CONSTANTS: predefined values

Pin Mode
INPUT for reading from a pin
OUTPUT for writing to a pin

Pin Status
HIGH pin is "on" or "closed"
LOW pin is "off" or "open"

Boolean
true yes
false no

VARIABLE: containers to hold values of data elements: each ends with semi-colon - ";"

int counter; // declares a variable
int counter = 0; // declares a variable and sets its initial value

STATEMENT: single command that gets executed: each ends with semi-colon - ";"

digitalWrite(13, HIGH); // turns pin 13 "on"
counter = 10; // sets a variable to the value 10
counter = counter + 1; // adds 1 to a variable (increments it by 1)

BLOCK: a group of commands that get executed: surrounded by braces - "{" and "}"
OPERATORS: for expressions

Assignment
  = replaces left-side with evaluation of right side (e.g. counter = 0;)

Arithmetic
  + addition
  − subtraction
  * multiplication
  / division
  % modulo (remainder after division)

Comparison
  == equal to
  != not equal to
  < less than
  > greater than
  <= less than or equal to
  >= greater than or equal to

Logical
  && and (boolean both)
  || or (boolean either)
  ! not (boolean opposite)
CONDITIONAL: if / then / else statements

```java
if (counter == 1) { // simple if
    // do something
}

if (counter == 1) { // simple if / else
    // do something
} else {
    // do something different
}

if (counter == 1) { // compound if / else
    // do something
} else if (counter == 2){
    // do something different
} else if (counter == 3) {
    // do another thing
} else {
    // do this if none of the above is true
}
```

Arduino Quick Reference

page 3 of 4
FOR LOOP: repeats a group of statements a number of times

```c
for (int i = 1; i <= 10; i++) {
    digitalWrite(10, HIGH); // these 4 statements in here are repeated
    delay(500);             // because they are inside the block braces
    digitalWrite(10, LOW);  // because they are inside the block braces
    delay(500);             // because they are inside the block braces
}
```

WHILE LOOP: repeats a group of statements conditionally

```c
counter = 0;
while (counter < 10) {
    counter = counter + 1; // these 2 statements in here are repeated
    delay(500);            // as long as counter is less than 10
}
```