

Inhaltsverzeichnis

1	Einführung	1
1.1	Einleitung	1
1.2	Installation von Anaconda	1
1.3	Verwenden von Jupyter Notebooks	2
1.4	Spezielle Eigenschaften von Python	4
1.5	IPython	5
1.6	Was kann eine Programmiersprache?	5
2	Prozedurale Programmierung in Python	9
2.1	Der Datentyp <code>int</code>	9
2.2	Operatoren	10
2.3	Der Datentyp <code>float</code>	12
2.4	Der Datentyp <code>bool</code>	13
2.5	Der Datentyp <code>str</code>	15
2.5.1	String-Operatoren	17
2.5.2	Methodenschreibweise	18
2.5.3	<code>str</code> -Methoden	19
2.6	Der Datentyp <code>list</code>	23
2.6.1	<code>list</code> -Operatoren	25
2.6.2	<code>list</code> -Methoden	26
2.7	Referenzen	27
2.8	Der Datentyp <code>tuple</code>	29
2.9	Indizierung	30
2.9.1	Indizierungsart 1: Auswahl eines einzelnen Elements	31
2.9.2	Indizierungsart 1b: Indizierung mit negativer Zahl	32
2.9.3	Indizierungsart 2: Auswahl einer Teilsequenz (Slicing)	32
2.9.4	Indizierung eines Teilbereichs mit Schrittweite (Extended Slicing)	34
2.10	Vergleiche zusammengesetzter Typen	35
2.11	Kommandos zur Kontrollfluss-Steuerung	36
2.11.1	Einrückung	37

2.11.2	while -Schleifen	38
2.11.3	for -Schleifen	40
2.11.4	Das if -Kommando	42
2.12	Funktionen	44
2.12.1	Beliebig viele Parameter	48
2.12.2	Named Parameter vs. Positional Parameter	49
2.12.3	Scoping (Sichtbarkeitsregeln)	50
2.12.4	Weitere Aufgaben zu Funktionen	53
2.13	Kommando vs. Ausdruck	57
2.14	if -Ausdruck	60
2.15	Lazy Evaluation	60
2.16	Der Datentyp set	61
2.16.1	Verwendung	62
2.16.2	Iteration und Methoden	64
2.17	Der Datentyp dict	67
2.17.1	Dictionary-Operationen	68
2.17.2	Weitere Eigenschaften von dict	70
2.17.3	dict -Methoden	70
2.18	Weitere Aufgaben	73
3	Objektorientierte Programmierung	77
3.1	Was ist das Neue in der OO-Programmierung?	78
3.2	Definition einer Klasse	78
3.3	Aufruf einer Methode	79
3.4	Erstes Beispiel: Die Klasse Point	80
3.4.1	Konstruktor	80
3.4.2	Attribute	80
3.4.3	Die __init__ -Funktion und self	81
3.4.4	__init__ mit Parametern	82
3.4.5	Weitere Methoden der Klasse Point	83
3.4.6	String-Repräsentation eines Objekts	85
3.4.7	Implementierung von Operatoren	86
3.5	Zweites Beispiel: Die Klasse Auto	88
3.5.1	Methoden	88
3.5.2	Klassen-Attribute vs. Objekt-Attribute	89
3.5.3	Verbindungen zwischen Klassen	90
3.5.4	Vererbung: Eine Subclass von Auto	92
3.6	Drittes Beispiel: Die Klasse Queue	95
3.6.1	Methoden	96
3.7	Viertes Beispiel: Die Klasse Counter	98
3.7.1	Erster Ansatz: dict -Attribut	98
3.7.2	Methoden	99
3.7.3	Indizierung	100
3.7.4	Zweiter Ansatz: Entwurf durch Vererbung von dict	101
3.8	Fünftes Beispiel: EventL – Vererbung von einer Basisklasse	102

3.9	Sechstes Beispiel: Bäume	105
3.9.1	graphviz	105
3.9.2	Binäre Suchbäume	106
3.9.3	Repräsentation in Python	108
3.9.4	Die Klasse <code>BTree</code>	108
3.9.5	Zeichnen eines Baumes	109
3.9.6	Einfügen	111
3.10	Weitere Aufgaben	113
4	Funktionale Programmierung	123
4.1	Ausdrücke vs. Kommandos	123
4.1.1	Kommando- <code>if</code> vs. Ausdrucks- <code>if</code>	124
4.2	Funktionsausdrücke	125
4.3	Die <code>map</code> -Funktion	126
4.3.1	Weitere <code>map</code> -Beispiele aus der Welt der Zeichen	128
4.4	Die <code>filter</code> -Funktion	131
4.5	<code>any</code> und <code>all</code>	132
4.6	List-Comprehensions	133
4.6.1	Suche im Dateisystem	136
4.7	Die <code>enumerate</code> -Funktion	137
4.8	Die <code>reduce</code> -Funktion	138
4.9	Weitere Aufgaben	141
5	Numpy	149
5.1	Numpy Arrays: Der Datentyp <code>ndarray</code>	149
5.2	Vektorisierung	150
5.2.1	Vektorisierte Funktionsauswertung und Plotting	153
5.3	Fehlende Werte: <code>np.nan</code>	153
5.4	Indizierung	154
5.4.1	Indizierung mit Skalaren Werten	155
5.4.2	Bereichsindizierung	155
5.4.3	Indizierung mit einer Sequenz	155
5.4.4	Indizierung mit Booleschem Array	155
5.5	Zuweisungen	157
5.6	Kummulative Operationen	158
5.7	Matrizen	159
5.7.1	Indizierung von Matrizen	160
5.8	Anwendung einer Funktion entlang von Achsen	161
5.9	Broadcasting	163
5.10	2-d Plotting mit <code>meshgrid</code>	164
5.11	Weitere Aufgaben	167
6	Lösungen	169
	Index	193