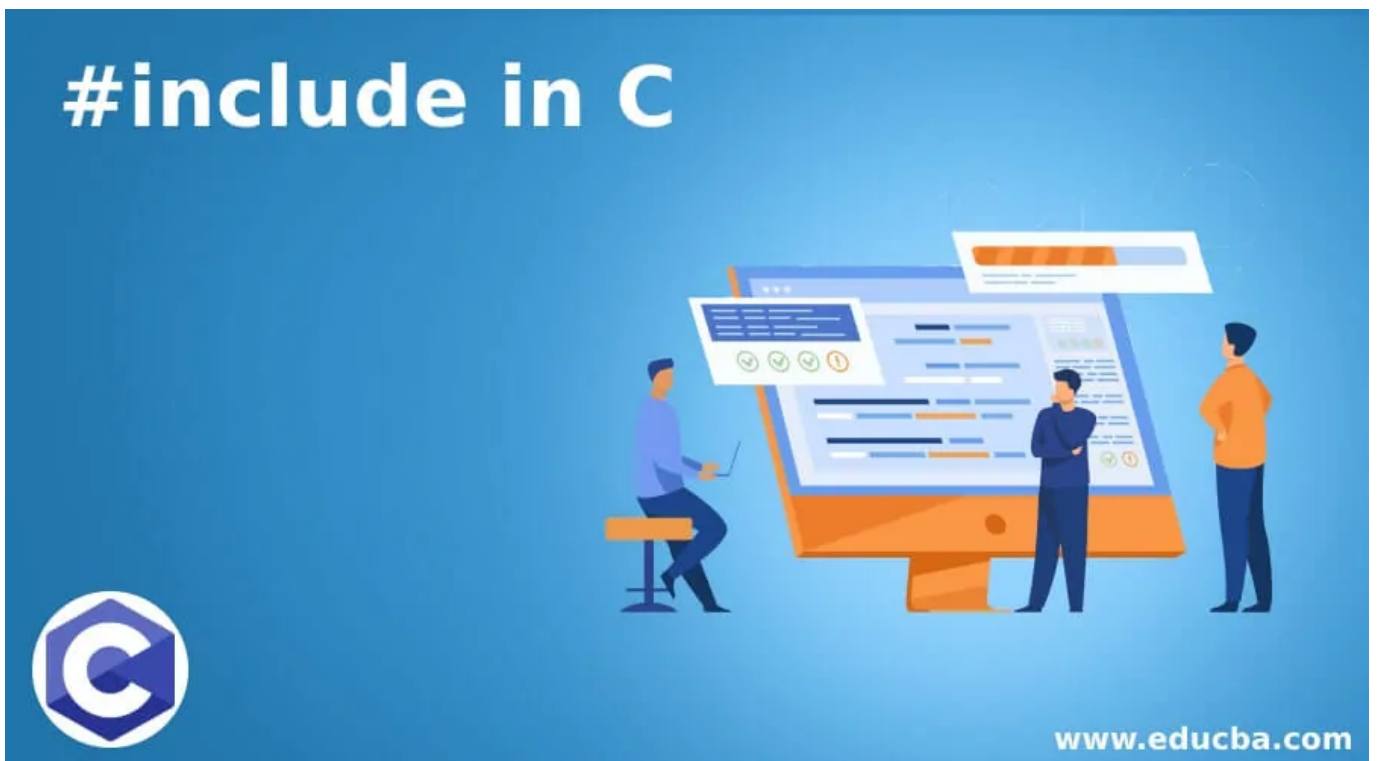




[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

← [\(https://www.educba.com/sharp-undef-in-c/\)](https://www.educba.com/sharp-undef-in-c/)

→ [\(https://www.educba.com/macros-in-c/\)](https://www.educba.com/macros-in-c/)



Introduction to #include in C

If you have ever worked or studied the C language, you must have seen that the very first line of the code starts with #include directive. Let us see what is #include and why is it used before writing any code in C language. #include is basically a preprocessor directive (as it is read by





(<https://www.educba.com/software-development/>)

language, different files either user defined or system files performing having specific functions or operations are kept separately for the easy manageable organization and accessibility and are included in the respective file once required. In C, these files are imported in the code using #include directive by the compiler and are mentioned in the starting of the code.

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

Syntax of #include in C

Header files included using the #include directive can be system files or the user defined files.

- **System files are standard files:** These files basically contain the declaration of functions and macros definition that is shared between various source files of C. These files are by default present in the C library (there is no need to create them).
- **User-defined files:** These files are almost similar to the system or standard files with the only difference being that they are written by the user in order to reduce the repetition or the proper management of code.

Before using the #include directive we need to provide the information to the preprocessor from where to look for the respective header file.

Below given is the basic syntax of using the C #include directive for including both type in the code:



1. #include <filename> While including the file using <>, the preprocessor will search the



(<https://www.educba.com/software-development/>)



C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,644 ratings)

Course Price

\$79 ~~\$399~~

[View Course](https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

Related Courses

[C++ Training \(4 Courses, 5 Projects, 4 Quizzes\)](https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

[Java Training \(40 Courses, 29 Projects, 4 Quizzes\)](https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

```
/* Including the system file */
#include <stdio.h>

void main()

{
/* C code to be written here */
}
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
/* Including the user defined file */  
#include "firstfile.h"  
  
void main()  
{  
/* C code to be written here */  
}
```

How #include Directive works in C?

There are two types of header files and two ways of including these files using the #include directive in any C program. Header files in C have an extension ".h". In C programs, the main motive is to direct the preprocessor to the specified header file in order to include its content (or the code written in that file). It works like a simple copy pasting the code of one file to the other file (but it is not done to keep the code manageable and easily accessible) and processing the code of the current (source) file. C allows the nesting of file inclusions using the #include directive.

C uses the above two syntax in order to include the header files in the source code. #include directs the preprocessor to look for the respective file and if there is an incomplete path inside the double quotes, it first looks for the file in the current source file only then in the standard folder. Preprocessor stops once the given file name matches the file which it is looking for. If not found in any case (wrong path or file name) in a local or standard folder, an error is thrown on the console. Comments inside the #include are not considered as it treats it as a normal text and starts looking with that respective file name.

Examples of #include in C

Given below are the examples mentioned :





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
// Inclusion of standard 'stdio.h' file
#include <stdio.h>

void main()
{
    printf("Hello we are learning the #include preprocessor
    directive");
}
```

Output:

Explanation:

- In the above code we are including the 'stdio.h' header file in our code. Preprocessor will first search for the system file 'stdio.h' in the standard directory of C header files and once found will include all the code of that file in the current source file before moving forward to the code of print statement in the above code.
- printf() is the predefined function which is present in the stdio.h header file, so there would be no error in the execution of the above program. If we will not include the above 'stdio.h' file, the compiler would throw an error of the missing function definition.

Example #2

Inclusion of user defined file using the #include " ".





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
void printing()
{
    printf("\n hello I am in new_file header file");
}

void working()
{
    printf("\n hello I used to work in header file");
}
```

File: header_learning.c

Code:

```
// including the system file using <> brackets in order to use
printf function
#include <stdio.h>

// including the user defined header file created above
#include "new_file.h"

void main()
{
    // calling of functions of header file
    printing();

    working();
    printf("\n hello I am in main/current file");
}
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Explanation:

- In the above code, we created one header file with the name 'new_file.h' having some functions in it. We have created a main 'header_learning.c' file including the above header file using the "#". On compiling the above program, the preprocessor will first look for the 'stdio.h' file in the standard directory of headers and then 'new_file.h' header file in the current directory.
- If we need to include the stdio.h using "#include" in the program, we need to ensure that this header file needs to be present in the current directory. Code of both the header files will get included in the main file so on calling both the functions (printing and working) of the header file (new_file.h) and printf function of header file (stdio.h), the program will execute properly without any issue.

Conclusion

Above description clearly explains the #include directive in C and how it is used in programming. Before learning of how to program in C language, the first line of code starts with #include directive. Most of us use it but do not actually know why it is used. Before moving forward it is very important to know every key point in order to have in-depth knowledge. There are some header files including the basic functions which are normally used in every program and hence imported every time.

Recommended Articles

This is a guide to #include in C. Here we discuss how #include directive works in C along with respective programming examples. You may also have a look at the following articles to learn





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

3. [Types or Errors in C \(https://www.educba.com/types-or-errors-in-c/\)](https://www.educba.com/types-or-errors-in-c/)
4. [Function Prototype in C \(https://www.educba.com/function-prototype-in-c/\)](https://www.educba.com/function-prototype-in-c/)

ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE (600+ COURSES, 50+ PROJECTS)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

Learn More

<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>

About Us

Blog (<https://www.educba.com/blog/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

source=footer)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

Certification Courses





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

[development/courses/python-certification-course/?source=footer\)](https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course ([https://www.educba.com/software-development/courses/java-course/?source=footer\)](https://www.educba.com/software-development/courses/java-course/?source=footer)

Become a Selenium Automation Tester ([https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer\)](https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer)

Become an IoT Developer ([https://www.educba.com/software-development/courses/iot-course/?source=footer\)](https://www.educba.com/software-development/courses/iot-course/?source=footer)

ASP.NET Course ([https://www.educba.com/software-development/courses/asp-net-course/?source=footer\)](https://www.educba.com/software-development/courses/asp-net-course/?source=footer)

VB.NET Course ([https://www.educba.com/software-development/courses/vb-net-course/?source=footer\)](https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course ([https://www.educba.com/software-development/courses/php-course/?source=footer\)](https://www.educba.com/software-development/courses/php-course/?source=footer)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

