# Anleitung zum Ausprobieren der Beispielprogramme / Musterlösungen des Buchs "Python Challenge"

Dieses Dokument beschreibt die Schritte, um das Begleitprojekt in pyCharm zu importieren, geeignet zu konfigurieren und benötigte Fremdbibliotheken einzubinden. Danach sollten Sie mit den Sourcen arbeiten und die Tests ausführen können.

© Michael Inden, 2021 Kontakt: michael\_inden@hotmail.com

# Voraussetzungen / Installationen

Sie benötigen zumindest ein installiertes Python 3 (idealerweise ein aktuelles Python 3.8 oder 3.9):.

• https://www.python.org/downloads/

Zur Arbeit mit bzw. zum Import des Projekts in PyCharm benötigen Sie eine aktuelle Version (die kostenfreie Community Edition ist ausreichend):

• https://www.jetbrains.com/de-de/pycharm/download

# Übungsaufgaben / Beispielprogramme "installieren"

Laden Sie das Projekt mit den Übungsaufgaben und Beispielen als ZIP-Archiv herunter und entpacken Sie dieses in ein beliebiges Verzeichnis. Danach kann ein Import in PyCharm erfolgen.

#### **Import in PyCharm**

Importieren Sie das Projekt in PyCharm. Das sollte in etwa wie folgt aussehen und gliedert sich in intro, solutions und util. In ersteren finden sich die Programme aus der jeweiligen Kapiteleinleitung, In solutions befinden sich die jeweiligen Lösungen. In Letzteren befindet sich ein Modul mit Hilfsfunktionalität.



#### **Import in Einzelschritten**

#### PyCharm direkt nach dem Start

Direkt nach dem Start können Sie über "Open" das vorgefertigte Python Challenge Projekt importieren.



#### PyCharm direkt nach dem Import

Nach dem Import zeigt PyCharm das Projekt ggf. geschlossen an. Drücken Sie dann auf "Project"



#### PyCharm-Konfiguration des Python-Interpreters

Im Projekt hatte ich einen Python-Interpreter mitgeliefert, klicken Sie auf den linken Link, um diesen zu aktivieren. ACHTUNG: Aus Gründen der Download-Größe habe ich den Interpreter entfernt. Nutzen Sie bitte einen beliebigen (aktuellen) Interpreter.



### Installation von Fremdbibliotheken in PyCharm

Nach dem erstmaligen Import müssen die Abhängigkeiten auf die externen Bibliotheken (pytest usw.) initialisiert werden. Wechseln Sie unten in das Terminal und geben dort die nachfolgend aufgelisteten Installationsanweisungen ein.

```
Terminal: Local 	imes Local (2) 	imes +
  (venv) Michaels-MBP-2:PythonChallenge michaeli$ open pytest-report.html
  (venv) Michaels-MBP-2:PythonChallenge michaeli$ pip install pytest
  Requirement already satisfied: pytest in ./venv/lib/python3.9/site-pack-ges (6.2.2)
  Requirement already satisfied: iniconfig in ./venv/lib/python3.9/site-packages (from pytest) (1.1.1)
  Requirement already satisfied: packaging in ./venv/lib/python3.9/site-packages (from pytest) (20.9)
 Requirement already satisfied: pluggy<1.0.0a1,>=0.12 in ./venv/lib/python3.9/site-packages (from pytest) (0.13.1)
 Requirement already satisfied: py>=1.8.2 in ./venv/lib/pyt>on3.9/site-packages (from pytest) (1.10.0)
Requirement already satisfied: toml in ./venv/lib/pyta3.9/site-packages (from pytest) (0.10.2)
*
  🗄 TODO 🕨 Run 🕕 Problems 🔮 Python Console 🗵 Terminal
    1.
            pip install pytest
    2.
            pip install pytest-html
    3.
            pip install pytest-json-report --upgrade
    4.
            pip install parameterized
    5.
            pip install numpy
```

ACHTUNG: Für Mac muss es pip3 statt pip heissen.

### Konfiguration von pytest als Standardtestrunner in PyCharm

Damit pytest zum Ausführen der Tests genutzt wird, müssen wir in PyCharm folgendes konfigurieren:

	Preferences	
Qr	Tools > Python Integrated Tools 🐵 For current project	Reset
	Daskaging	
Project: PythonChallenge	Packaging	
Project Interpreter	Package requirements file:	
Project Structure		
Build, Execution, Deployment	Pipenv	
Languages & Frameworks	Path to Pipenv executable:	
▼ Tools		
Web Browsers	Testing	
File Watchers	Default test runner: ovtest	-
External Tools		
Terminal 💿	Docstrings	
Database	Departules formation of Text	_
SSH Configurations	Docstring format: restructured fext	
SSH Terminal	🗹 Analyze Python code in docstrings	
Datalore 🗈	Render external documentation for stdlib	
Diagrams		
Diff & Merge	restructured lext	
External Documentation	Sphinx working directory:	<u></u>
Python Integrated Tools 🛛 🖻	Treat *txt files as reStructuredText	
Python Scientific		
Remote SSH External Tools		
Server Certificates		
Settings Repository		
Startup Tasks 💿		
► Tasks		
Vagrant 🐵		
2	Cancel Apply	ОК

Weitere Infos zur Integration in PyCharm liefert die folgende Webseite: https://www.jetbrains.com/help/pycharm/pytest.html#

### Ausführung von pytest

Um die Tests auszuführen, geben Sie python3 -m pytest im Terminal ein:

	Terminal: Local × Local (2) × +	¢ -	-
	tests/ch09_search_and_sort/ex04_insertion_sort_test.py .	[ 97%]	
	tests/ch09_search_and_sort/ex05_selection_sort_test.py .	[ 97%]	
	tests/ch09_search_and_sort/ex06_quick_sort_test.py	[ 97%]	
	tests/ch09_search_and_sort/ex07_bucket_sort_test.py	[ 98%]	
0	tests/ch09_search_and_sort/ex08_search_rotated_sorted_test.py	[100%]	
é			
k	(venv) Michaels-MBP-2:PythonChallenge michaeli\$ python3 -m pytest		
	🗄 TODO 🕨 Run 🚯 Problems 🍓 Python Console 🔯 Terminal	C Event Log	i

Eine Protokollierung und Aufbereitung als HTML erhält man durch python3 -m pytest -html=<report-filename>:

	Terminal: Local × Local (2) × +	\$	-
	tests/ch09_search_and_sort/ex05_selection_sort_test.py .	[ 97%]	
	tests/ch09_search_and_sort/ex06_quick_sort_test.py	[ 97%]	
	tests/ch09_search_and_sort/ex07_bucket_sort_test.py	[ 98%]	
	tests/ch89_search_and_sort/ex08_search_rotated_sorted_test.py	[100%]	
Favorites	<pre> generated html file: file:///Users/michaeli/PycharmProjects/PythonChallenge/pytest-report.html</pre>		
	Ξ TODO 🕨 Run 🕘 Problems 🚸 Python Conscle 🛛 🔯 Terminal	C Event Lo	)g

# Zum Öffnen geben Sie (unter Mac) folgendes ein open pytest-report.html:

pytest	t-report.htn	nl		
Report gene	rated on 15-Feb-2021 a	t 11:58:22 by pytest-html v3.1.1		
Environn	nent			
Packages	{"pluggy": "0.13.1", "py	": "1.10.0", "pytest": "6.2.2"}		
Platform	macOS-10.14.6-x86_6	4-1386-64bit		
Plugins	{"html": "3.1.1", "Json-n	eport": "1.2.4", "metadata": "1.11.0"}		
Python	3.9.1			
Summar	y			
34 tests rar	n in 2.94 seconds.			
Un)check th	e baxes to filter the res	ults.		
634 page	ed. 🛛 0 skipped. 🖓 (	falled. 🔽 0 errors. 🔽 0 expected fall	res. 7 0 unexpected pesses	
Deculto				
results				
Show all deb	ails / Hide all details			
Result		- Test		- Duration
Passed (sh	ow dotails)	tests/ch02_math/ex01_basiscs_tes	py::test_caid[6-7-0]	0.00
Passed (sh	ow datails)	tests/ch02_math/ex01_basiscs_tes	py::test_sald[3-4-6]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	pyntest_cald[5-5-5]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py=test_calc_v2[6-7-0]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	pymtest_calc_v2[3-4-6]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::lest_calc_v2[5-5-5]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::test_calc_sum_and_count_all_numbers_div_by_2_cr_7[3-expected0]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::test_calc_sum_and_count_all_numbers_div_by_2_cr_7[8-expected1]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::lest_calc_sum_and_count_all_numbers_div_by_2_or_7[15-expected2]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::test_is_even(1-False)	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::test_js_even(2-True)	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::test_is_even(3-Faise)	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::list_is_even(4-True)	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::test_is_odd[1-True]	0.00
Passed (sh	ow datails)	tests/ch02_math/ex01_basiscs_tes	py::test_is_odd[2-False]	0.00
Passed (sh	ow details)	tests/ch02_math/ex01_basiscs_tes	py::test_is_odd(3-True)	0.00
Passed (sh	ow dotails)	tests/ch02_math/ex01_basiscs_tes	py::test_is_odd[4-False]	0.00
Passed (sh	ow datails)	tests/ch02_math/ex02_number_as,	ext_test.py::Ex02_NumberAsText_Test::test_number_as_text_0	0.00
Passed (sh	ow details)	tests/ch02 math/ex02 number as	ext test.pv::Ex02_NumberAsText_Test:dest_number_as_text_1	0.00

# Start der Programme

Das Ausführen einiger Programme kann entweder in der IDE geschehen oder als Test wie zuvor beschrieben. Bei einfachen Programmen kann auch der Python-Kommandozeileninterpreter genutzt werden.

•		Pyth	onChallenge – ex04_reverse_string.py
honChallenge 👌 ch03_recursion 👌 solutions 👌 👰 ex04,	_reverse_string.py		🚽 ex04_reverse_string 💌 🌗 🔅 😘 🖓 🖛 🔳
Project 👻	() <u>=</u> ÷	tt — Utils.py	🗴 🎼 programming_style.py 🛛 🚛 pytest-report.html 👋 👫 array_utils.py 👋 👫 ex04_reverpy 👋 👫 ex07_rom : N
PythonChallenge ~/PycharmProjects/Python	Challenge	1	o# Beispielprogramm für das Buch "Pythan Challenge" ▲1 ±8 ヘ
> im.pytest_cache		2	<i>a</i>
> assets		3	# Copyright 2020 by Michael Inden
Image: Ch01_introduction		-4	
> lim ch02_math		5	· · · · ·
ch03_recursion		6	<pre>_def reverse_string(input):</pre>
> 🖿 intro		7	# rekursiver Abbruch
solutions		8	<pre>if len(input) &lt;= 1:</pre>
> is ex01_fibonacci.py		9	return input
is ex02_digits.py		10	
> 💑 ex03_gcd.py			first char = input[0]
> 🖡 ex04_reverse_string.py		12	remaining = input[1:]
> second seco		17	Conversity - any determined
> participation in py		15	A estimation Archier
is ex07_convert.py		14	# rekursiver Abstieg
> is ex08_power_of.py		15	<pre>peturn reverse_string(remaining) + first_char</pre>
> [6] ex09_pascal_triangle.py		16	
> ipple.ex10_number_palindrome.py		17	
is ex11_permutations.py		18	<pre>def main():</pre>
> substrings.py		19	<pre>print(reverse_string("ABC"))</pre>
> period ext3_ruler.py		20	<pre>print(reverse_string("Michael"))</pre>
> 🖿 util		21	<pre>print(reverse_string("RACEcar"))</pre>
Im ch04_strings		22	
> in ch05_datastructures		23	<pre>print("Michael"[::-1])</pre>
> m ch06_arrays		24	<pre>print("".join(reversed("Michael")))</pre>
> im ch07_recursion_advanced			reverse string()
			Laval sa "on niñi")
un: 🥏 ex04_reverse_string ×			\$
. A /Users/michaeli/PycharmProjects/F	PythonChallen	ge/venv/bin/	oython /Users/michaeli/PycharmProjects/PythonChallenge/ch03_recursion/solutions/ex04_reve
CBA			
leahciM			
TP racECAR			
== leahciM			
= leahciM			
•			
Process finished with exit code (	0		
E TCDO 🕨 Run 🕕 Problems 🜵 Python Console	E Terminal		Q Event Lo

# Weitere Informationen

Im Buch finden Sie weitere Informationen zu den Programmen. Die Quelltexte finden Sie in den jeweiligen Unterordern zu den Kapiteln. Das PyCharm-Projekt zum Import finden Sie im Hauptverzeichnis. That's all ;-)

Bei eventuellen Fragen erreichen Sie mich unter: michael\_inden@hotmail.com