

Arduino Cheat Sheet

by Maker Learners



Structures

Basic Structure

```
void setup(){  
    // Code here runs only once  
    // when Arduino powers on or resets.  
    // Used to initialize variables or  
    // pin modes and call libraries.  
}  
  
void loop(){  
    // Code here loops  
    // continuously, allowing your  
    // program to change and respond  
}
```

Control Structure

```
break; //exit loop or switch  
continue; //go to the next do/for/while loop  
return; //terminate function  
return x; //terminate function and return  
the value x  
if(x < 5) { ... } else { ... }  
for (int i = 0; i < 10; i++) { ... }  
while (x < 5) { ... } else { ... }  
do { ... } while (x < 5);  
switch (var) {  
    case 1:  
        ...  
        break;  
    case 2:  
        ...  
        break;  
}
```

Function Structure

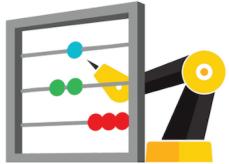
```
<type> <name> <parameters> { ... }  
e.g. int double ( int x ) { return x * 2; }
```

Operators

Mathematical Operators

Many of the mathematical operators look the same, but some are different.

- = Assignment
- + Add
- Subtract
- * Multiply
- / Divide
- % Modulo



Comparison Operators

- == Equal to
- != Not Equal to
- < Less than
- > Greater than
- <= Less than or Equal to
- >= Greater than or Equal to

Compound Operators

- ++ Increment
- Decrement
- += Compound addition
- = Compound subtraction
- *= Compound multiplication
- /= Compound division
- &= Compound bitwise and
- |= Compound bitwise or

Boolean Operators

- && And
- || Or
- ! Not

Bitwise Operators

- & Bitwise and
- ^ Bitwise xor
- << Shift left
- | Bitwise or
- ~ Bitwise not
- >> Shift right

Functions

Pin Inputs and Outputs

Digital I / O

```
pinMode( pin, [ INPUT / OUTPUT /  
INPUT_PULLUP ] )  
int digitalRead ( pin )  
digitalWrite ( pin, [ HIGH / LOW ] )
```

Analog In

```
int analogRead ( pin )  
analogReference ([ DEFAULT / INTERNAL /  
EXTERNAL ] )
```

PWM Out

```
analogWrite ( pin, value )
```

Time

```
delay ( millisec )  
delayMicroseconds ( microsec )
```

Tones

```
Tone ( pin, freqHz {, duration} )  
noTone ( pin ) //tone is limited to one pin  
at a time
```

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 (x, fromLow, fromHigh, toLow, toHigh)		

Type Conversions

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

Data

Data Types

boolean	true false
char	Any symbol on ASCII chart
byte	Number from 0 to 255
int	Number from -32,768 to 32,767
unsigned int	Number from 0 to 65,535
long	Number from -2,147,483,648 to 2,147,483,647
unsigned long	Number from 0 to 4,294,967,295
float	Number from -3.4028235 x10 ³⁸ to 3.4028235 x10 ³⁸
void	Declares a function has no return value

Strings

```
char Str1[ 8 ] = { 'a', 'r', 'd', 'u', 'i', 'n', 'o' };  
char Str1[ 8 ] = { 'a', 'r', 'd', 'u', 'i', 'n', 'o', '\0' };  
/* strings made as character arrays above  
need space for a null-terminator '\0' */  
char Str1[ ] = "arduino";  
char Str1[ 10 ] = "arduino";
```

Arrays

```
int myPins[ ] = { 2, 4, 8, 5 };  
int mInts[ 6 ];  
int mySensVals[ 6 ] = { 2, 4, -8, 3, 2 };  
char message[ 6 ] = "hello";
```

