Hochschule KarlsruheUniversity of
Applied Sciences

Fakultät für Informatik und Wirtschaftsinformatik



DSCB130 | Informatik für Data Science 1



Matthias Mruzek-Vering M.Sc. Foto: Quirin Leppert

Übersicht



- + Organisation und Überblick
 - Agenda
 - Themen
- + Fachlandkarte
- + Lernergebnisse
 - Allgemein
 - Erweitert
- + Lehrinhalte
 - Vorlesung
 - Übung
 - Hausarbeit

- + Lehrmethode
- + Lernkontrolle
- + Lernunterstützung
- + Begleitmaterialien
- + Einstiegsfrage

Organisation und Überblick

+I

Agenda

- Orientierung über diese und andere Lehrveranstaltungen im Modulhandbuch (pdf)
- + Workload für 6 ECTS Punkte
 - 4 SWS Vorlesung in Präsenz mit Videostream
 - 2 SWS Übung in Präsenz ohne Videostream
- + Tutorium in Präsenz ggf. mit Videostream
- + Sprechstunde: Donnerstags 14 Uhr, Raum E205 und nach Vereinbarung via mrma0001@h-ka.de
- + Terminänderungen im Stundenplan und IWI-News
- + Klausur in Präsenz handschriftlich 90 min (Stand: 26.09.2022)



Organisation und Überblick

┯

Themenblöcke

1 Informationsverarbeitung

2 Rechner und Netze

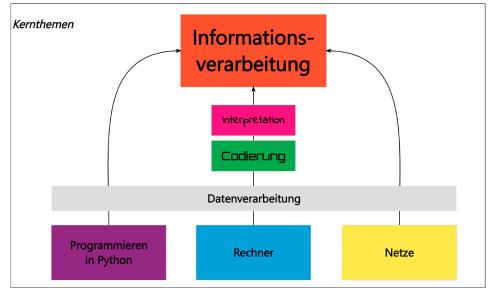
3 Programmieren



Fachlandkarte







Detallierte Ansicht der Fachlandkrate, siehe Collaboard im ILIAS-Kurs

Eigene Darstellung

Hochschule Karlsruhe



Lernergebnisse



Allgemein

Die Studierenden

- können die grundlegenden Elemente von IT-Systemen und deren Programmierung wiedergeben.
- **entwickeln** Fähigkeiten im Umgang mit Daten und Rechnernetze auf einfachem Niveau, um diese **darstellen**, **planen** und **handhaben** zu können.
- werden befähigt, die grundlegenden Konstrukte der Programmiersprache Python wie Variablen, Kontrollstrukturen, Funktionen und Collections zum Lösen einfacher Probleme **anzuwenden**.
- **erwerben vertiefend** Grundbegriffe und Denkweisen der Programmierung anhand der Programmiersprache Python. Dazu gehören: Sprachelemente, Datenabstraktion, Fehlerbehandlung und Laufzeit-Typinformationen.
- erlernen Programmier- und Dokumentationskonventionen, um Python-Programme lesbar zu schreiben.
- erkennen rekursive Problemstrukturen und lösen Sie mit rekursiven Algorithmen.
- wenden ihre Kenntnisse in Übungsaufgaben an.



Lernergebnisse



Erweitert

- + IT-Systeme werden stets weiterentwickelt, um sie robuster, effizienter und nicht zuletzt intelligenter zu machen. Diese Entwicklung beinhaltet die Erfahrung von vielen Jahrzenten.
- + Sie bekommen nur eine **Einführung in die IT-Welt**, die Ihnen zunächst eine Grundlage bietet, funktionierende und sinnvolle Programme zu schreiben sowie IT-Systeme beurteilen zu können.
- + In jedem Fall soll die Vorlesung Ihnen eine Orientierung erlauben darüber, was Sie lernen wollen und wo Sie die notwendigen Informationen finden.



Lehrinhalte der Vorlesung



1 INFORMATIONSVERARBEITUNG

- 1.1 Einführung in die Informatik
- 1.2 Zahlensysteme und binäre Arithmetik
- 1.3 Nachricht und Interpretation
- 1.4 Datenverarbeitung
- 1.5 Information und Codierung

2 RECHNER UND NETZE

- 2.1 Rechnerarchitekturen
- 2.2 Von-Neumann-Architektur
- 2.3 Der typische Desktop-PC und Laptop
- 2.4 Befehlssatz
- 2.5 Befehlsverarbeitung
- 2.6 Speicher
- 2.7 Evolution der der Rechner
- 2.8 Ein- und Ausgabe
- 2.9 Verbindungsstrukturen
- 2.10 Rechnernetze
- 2.11 Netzwerkprotokolle



Lehrinhalte der Vorlesung



3 PROGRAMMIEREN

- 3.1 Einführung in die Programmierung
- 3.2 Einführung in Python
 - Aufbau von Programmen
 - Erstes Programm schreiben
 - Fehler (Syntax und Laufzeit)
 - Variablen und Literale
 - Kommentare und Konventionen
 - Ein- und Ausgabefunktionen
 - Basis Datentypen und deren Umwandlung
 - Operatoren und Ausdrücke
 - Verarbeitung von Zeichenketten
 - Module einbinden
 - Zahlen und Mathe (Ganzzahlen und Gleitkomma)

- 3.3 Datenstrukturen
 - List, Tuple, String, Set, Dictionary
- 3.4 Boolesche Ausdrücke
 - Entscheidungslogik (if, elif, else)
- 3.5 Schleifen
 - for und while
 - break, continue und pass
- 3.6 Funktionen
 - Funktionsargumente
 - Anonyme Funktionen
- 3.7 Scope und Lifetime von Variablen
- 3.8 Module und Dateiverarbeitung



Lehrinhalte der Vorlesung



3.9 Algorithmen-Prinzipien

- Paradigmen
- Teile-und-Herrsche-Prinzip
- Versuch-und-Irrtum-Prinzip
- Heuristik
- Gierige Algorithmen

3.10 Algorithmen

- Begriffsklärung und Historie
- Schema zur Beschreibung von Algorithmen
- Anforderungen und Algorithmen
- Rekursion und Backtracking

3.11 Suchen und Sortieren

- Sequentielle Suche
- Binäre Suche
- Sortierverfahren

3.12 Komplexität

- O-Kalkül (Landau-Notation)
- Vergleich der relativen Laufzeiten von Sortierverfahren
- 3.13 Umsetzung einfacher Algorithmen in Programme
- 3.14 Funktionale Programmierung
 - lampda, map, filter, reduce
 - list-comprehension
 - packing und unpacking
 - zip
 - Elementare Programmiertechniken
- 3.15 Code Bugs finden und beheben
- 3.16 Ausnahmebehandlung



Lehrinhalte der Übung

+I

1 INFORMATIONSVERARBEITUNG

- Zahlensysteme
- Codierung und Interpretation



- Rechnerarchitekturen und Befehlsverarbeitung
- Netzwerktopologien und Netzwerk-Komponenten
- Kommunikations-Protokolle

3 PROGRAMMIEREN

- Zahlen-Umwandlung und wissenschaftliche Darstellung
- Zeichenkettenmanipulation
- Fehlersuche
- Datenstrukturen
- Algorithmen









Lehrinhalte der Hausarbeit



+ Programmierkenntnisse **anwenden**



+ Lösungsorientiertes Handeln, um ein lauffähiges Computerprogramm zu erstellen



+ Selbstständigkeit, Gruppenarbeit und Arbeitsteilung erleben



+ Ihren Code in einem Kolloquium **erklären**



Lehrmethode



Mein Ziel ist es, Sie zum Lernen zu motivieren

- + Wenn Sie Fragen haben, dann sprechen Sie mich an oder schreiben Sie mir eine E-Mail
- + Einstieg und Überblick in neue Themen vermittle ich in der Vorlesung
- + Detailwissen erarbeiten Sie sich im Selbststudium

Bitte verstehen Sie diese Lehr-Strategien als meine Aufgaben

- + **Sandwichstrategie:** Thema > Relevanz > Aktivierung > Präsentation
- + Expositorisches Lehrverfahren:
 Sie üben das Gelernte in Vertiefungs- oder
 Transferaufgaben. Hierzu gibt es die Arbeitsblätter,
 die Übungsaufgaben und die Hausarbeit.

Bitte verstehen Sie diese Lehr-Methoden als meine Aufgaben

- + Ampel-Methode: Umfragen mit Multiple-Response (a|b|c|d) Auswahl
- + **Directed Paraphrasing:**Bitte erklären Sie mit eigenen Worten ...
- + Application Cards:
 Bitte finden Sie Anwendungsbeispiele zu folgendem
 Thema...
- + Kriteriendiskussion:
 Bitte diskutieren Sie einmal und gewichten die Aussagen. Warum dies und nicht das andere?
 Ziel: Fachlich begründen können.
- + 3-2-1 Exit Tickets: Nach jeder Lehreinheit müssen Sie
 - 3 Dinge nennen, die Sie gelernt haben
 - 2 Dinge nennen, die Ihnen gefallen haben
 - eine weitere Frage zum aktuellen Thema stellen



Lernkontrolle



- + Wiederholungsfragen am Ende jedes Themenblocks
- + Tutorium
- + Quizze
- + Online-Tests

Lernunterstützung



Literatur (Auswahl)

+ Literaturverzeichnis in ILIAS

Python Tutorials (Auswahl)

- + Getting Started with Python in VS Code
- + The [official] Python Tutorial
- + Python 3 Tutorial (deutsch)
- + Real Python Tutorials
- + How To Code in Python
- + w3schools.com

Python Online Quizze (Auswahl)

- + Real Python
- + tutorialspoint
- + PyNative

Apps (Auswahl)

- + Learn Python
- + Mimo: Programmieren lernen



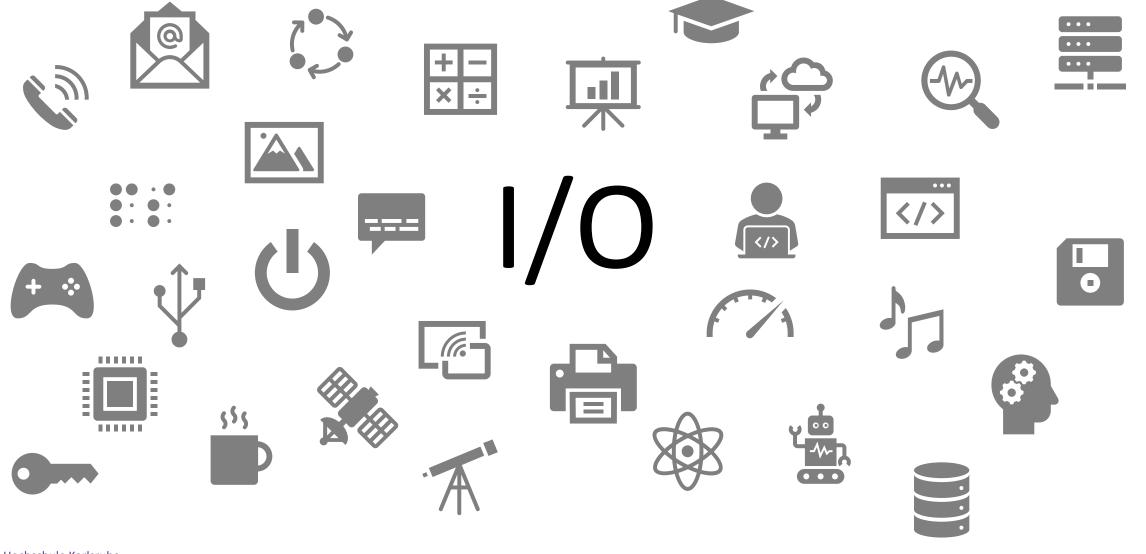
Begleitmaterialien in ILIAS



- + Vorlesungsunterlagen
- + Arbeitsblätter
- + Übungsblätter
- + Hausarbeit
- + Literatur (Auswahl)
- + Programmierumgebung Setup
- + Videoclips zu VS Code und Python Funktionen
- + Data Science Tools (Auswahl)
- + Weiterführendes Material (Web-Links)
- + Klausurvorbereitung: Sammlung an Musterlösungen

Was ist Informatik?







Hochschule Karlsruhe

University of Applied Sciences

Fakultät für

Informatik und Wirtschaftsinformatik



Hochschule Karlsruhe

University of Applied Sciences

Fakultät für

Informatik und Wirtschaftsinformatik

