
Daniel Braun
2. Auflage
mitp
Für Bukkit & Spigot
Unter Windows, Linux und macOS
LET'S PLAY
Komplett in Farbe
Programmieren lernen mit Python und Minecraft
Plugins erstellen ohne Vorkenntnisse
KEIN OFFIZIELLES MINECRAFTPRODUKT.
NICHT VON MOJANG GENEHMIGT ODER
MIT MOJANG VERBUNDEN.

Inhalt

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Einleitung

Liebe Leserin, lieber Leser,

die Welt von Minecraft steckt voller Dinge, die es zu entdecken gilt. Verschiedene Landschaften, Hunderte verschiedene Gegenstände und allerlei Tiere und Monster sind nur einige der Dinge, die dich erwarten.

Irgendwann ist aber selbst diese Vielzahl an Möglichkeiten erschöpft und man hat das Gefühl, alles schon einmal gesehen oder gemacht zu haben. Wenn es dir so geht, dann ist dieses Buch genau richtig für dich. Denn im Verlaufe dieses Buches lernst du, wie man mithilfe von Python und dem Bukkit- oder Spigot-Server eigene Erweiterungen für die Java-Edition von Minecraft programmiert, sogenannte Plugins, die du dann zusammen mit deinen Freunden auf deinem eigenen Minecraft-Server ausprobieren kannst.

Egal ob du neue Crafting-Rezepte entwerfen, ganze Häuser mit einem einfachen Chat-Befehl bauen oder sogar einen eigenen Spielmodus programmieren möchtest, mit eigenen Plugins steckt die Welt von Minecraft wieder voller Herausforderungen und Dinge, die entdeckt werden wollen. Und ganz nebenbei lernst du auch noch zu programmieren – und wer weiß, vielleicht kommt das nächste Minecraft eines Tages von dir!

Bevor es soweit ist, liegt allerdings noch ein ordentliches Stück Weg vor dir. Die ersten beiden Kapitel dieses Buches beschäftigen sich deshalb zunächst einmal damit, wie du deinen Computer für das Programmieren und Testen eigener Plugins vorbereitest. Dazu wird dir erklärt, wie du den Bukkit- oder Spigot-Server installierst, der in diesem Buch verwendet wird, ihn nach deinen Wünschen konfigurierst und wie du deinen Computer so einrichtest, dass du Python-Plugins schreiben kannst.

Direkt im Anschluss geht es im dritten Kapitel ohne Umschweife direkt los mit dem Programmieren deines ersten eigenen Plugins. Die ersten Schritte werden dir vielleicht noch etwas unspektakulär vorkommen, aber mit jedem der folgenden Kapitel wirst du immer mehr Möglichkeiten besitzen, um immer ausgeklügeltere Plugins zu programmieren. Schon im vierten Kapitel wirst du zum Beispiel lernen, wie du eigene Chat-Befehle programmieren und verwenden kannst.

Die Kapitel 5 bis 8 beschäftigen sich mit grundlegenden Konzepten des Programmierens im Allgemeinen und der Programmiersprache Python im Besonderen. Was du hier liest, wird dir nicht nur beim Programmieren von Minecraft-Plugins helfen, sondern beim Programmieren jedes Programms in jeder Programmiersprache. Trotzdem entstehen dabei natürlich auch einige praktische kleine Plugins, wie zum Beispiel das Mauer-Plugin, das es dir erlaubt, mit einem einfachen Chat-Befehl auf die Schnelle eine Mauer zu bauen, wenn du möchtest, sogar aus purem Gold.

Das neunte Kapitel widmet sich dann ganz der Baukunst. Häuser, Kreise und Kugeln – hier wird kein Block auf dem anderen gelassen. Und wenn du schon einmal versucht

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

hast, eine Kugel in Minecraft von Hand zu bauen, dann wirst du ganz besonders die Dienste des Kugel-Plugins zu schätzen wissen, das dir auf Knopfdruck eine nahezu perfekte Kugel zaubern kann. Weiter geht es danach mit dem Bau von Schildern, denen das gesamte zehnte Kapitel gewidmet ist.

Wenn dir selbst ein Knopfdruck zum Bauen noch zu viel ist, dann wird dir das elfte Kapitel besonders gefallen. Dort geht es nämlich um Plugins, die vollautomatisch auf Geschehnisse in der Spielwelt reagieren. Egal ob ein Creeper über die Karte schleicht, ein Spieler etwas isst oder ein Baum wächst, hier lernst du wie deinem Plugin nichts mehr von dem entgeht, was auf deinem Server passiert und natürlich auch, wie du darauf reagieren kannst.

Ein sehr grundlegendes und wichtiges Konzept moderner Programmiersprachen, die objektorientierte Programmierung, lernst du in Kapitel 12 kennen, hier dreht sich alles um Objekte und Klassen.

Falls du dich um die umherschleichenden Creeper aber doch lieber ganz manuell kümmern möchtest, kannst du die Informationen aus Kapitel 13 nutzen, um ganz eigene Waffen zu kreieren. In diesem Kapitel geht es nämlich um das Erstellen eigener Crafting-Rezepte und ein Beispiel, das dir dort begegnen wird, ist ein Rezept für ein Flammenschwert, das alles in Brand setzt, worauf es trifft.

Das vierzehnte Kapitel ist dann wieder etwas technischer, aber nicht weniger nützlich. Hier lernst du nämlich, wie du Informationen dauerhaft speichern kannst, die auch dann erhalten bleiben, wenn der Server zwischenzeitlich ausgeschaltet wird. Das ist zum Beispiel praktisch, wenn du wie in Kapitel 15 eigene Spielmodi kreieren willst, also sozusagen ein Spiel im Spiel. Wir wäre es zum Beispiel mit einem Schneeballschlacht-Mod mit eigener Highscore-Liste, die die Treffer zählt? Oder lieber ein lustiges Suchspiel, bei dem der Gewinner mit Erfahrungspunkten oder wertvollen Gegenständen belohnt wird? Ganz wie du möchtest: Deiner Kreativität sind keine Grenzen gesetzt!

Im letzten Kapitel bekommst du dann noch einen kurzen Ausblick darauf, was du mit deinen neu gewonnenen Programmierfähigkeiten noch anstellen kannst, außer Minecraft-Plugins zu programmieren. Denn wenn du am Ende des Buches angelangt bist, hört der Spaß noch lange nicht auf: Nun hast du alle Werkzeuge und alles Wissen, das du benötigst, um ganz eigene Plugins ganz nach deinen Vorstellungen zu entwerfen. Dabei helfen dir einige Listen im Anhang des Buches, in denen du Befehle und besonders häufig benötigte Dinge schnell nachschlagen kannst. Denn egal wie erfahren man als Programmierer ist, alles kann und muss man nicht auswendig können, man muss nur wissen, wo man es nachschlagen kann – und genau dazu dient der Anhang dieses Buches.

Falls du Fragen, Kritik oder Anregungen zum Buch oder generell zu Minecraft-Plugins hast, kannst du mich gerne jederzeit kontaktieren. Du erreichst mich per Mail an *info@daniel-braun.com* oder über meine Website *www.daniel-braun.com*.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Downloads zum Buch

Unter der Webadresse *buch.daniel-braun.com* findest du:

- Links zu allen Downloads, die du benötigst
- Alle Plugins, die du im Rahmen des Buches programmieren wirst, falls du den Code nicht aus dem Buch abtippen möchtest

Mein besonderer Dank gilt Karl-Heinz Barzen, der den Entstehungsprozess dieses Buches unermüdlich mit zahlreichen hilfreichen Kommentaren und Anmerkungen begleitet und damit einen wichtigen Beitrag dazu geleistet hat, dass dieses Buch möglichst verständlich und einsteigerfreundlich wird.

Nun wünsche ich dir aber vor allem viel Spaß beim Lesen, Programmieren und Entdecken!

Daniel Braun

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Kapitel 2

Python

Ob bewusst oder unbewusst, eine der wichtigsten Entscheidungen, die man auf dem Weg zum Programmierer zu treffen hat, hast du bereits getroffen: welche Programmiersprache du lernen möchtest. Mit diesem Buch hast du dich nämlich für die Programmiersprache Python entschieden. Bevor wir aber einen Blick auf die Besonderheiten von Python werfen, soll es zunächst um die Frage gehen, was Programmiersprachen eigentlich sind und warum sie benötigt werden.

2.1 Programmiersprachen

Beim Programmieren geht es im Wesentlichen darum, dass der Programmierer dem Computer eine bestimmte Aufgabe gibt, die dieser erledigen soll. Damit er das kann, braucht der Computer eine genaue Handlungsvorschrift, die auch *Algorithmus* genannt wird. Auch im Alltag begegnen uns oft Handlungsvorschriften, zum Beispiel in Form eines Rezepts:

1. 250 Gramm Mehl in eine Schüssel geben
2. 500 Milliliter Milch dazugeben
3. 2 Eier hinzugeben
4. Mit einer Prise Salz würzen
5. Umrühren

Fertig ist der Crêpes-Teig! Damit eine Handlungsvorschrift korrekt ausgeführt werden kann, müssen sich beide Seiten auf eine gemeinsame Sprache einigen. Wenn dir jemand ein chinesisches Rezept gibt, kannst du vermutlich nicht viel damit anfangen.

Computer »sprechen« in Einsen und Nullen, also in einer Sprache, mit der Menschen nicht besonders gut umgehen können. Unsere Sprache wiederum, egal ob es sich um Deutsch, Englisch oder Chinesisch handelt, ist für den Computer viel zu ungenau. Nehmen wir zum Beispiel den Satz »Da vorne ist eine Bank.« Obwohl es sich dabei um einen vollkommen korrekten deutschen Satz handelt, ist doch nicht eindeutig klar, was mit dem Satz eigentlich gemeint ist. Steht da vorne eine Parkbank, auf die man sich setzen kann, oder ist dort die Filiale einer Bank, auf der man Geld einzahlen und abheben kann? Es wäre ein recht kostspieliger Fehler, wenn dein Computer beim Online-Shoppen aus Versehen die Deutsche Bank statt einer Bank für den Garten kauft.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Algorithmen müssen deshalb nicht nur Handlungsvorschriften sein, sondern *eindeutige Handlungsvorschriften*. Auch mit Begriffen wie »eine Prise« kann ein Computer wenig anfangen, weil sie einfach zu ungenau sind. Aus diesem Grund nutzen wir Programmiersprachen, denn sie ermöglichen es uns, eindeutige Handlungsvorschriften festzulegen. Und obwohl sie auf den ersten Blick recht kompliziert scheinen, können wir sie doch leichter lernen als eine Sprache aus Nullen und Einsen.

Viele Programmiersprachen, wie zum Beispiel C, verwenden sogenannte Compiler, die die Programmiersprache in Maschinensprache übersetzen. Andere, wie auch Python, verwenden einen sogenannten Interpreter. Wir werden in diesem Buch einen besonderen Interpreter mit dem Namen Jython verwenden. Ein Vorteil des Interpreters ist, dass er den Programmcode bei jedem Ausführen neu »interpretiert«, also für den Computer übersetzt, Deshalb werden Änderungen am Programmcode bei einem Interpreter sofort wirksam, während man bei einer Compilersprache den Quellcode zuerst neu kompilieren muss, bevor die Änderungen auch im Programm ankommen. Außerdem sind Interpretersprachen unabhängiger vom Betriebssystem, deine fertigen Plugins kannst du daher später auch problemlos an Freunde weitergeben, egal ob sie Windows, GNU/Linux oder OS X nutzen.

Merke

- Ein *Algorithmus* ist eine eindeutige Handlungsvorschrift.
- Ein *Interpreter* übersetzt Programmiersprache bei Ausführung in Maschinensprache.

2.2 Besonderheiten von Python

Python wurde Anfang der 1990er-Jahre vom niederländischen Informatiker Guido van Rossum entwickelt. Die Programmiersprache ist heute weit verbreitet und dafür bekannt, dass sie besonders leicht zu erlernen ist, genau die richtige Sprache also für den Einstieg in die Programmierung.

Python gilt aber auch als besonders übersichtlich und minimalistisch. Gut sichtbar wird das am sogenannten »Hallo Welt!«-Programm. Programmierer nutzen »Hallo Welt!«-Programme gerne, um auf besonders einfache Weise den Aufbau einer Programmiersprache zu verdeutlichen, denn dieses kleine Programm tut nichts anderes, als den Text »Hallo Welt!« auf dem Bildschirm auszugeben.

In Listing 2.1 kannst du den Code des »Hallo Welt!«-Programms in Java sehen, also der Programmiersprache, in der auch Minecraft geschrieben wurde. Das sieht schon ziemlich kompliziert aus, es gibt viele Wörter und auffällig viele sogenannte geschweifte Klammern, also { und }. Insgesamt hat das Programm eine Länge von fünf Zeilen.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

```
1 class Hallo {
2     public static void main(String[] args) {
3         System.out.print("Hallo Welt!");
4     }
5 }
```

Listing 2.1: »Hallo Welt!« in Java

Zeilen und Zeilennummern, das wirst du später noch sehen, sind für Programmierer wichtige Informationen, denn wenn sie auf eine bestimmte Stelle in einem Programm hinweisen wollen, so tun sie das meist mithilfe der Zeilennummer. Oft sagt die Länge eines Programmes auch etwas über die Komplexität aus, zumindest aber über den Aufwand, deshalb sind Programmierer oft bemüht, Programme möglichst kurz zu halten.

In Listing 2.2 kannst du den Code des »Hallo Welt!«-Programms in der Programmiersprache C sehen. Obwohl das Programm wegen anderer Formatierung, sogar eine Zeile länger ist sieht man doch sofort, dass es schon deutlich übersichtlicher ist als sein Java-Pendant, selbst wenn man nicht weiß, wofür die Befehle eigentlichen stehen.

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     puts("Hallo Welt!");
6 }
```

Listing 2.2: »Hallo Welt!« in C

Formatierung, also zum Beispiel das Einrücken von Zeilen, oder das Beginnen einer neuen Zeile, ist beim Programmieren extrem wichtig, das gilt für Python noch viel mehr als für jede andere Programmiersprache. Denn wie wir später noch sehen werden, macht die Einrückung einer Zeile in Python nicht nur einen optischen Unterschied, sondern kann auch die Funktionalität eines Programmes verändern.

Die Python-Version des »Hallo Welt!«-Programms siehst du in Listing 2.3. Mit gerade einmal einer Zeile ist es ohne Zweifel die kürzeste der drei Versionen. Auffallend ist auch die Abwesenheit von Sonderzeichen wie dem Semikolon und den geschweiften Klammern, die es in den beiden anderen Programmiersprachen gibt.

```
print("Hallo Welt!")
```

Listing 2.3: »Hallo Welt!« in Python

Wie du also sehen kannst, wird dir Python das Leben als Programmierer nicht unnötig schwer machen, ganz im Gegenteil, es wird dir einen einfacheren Einstieg ermöglichen als die meisten anderen Programmiersprachen. Und obwohl die Arbeit mit Python vergleichsweise einfach ist, wirst du trotzdem im Verlaufe des Buches alle wichtigen

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Programmiertechniken kennenlernen, wie sie auch die anderen Programmiersprachen bieten. Da sich diese oft kaum zwischen den verschiedenen Sprachen unterscheiden, wird es dir, wenn du erst einmal Python gelernt hast, später nicht schwer fallen, noch weitere Programmiersprachen zu lernen, falls du das möchtest.

2.3 Einrichtung

Wie bereits erwähnt, ist Minecraft selbst in Java programmiert. Daher benötigst du, bevor du Python-Plugins auf deinem Server ausführen kannst, einen Interpreter, der deinen Python-Code in Java übersetzt, und ein Plugin, das diesen Interpreter an deinen Server koppelt. Beides findest du auf der Website zum Buch, unter *buch.daniel-braun.com* zum Download.

2.3.1 Jython

Zunächst solltest du von dort den Interpreter herunterladen, dieser trägt den Namen **Jython**, ein Kofferwort aus Java und Python. Jython besteht lediglich aus einer einzelnen Datei, diese trägt den Namen `jython.jar`. Wenn du die Datei heruntergeladen hast, musst du nur noch einen neuen Ordner mit dem Namen `lib` in deinem Serververzeichnis erzeugen und die Datei dorthin kopieren. Damit ist die Einrichtung des Interpreters bereits abgeschlossen.

2.3.2 PPLoader

Nun fehlt nur noch das Plugin, das den Interpreter mit deinem Server, egal ob Bukkit oder Spigot, verbindet. Das benötigte Plugin trägt den Namen **PPLoader**, was für *Python Plugin Loader* steht. Auch das Plugin besteht lediglich aus einer einzigen Datei, sie trägt den Namen `pploader1.2.0.jar`. Sie muss nach dem Download in den im Server-Ordner bereits vorhandenen Ordner `plugins` kopiert werden.

Merke

Bevor du weiterliest, solltest du noch einmal überprüfen, ob du alle nötigen Schritte ausgeführt hast, um deinen Server auf die Verwendung von Python-Plugins vorzubereiten:

1. Jython herunterladen von *buch.daniel-braun.com*
2. Im Server-Ordner ein neues Verzeichnis `lib` anlegen und die `jython.jar` dorthin kopieren
3. PPLoader herunterladen von *buch.daniel-braun.com*
4. Die Datei `pploader1.2.0.jar` in den `plugins`-Ordner im Server-Verzeichnis kopieren

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Um zu testen, ob alles funktioniert hat, kannst du den Server über die Eingabeaufforderung beziehungsweise das Terminal mit dem Befehl `java -jar craftbukkit-1.15.2.jar`, beziehungsweise `java -jar spigot-1.15.2.jar`, starten. Funktioniert alles ohne Probleme, so wird im Server-Fenster die in Abbildung 2.1 markierte Zeile angezeigt, die bestätigt, dass der PPLoader erfolgreich geladen wurde.

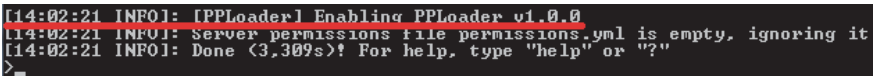

Abbildung 2.1: Erfolgreich geladener PPLoader

2.4 Editor

Bevor wir uns endgültig dem Programmieren von Plugins widmen, soll zunächst noch die Frage geklärt werden, welches Programm man eigentlich verwendet, um Python-Plugins zu schreiben. Grundsätzlich kannst du dazu nahezu jedes Programm verwenden, mit dem man Texte verfassen kann. Der mit Windows mitgelieferte EDITOR, den du im Startmenü unter ZUBEHÖR findest, ist zum Beispiel prinzipiell ausreichend. OS X bringt das Programm TEXTEDIT mit, das sich im Ordner Programme befindet oder über die Suche gefunden werden kann.

Solltest du dich für den Windows-Editor entscheiden, so musst du beim SPEICHERN darauf achten, dass du als DATEITYP den Eintrag ALLE DATEIEN auswählst. Beim Programm TEXTEDIT sollte nach dem Neuanlegen eines Dokuments der Menüpunkt FORMAT | IN REINEN TEXT UMWANDELN ausgewählt werden. Und unabhängig davon, welches Programm du verwendest, solltest du beim Speichern an den Dateinamen die Endung `.py` anhängen, damit dein Computer weiß, dass es sich bei der gespeicherten Datei um Python-Code handelt.

Du kannst dir das Leben aber auch deutlich einfacher machen, indem du einen Editor verwendest, der speziell für das Schreiben von Python-Code ausgelegt ist. Solche Editoren bieten oft zahlreiche Vorteile wie automatische Vervollständigung, farbliche Hervorhebung und weitere Komfortfunktionen. Im Internet kannst du dutzende, wenn nicht gar hunderte solcher Editoren finden. **Eclipse** und **PyCharm** sind zum Beispiel nur zwei bekanntere Vertreter. Für Einsteiger sehr empfehlenswert ist **Komodo Edit**, einen Link zum kostenlosen Download, für Windows, GNU/Linux und OS X, findest du auf der Website zum Buch unter *buch.daniel-braun.com*. Komo Edit bietet zahlreiche Komfortfunktionen, ist aber nicht so überladen wie zum Beispiel Eclipse und eignet sich daher besonders für Anfänger.

Nach dem Download der Installationsdatei kann diese wie gewohnt mit einem Doppelklick geöffnet werden. Daraufhin öffnet sich die in Abbildung 2.2 gezeigte Installationsroutine.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

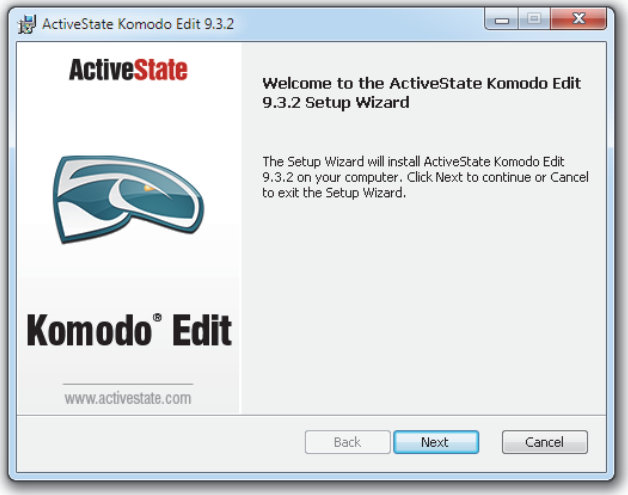

Abbildung 2.2: Schritt 1 der Installation

Dort kannst du mit einem Klick auf den Button NEXT zum zweiten Schritt der Installation springen, der in Abbildung 2.3 gezeigt ist.

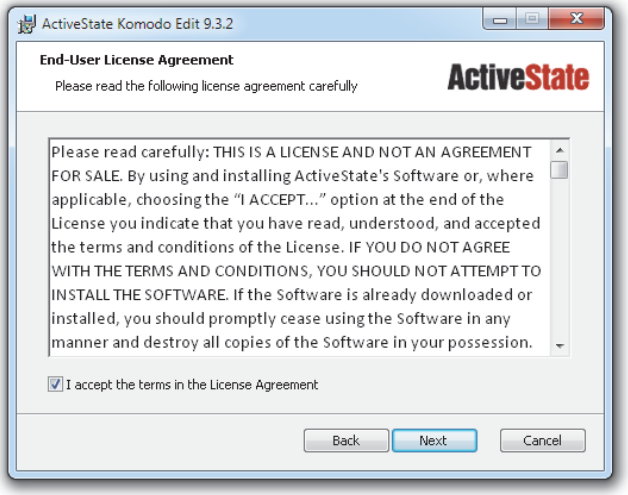

Abbildung 2.3: Schritt 2 der Installation

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Hier werden nun die Lizenzbestimmungen angezeigt, die, nach ausführlicher Lektüre, mit einem Klick auf die Box neben I ACCEPT THE TERMS IN THE LICENSE AGREEMENT bestätigt werden müssen, damit der Editor installiert werden kann.

Ein erneuter Klick auf den NEXT-Button bringt dich dann zum dritten und vorletzten Schritt der Installation, der in Abbildung 2.4 gezeigt ist.

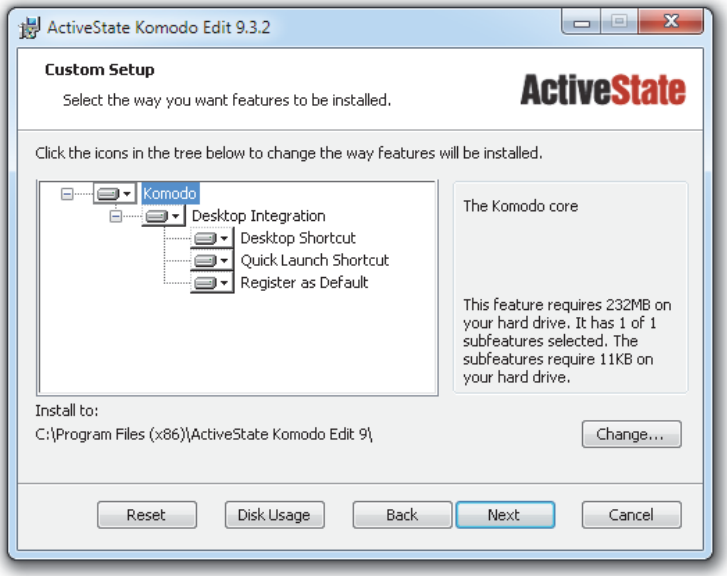

Abbildung 2.4: Schritt 3 der Installation

Hier kannst du nun noch auswählen, ob du ein Desktop-Icon und eine Startmenü-Verknüpfung einrichten möchtest. Auch dieses Fenster kannst du wieder mit einem Klick auf den NEXT-Button verlassen, was dich zum letzten Schritt der Installation führt. Hier musst du nun nur noch den Button INSTALL drücken und schon wird Komodo Edit auf deinen Computer installiert.

Beim ersten Starten des Programms erwartet dich dann das in Abbildung 2.5 gezeigte Fenster.

Neben der Menüleiste am oberen Bildschirmrand besteht es im Wesentlichen aus zwei Bereichen. Auf der linken Seite werden dir die Dateien und Ordner auf deinem Computer angezeigt, so kannst du schnell und bequem auf verschiedene Dateien zugreifen, im großen, dunklen Bereich auf der rechten Seite werden später geöffnete Datei angezeigt.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

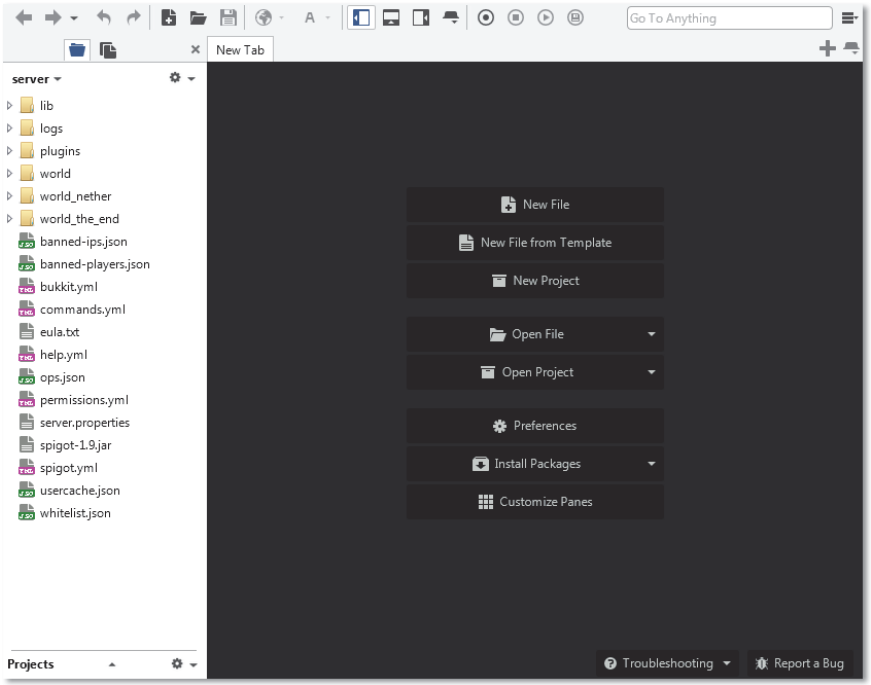

Abbildung 2.5: Startbildschirm von Komodo Edit

Um eine neue Datei anzulegen, kannst du entweder in diesem Bereich auf NEW FILE klicken, oder oben in der Menüleiste auf das Blatt mit dem Pluszeichen. In Abbildung 2.6 kannst du sehen, wie Python-Code in Komodo Edit dargestellt wird.

Der Inhalt ist an dieser Stelle noch egal, aber im Vergleich zum Beispiel zum Windows Editor sticht hier sofort die farbliche Hervorhebung ins Auge, außerdem gibt es eine Zeilennummerierung und mithilfe der kleinen Dreiecke am linken Rand lassen sich einzelne Teile des Codes ein- und ausklappen.

Noch mögen dir diese Unterschiede vielleicht banal erscheinen, aber je länger du programmierst, desto mehr wirst du diese kleinen Annehmlichkeiten schätzen lernen. Für den weiteren Verlauf des Buches ist es jedoch egal, für welchen Editor du dich entscheidest, denn in den folgenden Kapiteln werden wir uns ganz auf den Inhalt konzentrieren und der ist, unabhängig vom Editor, immer gleich.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

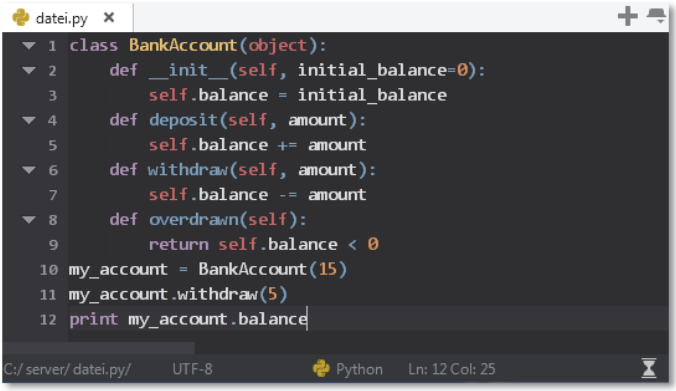

Abbildung 2.6: Python-Code in Komodo Edit

Auch bei Komodo Edit musst du beim Speichern von Dateien darauf achten, die korrekte Endung auszuwählen. Dateien speichern kannst du wie gewohnt über das Disketten-Symbol in der oberen Symbolleiste. Daraufhin öffnet sich der in Abbildung 2.7 gezeigte Speicherdialog. Als DATEITYP musst du hier den Eintrag ALL FILES auswählen und an den DATEINAMEN die Endung `.py` anhängen.

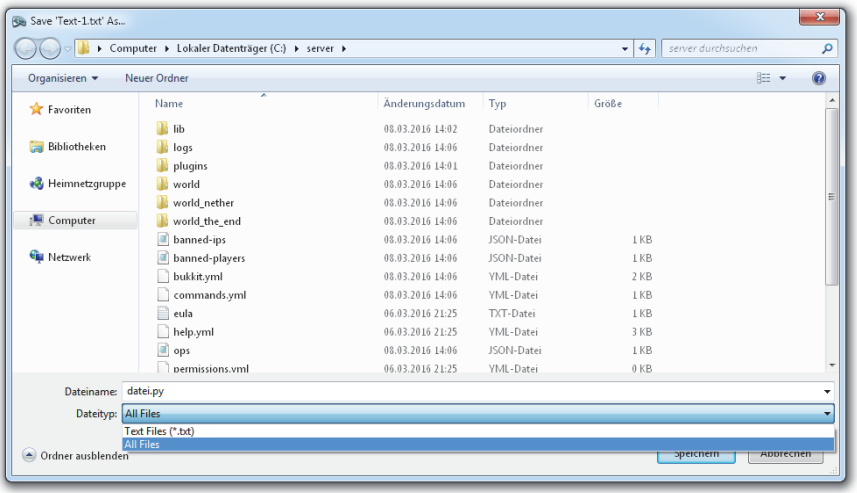

Abbildung 2.7: Speicherdialog

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Damit aber auch genug der Vorbereitungen, im nächsten Kapitel wirst du nun endlich dein erstes eigenes Plugin programmieren lernen. Und dank der bereits getroffenen Vorbereitungen wird das vermutlich schneller gehen, als du denkst.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Kapitel 3

Das erste Plugin

Wie bereits im letzten Kapitel erwähnt, bieten »Hallo Welt!«-Programme einen einfachen und schnellen Einstieg in eine Programmiersprache. Deshalb wird auch unser erstes Plugin ein »Hallo Welt!«-Plugin.

3.1 Ordner anlegen

Zunächst solltest du für jedes Plugin, das du programmieren möchtest, einen neuen Ordner anlegen. Jedes Plugin muss sich in einem eigenen Unterordner innerhalb des `plugins`-Ordners im Serververzeichnis befinden. Der Name des Unterordners muss immer mit `.py.dir` enden, davor kann ein beliebiger Name stehen, allerdings ohne Leer- oder Sonderzeichen. Für das »Hallo Welt!«-Plugin könntest du den Ordner also zum Beispiel `hallowelt.py.dir` nennen.

Merke

Jedes Plugin muss sich in einem eigenen Unterordner im `plugins`-Ordner des Serververzeichnisses befinden. Der Name dieser Unterordner muss immer mit `.py.dir` enden und darf keine Leer- oder Sonderzeichen enthalten.

3.2 plugin.py

Im nächsten Schritt solltest du dann im Editor deiner Wahl eine neue Datei anlegen und sie mit den Inhalt aus Listing 3.1 füllen. Diese Datei musst du dann im eben angelegten Ordner `hallowelt.py.dir` unter dem Namen `plugin.py` speichern.

```
1 class HalloWeltPlugin(PythonPlugin):
2     def onEnable(self):
3         self.getLogger().info("Hallo Welt!")
```

Listing 3.1: »Hallo Welt!«-Plugin

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Tipp

Es ist eine gute Übung, die Programme aus dem Buch abzutippen, da du dir so die verschiedenen Befehle schneller einprägen kannst. Allerdings musst du darauf achten, dabei sehr exakt vorzugehen, denn schon ein einziger Tippfehler kann dafür sorgen, dass dein Programm nicht mehr funktioniert. Möchtest du die Programme nicht abtippen, so findest du sie, wie bereits in der Einleitung erwähnt, unter *buch.daniel-braun.com* zum Download.

Das sind auf den ersten Blick erst einmal ziemlich viele Wörter und Zeichen, die wenig Sinn ergeben. Der Text in Listing 3.1 wird Programmcode, Sourcecode oder Quellcode genannt. Er enthält die Handlungsvorschriften, die der Computer ausführen soll. In diesem Fall also die Vorschrift: Zeige beim Starten des Servers die Nachricht »Hallo Welt!« an. Im Laufe der nächsten Kapitel wirst du alle Befehle genau kennen und verstehen lernen. Für den Anfang soll aber zunächst einmal ein grober Überblick reichen.

Wie dir sicher schon aufgefallen ist, ist das Plugin mit gerade einmal drei Zeilen zwar immer noch sehr kurz, aber immerhin schon einmal zwei Zeilen länger als das »Hallo Welt!«-Programm in Python aus dem letzten Kapitel. In der ersten Zeile wird der sogenannte Klassenname auf `HalloWeltPlugin` festgelegt, für den Anfang kannst du dir diesen einfach wie einen ganz normalen Namen vorstellen, mit dessen Hilfe das Plugin gerufen werden kann. Theoretisch gibt es für diesen Namen außer einem Verbot für Leer- und Sonderzeichen kaum Beschränkungen. In der Praxis gibt es unter Python-Programmierern aber einige ungeschriebene Gesetze oder Konventionen, die du beachten solltest. Zunächst einmal sollte der Name aussagekräftig sein, also eine gute Beschreibung dessen, was das Plugin tut. Außerdem sollten Klassennamen immer mit einem Großbuchstaben beginnen. Wenn sie sich, wie bei `HalloWeltPlugin`, aus mehreren Wörtern zusammensetzen, sollte außerdem der Anfangsbuchstabe jedes Wortes großgeschrieben werden, um die Lesbarkeit zu verbessern.

Merke

Konventionen für Klassennamen:

- Aussagekräftig
- Beginnen mit einem Großbuchstaben
- Bei einer Zusammensetzung mehrerer Wörter wird der erste Buchstabe jedes Wortes großgeschrieben
- Keine Verwendung von Leer- und Sonderzeichen

In den Klammern hinter dem Klassennamen steht außerdem noch `PythonPlugin`, das sagt dem Computer, dass es sich bei diesem Python-Code um ein Minecraft-Python-Plugin handelt.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

In der zweiten Zeile wird die sogenannte `onEnable`-Funktion definiert. Was genau eine Funktion beim Programmieren ist, damit werden wir uns später noch ausführlich beschäftigen. `onEnable` ist Englisch und bedeutet so viel wie »beim Start«. Hier kannst du dem Plugin sagen, was es machen soll, wenn es gestartet wird. Wie du sehen kannst, ist die zweite Zeile eingerückt, sie steht also weiter rechts als die vorherige Zeile; das ist wichtig zu beachten, denn nur so funktioniert dein Plugin später auch. In Python zeigen solche Einrückungen nämlich an, dass die entsprechenden Zeilen zu etwas gehören. In diesem Fall also zum Beispiel, dass die Funktion `onEnable` zum Plugin mit dem Namen `HalloWeltPlugin` gehört. Wie weit man die Zeilen genau einrückt, ist eine Geschmacksfrage, allerdings sollte man zumindest konsistent bleiben, also die Zeilen immer gleich weit einrücken. In diesem Buch werden wir jeweils vier Leerzeichen zur Einrückung verwenden.

In der dritten und letzten Zeile steht nun endlich, was das Plugin machen soll, nämlich den Text »Hallo Welt!« anzeigen.

3.3 plugin.yml

Bevor du dein erstes Plugin testen kannst, benötigst du noch eine zweite Datei, die sogenannte `plugin.yml`. Diese besteht, wie du in Listing 3.2 sehen kannst, ebenfalls nur aus drei Zeilen.

```
1 name: Hallo Welt-Plugin
2 main: HalloWeltPlugin
3 version: 1.0
```

Listing 3.2: `plugin.yml` für das Hallo Welt!«-Plugin

Die `plugin.yml` gibt dem Server wichtige Informationen über dein Plugin, die er schon vor dessen Start benötigt. In der ersten Zeile wird mit dem Stichwort `name` der Name des Plugins festgelegt. Im Gegensatz zum Klassennamen kannst du deiner Kreativität hier freien Lauf lassen, auch Leer- und Sonderzeichen können verwendet werden.

Wichtig wird der Klassenname aber wieder in der zweiten Zeile, dort muss dieser nämlich hinter dem Stichwort `main` eingetragen werden und zwar exakt so, wie er in der Datei `plugin.py` steht, nur so weiß der Server später, wie er dein Plugin ausführen kann.

Zu guter Letzt kannst du noch eine Versionsnummer für dein Plugin angeben, hinter dem Stichwort `version`. Solltest du eines deiner Plugins später einmal verändern oder weiterentwickeln, so kannst du – und andere, die es benutzen – mit der Versionsnummer leicht den Überblick behalten, ob eine Kopie deines Plugins schon auf dem neuesten Stand ist oder noch nicht.

Merke
Jedes Plugin besteht aus mindestens zwei Dateien, der `plugin.py` und der `plugin.yml`.

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

3.4 Testen

Nun wird es aber Zeit, dein erstes eigenes Plugin zu testen. Praktischerweise musst du dafür einfach nur den Server starten. Solltest du Bukkit benutzen, musst du also den Befehl `java -jar craftbukkit-1.15.2.jar` eingeben, benutzt du dagegen Spigot, so lautet der Befehl `java -jar spigot-1.15.2.jar`. Der PPLoader durchsucht dann automatisch den `plugins`-Ordner nach Ordnern mit der Endung `.py.dir` und öffnet die darin befindlichen Plugins.

Was du dann auf dem Bildschirm sehen kannst, wenn der Server alles geladen hat, kannst du in Abbildung 3.1 sehen. Besonders interessant ist dabei die drittletzte Zeile, denn hier kannst du dein Plugin in Aktion sehen, die Meldung »Hallo Welt!« wird angezeigt. Zugegebenermaßen ist das noch nicht besonders spektakulär, aber aller Anfang ist ja bekanntlich schwer und du hast damit die ersten Schritte auf dem Weg, ein Programmierer zu werden, gemacht. Mit jedem der folgenden Kapitel wirst du neue Dinge lernen, die es dir ermöglichen werden, immer spannendere Plugins zu programmieren.

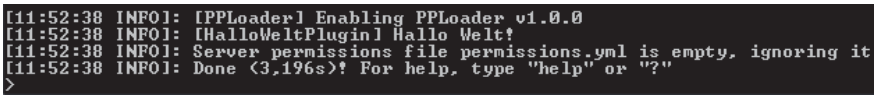

Abbildung 3.1: Ausgabe des »Hallo Welt«-Plugins nach dem Start

3.5 Fehler finden

Sollte beim Starten deines Servers nicht der in Abbildung 3.1 gezeigte Text angezeigt werden, dann liegt das höchstwahrscheinlich daran, dass du in deinem Plugin einen Tippfehler hast. Computer nehmen es ziemlich genau, ein einziges falsches Zeichen kann dazu führen, dass dein komplettes Plugin nicht mehr funktioniert. In den meisten Fällen wird dir dann eine lange und kryptisch wirkende Fehlermeldung wie in Listing 3.3 angezeigt.

```
[11:29:21] [Server thread/INFO]: [PPLoader] Loading Plugin
hallowelt.py.dir
[11:29:22] [Server thread/WARN]: org.bukkit.plugin.
InvalidPluginException: SyntaxError: ("no viable alternative at
input '\\n'", ('<iostream>', 3, 29, '        self.getLogger().
info("Hallo Welt!)\n'))
[11:29:22] [Server thread/WARN]:   at org.cyberlis.pyloader.
PythonPluginLoader.loadPlugin(PythonPluginLoader.java:244)
[11:29:22] [Server thread/WARN]:   at org.cyberlis.pyloader.
PythonPluginLoader.loadPlugin(PythonPluginLoader.java:118)
```


© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

Index

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260

© des Titels »Let's Play: Programmieren lernen mit Python und Minecraft« (ISBN 9783747502600) 2020 by mitp Verlags GmbH & Co. KG, Frechen. Nähere Informationen unter: http://www.mitp.de/0260