

Inhalt

Vorwort — V

1	Was ist Informatik? — 1
2	Unix/Linux und Shell-Programmierung — 8
2.1	Grundlegendes — 8
2.1.1	Wozu dient ein Betriebssystem? — 8
2.1.2	Unix vs. Linux — 10
2.1.3	Der Aufbau von Linux — 10
2.1.4	Die Shell — 11
2.1.5	Multitasking — 12
2.1.6	Das Dateisystem von Unix — 13
2.2	Erste wichtige Kommandos — 14
2.2.1	Aufbau von Shell-Kommandos — 15
2.2.2	Befehle für Verzeichnisse — 16
2.2.3	Befehle für Dateien — 17
2.2.4	Befehle für die Benutzerverwaltung — 19
2.2.5	Befehle des Prozesssystems — 20
2.2.6	Sonstige Befehle — 21
2.3	Textdateien erstellen und editieren mit <code>vi</code> — 22
2.4	Features der Shell — 24
2.4.1	Eingabe — 24
2.4.2	Wildcards — 25
2.4.3	Umleitungen und Pipes — 26
2.4.4	Shellvariablen — 29
2.4.5	Ausblendung von Sonderbedeutungen — 31
2.4.6	Verknüpfungen von Kommandos — 32
2.5	Weitere Kommandos — 34
2.5.1	Das <code>find</code> -Kommando — 34
2.5.2	Das <code>grep</code> -Kommando — 36
2.5.3	Der <code>cut</code> -Befehl — 39
2.5.4	Das <code>sort</code> -Kommando — 40
2.5.5	Die <code>head</code> - und <code>tail</code> -Kommandos — 41
2.6	Shell-Programmierung — 41
2.6.1	Dateneingabe — 42
2.6.2	Kommandozeilenparameter — 43
2.6.3	Bedingungen testen — 44
2.6.4	Programmschleifen — 47

3	Python-Programmierung — 51
3.1	Arbeiten mit Python — 52
3.1.1	Python 3 vs. Python 2 — 52
3.1.2	Installation — 53
3.1.3	Ein erstes Python-Programm — 53
3.1.4	Die Python-Shell — 54
3.1.5	Python Notebooks — 54
3.2	Einfache Datentypen — 56
3.2.1	Zahlen — 56
3.2.2	Boolesche Werte — 56
3.2.3	Strings — 57
3.2.4	Variablen — 57
3.2.5	Operatoren — 58
3.3	Grundlegende Konzepte — 60
3.3.1	Einrücktiefe — 60
3.3.2	Kontrollfluss — 61
3.3.3	Schleifenabbruch — 65
3.3.4	Anweisungen vs. Ausdrücke — 65
3.3.5	Funktionen — 67
3.4	Zusammengesetzte Datentypen — 70
3.4.1	Listen und Sequenzen — 70
3.4.2	Allgemeine Sequenzoperationen — 72
3.4.3	Wichtige Operationen auf Listen — 74
3.4.4	Referenzen — 76
3.4.5	Tupel — 77
3.4.6	Dictionaries — 77
3.4.7	Strings (Fortsetzung) — 80
3.5	Funktionale Programmierung — 83
3.5.1	Listenkompansionen — 84
3.5.2	Die <code>map</code> -Funktion — 89
3.5.3	Die <code>filter</code> -Funktion — 91
3.5.4	Die <code>reduce</code> -Funktion — 93
3.6	Dateien und Verzeichnisse — 96
3.6.1	Datei-Objekte — 97
3.6.2	Dateimanipulation mit Listenkompansionen — 100
3.6.3	Verzeichnisse — 103
3.7	Objektorientierte Programmierung — 105
3.7.1	Definition und Verwendung einer Klasse — 106
4	Programmierung mit regulären Ausdrücken — 111
4.1	Verwendung Regulärer Ausdrücke in Python — 111
4.1.1	Das Kommando <code>re.findall</code> — 112

4.1.2	Das Kommando <i>re.sub</i> — 113
4.1.3	Das Kommando <i>re.search</i> — 116
4.2	Komponenten Regulärer Ausdrücke — 117
4.2.1	Einfache Konstrukte — 118
4.2.2	Rückwärtsreferenzen (Backreferences) — 124
4.2.3	Greedy vs. Non-Greedy — 126
4.2.4	Lookahead — 127
4.3	Reguläre Ausdrücke vs. Suchausdrücke mit Listenkomprehensionen — 130
5	Datenbanken und Datenbankprogrammierung — 135
5.1	Wozu Datenbanken? — 135
5.1.1	Daten-Persistenz — 135
5.1.2	Dateisystem als Datenspeicher — 135
5.1.3	Anforderungen an Persistenzmechanismen — 136
5.2	Datenbankmanagementsysteme (DBMS) — 137
5.2.1	Transaktionskonzept — 137
5.2.2	Funktionsweise eines DBMS — 138
5.2.3	Einsatz von DBMS — 139
5.3	Relationale DBMS — 139
5.3.1	Tabellen, Schemata, Zeilen, Spalten — 139
5.3.2	Erstellen einer Tabelle in MySQL mit Python — 140
5.4	SQL-Abfragen — 144
5.4.1	Relationenalgebra — 144
5.4.2	Das SELECT-Kommando — 145
5.4.3	Zählen und Statistiken — 148
5.4.4	Joins: Verknüpfung von Tabellen — 149
5.5	Entwurf relationaler DBMS — 150
5.5.1	Entity-Relationship-Diagramme — 150
5.5.2	Umsetzung in ein relationales Modell — 153
5.5.3	Normalisierung — 155
5.6	Nicht-Relationale DBMS — 158
5.6.1	CouchDB: Datenverfügbarkeit vs. Datenkonsistenz — 158
5.6.2	Funktionsprinzipien — 159
5.6.3	JSON — 162
5.6.4	Erzeugen einer CouchDB-Datenbank mit Python — 163
5.6.5	Benutzerdefinierte Views mit MapReduce — 165
6	Internet und Internetprogrammierung — 170
6.1	Einführung in die Funktionsweise des Internet — 170
6.1.1	Geschichtliches — 170
6.1.2	Netzwerk-Protokolle — 171

6.1.3	Das TCP/IP-Referenzmodell —	171
6.1.4	Internetworking —	173
6.1.5	Sockets —	174
6.1.6	Host, Server, Client —	175
6.2	Socketprogrammierung —	176
6.3	Dateitransfer mit FTP —	179
6.4	HTML und Datentransfer von URLs —	181
6.4.1	HTML —	181
6.4.2	Datentransfer von URLs —	182
6.5	Dynamische Web-Seiten —	184
6.5.1	htmlgen: Generierung von HTML-Code —	184
6.5.2	Ein einfacher Web-Server —	186
6.5.3	Ein erstes CGI-Skript —	187
6.5.4	Komplexere CGI-Skripte —	188
7	Nebenläufige und Parallele Programmierung —	191
7.1	Grundlegendes —	191
7.1.1	Prozesse, Tasks und Threads —	191
7.1.2	Nebenläufigkeit vs. Parallelität —	192
7.1.3	Multithreading, Time-Sharing und Threadzustände —	193
7.1.4	Programmierung mit Threads vs. Multi-Core-Programmierung —	194
7.2	Parallele Rechnerarchitekturen —	194
7.2.1	NOWs —	195
7.2.2	SMPs und Mehrkern-Prozessoren —	195
7.2.3	GPUs —	196
7.2.4	Hardware-seitiges Multithreading —	198
7.3	Techniken Paralleler Programmierung —	198
7.3.1	Locks —	198
7.3.2	Message-Passing —	199
7.3.3	Bulk Synchronous Parallel Model (BSP) —	199
7.4	Multithread-Programmierung in Python —	200
7.4.1	Das <code>threading</code> -Modul —	201
7.4.2	Verwendung von Locks —	202
7.4.3	Das <code>queue</code> -Modul —	205
7.5	Multicore-Programmierung in Python —	208
7.5.1	Das <code>multiprocessing</code> -Modul —	208
7.5.2	Das <code>mpi4py</code> -Modul —	211

Literatur — 219

Stichwortverzeichnis — 220