Menu ▾ Log in

☰ HTML CSS

C Strings

❮ Previous Next ❯

Strings

Strings are used for storing text/characters.

For example, "Hello World" is a string of characters.

Unlike many other programming languages, C does not have a **String type** to easily create string variables. However, you can use the `char` type and create an array of characters to make a string in C:

```
char greetings[] = "Hello World!";
```

Note that you have to use double quotes.

To output the string, you can use the `printf()` function together with the format specifier `%s` to tell C that we are now working with strings:

Example

```
char greetings[] = "Hello World!";
printf("%s", greetings);
```

Try it Yourself »

Access Strings

Since strings are actually arrays in C, you can access a string by referring to its index number inside square brackets `[]`.

This example prints the **first character (0)** in **greetings**:

Example

```
char greetings[] = "Hello World!";
printf("%c", greetings[0]);
```

Try it Yourself »

Note that we have to use the `%c` format specifier to print a **single character**.

Modify Strings

To change the value of a specific character in a string, refer to the index number, and use **single quotes**:

Example

```
char greetings[] = "Hello World!";
greetings[0] = 'J';
printf("%s", greetings);
// Outputs Jello World! instead of Hello World!
```

Try it Yourself »

Another Way Of Creating Strings

In the examples above, we used a "string literal" to create a string variable. This is the easiest way to create a string in C.

You should also note that you can create a string with a set of characters. This example will produce the same result as the example in the beginning of this page:

Example

```
char greetings[] = {'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd',
'!', '\0'};
printf("%s", greetings);
```

Try it Yourself »

Why do we include the `\0` character at the end? This is known as the "null terminating character", and must be included when creating strings using this method. It tells C that this is the end of the string.

Differences

The difference between the two ways of creating strings, is that the first method is easier to write, and you do not have to include the `\0` character, as C will do it for you.

You should note that the size of both arrays is the same: They both have **13 characters** (space also counts as a character by the way), including the `\0` character:

Example

```
char greetings[] = {'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd',
'!', '\0'};
char greetings2[] = "Hello World!";

printf("%lu\n", sizeof(greetings));   // Outputs 13
printf("%lu\n", sizeof(greetings2));  // Outputs 13
```

Try it Yourself »

C Exercises

Test Yourself With Exercises

Exercise:

Fill in the missing part to create a "string" named **greetings**, and assign it the value "Hello".

```
                    =          ;
```

Submit Answer »

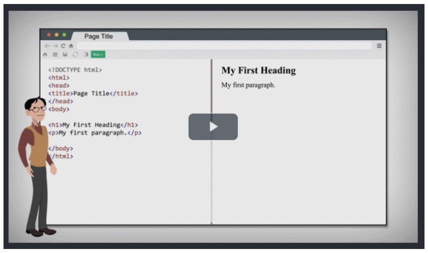
Start the Exercise

❮ Previous

Next ❯

NEW

We just launched
W3Schools videos



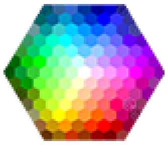
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

My First Heading

My first paragraph.

Explore now

COLOR PICKER



Get certified
by completing
a course today!

Report Error

Spaces

Pro

Get Certified

Top Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

Top References

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples

XML Examples
jQuery Examples

Get Certified

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate

FORUM | ABOUT

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

W3Schools is Powered by W3.CSS.