

Variables

Strings `String variableName = "Text";`

Integers `int variableName = 0;`

Doubles `double variableName = 0.1;`

Longs `long variableName = 999999999;`

Booleans `boolean variableName = false;`

You can also create empty variables by default by omitting the equals sign. For example, `String variableName;` will create a new empty String variable.

Arrays

Array `string[4] array;`
that can hold 4 Strings

Array `string[3] array = {"Banana", "Apple"};`
with pre-set content

Access `array[n-1];`
the nth value of an array

Assign `array[n-1] = "Pear";`
a value to an array

Changing Values

Create a new String variable `String variableName;`

Assign `variableName = "Pear";`
"- Pear" to the variable

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.

While Loops

Functions

Declare a function `void myFunc() { // code here }`

Declare a function that returns an int `int myFunc() { return 1; }`

Declare a function that takes a String parameter `void myFunc (String param) { // code here }`

Store the return value of a function in a variable `int variableName = myFunc();`

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.

Arduino

Set digital pin n to INPUT. `pinMode(n, INPUT);`

Set digital pin n to OUTPUT. `pinMode(n, OUTPUT);`

Read digital pin n (returns boolean) `digitalRead(n);`

Write to digital pin n (LOW or HIGH) `digitalWrite(n, LOW);`

Read value of analog pin A0 (returns int) `analogRead(A0);`

Motor

Encoder

Create a new encoder plugged in pins 2 and 3 `Encoder myEncoder(2, 3);`

Read encoder (returns long) `Encoder.read();`

Write to encoder `Encoder.write(0);`

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); plugged into port M1

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

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



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

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

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>