ARDUINO CHEAT SHEET

For more information visit: http://arduino.cc/en/Reference/

Structure

```
/* Each Arduino sketch must contain the
following two functions. */
void setup()
{
/* this code runs once at the beginning of
the code execution. */
}

void loop()
{
/* this code runs repeatedly over and over
as long as the board is powered. */
}
```

Comments

```
// this is a single line
/* this is
a multiline */
```

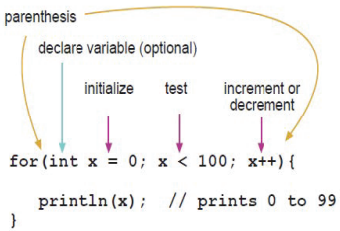
Setup

```
pinMode(pin, [INPUT \ OUTPUT \ INPUT_PUL-
LUP]);
/* Sets the mode of the digital I/O pin.
It can be set as an input, output, or an
input with an internal pull-up resistor.
*/
```

Control Structures

```
if(condition)
{
// if condition is TRUE, do something here
}
else
{
// otherwise, do this
}

for(initialization; condition; increment)
{
// do this
}
/* The 'for' statement is used to repeat
a block of statements enclosed in curly
braces. An increment counter is usually
used to increment and terminate the loop.
*/
```



Digital I/O

```
digitalWrite(pin, val);
/* val = HIGH or LOW write a HIGH or a LOW
value to a digital pin. */
int var = digitalRead(pin);
/* Reads the value from a specified digital
pin, either HIGH or LOW. */
```

Analog I/O

```
analogWrite(pin, val);
/* Writes an analog value to a pin.
val = integer value from 0 to 255 */
int var = analogRead(pin);
/* Reads the value from the specified
analog pin. */
```

Advanced I/O

```
tone(pin, freq);
/* Generates a square wave of the specified
frequency to a pin. Pin must be one of the
PWM (~) pins. */
tone(pin, freq, duration);
/* Generates a square wave of the specified
frequency to a pin for a duration in
milliseconds. Pin must be one of the PWM (~)
pins. */
noTone(pin);
// Turns off the tone on the pin.
```

Time

```
delay(time_ms);
/* Pauses the program for the amount of time
(in milliseconds). */
delayMicroseconds(time_us);
/* Pauses the program for the amount of time
(in microseconds). */
millis();
/* Returns the number of milliseconds since
the board began running the current program.
max: 4,294,967,295 */
micros();
/* Returns the number of microseconds since
the board began running the current program.
max: 4,294,967,295 */
```

Data Types

```
void      // nothing is returned
boolean   // 0, 1, false, true
char      // 8 bits: ASCII character
byte      // 8 bits: 0 to 255, unsigned
int       // 16 bits: 32,768 to 32,767, signed
long      /* 32 bits: 2,147,483,648
             to 2,147,483,647, signed */
float     // 32 bits, signed decimal
```

Constants

```
HIGH \ LOW
INPUT \ OUTPUT
true \ false
```

Mathematical Operators

```
=   // assignment
+   // addition
-   // subtraction
*   // multiplication
/   // division
%   // modulus
```

Logical Operators

```
==  // boolean equal to
!=  // not equal to
<   // less than
>   // greater than
<=  // less than or equal to
>=  // greater than or equal to
&&  // Boolean AND
||  // Boolean OR
!   // Boolean NOT
```

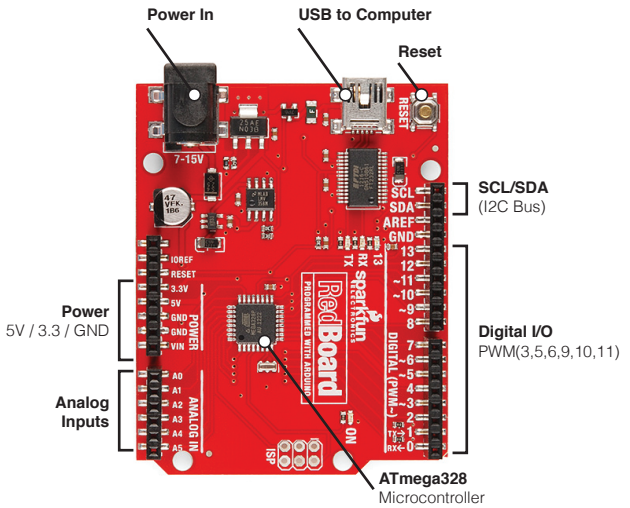
Bitwise Operators

```
&     // bitwise AND
|     // bitwise OR
^     // bitwise XOR
~     // bitwise INVERT
var << n    // bitwise shift left by n bits
var >> n    // bitwise shift right by n bits
```

Libraries

```
#include <libraryname.h>
/* this provides access to special
additional functions for things such as
servo motors, SD card, wifi, or bluetooth.
*/
```

RedBoard:



LilyPad ProtoSnap Simple: