

Python

Python ist eine **Programmiersprache**. Es liest einen Text ein, den du geschrieben hast (auch **Code** genannt), wandelt ihn in Befehle um, die für den Computer verständlich sind und führt diese dann aus. Du wirst lernen selbst Code zu schreiben, der coole und nützliche Dinge tut. Du wirst in der Lage sein, Programme selber zu schreiben, um mit Deinem Computer Dinge zu tun und nicht mehr nur Software von anderen Programmierern verwenden müssen.

Zuerst wirst du lernen Python zu benutzen und mit ihm zu kommunizieren. Es gibt mehrere Wege das zu tun; der erste ist, mit dem Python Interpreter über die Textkonsole deines **Betriebssystems** in Kontakt zu treten.

Eine **Konsole** (auch 'Terminal' oder 'Prompt') ist ein *schriftlicher* Weg deinem Betriebssystem Befehle zu erteilen, so wie der 'Desktop' zusammen mit deiner Maus ein *grafischer* Weg zu deinem System ist.

Wie öffne ich eine Konsole?

Die Windows-Konsole wird *Eingabeaufforderung* genannt, oder *cmd*. Die Tastenkombination Windows+R (Windows meint die Taste mit dem Windows-Logo) öffnet dieses als neues Fenster mit schwarzem Hintergrund. Gib dann *cmd* ein und klick Enter. Die erste Zeile sollte in etwa so aussehen:

```
C:\Users\myusername>
```

Du kannst den Python Interpreter (siehe unten) auch direkt ausführen, oder das IDLE-Programm das Python mitliefert. Diese findest Du im Startmenü.

Python verwenden

Python verhält sich standardmäßig als etwas das sich *Interpreter* nennt. Ein *Interpreter* nimmt Befehle entgegen und führt sie dann aus – super praktisch um Dinge auszuprobieren!

Tippe in deine Konsole *python* ein und drücke Enter um den Python-Interpreter zu starten. Sobald Python startet, zeigt es Dir einige Kontextinformationen ähnlich zu diesen:

```
Python 3.3.2 (default, January 21 2014, 15:40:45)
[GCC 4.8.0 20130502 (prerelease)] on linux
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```

Tipp: Der Prompt >>> in der letzten Zeile signalisiert, dass du dich nun in der interaktiven *Python Konsole* befindest, auch *Python Shell* genannt. **Diese Konsole ist etwas anderes als das normale Kommandozeilenterminal!**

Theoretisch ist es also möglich in dieser *Python Shell* zu programmieren, jedoch dient es in den meisten Fällen nur für sehr kurze Anweisungen oder Tests. Für die Programmierung von richtigen Programmen verwendet man auch hier eine Entwicklungsumgebung. Als Entwicklungsumgebung reicht für Python theoretisch schon der ganz normale *Editor* unter Windows (nicht zu verwechseln mit Wordpad oder Word). Damit die Handhabung für dich aber

leichter ist, verwenden wir in dieser Einheit die Entwicklungsumgebung Geany, mit welcher sich unter anderem auch Python Programme schreiben, testen und ausführen lassen.

Python Installation

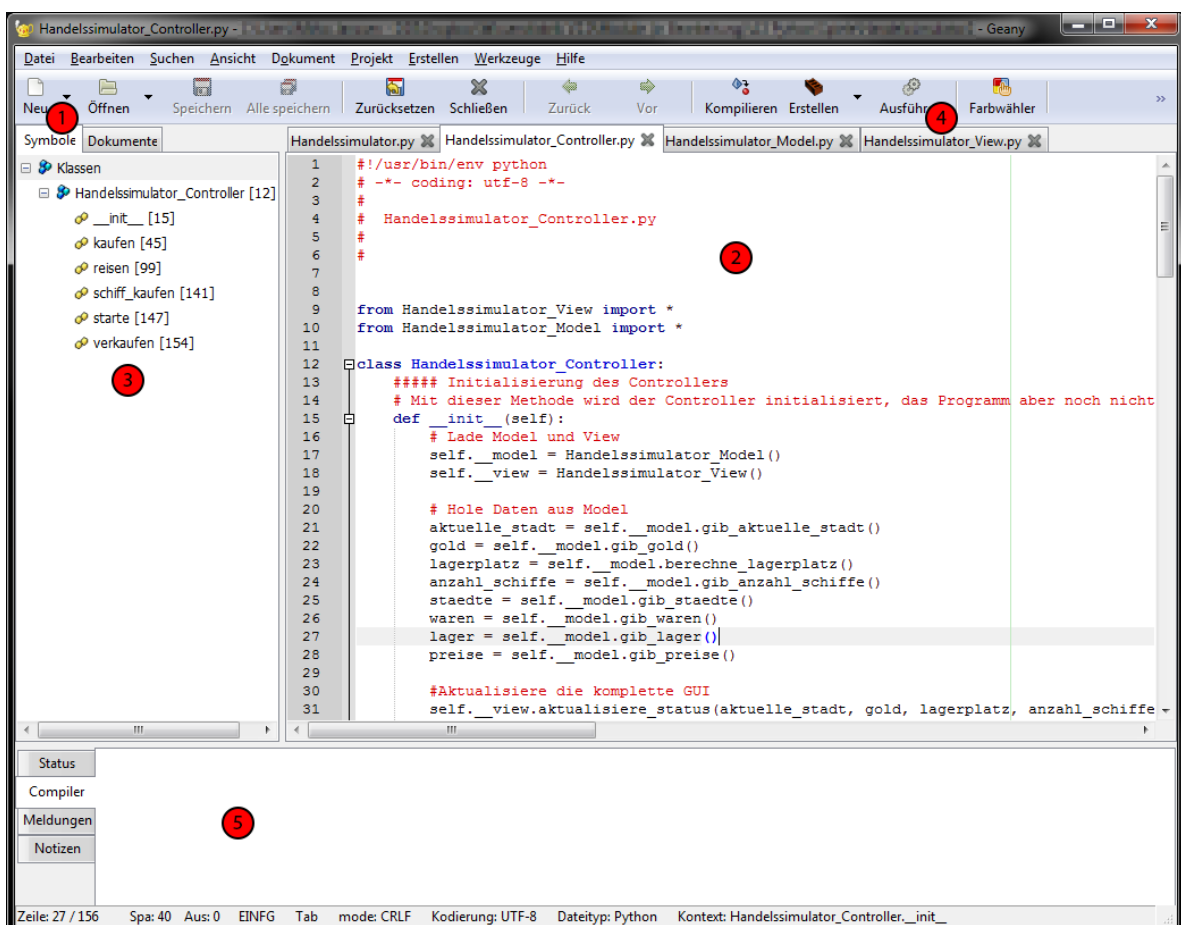
Wenn Python noch nicht auf deinem Computer installiert ist, findest du die aktuellen offiziellen Installationspakete hier:

<http://python.org/download/>

Wir verwenden Python 2.7 in dieser Unterrichtseinheit, prinzipiell sind beide Versionen nutzbar. Python 3 ist die eigentliche neueste Version, Python 2 wird jedoch parallel noch weiter mit entwickelt. Unter <https://wiki.python.org/moin/Python2orPython3> findest du die Hauptunterschiede und Gemeinsamkeiten beider Versionen.

Hinweis: Unter Windows ist es nötig, dass du Python zu deinem Systempfad hinzufügst, sodass anderen Programmen in der Lage sind Python zu finden. Eigentlich sollte das in dem Setup-Programm bereits ausgewählt und damit automatisch gemacht werden. Für den Fall, dass du beim späteren Programmieren merkst, dass dies nicht der Fall ist, kannst du in das Installationsverzeichnis (C:\Python33\) navigieren, den Tools-Ordner und anschließend den Scripts-Ordner öffnen, durch ausführen der Datei win_add2path.py wird die Umgebungsvariable dann gesetzt.

Geany



1. Hier kann eine neue Datei angelegt werden. Wichtig zu erwähnen ist hier, dass verschiedene Formate (.py für Python) als Vorlagen ausgewählt werden können.
2. In diesem Bereich befindet sich der Quellcode des Programms.
3. Hier werden die Klassen und deren Methoden aus der aktuellen Datei aufgelistet.
4. Mit diesem Button kann das Programm ausgeführt werden.
5. In diesem Bereich werden alle Benachrichtungen in Bezug auf das Kompilieren bzw. Ausführen des Programmes angezeigt. Fehlermeldungen werden grundsätzlich im Terminal des ausgeführten Programmes angezeigt.