

# C library - strcat() function

The C Library **strcat()** function accepts two pointer variable as parameters(say **dest**, **src**) and, appends the string pointed to by src to the end of the string pointed to by dest. This function only concatenates the string data type. We cannot use any other data types such int, float, char, etc.

## Syntax

Following is the syntax of the C library **strcat()** function –

```
char *strcat(char *dest_str, const char *src_str)
```

## Parameters

This function accepts the following parameters –

- **dest\_str** – This is a pointer to the destination array, which should contain a C string, and should be large enough to contain the concatenated resulting string.
- **src\_str** – This is the string to be appended. This should not overlap the destination.

Explore our **latest online courses** and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## Return Value

This function returns a pointer to the resulting string i.e. dest.

Note that, the function dynamically combine strings during runtime.

## Example 1

Following is the basic C library program that illustrates the code snippet on string concatenation using **strcat()** function.



</>

Open Compiler

```
#include <stdio.h>
#include <string.h>
int main() {
```

```

char dest_str[50] = "C ";
const char src_str[] = "JAVA";

// Concatenate src_str to dest_str
strcat(dest_str, src_str);

// The result store in destination string
printf("%s", dest_str);
return 0;
}

```

## Output

On executing the above code, we get the following output–

Final destination string : |This is destinationThis is source|

## Example 2

In the following example, we are using the **strcpy()** along with the **strcat()** function, to merge the new string ("point") with the existing string ("tutorials").

</>

Open Compiler

```

#include <stdio.h>
#include <string.h>
int main()
{
    char str[100];

    // Copy the first string to the variable
    strcpy(str, "Tutorials");

    // String concatenation
    strcat(str, "point");

    // Show the result
    printf("%s\n", str);
    return 0;
}

```



## Output

The above code produces the following output–

Tutorialspoint

### Example 3

While concatenating two different strings always requires two parameters, but here we are using integer values as parameters for `strcat()` that results in errors.

</>

Open Compiler

```
#include <stdio.h>
#include <string.h>
int main()
{
    int x = 20;
    int y = 24;
    strcat(x, y);
    printf("%d", x);
    return 0;
}
```

## Output

After executing the above code, we get the following result –

main.c: In function 'main':

main.c:7:12: warning: passing argument 1 of 'strcat' makes pointer from integer without

```
7 |   strcat(x, y);
  |         ^
  |         |
  |         int
```

In file included from main.c:2:

/usr/include/string.h:149:39: note: expected 'char \* restrict' but argument is of type 'int

```
149 | extern char *strcat (char *__restrict __dest, const char *__restrict __src)
    |                   ~~~~~^~~~~~
```

main.c:7:15: warning: passing argument 2 of 'strcat' makes pointer from integer without

```
7 |   strcat(x, y);
```

```
|      ^  
|      |  
|      int
```

In file included from main.c:2:

/usr/include/string.h:149:70: note: expected 'const char \* restrict' but argument is of type 'int'

149 | extern char \*strcat (char \*\_\_restrict \_\_dest, const char \*\_\_restrict \_\_src)

```
|
```

