

# Hello EiP

## Übung 1

### Einführung in die Programmierung

#### Über dieses Übungsblatt



Auf diesem Übungsblatt geht es lediglich darum, die in diesem Semester notwendigen Tools zu installieren und sich mit diesen etwas auseinanderzusetzen. Es erfolgt keine Abgabe dieses Übungsblattes. In der Vorlesung / Tutorium der ersten Woche werden Sie durch den Installationsprozess begleitet. Benötigt werden:

- Eine funktionsfähige, neuere Python-Installation. Da wir sehr wenige fortgeschrittene Python-spezifische Features verwenden werden, sollte die Version nicht besonders wichtig sein so, lange sie mit Python3 arbeiten. Nichtsdestotrotz können die Korrektur nur für auf Python 3.10 funktionsfähige Programme garantieren.
- Eine Entwicklungsumgebung (IDE). Wir empfehlen an dieser Stelle [VS Code \(Microsoft, kostenlos\)](#) (vielleicht etwas simpler für Anfänger) oder [Pycharm \(Jetbrains, kostenlos für Studis\)](#). Ein einfacher Texteditor wie Notepad++ sollte allerdings auch ausreichen. Wir empfehlen Ihnen nach der Installation sich kurz in ihrer IDE zurechtzufinden: zu beiden vorgeschlagenen IDEs gibt es eine offizielle Dokumentation und zahllose Tutorials im Internet.
- (Optional) eine [Git](#) Installation um mit ihren Kommilitonen kollaborativ zu arbeiten. Git ist eine sog. Versionierungssoftware und wird *quasi überall* verwendet, sowohl im späterem Studium als auch bei jedem Arbeitgeber der Software produziert. Wir empfehlen die [folgende Webseite](#) um spielerisch zu erlernen, wie Git funktioniert. Das Land Rheinland-Pfalz stellt Studierenden einen kostenlosen [Git-Webserver](#) zur Verfügung, den Sie zum Speichern/Austausch von selbst erstellter Software nutzen können.

# Aufgabe Hello World

*Letzte Änderung: 13. April 2022, 13:39 Uhr*

0 Punkte — [im Detail](#)

Ansicht:  |  Ein fundamentaler Unterschied zwischen interpretierten Programmiersprachen wie Python und Kompilierten Programmiersprachen wie z.B. C++ ist, dass Python einen sogenannten Interpreter verwendet, der Programmcode *zur Laufzeit* übersetzt. Dies bietet einige Vorteile, wie z.B. Programmcode zur Laufzeit einfügen zu können, aber auch Nachteile wie eventuell längere Laufzeiten. Wir möchten uns deshalb mit dem [Python-Interpreter](#) vertraut machen.



**1)** Öffnen Sie den Python-Interpreter in der Kommandozeile und geben Sie den Befehl `print("Hello world")` ein. Sie sollten nun die entsprechende Ausgabe angezeigt bekommen. Glückwunsch, Sie haben ihr erstes Programm in Python ausgeführt!

**2)** Erstellen Sie eine neue Datei "hello\_world.py" und öffnen Sie sie in einem beliebigen Texteditor. Fügen Sie ebenfalls den Befehl `print("Hello world")` ein und speichern Sie die Datei. Führen Sie die Datei nun mit dem Befehl `python hello_world.py` aus der Kommandozeile aus.

# Aufgabe Debugger

Letzte Änderung: 15. June 2023, 12:38 Uhr

0 Punkte — [im Detail](#)

Ansicht:  |  Wie in Aufgabe 1 erwähnt, kann man in Python **zur Laufzeit** in den Programmcode eingreifen. In dieser Aufgabe möchten wir den grundlegenden Umgang mit einem Debugger in Python üben. Wir stellen im Tutorium zwei Möglichkeiten vor dies zu tun: entweder mit dem Python-internen Debugger [pdb](#) bzw. die etwas verbesserte Version [ipdb](#) oder mit dem in Ihrem IDE integrierten Debugger: [Pycharm](#) bzw. [VS-Code](#).

- 1) Öffnen Sie das beigefügte Programm fibonacci.py aus der Kommandozeile und überprüfen Sie dessen Funktionalität für  $n = 50$ .
- 2) Plazieren Sie einen Haltepunkt (Breakpoint) in Zeile 10 des Programms. Inserieren Sie dazu entweder eine neue Zeile `breakpoint()` oder klicken Sie in ihrem IDE rechts neben der Zeilennummer um einen Breakpoint zu platzieren. Führen Sie das Programm erneut mit dem gleichen Input aus (achten sie in ihrem IDE darauf das Programm im Debug-Modus zu starten) und warten Sie bis das Programm an der angegebenen Stelle hält.
- 3) Machen Sie sich mit den Befehlen `(n)` - `next`, `(b)` - `break` und `(c)` - `continue` vertraut. Nachzulesen sind diese in der Dokumentation von [pdb](#).
- 4) Lassen Sie sich den Wert der Variablen `f0` und `f1` ausgeben.
- 5) Verändern Sie die Variable `p_old`, indem Sie ihr einen neuen Wert zuweisen, z.B. mit `p_old = 15`. Deaktivieren Sie nun den Breakpoint und lassen Sie das Programm zu Ende laufen. Hat sich die Ausgabe verändert?

# Aufgabe Basic-IO Installation

Letzte Änderung: 15. June 2023, 12:38 Uhr

0 Punkte — [im Detail](#)

Ansicht:  | 



So sollte der erfolgreiche Testbildschirm von Basic-IO aussehen.

Basic-IO ist unser hausgemachtes, simples Grafik-Framework basierend auf [QT](#). Mit Basic-IO können Sie mit *einfachen* Befehlen simple Formen (Quadrat, Rechteck) zeichnen, aber auch Tastatureingaben abfragen. Wir empfehlen Basic-IO für Grafik-Aufgaben der kommenden Übungsblätter zu nutzen. Sie können gerne komplexere Alternativen wie [PyGame](#) nutzen, wir können Ihnen allerdings keine Hilfe dafür anbieten.

Python nutzt den Package-Manager [pip](#) zum Installieren von Paketen. Pip ist sehr einfach zu handhaben und ermöglicht es Python-Programmierer\*Innen schnell und einfach von anderen Programmierer\*Innen erstellte Softwarepakete zu installieren. Ganz nach dem Motto [Zwerge auf den Schultern von Riesen](#) verwenden Programmierer\*Innen seit der Erfindung des modernen Computers Code von anderen Programmierer\*Innen um immer größere und komplexere Software zu schreiben. Wir möchten Sie in dieser Aufgabe in dem Prozess anleiten ein Softwarepaket über pip zu installieren.

**1)** Um Basic-IO zu installieren, sollte der Befehl `pip install jguvc_eip` ausreichen. Danach sollte es möglich sein Basic-IO in der Python-Konsole mit `from jguvc_eip import basic_io` zu importieren. Wenn das nicht funktionieren sollte, überprüfen Sie ob das `pip` und `python` aus ihrer Konsole zur gleichen Version gehören mit den Befehlen `python --version` und `pip --version`. Als "Quickfix" können Sie *immer* auf die zu `python` passende `pip`-Version mit `python -m pip` zugreifen.

**2):** Führen sie die auf LMS bereitgestellte Datei `test_basic_io.py` aus und überprüfen sie somit ob Basic-IO korrekt funktioniert.

**3)** Wenn Sie noch Zeit haben und sich etwas Arbeit in der Zukunft vorwegnehmen wollen, fangen Sie an das bereitgestellte [PDF-Handbuch](#) auf Deutsch oder die automatisch generierte [Online-Dokumentation](#) auf Englisch zu lesen.