

Einführung: Git, Entwicklungsumgebung, Python-Tutorial

Git-Repository

In dieser Übung soll das für alle weiteren Übungen genutzte Gitlab-Repository aufgesetzt werden. Diese Übung ist nicht abgabepflichtig. Loggen Sie sich hierzu unter <https://git.tu-berlin.de/> mit Ihrer LoginID ein und klicken Sie auf **New project** --> **Create blank project** ein neues Repository für Ihre Gruppe zu erstellen. Vergeben Sie den Namen des Repositories, und setzen Sie den Wert für **Visibility Level** auf **Private**. Nachdem das Repository erstellt wurde, können Sie unter

Manage --> **Members** --> **invite members**

den weiteren Mitgliedern Ihrer Gruppe Zugriff auf das Repository gewähren (unter **Choose a role permission** sollte hierbei **Developer** ausgewählt werden). Klicken Sie anschließend auf der Hauptseite des Projekts auf  --> **New file**. Klicken Sie danach auf "Commit changes" um die Datei Ihrem Repository hinzuzufügen. Sie können beispielsweise das Remote-Repository mit den entsprechenden Befehlen aktualisieren ([um Git zu installieren](#)):

```
~/path-to-my-repository$ git add --all
~/path-to-my-repository$ git commit -m "Eine Datei hinzugefügt."
~/path-to-my-repository$ git push
```

Mit dem Befehl `git pull` können nun alle weiteren Mitglieder Ihrer Gruppe Ihre lokalen Repositories entsprechend aktualisieren.

Reihenfolge: `git pull` → `git add` → `git commit` → `git push`.

Anaconda (Optional)

Für die Python-Programmierung empfehlen wir die Installation von Anaconda. Durch ihre einfache und MATLAB-ähnliche Syntax ist Python besonders leicht zu erlernen. Der Funktionsumfang der Sprache wird in Paketen (engl. Packages) organisiert. Die Python-Distribution Anaconda enthält für die Betriebssysteme Windows, macOS und Linux bereits viele der benötigten Pakete (z.B. Matplotlib oder NumPy). Bitte installieren Sie die aktuelle Anaconda-Version für ihr Betriebssystem. Der [Blog](#) und das [Video](#) zeigen

die Installation der Anaconda-Distribution von Python für die Betriebssysteme Windows, macOS und Linux und den Aufruf der integrierten Entwicklungsumgebung PyCharm.

Virtuelle Umgebung für Python 3.6 (Optional)

Für das Projekt benötigen Sie die Python-Version 3.6. Nach der Installation von Anaconda können Sie schnell eine entsprechende virtuelle Umgebung mit dem Namen "ASE" erstellen:

```
conda create -n ASE python=3.6 anaconda
```

Conda ist der Paketmanager von Anaconda, der aber auch andere Aufgaben wie das Anlegen von virtuellen Umgebungen übernimmt. Nun können Sie über die Konsole die virtuelle Umgebung wie folgt aktivieren und deaktivieren:

```
conda activate ASE
conda deactivate
```

PyCharm

Als Entwicklungsumgebung bietet sich PyCharm-Community an. Die ersten Schritte beinhalten das Erstellen eines neuen Python-Projekts:

- Klicken Sie auf Menü **File --> New Project...**
- Bitte ändern Sie den Standardnamen 'pythonProject' in 'ASE' im Feld **Location**
- Wählen Sie den passenden Interpreter unter **Base interpreter** aus (Python 3.6).
- Deaktivieren Sie die Option **Create a main.py welcome script**.

Jetzt können Sie Ihr erstes Python-Skript erstellen:

- Gehen Sie zu Menü **File --> New...**, klicke auf **Python File**.
- Geben Sie dem Skript einen Namen, zum Beispiel 'Hello_World'

Run und Debug

Schreiben Sie in das Skript:

```
a = 1
b = 2
c = a + b
print('Hello world')
```

Figure 1 zeigt die Entwicklungsumgebung. Wenn Sie auf 'Run' klicken, wird das Programm bis zum Ende ausgeführt und die Zeichenfolge "Hello world" wird ausgegeben. Wenn Sie neben eine Zeile klicken, wird ein roter Punkt angezeigt. Dieser rote Punkt ist ein Haltepunkt. Wenn Sie nun auf 'Debug' klicken, wird das Programm bis zu diesem Haltepunkt ausgeführt, und im **Debugger**-Fenster werden die Werte der vorherigen Variablen angezeigt. Alternativ können Sie auch das `breakpoint()`-Kommando verwenden, um einen Haltepunkt zu setzen.

Python Tutorial

Im ISIS-Kurs befindet sich ein Python-Tutorial. Wir empfehlen Ihnen, die Kapitel 1-6 in dieser und der kommenden Woche durchzuarbeiten und in Ihr Git hochzuladen.

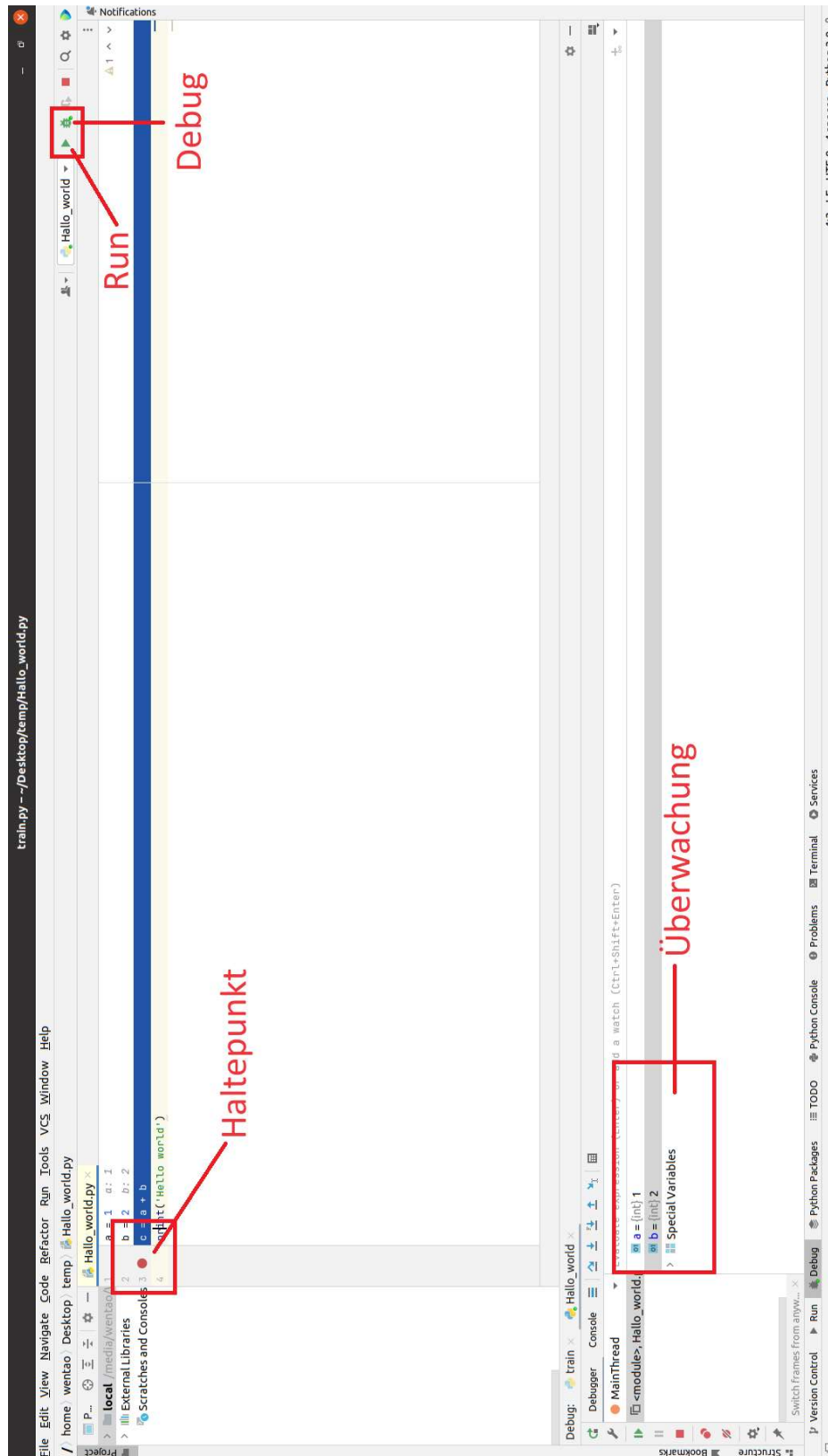


Figure 1: PyCharm Fenster