



# **1 ERSTE SCHRITTE MIT C#**

Am besten legen wir gleich los mit dem Programmieren. Nach dem Start des Computers und dem Auftauchen von Windows können wir uns schon dem ersten C#-Projekt widmen. Es wird natürlich noch kein Computerspiel sein, aber es gibt schon einiges zum Herumspielen.

In diesem Kapitel lernst du

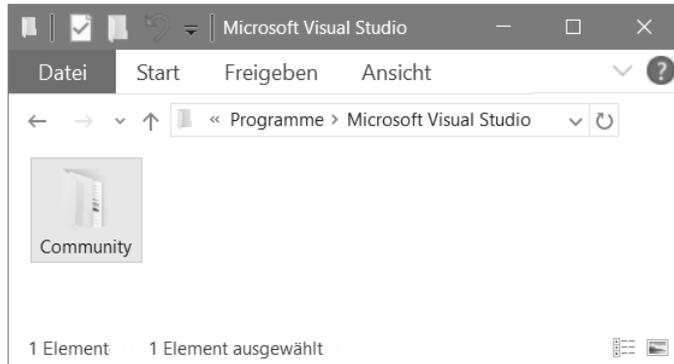
- ⊙ wie man Visual Studio startet
- ⊙ wie man ein Programm erstellt und ausführt
- ⊙ einiges über (mögliche) Fehler
- ⊙ was `WriteLine()` und `ReadLine()` bedeuten
- ⊙ etwas über Variablen
- ⊙ wie man ein Projekt speichert
- ⊙ wie man Visual Studio beendet

## VISUAL STUDIO STARTEN

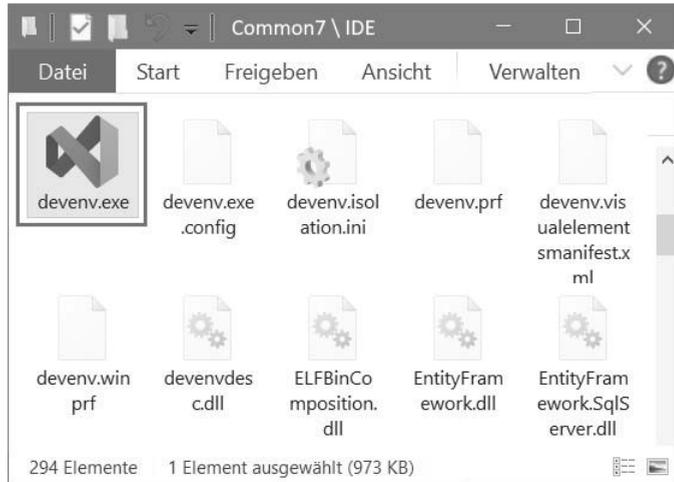
Bevor wir mit dem Programmieren anfangen können, muss Visual Studio erst installiert werden. Die Installation übernimmt ein Programm namens SETUP. Genaues erfährst du im Anhang A. Hier musst du dir von jemandem helfen lassen, wenn du dir die Installation nicht allein zutraust.

Eine Möglichkeit, Visual Studio zu starten, ist diese:

- Öffne den Ordner, in dem du Visual Studio untergebracht hast (z.B. C:\PROGRAMME\MICROSOFT VISUAL STUDIO).

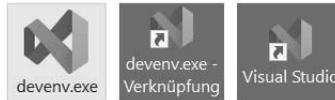


- Dort musst du nun weiter über einige Unterordner wie COMMUNITY und COMMON7 in IDE wechseln:



- Hier suchst du unter den zahlreichen Symbolen eines derjenigen heraus, bei denen etwas aussieht wie eine gekippte lila 8, und zwar das mit dem Namen DEVENV.EXE. Das Symbol wird nicht so schnell zu finden sein.

- Doch dann kannst du das Programm mit einem Doppelklick auf das Symbol starten:



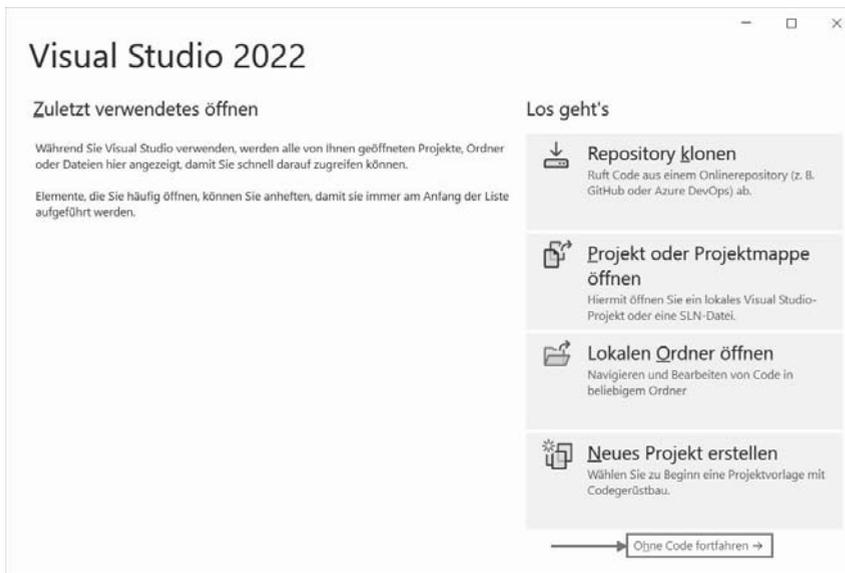
Ich empfehle dir, eine **Verknüpfung** auf dem Desktop anzulegen:

- ❖ Dazu klickst du mit der rechten Maustaste auf das Symbol für Visual Studio (DEVENV.EXE). Im Kontextmenü wählst du KOPIEREN.
- ❖ Dann klicke auf eine freie Stelle auf dem Desktop, ebenfalls mit der rechten Maustaste. Im Kontextmenü wählst du VERKNÜPFUNG EINFÜGEN.
- ❖ Es ist sinnvoll, für das neue Symbol auf dem Desktop den Text DEVENV.EXE – VERKNÜPFUNG durch VISUAL STUDIO zu ersetzen.

Von nun an kannst du auf das neue Symbol auf dem Desktop **doppelklicken** und damit Visual Studio starten.



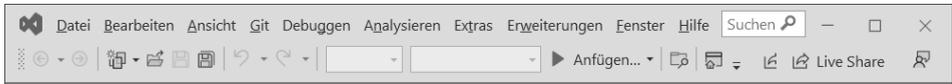
Je nach Computer kann es eine Weile dauern, bis Visual Studio geladen ist. Was dich schließlich erwartet, könnte ungefähr so aussehen:



Wenn du hier auf NEUES PROJEKT ERSTELLEN klickst, beginnt dein erstes Projekt. Du kannst also gleich loslegen, wenn du willst. Oder:

- Du klickst erst einmal auf OHNE CODE FORTFAHREN, um im Hauptfenster von Visual Studio zu landen.

Dort schauen wir uns jetzt ein bisschen um. Sieht ganz schön leer aus, doch uns soll jetzt nur die Menüleiste interessieren – ganz oben:



Links darunter befinden sich jede Menge Symbole, die man mit der Maus anklicken kann. Zu einigen davon kommen wir im Laufe der folgenden Kapitel noch.

Diese Menüs von Visual Studio wirst du wahrscheinlich am meisten benutzen:

- ❖ Über das DATEI-Menü kannst du Dateien speichern, laden (öffnen), ausdrucken, neu erstellen oder Visual Studio beenden.
- ❖ Das BEARBEITEN-Menü hilft dir bei der Bearbeitung deines Programmtextes, aber auch bei anderen Programmelementen. Außerdem kannst du dort bestimmte Arbeitsschritte rückgängig machen oder wiederherstellen.
- ❖ Im ANSICHT-Menü hast du unter anderem die Möglichkeit, zusätzliche Hilfsfenster und Boxen ein- oder auszublenden.
- ❖ Über das DEBUGGEN-Menü sorgst du dafür, dass dein Programmprojekt ausgeführt wird.
- ❖ Und das HILFE-Menü bietet dir vielfältige Hilfe-Informationen an.

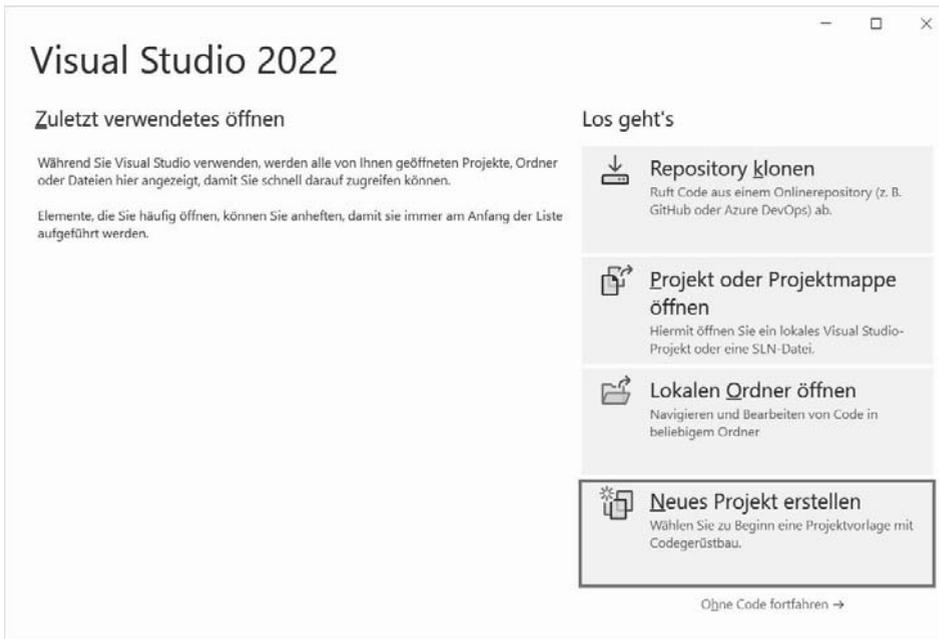


Einige wichtige Menüeinträge sind in einem sogenannten **Popup**-Menü zusammengefasst. Das heißt so, weil es dort aufklappt, wo du gerade mit der rechten Maustaste hinklickst.

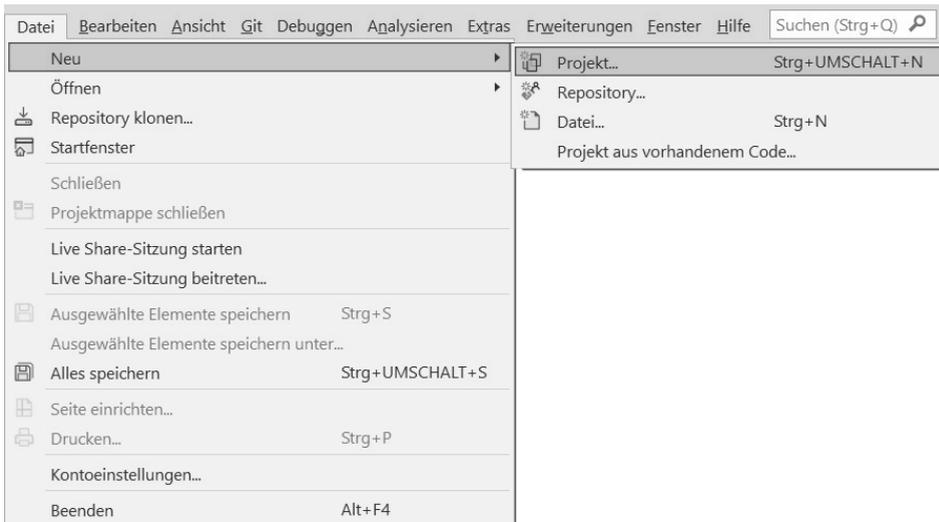
Ein Editorfenster, wie du es vielleicht von einem Editor oder Textverarbeitungsprogramm her kennst, ist gerade nicht in Sicht. Aber das lässt sich ändern: Ziel ist es ja, ein neues Projekt – unser Erstlingswerk – zu erstellen. Also los!

## DAS ERSTE PROGRAMM

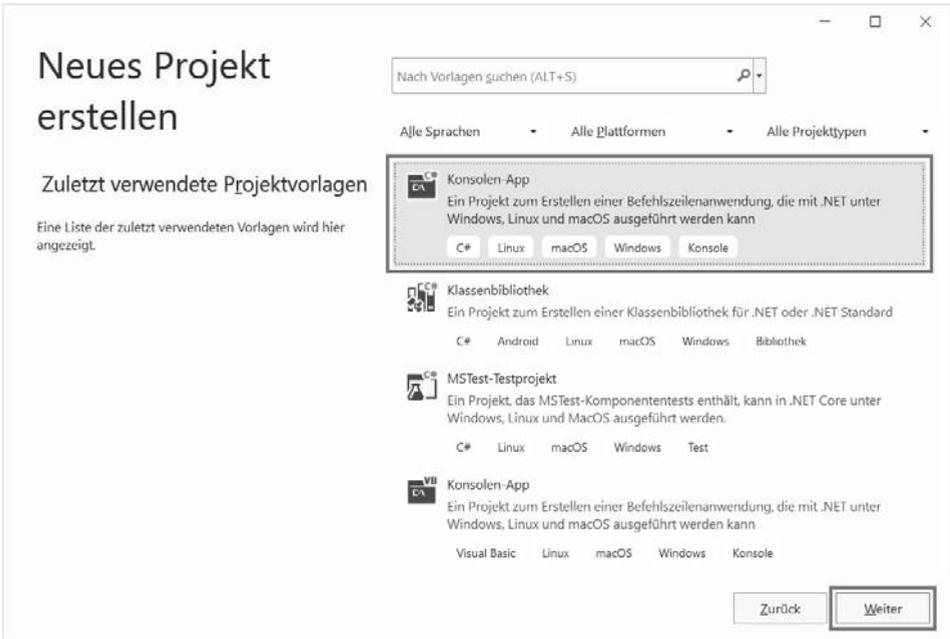
- Es gibt nun diese zwei Wege. Entweder du schließt das Fenster von Visual Studio (Klick auf das X rechts oben) und startest es neu – womit du in diesem Fenster landest:



- Dann klickst du auf NEUES PROJEKT ERSTELLEN.
- Oder du hast dich entschieden, bei der Menüleiste von Visual Studio zu bleiben. Dann klickst du dort auf DATEI und im sich öffnenden Menü auf NEU und dann auf PROJEKT.



Es erscheint ein Dialogfeld, in dem es offenbar mehrere Möglichkeiten gibt, wie du dein Projekt erstellst:



➤ Wähle rechts oben die Option KONSOLEN-APP (dort findest du auch den Eintrag C#). Dann klicke unten rechts auf WEITER.



## DER QUELLTEXT

---

- Gib dem Projekt einen Namen und Sorge dafür, dass hinter SPEICHERORT der Pfad steht, wo du dein Projekt unterbringen willst. Dann klicke abschließend auf WEITER.

Ich habe das Ganze einfallslos erst mal PROJEKT1 genannt. Du kannst den vorgegebenen Speicherort übernehmen, ich empfehle dir aber, besser einen eigenen Ordner zu erzeugen und den dort anzugeben. Bei mir ist das der Ordner PROJEKTE auf Laufwerk D:.



Weitere Informationen

Konsolen-App C# Linux macOS Windows Konsole

Framework ⓘ

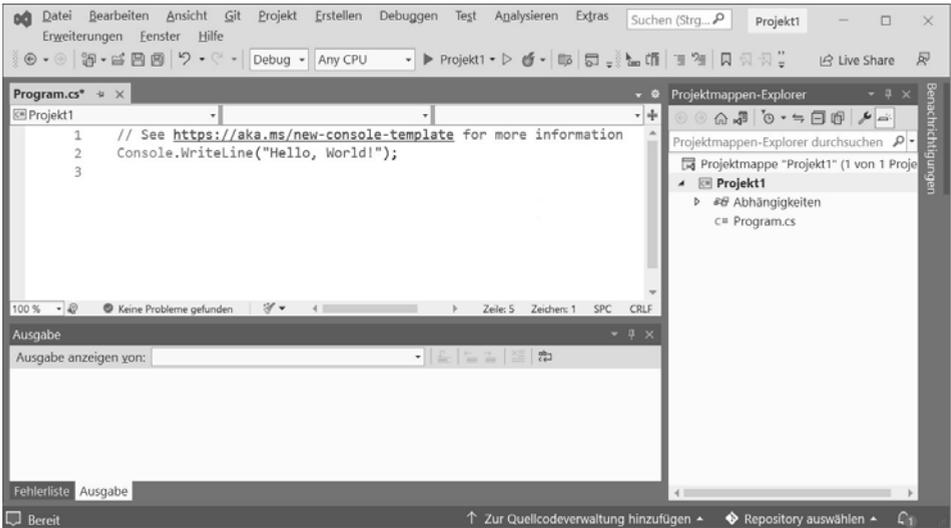
.NET 6.0 (Langfristige Unterstützung)

Zurück Erstellen

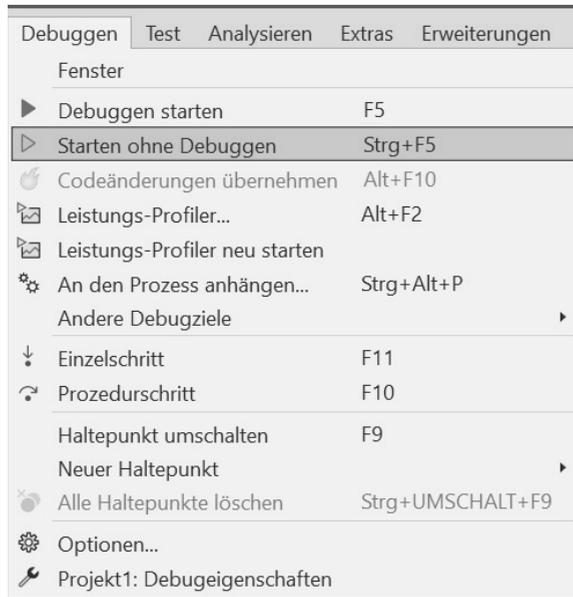
- Im nächsten Dialogfeld erzeugst du mit einem abschließenden Klick auf ERSTELLEN dein erstes Mini-Projekt in C#.

## DER QUELLTEXT

Wie du im linken großen Fensterbereich sehen kannst, bietet Visual Studio schon ein kleines Gerüst-Programm, das du natürlich auch starten kannst, allerdings passiert noch nichts Aufregendes.



➤ Probiere ruhig mal aus, was sich tut, wenn du in der Menüleiste auf **DEBUGGEN** und **STARTEN OHNE DEBUGGEN** klickst. (Alternativ kannst du auch die Tastenkombination **Strg + F5** drücken.)



Und nach dem Start des Programms tut sich tatsächlich etwas: Unten im Ausgabe-fenster steht einiges an Meldungen, die du jetzt einfach ignorieren solltest. Wichtiger ist das neue Fenster mit schwarzem Hintergrund, das sich öffnet und sich vor Visual Studio schiebt. Das ist das Konsolenfenster. Und in dem steht zum einen ein kurzer Gruß, darunter dann etwas momentan vielleicht für dich unverständliches

Kauderwelsch. Wichtig ist der letzte Satz, die Aufforderung, eine beliebige Taste zu drücken.



```
Microsoft Visual Studio-Debugging-Konsole
Hello, World!
D:\Projekte\Projekt1\bin\Debug\net6.0\Projekt1.exe wurde mit Code "0" beendet.
Drücken Sie eine beliebige Taste, um dieses Fenster zu schließen.
```

➤ Folge dieser Aufforderung (oder Bitte), um das Fenster wieder zu schließen.

Na ja, überwältigend war deine erste Begegnung mit einem C#-Programm nicht, aber es liegt an dir, daraus mehr zu machen. Zuerst aber schauen wir uns jetzt das näher an, was da links oben im Editorfenster steht:

```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```

Das, was du da siehst, wird **Quelltext** genannt. Die erste Zeile mit den beiden Schrägstrichen (//) am Anfang ist ein **Kommentar**. Der weist in diesem Falle darauf hin, dass du mehr unter der angezeigten Internet-Adresse erfahren kannst.

Als Kommentar könnte man aber auch so etwas hinschreiben:

```
// Mein erstes Projekt
```

Oder man lässt diesen Kommentar einfach ganz weg. Für ein funktionierendes Programm ist nur diese Zeile nötig:

```
Console.WriteLine("Hello, World!");
```

Und da steht schon einiges, was für uns interessant ist. Offenbar bewirkt diese Zeile, dass ein bestimmter Text ausgegeben wird. In diesem Fall ist es »Hello, World«, aber das lässt sich leicht ändern:

```
Console.WriteLine("Hallo, wer da?");
```

- Passe den Programmtext an, indem du einiges löschst und ersetzt. Dann starte das Programm erneut über DEBUGGEN und STARTEN OHNE DEBUGGINGoder mit `Strg` + `F5`.

```

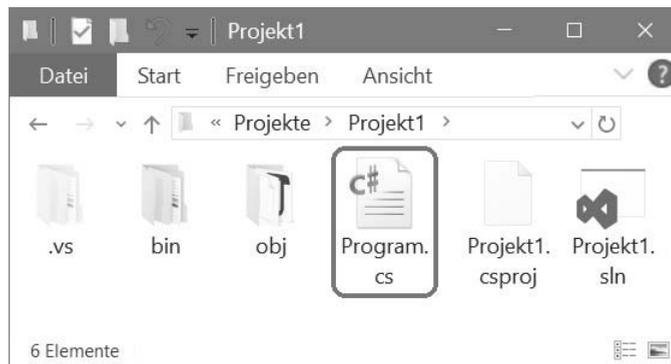
Microsoft Visual Studio-Debugging-Konsole
Hallo, wer da?

D:\Projekte\Projekt1\bin\Debug\net6.0\Projekt1.exe wurde mit Code "0" beendet.
Drücken Sie eine beliebige Taste, um dieses Fenster zu schließen.

```

Und wie du siehst, zeigt das Programm nun **deinen** Text an.

Das soll alles sein? Schauen wir mal im PROJEKTE-Ordner nach. Dort liegt ein Unterordner mit dem Namen PROJEKT1. Wenn du den öffnest, erwartet dich ein solches Bild:



Da liegt ja einiges an Dateien herum und in den Unterordnern findest du noch mehr. All das hat Visual Studio automatisch erzeugt, es muss dich aber nicht weiter interessieren, denn die Datei, um die es für dich geht, heißt PROGRAM.CS.



»CS« ist die Kennung für Dateien mit C#-Quelltext und kürzt »C-sharp« ab, so spricht man C# auch aus. Wenn du diese Datei mit einem normalen Texteditor öffnest, findest du darin die obigen Programmzeilen.

Und nun klären wir erst einmal, was diese eine Textzeile, aus der unser Programm besteht, bedeutet. Den Kommentar darüber kann man entsorgen, also löschen. Oder du trägst hinter die zwei Schrägstriche einen eigenen Infotext ein.

Es handelt sich hier um eine **Anweisung**, die den Computer dazu bringt, dich freundlicher zu grüßen:

```
Console.WriteLine("Hallo, wer da?");
```

Fangen wir bei dieser Anweisung von hinten an. Dort steht der Text, der angezeigt werden soll, eingepackt in Anführungsstriche und zusätzlich in runden Klammern:

```
("Hallo, wer da?")
```

Für die Anzeige sorgt die Anweisung `WriteLine()`, was ausführlich heißt: »Schreib etwas auf dem Bildschirm und gehe danach in die nächste Zeile.«

Es ist übrigens nicht egal, ob für die Wörter große oder kleine Buchstaben benutzt werden. C# unterscheidet eindeutig zwischen Groß- und Kleinschreibung.

Achte also beim Eintippen genau darauf, wann mal ein großer Buchstabe zwischen den vielen Kleinbuchstaben steht. Du kannst also nicht z.B. `writeline` oder `WriteLine` schreiben!



Eine solche Anweisung nennt man auch Methode. In C# gibt es zahlreiche Methoden, sie gehören immer zu einem sogenannten Objekt oder zu einer Klasse – hier `Console` heißt. Damit ist das Fenster mit dem schwarzen Hintergrund gemeint, das nur Text anzeigen kann.

**Objekte** kennen wir aus unserer Umgebung, z.B. Häuser, Bäume, Autos, Leute. Auch du bist ein Objekt. Und zwar vom Typ `Mensch`. Objekte in einer Programmiersprache sind natürlich nur künstlich.

Dabei kann es in C# durchaus mehrere Objekte eines Typs geben – so wie es im richtigen Leben auch (viele) verschiedene Menschen gibt. Daher spricht man hier von **Objekttyp**, das ist dasselbe wie **Klasse**. Und ein Objekt wird auch als **Instanz** einer Klasse bezeichnet.

Demnach bist du eine Instanz der Klasse `Mensch`. Und mit `Console` kennst du ein Beispiel für eine C#-Klasse. Objekte lernst du später noch kennen.

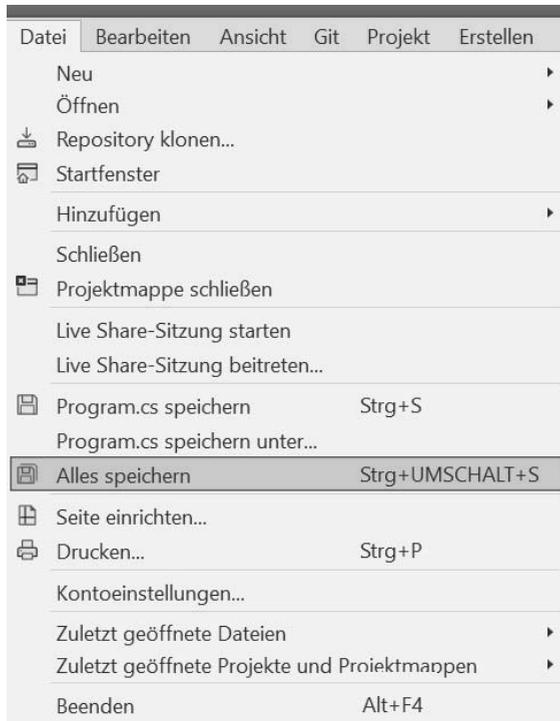


Man spricht bei `WriteLine()` auch von einer **Funktion**. Methode und Funktion meinen also hier dasselbe. Und das, was in den runden Klammern steht, wird als **Parameter** bezeichnet.

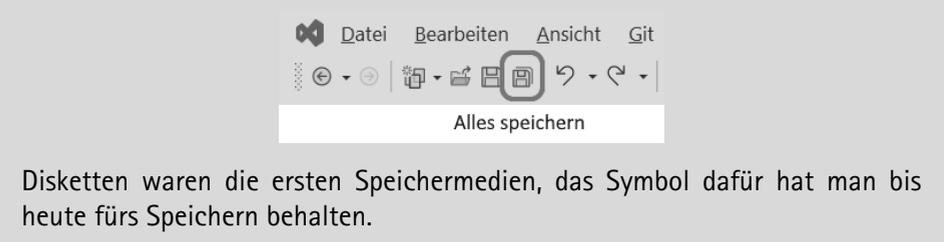


Wichtig ist: Jede (!) Methode beziehungsweise Funktion hat am Ende ihres Namens zwei runde Klammern, die können auch leer sein, sie dürfen aber niemals fehlen.

- Speichere dein Projekt jetzt erst mal. Dazu klickst du auf ALLES SPEICHERN im DATEI-Menü, damit werden alle beteiligten Dateien gesichert.



Alternativ dazu kannst du auch in der Symbolleiste auf den rechten der beiden Buttons mit dem Disketten-Zeichen klicken.



Disketten waren die ersten Speichermedien, das Symbol dafür hat man bis heute fürs Speichern behalten.

Diese Leseprobe haben Sie beim  
 **edv-buchversand.de** heruntergeladen.  
Das Buch können Sie online in unserem  
Shop bestellen.

[Hier zum Shop](#)