

Einführung in die Programmierung für Studierende der Physik

*JOHANN WOLFGANG GOETHE UNIVERSITÄT
12.04.2022*

MATTHIAS HANAUSKE

*FRANKFURT INSTITUTE FOR ADVANCED STUDIES
JOHANN WOLFGANG GOETHE UNIVERSITÄT
INSTITUT FÜR THEORETISCHE PHYSIK
ARBEITSGRUPPE RELATIVISTISCHE ASTROPHYSIK
D-60438 FRANKFURT AM MAIN
GERMANY*

1. Vorlesung

Plan für die heutige Vorlesung

- Internetseite der Vorlesung
- Lernplattform OLAT
- Warum und wozu Programmierung?
- Was benötigen Sie? *Computer, Betriebssystem, C++, Python*
- Das Betriebssystem Linux: Ein kurzer Überblick
- Programmiersprachen: Ein kurzer Überblick
- Übungsaufgaben, Tutorien und der Login-Account für die ITP-Rechner

Allgemeines zur Vorlesung

- Ort und Zeit:
Die Vorlesungen finden jeweils dienstags von 15.00-16.00 Uhr und donnerstags von 14.00-16.00 Uhr im Raum Phys-0.111 statt
Vielleicht hybride Online-Version (Zoom Meetings):
ID: 794 847 5614, PWD: 785453
- Vorlesungs-Materialien (Online):
<http://th.physik.uni-frankfurt.de/~harauske/VPROG/>
- Übungsgruppen/Praktika auf der Online-Lernplattform OLAT:
<http://olat.server.uni-frankfurt.de>
- Generelles zur Vorlesung:
Bei erfolgreicher Teilnahme 6 Creditpoints
Klausurtermine stehen noch nicht fest
- Voraussetzungen:
Laptop/PC mit C++ und Python

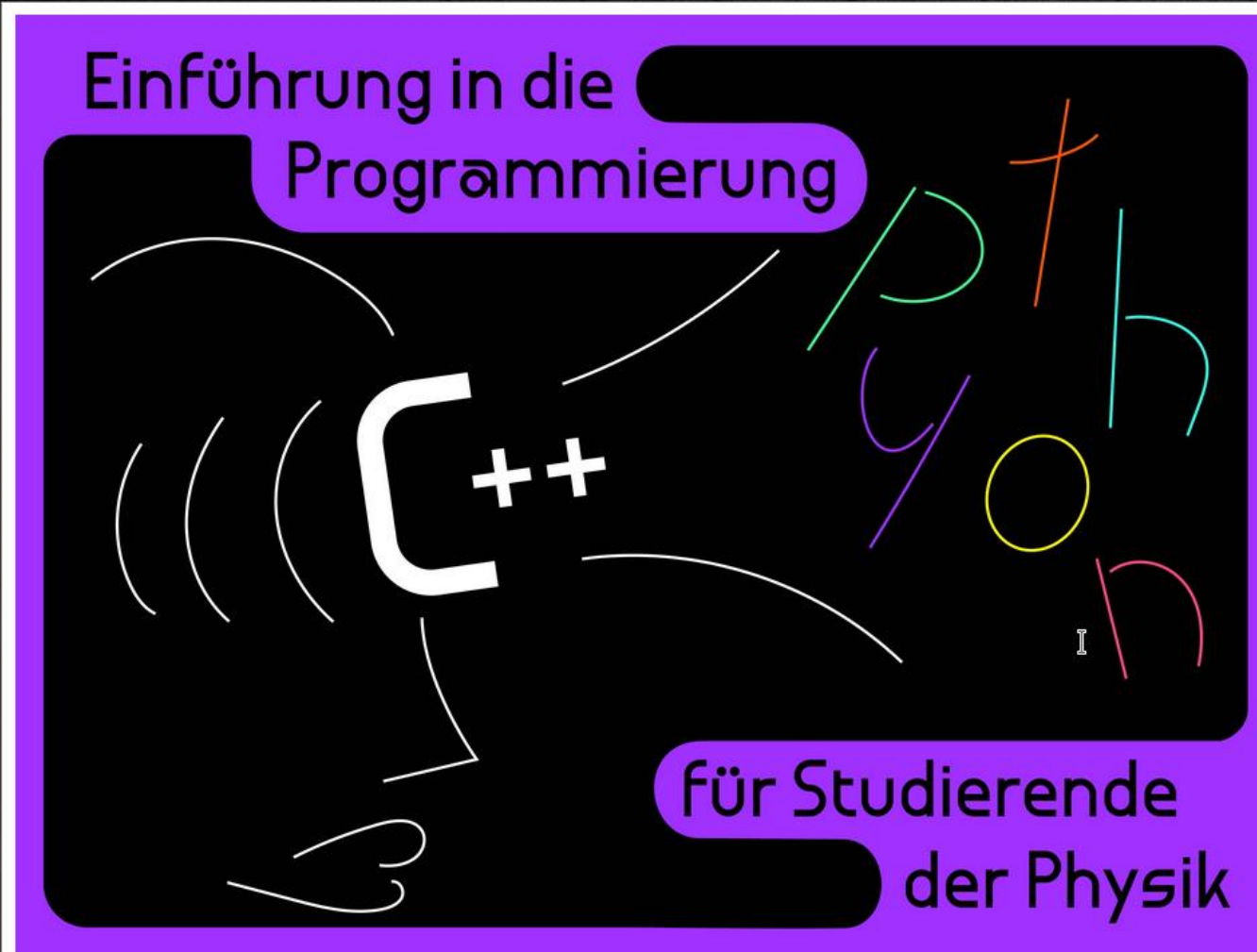


Illustration: Deborah Moldawski

Nächster Zoom Link am 12.04.2022, 15:00-16:00 Uhr: ID: 794 847 5614, PWD: 785453

**Einführung in die Programmierung für Studierende der Physik
(Introduction to Programming for Physicists)
Vorlesung SS 2022**

Diese Internetseite fasst die Online-Angebote der Vorlesung *Einführung in die Programmierung für Studierende der Physik* zusammen. Die Vorlesungstermine finden jeweils dienstags von 15.00-16.00 Uhr und donnerstags von 14.00-16.00 Uhr im Raum Phys-0.111 statt. Die Termine der Übungen/Praktika finden Sie auf der [Online-Lernplattform OLAT](#) und die Übungsaufgaben werden im linken Panel unter der jeweiligen Vorlesung und zusätzlich auf [OLAT](#) bereitgestellt.

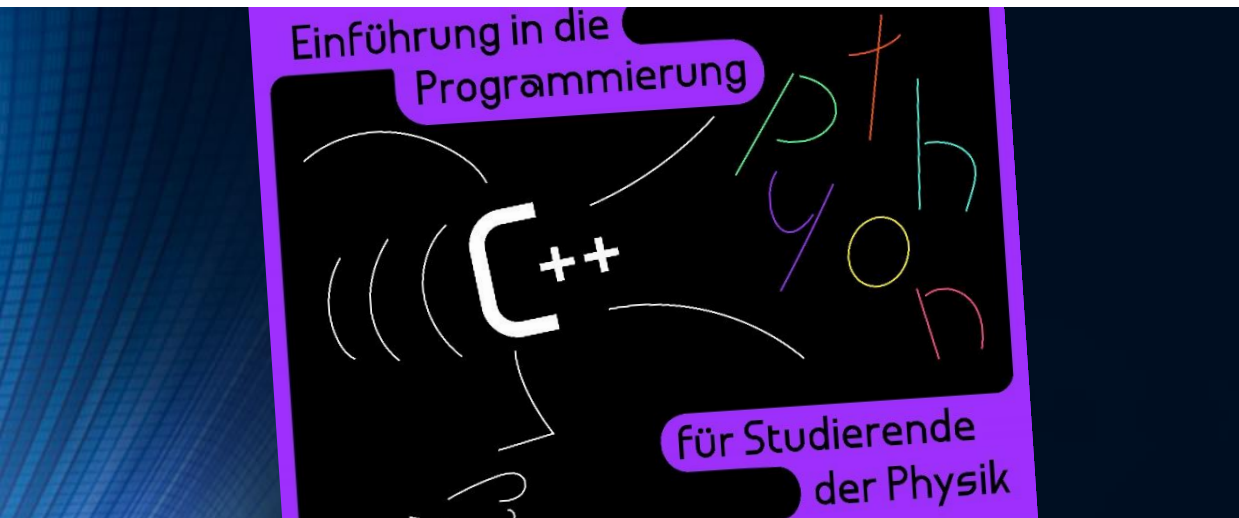
Die Vorlesung gibt einerseits eine Einführung in die Objektorientierte Programmiersprache C++ und vermittelt andererseits einige wesentliche Grundlagen der numerischen Mathematik. Es werden die grundlegenden Elemente der Programmiersprache, das Programmierparadigma der Objektorientierung und Simulationen von komplexen physikalischen Problemen behandelt. Das Hauptanliegen der Vorlesung besteht darin, dass die Studierenden die numerische Lösung eines komplexen physikalischen Problems auf dem Computer erstellen können. Der Schwerpunkt wird hierbei auf der Programmiersprache C++ liegen, wobei für die Visualisierung der berechneten Daten die Skriptsprache Python benutzt wird. Zusätzlich werden die berechneten mathematisch/physikalischen Gleichungen mittels Python Jupyter Notebooks analysiert und illustriert.

Literatur zu C++

- Prof. Dr. Marc Wagner, Vorlesung im WS 2019/20: [Einführung in die Programmierung für Physiker](#)
- B. W. Kernighan, D. M. Ritchie, Hanser: [Programmieren in C](#)
- Bjarne Stroustrup 2015: [Die C++ Programmiersprache](#)
- Bjarne Stroustrup 2009: [Programming: Principles and Practice Using C++](#)
- Prof. Dr. Claudius Gros, Vorlesung im WS 2021/22: [Advanced](#)

Diese Internetseite fasst die Online-Angebote der Vorlesung *Einführung in die Programmierung für Studierende der Physik* zusammen. Die

Vorlesungstermine finden jeweils dienstags von 15.00-16.00 Uhr und donnerstags von 14.00-16.00 Uhr im Raum Phys-0.111 statt. Die Termine der Übungen/Praktika finden Sie auf der Online-Lernplattform OLAT und die Übungsaufgaben werden im linken Panel unter der jeweiligen Vorlesung und zusätzlich auf OLAT bereitgestellt.



Die Vorlesung *Einführung in die Programmierung für Studierende der Physik* stellt ein Pflichtmodul im Bachelor Studium Physik der Goethe-Universität Frankfurt dar. Bei regelmäßiger und erfolgreicher Teilnahme an den Übungen/Praktika erhalten Sie eine Zulassung zur Klausur. Den benoteten Schein und sechs Creditpoints erhält man schließlich bei bestandener Klausur. Falls Sie bereits in einem vergangenen Semester (nach der alten Studienordnung) die Zulassung zur Klausur erhalten haben, können Sie direkt an der abschließenden Klausur teilnehmen. Jedoch rate ich Ihnen, die Vorlesung und die Übungen/Praktika trotzdem nochmals zu belegen, da sich die Inhalte und Schwerpunkte zu den vergangenen Vorlesungen unterscheiden könnten.

Literatur zu C++

- Prof. Dr. Marc Wagner, Vorlesung im WS 2019/20: Einführung in die Programmierung für Physiker
 - B. W. Kernighan, D. M. Ritchie, Hanser: Programmieren in C
 - Bjarne Stroustrup 2015: Die C++ Programmiersprache
 - Bjarne Stroustrup 2009: Programming: Principles and Practice Using C++
 - Prof. Dr. Claudius Gros, Vorlesung im WS 2021/22: Advanced Introduction to C++, Scientific Computing and Machine Learning
 - WikiBooks: C++ Programmierung
 - C++ reference
 - cplusplus.com

Literatur zu Python

- Python-Onlinekurs auf Deutsch
 - Python 3 documentation
- Hans Petter Langtangen: A Primer on Scientific Programming with Python
 - David M. Beazley: Python - Essential Reference
 - B. Slatkin: Effective Python

Literatur zur Numerischen Mathematik

- R.L. Burden und J.D. Faires 2000: Numerische Methoden

Die OLAT Seite

<http://olat.server.uni-frankfurt.de>



Auf der OLAT Seite der Vorlesung finden Sie Informationen zur Organisation der Übungen

Vorlesung (Beginn 12.04.):		Dozent:
Di 15-16 Uhr		Matthias Hanauske (hanauske@fias.uni-frankfurt.de)
Do 14-16 Uhr		
jeweils Raum Phys 0.111		
Kurswebsite		
Übungen (Beginn 18.04.):		Tutorienleitung:
Raum Phys 01.120		Niklas Götz (goetz@fias.uni-frankfurt.de)
Übungsgruppen		Tutoren
Tutorium 1 (Di 9.15 - 11.00 - Englisch)		Anjlit Dutia (dunia@tp.uni-frankfurt.de)
Tutorium 2 (Di 11.15 - 13.00 - Englisch)		Youjiang Xu (yxu@tp.uni-frankfurt.de)
Tutorium 3 (Mi 9.15 - 11.00)		Nils Saß (nass@fias.uni-frankfurt.de)
Tutorium 4 (Mi 14.15 - 16.00)		Keneth Miller (millerkeneth@gmail.com)
Tutorium 5 (Do 9.15 - 11.00)		Justin Mohs (mohs@fias.uni-frankfurt.de)
Tutorium 6 (Fr 9.15 - 11.00)		Michael Fromm (mfromm@tp.uni-frankfurt.de)

Organisation der Übungen

Hier findet ihr alle wichtigen Informationen zu den Übungen

Aus- und Abgabe der Übungen

- In jeder Woche des Semesters wird auf der [Kurswebsite](#) das Skript zur jeweiligen Woche sowie das zugehörige Übungsblatt veröffentlicht.
- Für jedes Übungsblatt habt ihr bis Freitag 23:59 Uhr der darauf folgenden Woche Zeit, die Lösung hier in OLAT hochzuladen. Nutzt dazu das entsprechende Aufgabefeld unter "Kursinhalt".
- Sofern nicht anders von eurer*em Tutor*in gewünscht, ladet ihr dort eine .zip-File hoch, welche für jede Aufgabe einen einzelnen Ordner enthält. Für Programmieraufgaben werden dort die Quellcode-Files (KEINE kompilierten Programme) und eine .txt-File mit dem Befehl zum Kompilieren erwartet. Nutzt Kommentare, um auf weiterführende Fragen in der Aufgabe einzugehen und euren Lösungsweg zu erläutern. Für Aufgaben, bei denen ihr Bugs sucht, wird eine verbesserter Code und Erläuterungen der Fehler erwartet.
- Stellt sicher, dass sich eure Abgabe kompilieren lässt, da die*der Tutor*in ansonsten auf diesen Teil der Abgabe keine Punkte geben muss.
- Die Musterlösungen der Übungen werden auf der [Kurswebsite](#) in der Woche nach der Korrektur veröffentlicht. Sie stellen dabei nur einen Lösungsweg dar.
- Über OLAT informieren euch die Tutor*innen in der Woche während der Besprechung des Übungsblatts über eure erreichte Punktzahl und die Verbesserungsvorschläge.

Format der Übungen

- In den Übungen herrscht Anwesenheitspflicht. Solltet ihr verhindert sein, gebt bitte eurer*em Tutor*in im Vorfeld Bescheid.
- Kerninhalt der Übungen ist die Vorstellung eurer Lösungsverschlüsse. Jeder von euch sollte mehrmals im Semester seine Lösung präsentieren. Hierzu stellt ihr zu einer Aufgabe eure Abgabe in OLAT euren Mitstudierenden vor, erklärt euren Ansatz und diskutiert diesen mit der*dem Tutor*in.
- Gegebenenfalls werden in den Übungen auch Inhalte der Vorlesungen vertieft oder auf häufige Probleme in den Abgaben eingegangen.
- In zweiten Woche des Semesters, wenn es noch keine Abgaben zu Besprechen gibt, helfen euch die Tutor*innen beim Aufsetzen eurer Entwicklungsumgebung.

Zulassung zur Klausur

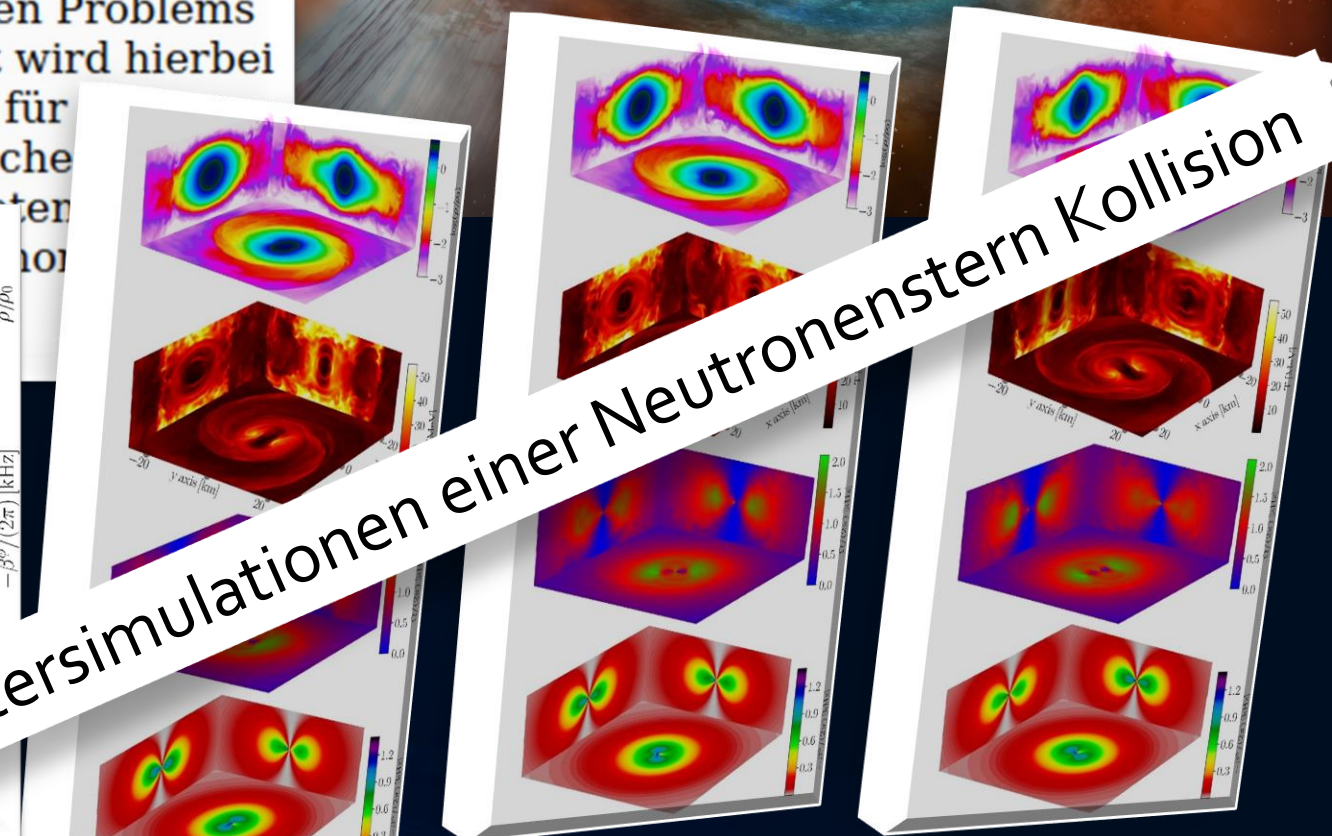
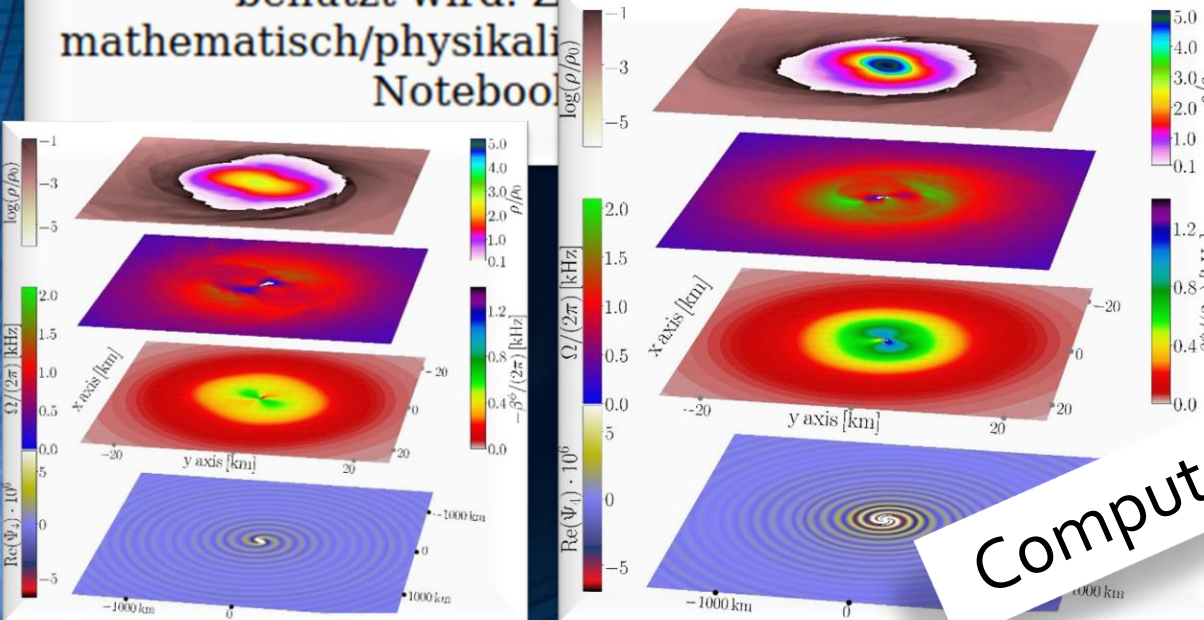
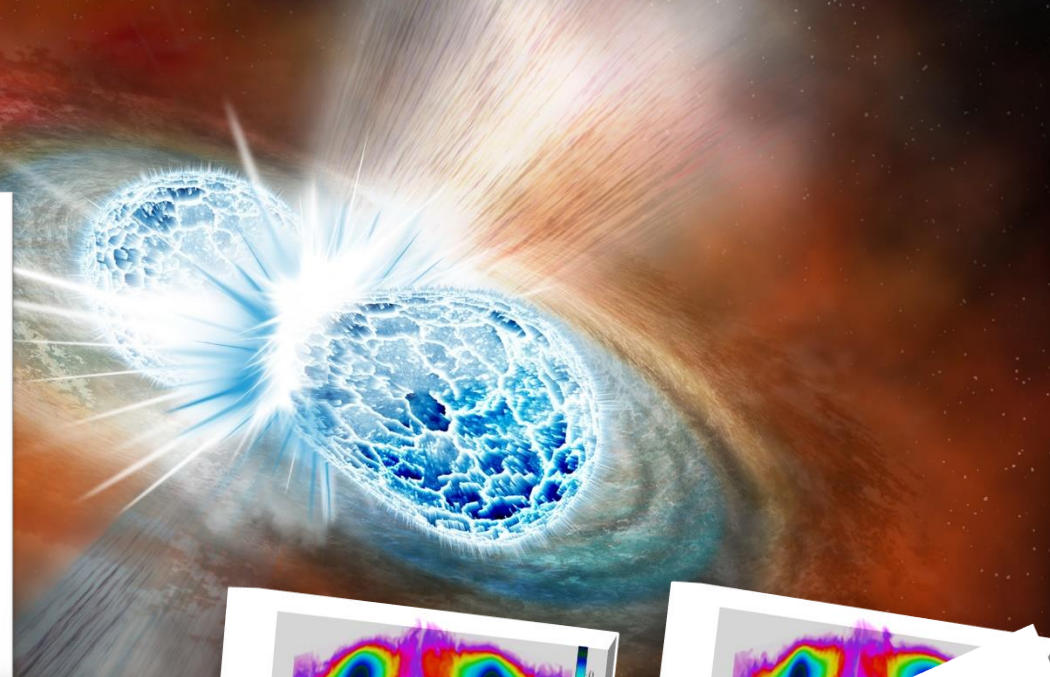
- Um an der Klausur teilzunehmen, müssen mindestens 50% der Punkte in den Übungsaufgaben erreicht werden. Daneben müsst ihr regelmäßig zu den Tutorien erschienen sein und mindestens einmal eure Lösung vorgestellt haben.
- Solltet ihr bereits einen Schein von einem vorherigen Semester besitzen, gebt bitte [Niklas](#) Bescheid. Aktive Teilnahme an den Übungen und der Vorlesung ist dennoch dringend empfohlen.

Entwicklungsumgebung

- Am einfachsten ist es, wenn ihr von eurem eigenen Computer/Laptop entwickelt. Solltet ihr nur ein Windows oder macOS Betriebssystem zur Verfügung haben, müsst ihr zur Nutzung des Terminals ggf. einige Anpassungen vornehmen. Hierzu gibt es sehr viele Tutorials im Internet, eure*er Tutor*in kann euch hierbei auch behelflich sein.
- Steht euch kein Computer zur Verfügung, oder ihr möchtet mit einem bereits fertig eingerichteten System arbeiten, so gibt es die Möglichkeit an den PCs im Übungsraum zu arbeiten. Diese haben ein Ubuntu-Betriebssystem und alle notwendigen Programme bereits installiert. Wir stellen euch hierfür die Login-Daten zur Verfügung. Ihr könnt dann von zu Hause aus mittels ssh auf die PCs zugreifen (mehr Details dazu in der Vorlesung)
- Welche Tools ihr zum Programmieren verwenden wollt, ist euch freigestellt. Theoretisch reichen das Terminal, ein Compiler, ein Texteditor und Jupyter für die Vorlesung aus. Moderne integrierte Entwicklungsumgebungen (IDE) können euch jedoch helfen, Probleme in eurem Code zu erkennen und schneller zu programmieren. Ein Beispiel für eine anfangersfreundliche IDE ist [Visual Studio Code](#). Solche IDEs werden in allen Bereichen von Forschung und Wirtschaft beim Programmieren verwendet, weshalb es sinnvoll ist, sich damit vertraut zu machen. Solltet ihr beim Installieren oder bei der Nutzung auf Probleme stoßen, könnt ihr euch an eure*n Tutor*in wenden.

Warum und wozu Programmierung?

Die Vorlesung gibt einerseits eine Einführung in die Objektorientierte Programmiersprache C++ und vermittelt andererseits einige wesentliche Grundlagen der numerischen Mathematik. Es werden die grundlegenden Elemente der Programmiersprache, das Programmierparadigma der Objektorientierung und Simulationen von komplexen physikalischen Problemen behandelt. Das Hauptanliegen der Vorlesung besteht darin, dass die Studierenden die numerische Lösung eines komplexen physikalischen Problems auf dem Computer erstellen können. Der Schwerpunkt wird hierbei auf der Programmiersprache C++ liegen, wobei für die Visualisierung der berechneten Daten die Skriptsprache Matplotlib benutzt wird. Zusätzlich werden die berechneten Daten in einem Jupyter Notebook dargestellt.



Computersimulationen einer Neutronenstern Kollision

MATTHIAS HANAUSKE
FRANKFURT INSTITUTE FOR ADVANCED STUDIES
JOHANN WOLFGANG GOETHE UNIVERSITÄT
INSTITUT FÜR THEORETISCHE PHYSIK
ARBEITSGRUPPE RELATIVISTISCHE ASTROPHYSIK
D-60438 FRANKFURT AM MAIN

JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

ICRANet
International Center for Relativistic Astrophysics Network

Gravitational-wave signatures of the hadron-quark phase transition in binary compact star mergers

Parallel session: Numerical Relativity and Gravitational Wave Observations
05.07.2021, 17:50

*In collaboration with Lukas Weih, Elias R. Most,
Jens Papenfort, Luke Bovard, Gloria Montana,
Laura Tolos, Jan Steinheimer, Anton Motornenko,
Veronica Dexheimer, Horst Stöcker, and Luciano Rezzolla*

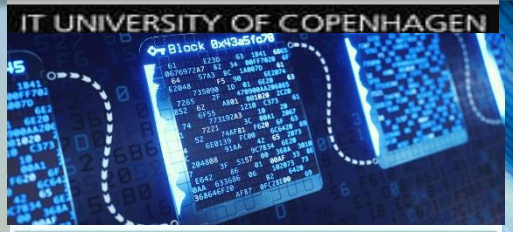
MG16 5-10 JULY 2021
SIXTEENTH MARCEL GROSSMANN MEETING
ON RECENT DEVELOPMENTS IN THEORETICAL AND EXPERIMENTAL GENERAL RELATIVITY, ASTROPHYSICS AND RELATIVISTIC FIELD THEORIES

VIRTUAL MEETING
websites:
<http://www.kira.it/mg16/>
<https://indico.icranet.org/event/1/>
email:
mg16@icranet.org
6:30-19:30 CENTRAL EUROPEAN SUMMER TIME

50TH ANNIVERSARY OF
"INTRODUCING THE BLACK HOLE"



MATTHIAS HANAUSKE
FRANKFURT INSTITUTE FOR ADVANCED STUDIES
JOHANN WOLFGANG GOETHE UNIVERSITÄT
INSTITUT FÜR THEORETISCHE PHYSIK
ARBEITSGRUPPE RELATIVISTISCHE ASTROPHYSIK
D-60438 FRANKFURT AM MAIN



EUROPEAN
BLOCKCHAIN
CENTER

Quantum Game Theory and Social Dilemmas

*Talk at the IT University of Copenhagen
Organized by the European Blockchain Center
Copenhagen, Denmark
28.03.2022*



MATTHIAS HANAUSKE
 FRANKFURT INSTITUTE FOR ADVANCED STUDIES
 JOHANN WOLFGANG GOETHE UNIVERSITÄT
 INSTITUT FÜR THEORETISCHE PHYSIK
 ARBEITSGRUPPE RELATIVISTISCHE ASTROPHYSIK
 D-60438 FRANKFURT AM MAIN



Lateral Thoughts: Matthias Hanauske

Black holes and the German Reichstag

One day a couple of years ago I was attending a meeting of the German Astronomical Society in Berlin, when I was gripped with an almost irreplaceable sense of inner unrest. There was no other option – I simply had to leave the lecture halls of the Technical University and carefully tape the entrance outside. Before I left, however, I carefully taped my poster to the wall between the entrances to the men's and women's toilets, which seemed the perfect spot for it. Every congress delegate would now be forced – subliminally at least – to notice my creation.

After leaving the university buildings, I first soaked up the summer sunshine in the zoological gardens before heading towards the Reichstag – the home of the German parliament. As I did so, my thoughts wandered off in a different direction. What a waste of time. What physics desperately needs are. What an exciting way of presenting

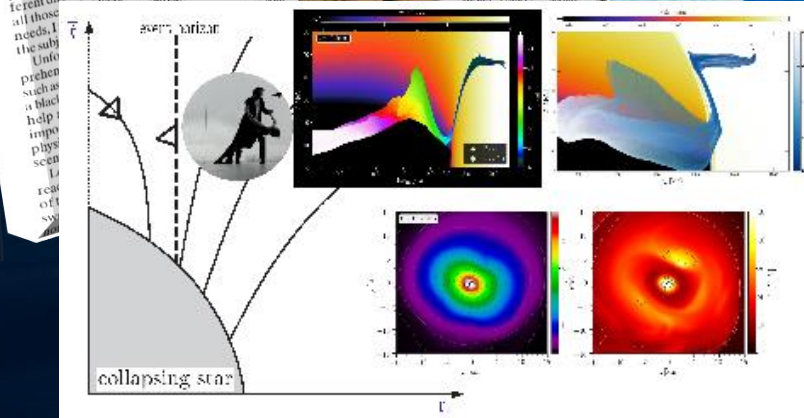
A Report to an Academy

Humanity is of a contradictory nature. On the one hand, humans are able to understand and analyze the evolution of the whole universe and brilliant ideas like the prediction of gravitational waves by Albert Einstein and black holes have been recently observed. On the other hand, the dominant nature of man manifests itself in a stupidity that screams to heaven

<http://itp.uni-frankfurt.de/~hannauske/new/etc/pdf/MG16-Hanauske.pdf>

General Relativity in the Theater of the Absurd

Parallel session: Education
 Teaching Einsteinian Physics to School Students
 08.07.2021, 17:20



MG16 5-10 JULY 2021

SIXTEENTH MARCEL GROSSMANN MEETING

ON RECENT DEVELOPMENTS IN THEORETICAL AND EXPERIMENTAL GENERAL RELATIVITY, ASTROPHYSICS AND RELATIVISTIC FIELD THEORIES

VIRTUAL MEETING
 websites: <http://www.icranet.org/mg16/> <https://indico.icranet.org/event/11/>
 email: mg16@icranet.org
 6:30-19:30 CENTRAL EUROPEAN SUMMER TIME

50TH ANNIVERSARY OF "INTRODUCING THE BLACK HOLE"

"Peace cannot be kept by force. It can only be achieved by understanding."
 Albert Einstein

Warum und wozu

Programmierung

Keine analytische Lösung des betrachteten Systems möglich

Computer Simulation

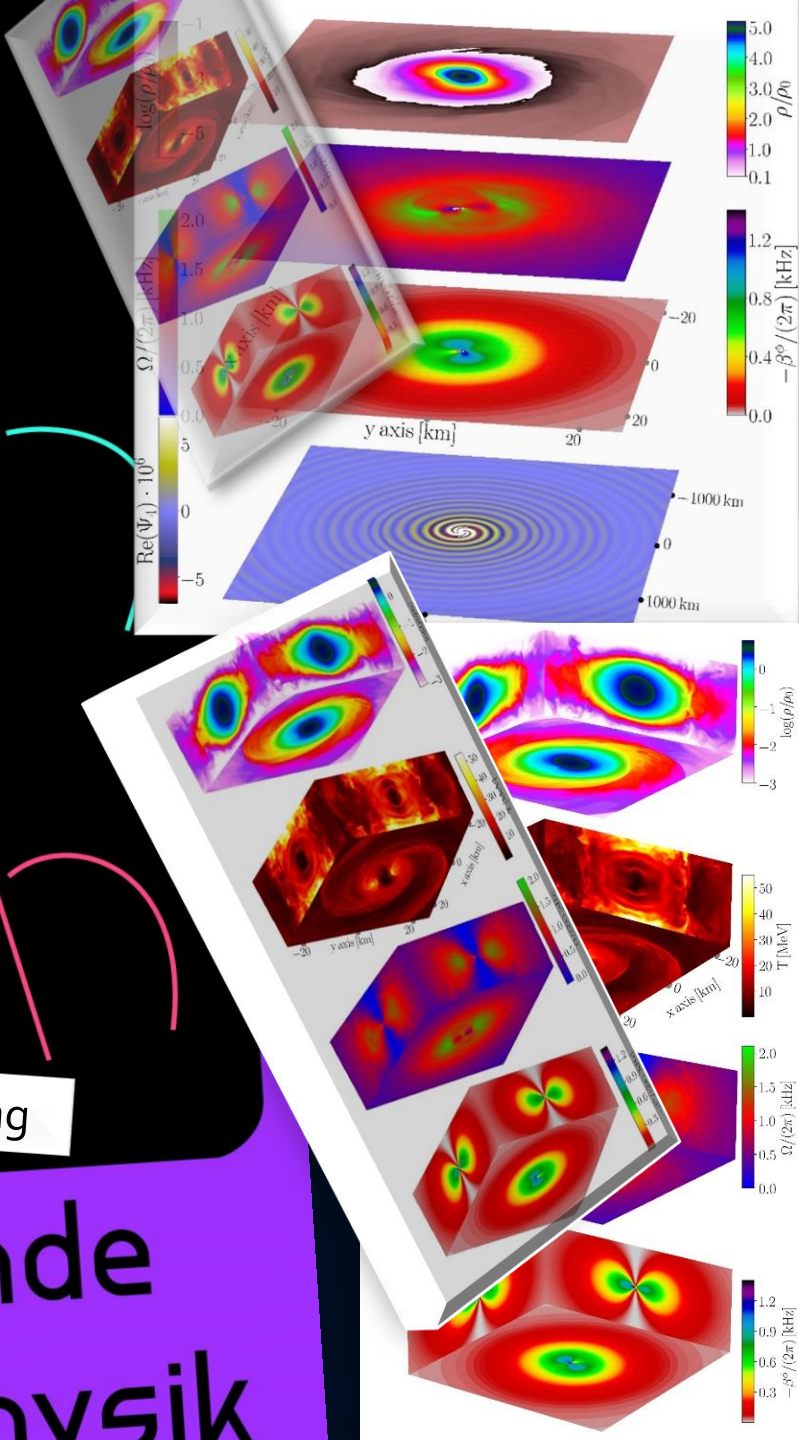


einstein toolkit

Visualisierung

für Studierende der Physik

$$R_{\mu\nu} - \frac{1}{2}R g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$



Eine kleine Einführung in Linux

Damit man ein Computerprogramm erstellen und ausführen kann, benötigt man zunächst einen Computer (PC oder Laptop) und ein Betriebssystem, welches die Systemkomponenten des Computers (z.B. Arbeitsspeicher, Festplatten) verwaltet.

Betriebssysteme, wie Windows von Microsoft, macOS von Apple oder Linux bilden die Schnittstelle zwischen den Hardware-Komponenten und der

Anwendungssoftware des Benutzers des Computers. Das frei erhältliche Betriebssystem Linux wird weltweit auf vielen Großrechnern, Computerclustern und Supercomputern eingesetzt und bildet auch, durch seine führende Rolle als Internet-Server-System, die grundlegende Struktur des Internets. Man kann mittlerweile Linux relativ einfach auf seinem eigenen Computer installieren (siehe

Übungsaufgabe 1.1) und mittels einiger Linux-Shell-Befehle ist eine kommandobasierte Benutzung des Computersystems möglich. Zusätzlich wird in diesem Unterpunkt auch vorgestellt, wie man sich mittels des kryptografischen Netzwerkprotokolls SSH auf die Rechner des Instituts für Theoretische Physik der Goethe-Universität einloggt. Die dafür nötigen Login-Accounts werden in den ersten beiden Vorlesungen an die Studierenden verteilt. Beim Klicken auf das nebenstehende Bild erfahren Sie mehr.



Programmiersprachen

Höhere Programmiersprachen gibt es schon seit den 1950er Jahren und die älteste noch in weitem Gebrauch befindliche Programmiersprache ist *Fortran* (FORmula TRANslator). Die Programmiersprache *BASIC* (Beginner's All-purpose Symbolic Instruction Code) wurde Ende der 1970er Jahre, aufgrund der für Jedermann erschwinglichen Heimcomputern (z.B. der Commodore C-64) populär. Die Programmiersprache C entstand 1972 und die Objekt-orientierte Variante von C (C++) wurde im Jahre 1983 von Bjarne Stroustrup vorgestellt. In der Vorlesung wird der Schwerpunkt auf der Programmiersprache C++ liegen und für die Visualisierung der berechneten Daten wird die Skriptsprache *Python* benutzt, die Anfang der 1990er Jahre entwickelt wurde (siehe [Zeittafel der Programmiersprachen](#)). Die Installation von C++ und *Python* ist auf allen gängigen Betriebssystemen möglich und wird am Beispiel des Betriebssystems Linux unter dem folgenden Link kurz beschrieben: [Programmiersprachen](#).

Anwendungsbeispiel: Erstellen einer eigenen Internet-Homepage

In diesem Unterpunkt werden einige der besprochenen Linux-Shell-Befehle angewendet, um die Erstellung einer eigenen Internet-Homepage zu realisieren. Zusätzlich wird das *Secure File Transfer Protocol (SFTP)* vorgestellt, mit dem man in verschlüsselter Form von einem Rechner auf einen anderen Rechner Daten übertragen kann (näheres siehe [Anwendungsbeispiel: Erstellen einer eigenen Internet-Homepage](#)).



Eine kleine Einführung in Linux

Das Betriebssystem *Linux* ist das empfohlene Betriebssystem dieser Vorlesung, weil es einerseits kostenlos verfügbar (siehe Unterpunkt "Installation von Linux") ist und zahlreiche kostenlose Programme für praktisch jede wissenschaftliche (und nicht-wissenschaftliche) Anwendung existieren (z.B. Office, Editoren, Compiler, Plotprogramme, ...). *Linux* ist an mathematischen und physikalischen Fachbereichen von Universitäten weit verbreitet und wird an vielen Großrechnern, Computerclustern und Supercomputern im Bereich der Forschung eingesetzt. Ich empfehle allen Studierenden dieser Vorlesung, sich zumindest als Zweitsystem Linux zu installieren und die Programmierung auf einem Linux System auszuführen. Falls man sich dennoch gegen eine Installation entscheidet, so kann man die C++ und Python Programme des Kurses auch auf einem anderen Betriebssystem, z.B. Windows oder macOS bearbeiten, kompilieren und ausführen. Eine weitere Möglichkeit, das Linux Betriebssystem zu nutzen, ohne es zu installieren, besteht in einer Secure Shell Verbindung (SSH Verbindung mittels eines kryptografischen Netzwerkprotokolls), welches im Unterpunkt "Login auf die Rechner des Instituts für Theoretische Physik (ITP) der Goethe-Universität" beschrieben wird.

Linux ist ein freies, Unix-ähnliches Mehrbenutzer-Betriebssystem, welches auf dem Linux-Kernel basiert. Die historische Entwicklung des Betriebssystems begann im Jahre 1991, als Herr Linus Torvalds einen Terminal-Emulator entwickelte, welcher dann ab 1992 unter der freien Lizenz GPL verbreitet wurde. Linux, lässt sich als alleiniges Betriebssystem betreiben, aber kann auch innerhalb eines Multi-Boot-Systems eingesetzt werden. Es existieren mehrere unterschiedliche moderne Linux-Distribution (z.B. OpenSUSE, Debian, MX Linux, Ubuntu oder Kubuntu), die den Nutzer mithilfe von grafischen Benutzeroberflächen durch die Installation leiten und eine Vielzahl von kostenlosen Programmen bereitstellen. Trotz des Sicherheitsvorsprungs gegenüber dem am weitesten verbreiteten Betriebssystem Windows wird Linux auf Desktoprechnern immer noch nur zögerlich eingesetzt. Dagegen konnte sich Linux auf dem Internet-Servermarkt relativ schnell etablieren und der Marktanteil in diesem Bereich stieg stetig. So wurden z.B. im Jahr 2017 mindestens 34 % aller Websites mittels eines Linux-Servers verfügbar gemacht (tatsächlicher Marktanteil wohl etwa 60 %).

Im Folgenden werden wir die unterschiedlichen Varianten einer Installation von Linux kurz erläutern. Danach beschreiben wir den SSH-Login auf die Rechner des ITP und stellen einige wichtige Linux-Shell-Befehle vor.

Installation von Linux

Man kann mittlerweile Linux relativ einfach auf seinem eigenen Computer installieren (siehe Übungsaufgabe 1.1). Man geht hierbei gewöhnlich in den folgenden Schritten vor:

Bitte lesen Sie, bevor Sie die Installation von Linux auf Ihrem Rechner beginnen, auch noch weitere Internetquellen die die Installation beschreiben (z.B. [Link 1](#), [Link 2](#), [Link 3](#)).

1. Datensicherung

Wenn sich auf dem Computer schon ein Betriebssystem (wie beispielsweise Windows, Mac OS X oder ein anderes Linux-System) oder wichtige Daten befinden, sollte vor einer Installation unbedingt eine System- bzw. eine Datensicherung erstellt werden. Wie bei jeder Veränderung am Computer kann immer auch etwas schiefgehen.

2. Herunterladen der Installationsdateien (ISO-Image) einer Linux-Distribution aus dem Internet

Zunächst müssen Sie sich für eine [Linux-Distribution](#) entscheiden und das ISO-Image dieser aus dem Internet herunterladen. Falls Sie hier keine Vorlieben haben, empfehle ich z.B. die Downloadseiten [Ubuntu 20.04.4 LTS](#) und [Kubuntu 20.04.4 LTS](#), vielleicht auch [MX Linux](#)

3. Bootfähigen USB-Stick erstellen

Das heruntergeladene ISO-Image der Linux-Distribution muss nun auf einen bootfähigen USB-Stick übertragen werden (ein einfaches Kopieren auf den USB-Stick funktioniert leider nicht). Damit eine Linux Distribution von einem USB-Stick startet, muss zum einen der Computer von USB-Sticks booten können und zum anderen muss der USB-Stick bootbar sein. Letzteres erledigt z.B. der LinuxLive USB Creator [Linux Live USB Creator](#), bzw. [LinuxLive USB Creator](#). Die Software überträgt das Image der Linux-Distribution so auf den USB-Stick, dass er bootfähig ist und der Computer das Linux-System startet, das sich so portable auch an anderen Computern nutzen lässt.

4. Einstecken des USB-Sticks in den Laptop/Desktop und vom USB-Stick Booten

Fahren Sie dann Ihren Computer herunter, stecken sie danach den USB-Stick in Ihren Laptop/Desktop und starten den Computer erneut. Zuvor müssen Sie unter Umständen noch im Setup Ihres Computers die Boot-Reihenfolge ändern, sodass Sie vom USB-Sticks booten können. Dann müssen Sie kurz nach dem Start ihres Computers eine Funktionstaste drücken, wonach der Computer eine Liste mit Laufwerken anzeigt, von denen das des USB-Sticks für den Bootvorgang auszuwählen ist. Welche Funktionstaste zu drücken ist, hängt vom Computerhersteller ab; oft ist es die F12- oder Esc-Taste.

5. Die Linux-Installation startet

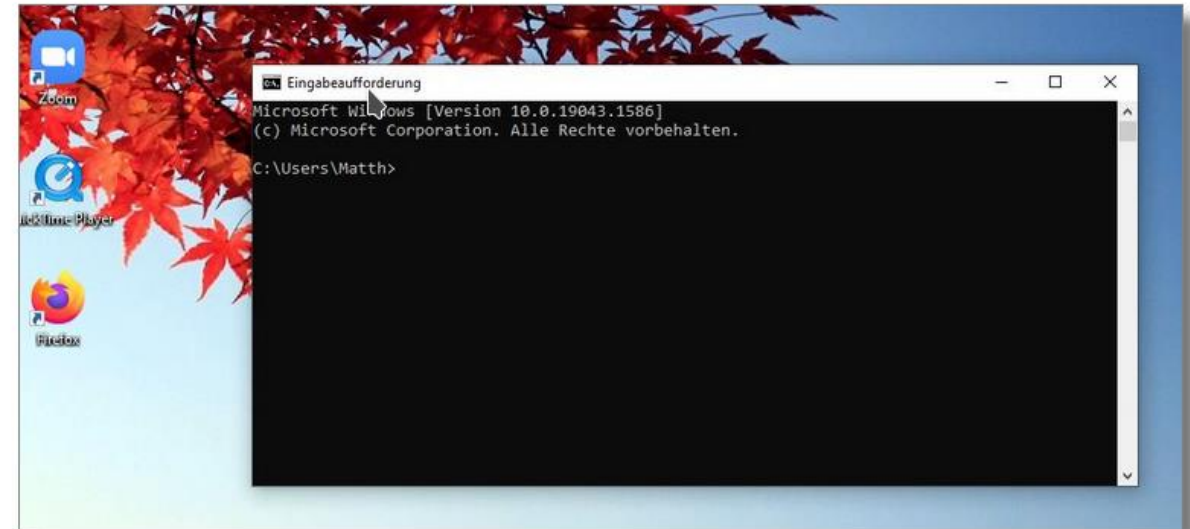
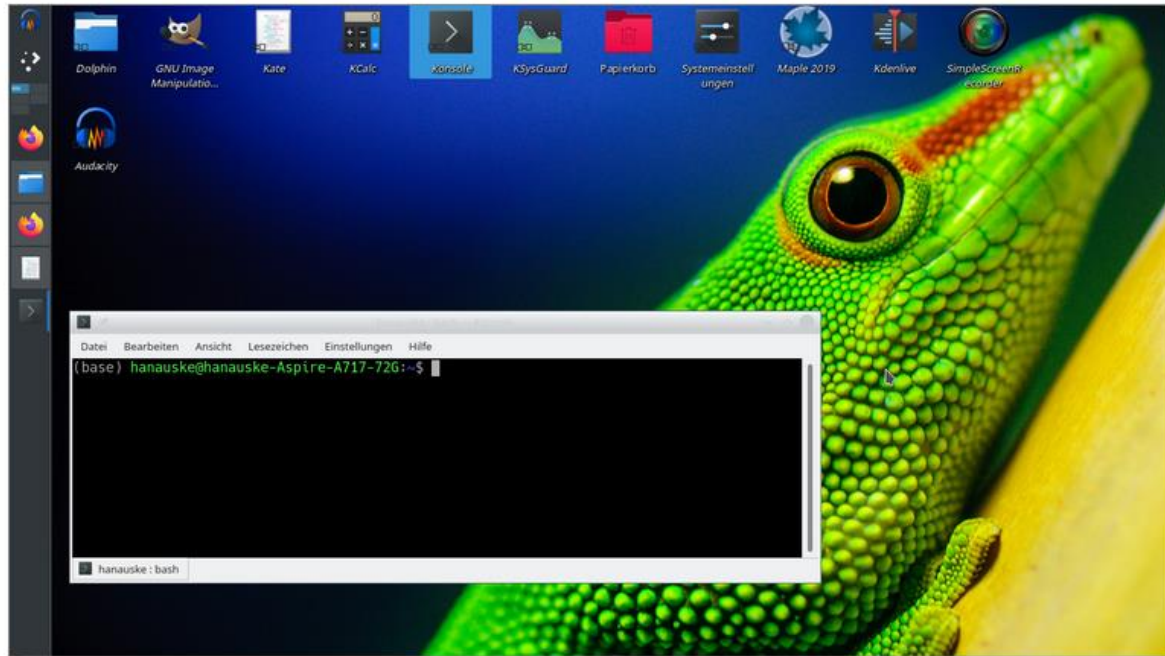
Die grafischen Benutzeroberflächen der Linux-Distribution leitet Sie nun durch die Installation. Jetzt müssen Sie entscheiden, ob Sie ihre gesamte Festplatte für Linux nutzen möchten (in diesem Fall gehen alle Daten die sich auf Ihrem Laptop/Desktop befinden unwiderruflich verloren und werden gelöscht), oder ob Sie Linux neben Ihrem sonstigen Betriebssystem laufen lassen möchten. Falls Sie noch unsicher sind und das Risiko eines Verlustes ihrer Daten nicht in Kauf nehmen wollen, können Sie auch Linux erst einmal ausprobieren und den Linux Live USB Modus starten. In diesem Fall wird Ihr Computer nicht verändert und Linux befindet sich nur auf dem USB-Stick.

Falls Sie an einer Stelle während der Installation Probleme haben und z.B. nicht wissen, ob Sie die vorgeschlagenen Veränderung an Ihrem Computersystem durchführen wollen, dann brechen Sie lieber die Installation ab und fragen einen Ihrer Kommilitonen nach Rat.

Sollten Sie beim Installieren oder bei der Nutzung auf Probleme stoßen, können Sie sich auch jederzeit an den Tutor Ihrer Übungsgruppe wenden bzw. zu den wöchentlichen offenen Fragestunden (immer dienstags nach der Vorlesung) kommen.

Login auf die Rechner des Instituts für Theoretische Physik (ITP) der Goethe-Universität

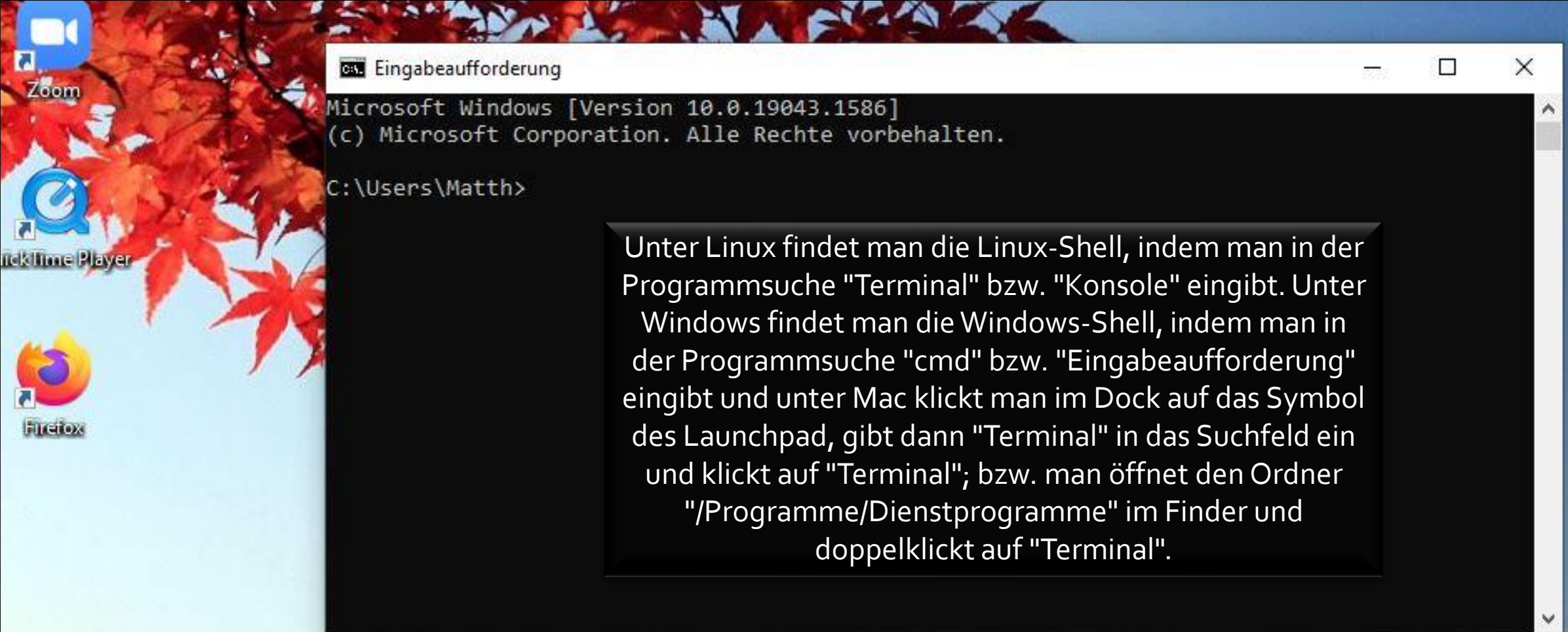
Auch wenn Sie den vorigen Unterpunkt noch nicht erledigt haben, bzw. Linux nicht auf Ihrem eigenen Rechner installieren möchten, können Sie dennoch dieses Teilkapitel durchgehen. Vorausgesetzt ist hierbei jedoch, dass Sie einen Login-Account für die Rechner des Instituts für Theoretische Physik (ITP) der Goethe-Universität haben. Die Login-Accounts werden in den ersten beiden Vorlesungen an die Studierenden verteilt. Bitte lesen Sie die Allgemeine Nutzungsordnung für die Informationsverarbeitungs- und Kommunikations-Infrastruktur der Johann Wolfgang Goethe-Universität, bevor Sie den Login-Account das erste Mal verwenden. Der einfachste Weg einen Login auf die Rechner des ITP aufzubauen ist eine SSH-Verbindung herzustellen. Secure Shell oder SSH bezeichnet ein kryptographisches Netzwerkprotokoll welches eine sichere, authentifizierte und verschlüsselte Verbindung zwischen zwei Rechnern über ein unsicheres Netzwerk aufbaut. Um eine solche Verbindung herzustellen, müssen Sie zunächst eine Shell (Terminal, Konsole) öffnen, die eine text- bzw. kommandobasierte Benutzung Ihres Computersystems ermöglicht. Ein solches Terminal kann man unter allen gängigen Betriebssystemen öffnen und die nachfolgenden Abbildungen zeigen eine solche Shell im Betriebssystem Linux (linke Abbildung) und Windows (rechte Abbildung).



Unter Linux findet man die Linux-Shell, indem man in der Programmsuche "Terminal" bzw. "Konsole" eingibt. Unter Windows findet man die Windows-Shell, indem man in der Programmsuche "cmd" bzw. "Eingabeaufforderung" eingibt und unter Mac klickt man im Dock auf das Symbol des Launchpad, gibt dann "Terminal" in das Suchfeld ein und klickt auf "Terminal"; bzw. man öffnet den Ordner "/Programme

Login auf die Rechner des Instituts für Theoretische Physik (ITP) der Goethe-Universität

Um eine solche Verbindung herzustellen, müssen Sie zunächst eine Shell (Terminal, Konsole) öffnen, die eine text- bzw. kommandobasierte Benutzung Ihres Computersystems ermöglicht. Ein solches Terminal kann man unter allen gängigen Betriebssystemen öffnen und die nachfolgende Abbildung zeigt eine solche Shell im Betriebssystem Windows.



Unter Linux findet man die Linux-Shell, indem man in der Programmsuche "Terminal" bzw. "Konsole" eingibt. Unter Windows findet man die Windows-Shell, indem man in der Programmsuche "cmd" bzw. "Eingabeaufforderung" eingibt und unter Mac klickt man im Dock auf das Symbol des Launchpad, gibt dann "Terminal" in das Suchfeld ein und klickt auf "Terminal"; bzw. man öffnet den Ordner "/Programme/Dienstprogramme" im Finder und doppelklickt auf "Terminal".

Login auf die Rechner des Instituts für Theoretische Physik (ITP) der Goethe-Universität

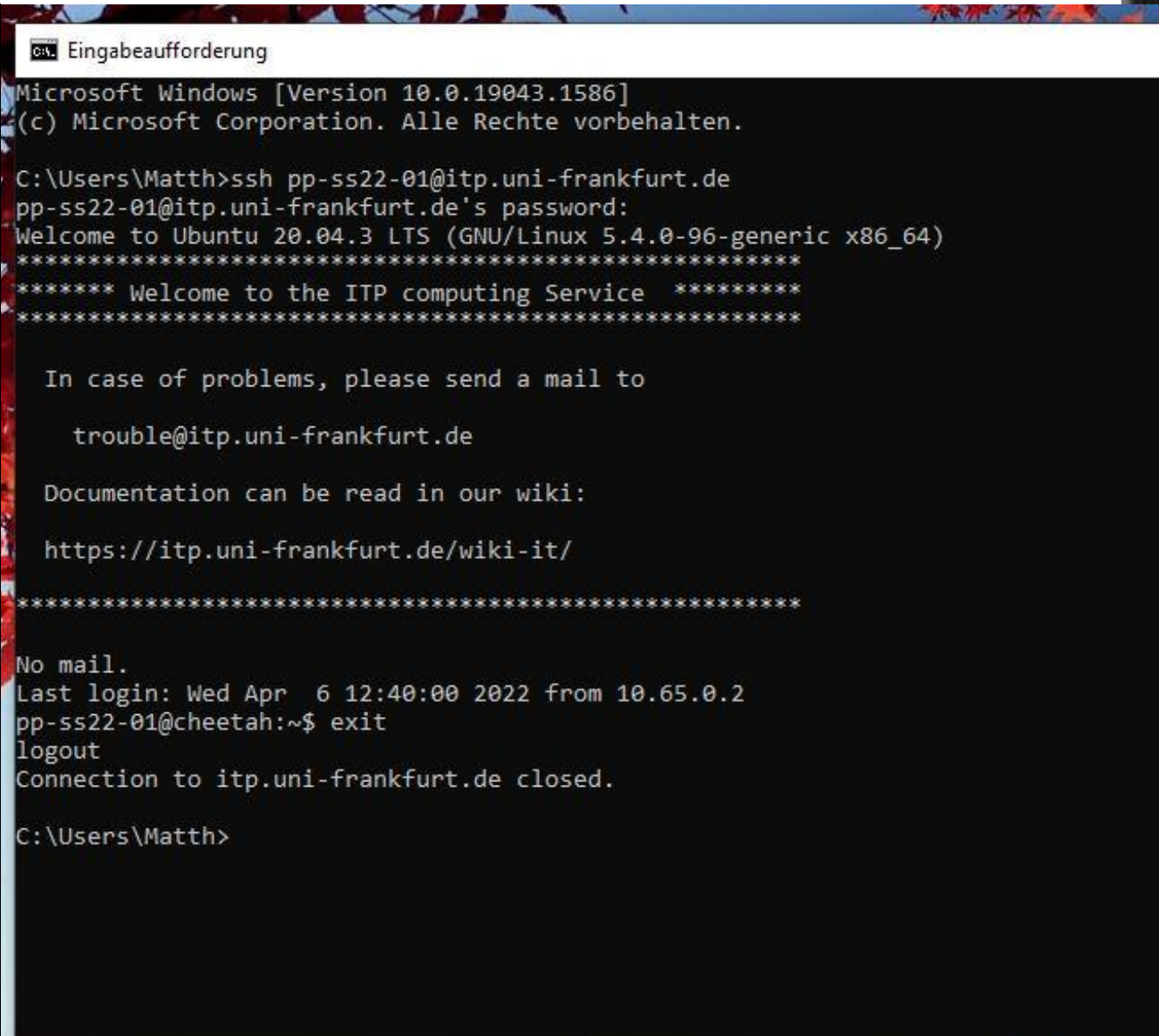
Die Verbindung zu den Rechnern des ITP baut man nun mit dem folgenden Befehl auf:

```
ssh pp-ss22-...@itp.uni-frankfurt.de
```

, wobei "pp-ss22-..." Ihr persönlicher Login-Account ist. Die rechte Abbildungen zeigt den Aufbau einer solchen SSH-Verbindung im Betriebssystem Windows, wobei hier der Login-Account "pp-ss22-01" verwendet wurde.

Nachdem man den Befehl in die Konsole eingegeben und auf "Return" gedrückt hat, muss man sein Passwort eintippen und wieder "Return" drücken. Hat das alles funktioniert, dann ist man mittels des Terminals auf den Rechnern des ITP und kann dort das Betriebssystem Linux ausprobieren.

In der rechten Abbildung der Windows-Shell loggen wir uns danach wieder aus, indem wir den Linux-Befehl "exit" in die Konsole eingeben.



```
Eingabeaufforderung
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Matth>ssh pp-ss22-01@itp.uni-frankfurt.de
pp-ss22-01@itp.uni-frankfurt.de's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-96-generic x86_64)
*****
***** Welcome to the ITP computing Service *****
*****

In case of problems, please send a mail to

    trouble@itp.uni-frankfurt.de

Documentation can be read in our wiki:

    https://itp.uni-frankfurt.de/wiki-it/

*****

No mail.
Last login: Wed Apr  6 12:40:00 2022 from 10.65.0.2
pp-ss22-01@cheetah:~$ exit
logout
Connection to itp.uni-frankfurt.de closed.

C:\Users\Matth>
```

Linux Shell-Befehle

Wir wollen nun das Betriebssystem Linux besser kennenlernen und stellen deshalb einige wichtige Linux Shell-Befehle vor. Eine umfangreiche Befehlsübersicht finden Sie unter [Linux-Shell-Befehle](#).

```
https://itp.uni-frankfurt.de/wiki-it/

*****
No mail.
Last login: Fri Mar 25 18:05:37 2022 from 10.65.0.2
pp-ss22-01@cheetah:~$ passwd
Current Password:
New password:
Retype new password:
passwd: password updated successfully
pp-ss22-01@cheetah:~$ exit
logout
Connection to itp.uni-frankfurt.de closed.
(base) hanauske@hanauske-Aspire-A717-72G:~$
```

```
Last login: Wed Apr 6 11:04:29 2022 from 10.65.0.2
pp-ss22-01@cheetah:~$ ls
lost+found Maildir public_html
pp-ss22-01@cheetah:~$ ls -la
total 68
drwxr-x--- 10 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:06 .
drwxr-xr-x 30 root root 0 Apr 6 11:12 ..
drwxr-xr-x 4 pp-ss22-01 pp-ss22-01 4096 Jan 25 2021 .ancillary
-rw----- 1 pp-ss22-01 pp-ss22-01 20 Apr 6 11:03 .bash_history
-rw-r--r-- 1 pp-ss22-01 pp-ss22-01 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 pp-ss22-01 pp-ss22-01 2327 Jan 25 2021 .bashrc
drwx----- 4 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:05 .cache
drwx----- 2 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:05 .config
drwxr-xr-x 3 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:05 .local
drwx----- 2 pp-ss22-01 pp-ss22-01 16384 Mar 25 10:16 lost+found
lrwxrwxrwx 1 pp-ss22-01 pp-ss22-01 29 Mar 25 10:16 Maildir -> ../nocache/pp-ss22-01/Maildir
-rw-r--r-- 1 pp-ss22-01 pp-ss22-01 1847 Mar 27 2020 .mkshrc
-rw-r--r-- 1 pp-ss22-01 pp-ss22-01 807 Feb 25 2020 .profile
drwxr-x--- 3 pp-ss22-01 pp-ss22-01 4096 Jan 25 2021 public_html
lrwxrwxrwx 1 pp-ss22-01 pp-ss22-01 35 Mar 25 10:16 .spamassassin -> ../nocache/pp-ss22-01/.spamassassin
in
drwx----- 4 pp-ss22-01 pp-ss22-01 4096 Jan 25 2021 .squirrelmail
pp-ss22-01@cheetah:~$
```

Als Erstes ist es dringend empfohlen, das Passwort des eigenen ITP-Accounts zu ändern. Dies wird mittels des Linux-Benutzerverwaltungsbefehls **passwd** gemacht (siehe linke Abbildung). Nachdem man zunächst das alte und danach zweimal das neue Passwort eingegeben hat, ist das Passwort unwiderruflich abgeändert. Man kann dies überprüfen, indem man sich gleich danach mit "exit" ausloggt und dann wieder eine neue SSH-Verbindung aufbaut.

Eine detailliertere, umfangreiche Befehlsübersicht finden Sie z.B. unter [Linux-Shell-Befehle](#)
<http://wiki.ubuntuusers.de/Shell/Befehlsübersicht/>

Passwort ändern

```
https://itp.uni-frankfurt.de/wiki-it/
```

Ihr Login-Name auf dem ITP-Rechner

Computernamen des ITP-Rechners

Sie befinden sich in Ihrem Heimverzeichnis

Eingegebener Linux-Befehl

```
No mail.  
Last login: Fri Mar 25 18:05:37 2022 from 10.65.0.2  
pp-ss22-01@cheetah:~$ passwd  
Current Password:  
New password:  
Retype new password:  
passwd: password updated successfully  
pp-ss22-01@cheetah:~$ exit  
logout  
Connection to itp.uni-frankfurt.de closed.  
(base) hanauske@hanauske-Aspire-A717-72G:~$
```

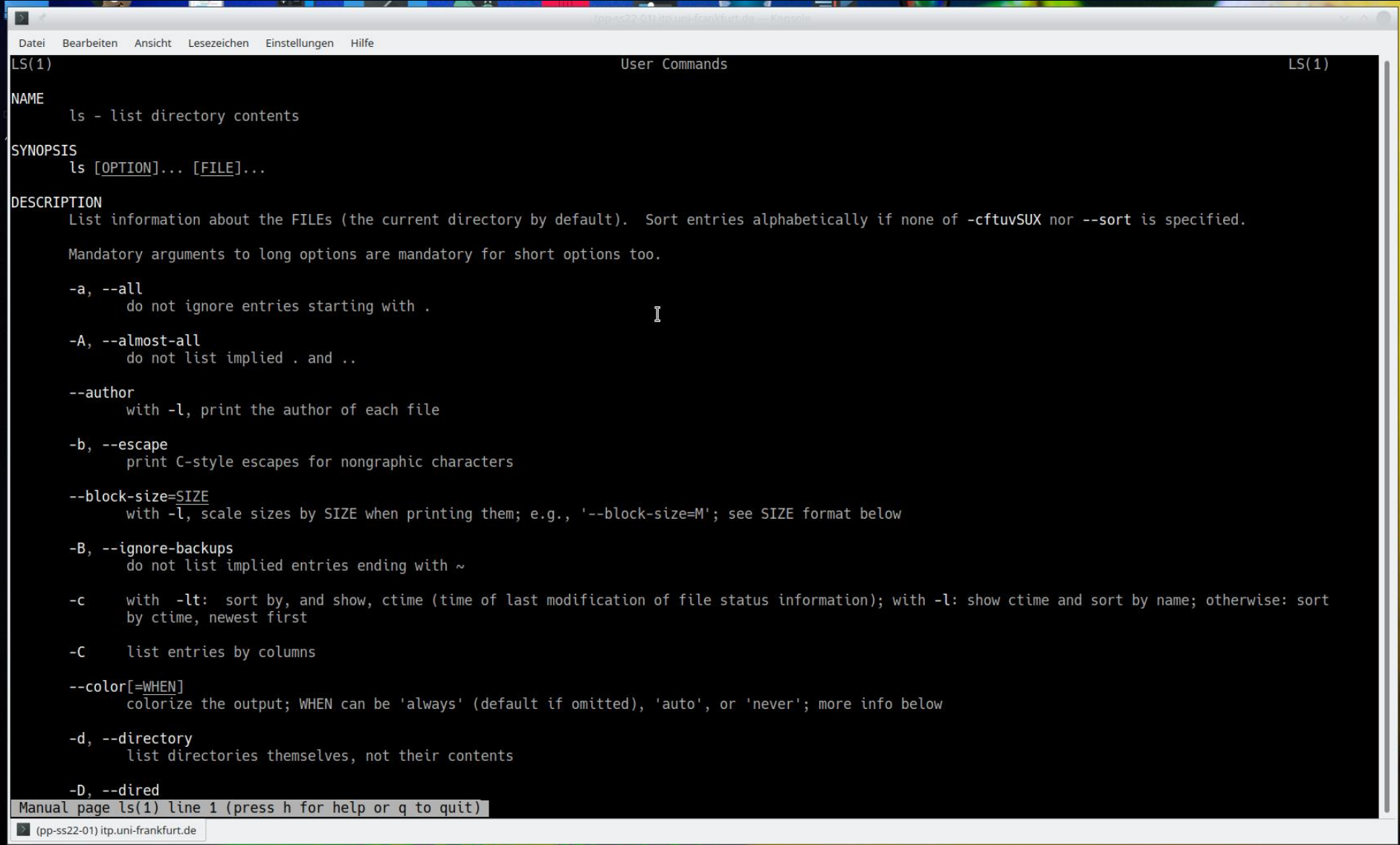
Der Befehl ls

Liste der die im aktuellen Verzeichnis befindlichen Datei und Ordner

```
Last login: Wed Apr  6 11:04:29 2022 from 10.65.0.2
pp-ss22-01@cheetah:~$ ls
lost+found Maildir public_html
pp-ss22-01@cheetah:~$ ls -la
total 68
drwxr-x--- 10 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:06 .
drwxr-xr-x 30 root      root      0 Apr  6 11:12 ..
drwxr-xr-x  4 pp-ss22-01 pp-ss22-01 4096 Jan 25  2021 .ancillary
-rw-----  1 pp-ss22-01 pp-ss22-01   20 Apr  6 11:03 .bash_history
-rw-r--r--  1 pp-ss22-01 pp-ss22-01  220 Feb 25  2020 .bash_logout
-rw-r--r--  1 pp-ss22-01 pp-ss22-01 2327 Jan 25  2021 .bashrc
drwx-----  4 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:05 .cache
drwx-----  4 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:05 .config
drwx-----  2 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:05 .gconf
drwxr-xr-x  3 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:05 .local
drwx-----  2 pp-ss22-01 pp-ss22-01 16384 Mar 25 10:16 lost+found
lrwxrwxrwx  1 pp-ss22-01 pp-ss22-01   29 Mar 25 10:16 Maildir -> ../nocache/pp-ss22-01/Maildir
-rw-r--r--  1 pp-ss22-01 pp-ss22-01 1847 Mar 27  2020 .mkshrc
-rw-r--r--  1 pp-ss22-01 pp-ss22-01  807 Feb 25  2020 .profile
drwxr-x---  3 pp-ss22-01 pp-ss22-01 4096 Jan 25  2021 public_html
lrwxrwxrwx  1 pp-ss22-01 pp-ss22-01   35 Mar 25 10:16 .spamassassin -> ../nocache/pp-ss22-01/.spamassassin
drwx-----  4 pp-ss22-01 pp-ss22-01 4096 Jan 25  2021 .squirrelmail
pp-ss22-01@cheetah:~$
```

Der Befehl `man`

Möchte man z.B. näheres zum Befehl "ls" erfahren, gibt man `man ls` in das Terminal ein. Es erscheint dann das folgende Bild im Terminal:



```
LS(1) User Commands LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      with -l, print the author of each file

  -b, --escape
      print C-style escapes for nongraphic characters

  --block-size=SIZE
      with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

  -B, --ignore-backups
      do not list implied entries ending with ~

  -c
      with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first

  -C
      list entries by columns

  --color[=WHEN]
      colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

  -d, --directory
      list directories themselves, not their contents

  -D, --dired

Manual page ls(1) line 1 (press h for help or q to quit)
(pp-ss22-01) itp.uni-frankfurt.de
```

Man kann diese Beschreibung des Befehls "ls" wieder verlassen, indem man "q" im Terminal eingibt.

Der Befehl cd

Innerhalb der Ordnerstruktur navigieren

```
pp-ss22-01@cheetah:~$ cd public_html/  
pp-ss22-01@cheetah:~/public_html$ ls -la  
total 12  
drwxr-x---  3 pp-ss22-01 pp-ss22-01 4096 Jan 25  2021 .  
drwxr-x--- 10 pp-ss22-01 pp-ss22-01 4096 Mar 25 18:06 ..  
drwxr-xr-x  2 pp-ss22-01 pp-ss22-01 4096 Jan 25  2021 cgi-bin  
lrwxrwxrwx  1 pp-ss22-01 pp-ss22-01   23 Jan 25  2021 webmail -> /usr/share/squirrelmail  
pp-ss22-01@cheetah:~/public_html$ ls  
cgi-bin  webmail  
pp-ss22-01@cheetah:~/public_html$ cd ..  
pp-ss22-01@cheetah:~$ ls  
lost+found  Maildir  public_html  
pp-ss22-01@cheetah:~$ █
```

Der Befehl mkdir

Neue Verzeichnisse erstellen

```
pp-ss22-01@cheetah:~$ ls  
lost+found  Maildir  public_html  
pp-ss22-01@cheetah:~$ mkdir VPROG  
pp-ss22-01@cheetah:~$ ls  
lost+found  Maildir  public_html  VPROG  
pp-ss22-01@cheetah:~$ cd VPROG/  
pp-ss22-01@cheetah:~/VPROG$ ls -la  
total 8  
drwxrwxr-x  2 pp-ss22-01 pp-ss22-01 4096 Apr  6 11:30 .  
drwxr-x--- 11 pp-ss22-01 pp-ss22-01 4096 Apr  6 11:30 ..  
pp-ss22-01@cheetah:~/VPROG$ ls  
pp-ss22-01@cheetah:~/VPROG$ █
```

Weitere wichtige Linux-Befehle

Weitere wichtige Befehle sind:

cp: Kopieren von Dateien oder Verzeichnissen. **cp a.txt ./VPROG/** kopiert z.B. die Datei "a.txt" in das Verzeichnis "VPROG" (mehrere Dateien können mit **cp -r** kopiert werden).

mv: Verschieben von Dateien oder Verzeichnissen. **mv a.txt ./VPROG/** verschiebt z.B. die Datei "a.txt" in das Verzeichnis "VPROG". Der Unterschied zum Befehl "cp" ist, dass die ursprüngliche Datei an ihrem Ort gelöscht wird.

rm: Löschen von Dateien oder Verzeichnissen. **rm a.txt** löscht z.B. die Datei "a.txt". Der Unterschied zum Befehl "cp" ist, dass die ursprüngliche Datei an ihrem Ort gelöscht wird. Die Datei landet hierbei nicht im "Papierkorb", sondern wird unwiderruflich gelöscht.

chmod: Zugriffsrechte von Dateien oder Verzeichnissen ändern. Wir hatten bei dem Listbefehl "ls -la" die Zugriffsrechte schon erwähnt, jedoch ihren Inhalt noch nicht besprochen. Jede Datei und jedes Verzeichnis verfügt über ihre eigenen Zugriffsrechte (erste 10 Zeichen beim "ls -la"-Befehl). Das erste Zeichen gibt an, ob es sich um eine reguläre Datei ("-"), ein Verzeichnis ("d") oder einen Link ("l") handelt. Die Zeichen 2 bis 4 spezifizieren die eigenen Zugriffsrechte (user) und es kann hierbei ein "r" für lesbar (read), ein "w" für überschreibbar (write) oder ein "x" für ausführbar (execute) stehen. Sind Zugriffsrechte nicht gesetzt, erscheint ein Minus "-". Die Zeichen 5 bis 7 spezifizieren in gleicher Weise die Zugriffsrechte der Mitglieder der eigenen Gruppe (group) und die Zeichen 8 bis 10 die Zugriffsrechte aller anderen Benutzer, die Zugang zum Dateisystem haben (others). Nehmen wir z.B. eine Datei, z.B. "a.txt", welche die folgenden Zugriffsrechte besitzt "-rw-rw-r--": Dies bedeutet, dass es sich um eine reguläre Datei handelt (erstes Minus), dass der eigene User sie lesen und überschreiben kann, sie jedoch nicht ausführen kann (es sich somit um kein ausführbares Programm handelt). Die gleichen Zugriffsrechte besitzen auch die Mitglieder der Gruppe, aber alle anderen Benutzer dürfen die Datei nur lesen. Mit dem Linux-Befehl **chmod** kann man nun diese Zugriffsrechte ändern. Möchte man z.B. die Datei "a.txt" auch für alle anderen Benutzer überschreibbar machen, schreibt man **chmod o+w a.txt** oder möchte man die Schreibrechte für Gruppenmitglieder nicht zulassen, gibt man **chmod u-w a.txt** ein.

ps: Zeigt die laufenden Prozesse an. **ps -ux** zeigt z.B. die Liste der laufenden Prozesse und viele Eigenschaften dieser an.

kill: Beendet laufende Prozesse. **kill 3633** beendet z.B. den Prozess mit der Prozess-ID 3633 beenden (hilfreich z.B., wenn ein Programm "abstürzt" ist und das zugeordnete Fenster nicht mehr reagiert).

Eine detailliertere, umfangreiche Befehlsübersicht finden Sie z.B. unter [Linux-Shell-Befehle](#).

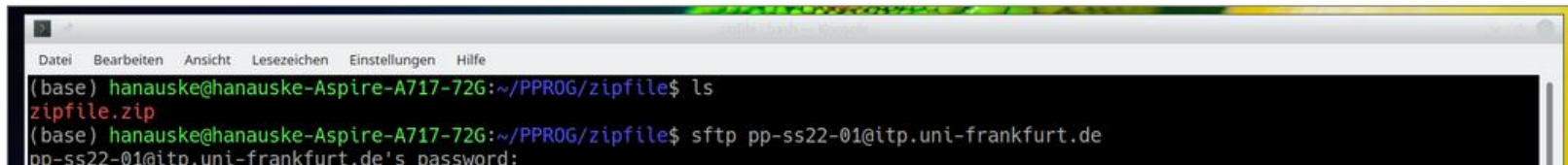
Eine detailliertere, umfangreiche Befehlsübersicht finden Sie z.B. unter [Linux-Shell-Befehle](#)
<http://wiki.ubuntuusers.de/Shell/Befehlsübersicht/>

Anwendungsbeispiel: Erstellen einer eigenen Internet-Homepage

Wie schon im vorigen Unterpunkt Eine kleine Einführung in Linux) besprochen, wird das Linux Betriebssystem auch oft als Internet-Server betrieben. In diesem Unterpunkt werden wir auf den Themenbereich der Linux Internet-Server eingehen und eine eigene kleine Homepage erstellen. Mit der Bereitstellung Ihres eigenen Login-Accounts für die Rechner des Instituts für Theoretische Physik der Goethe-Universität haben Sie auch die Möglichkeit erhalten, eine eigene Homepage zu erstellen und Jederman zur Verfügung zu stellen (bitte lesen Sie hierfür auch die Allgemeine Nutzungsordnung für die Informationsverarbeitungs- und Kommunikations-Infrastruktur der Johann Wolfgang Goethe-Universität). Wir werden einige der besprochenen Linux-Shell-Befehle anwenden und um die Erstellung einer eigenen Internet-Homepage zu realisieren. Zunächst werden wir jedoch einige Dateien, die wir für die Homepage und später für den Unterpunkt Programmiersprachen benötigen, mittels des *Secure File Transfer Protocol (SFTP)* auf den Rechner des ITP übertragen.

Das Secure File Transfer Protocol (SFTP)

Das Secure File Transfer Protocol (SFTP) ermöglicht es, Daten in verschlüsselter Form von einem Rechner auf einen anderen Rechner zu übertragen. Wir hatten im Unterpunkt "Login auf die Rechner des ITP" bereits die verschlüsselte Verbindung zwischen zwei Rechnern über ein unsicheres Netzwerk mittels einer SSH-Verbindung kennengelernt. Mittels einer SFTP-Verbindung kann man in ähnliche Weise eine Verbindung zu einem anderen Rechner aufbauen, hierbei ist es jedoch auch möglich Dateien von einem Rechner auf den anderen zu übertragen. Alternativ gibt es auch noch den Befehl **scp**, der hier jedoch nicht besprochen werden soll.




```
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile$ ls
zipfile.zip
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile$ sftp pp-ss22-01@itp.uni-frankfurt.de
pp-ss22-01@itp.uni-frankfurt.de's password:
```

Nehmen wir z.B. an, dass Sie mehrere Daten haben, die Sie von Ihrem Heimrechner auf Ihren Account am ITP übertragen möchten. Es ist hierbei oft

Das Secure File Transfer Protocol SFTP

Das Secure File Transfer Protocol (SFTP) ermöglicht es, Dateien und Verzeichnisse in verschlüsselter Form von einem Rechner auf einen anderen Rechner zu übertragen.



```
zipfile : bash — Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile$ ls
zipfile.zip
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile$ sftp pp-ss22-01@itp.uni-frankfurt.de
pp-ss22-01@itp.uni-frankfurt.de's password:
Connected to itp.uni-frankfurt.de.
sftp> ls
Maildir          VPROG          lost+found      public_html
sftp> cd VPROG/
sftp> ls
sftp> put ./zipfile.zip
Uploading ./zipfile.zip to /home/pp-ss22-01/VPROG/zipfile.zip
./zipfile.zip                                     100%  35KB 357.6KB/
sftp> ls
zipfile.zip
sftp> exit
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile$
```

<http://th.physik.uni-frankfurt.de/~hanauske/VPROG/pics/V1/zipfile.zip>

Laden Sie sich bitte hierfür zunächst den zip-File auf Ihren eigenen Computer herunter und speichern diesen in einem Ordner ([Download zip-File](#)). Öffnen Sie dann ein Terminal in diesem Verzeichnis. In oberen Abbildung können Sie in der obersten Zeile sehen, dass ich z.B. den zip-File auf meinem Heimrechner in das Verzeichnis "PPROG/zipfile/" gespeichert habe.

Wieder einloggen mit ssh

```
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile$ ssh pp-ss22-01@itp.uni-frankfurt.de
pp-ss22-01@itp.uni-frankfurt.de's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-96-generic x86_64)
*****
***** Welcome to the ITP computing Service *****
*****

In case of problems, please send a mail to
    trouble@itp.uni-frankfurt.de

Documentation can be read in our wiki:
    https://itp.uni-frankfurt.de/wiki-it/

No mail.
Last login: Wed Apr  6 11:12:07 2022 from 10.65.0.2
pp-ss22-01@cheetah:~$ cd VPROG/
pp-ss22-01@cheetah:~/VPROG$ ls
zipfile.zip
pp-ss22-01@cheetah:~/VPROG$ unzip zipfile.zip
Archive:  zipfile.zip
  creating: C++/
  inflating: C++/HelloWorld.cpp
  creating: HTML/
  inflating: HTML/index.html
  creating: Jupyter_Notebooks/
  inflating: Jupyter_Notebooks/VisDataPython.ipynb
  creating: Python_Skripts/
  inflating: Python_Skripts/.py
  extracting: Python_Skripts/HelloWorld.py
pp-ss22-01@cheetah:~/VPROG$ ls
C++  HTML  Jupyter_Notebooks  Python_Skripts  zipfile.zip
pp-ss22-01@cheetah:~/VPROG$
```

zip-File entpacken mit
unzip

Der Befehl cp

Kopieren von Dateien oder Verzeichnissen

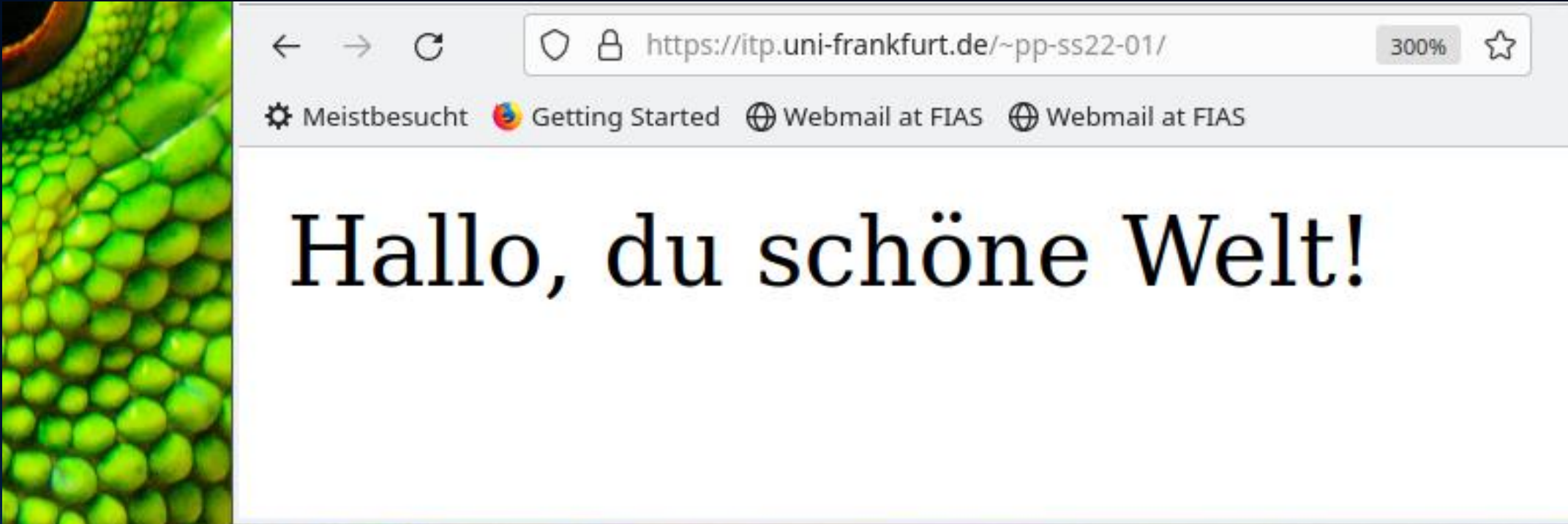
```
pp-ss22-01@cheetah:~/VPROG$ cd HTML/
pp-ss22-01@cheetah:~/VPROG/HTML$ ls
index.html
pp-ss22-01@cheetah:~/VPROG/HTML$ cp index.html ~/public_html/
pp-ss22-01@cheetah:~/VPROG/HTML$ cd ..
pp-ss22-01@cheetah:~/VPROG$ cd ..
pp-ss22-01@cheetah:~$ cd public_html/
pp-ss22-01@cheetah:~/public_html$ ls
cgi-bin  index.html  webmail
pp-ss22-01@cheetah:~/public_html$ ls -la
total 16
drwxr-x---  3 pp-ss22-01 pp-ss22-01 4096 Apr  6 12:13 .
drwxr-x--- 11 pp-ss22-01 pp-ss22-01 4096 Apr  6 11:30 ..
drwxr-xr-x  2 pp-ss22-01 pp-ss22-01 4096 Jan 25  2021 cgi-bin
-rw-rw-r--  1 pp-ss22-01 pp-ss22-01  132 Apr  6 12:13 index.html
lrwxrwxrwx  1 pp-ss22-01 pp-ss22-01   23 Jan 25  2021 webmail -> /usr/share/squirrelmail
pp-ss22-01@cheetah:~/public_html$
```

Die Datei
index.html
in das Verzeichnis
~/public_html/
kopieren



Bitte gehen Sie dann mittels `cd` in das "public_html"-Verzeichnis und überprüfen Sie mit `ls -la` ob die Zugriffsrechte der Datei "index.html" so gesetzt sind, dass sie für alle lesbar ist.

Homepage fertig!



Nun können Sie überprüfen, ob Ihre Homepage auch im Internet für jeden zugänglich ist, indem Sie einen Internet-Browser öffnen und als http-Adresse "`http://itp.uni-frankfurt.de/~pp-ss22-.../`" eingeben (hierbei benutzen Sie bitte die Nummer Ihres Login-Accounts für "`pp-ss22-...`").

Homepage fertig!



https://itp.uni-frankfurt.de/~pp-ss22-01/

300%



Meistbesucht Getting Started Webmail at FIAS Webmail at FIAS

Hallo, du schöne Welt!

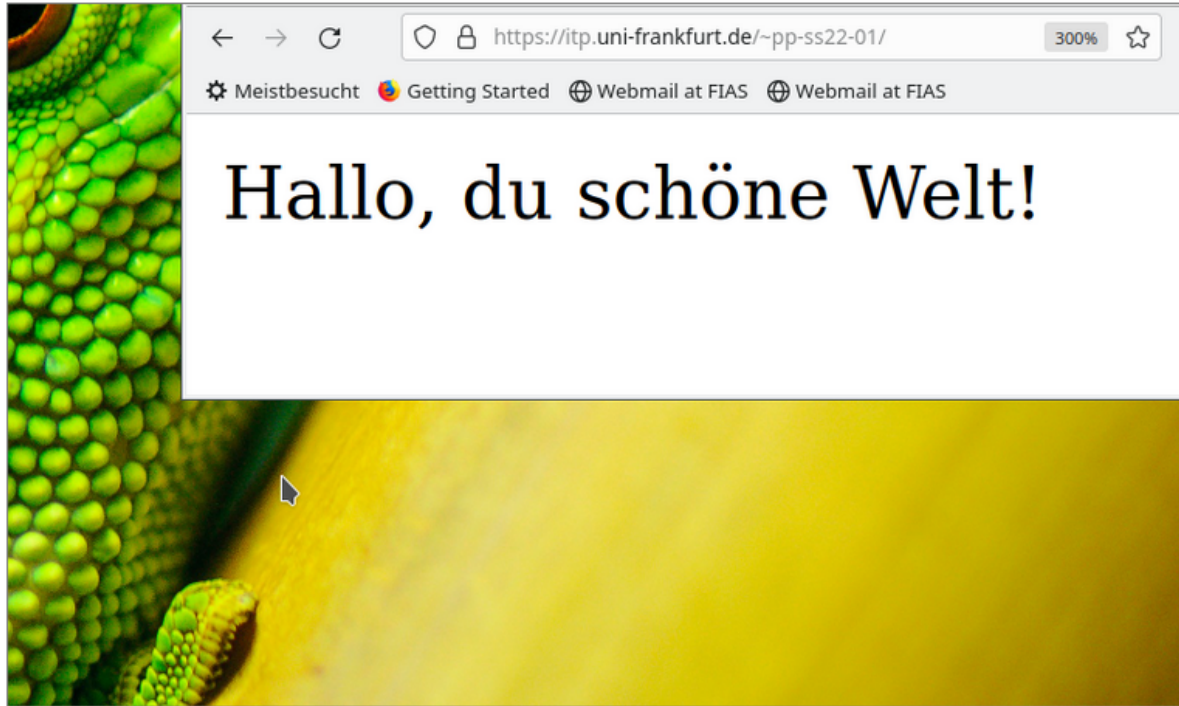
Der HTML Quelltext „index.html“ hat die folgende Struktur

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Homepage von ....</title>
  </head>
  <body>
    Hallo, du sch&ouml;ne Welt!
  </body>
</html>
```

Homepage fertig?

Die linke Abbildung zeigt die Homepage für den Account "pp-ss22-01". Falls die Zugriffsrechte der Datei "index.html" nicht für alle lesbar sind, erhalten Sie das Bild in der rechten unteren Abbildung.



Nun können Sie überprüfen, ob Ihre Homepage auch im Internet für jeden zugänglich ist, indem Sie einen Internet-Browser öffnen und als http-Adresse "http://itp.uni-frankfurt.de/~pp-ss22-.../" eingeben (hierbei benutzen Sie bitte die Nummer Ihres Login-Accounts für "pp-ss22-..."). Die nebenstehende Abbildung auf der linken Seite zeigt die Homepage für den Account "pp-ss22-01". Falls die Zugriffsrechte der Datei "index.html" nicht für alle lesbar sind, erhalten Sie das Bild in der rechten oberen Abbildung.

Gerne können Sie nun die "index.html"-Datei modifizieren und z.B. Ihren Namen angeben, Bilder einbinden (siehe [Einbindung mit img](#)) und Links zu anderen Internetseiten integrieren (siehe [Referenzieren in HTML](#)). Hierzu können Sie entweder einen textbasierten Editor (z.B. [VIM](#)) innerhalb der SSH-Verbindung benutzen und direkt auf dem ITP-Computer die Datei verändern, oder Sie können die Datei zunächst auf ihrem Heimcomputer mit einem komfortableren grafischen Editor bearbeiten (z.B. mit [gedit](#), [Kate](#) oder [Geany](#)) und dann per SFTP auf den ITP-Server übertragen. Weiteres zur Sprache HTML finden Sie z.B. unter [selfhtml Tutorials](#).

Programmiersprachen

Im vorigen Unterkapitel hatten wir die Skriptsprache des Internets ("HTML") kennengelernt. HTML ist jedoch keine Programmiersprache, da man mit ihr keine komplexen Rechenvorschriften formulieren kann, die in Form von Algorithmen dann vom Computer ausgeführt werden können. In diesem Unterkapitel werden wir die Programmiersprachen *C++* und *Python* kennenlernen. Beide Programmiersprachen sind bereits auf den Rechnern des Instituts für Theoretische Physik installiert und Sie können diese mittels Ihres ITP-Accounts und einer SSH-Verbindung nutzen (siehe Unterpunkt "Nutzung von C++ und Python auf den ITP-Rechnern"). Es ist jedoch ratsam sich diese Programmiersprachen auch auf dem eigenen Computer zu installieren und die Installation von *C++* und *Python* ist auf allen gängigen Betriebssystemen möglich und wird am Beispiel des Betriebssystems Linux im Unterpunkt "Installation von C++ und Python unter Linux" kurz beschrieben. Die Programmiersprache *C++* ist eine kompilierte Programmiersprache, d.h. der vom Programmierer geschriebene *Quelltext* des Computerprogramms muss zunächst mittels eines *Compilers* in ein ausführbares Programm umgewandelt werden, damit es genutzt werden kann. *Python* hingegen ist eine Skriptsprache, die den *Quelltext* des Programms zuerst in einen plattformunabhängigen Bytecode übersetzt, welcher dann von einem Interpreter ausgeführt wird.

Installation von C++ und Python unter Linux

Um ein C++ Programm ausführen zu können, benötigt man einen C++ Compiler, der den *Quelltext* des Computerprogramms in ein ausführbares Programm umgewandelt. Um zusätzliche Programme unter Linux zu installieren, benutzt man gewöhnlich den Linux-Befehl **sudo apt-get install ...**, wobei die ... als Platzhalter für das zu installierende Programm stehen. Es gibt eine Vielzahl an C++-Compilern und wir werden in dieser Vorlesung den Compilierungsprozess meist mit dem "Gnu Compiler g++" machen. Für die Installation von diesem und einigen anderen Compilern gibt man den folgenden Befehl in das Linux-Terminal ein: **sudo apt-get install build-essential**. Es handelt sich bei diesem Installationspaket um die Gnu Compiler Collection.

Die Programmiersprache Python ist normalerweise in einer Ubuntu-Standardinstallation bereits enthalten. Falls nicht, kann man *Python 3* einfach mittels des folgenden Befehls installieren: **sudo apt-get install python3** Zusätzlich ist es empfohlen, das Standardverwaltungswerkzeug für Python-Module (pip, Pip Installs Python) zu installieren: **sudo apt-get install python3-pip** Gewöhnlich reicht eine solche Installation um Python Skripte ausführen zu können. Da wir im Laufe der Vorlesung auch sogenannte *Python Jupyter Notebooks* ausführen möchten, ist es zusätzlich nötig Jupyter zu installieren. Mittels **pip** ist eine solche Installation mit folgendem Befehl einfach durchzuführen: **pip install notebook** Weitere, im Laufe der Vorlesung benötigte Python-Module/Bibliotheken kann man auf die gleiche Weise installieren. Die Python Visualisierungsbibliothek matplotlib installiert man z.B. mit **pip install matplotlib**.

Nutzung von C++ und Python auf den ITP-Rechnern

Nutzung von C++ und Python auf den ITP-Rechnern

Wie schon oben erwähnt, sind die Programmiersprachen *C++* und *Python* bereits auf den Rechnern des Instituts für Theoretische Physik installiert und Sie können diese mittels Ihres ITP-Accounts und einer SSH-Verbindung nutzen.

```
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile$ ssh pp-ss22-01@itp.uni-frankfurt.de
pp-ss22-01@itp.uni-frankfurt.de's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-96-generic x86_64)
*****
***** Welcome to the ITP computing Service *****
*****

In case of problems, please send a mail to
    trouble@itp.uni-frankfurt.de

Documentation can be read in our wiki:
    https://itp.uni-frankfurt.de/wiki-it/
```

Sie bauen zunächst eine SSH-Verbindung mit einem ITP-Rechner auf und extrahieren das zip-File, wie im vorigen Kapitel Anwendungsbeispiel: Erstellen einer eigenen Internet-Homepage besprochen. Gehen Sie nun mit **cd ~/VPROG/C++/** in das Verzeichnis "C++" und listen mit **ls** die im Ordner befindlichen Dateien (siehe untere linke Abbildung). Man erkennt, dass sich eine Datei HelloWorld.cpp darin befindet. Es handelt sich hierbei um den Quelltext eines C++ Programms, welches wir in der nächsten Vorlesung im Detail besprechen werden und der Inhalt der Datei ist in der linken unteren Abbildung dargestellt.

```
pp-ss22-01@cheetah:~/VPROG/C++$ ls
HelloWorld.cpp
pp-ss22-01@cheetah:~/VPROG/C++$ g++ HelloWorld.cpp
pp-ss22-01@cheetah:~/VPROG/C++$ ls
a.out HelloWorld.cpp
pp-ss22-01@cheetah:~/VPROG/C++$ ./a.out
Hallo, du schöne Welt!
pp-ss22-01@cheetah:~/VPROG/C++$ ./a.out > textfile.txt
pp-ss22-01@cheetah:~/VPROG/C++$ ls
a.out HelloWorld.cpp textfile.txt
```

```
No mail.
Last login: Wed Apr  6 11:12:07 2022 from 10.65.0.2
pp-ss22-01@cheetah:~$ cd VPROG/
pp-ss22-01@cheetah:~/VPROG$ ls
zipfile.zip
pp-ss22-01@cheetah:~/VPROG$ unzip zipfile.zip
Archive:  zipfile.zip
   creating: C++/
  inflating: C++/HelloWorld.cpp
   creating: HTML/
  inflating: HTML/index.html
   creating: Jupyter_Notebooks/
  inflating: Jupyter_Notebooks/VisDataPython.ipynb
   creating: Python_Skripts/
  inflating: Python_Skripts/.py
  extracting: Python_Skripts/HelloWorld.py
pp-ss22-01@cheetah:~/VPROG$ ls
C++ HTML Jupyter_Notebooks Python_Skripts zipfile.zip
pp-ss22-01@cheetah:~/VPROG$
```

HelloWorld.cpp

```
#include <iostream> // Ein- und Ausgabebibliothek
using namespace std; // Benutze den Namensraum std

int main(){ // Hauptfunktion
    cout << "Hallo, du schöne Welt!" << endl; // Ausgabe eines Textes
}
```

Um zu sehen, was das C++ Programm macht, muss man es zunächst

Kompilierung und Ausführung des C++ Programms

Quelltext des
C++ Programms
„HelloWorld.cpp“

HelloWorld.cpp

```
#include <iostream> // Ein- und Ausgabebibliothek
using namespace std; // Benutze den Namensraum std

int main(){ // Hauptfunktion
    cout << "Hallo, du schöne Welt!" << endl; // Ausgabe eines Textes
}
```

g++ HelloWorld.cpp



```
pp-ss22-01@cheetah:~/VPROG/C++$ ls
HelloWorld.cpp
pp-ss22-01@cheetah:~/VPROG/C++$ g++ HelloWorld.cpp
pp-ss22-01@cheetah:~/VPROG/C++$ ls
a.out HelloWorld.cpp
pp-ss22-01@cheetah:~/VPROG/C++$ ./a.out
Hallo, du schöne Welt!
pp-ss22-01@cheetah:~/VPROG/C++$ ./a.out > textfile.txt
pp-ss22-01@cheetah:~/VPROG/C++$ ls
a.out HelloWorld.cpp textfile.txt
pp-ss22-01@cheetah:~/VPROG/C++$
```

Um zu sehen, was das C++ Programm macht, muss man es zunächst kompilieren; dies funktioniert mit dem Befehl **g++ HelloWorld.cpp** Während des Kompilierung-Prozesses wurde eine neue Datei mit dem Namen "a.out" erzeugt, welche das ausführbare Programm darstellt. Führt man das Programm schließlich mit dem Befehl **./a.out** aus, so erscheint die folgende Ausgabe auf der Shell-Konsole: "Hallo, du schöne Welt!" Man kann sich die Terminalausgabe auch in eine Textdatei (z.B. in "textfile.txt") umleiten lassen, wenn man den folgenden Befehl ausführt: **./a.out > textfile.txt**

Ausführen eines Python Programms

Gehen Sie nun mit `cd ~/VPROG/Python_Skripts/` in das Verzeichnis "Python_Skripts" und listen mit `ls` die im Ordner befindlichen Dateien (siehe untere linke Abbildung). Man erkennt, dass sich eine Datei [HelloWorld.py](#) darin befindet. Es handelt sich hierbei um den Quelltext eines Python Skripts, welches in der linken unteren Abbildung dargestellt ist. Um zu sehen was das Python Skript macht, gibt man den Befehl `python3 HelloWorld.py` ein und es erscheint die folgende Ausgabe auf der Shell-Konsole: "Hallo, du schöne Welt!"

```
pp-ss22-01@cheetah:~/VPROG$ cd Python_Skripts/
pp-ss22-01@cheetah:~/VPROG/Python_Skripts$ ls
HelloWorld.py
pp-ss22-01@cheetah:~/VPROG/Python_Skripts$ python3 HelloWorld.py
Hallo, du schöne Welt!
pp-ss22-01@cheetah:~/VPROG/Python_Skripts$ █
```

(pp-ss22-01) itp.uni-frankfurt.de

HelloWorld.py

```
print("Hallo, du schöne Welt!")
```

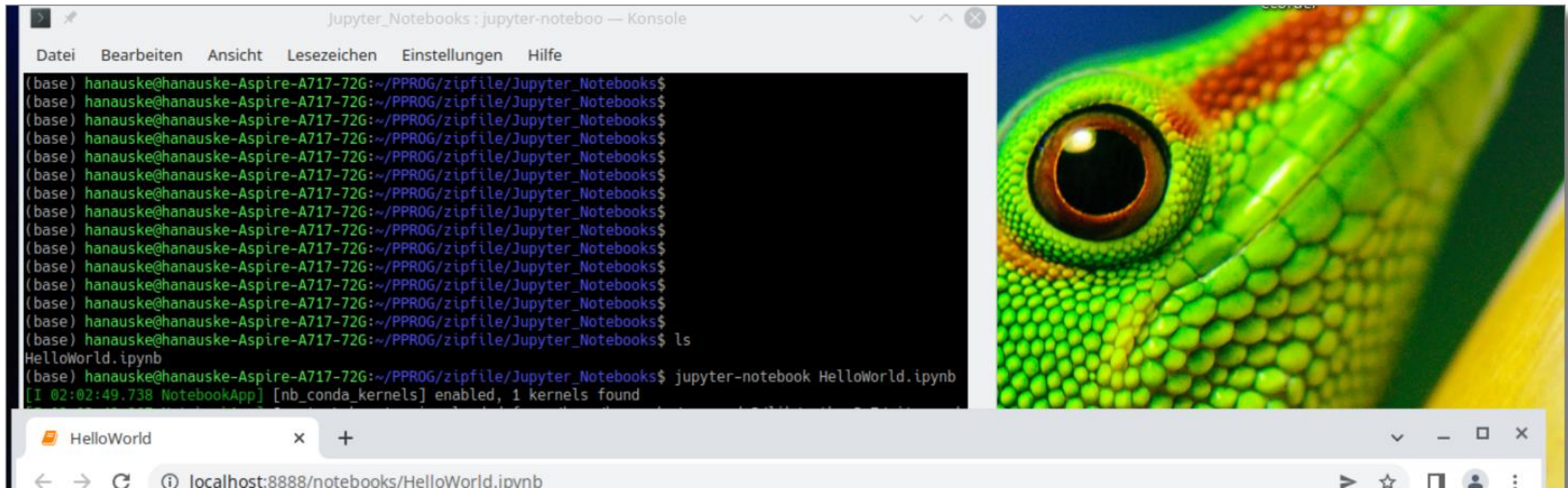
Quelltext des
Python Programms
„HelloWorld.py“

Jupyter Notebooks und Installation von C++ und Python unter dem eigenen Betriebssystem

Da Jupyter Notebooks eine grafische Benutzeroberfläche benötigen, ist es mittels einer normalen SSH-Verbindung nicht möglich diese Remote zu benutzen. Einen grafischen *Remote Login* von dem Heimcomputer zu den ITP-Computern ist jedoch möglich (näheres siehe [Graphical Remote Login with xrdp](#)). Sie können nun entweder einen solchen *Remote Login* aufbauen und die Jupyter-Notebooks mittels des Befehles **jupyter-notebook** starten, oder Sie installieren sich die gesamte Python Umgebung mittels [ANACONDA](#) bzw. [Miniconda](#) auf ihrem Heimrechner (für Windows und MacOS User empfohlen).

Für das Betriebssystem Windows 10 und 11 kann ein Linux-Betriebssystem ohne groß en Aufwand mittels Windows Subsystem for Linux (WSL) hinzugefügt werden. Die Installation wird hier [Windows Subsystem for Linux](#) erläutert. Es wird ein eigenes Dateisystem für Linux erstellt. Standardmäßig wird Ubuntu zur Verfügung gestellt, was sich gut für den Einstieg eignet. Nun kann im Linux-Subsystem die Einrichtung der Entwicklungsumgebung wie bei einer normalen Installation vorgenommen werden. Aus Windows heraus können auf Dateien im Linuxsystem zugegriffen werden, und umgekehrt. Linux- und Windows-Befehlen lassen sich außerdem kombinieren.

Haben Sie bereits Linux als Zweitsystem auf Ihrem eigenen Rechner installiert, dann starten Sie das Jupyter Notebook "HelloWorld" mit dem Befehl **jupyter-notebook HelloWorld.ipynb** In den Tutorien zu dieser Vorlesung wird auf die oben genannten Punkte noch genauer eingegangen. Die folgende Abbildung zeigt das Jupyter Notebook "HelloWorld.ipynb":



```
Jupyter_Notebooks : jupyter-notebook -- Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$ ls
HelloWorld.ipynb
(base) hanauske@hanauske-Aspire-A717-72G:~/PPROG/zipfile/Jupyter_Notebooks$ jupyter-notebook HelloWorld.ipynb
[I 02:02:49.738 NotebookApp] [nb_conda_kernels] enabled, 1 kernels found
```



er

HelloWorld

localhost:8888/notebooks/HelloWorld.ipynb

jupyter HelloWorld Last Checkpoint: vor 22 Minuten (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

Einführung in die Programmierung für Studierende der Physik

(Introduction to Programming for Physicists)

Vorlesung gehalten an der J.W.Goethe-Universität in Frankfurt am Main

(Sommersemester 2022)

von Dr.phil.nat. Dr.rer.pol. Matthias Hanauske

Frankfurt am Main 10.04.2022

```
In [1]: print("Hallo, du schöne Welt!")
Hallo, du schöne Welt!
```

```
In [ ]:
```

Installation von Jupyter

Auf den Rechnern des ITP ist Python und Jupyter schon vorinstalliert und man started ein Jupyter Notebook in einem Linux-Terminal mit dem Befehl „jupyter-notebook“.

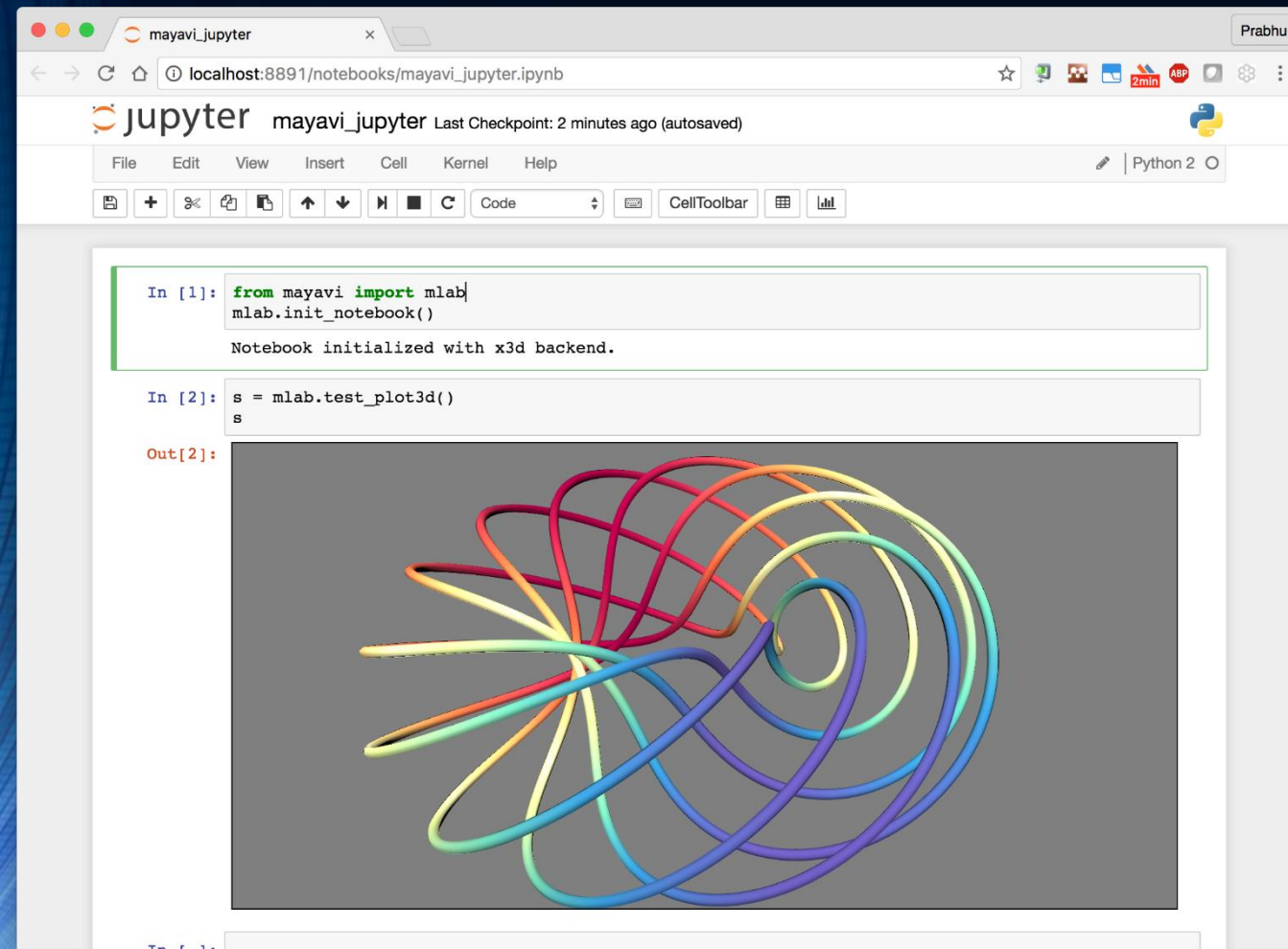
Unter Windows kann man Jupyter recht einfach mittel Anaconda installieren

The image shows two side-by-side screenshots from a Windows desktop. The left screenshot displays the Anaconda Navigator application window. The interface includes a sidebar with 'Home', 'Environments', 'Learning', and 'Community'. The main area shows 'Applications on base (root)' with four options: 'CMD.exe Prompt' (0.1.1), 'JupyterLab' (2.1.5), 'Jupyter Notebook' (6.0.3), and 'Powershell Prompt' (0.0.1). Each option has a 'Launch' button. The right screenshot shows a web browser window displaying a Jupyter Notebook. The notebook title is 'Physik der sozio-ökonomischen Systeme mit dem Computer (Physics of Socio-Economic Systems with the Computer)'. The content includes a title, author information (Dr. phil. nat. Dr. rer. pol. Matthias Hanauske), and an introduction to game theory. The notebook contains several code cells with Python code and their corresponding outputs. The code defines a payoff matrix for a 2-player game and calculates the transpose of the matrix. The output shows the matrix $\begin{bmatrix} -7 & -1 \\ -9 & -3 \end{bmatrix}$ and its transpose $\begin{bmatrix} -7 & -9 \\ -1 & -3 \end{bmatrix}$.

Python Jupyter Notebooks *eine web-basierte interaktive Programmierumgebung für Python*

Jupyter Notebooks

Nutzung ähnlich wie bei
Maple Worksheets oder *Mathematica Notebooks*

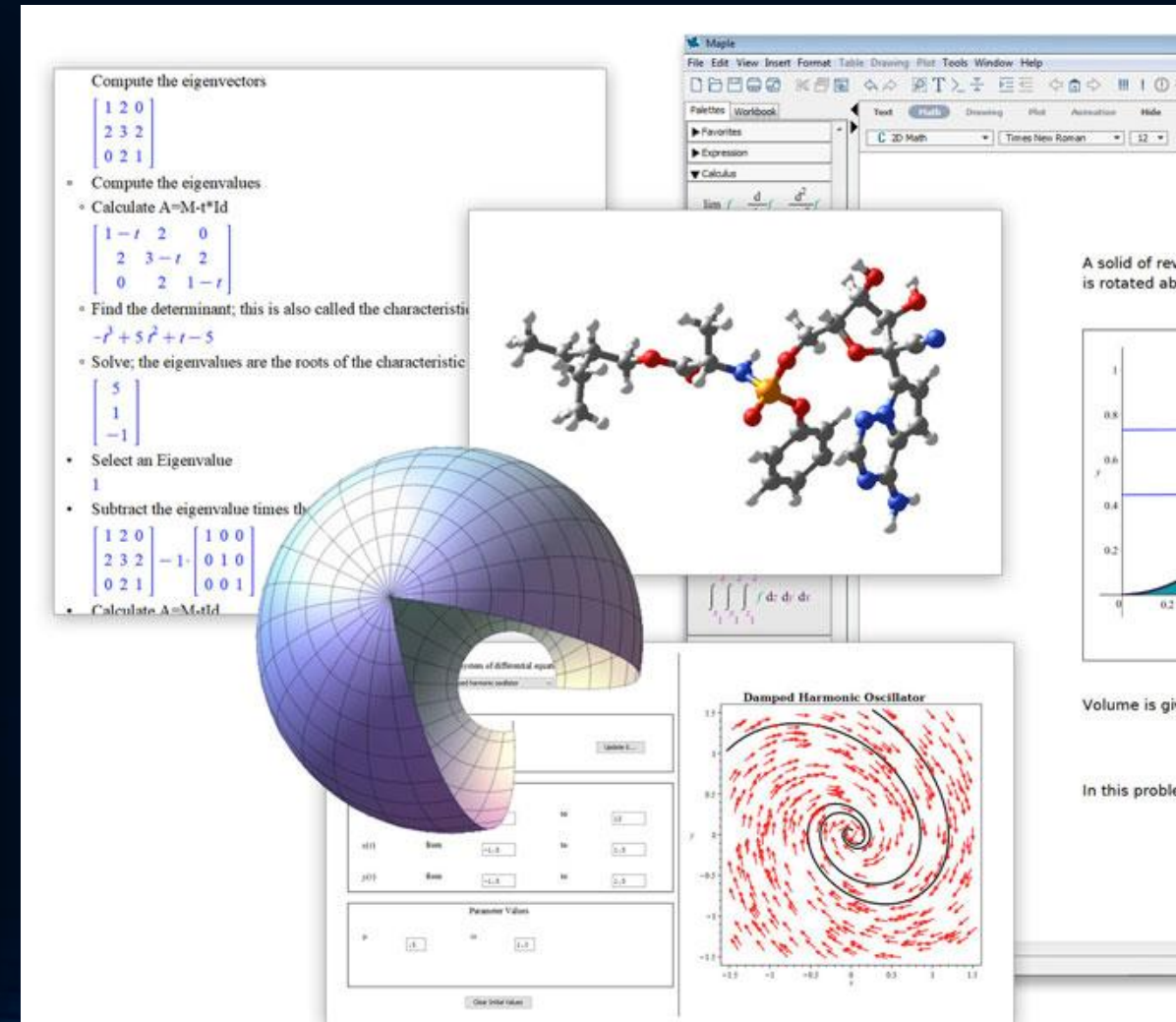


The screenshot shows a web browser window with the URL `localhost:8891/notebooks/mayavi_jupyter.ipynb`. The Jupyter interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar. The code cell contains:

```
In [1]: from mayavi import mlab
mlab.init_notebook()
Notebook initialized with x3d backend.

In [2]: s = mlab.test_plot3d()
s
```

The output cell displays a 3D plot of a complex, multi-colored, intertwined structure, likely a knot or a complex surface, rendered using the Mayavi library.



This collage illustrates the similarity in usage between Jupyter Notebooks and other mathematical software environments. It features several overlapping windows:

- Maple Worksheet:** Shows a list of tasks for computing eigenvectors and eigenvalues of a matrix $A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 3 & 2 \\ 0 & 2 & 1 \end{bmatrix}$. It includes steps like "Compute the eigenvalues", "Calculate $A = M^{-1} * Id$ ", and "Find the determinant".
- Mathematica Notebook:** Displays a 3D molecular model of a complex organic molecule with atoms represented by spheres (carbon in grey, oxygen in red, nitrogen in blue).
- Mathematica Notebook:** Shows a 3D plot of a sphere with a grid, illustrating a solid of revolution.
- Mathematica Notebook:** Displays a phase portrait of a "Damped Harmonic Oscillator" with a spiral trajectory and a vector field of red arrows.

Diverse Jupyter Notebooks auch in der Vorlesung

Allgemeine Relativitätstheorie mit dem Computer

[ART mit dem Computer \(Online\)](#) von Dr.phil.nat.Dr.rer.pol. [Matthias Hanauske](#)

Nächster Zoom Link am 09.07.2021, 15.00-17.00 Uhr:
ID: 794 847 5614, PWD: 785453

[Die Vorlesungen](#)

[Teil I](#)

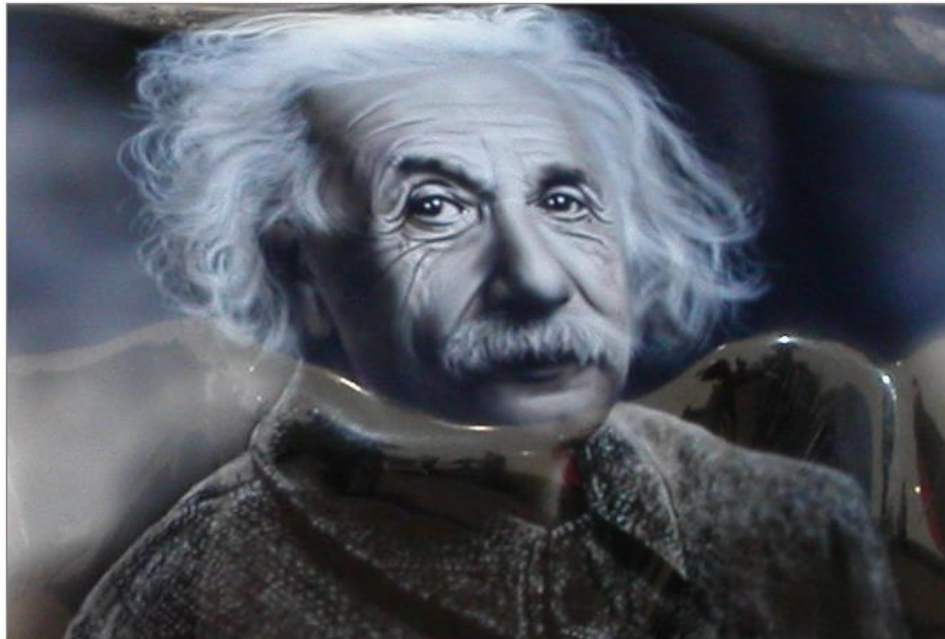
[Teil II](#)

[Teil III](#)

[E-Learning](#)

Vorwort

Die Vorlesung *Allgemeine Relativitätstheorie mit dem Computer* wurde im Sommersemester 2016 das erste Mal gehalten und viele der auf dieser Hauptseite erreichbaren Internetseiten basieren grundsätzlich auf dem damals erstellten Kurs. In der Vorlesung werden die mathematisch anspruchsvollen Gleichungen der Allgemeinen Relativitätstheorie (ART) in diversen Programmierumgebungen analysiert. Im ersten Teil des Kurses erlernen die Studierenden die Verwendung von Computeralgebra-Systemen (Python Jupyter Notebooks, Maple und Mathematica). Die oft komplizierten und zeitaufwendigen Berechnungen der tensoriellen Gleichungen der ART können mithilfe dieser Programme erleichtert werden.



Diverse Anwendungen der Einstein- und Geodätengleichung werden in Jupyter Notebooks (eine Open-Source, web-basierte interaktive Programmierumgebung) und Maple Worksheets implementiert, quasi analytische Berechnungen durchgeführt und entsprechende Lösungen berechnet und visualisiert. Der zweite Teil des Kurses befasst sich mit der numerischen Berechnung von Neutronensternen und Weißen Zwergen mittels eines C/C++ Programms. Nach einer kurzen Auffrischung der grundlegenden Programmierkenntnisse, wird ein Programm besprochen, das die Tolman-Oppenheimer-Volkoff Gleichung numerisch löst und die Ergebnisse werden visualisiert. Zusätzlich wird hierbei in die Grundkonzepte der parallelen Programmierung eingeführt und eine MPI- und OpenMP-Version des C/C++-Programms erstellt. Im dritten Teil des Kurses werden zeitabhängige numerische Simulationen der

Allgemeine Relativitätstheorie mit dem Computer (General Theory of Relativity on the Computer) Vorlesung SS 2021

Aufgrund der andauernden Corona-Krise findet die Vorlesung und die Übungstermine auch in diesem Semester nur Online statt!

Diese Internetseite fasst die Online-Angebote der Vorlesung *Allgemeine Relativitätstheorie mit dem Computer* zusammen. Auf der linken Seite finden Sie die einzelnen Vorlesungsaufzeichnungen (Videos), Vorlesungspräsentationen (pdf-Dateien) und weiterführende Links. Die Vorlesungstermine (Zoom Meetings, synchrones Lehrangebot) finden jeweils freitags von 15.00-17.00 Uhr statt. An den Online-Übungen können Sie entweder freitags vor (13.30-15.00 Uhr) oder nach der Vorlesung (17:00-18:30 Uhr) teilnehmen. Alle Lehrangebote werden mittel der Zoom Meeting Software gemacht und die jeweiligen Zoom-Links sind in der rechten oberen Ecke dieser Internetseite angegeben.

Die Inhalte der Vorlesung gliedern sich in drei Teile ([Teil I](#), [Teil II](#), [Teil III](#)), die Sie in der zweiten oberen Spalte einsehen können. Weiteres Zusatzmaterial und diverse Online-Aufgaben sind über die Online-Lernplattformen [OLAT](#) und [Lon Capa](#) erhältlich (siehe [E-Learning](#)).

Der Schwerpunkt der gesamten interaktiven Vorlesung liegt sowohl auf der Allgemeine Relativitätstheorie als auch auf der Vermittlung spezieller Programmierkenntnisse. Bei regelmäßiger Teilnahme an der Online-Vorlesung und den zugehörigen Übungseinheiten kann man einen benoteten bzw. unbenoteten Schein mit fünf Creditpoints zu erhalten.

Weiterführende Literatur

- *General relativity : An introduction for physicists* von M. P. Hobson, G. P. Efstathiou und

<http://itp.uni-frankfurt.de/~hanauske/VARTC/VARTC2021.html>

Diverse Jupyter Notebooks auch in der Vorlesung

Physik der sozio-ökonomischen Systeme mit dem Computer

Physik der sozio-ökonomischen Systeme mit dem Computer (Online) von Dr.phil.nat.Dr.rer.pol. [Matthias Hanauske](#)

Nächster Zoom Link am 18.02.2022, 15:00-17:00 Uhr:
ID: 794 847 5614, PWD: 785453

[Die Vorlesungen](#)

[Teil I](#)

[Teil II](#)

[Teil III](#)

[E-Learning](#)

Vorwort

Die Vorlesung *Physik der sozio-ökonomischen Systeme mit dem Computer* wurde im Wintersemester 2015/16 das erste Mal gehalten und viele der auf dieser Hauptseite erreichbaren Internetseiten basieren grundsätzlich auf dem damals erstellten Kurs. Das nebenstehende Video (ist in Arbeit!) gibt einen kurzen Überblick der Inhalte der Vorlesung. In der ersten Vorlesung (siehe Zoom Link in der rechten oberen Ecke) werden die Voraussetzungen besprochen, die man benötigt um einen benoteten bzw. unbenoteten Schein mit fünf Creditpoints zu erhalten.

Weiterführende Links



- [Zoom Meeting Software](#)
- [Online-Lernplattform](#)



- [OLAT](#)
- [Online-Lernplattform Lon Capa](#)

Physik der sozio-ökonomischen Systeme mit dem Computer (Physics of Socio-Economic Systems with the Computer) Vorlesung WS 2021/22

Auch in diesem Semester findet die Vorlesung nur Online statt!

Diese Internetseite fasst die Online-Angebote der Vorlesung *Physik der sozio-ökonomischen Systeme mit dem Computer* zusammen. Auf der linken Seite finden Sie die einzelnen Vorlesungspräsentationen (pdf-Dateien), Computerprogramme und weiterführende Links. Die Vorlesungstermine (Zoom Meetings, synchrones Lehrangebot) finden jeweils freitags von 15.00-17.00 Uhr statt. An den Online-Übungen können Sie entweder freitags vor (13.30-15.00 Uhr) oder nach der Vorlesung (17:00-18:30 Uhr) teilnehmen (Beginn der Online-Übungen erst am 29.10.2021). Alle Lehrangebote werden mittel der Zoom Meeting Software gemacht und die jeweiligen Zoom-Links sind in der rechten oberen Ecke dieser Internetseite angegeben.

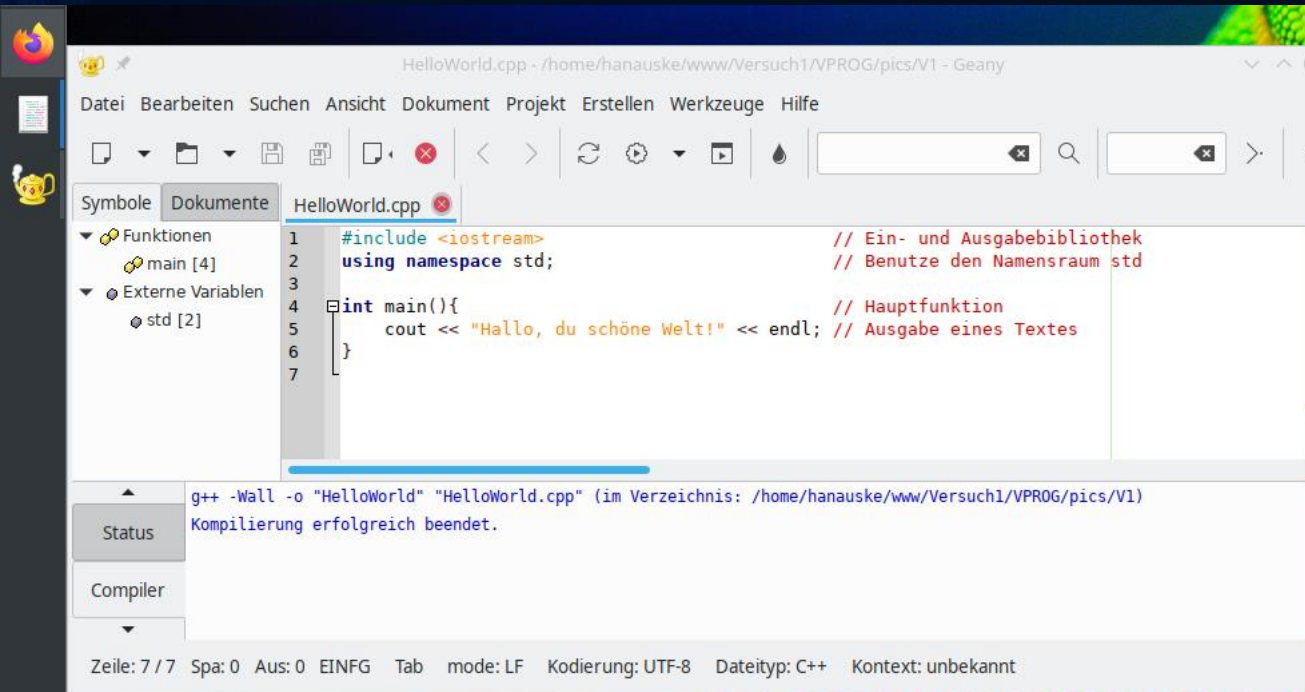
Die Inhalte der Vorlesung gliedern sich in drei Teile ([Teil I](#), [Teil II](#), [Teil III](#)), die Sie in der zweiten oberen Spalte einsehen können. Weiteres Zusatzmaterial und diverse Online-Aufgaben sind über die Online-Lernplattformen [OLAT](#) und [Lon Capa](#) erhältlich (siehe [E-Learning](#)).

Weiterführende Literatur

- Schlee, Walter, Einführung in die Spieltheorie, Vieweg 2004
- Hofbauer, Josef, and Karl Sigmund. Evolutionary games and population dynamics. Cambridge university press, 1998
- [Martin A. Nowak, Evolutionary Dynamics - Exploring the Equations of Life, 2006](#)
 - [Albert-Laszlo Barabasi, Network science, Cambridge university press, 2016](#)
- [Matthias Hanauske, Evolutionäre Quanten-Spieltheorie im Kontext sozio-ökonomischer Systeme, 2011](#)

<http://itp.uni-frankfurt.de/~hanauske/VPSOC/VPSOC2021.html>

Welches Tools ihr zum Programmieren von C++ und Python Skripts verwenden wollt, ist euch freigestellt. Theoretisch reichen das Terminal, ein Compiler für C++, eine Python3 Installation und ein einfacher Texteditor aus.

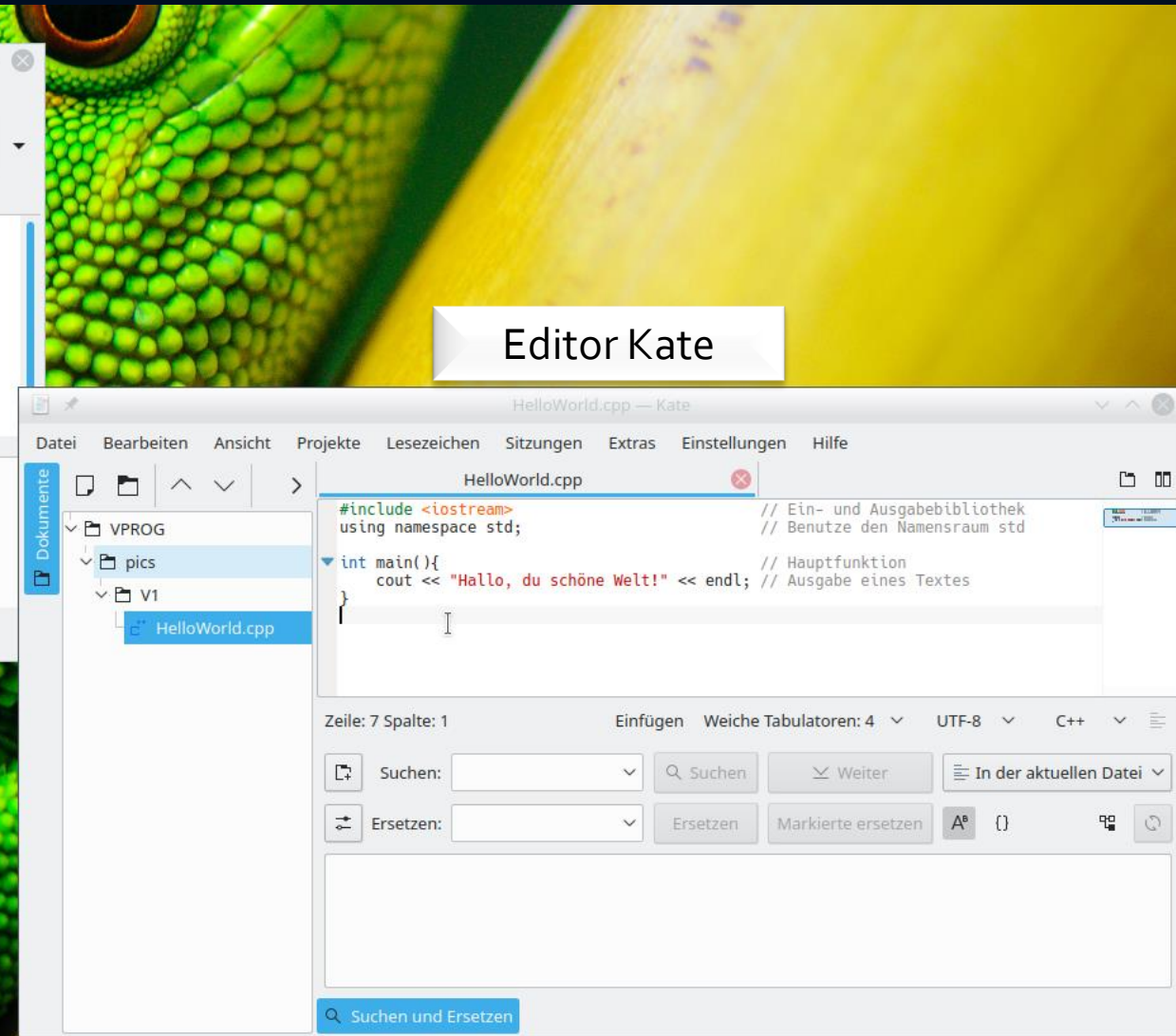


The screenshot shows the Geany IDE interface. The main window displays a C++ file named 'HelloWorld.cpp' with the following code:

```
1 #include <iostream> // Ein- und Ausgabebibliothek
2 using namespace std; // Benutze den Namensraum std
3
4 int main(){ // Hauptfunktion
5     cout << "Hallo, du schöne Welt!" << endl; // Ausgabe eines Textes
6 }
7
```

The status bar at the bottom indicates the compilation was successful: 'g++ -Wall -o "HelloWorld" "HelloWorld.cpp" (im Verzeichnis: /home/hanauske/www/Versuch1/VPROG/pics/V1) Kompilierung erfolgreich beendet.'

Editor Geany



The screenshot shows the Kate text editor interface. The main window displays the same C++ file 'HelloWorld.cpp' with the following code:

```
#include <iostream> // Ein- und Ausgabebibliothek
using namespace std; // Benutze den Namensraum std

int main(){ // Hauptfunktion
    cout << "Hallo, du schöne Welt!" << endl; // Ausgabe eines Textes
}
```

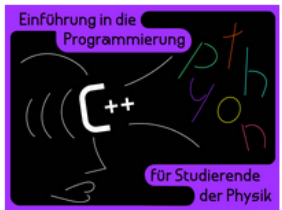
The status bar at the bottom shows the current cursor position: 'Zeile: 7 Spalte: 1' and the encoding: 'UTF-8'.

Editor Kate

Moderne integrierte Entwicklungsumgebungen (IDE) können euch jedoch helfen, Probleme in eurem Code zu erkennen und schneller zu programmieren. Ein Beispiel für eine anfängerfreundliche IDE ist [Visual Studio Code](#) oder [Eclipse](#).

Die OLAT Seite und die Vergabe der Login-Accounts

- 📁 Einführung in die Programmierung
 - 📄 Infos zu den Übungen
 - 📄 Einschreibung - Vorlesung
 - 📄 Einschreibung Tutorien
- 📁 Kursinhalt
 - 📄 Mitteilungen
 - 🗨️ Forum
 - 📁 Übungsabgaben
 - 📄 Übungsblatt 1
 - 🔗 Linkliste
 - ✉️ E-Mail an den Dozenten
- 👤 Gruppen



Sommersemester 2022
Einführung in die Programmierung
Verantwortliche/r: [Matthias Hanauske](#) | [Niklas Götz](#)
Einführung in die Programmierung für Studierende der Physik

Vorlesung (Beginn 12.04.): Di 15-16 Uhr Do 14-16 Uhr jeweils Raum Phys 0.111	Dozent: Matthias Hanauske (hanauske@fias.uni-frankfurt.de)
Kurswebsite	
Übungen (Beginn 18.04.): Raum Phys 01.120	Tutorienleitung: Niklas Götz (goetz@fias.uni-frankfurt.de)
Übungsgruppen	Tutoren
Tutorium 1 (Di 9.15 - 11.00 - Englisch)	Arijit Dutta (dutta@itp.uni-frankfurt.de)
Tutorium 2 (Di 11.15 - 13.00 - Englisch)	Youjiang Xu (yxu@itp.uni-frankfurt.de)
Tutorium 3 (Mi 9.15 - 11.00)	Nils Saß (nsass@fias.uni-frankfurt.de)
Tutorium 4 (Mi 14.15 - 16.00)	Keneth Miler (milerkeneth@gmail.com)
Tutorium 5 (Do 9.15 - 11.00)	Justin Mohs (mohs@fias.uni-frankfurt.de)
Tutorium 6 (Fr 9.15 - 11.00)	Michael Fromm (mfromm@itp.uni-frankfurt.de)

Schreibt euch bitte zur Teilnahme an dem Kurs auf der linken Seite in die Vorlesung und in die Übungsgruppe eurer Wahl ein. Dort findet ihr auch mehr Informationen über die Übungsmodalitäten.

Übungsblatt Nr. 1

Aufgabe 1 (10 Punkte)

In den ersten beiden Vorlesungen haben Sie Ihren Login-Account für die Rechner des Instituts für Theoretische Physik (ITP) der Goethe-Universität erhalten. Sollte dies jedoch nicht so sein, dann wenden Sie sich bitte direkt per E-Mail an Ihren Tutor (siehe Informationen zu den Übungen auf der [Online-Lernplattform OLAT](#)). Bitte lesen Sie die [Allgemeine Nutzungsordnung für die Informationsverarbeitungs- und Kommunikations-Infrastruktur der Johann Wolfgang Goethe-Universität](#), bevor Sie den Login-Account das erste Mal verwenden.

Führen Sie die folgenden Teilaufgaben aus und halten diese mittels eines Bildschirmfotos (Screenshots) fest:

- Bauen Sie eine SSH-Verbindung mit den Rechnern des ITP auf.
- Ändern Sie das Passwort Ihres ITP-Accounts.
- Erstellen Sie einen neuen Ordner "VPROG" und entzippen dort den folgenden zip-File: [Download zip-File](#).
- Erstellen Sie Ihre eigene Homepage. Es reicht hierbei aus, wenn Sie den im zip-File befindlichen HTML-File "index.html" in Ihr "public_html" Verzeichnis kopieren und überprüfen, dass die Zugriffsrechte für alle lesbar sind.
- Zeigen Sie durch ein Bildschirmfoto eines Internet-Browsers, dass ihre Homepage Online ist.
- Ändern Sie nun die Rechte der HTML-Datei "index.html", sodass nur noch Sie und Gruppenmitglieder die Datei lesen können und zeigen Sie dann mittels eines Bildschirmfotos eines Internet-Browsers, dass ihre Homepage nicht mehr zugänglich ist.
- Kompilieren Sie den C++ Quelltext "HelloWorld.cpp" und führen das Programm aus.
- Führen Sie das Python Programm "HelloWorld.py" aus.

Aufgabe 2 (5 Punkte)

Erstellen Sie einen bootfähigen USB-Stick mit einer Linux-Distribution Ihrer Wahl und starten Sie Linux im Live USB Modus. Machen Sie ein Bildschirmfoto, wenn das Linux Betriebssystem hochgefahren ist. Falls Sie Linux schon installiert haben, machen Sie ein Bildschirmfoto von Ihrem Heimrechner.

Aufgabe 3 (5 Punkte)

Installieren Sie Python und Jupyter auf Ihrem Heimrechner und öffnen Sie die das Jupyter Notebook "HelloWorld.ipynb" auf ihrem eigenen System. Alternativ können Sie auch mittels einer grafischen Remote-Verbindung das Jupyter Notebook auf den Rechnern des ITP öffnen. Machen Sie ein Bildschirmfoto.

Bitte laden Sie die einzelnen Bildschirmfotos der Aufgaben auf der [Online-Lernplattform OLAT](#) hoch; gerne auch als gebündeltes, komprimiertes .zip-File.

Vorlesung 2

Im Folgenden wird vorausgesetzt, dass Sie auf Ihrem eigenen Computer einen lauffähigen *C++ Compiler* und *Python 3* installiert haben. Wir beginnen, wie viele andere Einführungen in die Programmierung, mit der C++ Version des klassischen Hello World Programms. Es wird die Textausgabe im Linuxterminal und Eingabe von Zahlenwerten über die Tastatur vorgestellt. Will man numerische Berechnungen in der Programmiersprache C++ durchführen, muss man zunächst definieren, in welchem Zahlenraum die numerischen Variablen sich aufhalten und einen speziellen Typ den Variable zuordnen. Die grundlegenden, integrierten Datentypen von C++ und die mittels Operatoren integrierte Arithmetik werden in dieser Vorlesung vorgestellt. Am Ende der Vorlesung werden die Benutzereingabe und die formatierte Ausgabe von Zahlenwerten behandelt.

Das erste C++ Programm (Hello World)

In diesem Unterpunkt wird die Programmiersprache C++ am Beispiel des klassischen *Hello World Programms* vorgestellt. C++ ist eine kompilierte Programmiersprache, d.h. der vom Programmierer geschriebene *Quelltext* des Computerprogramms muss zunächst mittels eines *Compilers* in ein ausführbares Programm umgewandelt werden, damit es genutzt werden kann. Dieser Kompilierungsprozess und die Textausgabe des ausführbaren *Hello World Programms* im Linuxterminal (mittels `cout` bzw. `printf`) wird unter folgendem Link vorgestellt: [Das erste C++ Programm \(Hello World\)](#).

Datentypen und Variablen

In gleicher Weise, wie es in der Mathematik die unterschiedlichen Zahlenmengen (z.B. \mathbb{Z} , \mathbb{N} , \mathbb{R} , \mathbb{C}) gibt und man bei mathematischen Berechnungen den Wertebereich einer Variable oder Funktion vorher definieren muss, ist dies auch beim Programmieren in C++ nötig. In diesem Unterpunkt werden die Begriffe *Deklaration einer Variable*, *Datentyp*, *Wert einer Variable* und die *Initialisierung einer Variable* diskutiert. Die wichtigsten integrierten Datentypen von C++ sind `bool`, `char`, `int`, `float` und `double`. Näheres siehe unter folgendem Link: [Datentypen und Variablen](#)

Arithmetik und Operatoren

Möchte man mit Datentypen und Variablen mathematische Berechnungen durchführen oder die Datenwerte miteinander vergleichen, ist es zunächst nötig eine Arithmetik (die zum Zählen oder Rechnen gehörige Kunst) zu definieren. Hierzu wurden in C++ geeignete Operatoren definiert, die dem Programmierer neben den arithmetischen Grundrechenarten (Multiplikation, Addition, ...) noch diverse weitere integrierte Operatoren zur Verfügung stellen. Man unterscheidet hierbei die *arithmetischen Operatoren*, die *logischen Vergleichsoperatoren* und weitere *spezifische Operatoren*. Zusätzlich sind einige wichtige mathematische Funktionen (Sinus, Cosinus, ..) in der Standardbibliothek `<cmath>` vordefiniert (näheres siehe [Arithmetik und Operatoren](#)).

Die Ein- und Ausgabe

Vorlesung 2

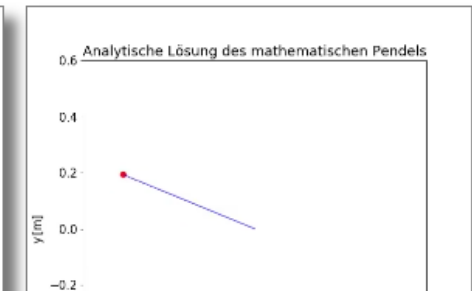
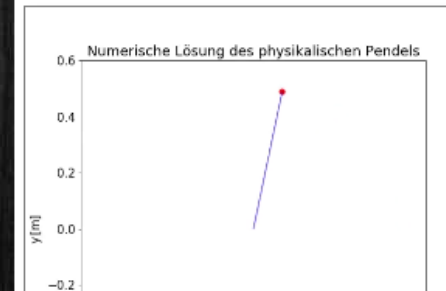
Es geschieht einem theoretischen Physiker leider nur zu oft, dass er ein System beschreiben möchte, welches er auf analytischem, mathematischem Weg nicht lösen kann. Betrachten wir z.B. das einfache physikalische Pendel aus dem Themenbereich der Mechanik (siehe z.B. [W.Greiner, Klassische Mechanik I](#)). Die zugrundeliegende Differentialgleichung (DGL) des Problems lautet

$$\frac{d^2\phi(t)}{dt^2} = \frac{g}{l} \cdot \sin(\phi(t)) \quad ,$$

wobei g die Erdbeschleunigung, l die Länge des Pendels und $\phi(t)$ die zeitliche Entwicklung des Pendelwinkels beschreibt.

Ungünstigerweise besitzt diese DGL keine analytisch darstellbare Lösung und gewöhnlich vereinfacht man dann das zu beschreibende System (z.B. durch Approximation zum mathematischen Pendel) und gelangt auf diesem Wege doch zu analytischen Lösungen, deren Verhalten man studieren kann. Vergleicht man die Lösung dann mit der Realität, stimmt diese oft nur unter gewissen Randbedingungen.

Nimmt man jedoch Anfangswerte des Pendels, die außerhalb des Gültigkeitsbereiches der approximierten Lösung liegen, weichen die Vorhersagen der Pendelbewegung deutlich von der wirklichen Lösung ab. Stoßen wir z.B. das Pendel mit einem sehr starken "Schubs" aus seiner Ruhelage, so stimmt die analytische Lösung des mathematischen Pendels nicht und das wirkliche Pendel verhält sich vollkommen anders. Die folgenden Animationen entstammen einem Python Jupyter Notebook und sie vergleichen die numerische Lösung des physikalischen Pendels mit der analytischen Lösung des mathematischen Pendels, bei der ein Pendelüberschlag nicht möglich ist.



Vorlesung 3

C++ Anweisungen: Die for-, while- und do-Schleifen

Möchte man als Programmierer ein Problem mittels eines C++ Programms lösen, so muss man dem Computer in Form von Anweisungen sagen, was er zu erledigen hat. In diesem Unterpunkt behandeln wir eine der wichtigsten Anweisungsarten, die sogenannten *Schleifenanweisungen*. Im Prinzip ist jede Programmzeile, die mit einem Semikolon endet, eine Anweisung an den Computer, jedoch stellen die *Schleifenanweisungen* eine besondere Art von iterativen Anweisungsprozessen dar, und sind ein oft verwendetes Hilfsmittel der prozeduralen Programmierung. Eine Schleifenanweisung kann als eine **for**-, **while**- oder **do**-Anweisung ausgedrückt werden (näheres siehe C++ Anweisungen: Die for-, while- und do-Schleifen).

Anwendungsbeispiel: Folgen und Reihen

Die im vorigen Unterpunkt besprochenen *Schleifenanweisungen* finden in vielen C++ Programmen ihre Anwendung. In diesem Unterpunkt wird ihre Anwendung im Bereich der mathematischen Folgen und Reihen diskutiert. Am Beispiel der konvergenten Folge der Eulersche Zahl e und der Leibniz-Reihe zur Berechnung der Kreiszahl π wird die Verwendung der while-Schleife vorgestellt. Das Umschreiben der Programme unter Verwendung einer for-Schleife ist Teil der Aufgabe 2 des (siehe Übungsblattes Nr. 3). Es wird unter anderem das Konvergenzverhalten der Folge für die ersten Folgenglieder untersucht und die von vom Programm ausgegebenen Werte visualisiert. Hierzu werden die ausgegebenen Daten in eine separate Datei umgeleitet und dann mittels Gnuplot bzw. Python-Matplotlib dargestellt (näheres siehe Anwendungsbeispiel: Folgen und Reihen).

Eine kleine Einführung in die Programmiersprache Python

Die Programmiersprache Python ist auch eine sehr gute Objekt-orientierte Programmiersprache und im Prinzip hätten wir die gesamte Vorlesung nur mittels Python gestalten können. Die in dieser Vorlesung behandelten Python-Skripte und Python Jupyter Notebooks werden jedoch lediglich zur Visualisierung von Daten und im Bereich der Illustration von mathematisch/physikalischen Gleichungen benutzt. Mittels des Python-Moduls "matplotlib" (siehe Matplotlib: Visualization with Python) können auf einem einfachen Weg Bilder and Animationen des zuvor mit C++ simulierten Systems erzeugt werden. Zusätzlich werden wir, die im nächsten Unterpunkt besprochene C++ Computerarithmetik mittels eines Python Jupyter Notebooks verdeutlichen und dabei die Verwendung von Listen, Arrays und for-Schleifen in der Programmiersprache Python kennenlernen (näheres siehe Eine kleine Einführung in die Programmiersprache Python).

Die Computerarithmetik und der Fehler in numerischen Berechnungen

In diesem Unterpunkt werden wir zwei Klassen von Fehlerquellen in numerischen Berechnungen besprechen, den sogenannten *Rundungsfehler*, der aufgrund der Computerarithmetik entsteht und der sogenannte *Approximierungsfehler* (*Abschneidefehler* bzw. "Truncation error"), der immer dann auftritt, wenn der Programmierer eine exakte mathematische Gleichung mittels approximativer Ausdrücke annähert (näheres siehe Die Computerarithmetik und der Fehler in numerischen Berechnungen).

Vorlesung 3

In dieser Vorlesung werden wir die wohl wichtigste Form von C++ Anweisungen, die sogenannten *Schleifenanweisungen*, kennenlernen. Die *Schleifenanweisungen* stellen einen iterativen Anweisungsprozess dar und können in Form von **for**-, **while**- oder **do**-Anweisung ausgedrückt werden. Möchte man z.B. die natürlichen Zahlen von Null bis 100 im Terminal ausgeben lassen, so kann man dies in einer einfachen Weise mittels einer Schleifenanweisung dem Computer sagen. Die Anwendung von Schleifenanweisung wird danach am Beispiel einer mathematischen Folge und Reihe diskutiert. Die Visualisierung von Daten, die mittels C++ Programmen erstellt wurden, ist ein wichtiges Teilgebiet eines Programmierers im Bereich der Physik. In dieser Vorlesung werden diverse Python-Skripte und Python Jupyter Notebooks vorgestellt, die zur Visualisierung von Daten und im Bereich der Illustration von mathematisch/physikalischen Gleichungen benutzt werden. Mittels des Python-Moduls "matplotlib" (siehe Matplotlib: Visualization with Python) können auf einem einfachen Weg Bilder and Animationen des zuvor mit C++ simulierten Systems erzeugt werden.

Am Ende der Vorlesung kommen wir nochmals auf den, bereits im Unterkapitel Datentypen und Variablen angesprochenen Zahlenraum des Computers (\mathbb{R}_C) zurück und diskutieren die zwei wichtigsten Fehlerquellen bei numerischen Berechnungen (*Rundungsfehler* und *Approximierungsfehler*). Der *Rundungsfehler* ist darin begründet, dass der Zahlenraum des Computers (\mathbb{R}_C), nur eine relativ kleine Teilmenge der reellen Zahlen benutzt und somit reellwertige Zahlen mit einer unendlichen Anzahl von Nachkommastellen nicht speichern kann. Diese Teilmenge \mathbb{R}_C umfasst nur rationale Zahlen und speichert den gebrochenen Teil, der mit *Mantisse* bezeichnet wird, zusammen mit dem exponentiellen Teil, welchen man *Charakteristik* nennt als eine binäre Liste von Nullen und Einsen. Die zweite Klasse von numerischen Fehlern, die *Approximierungsfehler* werden am Beispiel der approximativen Annäherung an die Kreiszahl π mittels der alternierenden Leibniz-Reihe besprochen. Bei dieser Art von Fehlern hat es der Programmierer selbst in der Hand, wie genau er in seinem Programm die Berechnung durchführen möchte (näheres siehe Die Computerarithmetik und der Fehler in numerischen Berechnungen).

Vorlesung 4

....

C++ Anweisungen: Auswahlanweisungen mit if und switch

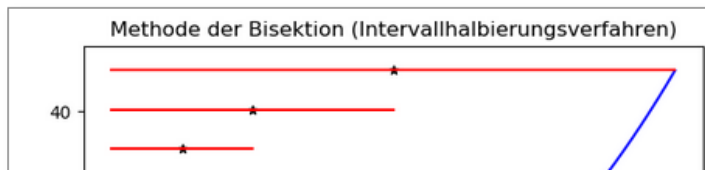
Die Programmiersprache C++ bietet einen konventionellen und flexiblen Satz von Anweisungen. Im Prinzip ist jede Programmzeile, die mit einem Semikolon endet, eine Anweisung. In diesem Unterpunkt werden wir die sogenannten *Auswahlanweisungen* behandeln, wobei wir in der vorigen Vorlesung die *Schleifenanweisungen* vorstellt hatten (siehe C++ Anweisungen: Die for-, while- und do-Schleifen). *Auswahlanweisungen* werden auch als *Verzweigung*, *Tests* oder *bedingte Anweisungen* bezeichnet und sind immer dann anzuwenden, wenn das Programm bei einem gewissen Ereignis bzw. unter einer gewissen Bedingung etwas Bestimmtes tun soll. Es teilt somit das Programm in unterschiedliche Anweisungspfade auf. *Auswahlanweisungen* können in Form einer **if-** (**if-else**)- oder **switch**-Anweisung ausgedrückt werden. C++ Anweisungen: Auswahlanweisungen mit if und switch

Funktionen in C++



Die Definition einer C++ Funktion ist im Grunde nichts Anderes, als eine Code-Block (*Anweisungsblock*) mit einem Funktionsnamen zu verbinden. C++ Funktionen werden außerhalb der main()-Hauptfunktion definiert und vereinfachen somit das Verständnis und die Lesbarkeit des Quelltextes. C++ Funktionen sind ein wichtiges Werkzeug, um den Quelltext eines Programms zu ordnen und wesentliche Algorithmen und zusammenhängende Anweisungsblöcke der main()-Hauptfunktion in einer zusammenhängenden Form auszulagern. Die Definition einer C++ Funktion besteht aus einer *Deklaration* und einem *Anweisungsblock* und sie ist der formalen Definition einer mathematischen Funktion nicht unähnlich: "Eine C++ Funktion ist eine Abbildung von dem Datenraum der *Argumentenliste* in den Datenraum des *Rückgabetyps*. Die dabei benutzte Abbildungsvorschrift findet sich in dem *Anweisungsblock* der Funktion. Der 'Rückgabe Typ' kann hierbei eine der schon besprochenen Datentypen (z.B. `int` oder `double`) oder ein Daten-Array (siehe nächste Vorlesung) sein. Die *Argumentenliste* setzt sich aus einer Liste von Datentypen der formalen Argumente (Parameter) der Funktion zusammen, die jeweils mit einem Komma voneinander getrennt sind. In C++ hat das Wort Funktion eine allgemeinere Bedeutung als im Bereich der Mathematik und die in der Mathematik und Physik definierte Funktionen stellen eine echte Teilmenge der C++ Funktionen dar (näheres siehe Funktionen in C++).

Anwendungsbeispiel: Nullstellensuche einer Funktion



In diesem Unterpunkt möchten wir ein grundlegendes Problem der numerischen Mathematik behandeln, die Nullstellensuche einer Funktion. Wir betrachten im Speziellen die Funktion $f(x) = e^x - 10$ und wollen berechnen, bei welchem x-Wert die Funktion Null wird ($f(x) = 0$). Hat eine Funktion im Teilintervall $[a, b] \in \mathbb{R}$ eine

Vorlesung 4

....

Die weiteren
Vorlesungen
sind in
Bearbeitung