

## Re: What does const char \*s mean? is s a constant or \*s?

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2005-01/2602.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-01/2602.html)

---

**From:** Jonathan Burd ([jonathan.burd\\_at\\_REMOVEMEgmail.com](mailto:jonathan.burd_at_REMOVEMEgmail.com))

**Date:** 01/24/05

Date: Mon, 24 Jan 2005 14:44:15 +0530

G Patel wrote:

- > *I've been looking at some code for string functions (certain*
- > *implementation of them) and the non modified string is usually declared*
- > *as a const char \*s in the parameter list.*
- >
- > *I was wondering, what exactly does that mean. Does that mean the*
- > *function is restricted from changing the address (changing s) or*
- > *changing what's being pointed to (\*s)?*
- >

See what the compiler tells you when you try to compile it:  
(gcc -pedantic -Wall -std=c89 -o complain complain.c)

```
/* complain.c */
#include <stdio.h>

void
replace_spaces_with_hyphens (/* >> */ const char *, const char *);

void
replace_spaces_with_hyphens (/* >> */ const char *buf, const char *str)
{
    int ch;
    while ((ch = *str))
    {
        if (*str == ' ')
            ch = '-';
        *buf = ch;
        ++buf;
        ++str;
    }
    *buf = '\0';
}

int
main (int argc, char **argv)
```

comp.lang.c: Re: What does const char \*s mean? is s a constant or \*s?

```
{  
char buf[100];  
const char *str = "foo licious";  
  
replace_spaces_with_hyphens (buf, str);  
puts (buf);  
return 0;  
}
```

Regards,  
Jonathan.

--  
"Women should come with documentation." - Dave