

Übersicht

#include	Fügt Text aus einer anderen Quelltextdatei ein.
#define	Definiert ein Makro
#undef	Entfernt ein Makro
#if	Bedingte Compilierung in Abhängigkeit der nachstehenden Bedingung
#elif ²	Alternative Compilierung in Abhängigkeit der nachstehenden Bedingung, wenn die Bedingung des vorstehenden #if, #elif, #ifdef oder #ifndef nicht erfüllt ist. (Also wie else if).
defined ¹	Liefert eine 1, wenn der nachstehende Makroname definiert ist, sonst eine 0. <i>Kann nur zusammen mit #if oder #elif verwendet werden.</i>
#ifdef	Bedingte Compilierung in Abhängigkeit, ob ein Makroname definiert ist.
#ifndef	Bedingte Compilierung in Abhängigkeit, ob ein Makroname nicht definiert ist.
#else	Alternative Compilierung, wenn die Bedingung des vorstehenden #if, #elif, #ifdef oder #ifndef nicht erfüllt ist
#endif	Beendet den Block mit der bedingten Compilierung
#line	Liefert eine Zeilennummer für Compilermeldungen
#operator ¹	Ersetzt innerhalb eines Makros einen Makroparameter durch eine konstante Zeichenkette, die den Wert des Parameters enthält.
## operator ¹	Erzeugt ein einzelnes Token aus zwei hintereinander stehenden Tokens.
#pragma ¹	Gibt Compiler- und Systemabhängige Informationen an den Compiler
#error ^{1,3}	Erzeugt einen Fehler während der Compilierung mit der angegebenen Fehlermeldung.

¹ Neu in C99.

² Gehört nicht zum Standard-C, obwohl es von vielen Compilern unterstützt wird.

³#pragma Befehle können nur entweder in der obersten Ebene **vor der ersten Deklaration** oder innerhalb eines Blockes **vor der ersten Deklaration** eingesetzt werden!

Vordefinierte Makros

__LINE__	Zeilennummer innerhalb der aktuellen Quellcodedatei
__FILE__	Name der aktuellen Quellcodedatei
__DATE__	Datum, wann das Programm kompiliert wurde (als Zeichenkette)
__TIME__	Uhrzeit, wann das Programm kompiliert wurde (als Zeichenkette)
__STDC__	Liefert eine 1, wenn sich der Compiler nach Standard-C richtet
__STDC_VERSION__	Liefert die Zahl 199409L, wenn sich der Compiler nach C95 richtet 199901L, wenn sich der Compiler nach C99 richtet sonst ist dieses Makro nicht definiert

Betriebssystemabhängige vordefinierte Makros

__unix__	UNIX System
__unix	UNIX System
__WIN32__	MS Windows ab 95
__Windows	Zielsystem MS Windows
__linux__	Linux-System
__FreeBSD__	FreeBSD
__OpenBSD__	OpenBSD

Einfügen und Definieren (Textersetzungen)

#include	#include <stdio.h> //Aus Includeverzeichnis #include "projekt.h" //Aus akt. Verzeichnis
#define	#define PI 3.14159265 int main() { printf("PI = %f\n", PI); return 0 } //Ersetzt das PI im Quelltext durch 3.14159265
#undef	#undef PI
# operator	#define Makro(a) printf(#a) //ersetzt bspw. Makro(7); //durch: printf("7");

Einfügen und Definieren (Textersetzungen) (cont)

```
## Operator      #define Makro(i) temp ## i
                  int temp1, temp2;
                  Makro(1) = 5;
                  //Wird zu
                  temp1 = 5;
```

Bedingte Kompilierung

```
#if      #if defined PI
          //Kompiliert diesen Bereich nur, wenn PI
          definiert wurde.
          #endif
          #if _Windows
          //Kompiliert diesen Bereich nur in Windows.
          #endif
```

```
#ifndef  #ifndef projekt_h
          //Kompiliert diesen Bereich nur, wenn projekt_h
          noch nicht definiert wurde
          #endif
```

```
#ifdef  #ifdef projekt_h
          //Kompiliert diesen Bereich nur, wenn projekt_h
          definiert wurde
          #endif
```

```
#else    #ifdef DEBUG
          //Kompiliert diesen Bereich nur, wenn DEBUG
          definiert wurde
          #else
          //Ansonsten wird dieser Bereich kompiliert
          #endif
```

#endif muss am Ende von jedem Bedingungsblock stehen!

Compilerbezogene Befehle

```
#line      #line 12345 "Neuer_Dateiname.c"
            printf("Dieser Text steht in Zeile %d in Datei
            %s. \n", __LINE__, __FILE__);
            //Gibt "Dieser Text steht in Zeile 12345 in Datei
            Neuer_Dateiname.c" aus.
            Egal in welcher Datei und welcher Zeile der Code wirklich
            steht.
```

```
#pragma    #pragma FENV_ACCESS ON
            //Weist den Compiler an, beim Kompilieren Überwachungen &
            Exceptions rund um Fließkomma-Arithmetik einzubauen.
```

Compilerbezogene Befehle (cont)

```
#error    #ifndef Labelname
            #error Labelname ist nicht definiert!
            #endif
            //Gibt einen Compilerfehler aus, wenn Labelname nicht definiert
            wurde.
```

#pragma Befehle können nur entweder in der obersten Ebene **vor der ersten Deklaration** oder innerhalb eines Blockes **vor der ersten Deklaration** eingesetzt werden!