



Arduino Programming Cheat Sheet



Structure & Flow

Basic Program Structure

```
void setup() {
  // runs once when sketch starts
}

void loop() {
  // runs repeatedly
}
```

Control Structures

```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
do { ... } while (x < 5);
for (int i = 0; i < 10; i++) { ... }
break; // exit a loop immediately
continue; // go to next iteration
switch (myVar) {
  case 1:
    ...
    break;
  case 2:
    ...
    break;
  default:
    ...
}
return x; // or "return;" for voids
```

Operators

General Operators

```
= (assignment operator)
+ (add) - (subtract)
* (multiply) / (divide)
% (modulo)
== (equal to) != (not equal to)
< (less than) > (greater than)
<= (less than or equal to)
>= (greater than or equal to)
&& (and) || (or) !(not)
```

Compound Operators

```
++ (increment)
-- (decrement)
+= (compound addition)
-= (compound subtraction)
*= (compound multiplication)
/= (compound division)
&= (compound bitwise and)
|= (compound bitwise or)
```

Bitwise Operators

```
& (bitwise and) | (bitwise or)
^ (bitwise xor) ~ (bitwise not)
<< (shift left) >> (shift right)
```

Built-in Functions

Pin Input/Output

```
Digital I/O (pins: 0-13 A0-A5)
pinMode(pin, [INPUT, OUTPUT])
int digitalRead(pin)
digitalWrite(pin, value)
// Write HIGH to an input to
// enable pull-up resistors
```

```
Analog In (pins: 0-5)
int analogRead(pin)
analogReference(
  [DEFAULT, INTERNAL, EXTERNAL])
PWM Out (pins: 3 5 6 9 10 11)
analogWrite(pin, value)
```

Time

```
unsigned long millis()
// overflows at 50 days
unsigned long micros()
// overflows at 70 minutes
delay(msec)
delayMicroseconds(usec)
```

Type Conversions

```
char() byte()
int() word()
long() float()
```

Math

```
min(x, y) max(x, y) abs(x)
sin(rad) cos(rad) tan(rad)
sqrt(x) pow(base, exponent)
constrain(x, minval, maxval)
map(val, fromL, fromH, toL, toH)
```

Random Numbers

```
randomSeed(seed) // long or int
long random(max)
long random(min, max)
```

Bits and Bytes

```
lowByte(x) highByte(x)
bitRead(x, bitn)
bitWrite(x, bitn, bit)
bitSet(x, bitn)
bitClear(x, bitn)
bit(bitn) // bitn: 0=LSB 7=MSB
```

Variables, Arrays, and Data

Data Types

```
void
boolean (0, 1, true, false)
char (e.g. 'a' -128 to 127)
int (-32768 to 32767)
long (-2147483648 to 2147483647)
unsigned char (0 to 255)
byte (0 to 255)
unsigned int (0 to 65535)
word (0 to 65535)
unsigned long (0 to 4294967295)
float (-3.4028e+38 to 3.4028e+38)
double (currently same as float)
```

Constants

```
HIGH | LOW
INPUT | OUTPUT
true | false
143 (Decimal)
0173 (Octal - base 8)
0b10111111 (Binary)
0x7B (Hexadecimal - base 16)
7U (force unsigned)
10L (force long)
15UL (force long unsigned)
10.0 (force floating point)
2.4e5 (2.4*10^5 = 240000)
```

Arrays

```
int myInts[6]; //array of 6 ints
int myPins[]={2, 4, 8, 3, 6};
int mySensVals[6]={2, 4, -8, 3, 2};
myInts[0]=42; //assigning first
//index of myInts
myInts[6]=12; //ERROR! Indexes
//are 0 though 5
```

Strings

```
char S1[15];
char S2[8]='G','a','l','i','l','e','o';
char S3[8]='G','a','l','i','l','e','o';
char S4[]="Galileo";
char S5[8] = "Galileo";
```

Pointer Access

```
& (reference: get a pointer)
* (dereference: follow a pointer)
```

Qualifiers

```
static // persists between calls
volatile // use RAM (nice for ISR)
const // make read only
PROGMEM // Use flash
```