

HowTo use Cmd/Konsole/Terminal

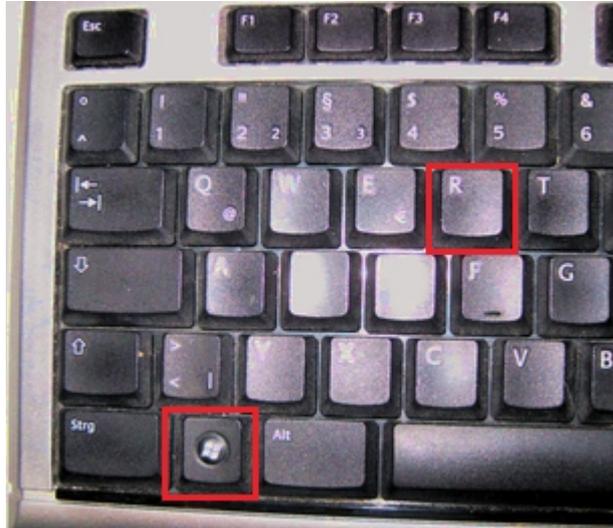
Die Konsole ist eine Möglichkeit um mit dem Betriebssystem in Interaktion zu treten. Weiterführend: <https://de.wikipedia.org/wiki/Kommandozeile> Damit das Themengebiet nicht ausufernd wird, wird sich hier auf die grundlegendsten Befehle beschränkt.

Übersicht

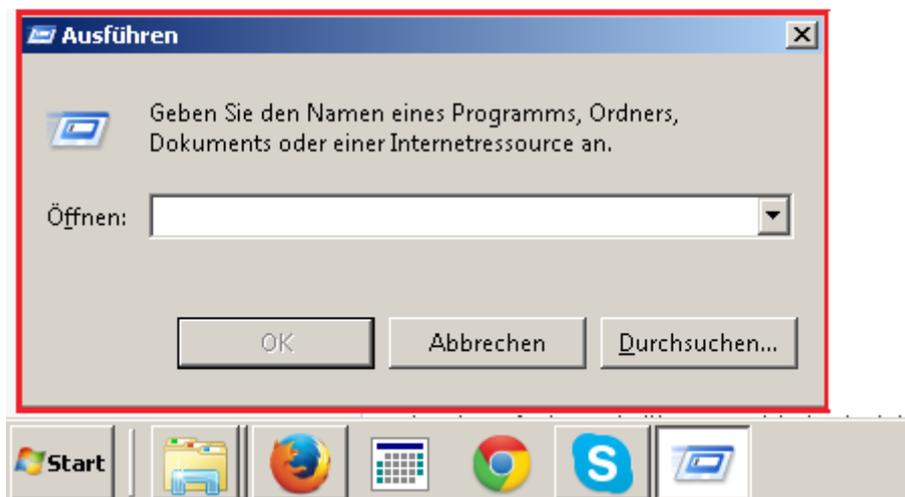
- Windows Systeme (XP, 7, 8, 10, NT)
 - Allgemeines zu Windows/Fenster (gilt für Windows, Linux)
- Linux (allgemein)
- Mac OS X
- Tablet, E-Reader, SmartDevices

Windows Systeme (XP, 7, 8, 10, NT)

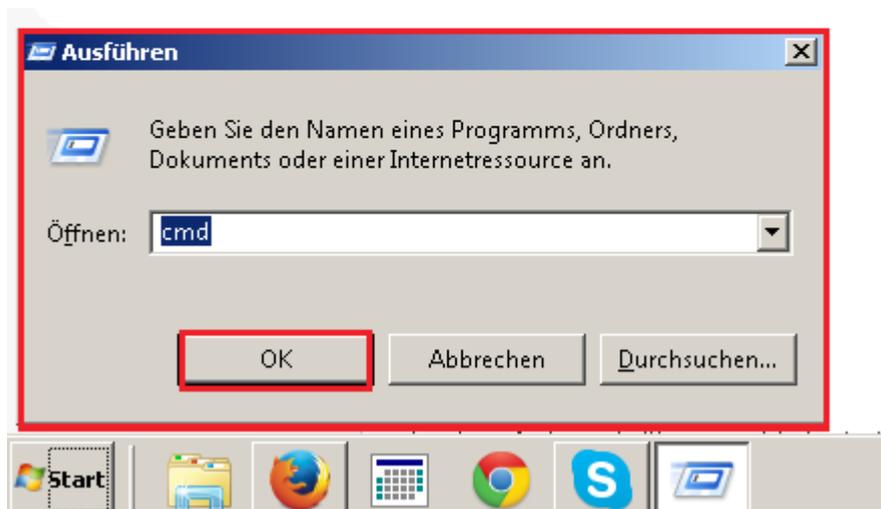
[Windows-Taste] + [R-Taste] drücken



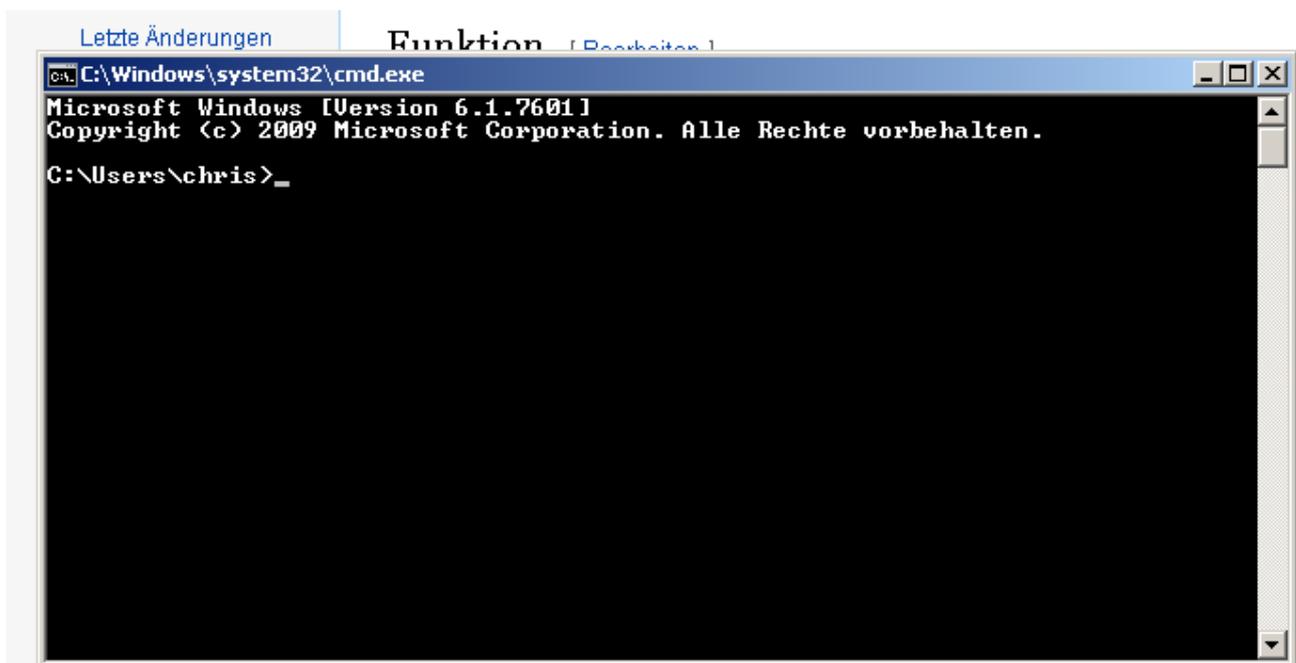
-> Das Ausführen-Fenster erscheint



cmd eingeben [Enter-Taste] drücken



-> eine Console öffnet



-> Befehle können eingetippt werden

Alternativ: Start -> Alle Programme -> Zubehör -> Eingabeaufforderung

Ein paar Befehle

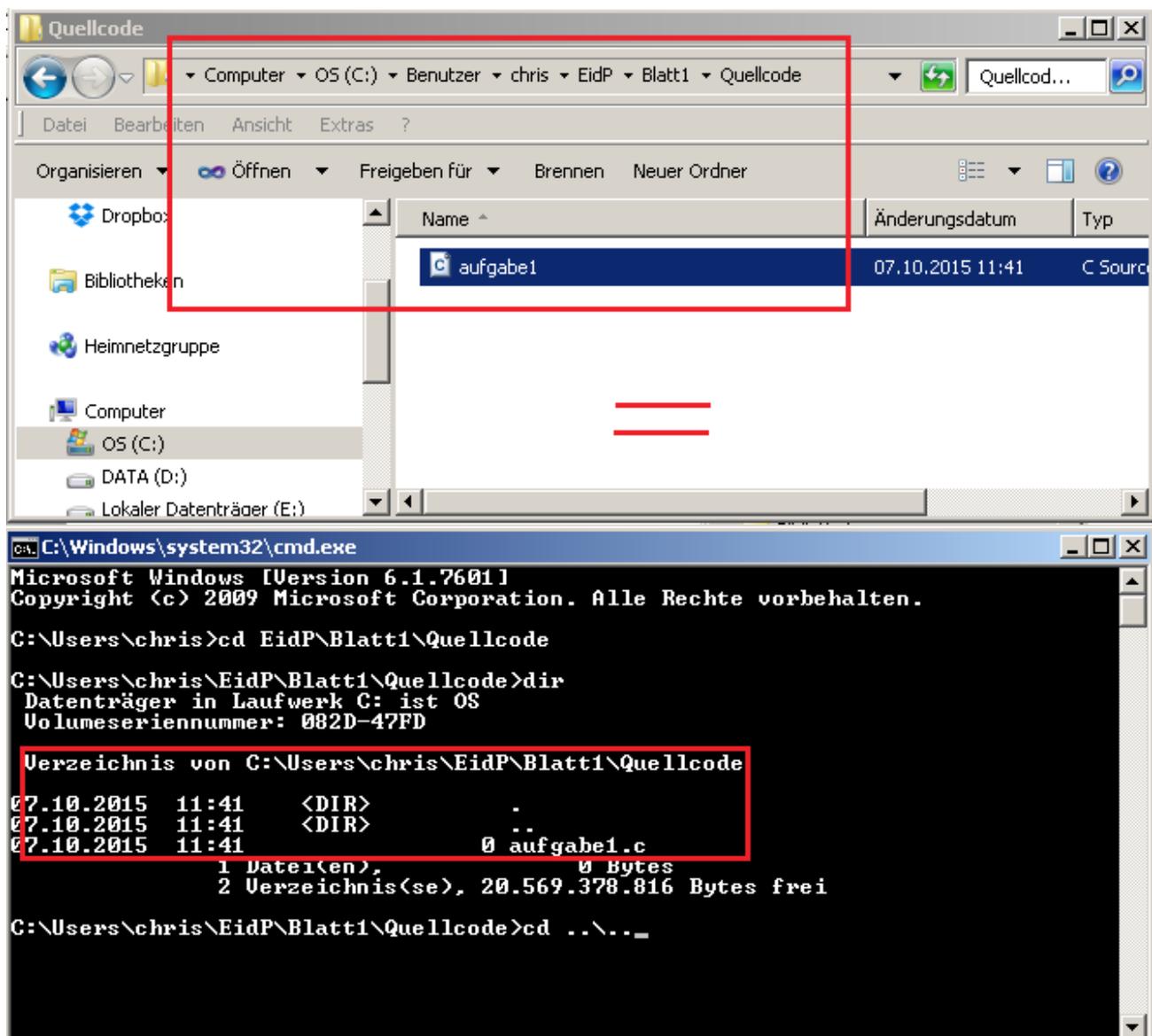
dir – zeigt das aktuelle Verzeichnis an

mkdir – erstellt ein Verzeichnis (**mkdir Blatt1**)
cd – wechselt zwischen Verzeichnissen (**cd Blatt1**)
cd .. - wechselt in das nächst höhere Verzeichnis

konkretes Beispiel:

mkdir EidP
cd EidP
mkdir Blatt1
cd Blatt1
mkdir Quellcode
cd ..\..
dir EidP\Blatt1
cd EidP\Blatt1\Quellcode – wechselt zum Beispiel direkt in einen Ordner

Ergebnis: Konsolen Ansicht und GUI Ansicht zeigen beide dasselbe. (Das Betriebssystem verwaltet nur die Ordner in englischer Sprache, für die GUI werden diese übersetzt.)



-> Änderungen (kopieren, löschen, erstellen) in GUI oder Konsole sind bei beiden zu "sehen".

Der gcc Compiler

gcc Quellcode-Datei – erstellt eine a.exe

gcc Quellcode-Datei -o Ausgabe-Name - erstellt eine Ausgabe-Name.exe

konkretes Beispiel:

gcc A1.c -o aufgabe1

-> erstellt eine aufgabe1.exe aus einer vorhandenen A1.c Datei

Optionen des gcc Compilers (ausgewählte Optionen)

(Weitere Optionen siehe Ende des Dokuments oder klick auf den Link: Optionen des gcc Compilers (ausgewählte Optionen))

Ein wichtiges Beispiel zu gcc

gcc A1.c -o aufgabe1 -lm

Hierbei wird die **math.h** Bibliothek geladen, dies ist besonders bei **Erstinstallation von gcc** oder beim erstmaligen Benutzen von **#include <math.h>** empfohlen.

Hilfe zu Befehlen in cmd allgemein

Für jeden cmd Befehl gibt es die Option für Hilfe: befehl /?

konkretes Beispiel:

cd /?

Befehle auflisten: help

Allgemeines zu Windows/Fenster (gilt für Windows, Linux)

- [Windows-Taste] + [Pfeiltaste] = passt das aktuell ausgewählte Fenster auf dem aktuell befindlichen Bildschirm an (Hälfte, Vollbild, ...)
- Tabulator-Taste, [Tab] vervollständig Ordner oder Dateinamen, dies erspart sehr viel Tipp-Arbeit und eine Überprüfung der Dateinamen.



-> Autovervollständigung: Anfangsbuchstaben eines Wortes tippen [Tab] drücken

konkretes Beispiel:

cd Ei [Tab] drücken -> cd EidP erscheint, vorausgesetzt Sie haben das Vorangegangene getestet

Linux (allgemein)

Terminal/Bash öffnen: häufig [Strg] + [Alt] + [T] drücken

sehr häufig [Strg] + [Alt] + [F1] ... F2, ... F5 ... F7

[Alt] + [F2] -> Terminal -> [Enter]

-> Terminal erscheint

Ein paar Befehle

ls	– zeigt das aktuelle Verzeichnis an
mkdir	– erstellt ein Verzeichnis (mkdir Blatt1)
cd	– wechselt zwischen Verzeichnissen (cd Blatt1)
cd ..	- wechselt in das nächst höhere Verzeichnis
pwd	– zeigt den gesamten Pfad, indem man sich befindet an

konkretes Beispiel:

mkdir EidP

cd EidP

mkdir Blatt1

cd Blatt1

mkdir Quellcode

cd ../..

ls EidP/Blatt1

cd EidP/Blatt1/Quellcode – wechselt zum Beispiel direkt in einen Ordner

Der gcc Compiler

gcc Quellcode-Datei – erstellt eine .a.out Datei,

gcc Quellcode-Datei **-o** Ausgabe-Name - erstellt eine .Ausgabe-Name

konkretes Beispiel:

gcc A1.c -o aufgabe1

-> erstellt eine .aufgabe1 aus einer vorhandenen A1.c Datei

in Terminal: ./aufgabe1 um sie auszuführen, bei zugriffsproblemen siehe `chmod +x aufgabe1`

[weiterführend](#)

Optionen des gcc Compilers (ausgewählte Optionen) (siehe letzte Seite des Dokuments)

Ein wichtiges Beispiel zu gcc

gcc A1.c -o aufgabe1 **-lm**

Hierbei wird die **math.h** Bibliothek geladen, dies ist besonders bei **Erstinstallation von gcc** oder beim erstmaligen Benutzen von **#include <math.h>** empfohlen.

konkretes Beispiel zu gcc Compiler Optionen:

gcc datei.c -o datei.c **-Wall**

Durch die Option **-Wall** weist der Compiler den Programmierer auf "Fehler" oder "schlechten" Quellcode hin. Gerade für Programmier-Einsteiger ist **-Wall** hilfreich für die Fehlersuche.

Hilfe zu Befehlen

Für jeden cmd Befehl gibt es die Option für Hilfe: befehl - -help, oder man befehl

konkretes Beispiel:

cd - -help

oder

man cd

Befehle auflisten: help

[manpages online](#)

Wahl des Editors

Unter Linux-Systemen ist es Geschmackssache welchen Editor man verwendet. Der Autor empfiehlt nano, vi oder vim ist allerdings auf Konsolenebene auch stark verbreitet. Als grafischen Editor empfiehlt der Autor sublime.

Mac OS X

[**cmd**-Taste] + [**Leertaste**] -> Terminal eintippen -> [**Enter**-Taste]

oder [strg] + [alt] + [t]

Alternativ

Finder -> Terminal suchen -> Konsole/Terminal öffnen

Rest: siehe Linux

für die gcc Installation:

im Terminal **gcc** eingeben und [**Enter**-Taste] -> bestätigen Download/Installation zum Programmieren kann X-Code oder ein anderer Text-Editor verwendet werden.

Tablet, E-Reader, SmartDevices

- Es ist dem Autor bekannt das es *gcc Compiler* für Android-, iPad-Tablet, Surface oder auch für SmartDevices gibt. Soweit dem Autor bekannt ist, ist dieser gcc Compiler nach einigen

Malen des compilierens kostenpflichtig! Ausserdem rät der Autor dieses Dokuments davon ab, gcc auf so einem Gerät zu benutzen.

Optionen des gcc Compilers (ausgewählte Optionen)

Parameter	Erklärung
-o	Bestimmt den Namen der Ausgabedatei (Standard: „a.out“)
-c	Der Quellcode wird nur kompiliert, aber noch nicht gelinkt (neue Endung oft „.o“)
-Ox	Optimiert den Quellcode, der Grad der Optimierung (Platzhalter X) kann 0 (nichts), 1 (Größe und Geschwindigkeit), 2 (Geschwindigkeit), 3 (Geschwindigkeit, aggressiv) oder s (Größe) sein)
-march=nativ	Optimiert speziell für den vorhandenen Prozessor und wird auf anderen Architekturen nicht laufen. Ist für distcc nicht geeignet
-lXXX	Bindet weitere Bibliotheken ein (z.B. -lcurses für libcurses)
-std=XXX	Legt den Sprachstandard fest (XXX kann z.B. c99, c1x, c11, gnu++98, c++98, c++0x oder c++11 sein)
-ansi	Erlaubt nur ANSI C-konforme Konstrukte
-Wall	Aktiviert sinnvolle Warnungen, die vom gcc-Compiler unterstützt werden.
-Wextra	Aktiviert noch mehr Warnungen, die vom gcc-Compiler unterstützt werden.
-pedantic	Weist auf nicht ISO konforme mit einer Warnung hin (-pedantic-errors erzwingt Fehler)
-v	Gibt die Kommandos und Schritte aus, die der Compiler gerade ausführt (Präprozessor, Compiler, Assembler, Linker...)
-g	Fügt Debugging-Symbole in die Binärdatei ein
-ggdb	Fügt weitere Debugging-Symbole für den GDB hinzu. Mit -ggdb3 wird alles mögliche unterstützt.

Weitere: <https://gcc.gnu.org/onlinedocs/gcc/Option-Summary.html>

konkretes Beispiel:

```
gcc datei.c -o datei.c -Wall
```

Durch die Option **-Wall** weist der Compiler den Programmierer auf "Fehler" oder "schlechten" Quellcode hin. Gerade für Programmier-Einsteiger ist **-Wall** hilfreich für die Fehlersuche.

Ein wichtiges Beispiel zu gcc

```
gcc A1.c -o aufgabe1 -lm
```

Hierbei wird die **math.h** Bibliothek geladen, dies ist besonders bei **Erstinstallation von gcc** oder beim erstmaligen Benutzen von **#include <math.h>** empfohlen.

Autor: Christoph Schwalbe, Oktober 2015