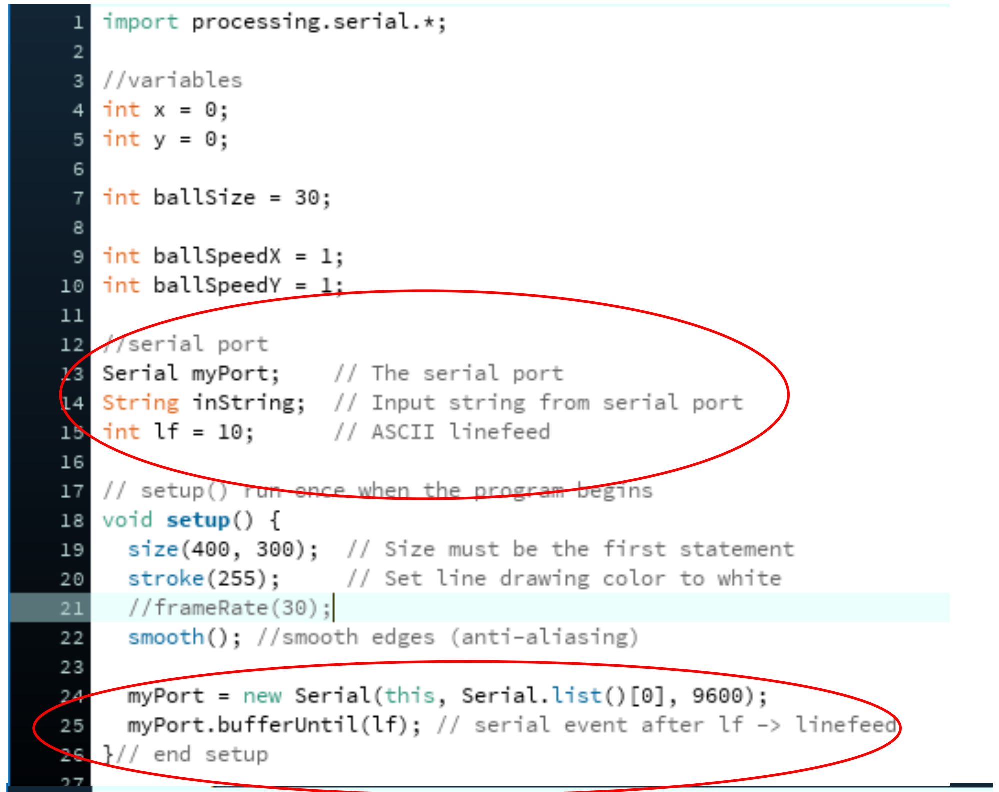
```
import processing.serial.*;

//variables
int x = 0;
int y = 0;

int ballSize = 30;
```

Processing: serial

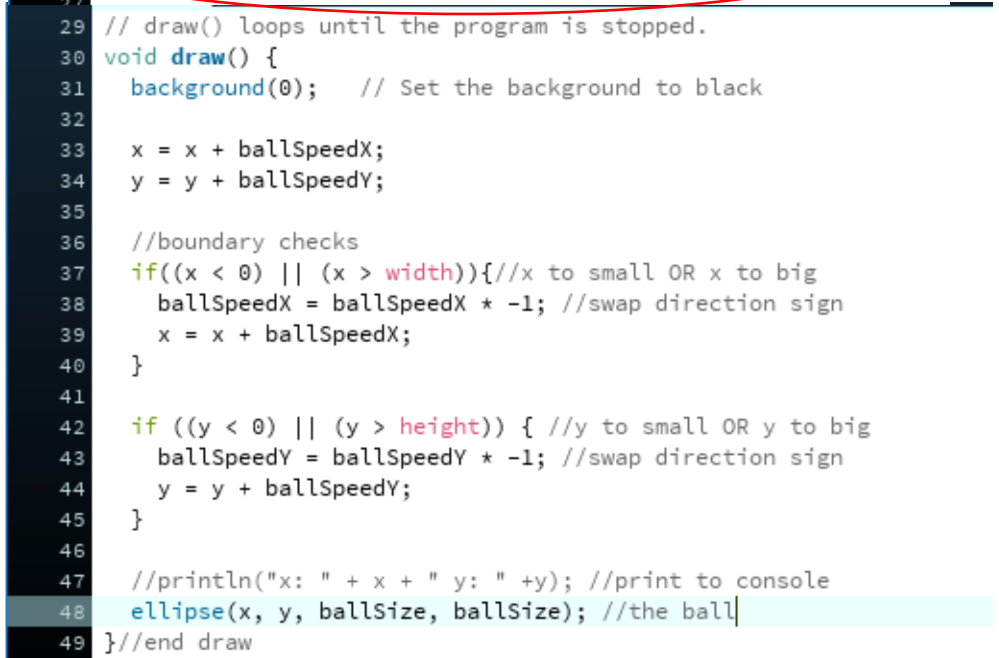
- Add code to receive and handle data



```
void serialEvent(Serial p) {
  try{
  inString = p.readString();

  if (inString != null) {
    inString = trim(inString);

    if(ballSpeedX < abs(ballSpeedX)){ //if ballSpeedX is negative
      ballSpeedX = (int(Integer.parseInt(inString)/10)) *-1;
    }else{
      ballSpeedX = int(Integer.parseInt(inString)/10);
    }

    if(ballSpeedY < abs(ballSpeedY)){ //if ballSpeedY is negative
      ballSpeedY = (int(Integer.parseInt(inString)/10)) *-1;
    }else{
      ballSpeedY = int(Integer.parseInt(inString)/10);
    }

    println("ballSpeedX: " + ballSpeedX + " ballSpeedY: " + ballSpeedY); //debug to console
  }
  }//end try

  catch(RuntimeException e) {
    //e.printStactTrace();)

  }//end catch
}//end serialEvent
```



```
import processing.serial.*;

//variables
int x = 0;
int y = 0;

int ballSize = 30;

int ballSpeedX = 1;
int ballSpeedY = 1;

//serial port
Serial myPort;    // The serial port
String inString;  // Input string from serial port
int lf = 10;      // ASCII linefeed

// setup() run once when the program begins
void setup() {
  size(400, 300);  // Size must be the first statement
  stroke(255);     // Set line drawing color to white
  //frameRate(30);
  smooth(); //smooth edges (anti-aliasing)

  myPort = new Serial(this, Serial.list()[0], 9600);
  myPort.bufferUntil(lf); // serial event after lf -> linefeed
}// end setup
```

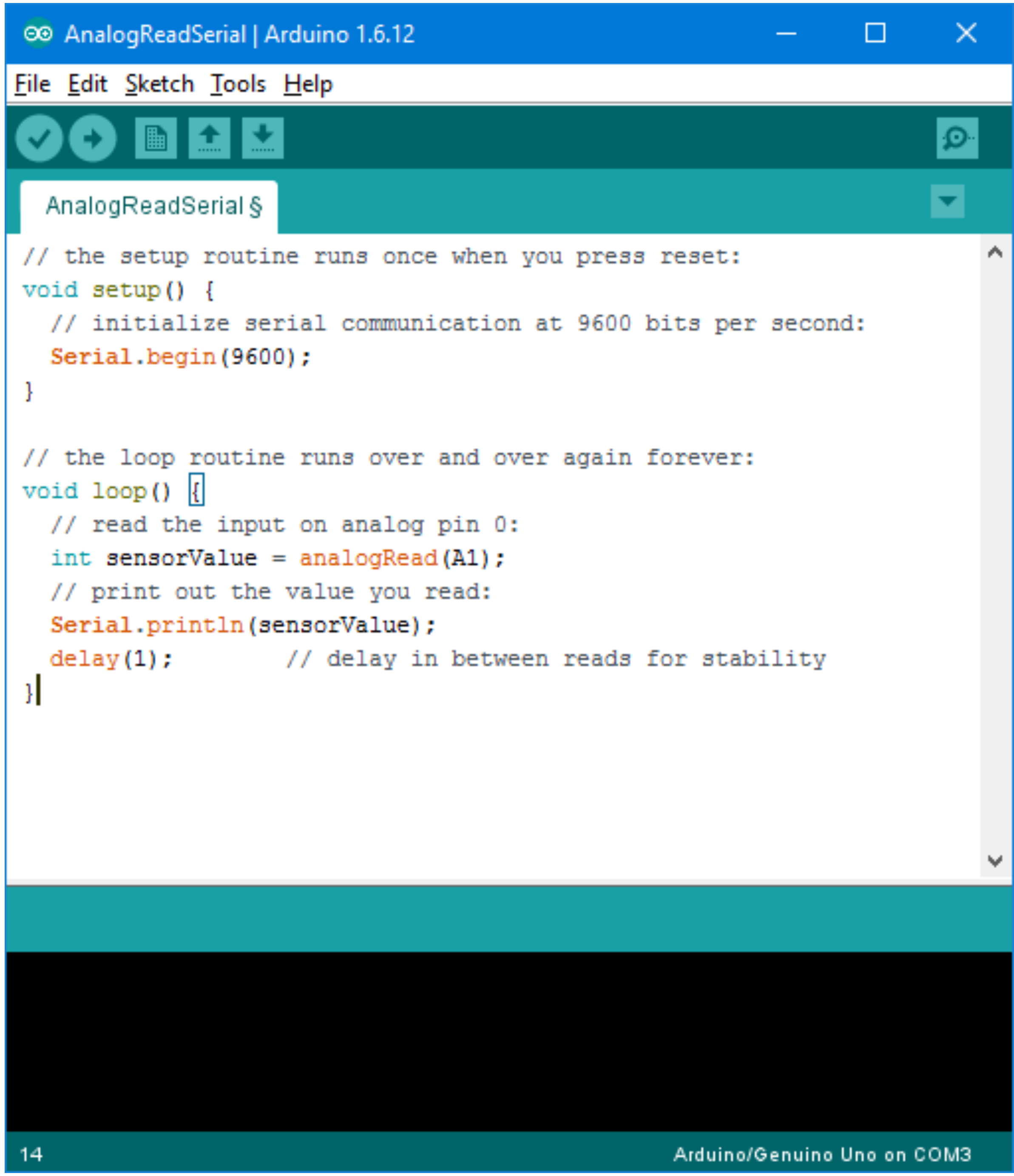


```
// draw() loops until the program is stopped.
void draw() {
  background(0);   // Set the background to black

  x = x + ballSpeedX;
  y = y + ballSpeedY;

  //boundary checks
  if((x < 0) || (x > width)){//x to small OR x to big
    ballSpeedX = ballSpeedX * -1; //swap direction sign
    x = x + ballSpeedX;
  }

  if ((y < 0) || (y > height)) { //y to small OR y to big
    ballSpeedY = ballSpeedY * -1; //swap direction sign
    y = y + ballSpeedY;
  }

  //println("x: " + x + " y: " +y); //print to console
  ellipse(x, y, ballSize, ballSize); //the ball
}//end draw
```

Processing: receive data from Arduino

Arduino: serial sensor

- Reading analog value from pin A1
- Sending on serial port
- Print out


AnalogReadSerial | Arduino 1.6.12
File Edit Sketch Tools Help
AnalogReadSerial §

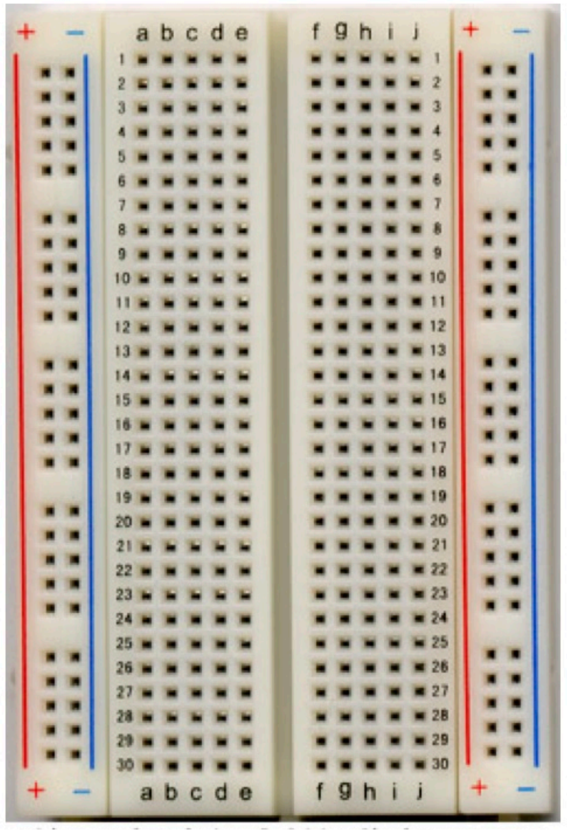
```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A1);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

14
Arduino/Genuino Uno on COM3

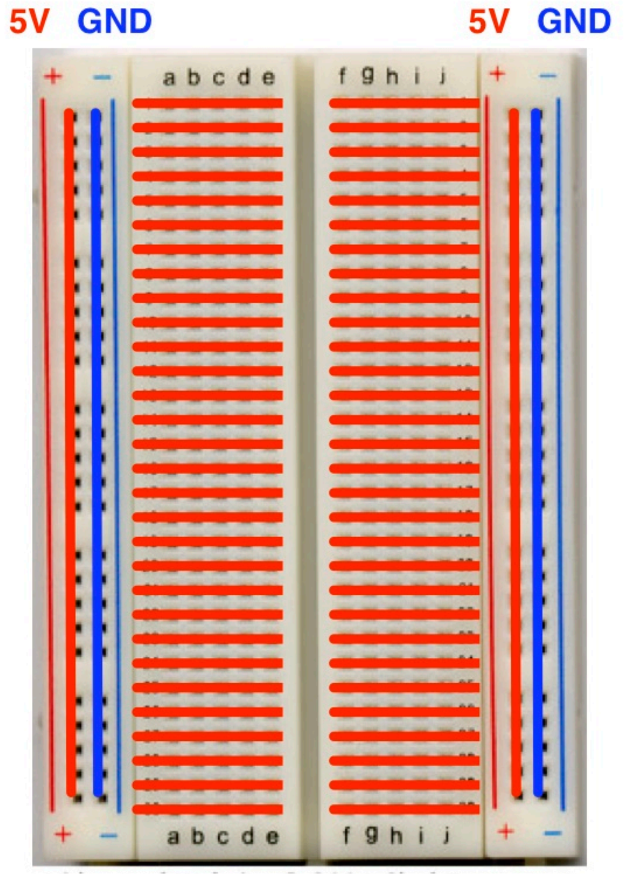
Arduino: breadboard

- Connections on the breadboard

Breadboard (photo)



Breadboard (schematic)


5V
GND
5V
GND
a b c d e
f g h i j