

# 1 Allgemeines

Für das Informatik3-Praktikum werden die folgenden Programmiersprachen und Tools benötigt.

Installiere von allen Sprachen jeweils die neuste Version. Es wird außerdem empfohlen, dass alle Gruppenmitglieder dieselben IDEs (= Integrated Development Environment = Entwicklungsumgebung) verwenden. Erfahrungsgemäß ist es für Gruppen, in denen gemischte Betriebssysteme verwendet werden, sinnvoll, Virtual Machines zu verwenden, um Kompatibilitätsprobleme zu vermeiden.

## 2 Setup

Installiere grundsätzlich zuerst die Compiler/ Interpreter der Programmiersprache und erst anschließend die IDE (sofern diese nicht in der IDE integriert oder bereits auf deinem System vorhanden sind). Es handelt sich bei den angebotenen IDEs lediglich um Vorschläge. Generell darf gerne jede andere IDE verwendet werden, solange es innerhalb der Gruppe nicht zu Inkompatibilitäten kommt.

### 2.1 C# und C++

#### 2.1.1 Installation

Installiere die Community-Edition von Visual Studio: <https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx>. Dieses beinhaltet auch die Möglichkeit, in Python in VS zu programmieren. Wähle bei der Installation unter "Workloads" die Elemente ".NET-Desktopentwicklung" und "Desktopentwicklung mit C++" aus.

#### 2.1.2 Hello C\*-World

Lege nun je ein C#- und ein C++-Projekt (New → New Project → Visual C# → Windows → Consoleapplication, bzw. New → New Project → Visual C++ → Win32 → Win32-Consoleapplication) beliebigen Namens an. Visual Studio wird in beiden Projekten Standard-Dateien erstellen. Kompiliere und führe diese aus, um sicherzustellen, dass die Einrichtung erfolgreich war. Anschließend können beide Projekte wieder gelöscht werden.

### 2.2 Python

#### 2.2.1 Installation und Hello Python-World

Folge dem Tutorial von Lars Vogel zum Thema Python und PyDev ab Sektion 2 bis mindestens Sektion 3, um Python und IDE auf deinem Rechner aufzusetzen. Die übrigen Sektionen werden ebenfalls empfohlen, um einen kurzen Überblick

über Python und PyDev zu erhalten.

**Achtung:** der Beispielcode verwendet noch die alte Syntax "`print x`". Seit Python 3.0 ist `print` kein Schlüsselwort mehr, sondern eine Funktion. Die neue Benutzung ist daher "`print(x)`". Beachte außerdem, dass Python Blöcke nicht wie aus Java bekannt mit geschweiften Klammern zuordnet, sondern mittels Einrückung. Achte daher hier besonders auf eine saubere Einrückung. Außerdem solltest du die aktuellste Version von Python (Stand bei Erstellung des Tutorials: 3.6.0) installieren.

## 2.3 Git und Github

Installiere Git für die Benutzung mit der Kommandozeile und lese dich mittels der Dokumentation in die Benutzung der Kommandozeile ein. Einige IDEs bieten möglicherweise nativ oder über ein Plugin Unterstützung für Git an. Da bei unserem Projekt ein einziges Repository für alle verwendeten Sprachen benutzt wird, wird hier empfohlen, ausschließlich mit der Kommandozeile zu arbeiten, um etwaige Konflikte zwischen den einzelnen IDEs zu vermeiden.

Es dürfen anstatt eines Github-Repositories auch GitLab, bitbucket oder andere Anbieter verwendet werden, solange diese ein privates Repository ermöglichen. Hier bietet sich der GitLab Server der Hochschule an, da alle bereits ein Konto dort haben sollten (automatisch).

Lege dir sich ein Benutzerkonto auf Github an (**jedes** Gruppenmitglied). **Ein** Gruppenmitglied legt dort dann ein **private**<sup>1</sup> Repository an und fügt alle anderen Gruppenmitglieder als Mitarbeiter hinzu. Wähle dazu das Repository über den Browser an und gib im Menü "Settings" die Namen deiner Mitglieder unter "Collaborators" ein, um diesen Schreibrechte für das Repository zu gewähren.<sup>2</sup>

**Ein** Gruppenmitglied kann nun lokal auf dem eigenen Rechner das erstellte Repository klonen, die in RELAX mitgelieferte Ordnerstruktur einfügen und den ersten Commit erstellen:

```
git clone https://github.com/<Pfad zum Repository>.git
< Dann die Ordnerstruktur einfügen >
< Danach >
git add -A
git commit -m "initialized directory structure"
git push
```

---

<sup>1</sup>Studenten können sich für das Student-Programm bei GitHub anmelden, mittlerweile hat aber jeder die Möglichkeit, kostenlose private Repositories zu erstellen.

<sup>2</sup>Auf GitLab muss der Tutor mindestens "Reporter" Rechte haben

Alle anderen Gruppenmitglieder können nun ebenfalls das Repository klonen:

```
git clone https://github.com/<Pfad zum Repository>.git
```

Möglicherweise fragt Git bei der ersten Benutzung nach der Angabe von Benutzer und Email-Adresse. Die entsprechenden Befehle werden direkt angezeigt. Sollte Git beim Pushen von Änderungen nicht mehr reagieren, könnte der folgende Befehl helfen:

```
git config --global core.askpass "git-gui--askpass"
```

**Ein** Gruppenmitglied stellt nun den Link eures Repositories in die entsprechende Abgabe auf RELAX.