

# **Tutorium 01 - 2024-10-17**

**Vorstellen, Orga, Zusammenfassung, Installation von WSL/VS Code**

# Über mich

- Nils Pukropp
- Informatik B.Sc. 5. Semester
- EidP Tutor seit zwei Semestern
- Gerne einfach "Du"

```
print("Hello", "Tutorium", "2024!")  
>>> "Hello Tutorium 2024!"
```

# Wie wird das Tutorium ablaufen?

- Vorstellen des letzten Blatt
- Wichtiges aus Vorlesung/für nächstes Blatt
- Fragen zum nächsten Blatt
- Allgemeine Fragen zur Vorlesung
- Am Ende QR-Code für Anwesenheit
- Ihr könnt mir gerne am Ende privat noch Fragen über die Vorlesung oder das Studium stellen

# Kontakt

- Mail: [nils@narl.io](mailto:nils@narl.io)
- Tutorium-Files: [git.narl.io/nvrl/eidp-2024](https://git.narl.io/nvrl/eidp-2024)
- Telegram: [@narl\\_np](https://t.me/narl_np)

# Orga

**Was müsst ihr machen/wissen**

# Studienleistung bekommen

- Im HisInOne sich für dieses (oder ein anderes) Tutorium anmelden + Studienleistung (wird noch freigeschaltet)
- 14 Blätter, jeden Dienstag auf der EidP Website
- 2x Vorrechnen
  - Muss nicht korrekt sein
  - Ich werde gute Abgaben fragen
- 3 Punkte für Anwesenheit
  - Ab nächstem Tutorium
  - Über QR-Code
- Ihr braucht insgesamt mehr als 50% der Punkte (>50%)

# Prüfung

- Nach den Vorlesungen (Ende Februar/März)
- Setzt die Studienleistung voraus
- Schwerere/Schwerste Klausur im ersten Semester

# Warum fällt vielen EidP so schwer?

- Zeitdruck in der Klausur
- In der Studienleistung schummeln ist einfach, in der Klausur nicht
  - fehlende Übung im Programmieren
  - Nicht ausreichend mit den Themen befasst
  - Wahrscheinlich die Übungsblätter nicht vollständig selbst bearbeitet
  - Nachdem man die Studienleistung hat (>50% Punkte) aufgehört sich mit der Vorlesung zu beschäftigen



# ChatGPT

- Wir tolerieren den Gebrauch
- **ABER:** Ihr müsst in der Klausur alles verstanden haben und genug Übung im Umgang mit Python haben
- Ob ihr jetzt Stunden braucht um die Lösung von ChatGPT nachzuvollziehen oder euch in Stunden die Lösung selbst erarbeitet und dabei praktische Programmiererfahrung sammelt ist eure Entscheidung.
- Aus eigenem Interesse würde ich generierte Stellen mit einem Kommentar versehen, damit ich einen Überblick habe welche Aufgaben schwerer fallen und wie viel ChatGPT verwendet wird

# Plagiat

- Wir tolerieren **kein** Plagiat
  - Nicht von anderen Abschreiben
  - Keine Lösungen aus dem Internet
- Erster Plagiat: 0 Punkte
- Zweiter Plagiat: Viel Erfolg im nächsten Jahr

# Abschließend

- Stellt Fragen im Tutorium
  - Im Tutorium für alle
  - Nach dem Tutorium für eine genauere Erklärung
- Nutzt die Studienleistung zum Üben
- Vor der Klausur werde ich wieder ein Zusatz-Tutorium veranstalten um euch Abschließend auf die Klausur vorzubereiten
- Gerne auch Fragen übers Studieren/Studium

# Zusammenfassung der Vorlesung

**Was müsst ihr Wissen?**

# Was muss ich installieren?

- Setup Guide
- Zusammengefasst:
  - WSL/Linux
  - Python 3.12.x
  - VS Code
  - VS Code-Extensions:
    - Python, Pylance, Flake8, autopep8
  - Git

# Warum Linux statt WSL?

- Ihr werdet euch im Studium sowieso mit Linux auseinandersetzen
- Endlich kein Windows mehr
- Wir haben nicht einen Tutor der Windows benutzt
- Deutlich kürzere Installation auch in zukünftigen Vorlesungen
- Ihr lernt den Umgang mit der Shell deutlich schneller

# Was ist die Shell?

- Die Shell ermöglicht wie die UI mit dem Betriebssystem zu interagieren
- Ermöglicht einen präziseren Umgang mit eurem Betriebssystem
- Ungewohnt am Anfang, aber mit etwas Übung deutlich effektiver/schneller als auf UIs angewiesen zu sein
- Viele Prozesse sind nur mit der Shell möglich (haben kein UI)

# Was ist Git?

- Git dient der Versionskontrolle
- Ihr könnt neue/veränderte Dateien zu Git hinzufügen um so eine Version einer Datei abzuspeichern
- Hat noch viele weitere Funktionen für die Software Entwicklung
- Für die Vorlesung braucht ihr lediglich `add`, `commit`, `pull`, `push`,  
( `status` )



## git status

- Zeigt neue/veränderte/staged Dateien an

## git add

- Fügt neue/veränderte Dateien in den `staged` Bereich von Git
- Diese Dateien sind noch nicht vollständig gespeichert
- Mit `git status` könnt ihr sehen welche bereits `staged` sind

```
git commit -m "commit message"
```

- "Speichert" die Dateien, welche mit `git add` hinzugefügt wurden

# Git-Server

- Die Dateien sind jetzt aber nur lokal gespeichert
- Git-Server ermöglichen den Austausch von Quellcode im Internet
- So können auch mehrere an einem Quellcode arbeiten

## `git pull`

- Der Git-Server wird nach neuen Änderungen gefragt
- Die neuen Änderungen werden heruntergeladen

## `git push`

- Lokal gespeicherte Änderungen werden auf den Git-Server hochgeladen
- Davor muss `git commit` verwendet worden sein

# Python

- Wird in `.py` Dateien geschrieben
- Wird mit dem Befehl `python filename.py` ausgeführt
- Python übersetzt die Datei, und führt diese Zeile für Zeile aus

# Python Shell

- Interaktive Shell (ähnlich wie die Linux Shell)
- wartet auf Benutzereingaben
- interpretiert Benutzereingaben nacheinander
- Wird mit dem Befehl `python` ausgeführt

# Zahlentypen

## int

- Ganzzahlen  $\mathbb{Z}$
- In Python nur durch verfügbaren Speicher limitiert

## float

- Rationale Zahlen  $\mathbb{R}$
- neben Speicherlimitierung auch noch limitiert in der Genauigkeit (Floatpoint Precision)

# Rechenoperation

- Wie mans kennt eigentlich
  - Addition `+`
  - Subtraktion `-`
  - Multiplikation `*`
  - Division `/`
  - Ganzzahlige Division `//`
  - Potenzen `**`
  - Modulo `%`

# Übungsblatt 1 + Installationsprobleme