

COMMAND-LINE ARGUMENTS

http://www.tutorialspoint.com/objective_c/objective_c_command_line_arguments.htm Copyright © tutorialspoint.com

It is possible to pass some values from the command line to your Objective-C programs when they are executed. These values are called **command line arguments** and many times they are important for your program, especially when you want to control your program from outside instead of hard coding those values inside the code.

The command line arguments are handled using main function arguments where **argc** refers to the number of arguments passed, and **argv[]** is a pointer array, which points to each argument passed to the program. Following is a simple example, which checks if there is any argument supplied from the command line and take action accordingly:

```
#import <Foundation/Foundation.h>

int main( int argc, char *argv[] )
{
    if( argc == 2 )
    {
        NSLog(@"The argument supplied is %s\n", argv[1]);
    }
    else if( argc > 2 )
    {
        NSLog(@"Too many arguments supplied.\n");
    }
    else
    {
        NSLog(@"One argument expected.\n");
    }
}
```

When the above code is compiled and executed with a single argument, say "testing", it produces the following result.

```
2013-09-13 03:01:17.333 demo[7640] The argument supplied is testing
```

When the above code is compiled and executed with two arguments, say testing1 and testing2, it produces the following result.

```
2013-09-13 03:01:18.333 demo[7640] Too many arguments supplied.
```

When the above code is compiled and executed without passing any argument, it produces the following result.

```
2013-09-13 03:01:18.333 demo[7640] One argument expected
```

It should be noted that **argv[0]** holds the name of the program itself and **argv[1]** is a pointer to the first command-line argument supplied, and *argv[n] is the last argument. If no arguments are supplied, argc will be one, otherwise if you pass one argument, then **argc** is set at 2.

You pass all the command line arguments separated by a space, but if argument itself has a space, then you can pass such arguments by putting them inside double quotes "" or single quotes ". Let us re-write above example once again where we will print program name and we also pass a command-line argument by putting inside double quotes:

```
#import <Foundation/Foundation.h>

int main( int argc, char *argv[] )
{
    NSLog(@"Program name %s\n", argv[0]);

    if( argc == 2 )
```

```
{
  NSLog(@"The argument supplied is %s\n", argv[1]);
}
else if( argc > 2 )
{
  NSLog(@"Too many arguments supplied.\n");
}
else
{
  NSLog(@"One argument expected.\n");
}
return 0;
}
```

When the above code is compiled and executed with a single argument separated by space but inside double quotes say "Testing1 Testing2", it produces the following result.

```
2013-09-14 04:07:57.305 demo[8534] Program name demo
2013-09-14 04:07:57.305 demo[8534] The argument supplied is Testing1 Testing 2
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```