

# Preprocessor directive in C #

Preprocessor Directive directives provide instructions to the compiler to preprocess the information before the actual compilation process begins. All preprocessor directives in C # start with #, and only white-space characters can appear before a preprocessor directive in a line. Preprocessor directives in C # are not commands, so they do not end with a semicolon (;).

Preprocessor Directive directives provide instructions to the compiler to preprocess the information before the actual compilation process begins.

All preprocessor directives in C # start with #, and only white-space characters can appear before a preprocessor directive in a line. Preprocessor directives in C # are not commands, so they do not end with a semicolon (;).

The C # compiler does not have a separate preprocessor, however, these instructions are handled as if there was actually a separate preprocessor. In C #, preprocessor directives are used to help conditional compilation. Unlike preprocessor directives in C and C ++, they are not used to create macros. A mandatory preprocessor directive is a single instruction on a line.

## Preprocessor directives in C #

Below is a list of preprocessing directives available in C #:

Preprocessor directive #define It defines a sequence of characters, called symbols. #undef It allows you to not define (undefine) an icon. #if It allows checking whether an icon or multiple icons is true. #else It allows creating a conditional directive, along with #if. #elif It allows creating a conditionally complex directive. #endif Identify the end of a conditional directive. #line It allows you to modify the line number of the compiler and (optionally) the file name for Error and Warning. #error It allows to create an error from a specific location in your code. #warning It allows creating a warning level from a specific location in your code. #region It allows you to define a code block that you can expand or collapse using the Visual Studio Code Editor feature. #endregion It marks the end of a #region block.

## #Define preprocessor directive in C #

#Define preprocessor directive in C # creates symbol constants.

#define allows you to create an icon so that, when using an expression symbol passed to #if preprocessor directive, the expression evaluates to true. Its syntax is as follows:

```
define symbol
```

The following example illustrates this:

```
#define PI using System ; namespace PreprocessorDApp1 { class QTM { static void
?ã ???c ??nh ngh?a" ); #else Console . WriteLine ( "PI ch?a ???c ??nh ngh?
a" ); #endif Console . ReadKey (); } } }
```

When running the C # program, we have the following results:

```
PI has been defined
```

## Conditional Directive in C #

You can use the #if preprocessor directive in C # to create a Conditional Directive. Conditional directives are useful when checking an icon or icons to check if they estimate to true. If they evaluate to true, the compiler estimates all code between the #if directive and the next directive.

Syntax for conditional directives in C # is:

```
if symbol [operator ký hi?u] .
```

Here, **symbol** is the name of the symbol you want to check. You can also use true and false or append the following symbol to the negative operator.

**operator symbol** is the **operator** used to estimate the symbol. Operators can be one of:

1. == (equals)
2. != (not equal)
3. && (and)
4. || (or)

You can also group symbols and operators by parentheses. Conditional directives are used to compile code for debugging or when compiling for a specific configuration. A conditional directive in C # starts with a #if preprocessor directive that must be explicitly terminated by a #endif directive.

The following example illustrates the use of conditional directives in C #:

```
#define DEBUG #define VC_V10 using System ; public class Tester { public sta
?ã ???c ??nh ngh?
a" ); #elif (! DEBUG && VC_V10 ) Console . WriteLine ( "VC_V10 ?ã ???c ??
nh ngh?
a" ); #elif ( DEBUG && VC_V10 ) Console . WriteLine ( "DEBUG và VC_V10 ?
ã ???c ??nh ngh?a" ); #else Console . WriteLine ( "DEBUG và VC_V10 ?ã ???c
??nh ngh?a" ); #endif Console . ReadKey (); } }
```

Running the C # code above gets the following result:

```
DEBUG and VC_V10 have been defined
```

### According to Tutorialspoint

Previous article: Namespace in C #

Next lesson: Regular Expression in C #

You finished reading the article "**Preprocessor directive in C #**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

---