

C Quick Reference

Common C Syntax



General

Description	Syntax	Example(s)
Function prototype	<i>returntype</i> <i>functionname(parameterList);</i>	<pre>int test(int, int); /* Names for parameters are optional */</pre>
Functions	<i>returntype</i> <i>functionname(parameterList)</i> { <i>declarations and Statements</i> }	<pre>int test(int one, int two) { return one + two; }</pre>
Creating a new name for data type	<i>typedef type name;</i>	<pre>typedef int myInt;</pre>
Size of a data type (in bytes)	<i>sizeof(type)</i>	<pre>sizeof(int)</pre>
Printing	<i>printf("format", arguments)</i>	<pre>int x = 0; printf("%d", x);</pre>
Reading	<i>scanf("format", arguments)</i>	<pre>int x; scanf("%d", &x);</pre>

Memory Allocation

Description	Syntax	Example(s)
Memory allocation (returns starting address of the memory allocated)	<i>malloc(size)</i> <i>calloc(nitems, itemsize)</i>	<pre>char *a; a = malloc(sizeof(char) * 5);</pre>
Memory Re-allocation	<i>realloc(ptr, size)</i>	<pre>/* From above */ a = realloc(a, sizeof(char)*6);</pre>
Free up memory allocation	<i>free(ptr)</i>	<pre>/* From above */ free(a);</pre>

Preprocessor

Description	Syntax	Example(s)
Include library file	<i>#include <filename></i>	<pre>#include <stdio.h></pre>
Include user file	<i>#include "filename"</i>	<pre>#include "myFunctions.h"</pre>
Replacement Text	<i>#define name text</i>	<pre>#define SIZE 10</pre>
Replacement Macro	<i>#define name(var) text</i>	<pre>#define double(A) 2 * (A)</pre>

C Quick Reference

Common C Syntax



String Manipulation

Description	Syntax	Example(s)
Length of a string	<code>strlen(char *)</code>	<pre>char string[] = "HELLO"; int length = strlen(string);</pre>
Copy a string to another location	<code>strcpy(char* dest, char* source)</code>	<pre>char source[] = "HI"; char *dest; strcpy(dest, source);</pre>
Concatenate two strings	<code>strcat(char *, char *)</code>	<pre>char one[] = "HI "; strcat(one, "WORLD");</pre>
Compare two strings	<code>strcmp(char *, char *)</code>	<pre>char one[] = "HI"; if(strcmp(one, "HI") == 0) { puts("SAME"); }</pre>
Convert string to an int	<code>atoi(char *)</code>	<pre>char string[] = "24"; int value = atoi(string);</pre>

Creating/Declaring variables

Description	Syntax	Example(s)
Declaring a pointer	<code>type *name;</code>	<pre>int *valuePtr;</pre>
Assigning a pointer to the address of a variable	<code>pointername = &variablename</code>	<pre>int value = 2; int *valuePtr = &value;</pre>
Declaring a structure	<pre>struct tag { type name; type name; } /* The tag is optional */</pre>	<pre>struct person { char *firstName; char *lastName; int age; }</pre>
Creating a structure	<code>struct tag name;</code>	<pre>/* From above */ struct person Person;</pre>
Accessing a member of a structure (dot operator)	<code>name.member</code>	<pre>/* From above */ Person.age = 10;</pre>
Accessing a member from a structure pointer	<code>name->member</code>	<pre>/* From above */ Struct person *personPtr; personPtr = &Person; personPtr->age = 20;</pre>
Creating an array	<code>type name[size];</code>	<pre>int array[10];</pre>
Initializing/Creating an array	<code>type name[] = {value, ...};</code>	<pre>int array[] = {1, 2, 3};</pre>