

# 12-01: Rückblick

## Herzlichen Glückwunsch!

Ihr habt den ersten Programmierkurs des Computerlinguistikstudiums erfolgreich abgeschlossen! Damit verfügt ihr jetzt über eine gute Grundlage für die weiterführenden Kurse in den nächsten Semestern.

## Organisatorisches

Falls ihr im Laufe des Semesters genügend Aufgaben erfolgreich gelöst habt (15 oder mehr), habt ihr euch den BN für diesen Kurs verdient. In Moodle könnt ihr selbst sehen, ob eure Punkte ausreichen. Die **digitalen BNs** werden wir in den nächsten Wochen (bis spätestens 31. März) ausstellen. Wenn ihr aus irgendwelchen Gründen einen **Papier-BN** benötigt, informiert am besten die Tutor\_innen. Die Papier-BNs werden dann bis Ende März im Sekretariat der allgemeinen Sprachwissenschaft hinterlegt.

Heute gibt es einen Rückblick auf das vergangene Semester und einen Ausblick auf Dinge, die wir im Kurs nicht behandeln konnten, die euch aber vermutlich in eurer Zukunft als Profi-Pythonistas begegnen werden.

## Rückblick

Was haben wir in diesem Kurs alles gelernt, und wofür waren die einzelnen Konzepte wichtig?

## Variablen

In Variablen können wir **Werte speichern**, um darauf später im Programm zugreifen zu können und die Werte abzufragen oder zu ändern. In den Übungsaufgaben wurde meistens ein Wert für eine vorgegebene Variable angegeben, mit dem wir gearbeitet haben, um davon einen passenden Ausgabewert abzuleiten.

## Datentypen

Verschiedene Arten von Informationen sollten unterschiedlich repräsentiert werden. Wir haben die folgenden eingebauten Python-Datentypen kennengelernt:

- Strings
- Listen
- Zahlen (Integer, Float)
- Booleans (Wahrheitswerte)
- Dictionarys
- Mengen (Sets)

- Tupel
- None

Jeder Datentyp hat eigene **Methoden**, die wir auf Objekte des jeweiligen Typs anwenden können. Manche Datentypen sind **veränderbar (mutable)** und andere nicht. Mit veränderbaren Datentypen müssen wir etwas anders umgehen als mit unveränderbaren Datentypen. Manche Datentypen haben eine **feste Reihenfolge** (Strings, Listen, Tupel), andere nicht (Dictionaries, Mengen).

## Umgang mit sequenziellen Datentypen

Wenn wir sequenzielle Datentypen wie Strings oder Listen verarbeiten, wollen wir manchmal auf jedes einzelne Element darin der Reihe nach zugreifen. Dafür können wir **Indexing** und **Slicing** benutzen.

## Strukturierung unserer Programme

Mit **for-Schleifen** können wir Teile unseres Programms für jedes Element einer Sequenz wiederholen, bis die Sequenz fertig durchgelaufen ist.

Wenn Teile unseres Programms nur unter bestimmten Bedingungen ausgeführt werden sollen, verwenden wir **if-Blöcke**, oft in Kombination mit **elif** und **else**.

## Funktionen

Für wiederkehrende Aufgaben oder Sinneinheiten in unserem Programm definieren wir **Funktionen** mit bestimmten **Eingabeargumenten** und **Rückgabewerten**, um Teilaufgaben zu kapseln und das Programm übersichtlicher zu machen.

Funktionen können sich selbst aufrufen. Das ist dann nützlich, wenn wir die Aufgabe, für die die Funktion verantwortlich ist, in immer einfachere Teilprobleme zerlegen können, bis wir bei einem trivialen Problem ankommen und dann nach und nach die Gesamtlösung zusammenfügen können. Funktionen, die so aufgebaut sind, werden **rekursiv** genannt. Da natürliche Sprache auch rekursive Eigenschaften hat, ist dieses Konzept besonders für uns in der Computerlinguistik nützlich.

Wenn Funktionen zusammen eine größere Sinneinheit bilden, können wir sie in eigene **Module** auslagern und diese Module dann in anderen Programmen importieren. Es gibt auch eingebaute Module in Python, wie zum Beispiel `random` oder `re`.

## Umgang mit Wissenslücken oder Programmierproblemen

Um neue Befehle für Python kennenzulernen, können wir die **offizielle Python-Dokumentation** lesen oder auf **StackOverflow** nach Antworten auf konkrete Fragen suchen. Beide Quellen sind mit Vorsicht zu genießen, und man sollte nie Code von einer externen Quelle übernehmen, ohne genau zu verstehen, was der Code macht.

Wenn unser Programm einen Fehler enthält, wir aber nicht genau wissen, wo er sich befindet, können wir mit dem **Debugger** das Programm Schritt für Schritt ausführen und immer wieder pausieren, um zu beobachten, welche Werte die Variablen haben und ob das Verhalten des Programms während der Laufzeit unseren Erwartungen entspricht.

## Professioneller Umgang mit Texten

In der Computerlinguistik verarbeiten wir viele Texte. Einige Werkzeuge aus dem Kurs sind dafür besonders wichtig:

Wir können Texte nicht nur in Variablen speichern, sondern sie auch aus **Dateien** einlesen oder Ergebnisse unseres Programms in eine Datei schreiben.

**Reguläre Ausdrücke** ermöglichen uns, Strings zu beschreiben, indem wir Bedingungen formulieren, die die Strings erfüllen sollen. So können wir viele verschiedene Strings in einem einzigen Muster beschreiben und Textdaten gut weiterverarbeiten.

Um Strings für die Ausgabe von Ergebnissen vorzubereiten, verwenden wir **String Formatting**. Damit können wir Platzhalter mit optionalen zusätzlichen Eigenschaften definieren und dann während der Laufzeit Variablenwerte an diesen Stellen im String einfügen.

## Und wie gehts weiter?

Dieser Kurs war so aufgebaut, dass ihr möglichst viele Konzepte und Strategien, die ihr hier gelernt habt, später wiederverwenden könnt:

- **Rekursion** ist hilfreich, wenn wir verschachtelte Strukturen verarbeiten. Syntaxbäume sind Beispiele für solche Strukturen und werden im Kurs "Einführung in die Syntax" (Modul L1, 2. Semester) behandelt.
- **Reguläre Ausdrücke** sind ein Universalwerkzeug, das jede\_r Computerlinguist\_in kennen sollte. Sie werden im Kurs "Einführung in die Computerlinguistik" (Modul CL1, 2. Semester) ausführlicher behandelt.
- **Schleifen, bedingte Ausführung und Funktionen** gibt es in vielen Programmiersprachen. Eure Fähigkeit, Programme zu strukturieren zu planen und umzusetzen, wird in jedem weiteren Programmierkurs hilfreich sein, auch wenn ihr in anderen Sprachen diese Konstrukte mit einer anderen Syntax umsetzt.
- Wir haben einige computerlinguistische Fragestellungen angedeutet und beispielhaft implementiert. Mehr über die theoretischen Hintergründe werdet ihr im Kurs "Einführung in die Computerlinguistik" erfahren.

Wenn ihr euch weiter mit Python beschäftigen und eure bisherigen Fähigkeiten noch weiter ausbauen möchtet, habt ihr im nächsten Semester die Gelegenheit dazu. Mehr Informationen dazu gibt es im nächsten Notebook/Video.

Diese Woche finden keine Übungssitzungen statt. Wer noch Fragen zum Kurs hat oder sich über das weitere Studium unterhalten möchte, kann das in den existierenden Kanälen tun (Rocketchat, Mail, Discord).