

- 11.6.1 Projekt: Motivator
- 11.7 Das Layout verfeinern
- 11.8 Raster-Layout
- 11.9 Projekt: 25 Farben – ein automatisches Farbfelder-Bild
- 11.10 Widgets zur Texteingabe
 - 11.10.1 Einzeilige Eingabe: Das Entry-Widget
 - 11.10.2 Mehrzeilige Eingabe: Das Text-Widget
 - 11.10.3 Projekt: Reimen mit Goethe
- 11.11 Radiobuttons
 - 11.11.1 Projekt: Währungsrechner
- 11.12 Dialogboxen
 - 11.12.1 Projekt: Texteditor
- 11.13 Parallele Abläufe: Threads
 - 11.13.1 Ein Experiment: Countdown
 - 11.13.2 Eine Funktion in einem eigenen Thread ausführen
- 11.14 Rückblick
- 11.15 Übungen
- 11.16 Lösungen zu den Fragen

12 Grafik programmieren

- 12.1 Bilder auf Schaltflächen und Labels
 - 12.1.1 Projekt: Würfelspiel
 - 12.1.2 Bilder verändern
 - 12.1.3 Projekt: Graustufen
- 12.2 Canvas
 - 12.2.1 Flächen gestalten
 - 12.2.2 Linien gestalten
 - 12.2.3 ID-Nummern: Elemente löschen oder bewegen
- 12.3 Projekt: Creative Coding
- 12.4 Die Python Imaging Library (PIL)
 - 12.4.1 Ein Image-Objekt aus einer Datei gewinnen

- 12.4.2 Ein Image-Objekt ohne Datei erzeugen
- 12.4.3 Attribute und Methoden von Image-Objekten
- 12.4.4 Bilder über Listen verarbeiten
- 12.4.5 Bilder einfügen
- 12.4.6 Projekt: Greenscreen
- 12.4.7 PIL.Image-Objekte in tkinter-Anwendungen
- 12.4.8 Projekt: Webcam-Viewer
- 12.5 Rückblick
- 12.6 Übungen

13 Fehler finden und vermeiden

- 13.1 Zusicherungen
- 13.2 Tracing
 - 13.2.1 Beispiel: Quicksort
- 13.3 Debugging mit IDLE
 - 13.3.1 Der Debugger der Python-Shell
 - 13.3.2 Das Programm schrittweise durchlaufen
 - 13.3.3 Haltepunkte setzen
- 13.4 Debugging mit Thonny
- 13.5 Rückblick
- 13.6 Lösungen zu den Fragen

14 Objektorientierte Programmierung

- 14.1 Klassen und Objekte
 - 14.1.1 Was ist Objektorientierung?
 - 14.1.2 Klassen entwerfen und grafisch darstellen – UML
 - 14.1.3 Definition einer Klasse
 - 14.1.4 Objekte einer Klasse erzeugen: Instanziierung
 - 14.1.5 Auf Attribute zugreifen
 - 14.1.6 Methoden aufrufen
 - 14.1.7 Objekte mit variablen Anfangswerten
 - 14.1.8 Metaphern in der Programmierung

- 14.2 Projekt: Geld
 - 14.2.1 Mit Geld-Objekten rechnen
 - 14.2.2 Klassenattribute
 - 14.2.3 Operatoren überladen – Polymorphie
- 14.3 Magische Methoden
- 14.4 Projekt: Abrechnung
- 14.5 Vererbung
- 14.6 Projekt: Farbtester
- 14.7 Projekt: Zahlenregen
- 14.8 Rückblick
- 14.9 Übungen
- 14.10 Lösungen zu den Fragen

15 Datenbanktechnik

- 15.1 Was ist ein Datenbanksystem?
- 15.2 Eine Datenbank entwerfen – das Entity-Relationship-Diagramm (ER)
- 15.3 Relationale Datenbanken
- 15.4 Relationen mit Python darstellen *
 - 15.4.1 Menge von Tupeln
 - 15.4.2 Benannte Tupel (named tuples)
- 15.5 Das Modul sqlite3 – Schnittstelle zu einer SQL-Datenbank
 - 15.5.1 Mit SQL Tabellen anlegen und Tupel eintragen
 - 15.5.2 Mit sqlite3 eine SQLite-Datenbank aufbauen
 - 15.5.3 Formulierung von Anfragen (Queries) mit SQL
 - 15.5.4 Datenbankanfragen in Python-Programmen
 - 15.5.5 SQL-Anweisungen mit variablen Teilen
- 15.6 Projekt: Zitatesammlung
 - 15.6.1 ER-Diagramm
 - 15.6.2 Tabellen (Beispiel)
 - 15.6.3 Administration der Zitatesammlung

- 15.6.4 Nach Zitaten suchen
- 15.7 Rückblick
- 15.8 Übungen
- 15.9 Lösungen zu den Fragen

16 Wissenschaftliche Projekte

- 16.1 NumPy – Rechnen mit Arrays
 - 16.1.1 Arrays
 - 16.1.2 Indizieren
 - 16.1.3 Slicing
 - 16.1.4 Arrays verändern
 - 16.1.5 Arithmetische Operationen
 - 16.1.6 Funktionen, die elementweise ausgeführt werden
 - 16.1.7 Matrizenmultiplikation mit dot()
 - 16.1.8 Array-Funktionen und Achsen
- 16.2 Datenvisualisierung mit matplotlib
 - 16.2.1 Liniendiagramme
 - 16.2.2 Mehrere Linien in einem Diagramm
 - 16.2.3 Histogramme
 - 16.2.4 Projekt: Würfeln
 - 16.2.5 Heatmaps
- 16.3 Projekt: Bewegungsprofil
- 16.4 Google Colaboratory – Colab
 - 16.4.1 Ein Colab-Notebook erzeugen
 - 16.4.2 Text-Zellen
 - 16.4.3 Bilder einfügen
 - 16.4.4 Notebooks speichern und öffnen
- 16.5 Projekt: Füchse und Hasen – Simulation eines Räuber-Beute-Systems
 - 16.5.1 Notebooks teilen
- 16.6 Rückblick
- 16.7 Übungen

16.8 Lösungen zu den Fragen

17 Dynamische Webseiten: CGI und WSGI

17.1 Dynamische Webseiten mit CGI

17.1.1 Projekt: Wie spät ist es?

17.1.2 Die Ausgabe eines CGI-Skripts

17.1.3 Wie ist ein CGI-Skript aufgebaut?

17.1.4 CGI-Skripte unter Windows

17.1.5 Aufruf mit dem Webbrowser

17.1.6 Ein HTTP-Server

17.1.7 Zugriff von einem anderen Computer im lokalen Netz

17.2 Interaktive Webseiten

17.2.1 Eingabekomponenten in einem HTML-Formular

17.2.2 Verarbeitung von Eingabedaten mit FieldStorage

17.3 Wie verarbeitet man Umlaute? *

17.4 Dynamische Webseiten mit WSGI

17.4.1 Das Applikationsobjekt

17.4.2 Skripte mit eigenem HTTP-Server – das Modul wsgiref

17.5 Projekt: Wie spät ist es? (II)

17.6 Projekt: Umfrage

17.6.1 Die HTML-Schablonen

17.6.2 Der algorithmische Teil

17.7 Einen Hosting-Dienst nutzen

17.7.1 Python Anywhere

17.7.2 Das vorgefertigte OWSGI-Programm ausprobieren

17.7.3 Projekt: Wie spät ist es? (III)

17.7.4 WSGI-Projekte modularisieren

17.8 Rückblick

17.9 Übungen

17.10 Lösung zur Frage: Interaktive Webseite

18 Professionelle Software-Entwicklung