

Informatik 1

Editoren & IDEs

Jonas Miederer

DHBW Stuttgart - Campus Horb

8. Januar 2021



Outline:

- ① Code-Editoren & IDEs
- ② Visual Studio Code
- ③ Bibliographie

Inhaltsverzeichnis

① Code-Editoren & IDEs

② Visual Studio Code

③ Bibliographie

Ausführungsarten von Python-Code (I)

Wie bereits beschrieben kann Python-Code auf unterschiedliche Arten und Weisen ausgeführt werden.

Eine Möglichkeit besteht darin, den Code direkt in den Python-Interpreter einzugeben. Hierdurch wird der Code sofort und unmittelbar ausgeführt, was insbesondere dann hilfreich ist, wenn schnell eine gewisse Funktionalität ausprobiert werden soll.

Der Nachteil besteht jedoch darin, dass der Code, der direkt im Python-Interpreter eingegeben wird, nicht persistiert wird. Das bedeutet, dass der Code nicht abgespeichert wird, sobald der Interpreter geschlossen wird ist der Code verloren. Zudem kann der Code auf diese Weise nicht mit anderen Mitarbeitern geteilt werden, der Code kann schlecht strukturiert werden, etc...

Ausführungsarten von Python-Code (II)

Um diese Nachteile zu umgehen, bietet es sich an, den Code in Python-Dateien zu speichern und diese Dateien dann auszuführen. So wird die Arbeit mit dem Python-Code deutlich einfacher, übersichtlicher und strukturierter.

Der Code innerhalb der Dateien kann ausgeführt werden, indem dem Python-Interpreter der Dateipfad angehängen wird, in der der Code gespeichert ist.

Ist der Code also beispielsweise in der Datei `main.py` gespeichert, kann der Code ausgeführt werden mit `python main.py` (sofern man sich im gleichen Verzeichnis wie die entspr. Datei befindet).

Alternativ kann auch ein absoluter Pfad übergeben werden, z.B.

```
python C:\Users\Jonas\Programming\main.py
```

 (unabhängig vom aktuellen Pfad)

Ausführungsarten von Python-Code (III)

Interpreter- vs. Datei-Modus

Wird lediglich der Befehl `python` ausgeführt, wird der Python Interpreter gestartet, während durch den Befehl `python datei.py` der Code aus der entsprechenden Datei ausgelesen wird und dem Interpreter zur Ausführung übergeben wird. Auf diese Weise gelangt man also nicht in den interaktiven ausführbaren Modus.

Ausführungsarten von Python-Code (IV)

Python Dateiendung

Dateien, in denen Python-Code gespeichert ist, werden i.d.R. mit der Dateiendung `.py` abgespeichert, damit offensichtlich ist, dass es sich dabei um Python-Dateien handelt (analog zu bspw. `.jpg` für Bilder oder `.txt` für Textdateien).

Die Dateiendung `.py` ist nicht zwingend notwendig, jedoch Best Practice bzw. Teil der Konvention und sollte daher immer genutzt werden.

Code-Editoren

Python-Code, der in Dateien gespeichert wird, kann mithilfe jeden beliebigen Text-Editor geschrieben werden, bspw. den standardmäßig vorinstallierten *Windows Editor* auf Windows-Systemen oder auch *TextEdit* auf macOS.

Dennoch sind diese simplen Texteditoren kaum dafür geeignet, Code zu verfassen, da sie keinerlei Hilfestellungen oder anderweitige Features bieten, die dabei helfen, fehlerfreien und sauberen Code zu schreiben.

Daher gibt es spezielle Code-Editoren, die als Text-Editoren mit derartigen Funktionen gesehen werden können.

Features von Code-Editoren

Code-Editoren bieten beispielsweise folgende Features:

- Code- / Syntax-Highlighting (unterschiedliche Farben für unterschiedliche Programm-Bestandteile (Variablen, Funktionsdefinitionen, Schlüsselwörter, ...))
- Autovervollständigung: Schlüsselwörter oder bereits zuvor deklarierte Variablennamen können vervollständigt werden
- Einrückung: Code kann korrekt eingerückt werden, keine Vermischung von Tabulator- und Leerzeichen
- Hervorheben von zusammengehörigen Blöcken: Der Bereich von if-Clauses, Schleifen oder Funktionen wird gekennzeichnet
- ...

Code-Editoren können also dazu beitragen, Fehler frühzeitig zu erkennen und zu beheben sowie die Produktivität des Entwicklers und Qualität des Codes zu steigern.

Bekannte Code-Editoren

Es gibt eine Vielzahl an Code-Editoren, die genutzt werden können. Viele davon sind kostenlos und quelloffen.

- Visual Studio Code
- Sublime Text 3
- Atom text editor
- Notepad++ (nur Windows)
- TextMate (nur macOS)

Integrierte Entwicklungsumgebungen (IDEs)

Neben Code-Editoren existieren auch sogenannte *Integrierte Entwicklungsumgebungen* (engl. *Integrated Development Environments (IDEs)*).

Diese bieten im Vergleich zu "einfachen" Code-Editoren erweiterte Funktionalitäten wie z.B.

- integrierte Versionsverwaltung
- integrierte Compiler / Interpreter / Linker
- Werkzeuge zur Steigerung der Produktivität
- Tools zur Unterstützung bei der Entwicklung graphischer Benutzeroberflächen
- ...

Bekannte IDEs

Im professionellen Umfeld werden zur Code-Entwicklung meist IDEs verwendet. Aufgrund des großen Funktionsumfangs sind IDEs häufig kostenpflichtig.

- PyCharm
- Eclipse
- Spyder
- IDLE (enthalten in der Standard Python Installation)
- Visual Studio Code

Inhaltsverzeichnis

① Code-Editoren & IDEs

② Visual Studio Code

③ Bibliographie

Visual Studio Code

In dieser Veranstaltung werden Übungen und Codebeispiele mithilfe von Visual Studio Code (kurz: VS Code) umgesetzt.

VS Code ist eine Mischung aus Code-Editor und IDE und wird von Microsoft entwickelt. Das Programm ist plattformunabhängig und kann auf Windows, macOS und anderen Systemen ausgeführt werden.

VS Code kann unter <https://code.visualstudio.com/> heruntergeladen werden.

Wahl des Code-Editors

Die Studierenden können ein beliebiges Programm ihrer Wahl zur Programmierung nutzen. Es wird aber empfohlen VS Code zu nutzen, da hierfür die beste Hilfestellung gegeben werden kann.

Visual Studio Code - Einrichtung (I)

Nach der Installation von VS Code und dem Öffnen des Programms sollte Folgendes zu sehen sein:

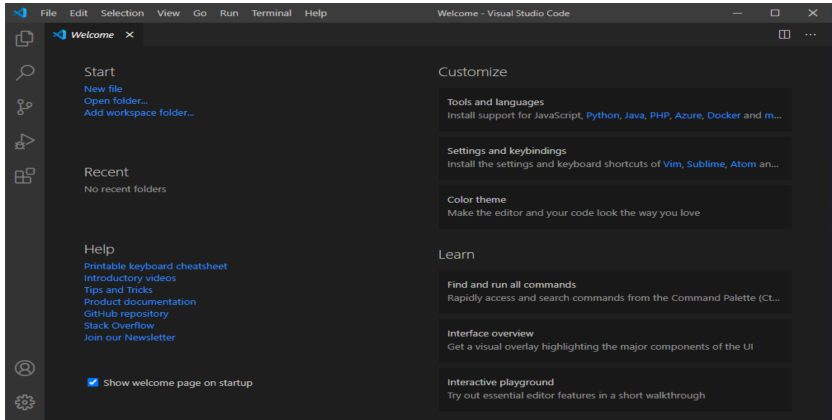


Abbildung 1: VS Code nach dem 1. Start

Visual Studio Code - Einrichtung (II)

Die Bearbeitung von Projekten/Dateien erfolgt in VS Code Workspacebasiert, d.h. es wird ein bestimmter Ordner als Workspace bestimmt, innerhalb dessen die Dateien verwaltet werden.

Ein Workspace kann angelegt werden, indem direkt auf der Startseite

`Add workspace folder...` oder unter `File` der Menüpunkt `Open Workspace...` ausgewählt wird.

Hier bietet es sich an, einen Ordner zu erstellen, in dem alle Übungen und Codebeispiele gespeichert werden. Dieser Ordner dient dann als Workspace.

Visual Studio Code - Einrichtung (III)

Nachdem der Workspace angelegt wurde, steht dieser zur Arbeit bereit. In der linken Leiste von VS Code gibt es verschiedene Reiter, der erste Reiter zeigt die Datei-/Ordnerstruktur des aktuellen Workspaces an.

Durch einen Rechtsklick können so neue Ordner und Dateien angelegt werden

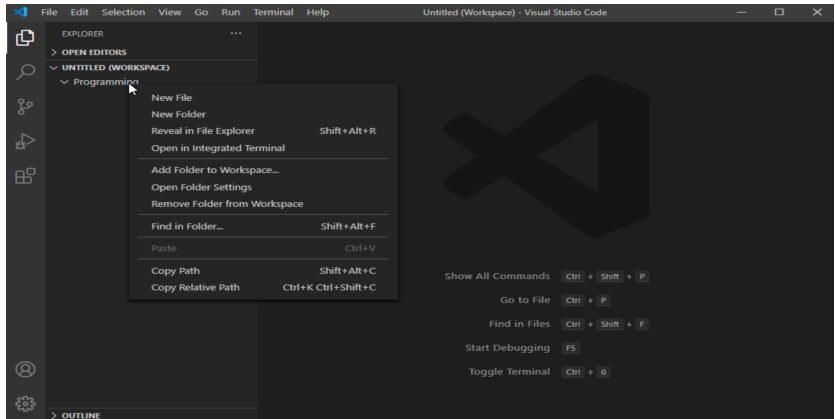


Abbildung 2: Anlegen einer neuen Datei oder Verzeichnisses

Visual Studio Code - Einrichtung (IV)

Wird eine neue `.py`-Datei angelegt, erkennt VS Code, dass es sich dabei um eine Python-Datei handelt und schlägt automatisch die Python-Erweiterung vor. Es wird empfohlen, diese zu installieren, um Python-spezifische Features zu aktivieren.

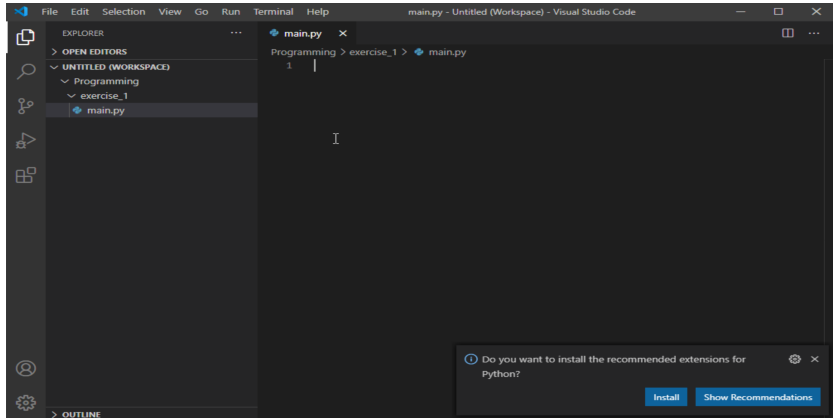


Abbildung 3: Vorschlag zur Installation der Python-Erweiterung

Visual Studio Code - Einrichtung (V)

Sobald die Python-Erweiterung installiert ist, wird automatisch ein installierter Python-Interpreter gesucht und links unten in der unteren Programmleiste angezeigt. Stellen Sie sicher, dass hier der korrekte Interpreter ausgewählt ist

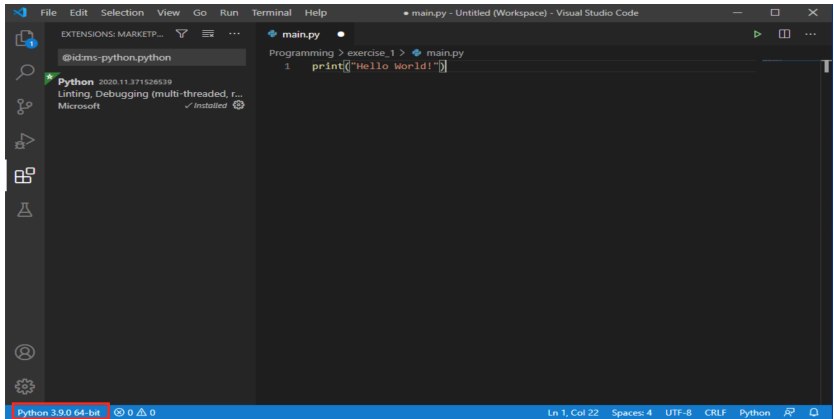


Abbildung 4: Anzeige des aktuell verwendeten Python Interpreters

Visual Studio Code - Einrichtung (VI)

Wird nun eine Python Datei geöffnet, so kann diese direkt aus dem Editor heraus ausgeführt werden, indem das grüne Dreieck im oberen rechten Bereich gedrückt wird. Hierdurch wird die Konsole im unteren Bereich geöffnet, in der die Ausgabe erfolgt.

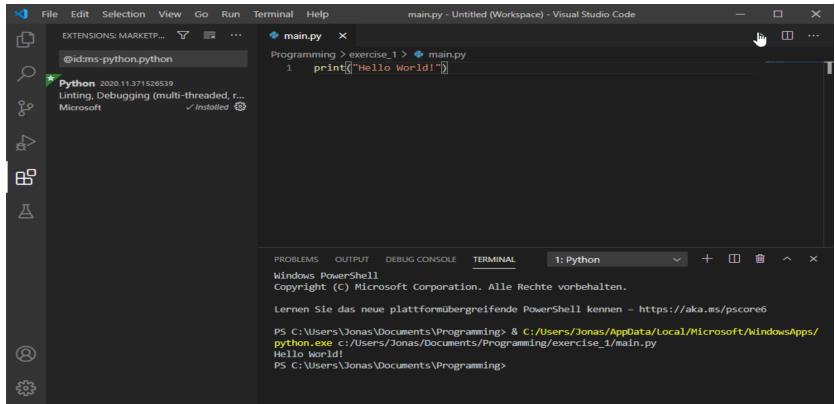
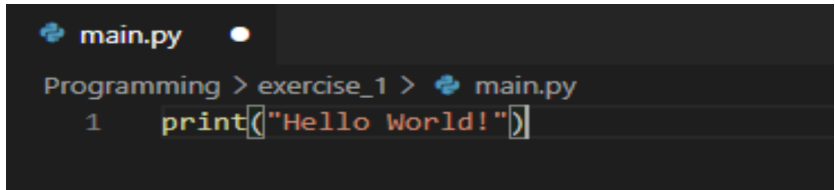


Abbildung 5: Ausführen der Python Datei

Visual Studio Code - Einrichtung (VII)

Achtung: Wird hinter dem Dateinamen ein weißer Punkt angezeigt, bedeutet das, dass die Änderungen in der Datei noch nicht gespeichert wurden.



```
main.py  
Programming > exercise_1 > main.py  
1 print("Hello World!")
```

Abbildung 6: Ungespeicherte Änderungen in der Datei main.py

Inhaltsverzeichnis

① Code-Editoren & IDEs

② Visual Studio Code

③ Bibliographie

Bibliographie I