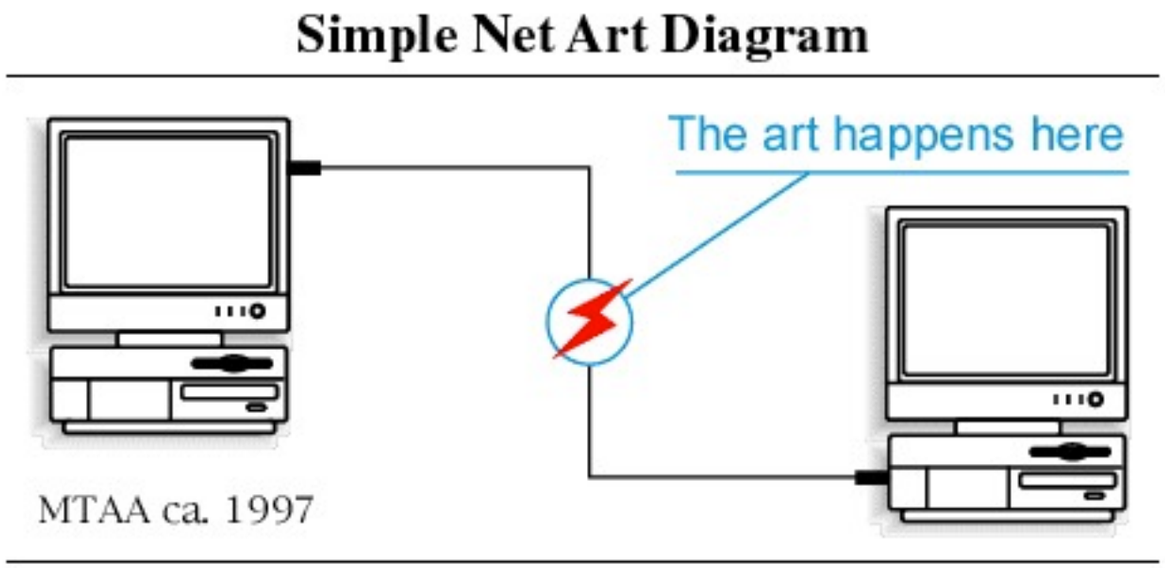
Arduino 3

Connecting to the net

Owen Mundy | Spring 2012

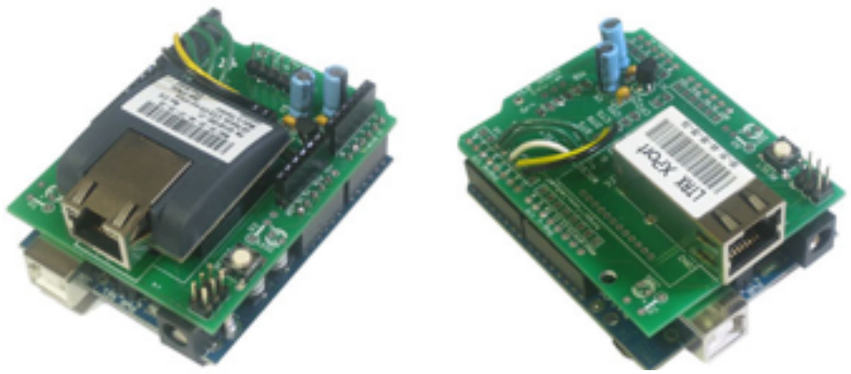


Overview

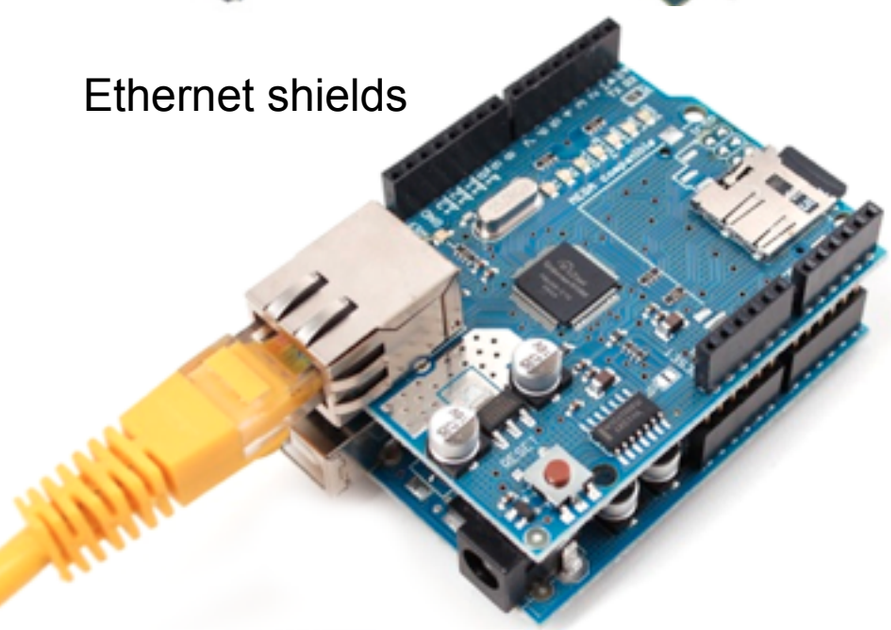
- Devices for connecting Arduino to the web
- Arduino + Processing + PHP theory of operation
- Loading local and remote files with Processing
- Basic Serial communication with Arduino + Processing
- Asynchronous loading with jquery
- Putting it all together

Arduino + internet

- There are lots of ways to connect to an Arduino to the internet.

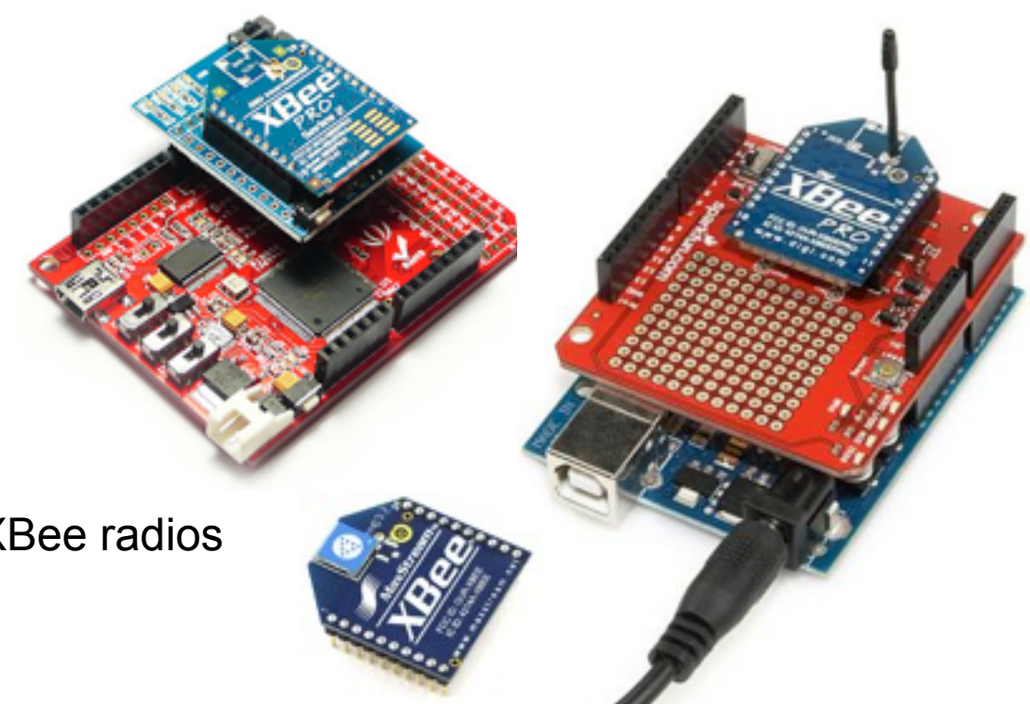


Ethernet shields



Standard Arduino + Processing

Arduino ethernet board

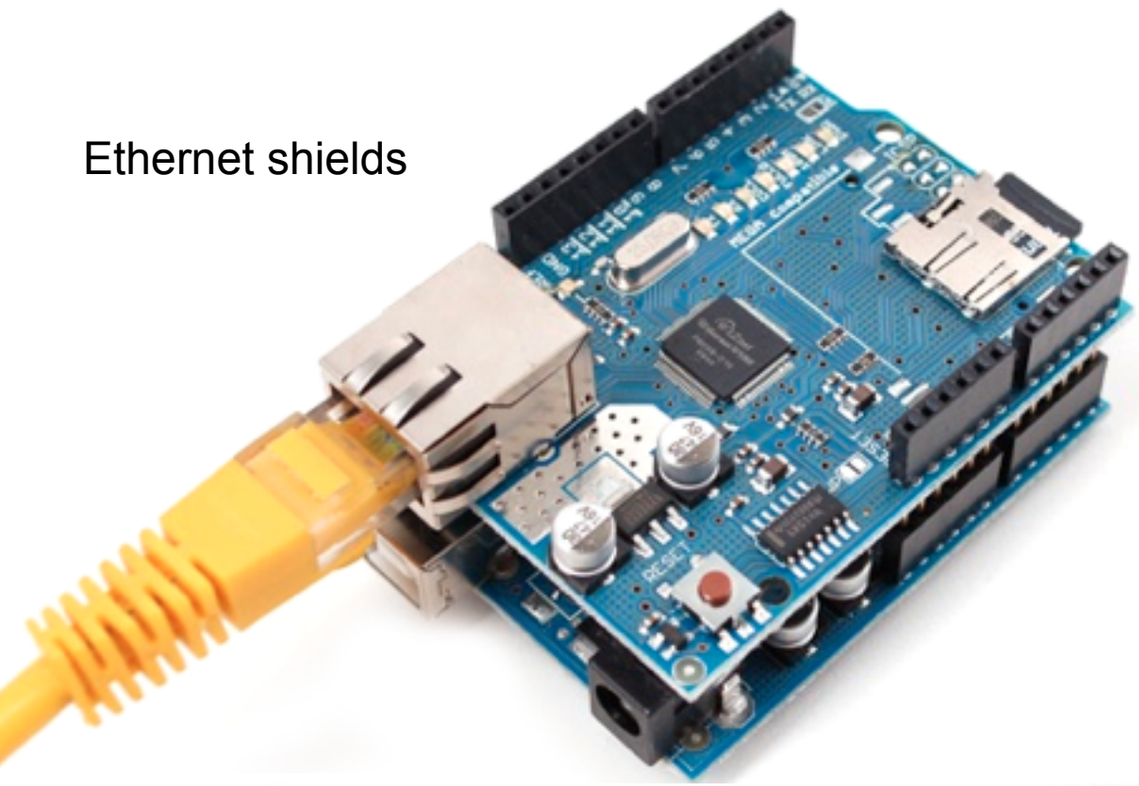


XBee radios

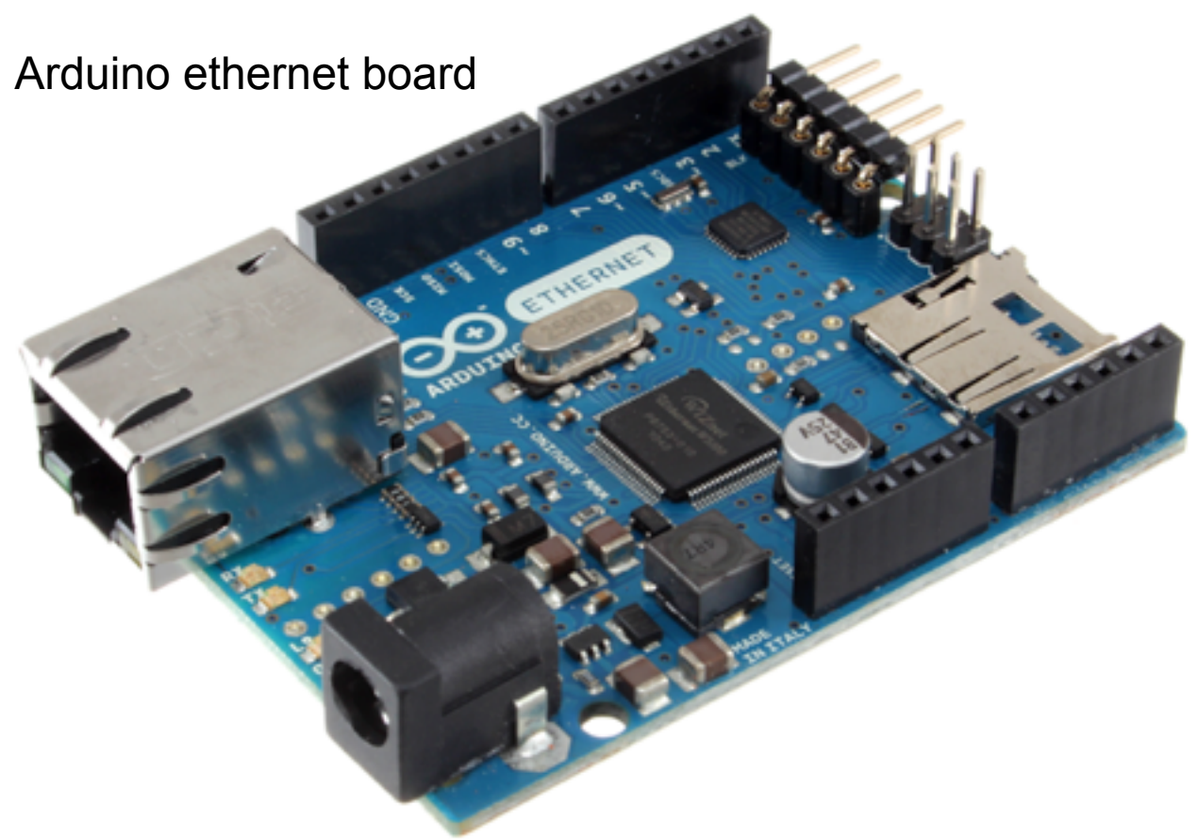
Arduino + internet

- Ethernet shields (~$46) and boards (~$55 w/oPOE | ~$80 w/POE) are expensive. And, if you fry the board you'll have to buy the whole setup again.
- Alternately, they both use the same open source library and can work as stand-alone devices. Meaning: once you program the board no computer is required.
- Example(s): Paris by Tim Schwartz

Ethernet shields



Arduino ethernet board



Arduino + internet

- XBee radios can transmit data wirelessly and operate without a computer.
- Drawbacks include complexity in software, wiring (breakout boards required), and spectrum interference (microwaves, WIFI, etc.), and price (~$23) because you have to have at least two.
- Example(s): Wind Up Birds by HC Gilje

XBee radios

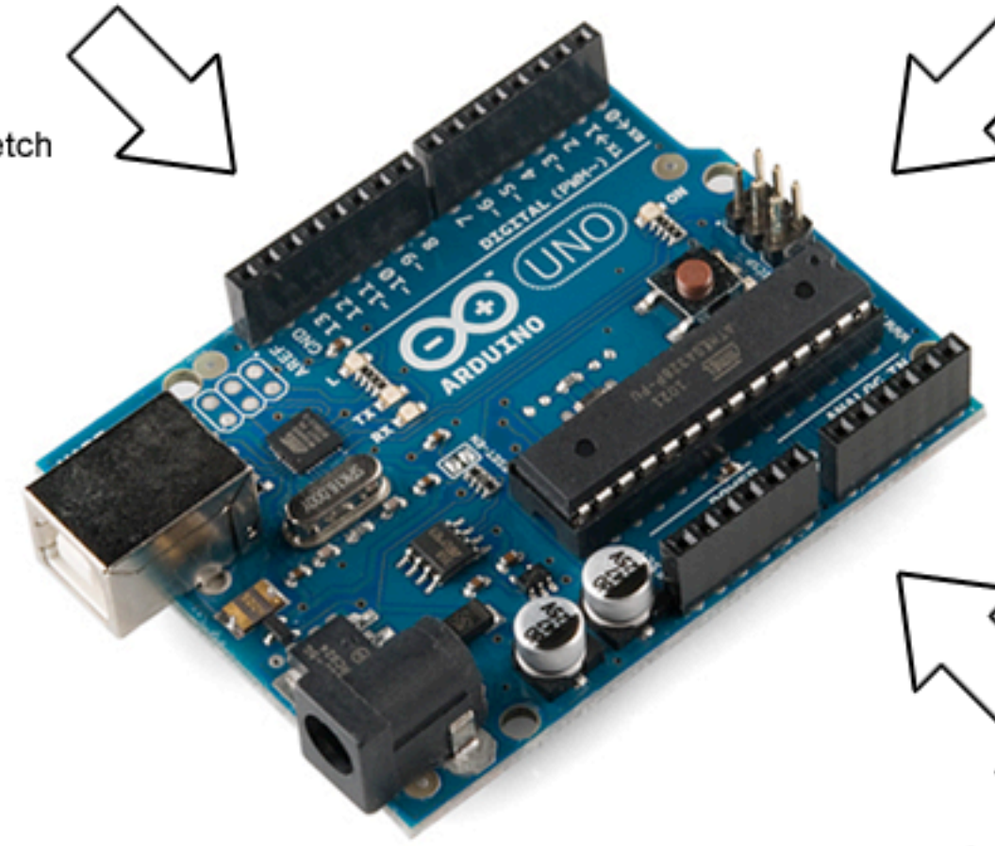
Arduino + internet

- Perhaps the simplest and most affordable method is to use Processing and a USB cable connected to your Arduino.

```
int randNum;

void setup(){
  Serial.begin(9600);
}

void loop(){
  randNum = random(0,100);
  Serial.print(randNum);
  delay(1000);
}
```

Processing can not only read *and* send data to the Arduino via USB, it can connect to the internet, communicating with PHP, databases, etc.

Upload sketch



Sensors, motors, etc.

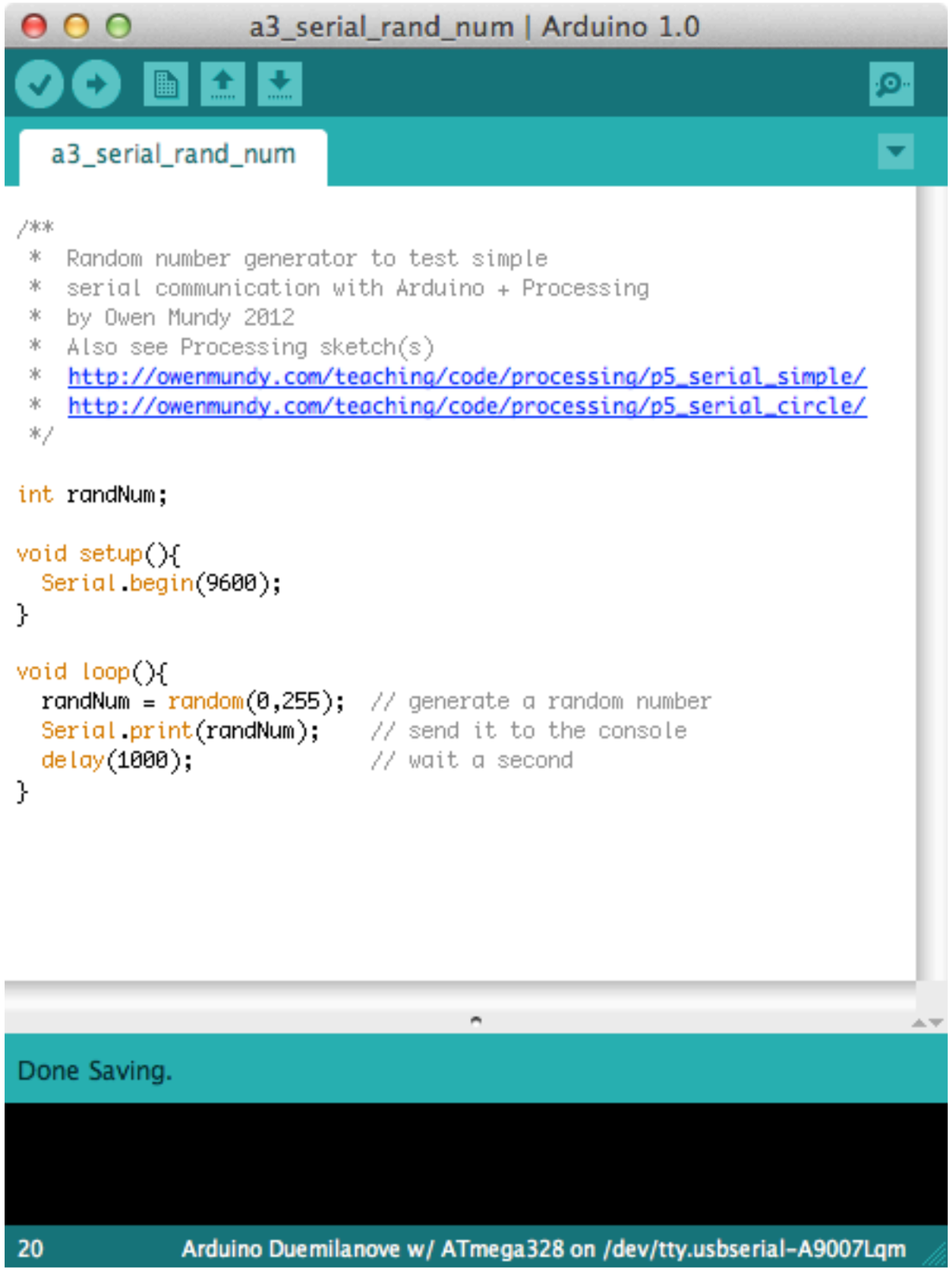
Serial communication

- Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus.
- Examples of serial communication architectures include (from slowest to fastest): Morse code (0.056 kbit/s), MIDI (31.25 kbit/s), telephone modem (56.0/48.0 kbit/s), Mobile 3G (384/384 kbit/s), Bluetooth (3 Mbit/s), IEEE 802.11g (54 Mbit/s), Mobile 4G (100/50 Mbit/s), Ethernet (100 Mbit/s), USB 2.0 (Universal Serial Bus) (480 Mbit/s), FireWire 800 (786.432 Mbit/s).
- When you choose Tools > Serial Port in Arduino software you can see a list of possible ports on which the computer can receive serial communication. You can also type the following in the Terminal application on your Mac to see a list of available serial ports: ls /dev/tty.*

Arduino serial communication

The Arduino software shows serial communication from the Arduino board in the serial monitor (the “console” in Processing). This sketch generates a random number and sends it to the computer. Click the serial monitor button to see the data after you upload the sketch.

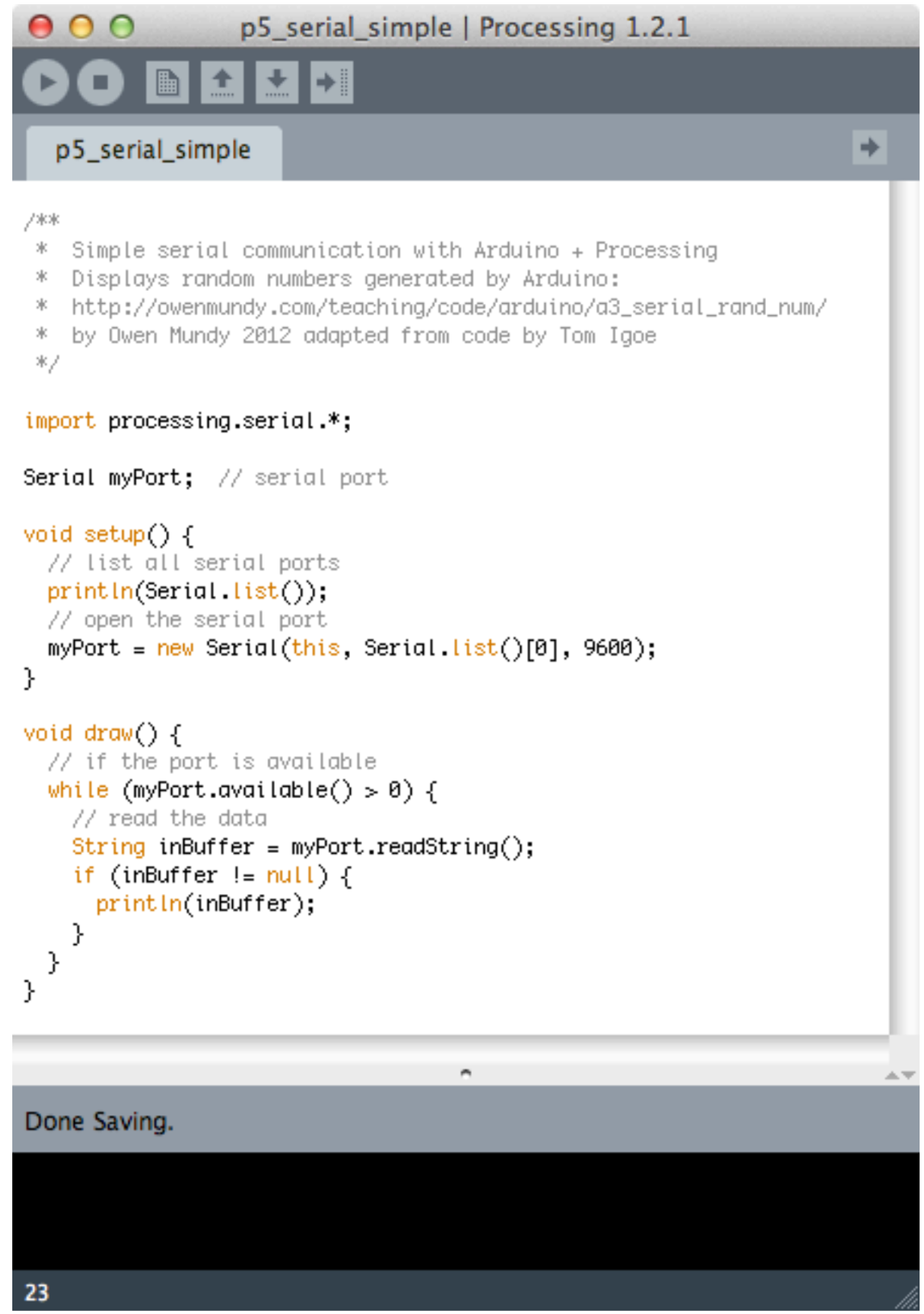
```
int randNum;

void setup(){
  Serial.begin(9600);
}

void loop(){
  randNum = random(0,255);
  Serial.print(randNum);
  delay(1000);
}
```



Processing serial communication

In Processing we can access data coming through the serial port with the console. This first part loads the serial library and creates and opens a serial port.

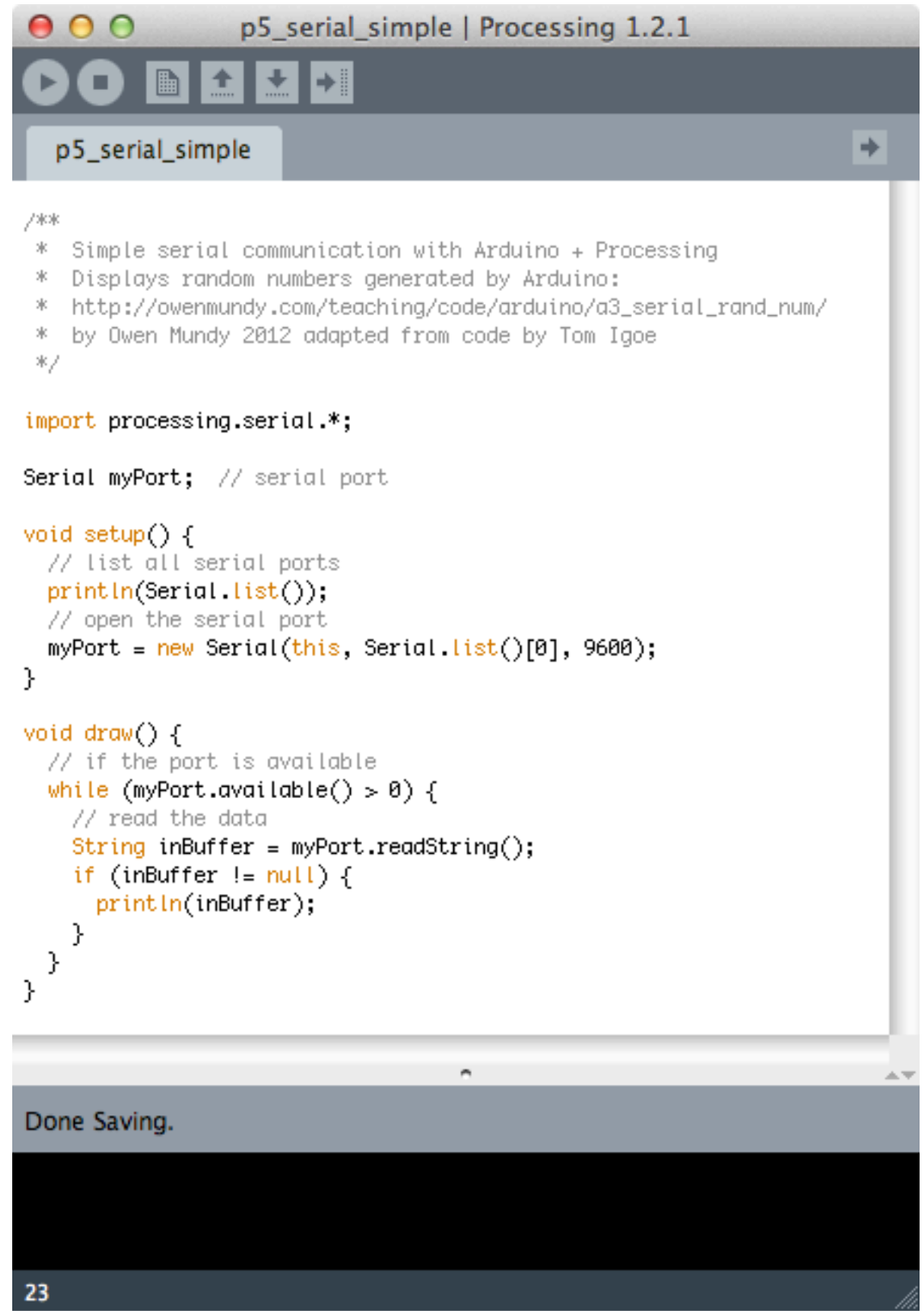
```
import processing.serial.*;

Serial myPort;  // serial port


void setup() {
  // list all serial ports
  println(Serial.list());
  // open the serial port
  myPort = new Serial(this, Serial.list()[0],
9600);
}
```



Processing serial communication

The draw() function in this sketch constantly reads the data in the serial port buffer and prints it to the console.

```
void draw() {
  // if the port is available
  while (myPort.available() > 0) {
    // read the data
    String inBuffer = myPort.readString();
    if (inBuffer != null) {
      println(inBuffer);
    }
  }
}
```



Processing serial communication

This sketch visualizes the random number coming from the Arduino via the serial port with a circle and rgb value. In addition to reading the serial data like the previous sketch, it also contains a custom function called, drawCircle(), that draws the circle with one value of the rgb fill() and the radius determined by the random number.

```
void drawCircle(){
  ...
  background(255);
  fill(255,rad,0);
  ellipse(150,150,rad,rad);
}
```

p5_serial_circle | Processing 1.2.1

p5_serial_circle

```
  // list all serial ports
  println(Serial.list());
  // open the serial port
  myPort = new Serial(this, Serial.list()[0], 9600);
}

void draw() {
  // if the port is available
  while (myPort.available() > 0) {
    // read the data
    String inBuffer = myPort.readString();
    if (inBuffer != null) {
      // store random number from Arduino
      // generated once every second
      num = int(inBuffer);
      println("----------------------------- NEW SIZE: " + num);
    }
  }
  drawCircle();
}

void drawCircle(){

  // change size of circle with easing
  if (num > rad){
    rad = floor(rad +((num-rad)/15));
  } else if (rad > num){
    rad = floor(rad -((rad-num)/30));
  }

  // reset background and change circle size and fill
  background(255);
  fill(255,rad,0);
  ellipse(150,150,rad,rad);
  println("current / new: "+ rad +" / "+ num);
}
```

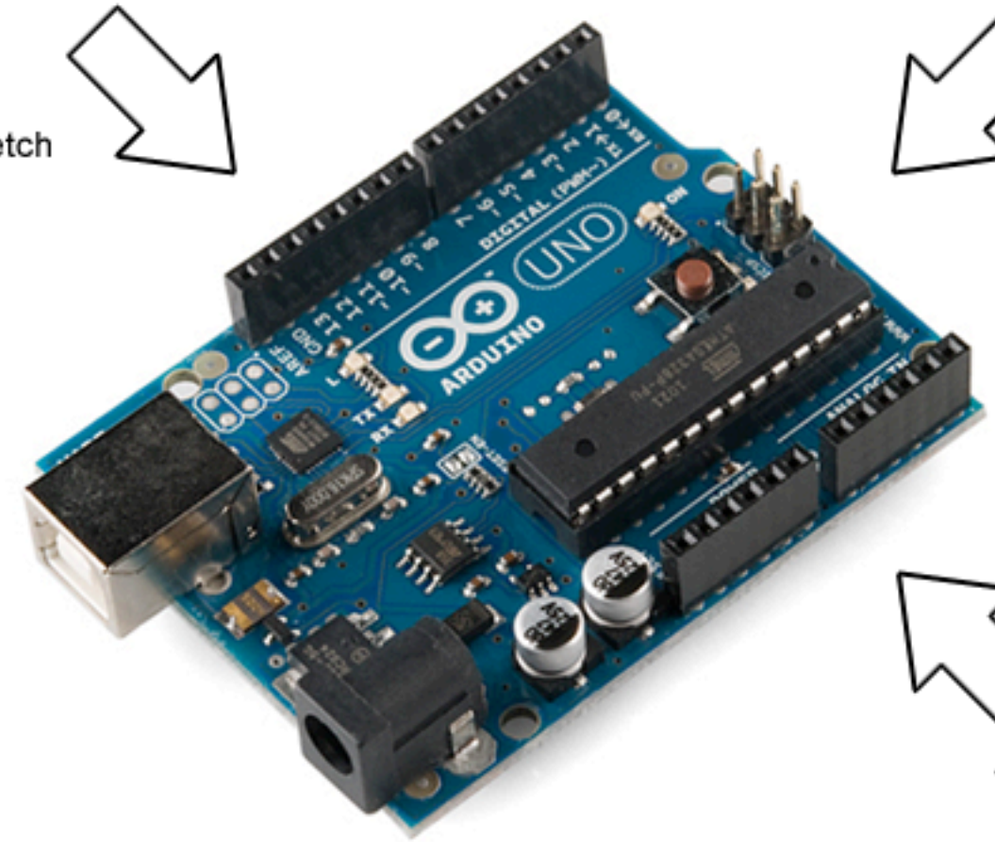
Arduino + internet

- Revisiting this graphic, we can see that Processing can also get and send data on the internet.

```
int randNum;

void setup(){
  Serial.begin(9600);
}

void loop(){
  randNum = random(0,100);
  Serial.print(randNum);
  delay(1000);
}
```

Processing can not only read *and* send data to the Arduino via USB, it can connect to the internet, communicating with PHP, databases, etc.

Upload sketch



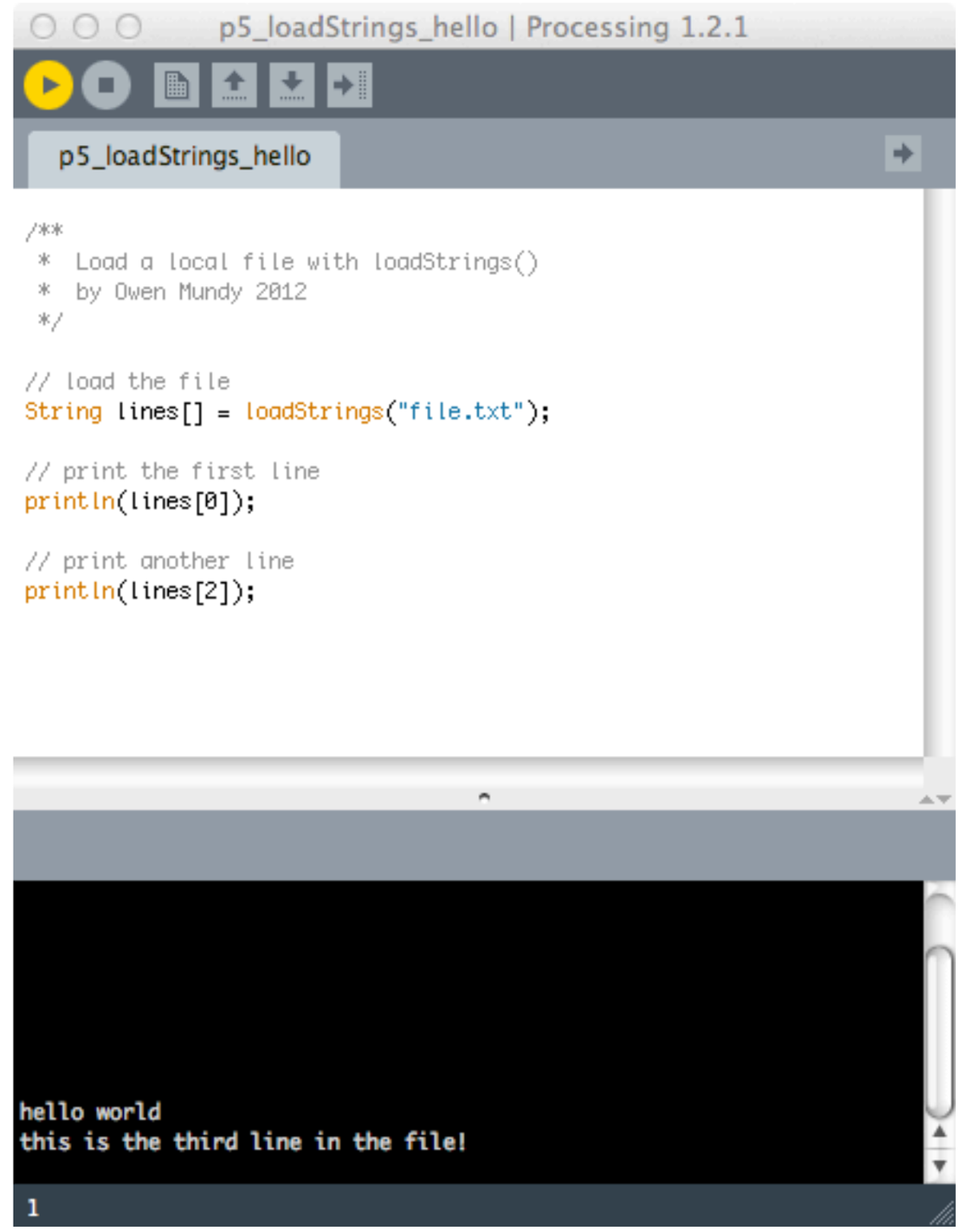
Sensors, motors, etc.

Processing: loadStrings() - Hello World

With Processing we can load practically any type of file, locally or from the web.

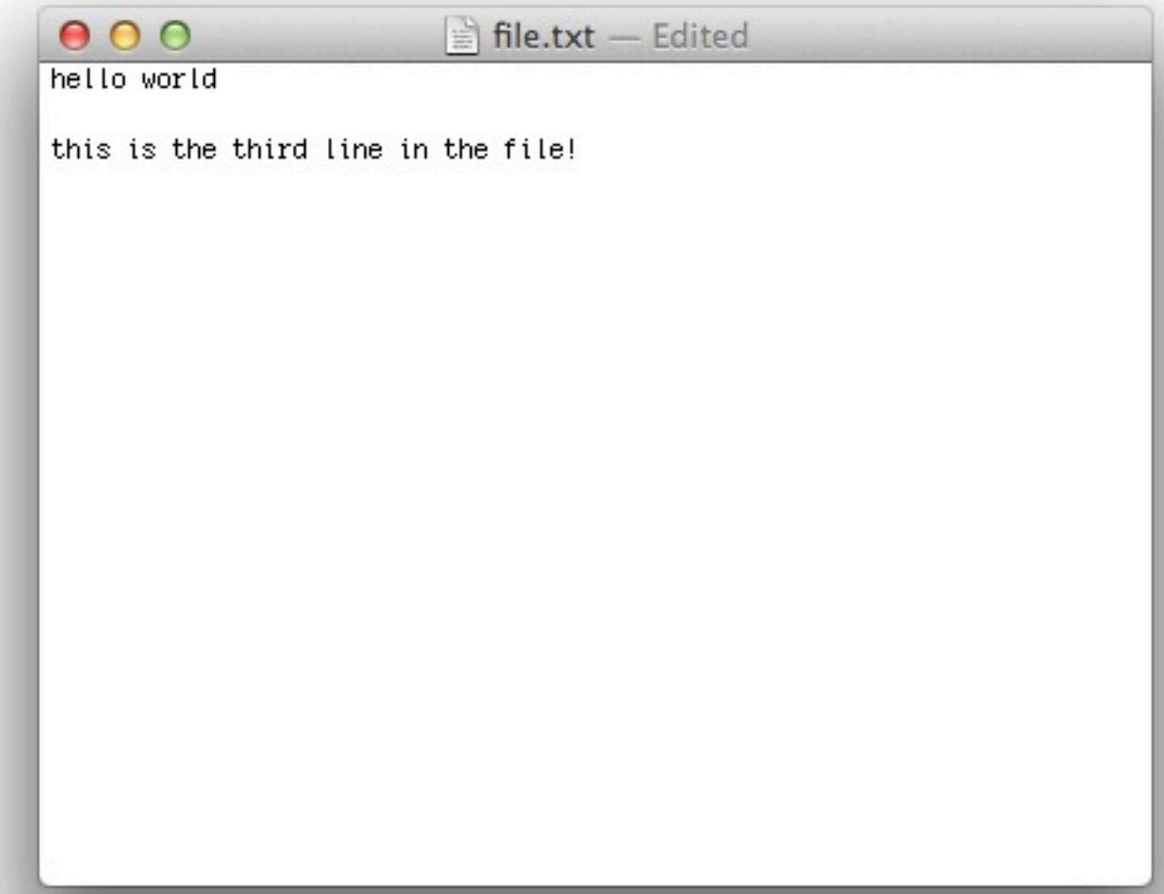
This sketch gets data from a text file and displays the contents in the Console. Files loaded locally must be stored inside a folder called, “data,” located inside the sketch folder.

```
// load the file
String lines[] = loadStrings("file.txt");

// print the first line
println(lines[0]);

// print another line
println(lines[2]);
```



Processing: loadStrings() - Hello World

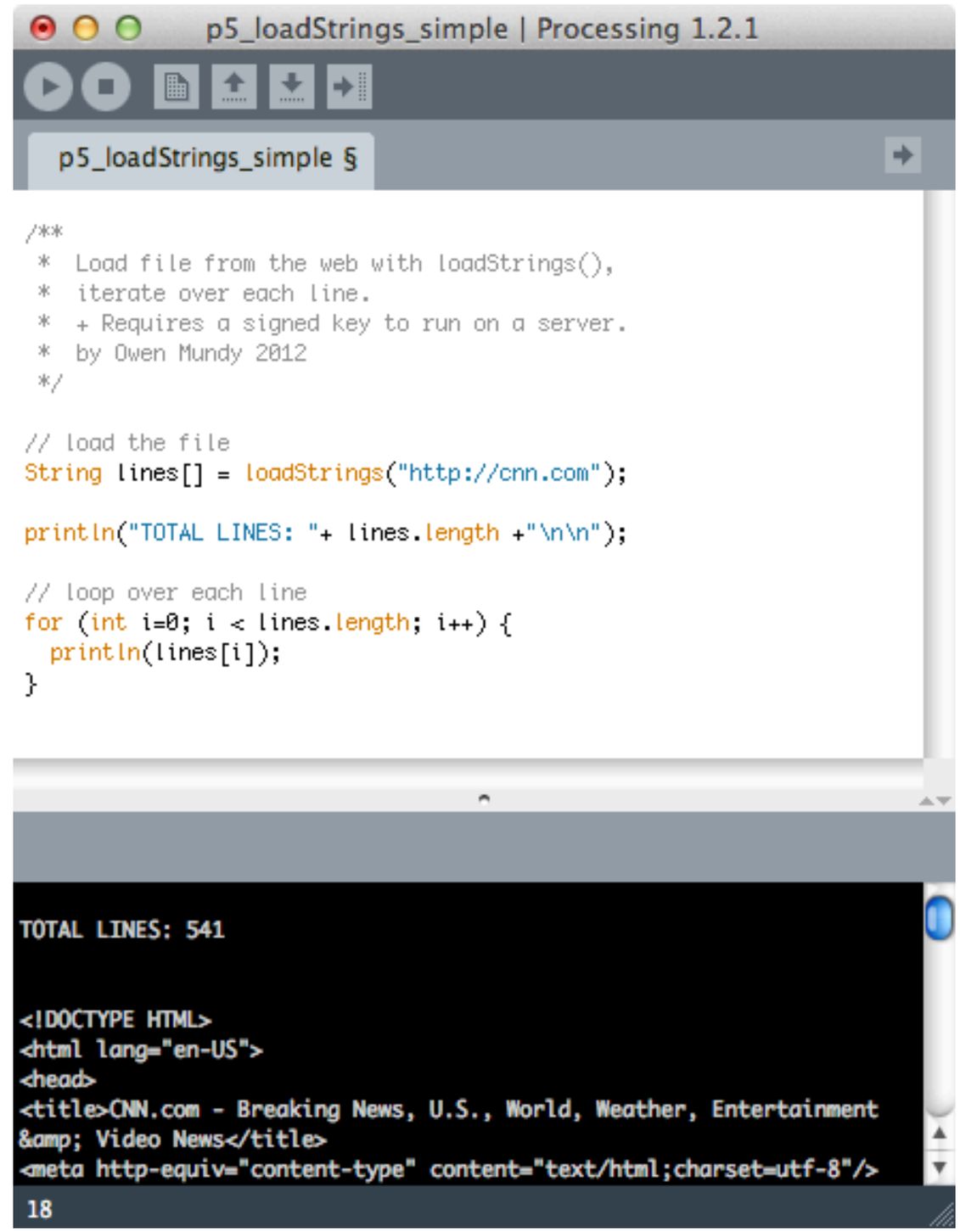
The text file looks like this.



Processing: loadStrings() - Scraper

This sketch loads a web page and displays the number of lines and the contents of the file in the Console.

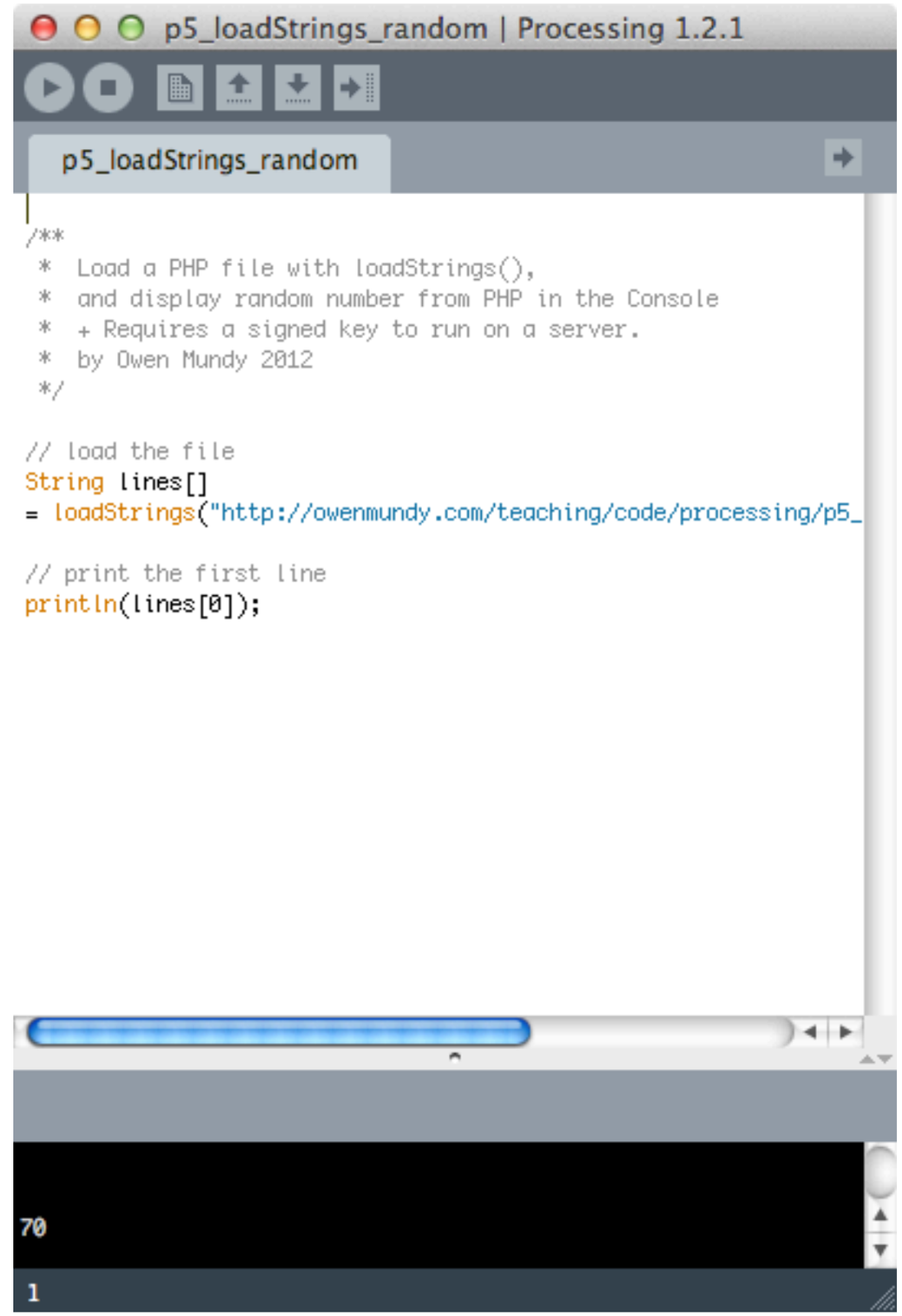
```
// load the file
String lines[] = loadStrings("http://cnn.com");

println("TOTAL LINES: "+ lines.length +"\n");

// loop over each line
for (int i=0; i < lines.length; i++) {
  println(lines[i]);
}
```



Processing: loadStrings() + PHP

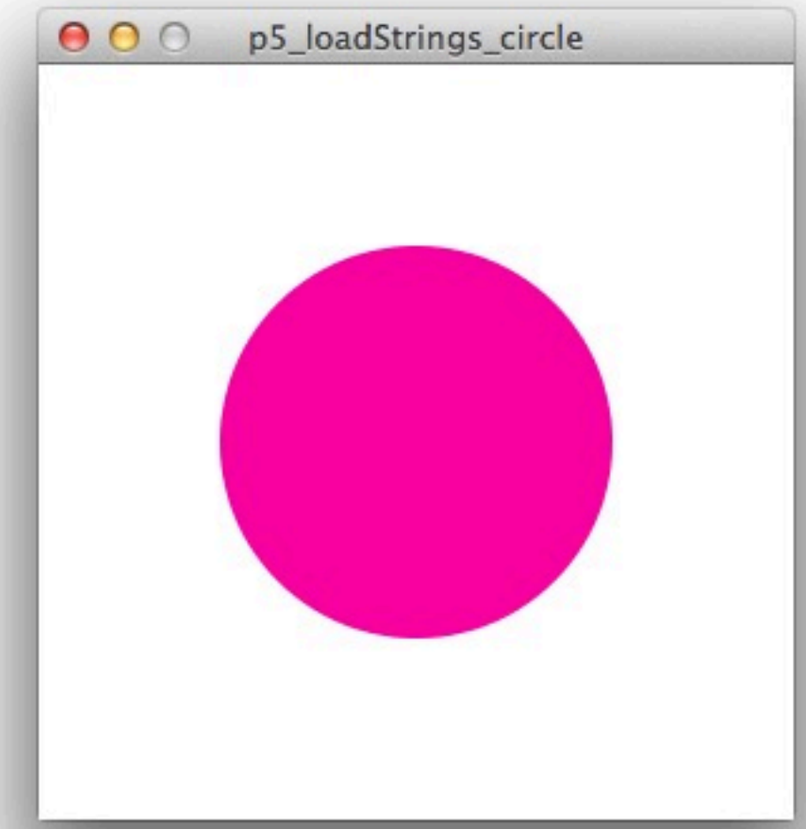
This sketch loads a PHP file that generates a random number and displays that number in the Console.

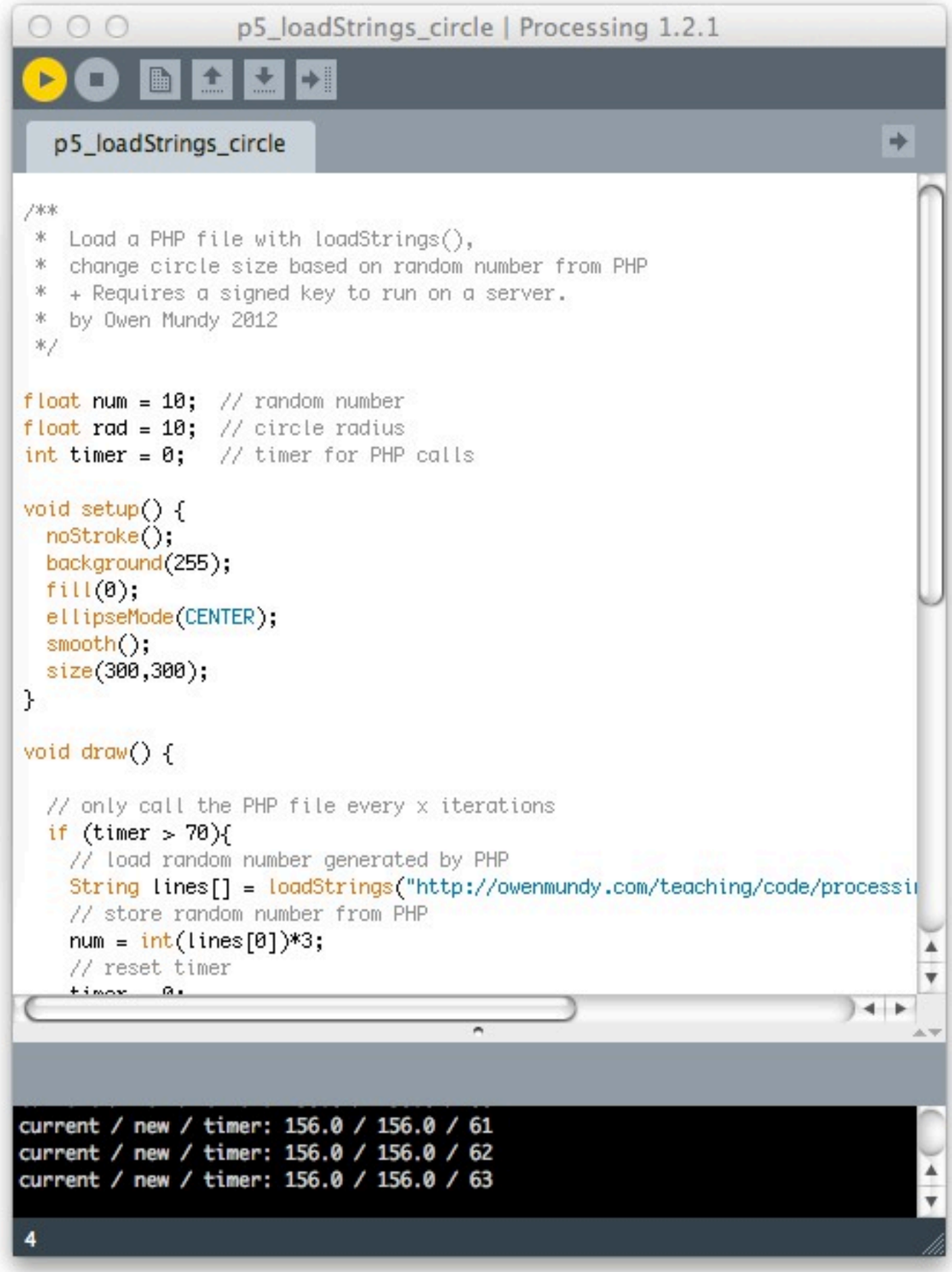
```
// load the file
String lines[] = loadStrings("http://owenmundy.com/teaching/code/processing/p5_loadStrings_random/random.php");

// print the first line
println(lines[0]);
```



Processing: loadStrings() + PHP (circle)

This sketch continuously loads a PHP file that generates a random number.





It displays that number in the Console and adjusts the size and fill of a circle according to that number. It's visual display of data from the net.

Processing: loadStrings() + PHP (circle)

1. **Arduino** sends temperature to **Processing** via serial connection.
2. **Processing** sends temperature and mouse X and Y position to **PHP**.
3. **PHP** generates a random number, sending it back to **Processing**, and saving it with the temperature and mouse positions in a TXT file.
4. **Processing** displays a circle with size and color (rgb(255,0,random)) based on the random number.
5. **jquery** works asynchronously, constantly retrieving and displaying data from the TXT file.

Display .txt file updated by P

owenmundy.com/teaching/code/processing/p5_loadStrings_circle/dis

random number from PHP:

128

mouse position in processing:

23 /

1. **Arduino** sends tempe
2. **Processing** sends ten
3. **PHP** generates a rand
4. **Processing** displays a
5. **jquery** works asynchro

HP.
, and sa
m)) bas
g data f

p5_loadStrings_circle

p5_loadStrings_circle | Processing 1.2.1

p5_loadStrings_circle

```
// only call the PHP file every x iterations
if (timer > 70){

  // get mouse X and Y position to send to PHP
  int xmouse = mouseX;
  int ymouse = mouseY;

  String url = "http://owenmundy.com/teaching/code/processing/";
  url += "p5_loadStrings_circle/random.php";
  url += "?xmouse=" + xmouse + "&ymouse=" + ymouse;

  // load random number generated by PHP
  String lines[] = loadStrings(url);
  // store random number from PHP
  num = int(lines[0]);
  // reset timer
  timer = 0;
}
timer++;
drawCircle();
}

void drawCircle(){

  // change size of circle with easing
  if (num > rad){
    rad = floor(rad +((num-rad)/15));
  } else if (rad > num){
    rad = floor(rad -((rad-num)/30));
  }

  // reset background and change circle size and fill
  background(255);
  fill(255,0,rad);
  ellipse(150,150,rad,rad);
  println("current / new / timer: "+ rad +" / "+ num +" / "+ timer);
}
```

```
current / new / timer: 114.0 / 128.0 / 64
current / new / timer: 114.0 / 128.0 / 65
current / new / timer: 114.0 / 128.0 / 66
```

39