

Variables	Functions	Encoder
<p>Strings <code>String variableName = "Text";</code></p> <p>Integers <code>int variableName = 0;</code></p> <p>Doubles <code>double variableName = 0.1;</code></p> <p>Longs <code>long variableName = 999999999;</code></p> <p>Booleans <code>boolean variableName = false;</code></p> <p>You can also create empty variables by default by omitting the equals sign. For example, <code>String variableName;</code> will create a new empty String variable.</p>	<p>Declare a function <code>void myFunc() { // code here }</code></p> <p>Declare a function that returns an int <code>int myFunc() { return 1; }</code></p> <p>Declare a function that takes a String parameter <code>void myFunc (String param) { // code here }</code></p> <p>Store the return value of a function in a variable <code>int variableName = myFunc();</code></p>	<p>Create a new encoder plugged in pins 2 and 3 <code>Encoder myEncoder(2, 3);</code></p> <p>Read encoder (returns long) <code>Encoder.read();</code></p> <p>Write to encoder <code>Encoder.write(0);</code></p>
<p>Arrays</p> <p>Array that can hold 4 Strings <code>string[4] array;</code></p> <p>Array with pre-set content <code>string[3] array = {"Banana", "Apple"};</code></p> <p>Access the nth value of an array <code>array[n-1];</code></p> <p>Assign a value to an array <code>array[n-1] = "Pear";</code></p>	<p>Combine these concepts to create different types of functions that suit your needs. You can have a function that returns an int value with a String parameter, for example. Use void if your function doesn't return anything.</p>	
<p>Changing Values</p> <p>Create a new String variable <code>String variableName;</code></p> <p>Assign "Pear" to the variable <code>variableName = "Pear";</code></p> <p>This method works for other variable types, too. Keep in mind of the data type that you are assigning to a variable, though. You cannot assign "Pear" to an int, for example.</p>	<p>Arduino</p> <p>Set digital pin n to INPUT. <code>pinMode(n, INPUT);</code></p> <p>Set digital pin n to OUTPUT. <code>pinMode(n, OUTPUT);</code></p> <p>Read digital pin n (returns boolean) <code>digitalRead(n);</code></p> <p>Write to digital pin n (LOW or HIGH) <code>digitalWrite(n, LOW);</code></p> <p>Read value of analog pin A0 (returns int) <code>analogRead(A0);</code></p>	
<p>While Loops</p>	<p>Motor</p>	

While loops will repeatedly run the code inside it until the condition is false, in which the loop will stop and the code after it will continue.

```
while (condition) {  
    // code here  
}
```

Create a new Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
motor
plugged into port M1

Set speed of motor -> setSpeed(100);
a motor (0-255)

Change motor -> run(FORWARD);
motor
direction
(FORWARD,
BACKWARD,
RELEASE)



By **Knoddy**
cheatography.com/knoddy/

Published 18th May, 2022.
Last updated 18th May, 2022.
Page 1 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>