Overview

• Variables: assigning and using
  ❖ int
  ❖ float
• Loops
  ❖ for loops

Assigning and Using Variables

Arduino web site
❖ http://www.arduino.cc/en/Tutorial/Variables

Defining and using variables:
❖ All variables must be declared before use
❖ Declaration consists of a type specification and the variable name
❖ A declaration may also include an assignment
❖ Use meaningful variable names
❖ Add comments to further clarify meaning

```c
int red_pin;          // declaration only
int blue_pin = 5;     // declaration and assignment
int greenPin = 0;
float scale = 5.0/1024.0;   // Convert 10-bit value
char name = 'Bob';
```
int Variables

An int is a 16 bit, signed integer
❖ Storage requires two bytes or 16 bits
❖ $2^{16} = 65536$
❖ Split into negative and positive range: –32,768 to 32,767
❖ Computations are rounded and rolled-over as needed

Examples:

```c
int sensorVal;        // declaration only
int sensorPin = 3;    // declaration and assignment
sensorVal = analogRead(sensorPin);     // assignment
```

float Variables

A float is a signed number with a fractional part
❖ Stored in 32 bits (twice as much memory as an int)
❖ Range: $-3.4028235 \times 10^{38}$ to $3.4028235 \times 10^{38}$
❖ Floating point arithmetic introduces rounding

Examples:

```c
int sensorVal;        // value returned from analog input
int sensorPin = 3;    // pin assigned to analog input
int range = 1024;     // Maximum range of 10 bit value
float voltage;        // Voltage of the input signal
float maxVoltage = 5.0; // Maximum range of analog input
sensorVal = analogRead(sensorPin);     // get reading

// convert to floating point voltage
voltage = float(sensorVal)*maxVoltage/float(range);
```

Try it! Measure photoresistor output

Build the photoresistor circuit
and run this program

```c
int sensorVal;
int sensorPin = 3;
float voltage;
float input2volts = 5.0/1024.0;

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

void loop () {
    sensorVal = analogRead(sensorPin);
    voltage = float(sensorVal)*input2volts;
    Serial.print("Voltage = ");
    Serial.println(voltage);
}
```
Loops

Loops allow code to be repeated
❖ Repeated code goes in a block, surrounded by { }
❖ for loops
  › need a counter
❖ while loops
  › need an escape

int i;                    // declare counter
for ( i=0; i<=12; i++ ) {  // standard structure
  Serial.println(i);  // send value of i to serial monitor
}

Loops

Initial value of counter
i=0 only on first pass through the loop

Stopping test: Continue while this condition is true

Increment: How to change i on each pass through the loop

int i;                    // declare counter
for ( i=0; i<=12; i++ ) {  // standard structure
  Serial.println(i);  // send value of i to serial monitor
}

Loops

Common loop: increment by one

for ( i=0; i<=12; i++ ) {  // increment by one
  ... code block goes here
}

Common loop: increment by two

for ( i=0; i<=12; i+=2 ) {  // increment by two
  ... code block goes here
}

Decrement by one

for ( i=12; i>=0; i-- ) {  // decrement by one
  ... code block goes here
}
Try it! Modify the photoresistor program

Change the loop function
(modify your previous code)

```cpp
void loop () {
  float sensorAve;
  int   sensorSum;
  int   nave=5;

  sensor_sum = 0.0;
  for ( i=1; i<=nave; i++ ) {
    sensorVal = analogRead(sensorPin);
    sensorSum = sensorSum + sensorVal;
  }
  sensorAve = float(sensorSum)/float(nave);
  voltage = sensorAve*input2volts;
  Serial.print("Average voltage = ");
  Serial.println(voltage);
}
```

This code contains errors that you will need to fix before it runs!

Test it! Break your code to learn how it works

Change nave
❖ Increase nave from 5 to 10, 50, 100, 500
❖ Why is the reading negative for large nave?
❖ How can you fix this by changing the variable type for sensorSum?

Add print statements inside the averaging loop
```cpp
Serial.print("\t Reading = ");
Serial.println(sensorVal);
```