



Universitätskurs
Python for Science and Engineering
an der
Technischen Universität Graz

Lehrplan

§ 1 Qualifikationsprofil

1. Ziele des Universitätskurses

In einer zunehmend datengetriebenen Welt ist Programmieren eine zentrale Kompetenz in Wissenschaft und Technik. Ob in der Forschung, Entwicklung oder Lehre, die Fähigkeit Daten auszuwerten, Prozesse zu automatisieren und Modelle zu erstellen, wird in nahezu allen Disziplinen vorausgesetzt. Python hat sich dabei als ein de-facto Standard in Bereichen wie wissenschaftliches Rechnen, Simulation, Data Science und numerische Analyse etabliert. Es verbindet einfache Syntax mit hoher Ausdruckskraft, bietet eine große Auswahl spezialisierter Bibliotheken und lässt sich interdisziplinär einsetzen, von der Naturwissenschaft bis zum Ingenieurwesen. Ziel dieses Universitätskurses ist es, die Grundlagen der Programmiersprache Python zu vermitteln und deren Anwendung im wissenschaftlich-technischen Kontext zu erlernen. Die Teilnehmenden arbeiten mit zentralen Sprachelementen (Kontrollstrukturen, Funktionen, Klassen, Datenstrukturen) sowie mit gängigen wissenschaftlichen Bibliotheken.

Python ist nicht nur ein Werkzeug, sondern auch ein Zugang zu modernen wissenschaftlichen Arbeitsweisen. Die Sprache ist offen, modular und reproduzierbar. Dieser Universitätskurs befähigt zur souveränen Anwendung von Python in Forschung und Technik und legt den Grundstein für weiterführende Anwendungen, Innovation und interdisziplinäre Zusammenarbeit.

2. Zielgruppen, an die sich das Angebot richtet

Als Zielgruppe für diesen Universitätskurs lassen sich primär Personen aus der Wissenschaft und Technik definieren, die Python als Werkzeug in ihren jeweiligen Disziplinen nutzen möchten.

„**Modul I: Python - Basics**“ bietet eine zielgruppengerechte Einführung in das erfolgreiche Arbeiten mit Python. Adressiert werden hier insbesondere Fachkräfte aus der Forschung und Entwicklung sowie Softwareentwickler*innen, die sich Grundlagen für die Programmierung in Python aneignen und in ihrem Arbeitsumfeld einsetzen möchten.

„**Modul II: Python - Advanced**“ soll vor allem Fachkräfte aus der Informatik, Softwareentwicklung, Mathematik sowie den Ingenieur- und Naturwissenschaften ansprechen. Modul II baut inhaltlich auf Modul I auf und setzt somit Grundkenntnisse in Python voraus, welche optimalerweise durch Absolvierung von Modul I erlangt wurden. Das Absolvieren von Modul I stellt allerdings keine Voraussetzung dar, wenn sich die Teilnehmer*innen bereits auf anderem Weg diese Grundlagen angeeignet haben.

„**Modul III: Python - Numerical and Scientific Computing**“ richtet sich an Fachkräfte aus Forschung und Entwicklung, die Python in ihrem Arbeitsumfeld einsetzen möchten und eine Alternative oder Ergänzung zu MATLAB suchen. Für die Teilnahme wird die erfolgreiche Absolvierung der Module I und II empfohlen, stellt jedoch keine Zulassungsvoraussetzung dar, solange Grundkenntnisse in numerischer Mathematik sowie Programmieren in Python anderweitig erworben wurden.

3. Zukünftige Arbeitsfelder

Die Absolvent*innen des Universitätskurses sind befähigt, die Programmiersprache Python in unterschiedlichen wissenschaftlich-technischen Bereichen und Kontexten

ihres beruflichen Umfelds einzusetzen, u.a. in der Datenauswertung, in der Automatisierung, in der numerischen Simulation sowie zur Datenanalyse.

4. Lernergebnisse

Je nach gewähltem Modul sind die Teilnehmer*innen nach erfolgreicher Absolvierung des Universitätskurses in der Lage,

Modul I: Python - Basics	• die Grundlagen der Programmiersprache Python zu beherrschen. • einfache Automatisierungsaufgaben mithilfe von Python zu erledigen. • einfache Aufgaben in den Bereichen Datenauswertung zu lösen • und numerisches Rechnen, wie das Einlesen von Datenfiles, zu verstehen und einfachste Berechnungen Linearer Gleichungssysteme zu lösen.
Modul II: Python - Advanced	• erweiterte Funktionen der Programmiersprache Python zu verstehen und anwenden zu können. • komplexerer Automatisierungsaufgaben mithilfe von Python zu erledigen. • komplexere Aufgaben in den Bereichen Datenauswertung zu lösen und numerisches Rechnen, wie das Statistische Erfassen von Daten, zu verstehen und einfache Berechnungen zu lösen sowie Daten zu visualisieren. • und große Datenmengen zu analysieren und verarbeiten.
Modul III: Python - Numerical and Scientific Computing	• wissenschaftliche Prozesse zu automatisieren. • numerische Simulation und Modellierung durchzuführen. • und reproduzierbare Forschung zu schaffen und deren Visualisierung umzusetzen.

5. Lehr- und Lernkonzept

Der Universitätskurs ist modular aufgebaut, wobei für jedes Modul eine Präsenzphase und eine Selbstlern- und Prüfungsvorbereitungsphase vorgesehen ist. Unabhängig vom jeweiligen Modul erfolgt die Wissensvermittlung über einen Vortrag mit integrierten Übungsbeispielen. Die zu behandelnden Inhalte können auch, sofern sie argumentativ sinnvoll mit den zu lehrenden Inhalten kompatibel sind, an die Interessen der Teilnehmer*innen angepasst werden.

6. Beurteilungskonzept

Die Teilnahme an der Präsenzphase ist verpflichtend. Zusätzlich muss ein individuelles Programmierbeispiel abgeliefert werden. Die Beurteilung erfolgt nach Absolvierung der Präsenzphase und auf Basis der Ergebnisse der Arbeit, die in der Selbstlernphase von den Teilnehmenden erarbeitet wird. Die Ergebnisse sollen in Form einer Präsentation oder eines Prüfungsgesprächs abgeliefert werden. Die Art der Präsentation muss im Vorhinein mit der wissenschaftlichen Leitung abgesprochen werden. Die Beurteilung erfolgt durch die wissenschaftliche Kursleitung.

Module – Lehrveranstaltungen	Stunden	ECTS
Modul I: Python - Basics Präsenzphase - Einfache Datentypen, Container - Imperative Aspekte: Kontrollstrukturen (if, for, while), Funktionen - Objektorientierte Aspekte: Klassen, Objekte, Methoden - In/Output, File handling - Kurzüberblick für Daily-Use Tools: - o Pandas, excel-manipulation (xlsxwriter, openpyxl), DateTime... - o Plots (matplotlib) - o advanced (os, sys, filedialog (tkinter), nptdms...)	16	0,7
Selbstlern- und Prüfungsvorbereitungsphase	8	0,3
Gesamt Modul I	24	1
Modul II: Python - Advanced Präsenzphase - Erweiterte Objektorientierte Aspekte: Vererbung, Operator Overloading, Exceptions	16	0,7

- Funktionale Aspekte: Lambda Operator, Map, Filter, List Comprehensions, Iterators, Generators - Module und Namespaces - Besonderheiten der Sprache: Duck Typing, Doc Strings, Manipulation von Objekten und Klassen zur Laufzeit, Exceptions - Datenmodell - Erweiterte Themen: - o Decorators, Changing Classes on Runtime, exec function - o Advanced Filehandling: xml, json, yaml - o Software Development: Unittests, Doctests, Python Debugger - o Kurzeinführung: numpy - o Machine-learning: keras Einführung - o GUI: tkinter - o Codeoptimierung: ctypes, Cython		
Selbstlern- und Prüfungsvorbereitungsphase	8	0,3
Gesamt Modul II	24	1
Modul III: Python - Numerical and Scientific Computing Präsenzphase - Grundlagen für das Arbeiten mit wissenschaftlichen Bibliotheken in PYTHON 3 - Numpy Grundlagen: Arrays, Arithmetik, Indizierung - Lineare Algebra mit Numpy/Scipy (u. a. lineare Gleichungssysteme, Matrixoperationen, Sparse Matrices) - Nichtlineare Probleme mit Scipy: Nullstellensuche, Datafitting - Optimierung mit Scipy und CVXOPT - Erstellen von Grafiken mit Matplotlib - Tuning numerischer Codes: Vektorisieren, Cython und F2PY - Ausblick auf weiterführende Anwendungen: Computeralgebra mit SageMath, FEM mit Salome und Code_Aster	16	0,7
Selbstlern- und Prüfungsvorbereitungsphase	8	0,3
Gesamt Modul III	24	1
Gesamt	72	3

§ 5 Prüfungsordnung

Für den Abschluss bestehen folgende Möglichkeiten:

- Abschluss des Universitätskurses in vollem Umfang (Modul I, II und III)
- Getrennte Leistungsfeststellung zum separaten Abschluss der einzelnen Module I, II oder III.

Die Leistungsfeststellung muss spätestens bis sechs Monate nach der Präsenzphase erfolgen. Bei negativer Prüfungsleistung besteht die Möglichkeit, die jeweilige Modulabschlussprüfung zu wiederholen. Die Prüfungswiederholung muss bis spätestens ein Jahr nach dem Kursende erfolgen.

Die zugehörige Feststellung des Prüfungserfolgs obliegt der oder dem Vortragenden.

§ 6 Abschluss

Nach positivem Abschluss des Universitätskurses wird von der Technischen Universität Graz ein Zertifikat verliehen. Sollten nicht alle Module des Universitätskurses erfolgreich absolviert werden, so führt die erfolgreiche Absolvierung eines Einzelmoduls zur Verleihung eines Microcredentials mit je 1 ECTS. Teilnehmende, welche keine Prüfung ablegen, erhalten eine Teilnahmebestätigung der TU Graz.

§ 7 Universitätskursbeitrag

Der Universitätskursbeitrag schließt nur die Kosten des Universitätskurses gemäß § 8 für die Lehrveranstaltungen ein. Der Kursbeitrag ist der aktuellen Information auf der Homepage von TU Graz Life Long Learning zu entnehmen.

Die TeilnehmerInnen dieses Universitätskurses haben nur den Universitätskursbeitrag, nicht aber den Studienbeitrag zu entrichten. Sollten die TeilnehmerInnen als außerordentliche Hörer inskribiert sein, ist auch der ÖH-Beitrag zu bezahlen.

§ 8 Kosten des Universitätskurses

Die Kosten des Universitätskurses setzen sich aus den Aufwendungen für die Lehrenden und den sonstigen Aufwendungen für Leitung, Organisation, Durchführung der Kurse, Probenmaterialien etc. zusammen. Die dafür erforderlichen Mittel werden aus dem Universitätskursbeitrag und gegebenenfalls aus Drittmitteln aufgebracht. Der Universitätskurs kann nur abgehalten werden, wenn die für die Durchführung erforderlichen Mittel in entsprechender Höhe zur Verfügung stehen.

§ 9 Durchführung des Universitätskurses

Der Universitätskurs wird von TU Graz Life Long Learning durchgeführt. Die wissenschaftliche Leitung nimmt DI Dr. Stefan H. Reiterer wahr.

§ 10 Inkrafttreten

Der Lehrplan tritt am Tag nach der Verlautbarung im Mitteilungsblatt der TU Graz in Kraft.

Univ.-Prof. Dipl. Ing. Dr. techn. Stefan Vorbach

Vizerektor für Lehre
TU Graz