

Java Programming and NeoPixels with Arduino

Mr. Michaud

OEC Programs at Georgia Tech

Key Functions

- `void setup()`
 - Must be defined
 - Runs one time
 - Acts like a constructor in Java
- `void loop()`
 - Must be defined
 - Runs continuously (In a loop . . .)
 - Like the `act()` method in Greenfoot or Gridworld

Data Types and Variables

bool	0	0 or 1 true or false
byte	0000 0000	Whole number: -128 to 127
char	0000 0000	Can be a character value (ie "a")
float	0000 0000 0000 0000 0000 0000 0000 0000	decimal point number
long	0000 0000 0000 0000 0000 0000 0000 0000	Whole number: -2,147,483,648 to 2,147,483,647
int	0000 0000 0000 0000	Whole number: -32,768 to 32,767
ubyte	0000 0000	Whole number between 0 and 255

Assigning Variables:

```
int power = 100; // Creates an integer equal to 100
```

```
bool touched = false;
```

```
float batteryPower = 6.78;
```

```
long clockCycles = 84234;
```

Operators

=	Assigns a value
==	Compares - "is equal to"
>	Greater than
<	Less than
!	Not
	Or
&&	And
*	Multiply
/	Divide
++	Increase by 1
+	Addition
-	Subtraction
%	Modulo

Declaring Variables and Arrays

Variables:

```
int myAge = 14; // Integer
float angle = 2*PI; // Float
String name = "Mr. Michaud";
```

Arrays:

```
int pins [] = {3, 4, 5, 6};
String gospels [] = {"Matthew",
"Mark", "Luke", "John"};
```

For Loop

- Repeats section of code while counting up or down with an index variable
- Example

```
for (int i = 0; i < 10; i++) {  
    Serial.println(i);  
}
```

Returns:

0
1
2
3
4
5
6
7
8
9

```
for (int i = 0; i < 10; i++) {}
```

- `i++` means "i = i + 1"
- `int i` means "integer i"
- `for (int i = 0; i < 10; i++)` means "For index variable i starting at 0, while i is less than 10, count be 1."

Combined For Loop and Array

```
// Array Example

String gospels [] = {"Matthew", "Mark", "Luke", "John"};

// Setup
void setup() {
  Serial.begin(9600);
  // For Loop
  for (int i = 0; i < 4; i++) {
    Serial.println(gospels[i]);
  }
}

void loop(){
  // No Code here
}
```


Conditional Statements

- 'if statement': Checks if a given statement or expression is true and then executes a section of code

```
if (score > 9) {  
    Serial.println("You Win");  
}
```

While Loop

- Executes a Segment of Code while a Condition is True

```
// Initialize integer score as 0
int score = 0;

// Loop while score is less than 20
while (score < 20) {
    // Print the Score
    System.out.println("Your Score is: " + String.valueOf(score));
    score++; // Increase Score by 1
} // end while

System.out.println("All Done!");
```

References

- Key Vocabulary:
(<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>)
- Another Good Reference:
<http://processing.org/reference/>

Pin Output Control and Timing

- **pinMode(pin, Mode)**

```
pinMode(3, OUTPUT);
```

- **digitalWrite(pin, State)**

```
digitalWrite(3, HIGH);  
digitalWrite(3, LOW);
```

- **delay(milliseconds)**

```
delay(1000);
```

Program Control

- Two Key Functions in all Arduino Programs:

```
// Key Functions

void setup() {
    // Set Pinmodes
    // Other code you want run once
}

void loop() {
    // Code you wish to run
    // Repeats while device is powered
}
```

Sample Program

Declare and set variables pointing to pins.

loop() repeats turning on, then off the lights at ½ second intervals

```
// Two LED Program

// Declare ports for LED lights red and yellow
int red = 2;
int yellow = 3;

void setup() {
  // Set Pinmodes
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
}

void loop() {
  // Blink the Lights
  digitalWrite(red, HIGH);
  digitalWrite(yellow, HIGH);
  delay(500);

  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  delay(500);
}
```

setup() sets the pin mode to output – send signals.

For Loop

- Used to repeat a set of commands

```
for (int i = 0; i < 8; i++) {  
    // repeated commands  
}
```

Means:

“Set i to zero. While i is less than 8, add 1 to i.”

This sequence will loop 8 times:

“0, 1, 2, 3, 4, 5, 6, 7”

Example:

```
void loop() {  
  // Blink the Lights with Loops  
  for (int i = 0; i < 8; i++) {  
    digitalWrite(red, HIGH);  
    delay(500);  
    digitalWrite(red, LOW);  
    delay(500);  
  }  
  
  for (int i = 0; i < 8; i++) {  
    digitalWrite(yellow, HIGH);  
    delay(500);  
    digitalWrite(yellow, LOW);  
    delay(500);  
  }  
}
```


Adafruit_NeoPixel Object

- Constructor:

```
Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO_KHZ800);
```

Number of
Lights

Pin
Assignment

Signal Configuration
Constant

```
Adafruit_NeoPixel dot = Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO_KHZ800);
```

Setup for NeoPixel

```
// Import Library
#include <Adafruit_NeoPixel.h>

// Define Signal Pin
#define PIN 8

// Create Adafruit_NeoPixel instance: strip
Adafruit_NeoPixel dot = Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO_KHZ800);
```

API for NeoPixel (instance name: strip)

Function	Example	Description
<code>.begin();</code>	<code>strip.begin();</code>	Initializes the NeoPixels. Called in <code>setup()</code> function.
<code>.show();</code>	<code>strip.show();</code>	Pushes change of signal to the NeoPixels. Called after a change of value (color, brightness, index...)
<code>.Color(red, green, blue);</code>	<code>strip.Color(255, 0, 10);</code>	Calls a color value. Used in <code>.setPixelColor()</code> function.
<code>.setPixelColor(index, color);</code>	<code>strip.setPixelColor(10, strip.Color(0, 0, 255));</code>	Sets color for individual Pixel indexed from 0.

Example Calls

```
strip.begin();  
strip.show();  
strip.Color(255, 0, 0); // Red  
strip.setPixelColor(5, strip.Color(255, 0, 0));  
// Pixel index 5 is made red
```

Sample Program

```
//Mr. Michaud:  www.nebomusic.net

// Import Library
#include <Adafruit_NeoPixel.h>

// Define Signal Pin
#define PIN 8

// Create Adafruit_NeoPixel instance: strip
Adafruit_NeoPixel dot = Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO_KHZ800);

void setup() {};
  dot.begin();
  dot.show();
}

void loop() {HIGH};
  dot.setPixelColor(0, dot.Color(100, 0, 0));
  dot.show();
  delay(500);
  dot.setPixelColor(0, dot.Color(0, 0, 100));
  dot.show();
  delay(500);
}
```