

Programming with Python

6. Beispiele Herunterladen

Thomas Weise (汤卫思)

tweise@hfuu.edu.cn · <https://thomasweise.github.io>

School of Artificial Intelligence and Big Data
Hefei University
Hefei, Anhui, China

人工智能与大数据学院
合肥大学
中国安徽省合肥市

Programming with Python



Dies ist ein Kurs über das Programmieren mit der Programmiersprache Python an der Universität Hefei (合肥大学).

Die Webseite mit dem Lehrmaterial dieses Kurses ist <https://thomasweise.github.io/programmingWithPython> (siehe auch den QR-Code unten rechts). Dort können Sie das Kursbuch (in Englisch) und diese Slides finden. Das Repository mit den Beispielprogrammen in Python finden Sie unter <https://github.com/thomasWeise/programmingWithPythonCode>.



Outline



1. Einleitung
2. Herunterladen der Beispiele
3. Repository Klonen
4. Zusammenfassung



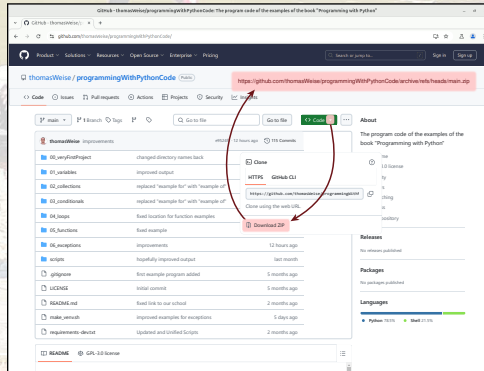


Einleitung



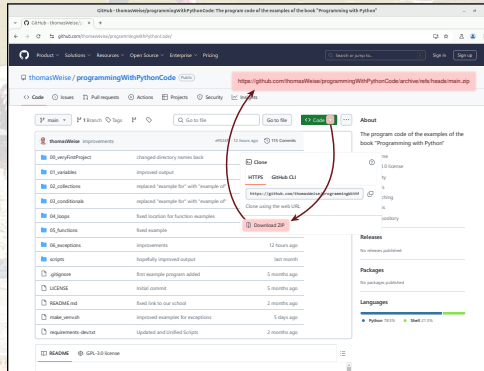
Einleitung

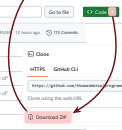
- Im Rest dieses Kurses werden wir intensiv mit praktischen Beispielen arbeiten.



Einleitung

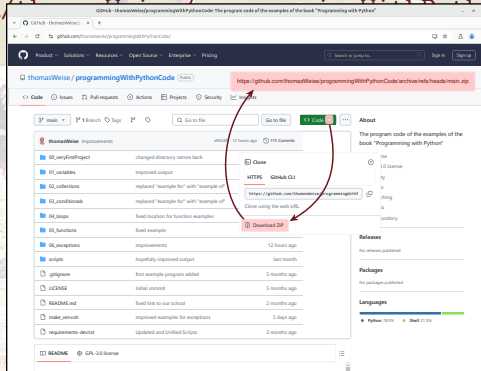
- Im Rest dieses Kurses werden wir intensiv mit praktischen Beispielen arbeiten.
- Wenn wir ein Konzept vorstellen, dann probieren wir das praktisch aus.





Einleitung

- Im Rest dieses Kurses werden wir intensiv mit praktischen Beispielen arbeiten.
- Wenn wir ein Konzept vorstellen, dann probieren wir das praktisch aus.
- Damit Sie diese Beispiele nachvollziehen können, stellen wir sie in einem GitHub repository zur Verfügung.
- Dieses Repository finden Sie unter <https://github.com/thomasweise/programmingWithPythonCode>.



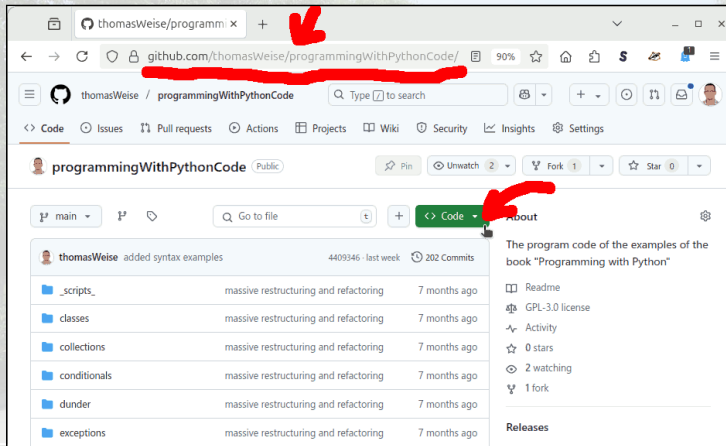


Herunterladen der Beispiele



Herunterladen der Beispiele

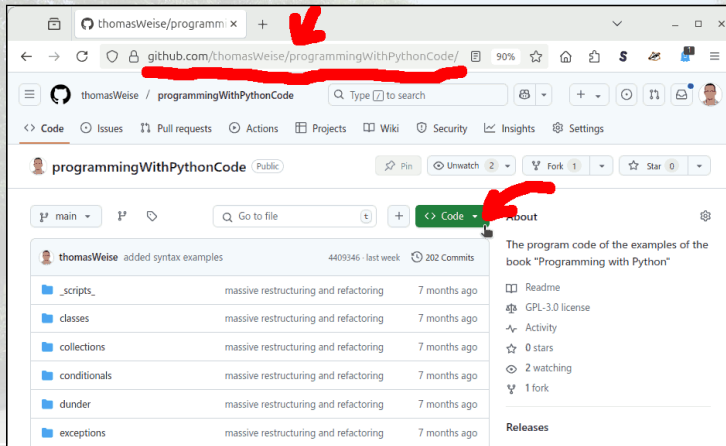
- Öffnen Sie einen Webbrowser und besuchen Sie die Webseite <https://github.com/thomasWeise/programmingWithPythonCode>.



Herunterladen der Beispiele



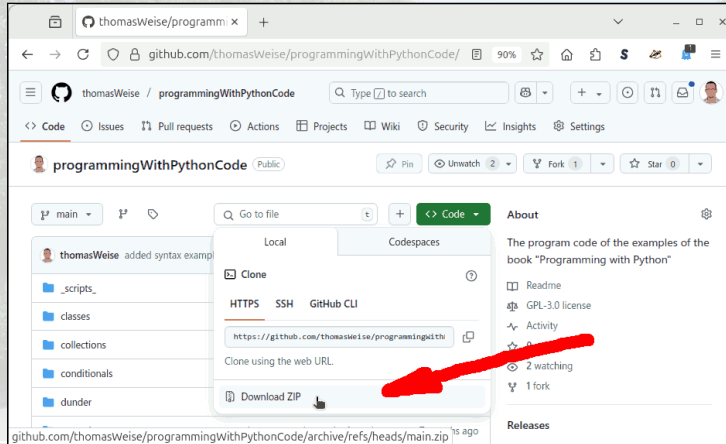
- Öffnen Sie einen Webbrowser und besuchen Sie die Webseite <https://github.com/thomasWeise/programmingWithPythonCode>. Klicken Sie auf das nach unten gerichtete Dreieck im Button `Code`.



Herunterladen der Beispiele

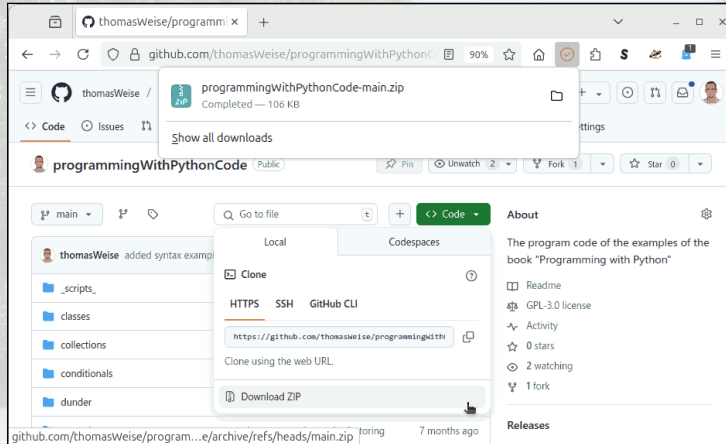


- In dem sich öffnenden Menü, klicken Sie auf **Download ZIP**.



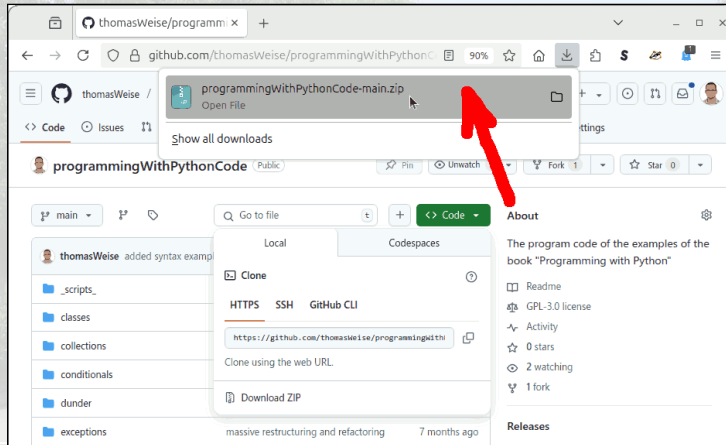
Herunterladen der Beispiele

- Der Download beginnt und ist irgendwann abgeschlossen.



Herunterladen der Beispiele

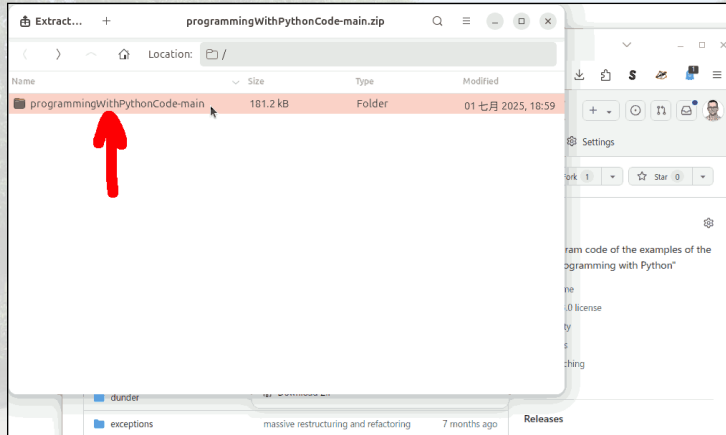
- Der Download beginnt und ist irgendwann abgeschlossen. Öffnen Sie die heruntergeladene Datei.



Herunterladen der Beispiele



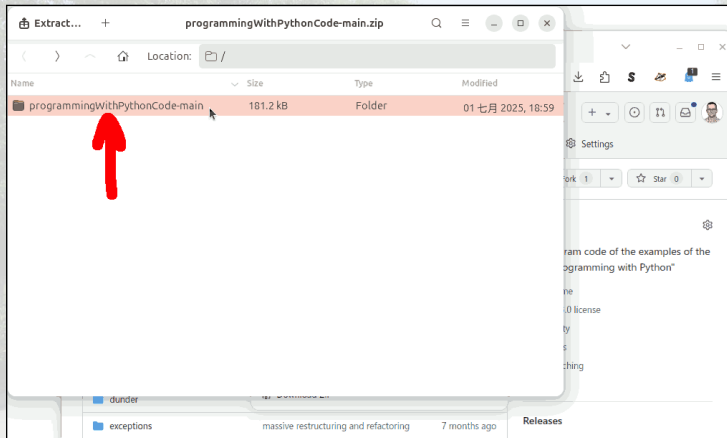
- Die Datei ist ein ZIP-Archiv, also eine Datei, die andere Dateien und Verzeichnisse enthält.



Herunterladen der Beispiele



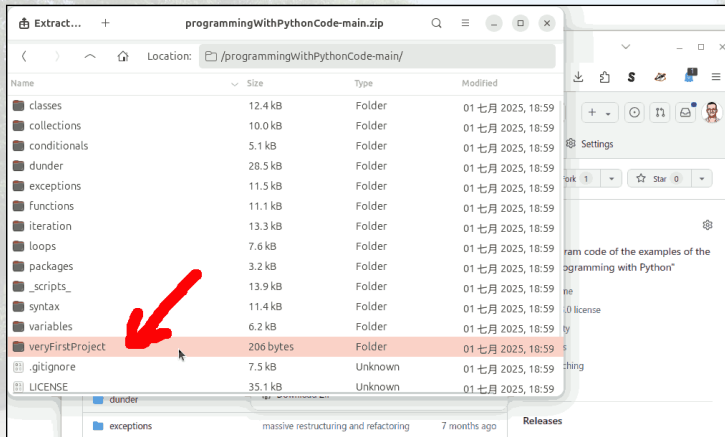
- Die Datei ist ein ZIP-Archiv, also eine Datei, die andere Dateien und Verzeichnisse enthält. Sie können Sie an einem Ihnen angenehmen Ort entpacken.



Herunterladen der Beispiele



- Die Datei ist ein ZIP-Archiv, also eine Datei, die andere Dateien und Verzeichnisse enthält. Sie können Sie an einem Ihnen angenehmen Ort entpacken. Das erste große Beispiel, mit dem wir als nächstes beginnen, befindet sich übrigens im Unterordner `veryFirstProject`.





Repository Klonen



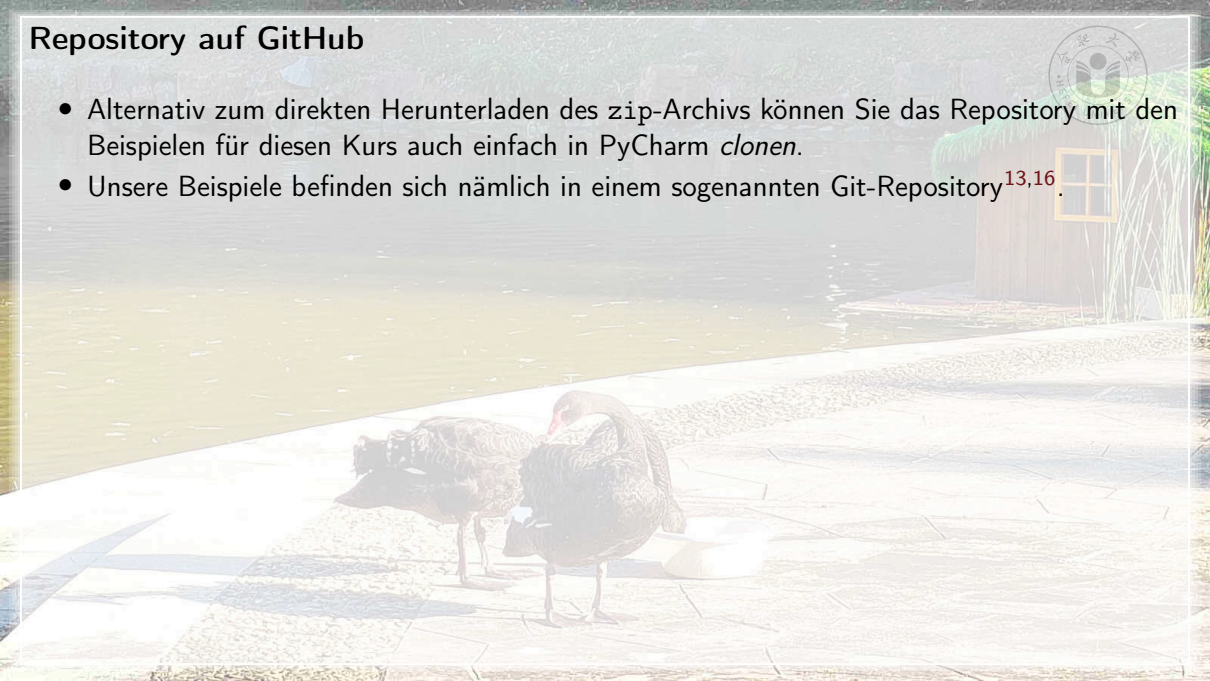
Repository auf GitHub

- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.



Repository auf GitHub

- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.



Repository auf GitHub



- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.
- Git ist ein Version Control Systems (VCS)^{13,16}, also ein Versionsmanagementsystem für Softwareentwicklung.



Repository auf GitHub



- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.
- Git ist ein Version Control Systems (VCS)^{13,16}, also ein Versionsmanagementsystem für Softwareentwicklung.
- Mit so einem System können wir schrittweise an unserem Code arbeiten und Änderungen in die Codebasis einfügen.



Repository auf GitHub



- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.
- Git ist ein Version Control Systems (VCS)^{13,16}, also ein Versionsmanagementsystem für Softwareentwicklung.
- Mit so einem System können wir schrittweise an unserem Code arbeiten und Änderungen in die Codebasis einfügen.
- Das VCS merkt sich die Geschichte unseres Projekts und erlaubt es uns, kollaborativ gemeinsam an dem Code zu arbeiten.

Repository auf GitHub



- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.
- Git ist ein Version Control Systems (VCS)^{13,16}, also ein Versionsmanagementsystem für Softwareentwicklung.
- Mit so einem System können wir schrittweise an unserem Code arbeiten und Änderungen in die Codebasis einfügen.
- Das VCS merkt sich die Geschichte unseres Projekts und erlaubt es uns, kollaborativ gemeinsam an dem Code zu arbeiten.
- Also wir werden nicht kollaborativ gemeinsam an dem Beispiel-Code arbeiten ... es sind ja Beispiele für diesen Kurs.

Repository auf GitHub



- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.
- Git ist ein Version Control Systems (VCS)^{13,16}, also ein Versionsmanagementsystem für Softwareentwicklung.
- Mit so einem System können wir schrittweise an unserem Code arbeiten und Änderungen in die Codebasis einfügen.
- Das VCS merkt sich die Geschichte unseres Projekts und erlaubt es uns, kollaborativ gemeinsam an dem Code zu arbeiten.
- Also wir werden nicht kollaborativ gemeinsam an dem Beispiel-Code arbeiten ... es sind ja Beispiele für diesen Kurs.
- Aber Git ist ein sehr weit verbreitetes VCS, also schade es nichts, zumindest grob zu wissen, wie das funktioniert.

Repository auf GitHub

- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.
- Git ist ein Version Control Systems (VCS)^{13,16}, also ein Versionsmanagementsystem für Softwareentwicklung.
- Mit so einem System können wir schrittweise an unserem Code arbeiten und Änderungen in die Codebasis einfügen.
- Das VCS merkt sich die Geschichte unseres Projekts und erlaubt es uns, kollaborativ gemeinsam an dem Code zu arbeiten.
- Also wir werden nicht kollaborativ gemeinsam an dem Beispiel-Code arbeiten ... es sind ja Beispiele für diesen Kurs.
- Aber Git ist ein sehr weit verbreitetes VCS, also schade es nichts, zumindest grob zu wissen, wie das funktioniert.
- Unsere Beispiele liegen in einem Git-Repository auf GitHub^{9,16}.

Repository auf GitHub

- Alternativ zum direkten Herunterladen des zip-Archivs können Sie das Repository mit den Beispielen für diesen Kurs auch einfach in PyCharm *clonen*.
- Unsere Beispiele befinden sich nämlich in einem sogenannten Git-Repository^{13,16}.
- Git ist ein Version Control Systems (VCS)^{13,16}, also ein Versionsmanagementsystem für Softwareentwicklung.
- Mit so einem System können wir schrittweise an unserem Code arbeiten und Änderungen in die Codebasis einfügen.
- Das VCS merkt sich die Geschichte unseres Projekts und erlaubt es uns, kollaborativ gemeinsam an dem Code zu arbeiten.
- Also wir werden nicht kollaborativ gemeinsam an dem Beispiel-Code arbeiten ... es sind ja Beispiele für diesen Kurs.
- Aber Git ist ein sehr weit verbreitetes VCS, also schade es nichts, zumindest grob zu wissen, wie das funktioniert.
- Unsere Beispiele liegen in einem Git-Repository auf GitHub^{9,16}.
- Clonen wir dieses Repository also!

Repository Klonen

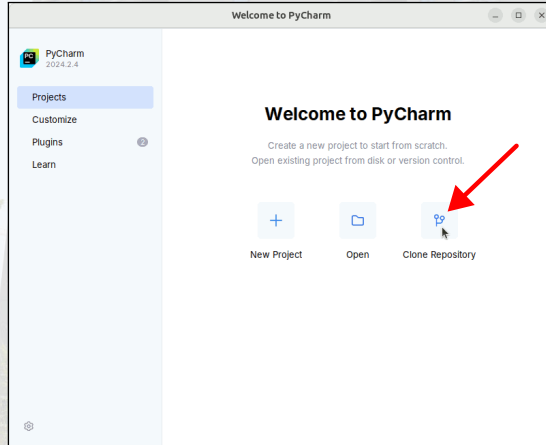
- Clonen wir dieses Repository also!



Repository Klonen



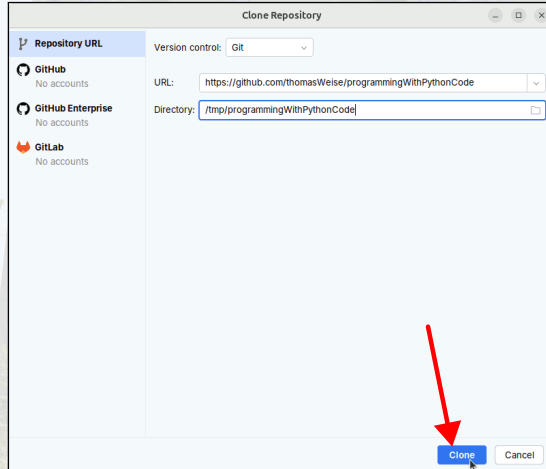
- Im PyCharm Willkommensbildschirm, klicken Sie auf **Clone Repository**.



Repository Klonen



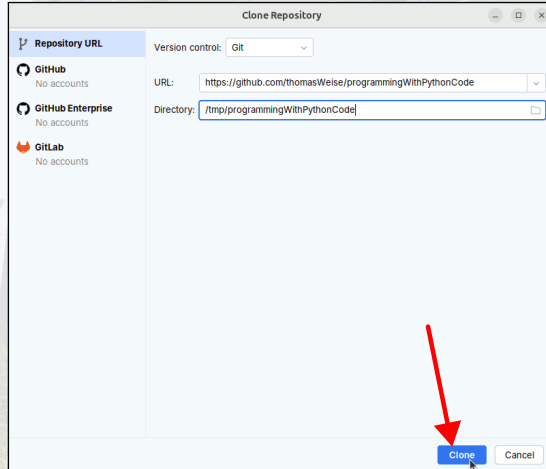
- Geben Sie <https://github.com/thomasWeise/programmingWithPythonCode> als URL: ein.



Repository Klonen



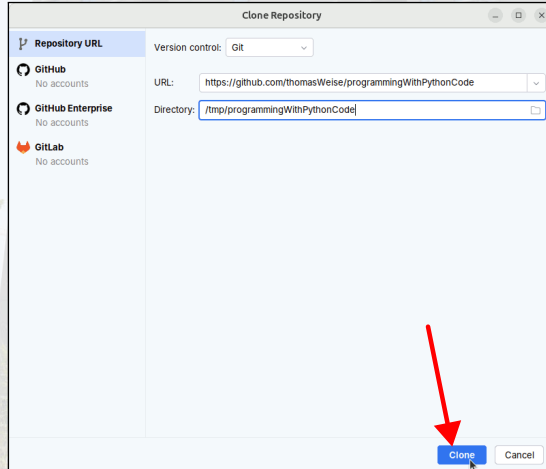
- Wählen Sie ein Verzeichnis als **Directory:** aus, wo das neue Projekt mit den Beispielen angelegt werden soll.



Repository Klonen



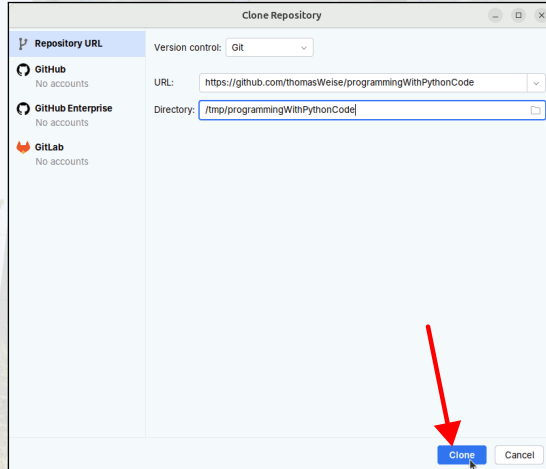
- Wählen Sie ein Verzeichnis als **Directory:** aus, wo das neue Projekt mit den Beispielen angelegt werden soll. Nehmen Sie nicht `/tmp`.



Repository Klonen

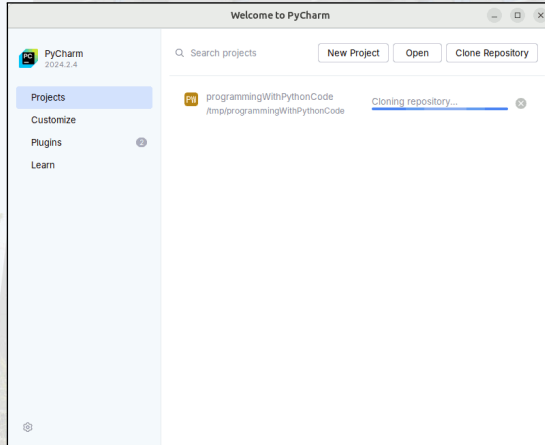


- Klicken Sie auf **Clone**.



Repository Klonen

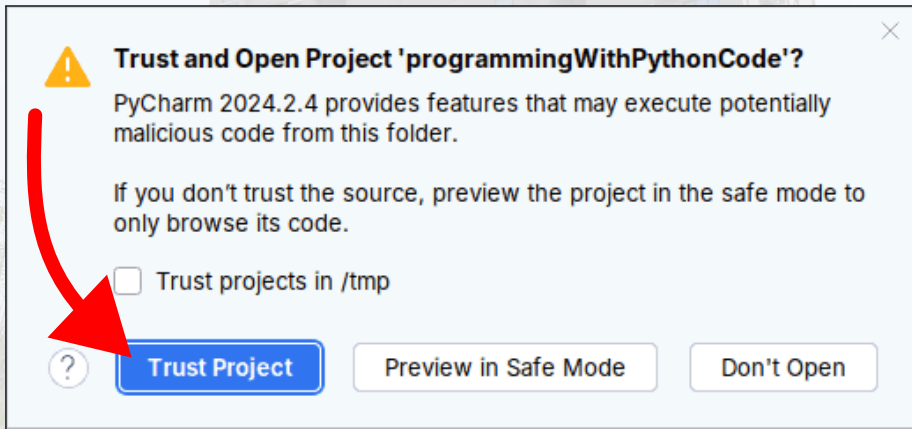
- Der Klon-Prozess beginnt.



Repository Klonen



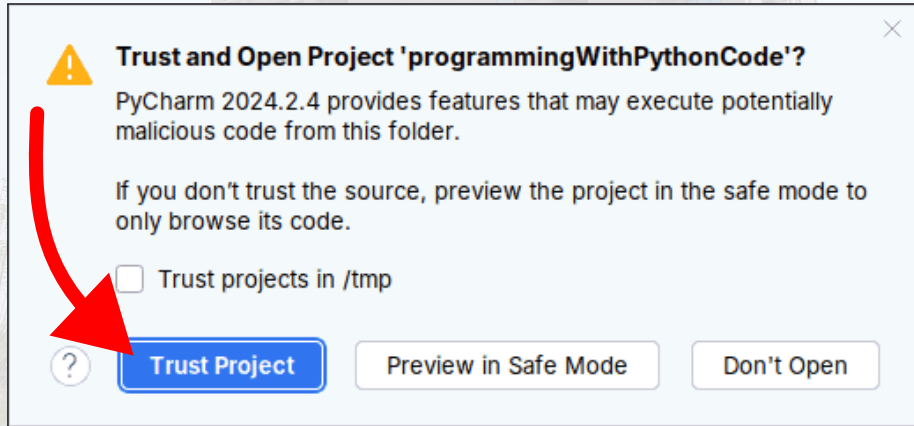
- Nach dem herunterladen wird PyCharm fragen, ob wir dem Projekt vertrauen.



Repository Klonen



- Nach dem herunterladen wird PyCharm fragen, ob wir dem Projekt vertrauen. Wenn ja, dann klicken Sie auf `Trust Project`.



Repository Klonen



- Nun können wir alle Beispieldateien sehen.

The screenshot shows a code editor interface. On the left, a 'Project' sidebar displays a file tree for 'programmingWithPythonCode'. The tree includes subfolders like '00_veryFirstProject', '01_variables', '02_collections', '03_conditionals', '04_loops', '05_functions', '06_exceptions', and 'scripts', along with files like '.gitignore', 'LICENSE', 'make_venv.sh', 'README.md', and 'requirements-dev.txt'. The main editor area shows the content of 'README.md'. The text in the README includes a title, a description, a table of contents, a license section, and a contact section. The status bar at the bottom indicates the file is 'README.md' in the 'programmingWithPythonCode' directory, using 'UTF-8' encoding with '4 spaces' and 'Python 3.12' syntax highlighting.

```
1 # Example Programs for the Book "Programming with Python"
2
3 Here we provide all the example programs and codes for our book
4
5 ## 1. Structure
6
7 1. A [Very First Project](...)
8 2. [Variables](...)
9 3. [Collections](...)
10 4. [Scripts](...)
11
12 ## 2. License
13 Copyright (C) 2023-2024 [Thomas Weise](...) (汤卫思教授)
14
15 The book "[Programming with Python](...)" is released under the
16 The code in this repository is provided as open source software
17
18 ## 3. Contact
19 If you have any questions or suggestions, please contact
```



Zusammenfassung





- Wir haben nun das Repository mit den Beispielen für diesen Kurs heruntergeladen.

Zusammenfassung



- Wir haben nun das Repository mit den Beispielen für diesen Kurs heruntergeladen.
- Damit haben Sie alle Programme, die wir im Folgenden verwenden, direkt zur Hand.

Zusammenfassung



- Wir haben nun das Repository mit den Beispielen für diesen Kurs heruntergeladen.
- Damit haben Sie alle Programme, die wir im Folgenden verwenden, direkt zur Hand.
- Sie können also unsere Beispiele sehr komfortabel nachvollziehen.

Zusammenfassung



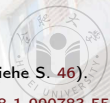
- Wir haben nun das Repository mit den Beispielen für diesen Kurs heruntergeladen.
- Damit haben Sie alle Programme, die wir im Folgenden verwenden, direkt zur Hand.
- Sie können also unsere Beispiele sehr komfortabel nachvollziehen.
- Als Seiteneffekt haben wir auch einen kurzen Blick auf Git und GitHub geworfen.



谢谢你们！
Thank you!
Vielen Dank!



References I

- 
- [1] Kent L. Beck. *JUnit Pocket Guide*. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2004. ISBN: **978-0-596-00743-0** (siehe S. 46).
- [2] Josh Centers. *Take Control of iOS 18 and iPadOS 18*. San Diego, CA, USA: Take Control Books, Dez. 2024. ISBN: **978-1-990783-55-5** (siehe S. 46).
- [3] John Hunt. *A Beginners Guide to Python 3 Programming*. 2. Aufl. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2023. ISBN: **978-3-031-35121-1**. doi:[10.1007/978-3-031-35122-8](https://doi.org/10.1007/978-3-031-35122-8) (siehe S. 46).
- [4] Kent D. Lee und Steve Hubbard. *Data Structures and Algorithms with Python*. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2015. ISBN: **978-3-319-13071-2**. doi:[10.1007/978-3-319-13072-9](https://doi.org/10.1007/978-3-319-13072-9) (siehe S. 46).
- [5] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: **978-1-0981-7130-8** (siehe S. 46).
- [6] A. Jefferson Offutt. "Unit Testing Versus Integration Testing". In: *Test: Faster, Better, Sooner – IEEE International Test Conference (ITC'1991)*. 26.–30. Okt. 1991, Nashville, TN, USA. Los Alamitos, CA, USA: IEEE Computer Society, 1991. Kap. Paper P2.3, S. 1108–1109. ISSN: **1089-3539**. ISBN: **978-0-8186-9156-0**. doi:[10.1109/TEST.1991.519784](https://doi.org/10.1109/TEST.1991.519784) (siehe S. 46).
- [7] Michael Olan. "Unit Testing: Test Early, Test Often". *Journal of Computing Sciences in Colleges (JCSC)* 19(2):319–328, Dez. 2003. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: **1937-4771**. doi:[10.5555/948785.948830](https://doi.org/10.5555/948785.948830). URL: <https://www.researchgate.net/publication/255673967> (besucht am 2025-09-05) (siehe S. 46).
- [8] Ashwin Pajankar. *Python Unit Test Automation: Automate, Organize, and Execute Unit Tests in Python*. New York, NY, USA: Apress Media, LLC, Dez. 2021. ISBN: **978-1-4842-7854-3** (siehe S. 46).
- [9] Yasset Pérez-Riverol, Laurent Gatto, Rui Wang, Timo Sachsenberg, Julian Uszkoreit, Felipe da Veiga Leprevost, Christian Fufezan, Tobias Ternent, Stephen J. Eglen, Daniel S. Katz, Tom J. Pollard, Alexander Kononov, Robert M. Flight, Kai Blin und Juan Antonio Vizcaíno. "Ten Simple Rules for Taking Advantage of Git and GitHub". *PLOS Computational Biology* 12(7), 14. Juli 2016. San Francisco, CA, USA: Public Library of Science (PLOS). ISSN: **1553-7358**. doi:[10.1371/JOURNAL.PCBI.1004947](https://doi.org/10.1371/JOURNAL.PCBI.1004947) (siehe S. 19–27, 46).
- [10] Ernest E. Rothman, Rich Rosen und Brian Jepson. *Mac OS X for Unix Geeks*. 4. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2008. ISBN: **978-0-596-52062-5** (siehe S. 46).

References II



- [11] Per Runeson. "A Survey of Unit Testing Practices". *IEEE Software* 23(4):22–29, Juli–Aug. 2006. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE). ISSN: **0740-7459**. doi:[10.1109/MS.2006.91](https://doi.org/10.1109/MS.2006.91) (siehe S. **46**).
- [12] Ahmad Sahar. *iOS 26 Programming for Beginners*. 10. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Nov. 2025. ISBN: **978-1-80602-393-6** (siehe S. **47**).
- [13] Anna Skoulikari. *Learning Git*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2023. ISBN: **978-1-0981-3391-7** (siehe S. **19–27, 46**).
- [14] Drew Smith. *Modern Apple Platform Administration – macOS and iOS Essentials (2025)*. Birmingham, England, UK: Packt Publishing Ltd, Feb. 2025. ISBN: **978-1-80580-309-6** (siehe S. **46**).
- [15] George K. Thiruvathukal, Konstantin Läufer und Benjamin Gonzalez. "Unit Testing Considered Useful". *Computing in Science & Engineering* 8(6):76–87, Nov.–Dez. 2006. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE). ISSN: **1521-9615**. doi:[10.1109/MCSE.2006.124](https://doi.org/10.1109/MCSE.2006.124). URL: <https://www.researchgate.net/publication/220094077> (besucht am 2024-10-01) (siehe S. **46**).
- [16] Mariot Tsitoara. *Beginning Git and GitHub: Version Control, Project Management and Teamwork for the New Developer*. New York, NY, USA: Apress Media, LLC, März 2024. ISBN: **979-8-8688-0215-7** (siehe S. **19–27, 46, 47**).
- [17] Bruce M. Van Horn II und Quan Nguyen. *Hands-On Application Development with PyCharm*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2023. ISBN: **978-1-83763-235-0** (siehe S. **46**).
- [18] Thomas Weise (汤卫思). *Programming with Python*. Hefei, Anhui, China (中国安徽省合肥市): Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2024–2026. URL: <https://thomasweise.github.io/programmingWithPython> (besucht am 2025-01-05) (siehe S. **46**).
- [19] Kevin Wilson. *Python Made Easy*. Birmingham, England, UK: Packt Publishing Ltd, Aug. 2024. ISBN: **978-1-83664-615-0** (siehe S. **46**).
- [20] Martin Yanev. *PyCharm Productivity and Debugging Techniques*. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2022. ISBN: **978-1-83763-244-2** (siehe S. **46**).

Glossary (in English) I



Git is a distributed Version Control Systems (VCS) which allows multiple users to work on the same code while preserving the history of the code changes^{13,16}. Learn more at <https://git-scm.com>.

GitHub is a website where software projects can be hosted and managed via the Git VCS^{9,16}. Learn more at <https://github.com>.

IDE An *Integrated Developer Environment* is a program that allows the user do multiple different activities required for software development in one single system. It often offers functionality such as editing source code, debugging, testing, or interaction with a distributed version control system. For Python, we recommend using PyCharm. On Apple systems, Xcode is often used.

iOS is the operating system that powers Apple iPhones^{2,14}. Learn more at <https://www.apple.com/ios>.

iPadOS is the operating system that powers Apple iPads². Learn more at <https://www.apple.com/ipados>.

macOS or Mac OS is the operating system that powers Apple Mac(intosh) computers^{10,14}. Learn more at <https://www.apple.com/macOS>.

PyCharm is the convenient Python Integrated Development Environment (IDE) that we recommend for this course^{17,19,20}. It comes in a free edition, so it can be downloaded and used at no cost. Learn more at <https://www.jetbrains.com/pycharm>.

Python The Python programming language^{3-5,18}, i.e., what you will learn about in our book¹⁸. Learn more at <https://python.org>.

unit test Software development is centered around creating the program code of an application, library, or otherwise useful system. A *unit test* is an *additional* code fragment that is not part of that productive code. It exists to execute (a part of) the productive code in a certain scenario (e.g., with specific parameters), to observe the behavior of that code, and to compare whether this behavior meets the specification^{1,6-8,11,15}. If not, the unit test fails. The use of unit tests is at least threefold: First, they help us to detect errors in the code. Second, program code is usually not developed only once and, from then on, used without change indefinitely. Instead, programs are often updated, improved, extended, and maintained over a long time. Unit tests can help us to detect whether such changes in the program code, maybe after years, violate the specification or, maybe, cause another, depending, module of the program to violate its specification. Third, they are part of the documentation or even specification of a program.

Glossary (in English) II



VCS A *Version Control System* is a software which allows you to manage and preserve the historical development of your program code¹⁶. A distributed VCS allows multiple users to work on the same code and upload their changes to the server, which then preserves the change history. The most popular distributed VCS is Git.

Xcode is offers the tools for developing, testing, and distributing applications as well as an IDE for Apple platforms such as macOS and iOS¹².