

[HOME](#) [JAVA](#) [PHP](#) [Python](#) [Ruby](#) [Perl](#) [HTML](#) [CSS](#) [Javascript](#) [MySQL](#) [C++](#) [UNIX](#) [MORE...](#)[REFERENCES](#) | [FORUM](#) | [ABOUT](#) | [CONTACT](#)

The C Standard Library

[C Library - Home](#)[C Library - <assert.h>](#)[C Library - <ctype.h>](#)[C Library - <errno.h>](#)[C Library - <float.h>](#)[C Library - <limits.h>](#)[C Library - <locale.h>](#)[C Library - <math.h>](#)[C Library - <setjmp.h>](#)[C Library - <signal.h>](#)[C Library - <stdarg.h>](#)[C Library - <stddef.h>](#)[C Library - <stdio.h>](#)[C Library - <stdlib.h>](#)[C Library - <string.h>](#)[C Library - <time.h>](#)

C library function - fread()

Advertisements

[Previous Page](#)[Next Page](#)

Description

The C library function **size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)** reads data from the given **stream** into the array pointed to by **ptr**.

Declaration

Following is the declaration for fread() function.

```
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)
```

Parameters

ptr -- This is the pointer to a block of memory with a minimum size of *size*nmemb* bytes.

size -- This is the size in bytes of each element to be read.

nmemb -- This is the number of elements, each one with a size of size bytes.

stream -- This is the pointer to a FILE object that specifies an input stream.

Return Value

The total number of elements successfully read is returned as a **size_t** object, which is an integral data type. If this number differs from the **nmemb** parameter, either an error occurred or the End Of File was reached.

[g+1](#)[Modern Baby Names](#)[Online Photo Editing](#)

Advertisements

C Programming Resources

[C - Programming Tutorial](#)[C - Useful Resources](#)

Selected Reading

[Developer's Best Practices](#)[Effective Resume Writing](#)[Computer Glossary](#)[Who is Who](#)

Example

The following example shows the usage of `fread()` function.

```
#include <stdio.h>
#include <string.h>

int main()
{
    FILE *fp;
    char c[] = "this is tutorialspoint";
    char buffer[20];

    /* Open file for both reading and writing */
    fp = fopen("file.txt", "w+");

    /* Write data to the file */
    fwrite(c, strlen(c) + 1, 1, fp);

    /* Seek to the beginning of the file */
    fseek(fp, SEEK_SET, 0);

    /* Read and display data */
    fread(buffer, strlen(c)+1, 1, fp);
    printf("%s\n", buffer);
    fclose(fp);

    return(0);
}
```

Let us compile and run the above program, this will create a file **file.txt** and write a content *this is tutorialspoint*. Next we use **fseek()** function to reset writing pointer to the beginning of the file and ready the file content which is as follows:

```
this is tutorialspoint
```

[Previous Page](#)[Print Version](#)[PDF Version](#)[Next Page](#)

Advertisements

JAVA FILE APIs

Convert
Print
Create
Modify
Combine



Try for FREE

ASPOSE
Your File Format APIs

<oXygen/> XML Editor

 oxygenxml.com

Validate and Execute XQuery
support over eXist Database. Easy
to use.

[ASP.NET](#) | [jQuery](#) | [AJAX](#) | [ANT](#) | [JSP](#) | [Servlets](#) | [log4j](#) | [iBATIS](#) | [Hibernate](#) | [JDBC](#) | [Struts](#) | [HTML5](#) | [SQL](#) | [MySQL](#) | [C++](#) | [UNIX](#)

Copyright © 2014 by tutorialspoint. All Rights Reserved.

