

## Minimum Sketch:

```
void setup() {}  
void loop() {}
```

## Variable Types

Variable Type Declaration	Bits	Number Range
bool myVar=0;	1	0 or 1, false or true, LOW or HIGH
char myVar=0;	8	-128 to 127
unsigned char myVar=0;	8	0 to +255 (same as byte)
byte myVar=0;	8	0 to +255
int myVar=0;	16	-32,768 to +32,767
unsigned int myVar=0U;	16	0 to +65535
long myVar=0L;	32	-2,147,483,648 to +2,147,483,647
unsigned long myVar=0UL;	32	0 to +4,294,967,295
float myVar=0.f;	32	-3.4028235×10 <sup>38</sup> to 3.4028235×10 <sup>38</sup>

## Comments:

```
//this is a line comment  
/* this is a section comment  
that can span multiple lines */
```

## Boolean Operators:

```
a&&b // logical AND (a AND b)  
a||b // logical OR (a OR b)  
!a // logical NOT (NOT a)  
a!=b // not equals (if a≠b)  
a!=b // equals not (a = NOT(b))
```

## String and char:

```
String msg1="Welcome";  
char letter1='A'; //or use UTF-8
```

## Casting: use brackets

```
int x=5;  
float y=(float)x/2; //int to float
```

## Arrays:

```
int x[3]={1,2,3}; //x[2] is 3  
byte y[2][3]={ //y[0][1] is 2  
  {1,2,3},  
  {4,5,6}  
}  
char myMessage[6]="hello";
```

# Section 3 Cheat Sheet

## Mathematical Operators

Operator	Function	Example:
+	add	answer=x+1;
-	subtract	answer=x-1;
*	multiply	answer=x*3;
/	divide	answer=x/2;
%	modulus (computes the remainder after dividing two integers)	answer=x%2; e.g. 19%8=3, because 8 goes into 19 twice, with 3 remaining.
pow()	exponent (x <sup>b</sup> )	answer=pow(x, 3);
exp()	e <sup>x</sup> function	answer=exp(x);
abs()	absolute value	answer=abs(x);
log()	natural log	answer=log(x);
log10()	base 10 log	answer=log10(x);
sq()	square (x*x)	answer=sq(x);
sqrt()	square root	answer=sqrt(x);

```
if(condition1 goes here) {  
    action if true;  
    another action if true;  
} else if(condition2 goes here) {  
    action if true;  
    another action if true;  
} else {  
    action if false;  
    another action if false;  
}
```

optional

```
switch() case:  
switch(variable) {  
    case first value:  
        action1;  
        break;  
    case second value:  
        action2;  
        break;  
    case third value:  
        action3;  
        break;  
    default:  
        action4;  
}
```

## Relational Operators

Logical Expression Syntax	Meaning
if(x<y) {	"if x is less than y..."
if(x<=y) {	"if x is less than or equal to y..."
if(x>y) {	"if x is greater than y..."
if(x>=y) {	"if x is greater than or equal to y..."
if(x==y) {	"if x is equal to y..."
if(x!=y) {	"if x is not equal to y..."

## Loops:

### for() Loop:

```
// count up from 0 to 2, loop runs 3X:  
for(int i=0; i<3; i++) {  
    actions to repeat;  
}
```

```
// count down from 2 to 0, loop runs 3X:
```

```
for(int i=2; i>=0; i--) {  
    actions to repeat;  
}
```

### do...while() Loop:

```
do {  
    actions to repeat;  
} while(condition goes here); //tests last
```

### while() Loop:

```
while(condition goes here) { //tests first  
    actions to repeat;  
}
```

```
break; // leave a loop here
```

```
delay(100); // wait 100 msec
```

```
while(true); //stop program here
```