ARDUINO CHEAT SHEET

Content for this Cheat Sheet provided by Gavin from Robots and Dinosaurs.
For more information visit: http://arduino.cc/en/Reference/Extended

Structure
void setup() { void loop();

Control Structures
if (x<5){ } // else {} switch (myvar) { case 1: break; case 2: break; default: }
}
for (int i=0; i <= 255; i++)
while (x<5){ } do {} while (x<5); continue; // Go to next in do/while loop
return x // Or “return;” for voids.
goto // considered harmful :;

Further Syntax
// (single line comment) /* (multi-line comments) */
#define DOZEN 12 / included by: 
#include <avr/pgmspace.h>

General Operators
= (assignment operator) += (addition) -= (subtraction) *= (multiplication) /= (division)
% (modulo) == (equal to) != (not equal to) <= (less than) > (greater than)
>= (less than or equal to) >= (greater than or equal to) && (and) || (or)

Pointer Access
& reference operator * dereference operator

Bitwise Operators
& (bitwise and) // (bitwise or)
^ (bitwise xor) ~ (bitwise not)
<< (bitshift left) >> (bitshift right)

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

Constants
HIGH I LOW INPUT | OUTPUT
true | false 143 // Decimal number
0173 // Octal number
0b11011111 //Binary
0x7B // Hex number
7U // Force unsigned
10L // Force long
15UL // Force long unsigned
10.0 // Forces floating point
2.4e5 // 240000

Data Types

void boolean (0, 1, false, true)
char (e.g. a‘-128 to 127)
unsigned char (0 to 255)
byte (0 to 255)
int (-32,768 to 32,767)
unsigned int (0 to 65535)
word (0 to 65535)
long (-2,147,843,648 to 2,147,843,647)
long (0 to 2,147,983,647)
float (-3.4028235E+38 to 3.4028235E+38)
double (currently same as float)
sizeof (returns 2 bytes)

Strings
char S1[15];
char S2[8]="a",r',d',u",'n",'o";
char S3[S]="a",r',d',u",'n",'n",'o",'o";
#include "null termination
char S4[] = "arduino";
char S5[6] = "arduino";
char S6[15] = "arduino";

Arrays
int myInts[6];
int myPins[] = {2,4,8,3,6};
int mySensVs[6] = {2,4,-8,3,2};

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

Qualifiers
static // persists between calls
volatile // use RAM (nice for ISR)
const // make read-only
PROMMEM // use flash

Digital I/O
tone(pin, freq, on/off)
tone(pin, freq, duration_ms)
tone(pin, freq, duration_ms)

Analog I/O
analogRead(pin) //Call twice if...

Advanced I/O

Servo
Servo (pin, min_us, max_us)

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

Random Numbers
randomSeed(seed) // Long or int long random(max)
long random(min, max)

Bits and Bytes

Wire
Wire (I2C)

SoftwareSerial(RxPin, TxPin)

Serial
Serial.begin(300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200)

External Interrupts
attachInterrupt(interrupt, function, [LOW,CHANGE,RISING,FALLING])
detachInterrupt(interrupt)

EEPROM

Libraries:

microSD

Duemilanove/ Nano/ Pro/ ProMini

Mega

# of IOs
14 + 5 analog (Nano has 14 + 8)
54 + 16 analog

Serial Pins
0 RX 1 TX
19 0 RX2 1 TX2
17 RX 16 TX2
15 RX4 14 TX4

Ext Interrupts
2 - [INT 0]
1 - [INT 1]

PWM Pins
5.6 - Timer 0
9.10 - Timer 1

SPI
10 - SCK
11 - MOSI
13 - MISO
12 - MOSI
13 - SCK

I2C
AnalogOut - SDA
AnalogOut - SCK

Servo
Servo

SoftwareSerial
SoftwareSerial

begin()
begin()

Wire
Wire

begin()
begin()