

The C-Language Keywords and Symbols

Keywords		Symbols			
MEMORY	CONTROL	CONTROL	LOGIC	MATH	BIT OP
01.void	21.return	#	==	*	
02.char	22.if	< >	!=	%	&
03.int	23.else	//	<	/	^
04.short	24.switch	/* */	>	+	~
05.long	25.case	()	<=	-	<<
06.float	26.default	{ }	>=		>>
07.double	27.while	;	&&		
08.signed	28.do	,			
09.unsigned	29.for	"	!		
10.struct	30.break	'			
11.union	31.continue	=			
12.enum	32.goto	[]			
13.const		:			
14.volatile		?			
15.auto		.			
16.extern		\			
17.static		MEMORY			
18.register		&			
19.typedef		*			
20.sizeof					

MATH OPERATIONS

These symbols allow us to do basic math with numbers

C-Language Math Operation Symbols

* Multiply, $a=b*c$; a = result from b multiply c
 % Modulus, $a=b\%c$; a = remainder of a division, b divided by c
 / Divide, $a=b/c$; a = result from b divided by c
 + Plus, $a=b+c$; a = result from b plus c
 - Minus, $a=b-c$; a = result from b minus c

Arduino IDE|Save PROGRAM as: **c_math_basic**

Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
    Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...\n");

    Serial.print("\n\nMultiply *");
    Serial.print("\n4 * 2 = ");Serial.print(4*2);

    Serial.print("\n\nDivide /");
    Serial.print("\n4 / 2 = ");Serial.print(4/2);

    Serial.print("\n\nAdd +");
    Serial.print("\n4 + 2 = ");Serial.print(4+2);

    Serial.print("\n\nMinus -");
    Serial.print("\n4 - 2 = ");Serial.print(4-2);

    Serial.print("\n\nModulus %");
    Serial.print("\n 8 % 5 = ");Serial.print(8%5);
    Serial.print("\n 1 % 5 = ");Serial.print(1%5);
    Serial.print("\n15 % 5 = ");Serial.print(15%5);
}
void loop(){}

```

We can also do shortcut coding when using math operation symbols

a = a+b; can also be coded as **a += b;** // this can also work for other operations

Increase and Decrease number by 1

b++; // use original value, then increase by 1 after operation is completed
 b--; // use original value, then increase by 1 after operation is completed
 ++b; // increase by 1 before operation starts
 --b; // decrease by 1 before operation starts

```
b=4, c=7;
a = b++ + ++c; // results = 4+8 = 12;
```

We can use the bracket () pair symbol to enforce math operation precedence

```
a = 2; b=4, c=7;
```

```
a = (b+(c-a))*a;
```

Step 1. This will start with most inner bracket pair "()" c-a
 Step 2. result of Step 1. plus b
 Step 3. result of Step 2. multiply a
 Step 4. result of Step 3. assigned to a