

Übersicht

<code>#include</code>	Fügt Text aus einer anderen Quelltextdatei ein.
<code>#define</code>	Definiert ein Makro
<code>#undef</code>	Entfernt ein Makro
<code>#if</code>	Bedingte Compilierung in Abhängigkeit der nachstehenden Bedingung
<code>#elif²</code>	Alternative Compilierung in Abhängigkeit der nachstehenden Bedingung, wenn die Bedingung des vorstehenden <code>#if</code> , <code>#elif</code> , <code>#ifdef</code> oder <code>#ifndef</code> nicht erfüllt ist. (Also wie else if).
<code>defined¹</code>	Liefert eine 1, wenn der nachstehende Makroname definiert ist, sonst eine 0. <i>Kann nur zusammen mit <code>#if</code> oder <code>#elif</code> verwendet werden.</i>
<code>#ifdef</code>	Bedingte Compilierung in Abhängigkeit, ob ein Makroname definiert ist.
<code>#ifndef</code>	Bedingte Compilierung in Abhängigkeit, ob ein Makroname nicht definiert ist.
<code>#else</code>	Alternative Compilierung, wenn die Bedingung des vorstehenden <code>#if</code> , <code>#elif</code> , <code>#ifdef</code> oder <code>#ifndef</code> nicht erfüllt ist
<code>#endif</code>	Beendet den Block mit der bedingten Compilierung
<code>#line</code>	Liefert eine Zeilennummer für Compilermeldungen
<code>#operator¹</code>	Ersetzt innerhalb eines Makros einen Makroparameter durch eine konstante Zeichenkette, die den Wert des Parameters enthält.
<code>##</code>	Erzeugt ein einzelnes Token aus zwei hintereinander stehenden Tokens.
<code>#pragma¹</code>	Gibt Compiler- und Systemabhängige Informationen an den Compiler
<code>#error^{1,3}</code>	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

<code>__LINE__</code>	Zeilennummer innerhalb der aktuellen Quellcode-datei
<code>__FILE__</code>	Name der aktuellen Quellcode-datei
<code>__DATE__</code>	Datum, wann das Programm kompiliert wurde (als Zeichenkette)
<code>__TIME__</code>	Uhrzeit, wann das Programm kompiliert wurde (als Zeichenkette)
<code>__STDC__</code>	Liefert eine 1, wenn sich der Compiler nach Standard-C richtet
<code>__STDC_VERSION__</code>	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

<code>__unix__</code>	UNIX System
<code>__unix</code>	UNIX System
<code>__WIN32__</code>	MS Windows ab 95
<code>_Windows</code>	Zielsystem MS Windows
<code>__linux__</code>	Linux-System
<code>__FreeBSD__</code>	FreeBSD
<code>__OpenBSD__</code>	OpenBSD

Einfügen und Definieren (Textersetzungen)

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

Einfügen und Definieren (Textersetzungen) (cont)

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

Compilerbezogene Befehle (cont)

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

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.
```

