

# C programmieren lernen für Dummies

Bearbeitet von  
Dan Gookin, Wilhelm Kulisch

1. Auflage 2017. Taschenbuch. 394 S. Softcover  
ISBN 978 3 527 71342 4  
Format (B x L): 14,8 x 21 cm

[Weitere Fachgebiete > EDV, Informatik > Programmiersprachen: Methoden > Programmier- und Skriptsprachen](#)

Zu [Inhalts-](#) und [Sachverzeichnis](#)

schnell und portofrei erhältlich bei

  
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung [beck-shop.de](http://beck-shop.de) ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Die Entwicklungsumgebung (IDE) Code::Blocks herunterladen

Ihr erstes Projekt vorbereiten

Den Code eingeben

Das Projekt und laufen lassen

# Kapitel 1

## Ein schneller Start für die Ungeduldigen

Sie sind wahrscheinlich begierig darauf, mit dem Programmieren in C zu beginnen. Ich will nicht Ihre Zeit vergeuden.

### Was Sie zum Programmieren benötigen

Um Ihren Computer, Ihr Tablet, Ihr Handy, Ihre Spielkonsole oder was auch immer vollständig zu beherrschen, benötigen Sie einige Softwaretools. Die gute Neuigkeit ist, dass zum jetzigen Zeitpunkt, im 21. Jahrhundert, all diese Tools über das Internet frei und einfach verfügbar sind. Sie müssen nur wissen, was erforderlich ist und wo Sie es bekommen.

### Tools zum Programmieren besorgen

Die beiden wichtigsten Dinge, die Sie für den Start Ihres Programmierabenteuers benötigen sind:

- ✓ Ein Computer
- ✓ Ein Internetzugang

Der Computer ist Ihr wichtigstes Instrument zum Schreiben und Erstellen von Programmen. Sogar wenn Sie nur ein Spiel für die Xbox schreiben, benötigen Sie zum Programmieren einen Computer. Der Computer kann ein PC oder ein Macintosh sein. Er kann mit Windows oder Linux arbeiten.

Der Internetzugang ist notwendig, um die restlichen Programmierertools zu bekommen. Sie benötigen einen Texteditor, um den Code zu schreiben, und einen Compiler, um den Code in ein Programm zu verwandeln. Der Compiler kommt gewöhnlich zusammen mit anderen Tools, die Sie brauchen, wie einem Linker und einem Debugger. All diese Tools finden Sie kostenfrei im Internet.

Flippen Sie nicht aus! Die Begriffe *Compiler*, *Linker* und *Debugger* werden alle in Kapitel 2 erläutert.

### Eine Entwicklungsumgebung (IDE) besorgen

Es ist völlig normal, dass Sie Ihre Reise in die Programmierwelt beginnen, indem Sie einen Texteditor, einen Compiler und andere Tools zusammenstellen. Die Verwendung von getrennten Programmen in der Kommandozeile in einem Bildschirmfenster ist die Art, in der ich in grauer Vorzeit das Programmieren gelernt habe. Das ist ein Weg, den man gehen kann, aber er ist etwas altmodisch.

Der geradlinige, professionelle Weg zum Erstellen von Codes besteht heutzutage in der Verwendung einer Integrierten Entwicklungsumgebung (engl. Integrated Development Environment), die von den echten Programmierern *IDE* genannt wird. Sie verbindet alle Tools, die Sie zum Programmieren benötigen, zu einer kompakten, furchterregenden und einschüchternden Einheit.

Sie verwenden die IDE zum Schreiben von Codes, zum Erstellen von Programmen, zum Auffinden von Fehlern und zur Durchführung aller Arten von Zauberei, obwohl es nostalgisch ist, einen separaten Texteditor oder Compiler zu verwenden. Alle Profis nutzen eine IDE. Das empfehle ich für den Gebrauch in diesem Buch ebenfalls.

Die Beschaffung einer IDE löst unzählige Probleme wie das Einrichten eines Compilers, die Konfigurierung des Texteditors und die Aufgabe, diese verschiedenen Elemente dazu zu bringen, zusammen zu arbeiten. Eine IDE zu haben, ist der beste Weg, um mit dem Programmieren zu beginnen – worauf Sie wahrscheinlich ziemlich scharf sind.

### Die IDE Code::Blocks betrachten

Sie werden im Internet mehr als genug IDEs finden und sie sind alle ziemlich gut. Doch um konsistent zu sein, arbeitet dieses Buch mit der IDE Code::Blocks, die unter Windows, Mac OS X und Linux läuft. Sie hat alles, was Sie benötigen.

Wenn Sie bereits eine IDE haben – großartig! Ich bin sicher, dass sie das gleiche wie Code::Blocks kann, obwohl in diesem Buch die Abbildungen und Beispiele, vor allem in den ersten Kapiteln, typisch für Code::Blocks sind.

## Code::Blocks installieren

Sie erhalten Code::Blocks im Internet auf folgender Webseite: [www.codeblocks.org](http://www.codeblocks.org)

Diese Webseite wird sich zweifellos im Verlauf der Zeit ändern, so dass sich die folgenden Schritte zur Installation der IDE etwas ändern können:

1. **Verwenden Sie den Webbrowser Ihres Computers, um die Webseite Code::Blocks zu besuchen.**
2. **Gehen Sie in den Bereich zum Herunterladen.**

Laden Sie eine binäre oder lauffähige Version von Code::Blocks herunter. Sie muss für das Betriebssystem Ihres Computers geeignet sein. Suchen Sie außerdem eine Version mit einem C-Compiler, wie etwa dem gebräuchlichen Compiler MinGW.

3. **Klicken Sie auf den Link zum Anzeigen der binären oder lauffähigen Installation von Code::Blocks.**

Als dieses Buch in den Druck ging, hieß der Link `DOWNLOAD THE BINARY RELEASE`.

4. **Wählen Sie das Betriebssystem Ihres Computers oder scrollen Sie zu dem Teil des Bildschirms, der Optionen für dieses Betriebssystem enthält.**

Sie finden Abschnitte (oder Seiten) für Windows, Linux und Mac OS X.

5. **Klicken Sie auf den Link, der den für Ihr Betriebssystem geeigneten Compiler und die entsprechende IDE herunterlädt.**

Die Version des IDEs und des Compilers für Windows ist folgendermaßen benannt:

```
codeblocks-xx.yy mingw-setup.exe
```

xx und yy sind dabei Platzhalter für die Nummern der jeweiligen Version des Code::Blocks.

In Linux können Sie je nach Ihrer Linux-Distribution, kurz *Distro*, die 32- oder die 64-Bit-Version und das von Ihnen gewünschte Dateiformat auswählen. Ich empfehle eine stabile Version.

Die Nutzer von Mac OS X können wählen, ob sie eine `.dmg`-Datei (Disk-Image-Datei) oder eine herkömmliche Zip-Datei herunterladen wollen.

6. **Extrahieren Sie das Code::Blocks-Installationsprogramm aus dem Archiv.**

Ungeachtet des *Für Dummies*-Titels dieses Buches gehe ich davon aus, dass Sie mit `.zip`, `.gz`, `.dmg` und anderen Dateiformaten in dem von Ihnen genutzten Betriebssystem erfolgreich arbeiten können.

7. **Starten Sie das Installationsprogramm.**

Achten Sie auf die Anweisungen auf dem Bildschirm. Führen Sie eine Standardinstallation durch; Sie müssen zu diesem Zeitpunkt nichts individuell einrichten.

Wenn Sie unter Windows arbeiten, stellen Sie sicher, dass Sie den MinGW-Compiler installieren. Wenn diese Option nicht im Fenster für die Wahl der Komponenten angezeigt wird, haben Sie die falsche Version von Code::Blocks heruntergeladen. Dann gehen Sie zurück zu Schritt 5.

**8. Beenden Sie die Installation, indem Sie Code::Blocks laufen lassen.**

Auf meinem Bildschirm erschien eine Anzeige, die fragte, ob ich Code::Blocks laufen lassen möchte. Ich klickte auf »Yes«. Wenn Sie diese Abfrage nicht sehen, verwenden Sie das Betriebssystem des Computers, um Code::Blocks so zu starten wie jedes andere Programm auch.

**9. Schließen Sie das Fenster zur Installation.**

Auch wenn Sie Code::Blocks über den Bildschirm laufen sehen, müssen Sie trotzdem die Installation beenden, indem Sie das Installationsfenster schließen.

Der folgende Abschnitt bietet einen Überblick über die Benutzeroberfläche des Code::Blocks-Programms.

## Eine Reise durch den Arbeitsbereich von Code::Blocks

Wenn Code::Blocks noch nicht gestartet ist, schreiten Sie voran, indem Sie es starten. Man startet es wie jedes andere Programm: Gehen Sie zu seinem Icon im Startmenü oder zu dem entsprechenden Icon auf der Bildschirmoberfläche. Dies ist der einfachste Weg zum Starten von Code::Blocks unter Windows 8.

Abbildung 1.1 zeigt den *Arbeitsbereich*, wie der offizielle Name des gewaltigen Mosaiks aus Fenstern lautet, die Sie auf dem Bildschirm sehen.

Die Einzelheiten in Abbildung 1.1 sind relativ klein, aber Sie sollten die wesentlichen Bereiche kennen, die in der Abbildung angegeben sind:

**Symbolleisten** (engl. Toolbars): Diese vertrackten Leisten, die mit verschiedenen Kommandozeichen verziert sind, sind am oberen Ende des Code::Blocks-Fensters angeordnet. Es gibt acht Leisten, die Sie umordnen, anzeigen oder verstecken können. Legen Sie sich nicht mit ihnen an, bevor Sie sich mit der Benutzeroberfläche angefreundet haben.

**Verwaltung** (engl. Management): Das Fenster im linken Bereich des Arbeitsbereichs zeigt vier Reiter (engl. Tabs), obwohl Sie vielleicht nicht alle vier gleichzeitig sehen. Es bietet eine praktische Beaufsichtigung Ihrer Programmierbemühungen.

**Statuszeile** (engl. Status bar): Am unteren Ende des Bildschirms sehen Sie Informationen über das Projekt, den Editor und über andere Aktivitäten, die in Code::Blocks ablaufen.

**Editor**: Das große Fenster in dem mittleren und rechten Bereich des Bildschirms ist der Bereich, in dem Sie den Code eingeben.

**Protokolldateien** (engl. Logs): Im unteren Teil des Bildschirms befindet sich ein Fenster mit sehr vielen Reitern. Jeder Reiter zeigt Informationen über Ihr Programmierprojekt an. Der Reiter, den Sie am häufigsten verwenden, heißt BUILD LOG.

Das Menü VIEW kontrolliert die Sichtbarkeit eines jeden in dem Fenster angezeigten Posten. Wählen Sie das dazugehörige Kommando sowie Manager aus dem Menü VIEW, um diesen Posten zu zeigen oder zu verstecken. Kontrollieren Sie die Symbolleisten, indem Sie das Untermenü VIEW|TOOLBARS verwenden.

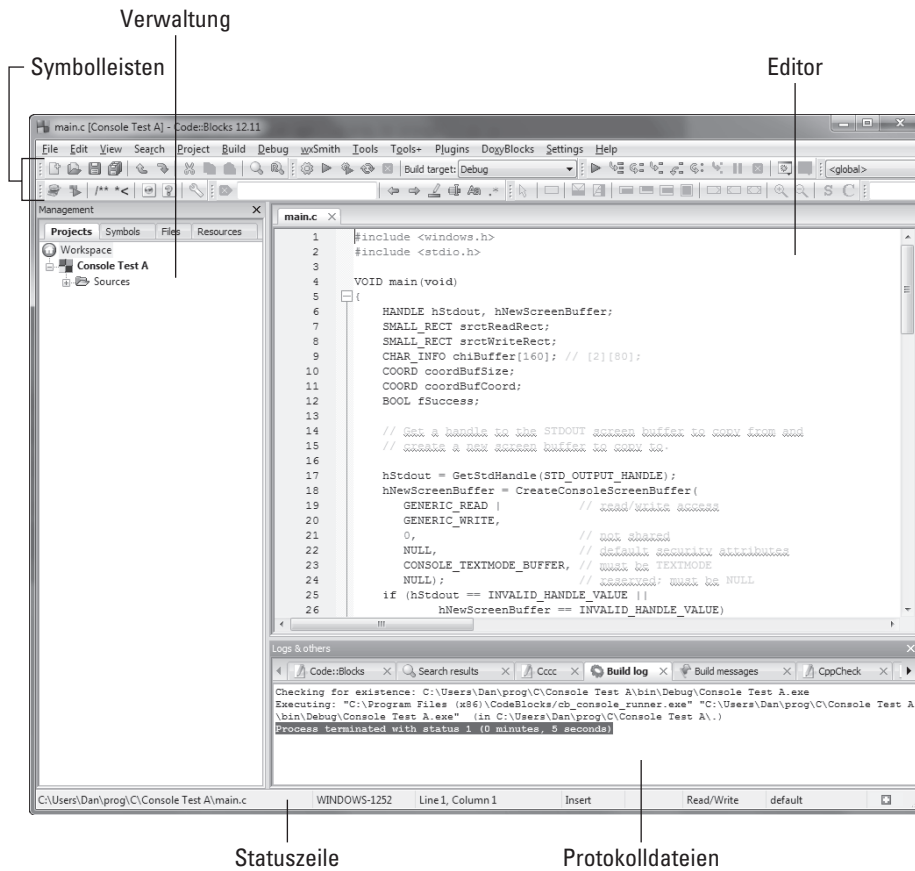


Abbildung 1.1: Der Arbeitsbereich von Code::Blocks



Das Wichtigste, was Sie bei der Benutzeroberfläche von Code::Blocks beachten müssen: Lassen Sie sich nicht von ihr verwirren. Eine IDE wie Code::Blocks kann höchst einschüchternd sein, auch wenn Sie sich als einen erfahrenen Programmierer betrachten. Keine Angst: Sie werden sich schnell zu Hause fühlen.

- ✓ Vergrößern Sie das Code::Blocks-Fenster, so dass es den gesamten Bildschirm füllt. Sie benötigen all diesen Platz.
- ✓ Jeder der verschiedenen Bereiche auf dem Bildschirm – Verwaltung, Editor, Protokolldateien – kann in seiner Größe verändert werden: Setzen Sie den Mauszeiger zwischen zwei Bereiche. Wenn sich der Mauszeiger in ein doppelpeilartiges Etwas verwandelt, können Sie die Maus so ziehen, dass sich die Größe eines Bereichs ändert.
- ✓ Der Editor und die Protokolldateien weisen gekennzeichnete Benutzeroberflächen auf. Jedes Fenster zeigt viele »Blätter« mit Informationen. Wechseln Sie zwischen den Blättern, indem Sie einen anderen Reiter wählen.

## Ihr erstes Projekt

Bei der herkömmlichen Computerprogrammierung verwendete man, voneinander getrennt, einen Texteditor, einen Compiler und einen Linker. Die *Eingabeaufforderungszeile* (engl. *command prompt*) wurde benutzt, um die Kommandos zum Editieren, Kompilieren und Linken einzugeben. Es war ein linearer Prozess, und war für kleine Projekte sehr gut geeignet. Mit dem Aufkommen neuer Betriebssysteme und der Programmierung von mobilen Geräten und Spielkonsolen wird diese lineare Methode jedoch höchst ineffizient.

Die moderne IDE besitzt immer noch die Elemente eines Editors, Compilers, Linkers, Debugger und anderer Programmierertools. Sie verfügt über Eigenschaften, die zum Erzeugen von grafischen Programmen und komplexen Projekten erforderlich sind. Daher ist die IDE darauf ausgerichtet, nicht nur mit einzelnen Programmen, sondern mit Projekten zu arbeiten.

In diesem Abschnitt wird der Prozess des Erstellens eines Projekts mit der IDE Code::Blocks erläutert.

### Ein neues Projekt erstellen

Die in diesem Buch vorgestellten Beispiele sind ausschließlich Tastaturanwendungen; das bedeutet, dass sie im Textmodus in einem Computerfenster laufen. Meiner Meinung nach ist das der beste Weg zum Lehren von grundlegenden Programmierkonzepten, ohne Sie mit einem großen, komplexen, grafischen Scheusal von Programm zu erschlagen. Auch wenn eine IDE viel leistungsfähiger ist, verwenden sie Sie in diesem Buch, um einfache, tastaturbasierte Programme zu erzeugen. Sie arbeitet folgendermaßen:

**1. Starten Sie Code::Blocks.**

Sie sehen den Start–Here–Bildschirm, der das Logo von Code::Blocks und einige Links anzeigt. Wenn Sie den Start–Here–Bildschirm nicht sehen, dann wählen Sie `FILE|CLOSE WORKSPACE`.

**2. Klicken Sie auf den Link CREATE A NEW PROJECT.**

Die in Abbildung 1.2 gezeigte Dialogbox `NEW FROM TEMPLATE` erscheint.

**3. Wählen Sie CONSOLE APPLICATION aus und klicken Sie auf das Feld GO.**

Der Tastaturanwendungs–Wizard erscheint.



Sie können eine Markierung bei `SKIP THIS PAGE NEXT TIME` machen, um den ersten Schirm des Wizards zu überspringen.

**4. Klicken Sie auf das Feld NEXT.**

**5. Wählen Sie C als die Sprache aus, die Sie verwenden wollen und klicken Sie dann auf das Feld NEXT.**

C unterscheidet sich stark von C++ – Sie können in der einen Sprache Sachen machen, die in der anderen nicht erlaubt sind.

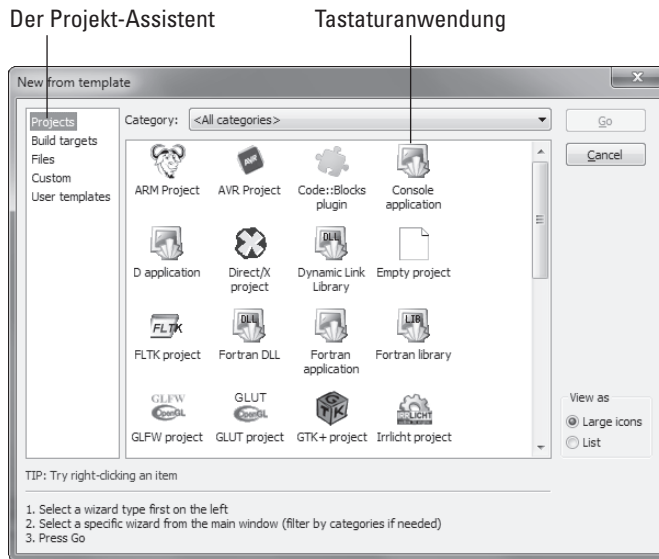


Abbildung 1.2: Hier starten neue Projekte.

## 6. Geben Sie `ex0101` als Projekttitel ein.

Alle Codes in diesem Buch sind nach diesem Schema benannt: Die Buchstaben `ex` (für `exercise` = Übung), gefolgt von einer zweistelligen Kapitelnummer und einer zweistelligen Projektnummer.

Wenn Sie den Projekttitel festlegen, wird der Dateiname des Projekts automatisch ausgefüllt.

## 7. Klicken Sie auf das `Browse`-Feld rechts in der Textbox mit dem Namen `FOLDER TO CREATE PROJECT IN`.

Ich empfehle, dass Sie einen gesonderten Ordner für alle Projekte in diesem Buch anlegen.

## 8. Verwenden Sie das Feld `MAKE NEW FOLDER` in der Dialogbox `BROWSE FOR FOLDER`, um einen neuen Projektordner zu erzeugen.

Wenn Sie bereits einen Ordner für Programmierprojekte haben, erzeugen Sie einen neuen Unterordner mit dem Namen `BegC4D` für die Projekte in diesem Buch. Der Name des Ordners beruht auf dem Titel der amerikanischen Ausgabe dieses Buches *Beginning Programming with C For Dummies*.

Wenn Sie keinen Ordner für Programmierprojekte im Hauptordner Ihres Benutzerkontos haben, erzeugen Sie einen Unterordner namens `prog` und dann einen weiteren Ordner namens `c`. Zum Schluss erzeugen Sie einen Unterordner namens `BegC4D` und wählen ihn zum Speichern des Projekts.

In meinem Computer, der mit Windows 7 arbeitet, sind die Projekte aus diesem Buch in folgendem Ordner gespeichert:

```
C:\Users\Dan\prog\c\BegC4D\
```



9. Klicken Sie auf das Feld OK, um den Ordner auszuwählen, und schließen Sie die Dialogbox.

10. Klicken Sie auf das Feld NEXT.

Der nächste Bildschirm (der letzte) ermöglicht Ihnen, einen Compiler auszuwählen und zu entscheiden, ob Sie eine Debug- oder eine Veröffentlichungsversion oder beides anlegen wollen.

Die Auswahl des Compilers ist in Ordnung; der Compiler GNU GCC (oder was auch immer in dem Fenster angezeigt wird) ist genau der, den Sie haben wollen.

11. Entfernen Sie das Häkchen an der Zeile CREATE DEBUG CONFIGURATION.

Sie benötigen diese Konfiguration nur, wenn Sie ein Programmierdilemma beheben müssen (engl. *debug*), das Ihnen Kopfzerbrechen bereitet. In Kapitel 25 wird diese Situation behandelt.

12. Klicken Sie auf das Feld FINISH.

Code::Blocks erzeugt ein Gerüst Ihres Projekts, das Sie möglicherweise dennoch nicht im Fenster des Programms sehen. Bei einem einfachen Kommandozeilenprojekt ist das Gerüst eine *Quelltextdatei* – eine Klartextdatei, die den C-Programmiercode enthält, um Ihnen zu helfen, Ihr Projekt zu starten.

Lesen Sie im folgenden Abschnitt weiter.



Eine Tastaturanwendung ist eine der einfachsten Programme, die Sie erstellen können. Wenn Sie ein Windowsprogramm erstellen, dann erzeugt Code::Blocks andere Elemente für Sie und zeigt Tools zum Entwerfen des Fensters, zum Erstellen eines Icons und zum Arbeiten mit anderem erforderlichen Krimskrams an. Die IDE macht es leicht, mit all diesen Dingen zu arbeiten.

## Den Quelltext überprüfen

Wenn Code::Blocks ein neues Projekt hervorzaubert, erzeugt es eine Handvoll grundlegender Elemente, damit Sie anfangen können. Bei einer Tastaturanwendung beinhaltet dies nur ein Element: Die Quelltextdatei `main.c`.

Eine Quelltextdatei ist eine Klartextdatei, die den Programmiercode enthält. In Kapitel 2 werden die Einzelheiten erläutert, doch im Moment reicht es aus zu wissen, dass Code::Blocks diese Datei jedes Mal automatisch erzeugt, wenn Sie ein neues Projekt starten. Es gibt Ihnen sogar einen Mustercode vor, mit dem Sie starten können, obwohl Sie meistens das Gerüst löschen und von vorn anfangen, indem Sie Ihren eigenen Code schreiben.

Projekte in der IDE werden in dem Kasten Management auf der linken Seite des Bildschirms (siehe Abbildung 1.1) aufgelistet. Wenn Sie diesen Kasten nicht sehen, drücken Sie  $\diamond + F2$  auf der Tastatur oder wählen Sie VIEW|MANAGER aus dem Menü aus.

Abbildung 1.3 zeigt den Managementkasten mit dem ausgewählten Projektreiter. Das derzeitige Projekt ist geöffnet, aber Sie können auch andere Projekte sehen.

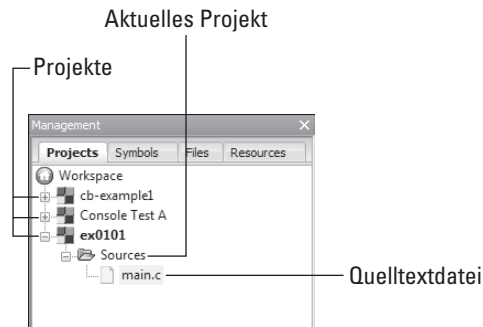


Abbildung 1.3: Projekte in der IDE

Bei einem neuen Tastaturanwendungsprojekt wird die Quelltextdatei `main.c` genannt. Sie wird in dem Quellordner des Projekts gespeichert. Wenn Sie die Quelltextdatei nicht sehen (siehe Abbildung 1.3), dann öffnen Sie den Quellordner, um ihn anzuzeigen.

Machen Sie einen Doppelklick auf das Element `main.c` im Quellordner, um die Datei im Editorfenster zu öffnen. Das vorgeschriebene Codegerüst wird in Listing 1.1 angezeigt. Es handelt sich um einen richtigen Code für ein reales Programm in C, welches Sie erstellen können, wie im nächsten Abschnitt erläutert wird.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello world!\n");
    return 0;
}
```

Listing 1.1: Das Code::Blocks–Gerüst

In Kapitel 2 werden die Einzelheiten zu Listing 1.1 erläutert. Im Moment können Sie den farbenprächtigen Text und die schöne Formatierung bewundern.

- ✓ Das Editorfenster zeigt Zeilennummern an. Wenn Sie sie nicht sehen, wählen Sie `EDIT|EDITOR TWEAKS|SHOW LINE NUMBERS`. In diesem Buch wird davon ausgegangen, dass Sie die Zeilennummern sehen können, um Bezug darauf nehmen zu können.
- ✓ Die Zeilennummern sind kein Teil des C–Codes. Sie werden zum Bearbeiten benutzt; aber auch wenn Fehler auftreten, beziehen sich die Angaben auf die Zeilennummern in dem Code.
- ✓ Sie können den Einzug des Tabulators festlegen, indem Sie `EDIT|EDITOR TWEAKS|TAB SIZE` auswählen und dann die Größe wählen. Ich bevorzuge vier Leerzeichen pro Einzug.
- ✓ Der Farbton des Textes wird mithilfe von `EDIT|HIGHLIGHT MODE|C/C++` festgelegt. Um die Hervorhebungen zu entfernen, wählen Sie das Element `PLAIN TEXT` aus dem Untermenü `HIGHLIGHT MODE` aus. (Es ist dort das erste Element.)

## Das Projekt erstellen und laufen lassen

Um ein Programm in Code::Blocks zu erzeugen, müssen Sie das Projekt erstellen. Dieser einzelne Schritt bedingt verschiedene Dinge; ich vermeide es allerdings, die Einzelheiten vor Kapitel 2 zu diskutieren. Wenn Sie bereits Ihr erstes Projekt `ex0101` begonnen haben, es geöffnet ist und in Code::Blocks angezeigt wird, sind Sie so weit, dass Sie es erstellen können. Beachten Sie folgende Schritte:

1. Stellen Sie sicher, dass das Projekt, das Sie erstellen wollen, im Managementfenster aktiviert ist.

Aktivierte Projekte erscheinen in Fettdruck. Wenn mehr als ein Projekt im Projektfenster angezeigt wird, aktivieren Sie dasjenige, mit dem Sie arbeiten wollen, indem Sie mit der rechten Maustaste den Projektnamen anklicken (über den Code::Blocks-Icon) und das Kommando `ACTIVATE PROJECT` wählen.

2. Wählen Sie `BUILD|BUILD` aus dem Menü.

Der Reiter `BUILD LOG` in der Protokolldatei zeigt die Ergebnisse des Erstellens eines Projektes an. Sie sehen dort einige Zeilen Text; die Einzelheiten dazu werden in Kapitel 2 erläutert.

Da Sie sich nicht in das Quelltextgerüst eingemischt haben, wird das Projekt ohne Fehler übersetzt; Sie sehen den Text im Überblick, der null Fehler und null Warnungen anzeigt. Gut. Wenn Fehler auftreten, was häufig genug passiert, machen Sie sich daran, sie zu beheben. Die Fehlermeldungen helfen in dieser Hinsicht; dieses Thema wird in Kapitel 3 erläutert.

Das Erstellen eines Projekts ist aber nur die halbe Arbeit. Die andere Hälfte besteht im Laufenlassen des Projekts, also das Ausführen des fertiggestellten Programms innerhalb der IDE.

Um das aktuelle Projekt laufen zu lassen, wählen Sie `BUILD|RUN` aus dem Menü. Sie sehen, dass das Bildschirmfenster erscheint und die Ausgaben des Programms sowie etwas überflüssiger Text aufgelistet sind, wie in Abbildung 1.4 gezeigt.









Abbildung 1.4: Laufenlassen eines Programms

Drücken Sie , um das Eingabeaufforderungsfenster zu schließen.









Und nun der Clue: Sie können ein Projekt erstellen *und* laufen lassen mithilfe eines einzigen Befehls: Wählen Sie BUILD|BUILD AND RUN.

- ✓ Die Tastenkombinationen für BUILD, RUN und BUILD AND RUN sind  + ,  +  und . Es ist aber nicht nötig, sich diese Abkürzungen zu merken – sie sind im Menü aufgelistet.
- ✓ Befehlsfelder zum Erstellen und Ausführen eines Projekts befinden sich in der Symbolleiste des Compilers. Sie finden ein gelbes Getriebe-Icon für BUILD, einen grünen Pfeil für RUN und eine Kombination aus beiden für den Befehl BUILD AND RUN.
- ✓ Die Ausgabe des Programms erscheint im oberen Teil des Eingabeaufforderungsfensters (siehe Abbildung 1.4). Die letzten beiden Zeilen werden von der IDE erzeugt, wenn das Programm läuft. Der Text enthält einen Wert, der vom Programm an das Betriebssystem zurückgegeben wird, eine null und die Angabe, wie lange das Programm läuft (9 Millisekunden). Die Anzeige Press any key to continue besagt, dass Sie mit  das Fenster schließen können.



## Speichern und schließen

Wenn Sie ein Projekt erstellt haben oder eine kleine Änderung vorgenommen haben, sollten Sie speichern. In einer IDE müssen Sie allerdings verschiedene Dinge speichern, etwa die Quelltextdatei, den Arbeitsbereich und das Projekt. Befehle zum Speichern dieser verschiedenen Punkte finden Sie in dem Menü FILE.

Ein praktischer Weg, um alles auf einmal zu speichern, ist die Verwendung des Befehls SAVE EVERYTHING. Diesen Befehl finden Sie in dem Menü FILE, die praktische Abkürzung auf der Tastatur ist  +  +  (oder  +  +  bei einem Macintosh).

Wenn Sie das Projekt bisher noch nicht gespeichert haben, machen Sie es jetzt.

Sie können das aktuelle Projekt auch schließen, wenn Sie FILE|CLOSE PROJECT wählen.

- ✓ Verlassen Sie Code::Blocks, wenn Sie eine Pause einlegen möchten.
- ✓ Keine Sorge! Code::Blocks ermahnt Sie, wenn Dateien oder Projekte noch ungespeichert sind. Speichern, dann schließen.
- ✓ Wenn der Befehl SAVE EVERYTHING nicht verfügbar ist, gehen Sie davon aus, dass alles in Ordnung und bereits gespeichert ist.
- ✓ Der Standardbefehl zum Speichern in Code::Blocks – entweder FILE|SAVE oder  +  – wird verwendet, um Dateien wie Ihre Quelltextdatei in der Sprache C zu speichern. Ich empfehle Ihnen dringend, Ihre Quelltextdateien zu speichern, wenn Sie sie bearbeitet haben.



## Verwendung der Man-Seiten

Ein Fenster des Arbeitsbereichs von Code::Blocks trägt den Namen `Man/Html Pages Viewer`. Es ist der Link zur Dokumentation der Bibliothek von C. `Man` ist die Kurzform von `Manual`. Das Fenster können Sie ansehen, indem Sie `VIEW | MAN PAGES VIEWER` auswählen.

Um den `Man Pages Viewer` zu verwenden, geben Sie ein Schlüsselwort oder eine Funktion der Sprache C in das Textfeld ein. Klicken Sie in das Feld `SEARCH` oder drücken Sie `[↵]`, um die gegebenen Elemente anzuschauen. Haben Sie Geduld; manchmal dauert es etwas, bis die Information zugänglich ist.

Wenn `Man Pages Viewer` die `Man-Seiten` nicht finden kann, können Sie im Internet nach der C-Funktionsdokumentation suchen. Ich empfehle folgende Seiten:

```
http://man7.org/linux/man-pages/dir_section_2.html  
http://man7.org/linux/man-pages/dir_section_3.html
```

Die Funktionen in C, auf die in diesem Buch verwiesen wird, findet man auf der `Man-Seite 2` oder der `Man-Seite 3`; sie sind mit den beiden Seiten verlinkt, die gerade angegeben wurden.

Wenn Sie ein Betriebssystem des Typs `Unix` haben, können Sie ein Bildschirmfenster aufrufen und den Befehl `man` verwenden, um Funktionsdokumente durchzusehen; zum Beispiel:

```
man rand
```

Dieser Befehl zeigt Informationen über die Funktion `rand ( )` an.