

Python Programmierung für Fortgeschrittene

Seminarunterlage

Version: 2.01

Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG
Karl-Schurz-Straße 19a
D-33100 Paderborn
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
An der alten Ziegelei 5
D-48157 Münster
Tel.: (+49) 02 51 / 9 24 35 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Welser Straße 9
D-86368 Gersthofen
Tel.: (+49) 08 21 / 507 492 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Kreuzberger Ring 13
D-65205 Wiesbaden
Tel.: (+49) 06 11 / 7 78 40 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Wikingerstraße 18-20
D-51107 Köln
Tel.: (+49) 02 21 / 8 70 61 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Südwestpark 67/2
D-90449 Nürnberg
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

Internet: <http://www.ordix.de>

Email: seminare@ordix.de

Inhaltsverzeichnis

1	Entwicklungsumgebung: Dependency Management	10
1.1	Einleitung	11
1.2	Tool Übersicht	12
1.3	Python Package Installer	13
1.4	Welche Abhängigkeiten hat ein Projekt?	14
1.5	Problem: unterschiedliche Anforderungen	15
1.6	Lösung: unterschiedliche Anforderungen	16
1.7	Pipenv	17
1.7.1	pipenv – shell	18
1.7.2	pipenv – Einrichtung der Umgebung	19
1.7.3	pipenv – Dependency Graph	20
1.7.4	Übersicht: Pipfile / Pipfile.lock / pipenv graph	21
1.7.5	pipenv install – Advanced	22
1.7.6	pipenv – dev	23
1.8	Zusammenfassung	24
2	Module vs. Packages	25
2.1	Rückblick: Warum ein Projekt modular aufbauen?	26
2.2	Rückblick: Modularität in Python	27
2.3	Rückblick: Module	28
2.4	Packages in Python	29
2.4.1	Packages in Python: Import	30
2.4.2	Packages in Python: From ... import	31
2.4.3	Packages in Python: __init__.py (1/2)	32
2.4.4	Packages in Python: __init__.py (2/2)	33
2.4.5	Packages in Python: Aufruf	34
2.4.6	Packages in Python: Zusammenfassung	35
3	Exceptions	36
3.1	Motivation	37
3.2	Laufzeitfehler vs. syntaktische Fehler	38
3.3	Exceptions gezielt "werfen" – raise	39
3.4	Exceptions gezielt "werfen" – assert	40
3.5	Exceptions abfangen – try and except)	41
3.6	Exceptiontyp – Vordefinierte Fehler	43
3.7	Exceptiontyp – Eigene Exceptions	44
3.8	Exceptions ans Betriebssystem weitergeben	45
4	Argumente Parsen	46
4.1	Motivation	47
4.2	sys.argv	48
4.3	Motivation	49
4.4	Grundlegendes	50
4.5	Beispiel	51
4.6	Erstellen eines Parser-Objektes	52
4.7	Argument – Namen	55
4.7.1	Argument – Datentyp	56
4.7.2	Argument – Optional vs. Required	57
4.7.3	Argument als Schalter	58
4.7.4	Argument mit mehreren Parametern	59
4.7.5	Argument mit mehreren Optionen	60
4.7.6	Argument – Hilfetext	61
4.7.7	add_argument – Die wichtigsten Argumente	62
4.8	Mutually Exclusive Groups	63
4.9	Weiterführende Inhalte	64
4.10	Sub Commands – Motivation	65
4.10.1	Sub Commands	67
4.10.2	Sub Commands Auswertung	68

4.10.3	Sub Commands.....	69
5	Logging	70
5.1	Motivation.....	71
5.2	Beispiel – Basic Logging.....	72
5.3	Log-Level.....	73
5.4	Basic Config.....	74
5.4.1	Basic Config – Parameter	75
5.4.2	Basic Config – Format.....	76
5.5	Logging – Exceptions.....	77
5.6	Logging / Logger	78
5.7	Weiterführende Inhalte.....	79
5.8	Logging mit Logger	80
6	pydoc.....	81
6.1	Motivation.....	82
6.2	Aufruf der Hilfe in Python	83
6.3	Hilfetexte erstellen.....	84
6.4	Aufruf der Hilfe in Python	86
6.5	Aufruf der Hilfe im Browser	87
6.6	Python Dokumentation.....	88
6.7	Beispiel reStructuredText.....	89
7	pylint.....	90
7.1	PEP8 – Style Guide for Python Code	91
7.2	Installation	92
7.2.1	Ein einfaches Beispiel	93
7.3	Konventionen	95
7.3.1	Ein einfaches Beispiel	96
7.4	Konventionen	97
7.4.1	Ein einfaches Beispiel	98
7.5	Namenskonventionen	99
7.5.1	Ein einfaches Beispiel	100
7.6	Meldungen unterdrücken	101
7.7	Ausgabe	102
7.8	Und nun ein komplexes Beispiel.....	103
7.9	Fazit.....	104
8	Testen mit pytest.....	105
8.1	Motivation.....	106
8.2	Installation	107
8.3	Ein einfacher Test	108
8.4	Erfolgreiche Test.....	109
8.5	Tests die Fehlschlagen	110
8.6	Exceptions.....	111
8.7	Aufruf pytest	114
8.8	Ausgabe pytest	115
8.9	Ausgabe pytest – Fehlermeldungen	116
8.10	Parametrisierte Tests	117
8.11	Testablauf – ohne Hilfsmittel.....	119
8.12	Testablauf – Setup mit Fixture	120
8.13	Testablauf – Teardown mit Fixture	121
8.14	Fixtures – Dokumentation	122
8.15	Parametrisierte Fixtures	123
8.16	Zusammenfassung.....	124
8.17	Fazit.....	125
9	Objektorientierung	126
9.1	Überblick	127
9.2	Klasse vs. Objekt	128

9.3	Klasse.....	129
9.4	Klasse – Getter / Setter.....	130
9.5	Special Functions.....	131
9.6	Ausgabe mit <code>__repr__</code>	132
9.7	Ausgabe mit <code>__str__</code>	133
9.8	String-Repräsentation von Objekten.....	134
9.9	Bedeutung von <code>self</code>	135
9.10	Special Functions – Comparison.....	137
9.10.1	Comparison Objekt – Default.....	138
9.10.2	Objekt Comparison – Implementiert.....	139
9.10.3	Objekt Comparison – Fehler.....	140
9.11	Special Functions – weitere Gruppen.....	141
9.12	Private Attribute / Eigene Methoden.....	142
9.13	Private Attribute / Eigene Methoden (ff).....	143
9.14	Special Functions – Teil 2.....	144
9.14.1	Special Function <code>__getitem__</code>	145
9.14.2	Special Functions – Teil 2.....	146
9.14.3	Special Function <code>__getitem__</code>	147
9.14.4	Operator in.....	148
9.15	Context Manager.....	149
9.16	Vererbung / Subclass.....	150
9.16.1	Vererbung.....	151
9.16.2	Vererbung – <code>isinstance()</code>	152
9.16.3	Vererbung – Method Resolution Order.....	153
9.16.4	Vererbung – <code>mro()</code>	154
9.16.5	Vererbung – <code>super()</code>	155
9.17	Class Variables.....	156
9.18	Special Function <code>__del__</code>	157
9.19	Static Methods / Class Methods.....	158
9.20	Static Method.....	159
9.21	Class Method.....	160
10	Iterable Iterator Generator.....	161
10.1	Was ist was?.....	162
10.2	Zielsetzung.....	163
10.3	Iterable: Voraussetzung.....	164
10.3.1	Iterable: Beispiel.....	165
10.3.2	Iterable: weitere Eigenschaften.....	166
10.4	Was ist was?.....	167
10.5	Iterator.....	168
10.5.1	Iterator Implementierung.....	169
10.5.2	Iterator: Beispiel.....	170
10.6	Iterable vs. Iterator.....	171
10.7	Was ist was?.....	172
10.8	Generator.....	173
10.9	Was ist was?.....	174
10.10	Iterator Funktionen.....	175
10.10.1	Funktion <code>zip</code>	176
10.10.2	Funktion <code>map</code>	177
10.10.3	Funktion <code>sum</code>	178
10.11	Was ist was?.....	179
10.12	Itertools.....	180
11	Generatoren.....	183
11.1	Generatoren.....	184
11.2	Generatoren – Beispiel.....	185
11.3	Generatoren – Einführung.....	186
11.4	Generatoren – Funktion.....	188
11.5	Generatoren – Arbeitsweise.....	189
11.6	Generator – Expressions.....	190

11.7	Generator – mehrere yield	191
12	@Decorator	192
12.1	Decorator – Einführung	193
12.2	Higher Order Function	194
12.3	Parameter Definition – Parameterliste	195
12.4	Parameter Definition – Keyword Liste	196
12.5	Higher Order Functions – mit Parametern	197
12.6	Subfunction	198
12.7	Rückgabe einer Funktion	199
12.8	Decorator – Version 1	200
12.9	Decorator – Version 2	201
12.10	Decorator – Version 3	202
12.11	Decorator – Version 4	203
12.12	Decorator – Version 4 – Problem	204
12.13	Decorator – Version 4 – Lösung	205
12.14	Decorator – vollständiges Beispiel	206
12.15	Decorator	207
13	Datetime	208
13.1	Motivation	209
13.2	Datum Datentypen	210
13.3	Date	211
13.3.1	date – Verwendung der Konstruktoren	212
13.3.2	Formatierung von Zeitdatentypen	213
13.4	time	214
13.4.1	time – Verwendung der Konstruktoren	215
13.5	datetime	216
13.5.1	datetime – Verwendung der Konstruktoren	217
13.6	timedelta	219
13.6.1	timedelta – Konstruktor	220
13.6.2	timedelta – Verwendung des Konstruktors	221
13.7	Rechnen mit Date-Datentypen	222
13.8	Naive/bewusste Objekte	223
13.9	Timezone	224
13.9.1	timezone – Verwendung des Konstruktors	225
13.9.2	Timezone – Module pytz	226
13.9.3	Timezone – Übersetzung von Zeiten	227
13.10	Modul calendar	228
14	SQLAlchemy	229
14.1	Agenda	230
14.2	SQL	231
14.3	CREATE/DROP DATABASE	232
14.4	CREATE TABLE	233
14.5	INSERT	234
14.6	SELECT	235
14.7	DELETE	236
14.8	Commit	237
14.9	Übersicht	238
14.10	SQLite	239
14.11	Übersicht	240
14.12	Datenbankverbindung SQLite	241
14.13	Datenbankverbindung MySQL	242
14.14	Das Cursor-Objekt – cursor()	243
14.15	CREATE TABLE / Datentypen SQLite	244
14.16	Einfügen und Löschen von Datensätzen – execute()	245
14.17	Abschließen der Transaktion	246
14.18	Ausführen eines Selects – execute()	247
14.19	Ergebnisse auswerten	248

14.20	Ergebnisse nacheinander abfragen und ausgeben	249
14.21	close().....	250
14.22	Weitere Informationen.....	251
14.23	Übung 1.....	252
14.24	SQLAlchemy	253
	14.24.1 Übersicht	254
	14.24.2 SQLAlchemy Core – Vorbereitung der Verbindung.....	255
14.25	Datentypen.....	256
14.26	Tabellen.....	257
14.27	Datenbankverbindung.....	258
14.28	Insert	259
14.29	Select	260
14.30	Delete	261
14.31	Übung 2.....	262
14.32	Übersicht	263
14.33	ORM	264
14.34	Kriterien für das Anlegen eines Mappers.....	265
14.35	Anlegen des Mappers	266
14.36	Definieren der Tabelle.....	267
14.37	Starten der Session.....	268
14.38	Einfügen von Daten.....	269
14.39	Abfragen von Daten	270
14.40	Filtering.....	271
14.41	Löschen von Daten	272
14.42	Session Zustände von Instanzen.....	273
14.43	Prüfen des Session Status.....	274
14.44	Übung 3.....	275
15	JSON.....	276
	15.1 Motivation.....	277
	15.2 JSON – Beispiel	278
	15.3 Serialisierung	279
	15.4 Deserialization.....	280
	15.5 Daten-Mapping.....	281
	15.6 Unterschiede load vs. dump.....	282
16	Python Webentwicklung mit Flask.....	283
	16.1 Idee	284
	16.2 Was ist Flask?.....	285
	16.3 Unterschiede zu Django.....	286
	16.4 Anwendungsbeispiele	287
	16.5 Installation und Import.....	288
	16.6 Die Funktion run()	289
	16.7 Routes.....	290
	16.7.1 Routes - Ausgabe.....	291
	16.7.2 Routes - Variablen.....	292
	16.8 Übung 1.....	293
	16.9 Templates.....	294
	16.9.1 Templates – Erstellen.....	295
	16.9.2 Templates – Zugriff	296
	16.9.3 Templates – Variablen	297
	16.10 Exkurs: Jinja 2.....	298
	16.11 Beiträge.....	299
	16.12 Exkurs: Jinja 2 – Ein weiteres Beispiel	300
	16.13 Struktur.....	301
	16.14 Layout.....	302
	16.15 Übung 2.....	303
	16.16 Was ist Flask-WTF?.....	304
	16.16.1 Wie funktioniert Flask-WTF?.....	305
	16.16.2 Wie nutzt man Flask-WTF?.....	306

16.17	FormFields	307
16.18	Nutzung von FormFields in Klassen	308
16.19	Klassen in Routes nutzen	309
16.20	Methoden GET und POST	310
16.21	request	311
16.22	Secret Key	312
16.23	Übung 3.....	313
16.24	Validatoren	314
	16.24.1 Validatoren – Nachrichten	315
	16.24.2 Validatoren - Schreibweise	316
	16.24.3 Validate_on_submit()	317
16.25	url_for() und redirect()	318
16.26	Flash-Nachricht	319
16.27	Übung 4.....	321
17	Python REST API mit Connexion.....	322
17.1	Was ist REST?.....	323
17.2	Was ist Connexion?	324
17.3	Was ist eine swagger-ui?	325
17.4	Installation und Import.....	326
17.5	Erstellen der API	327
17.6	Swagger yml	328
17.7	Das Back-End	330
17.8	Das Back-End - Beispiel	331
17.9	Hilfen und Zusammenfassung	332
18	Jupyter Notebook.....	334
18.1	Motivation	335
18.2	Installation mit pip	336
18.3	Installation mit conda	337
18.4	Das erste Notebook	338
18.5	Jupyter Notebook	339
19	Matplotlib.....	340
19.1	Motivation	341
19.2	Inhalt.....	342
19.3	Nicht Inhalt	343
19.4	Liniendiagramm Default	344
19.5	Mehrere Linien	345
19.6	Linien Styles.....	346
	19.6.1 Linien Styles – Farben.....	347
	19.6.2 Linien Attribute	348
	19.6.3 Linien Styles	349
19.7	Beschriftungen	350
19.8	Grid / Gitter.....	351
19.9	Styles.....	352
19.10	Antigravity – Easter Egg.....	353
19.11	Styles – Anti Gravity	354
19.12	Weiterführende Themen	355
20	Packages veröffentlichen	356
20.1	pip, pypi.org und pypiserver	357
20.2	Vorbereitung: Projektstruktur	358
20.3	Vorbereitung: Konfiguration mittels setup.py	359
20.4	Vorbereitung: Versionierung	360
20.5	Vorbereitung: Weitere Dateien hinzufügen	361
20.6	Packages bauen – setup.py ausführen.....	362
20.7	Packages hochladen – twine	363
20.8	Hilfreiche Tools	364

21	Float & Decimal.....	365
21.1	Datentyp Float.....	366
21.2	Unerwartete Effekte (1/3) – Rundungsfehler, Präzision	367
21.3	Unerwartete Effekte (3/3) – Runden	368
21.4	Interne Repräsentation / Struct.....	369
21.5	Spezialfälle	370
21.6	Erklärung – Rundungsfehler	371
21.7	Erklärung – Runden	372
21.8	Decimal / Konstruktor.....	373
21.9	Decimal – Runden.....	375
21.10	Genauigkeit	378
21.11	Traps	379
21.12	Zusammenfassung.....	382