

**Michael Inden**

# **Python Challenge**

**Fit für Prüfung, Job-Interview und Praxis  
– mit 100 Aufgaben und Musterlösungen**

 **dpunkt.verlag**

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
1.1	Aufbau der Kapitel .....	1
1.2	Grundgerüst des PyCharm-Projekts .....	3
1.3	Grundgerüst für die Unit Tests mit PyTest .....	4
1.4	Anmerkung zum Programmierstil .....	5
1.5	Anmerkung zu den Aufgaben .....	9
1.6	Ausprobieren der Beispiele und Lösungen .....	9
<b>I</b>	<b>Grundlagen</b> .....	<b>11</b>
<b>2</b>	<b>Mathematische Aufgaben</b> .....	<b>13</b>
2.1	Einführung .....	13
2.1.1	Römische Zahlen .....	17
2.1.2	Zahlenspielereien .....	18
2.2	Aufgaben .....	21
2.2.1	Aufgabe 1: Grundrechenarten (★☆☆☆☆) .....	21
2.2.2	Aufgabe 2: Zahl als Text (★☆☆☆☆) .....	22
2.2.3	Aufgabe 3: Vollkommene Zahlen (★☆☆☆☆) .....	22
2.2.4	Aufgabe 4: Primzahlen (★☆☆☆☆) .....	23
2.2.5	Aufgabe 5: Primzahlpaare (★☆☆☆☆) .....	23
2.2.6	Aufgabe 6: Prüfsumme (★☆☆☆☆) .....	23
2.2.7	Aufgabe 7: Römische Zahlen (★★★★☆) .....	24
2.2.8	Aufgabe 8: Kombinatorik (★☆☆☆☆) .....	24
2.2.9	Aufgabe 9: Armstrong-Zahlen (★☆☆☆☆) .....	25
2.2.10	Aufgabe 10: Max Change Calculator (★★★★☆) .....	25
2.2.11	Aufgabe 11: Befreundete Zahlen (★☆☆☆☆) .....	26
2.2.12	Aufgabe 12: Primfaktorzerlegung (★☆☆☆☆) .....	26
2.3	Lösungen .....	27
2.3.1	Lösung 1: Grundrechenarten (★☆☆☆☆) .....	27
2.3.2	Lösung 2: Zahl als Text (★☆☆☆☆) .....	29
2.3.3	Lösung 3: Vollkommene Zahlen (★☆☆☆☆) .....	31
2.3.4	Lösung 4: Primzahlen (★☆☆☆☆) .....	33

2.3.5	Lösung 5: Primzahlpaare (★★☆☆☆)	35
2.3.6	Lösung 6: Prüfsumme (★★☆☆☆)	39
2.3.7	Lösung 7: Römische Zahlen (★★★★☆)	40
2.3.8	Lösung 8: Kombinatorik (★★☆☆☆)	43
2.3.9	Lösung 9: Armstrong-Zahlen (★★☆☆☆)	46
2.3.10	Lösung 10: Max Change Calculator (★★★★☆)	49
2.3.11	Lösung 11: Befreundete Zahlen (★★☆☆☆)	50
2.3.12	Lösung 12: Primfaktorzerlegung (★★☆☆☆)	52
<b>3</b>	<b>Rekursion</b>	<b>55</b>
3.1	Einführung	55
3.1.1	Mathematische Beispiele	55
3.1.2	Algorithmische Beispiele	59
3.1.3	Typische Probleme: Endlose Aufrufe und <code>RecursionError</code>	64
3.2	Aufgaben	66
3.2.1	Aufgabe 1: Fibonacci (★★☆☆☆)	66
3.2.2	Aufgabe 2: Ziffern verarbeiten (★★☆☆☆)	66
3.2.3	Aufgabe 3: ggT / GCD (★★☆☆☆)	67
3.2.4	Aufgabe 4: Reverse String (★★☆☆☆)	68
3.2.5	Aufgabe 5: Array Sum (★★☆☆☆)	68
3.2.6	Aufgabe 6: Array Min (★★☆☆☆)	68
3.2.7	Aufgabe 7: Konvertierungen (★★☆☆☆)	69
3.2.8	Aufgabe 8: Exponentialfunktion (★★☆☆☆)	70
3.2.9	Aufgabe 9: Pascal'sches Dreieck (★★☆☆☆)	71
3.2.10	Aufgabe 10: Zahlenpalindrome (★★★★☆)	71
3.2.11	Aufgabe 11: Permutationen (★★★★☆)	72
3.2.12	Aufgabe 12: Count Substrings (★★☆☆☆)	72
3.2.13	Aufgabe 13: Lineal (★★☆☆☆)	73
3.3	Lösungen	74
3.3.1	Lösung 1: Fibonacci (★★☆☆☆)	74
3.3.2	Lösung 2: Ziffern verarbeiten (★★☆☆☆)	76
3.3.3	Lösung 3: ggT / GCD (★★☆☆☆)	78
3.3.4	Lösung 4: Reverse String (★★☆☆☆)	80
3.3.5	Lösung 5: Array Sum (★★☆☆☆)	81
3.3.6	Lösung 6: Array Min (★★☆☆☆)	83
3.3.7	Lösung 7: Konvertierungen (★★☆☆☆)	84
3.3.8	Lösung 8: Exponentialfunktion (★★☆☆☆)	87
3.3.9	Lösung 9: Pascal'sches Dreieck (★★☆☆☆)	90
3.3.10	Lösung 10: Zahlenpalindrome (★★★★☆)	93
3.3.11	Lösung 11: Permutationen (★★★★☆)	96
3.3.12	Lösung 12: Count Substrings (★★☆☆☆)	99
3.3.13	Lösung 13: Lineal (★★☆☆☆)	102

<b>4</b>	<b>Strings</b> .....	<b>105</b>
4.1	Einführung .....	105
4.2	Aufgaben .....	111
4.2.1	Aufgabe 1: Zahlenumwandlungen (★★☆☆☆) .....	111
4.2.2	Aufgabe 2: Joiner (★☆☆☆☆) .....	111
4.2.3	Aufgabe 3: Reverse String (★★☆☆☆) .....	112
4.2.4	Aufgabe 4: Palindrom (★★★☆☆) .....	112
4.2.5	Aufgabe 5: No Duplicate Chars (★★★☆☆) .....	113
4.2.6	Aufgabe 6: Doppelte Buchstaben entfernen (★★★☆☆) .....	113
4.2.7	Aufgabe 7: Capitalize (★☆☆☆☆) .....	114
4.2.8	Aufgabe 8: Rotation (★☆☆☆☆) .....	115
4.2.9	Aufgabe 9: Wohlgeformte Klammern (★☆☆☆☆) .....	115
4.2.10	Aufgabe 10: Anagramm (★☆☆☆☆) .....	116
4.2.11	Aufgabe 11: Morse Code (★☆☆☆☆) .....	116
4.2.12	Aufgabe 12: Pattern Checker (★☆☆☆☆) .....	117
4.2.13	Aufgabe 13: Tennis-Punktstand (★★★☆☆) .....	117
4.2.14	Aufgabe 14: Versionsnummern (★☆☆☆☆) .....	118
4.2.15	Aufgabe 15: Konvertierung <code>str_to_number</code> (★☆☆☆☆) .....	118
4.2.16	Aufgabe 16: Print Tower (★★★☆☆) .....	119
4.2.17	Aufgabe 17: Gefüllter Rahmen (★☆☆☆☆) .....	119
4.2.18	Aufgabe 18: Vokale raten (★☆☆☆☆) .....	119
4.3	Lösungen .....	120
4.3.1	Lösung 1: Zahlenumwandlungen (★☆☆☆☆) .....	120
4.3.2	Lösung 2: Joiner (★☆☆☆☆) .....	123
4.3.3	Lösung 3: Reverse String (★☆☆☆☆) .....	124
4.3.4	Lösung 4: Palindrom (★★★☆☆) .....	126
4.3.5	Lösung 5: No Duplicate Chars (★★★☆☆) .....	129
4.3.6	Lösung 6: Doppelte Buchstaben entfernen (★★★☆☆) .....	131
4.3.7	Lösung 7: Capitalize (★☆☆☆☆) .....	132
4.3.8	Lösung 8: Rotation (★☆☆☆☆) .....	136
4.3.9	Lösung 9: Wohlgeformte Klammern (★☆☆☆☆) .....	137
4.3.10	Lösung 10: Anagramm (★☆☆☆☆) .....	139
4.3.11	Lösung 11: Morse Code (★☆☆☆☆) .....	140
4.3.12	Lösung 12: Pattern Checker (★☆☆☆☆) .....	142
4.3.13	Lösung 13: Tennis-Punktstand (★★★☆☆) .....	144
4.3.14	Lösung 14: Versionsnummern (★☆☆☆☆) .....	147
4.3.15	Lösung 15: Konvertierung <code>str_to_number</code> (★☆☆☆☆) .....	148
4.3.16	Lösung 16: Print Tower (★★★☆☆) .....	151
4.3.17	Lösung 17: Gefüllter Rahmen (★☆☆☆☆) .....	153
4.3.18	Lösung 18: Vokale raten (★☆☆☆☆) .....	154

<b>5</b>	<b>Basisdatenstrukturen: Listen, Sets und Dictionaries .....</b>	<b>157</b>
5.1	Einführung .....	157
5.1.1	Sequenzielle Datentypen .....	157
5.1.2	Listen .....	159
5.1.3	Mengen (Sets).....	163
5.1.4	Schlüssel-Wert-Abbildungen (Dictionaries) .....	164
5.1.5	Der Stack als LIFO-Datenstruktur .....	166
5.1.6	Die Queue als FIFO-Datenstruktur .....	167
5.2	Aufgaben .....	171
5.2.1	Aufgabe 1: Gemeinsame Elemente (★★☆☆☆).....	171
5.2.2	Aufgabe 2: Eigener Stack (★★☆☆☆) .....	171
5.2.3	Aufgabe 3: List Reverse (★★☆☆☆) .....	171
5.2.4	Aufgabe 4: Duplikate entfernen (★★☆☆☆) .....	172
5.2.5	Aufgabe 5: Maximaler Gewinn (★★★☆☆) .....	172
5.2.6	Aufgabe 6: Längstes Teilstück (★★★☆☆) .....	173
5.2.7	Aufgabe 7: Wohlgeformte Klammern (★★☆☆☆) .....	173
5.2.8	Aufgabe 8: Pascal'sches Dreieck (★★★☆☆).....	174
5.2.9	Aufgabe 9: Check Magic Triangle (★★★☆☆) .....	174
5.2.10	Aufgabe 10: Häufigste Elemente (★★☆☆☆) .....	175
5.2.11	Aufgabe 11: Addition von Ziffern (★★★☆☆) .....	175
5.2.12	Aufgabe 12: List Merge (★★☆☆☆) .....	176
5.2.13	Aufgabe 13: Excel Magic Select (★★☆☆☆) .....	176
5.2.14	Aufgabe 14: Stack Based Queue (★★☆☆☆).....	177
5.3	Lösungen .....	178
5.3.1	Lösung 1: Gemeinsame Elemente (★★☆☆☆) .....	178
5.3.2	Lösung 2: Eigener Stack (★★☆☆☆) .....	180
5.3.3	Lösung 3: List Reverse (★★☆☆☆) .....	181
5.3.4	Lösung 4: Duplikate entfernen (★★☆☆☆) .....	184
5.3.5	Lösung 5: Maximaler Gewinn (★★★☆☆).....	186
5.3.6	Lösung 6: Längstes Teilstück (★★★☆☆) .....	188
5.3.7	Lösung 7: Wohlgeformte Klammern (★★☆☆☆) .....	190
5.3.8	Lösung 8: Pascal'sches Dreieck (★★★☆☆).....	194
5.3.9	Lösung 9: Check Magic Triangle (★★★☆☆) .....	196
5.3.10	Lösung 10: Häufigste Elemente (★★☆☆☆) .....	199
5.3.11	Lösung 11: Addition von Ziffern (★★★☆☆) .....	200
5.3.12	Lösung 12: List Merge (★★☆☆☆) .....	204
5.3.13	Lösung 13: Excel Magic Select (★★☆☆☆) .....	208
5.3.14	Lösung 14: Stack Based Queue (★★☆☆☆) .....	210

<b>6</b>	<b>Arrays</b>	<b>213</b>
6.1	Einführung	213
6.1.1	Eindimensionale Arrays	214
6.1.2	Mehrdimensionale Arrays	222
6.1.3	Typische Fehler	227
6.1.4	Besonderheiten	228
6.1.5	Rekapitulation: NumPy	229
6.2	Aufgaben	235
6.2.1	Aufgabe 1: Gerade vor ungeraden Zahlen (★★☆☆☆)	235
6.2.2	Aufgabe 2: Flip (★★☆☆☆)	235
6.2.3	Aufgabe 3: Palindrom (★★☆☆☆)	235
6.2.4	Aufgabe 4: Inplace Rotate (★★★☆☆)	236
6.2.5	Aufgabe 5: Jewels Board Init (★★★☆☆)	236
6.2.6	Aufgabe 6: Jewels Board Erase Diamonds (★★★★☆)	238
6.2.7	Aufgabe 7: Spiral-Traversal (★★★★☆)	239
6.2.8	Aufgabe 8: Add One to Array As Number (★★☆☆☆)	239
6.2.9	Aufgabe 9: Sudoku-Checker (★★★☆☆)	240
6.2.10	Aufgabe 10: Flood-Fill (★★☆☆☆)	241
6.2.11	Aufgabe 11: Array Min und Max (★★☆☆☆)	242
6.2.12	Aufgabe 12: Array Split (★★★☆☆)	243
6.2.13	Aufgabe 13: Minesweeper Board (★★★★☆)	244
6.3	Lösungen	246
6.3.1	Lösung 1: Gerade vor ungeraden Zahlen (★★☆☆☆)	246
6.3.2	Lösung 2: Flip (★★☆☆☆)	250
6.3.3	Lösung 3: Palindrom (★★☆☆☆)	253
6.3.4	Lösung 4: Inplace Rotate (★★★☆☆)	255
6.3.5	Lösung 5: Jewels Board Init (★★★☆☆)	259
6.3.6	Lösung 6: Jewels Board Erase Diamonds (★★★★☆)	265
6.3.7	Lösung 7: Spiral-Traversal (★★★★☆)	273
6.3.8	Lösung 8: Add One to Array As Number (★★☆☆☆)	277
6.3.9	Lösung 9: Sudoku-Checker (★★★☆☆)	278
6.3.10	Lösung 10: Flood-Fill (★★☆☆☆)	283
6.3.11	Lösung 11: Array Min und Max (★★☆☆☆)	287
6.3.12	Lösung 12: Array Split (★★★☆☆)	290
6.3.13	Lösung 13: Minesweeper Board (★★★★☆)	294

<b>II Fortgeschrittenere und kniffligere Themen</b>	<b>301</b>
<b>7 Rekursion Advanced</b>	<b>303</b>
7.1 Memoization	303
7.1.1 Memoization für Fibonacci-Zahlen	303
7.1.2 Memoization für Pascal'sches Dreieck	305
7.1.3 Memoization mit Python-Bordmitteln	307
7.2 Backtracking	311
7.2.1 n-Damen-Problem	311
7.3 Aufgaben	315
7.3.1 Aufgabe 1: Türme von Hanoi (★★★☆☆)	315
7.3.2 Aufgabe 2: Edit Distance (★★★★☆)	316
7.3.3 Aufgabe 3: Longest Common Subsequence (★★★☆☆)	316
7.3.4 Aufgabe 4: Weg aus Labyrinth (★★★☆☆)	317
7.3.5 Aufgabe 5: Sudoku-Solver (★★★★☆)	318
7.3.6 Aufgabe 6: Math Operator Checker (★★★★☆)	319
7.3.7 Aufgabe 7: Wassereimer-Problem (★★★☆☆)	320
7.3.8 Aufgabe 8: Alle Palindrom-Teilstrings (★★★★☆)	321
7.3.9 Aufgabe 9: n-Damen-Problem (★★★☆☆)	321
7.4 Lösungen	322
7.4.1 Lösung 1: Türme von Hanoi (★★★☆☆)	322
7.4.2 Lösung 2: Edit Distance (★★★★☆)	327
7.4.3 Lösung 3: Longest Common Subsequence (★★★☆☆)	330
7.4.4 Lösung 4: Weg aus Labyrinth (★★★☆☆)	334
7.4.5 Lösung 5: Sudoku-Solver (★★★★☆)	337
7.4.6 Lösung 6: Math Operator Checker (★★★★☆)	344
7.4.7 Lösung 7: Wassereimer-Problem (★★★★☆)	347
7.4.8 Lösung 8: Alle Palindrom-Teilstrings (★★★★☆)	350
7.4.9 Lösung 9: n-Damen-Problem (★★★☆☆)	354
<b>8 Binärbäume</b>	<b>361</b>
8.1 Einführung	361
8.1.1 Aufbau, Begrifflichkeiten und Anwendungsbeispiele	361
8.1.2 Binärbäume	362
8.1.3 Binärbäume mit Ordnung: binäre Suchbäume	363
8.1.4 Traversierungen	365
8.1.5 Balancierte Bäume und weitere Eigenschaften	367
8.1.6 Bäume für die Beispiele und Übungsaufgaben	369
8.2 Aufgaben	371
8.2.1 Aufgabe 1: Tree Traversal (★★☆☆☆)	371
8.2.2 Aufgabe 2: In-, Pre- und Postorder iterativ (★★★★☆)	371
8.2.3 Aufgabe 3: Tree-Höhe berechnen (★★☆☆☆)	371
8.2.4 Aufgabe 4: Kleinster gemeinsamer Vorfahre (★★★☆☆)	372

8.2.5	Aufgabe 5: Breadth-First (★★★★☆☆)	372
8.2.6	Aufgabe 6: Level Sum (★★★★☆☆)	373
8.2.7	Aufgabe 7: Tree Rotate (★★★★☆☆)	373
8.2.8	Aufgabe 8: Rekonstruktion (★★★★☆☆)	374
8.2.9	Aufgabe 9: Math Evaluation (★★☆☆☆☆)	374
8.2.10	Aufgabe 10: Symmetrie (★★☆☆☆☆)	375
8.2.11	Aufgabe 11: Check Binary Search Tree (★★☆☆☆☆)	376
8.2.12	Aufgabe 12: Vollständigkeit (★★★★★)	376
8.2.13	Aufgabe 13: Tree Printer (★★★★★)	378
8.3	Lösungen	381
8.3.1	Lösung 1: Tree Traversal (★★☆☆☆☆)	381
8.3.2	Lösung 2: In-, Pre- und Postorder iterativ (★★★★☆☆)	383
8.3.3	Lösung 3: Tree-Höhe berechnen (★★☆☆☆☆)	390
8.3.4	Lösung 4: Kleinster gemeinsamer Vorfahre (★★★★☆☆)	391
8.3.5	Lösung 5: Breadth-First (★★★★☆☆)	394
8.3.6	Lösung 6: Level Sum (★★★★☆☆)	396
8.3.7	Lösung 7: Tree Rotate (★★★★☆☆)	399
8.3.8	Lösung 8: Rekonstruktion (★★★★☆☆)	402
8.3.9	Lösung 9: Math Evaluation (★★☆☆☆☆)	407
8.3.10	Lösung 10: Symmetrie (★★☆☆☆☆)	408
8.3.11	Lösung 11: Check Binary Search Tree (★★☆☆☆☆)	413
8.3.12	Lösung 12: Vollständigkeit (★★★★★)	415
8.3.13	Lösung 13: Tree Printer (★★★★★)	423
<b>9</b>	<b>Suchen und Sortieren</b>	<b>433</b>
9.1	Einführung Suchen	433
9.1.1	Binärsuche	435
9.2	Einführung Sortieren	436
9.2.1	Insertion Sort	436
9.2.2	Selection Sort	438
9.2.3	Merge Sort	440
9.2.4	Quick Sort	441
9.2.5	Bucket Sort	443
9.2.6	Schlussgedanken	444
9.3	Aufgaben	445
9.3.1	Aufgabe 1: Contains All (★★☆☆☆☆)	445
9.3.2	Aufgabe 2: Partitionierung (★★★★☆☆)	445
9.3.3	Aufgabe 3: Binärsuche (★★☆☆☆☆)	446
9.3.4	Aufgabe 4: Insertion Sort (★★☆☆☆☆)	446
9.3.5	Aufgabe 5: Selection Sort (★★☆☆☆☆)	447
9.3.6	Aufgabe 6: Quick Sort (★★☆☆☆☆)	447
9.3.7	Aufgabe 7: Bucket Sort (★★☆☆☆☆)	448
9.3.8	Aufgabe 8: Suche in rotierten Daten (★★★★☆☆)	448



9.4	Lösungen .....	450
9.4.1	Lösung 1: Contains All (★★☆☆☆) .....	450
9.4.2	Lösung 2: Partitionierung (★★★★☆) .....	451
9.4.3	Lösung 3: Binärsuche (★★☆☆☆) .....	453
9.4.4	Lösung 4: Insertion Sort (★★☆☆☆) .....	456
9.4.5	Lösung 5: Selection Sort (★★☆☆☆) .....	457
9.4.6	Lösung 6: Quick Sort (★★★★☆) .....	458
9.4.7	Lösung 7: Bucket Sort (★★☆☆☆) .....	460
9.4.8	Lösung 8: Suche in rotierten Daten (★★★★☆) .....	461
<b>10</b>	<b>Schlusswort und ergänzende Literatur .....</b>	<b>467</b>
10.1	Schlusswort .....	467
10.1.1	Gelerntes pro Kapitel .....	467
10.1.2	Bedenkenswertes .....	469
10.2	Knobelaufgaben .....	471
10.2.1	Goldsäcke – Fälschung entdecken .....	471
10.2.2	Pferderennen – schnellste drei Pferde ermitteln .....	472
10.3	Ergänzende Literatur .....	475
<b>III</b>	<b>Anhang .....</b>	<b>479</b>
<b>A</b>	<b>Kurzeinführung Pytest .....</b>	<b>481</b>
A.1	Schreiben und Ausführen von Tests .....	481
A.1.1	Installation von Pytest .....	481
A.1.2	Beispiel: Ein erster Unit Test .....	482
A.1.3	Ausführen von Tests .....	482
A.1.4	Behandlung erwarteter Exceptions .....	484
A.1.5	Parametrisierte Tests mit Pytest .....	485
A.2	Weiterführende Literatur zu Pytest .....	486
<b>B</b>	<b>Kurzeinführung Dekoratoren .....</b>	<b>487</b>
<b>C</b>	<b>Schnelleinstieg O-Notation .....</b>	<b>493</b>
C.1	Abschätzungen mit der O-Notation .....	493
C.1.1	Komplexitätsklassen .....	494
C.1.2	Komplexität und Programmlaufzeit .....	496
	<b>Literaturverzeichnis .....</b>	<b>497</b>
	<b>Index .....</b>	<b>499</b>