



合肥大學  
HEFEI UNIVERSITY



# Programming with Python

## 4. PyCharm Installieren

Thomas Weise (汤卫思)  
[tweise@hfuu.edu.cn](mailto:tweise@hfuu.edu.cn)

Institute of Applied Optimization (IAO)  
School of Artificial Intelligence and Big Data  
Hefei University  
Hefei, Anhui, China

应用优化研究所  
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. PyCharm unter Ubuntu Linux installieren
3. PyCharm unter Microsoft Windows installieren
4. Zusammenfassung





# Einleitung



# Einleitung

- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.





# Einleitung



- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.
- OK, technisch gesehen, ist es genug um Python-Programme auszuführen.

# Einleitung



- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.
- OK, technisch gesehen, ist es genug um Python-Programme auszuführen.
- Aber um vernünftig und effizient Software zu entwickeln, braucht man schon mehr.

# Einleitung



- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.
- OK, technisch gesehen, ist es genug um Python-Programme auszuführen.
- Aber um vernünftig und effizient Software zu entwickeln, braucht man schon mehr.
- Wir wollen ja mit dem Microsoft Windows Notepad programmieren...

```
my_math_3.py - Notepad
File Edit Format View Help
"""A third version of our module with mathematics routines."""

from math import isfinite

# factorial is omitted here for brevity

def sqrt(number: float) -> float:
    """
    Compute the square root of a given 'number'.

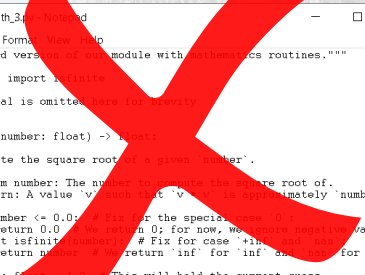
    :param number: The number to compute the square root of.
    :return: A value 'v' such that 'v * v' is approximately 'number'.
    """
    if number <= 0.0: # Fix for the special case '0':
        return 0.0 # We return 0; for now, we ignore negative values.
    if not isfinite(number): # Fix for case '+inf' and 'nan':
        return number # We return 'inf' for 'inf' and 'nan' for 'nan'.
    guess: float = 1.0 # This will hold the current guess
```



# Einleitung



- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.
- OK, technisch gesehen, ist es genug um Python-Programme auszuführen.
- Aber um vernünftig und effizient Software zu entwickeln, braucht man schon mehr.
- Wir wollen ja **nicht** mit dem Microsoft Windows Notepad programmieren...



```
my_math_3.py - Notepad
File Edit Format View Help
"""A third version of our module with mathematics routines."""

from math import inf

# factorial is omitted here for brevity

def sqrt(number: float) -> float:
    """
    Compute the square root of a given 'number'.

    :param number: The number to compute the square root of.
    :return: A value 'v' such that 'v * v' is approximately 'number'.
    """
    if number <= 0.0: # Fix for the special case '0':
        return 0.0 # We return 0; for now, we ignore negative values.
    if not isfinite(number): # Fix for case 'inf' and 'nan'
        return number # We return 'inf' for 'inf' and 'nan' for 'nan'.

    guess = 1.0 # This will hold the current guess
```

# Einleitung



- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.
- OK, technisch gesehen, ist es genug um Python-Programme auszuführen.
- Aber um vernünftig und effizient Software zu entwickeln, braucht man schon mehr.
- Wir wollen ja **nicht** mit dem Microsoft Windows NotePad programmieren. . .
- Wir brauchen ein Integrated Development Environment (IDE), ein Programm mit dem man mehrere der notwendigen Aufgaben in der Softwareentwicklung durch eine bequeme Benutzeroberfläche erledigen kann.

# Einleitung



- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.
- OK, technisch gesehen, ist es genug um Python-Programme auszuführen.
- Aber um vernünftig und effizient Software zu entwickeln, braucht man schon mehr.
- Wir wollen ja **nicht** mit dem Microsoft Windows NotePad programmieren. . .
- Wir brauchen ein Integrated Development Environment (IDE), ein Programm mit dem man mehrere der notwendigen Aufgaben in der Softwareentwicklung durch eine bequeme Benutzeroberfläche erledigen kann.
- Für Python empfehlen wir die PyCharm<sup>25,28,29</sup> IDE, deren Community Edition kostenlos ist.

# Einleitung



- Nur die Programmiersprache Python auf unserem System zu haben ist nicht genug.
- OK, technisch gesehen, ist es genug um Python-Programme auszuführen.
- Aber um vernünftig und effizient Software zu entwickeln, braucht man schon mehr.
- Wir wollen ja **nicht** mit dem Microsoft Windows NotePad programmieren. . .
- Wir brauchen ein Integrated Development Environment (IDE), ein Programm mit dem man mehrere der notwendigen Aufgaben in der Softwareentwicklung durch eine bequeme Benutzeroberfläche erledigen kann.
- Für Python empfehlen wir die PyCharm<sup>25,28,29</sup> IDE, deren Community Edition kostenlos ist.
- Die Installationsanweisungen für PyCharm finden sich unter <https://www.jetbrains.com/help/pycharm/installation-guide.html>.

# PyCharm unter Ubuntu Linux installieren



- Nachdem wir die Installationsanweisungen für die PyCharm Community Edition fertig gemacht haben, wurde angekündigt, dass diese von einer „unified edition“ ersetzt wird.

## An Important Update to PyCharm Community Edition

In our efforts to make PyCharm one powerful unified product, 2025.2 is the last supported version of PyCharm Community. With the 2025.3 release, we'll offer a seamless way to migrate to the unified PyCharm.

In the meantime, you can start using the unified PyCharm with Jupyter notebooks for free today without waiting for the migration.

[Download the unified PyCharm](#)

Free forever, with one month of Pro included.

[Read our extended FAQ](#)



# PyCharm unter Ubuntu Linux installieren



- Nachdem wir die Installationsanweisungen für die PyCharm Community Edition fertig gemacht haben, wurde angekündigt, dass diese von einer „unified edition“ ersetzt wird.
- Die folgenden Anweisungen sind daher wahrscheinlich etwas veraltet, sollten aber dennoch ausreichend instruktiv sein.

## An Important Update to PyCharm Community Edition

In our efforts to make PyCharm one powerful unified product, 2025.2 is the last supported version of PyCharm Community. With the 2025.3 release, we'll offer a seamless way to migrate to the unified PyCharm.

In the meantime, you can start using the unified PyCharm with Jupyter notebooks for free today without waiting for the migration.

[Download the unified PyCharm](#)

Free forever, with one month of Pro included.

[Read our extended FAQ](#)



# PyCharm unter Ubuntu Linux installieren



# PyCharm unter Ubuntu Linux installieren



- PyCharm steht unter Ubuntu Linux<sup>16,22</sup> als Snap-Package zur Verfügung.

# PyCharm unter Ubuntu Linux installieren



- PyCharm steht unter Ubuntu Linux<sup>16,22</sup> als Snap-Package zur Verfügung.
- Der Installationsprozess ist sehr einfach.



# PyCharm unter Ubuntu Linux installieren



- Zuerst öffnen wir ein Terminal in wir **Ctrl** + **Alt** + **T** drücken.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ sudo snap install pycharm-community --classic
```



# PyCharm unter Ubuntu Linux installieren



- Zuerst öffnen wir ein Terminal in wir `Ctrl` + `Alt` + `T` drücken.
- Dann schreiben wir das Kommando `sudo snap install pycharm-community --classic` und drücken `↵`.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ sudo snap install pycharm-community --classic
```

# PyCharm unter Ubuntu Linux installieren



- Zur Installation brauchen wir die sudo-Privilegien und müssen daher das sudo-Passwort eingeben.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ sudo snap install pycharm-community --classic  
[sudo] password for twiese: 
```

# PyCharm unter Ubuntu Linux installieren



- Die Installation verläuft automatisch.

```
tweise@weise-laptop: ~  
twaise@weise-laptop:~$ sudo snap install pycharm-community --classic  
[sudo] password for twaise: 
```

# PyCharm unter Ubuntu Linux installieren



- Die Installation verläuft automatisch.

```
tweise@weise-laptop: ~  
twaise@weise-laptop:~$ sudo snap install pycharm-community --classic  
[sudo] password for twaise:  
Download snap "pycharm-community" (388) from channel "stable"
```

# PyCharm unter Ubuntu Linux installieren



- Die Installation verläuft automatisch.

```
tweise@weise-laptop: ~  
twaise@weise-laptop:~$ sudo snap install pycharm-community --classic  
[sudo] password for twaise:  
Download snap "pycharm-commu" (388) from channel "st... 37% 1.37MB/s 5m26s
```



# PyCharm unter Ubuntu Linux installieren




- Die Installation verläuft automatisch.

```
tweise@weise-laptop: ~  
twaise@weise-laptop:~$ sudo snap install pycharm-community --classic  
[sudo] password for twaise:  
pycharm-community 2024.1.3 from jetbrains✓ installed  
twaise@weise-laptop:~$
```

# PyCharm unter Ubuntu Linux installieren




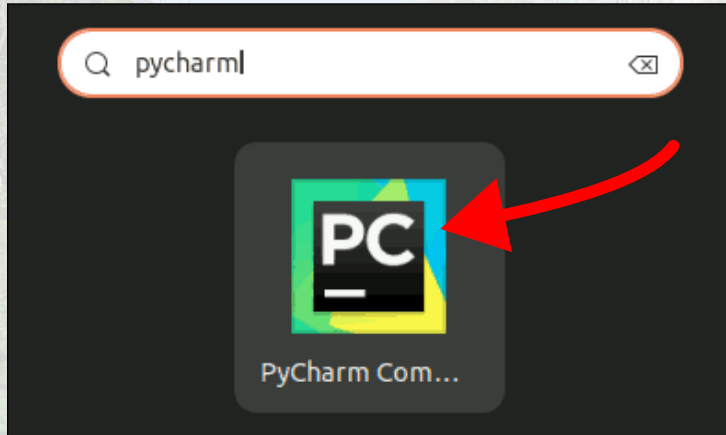
- Nachdem sie abgeschlossen ist, können Sie den Ubuntu Launcher durch Druck auf  öffnen und im Launcher-Fenster pycharm eingeben.



# PyCharm unter Ubuntu Linux installieren



- Nachdem sie abgeschlossen ist, können Sie den Ubuntu Launcher durch Druck auf  öffnen und im Launcher-Fenster pycharm eingeben. Klicken Sie auf das PyCharm-Symbol.



# PyCharm unter Ubuntu Linux installieren



- Der PyCharm Startbildschirm erscheint.



# PyCharm unter Ubuntu Linux installieren



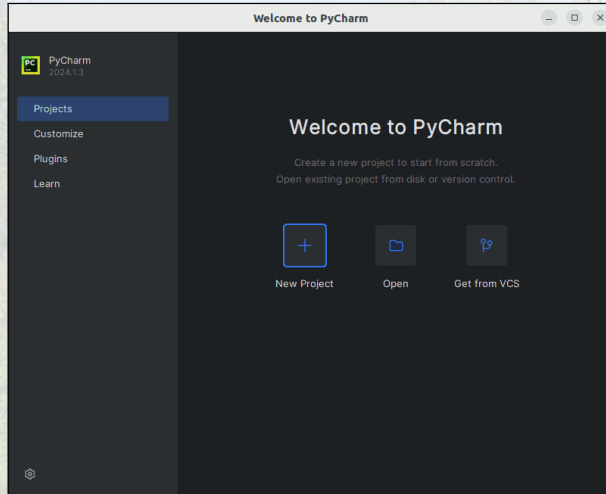
- Gegebenenfalls müssen Sie nun Lizenzbedingungen entweder zustimmen oder diese ablehnen.





# PyCharm unter Ubuntu Linux installieren

- Jetzt läuft PyCharm!





# PyCharm unter Microsoft Windows installieren



# PyCharm unter Microsoft Windows installieren



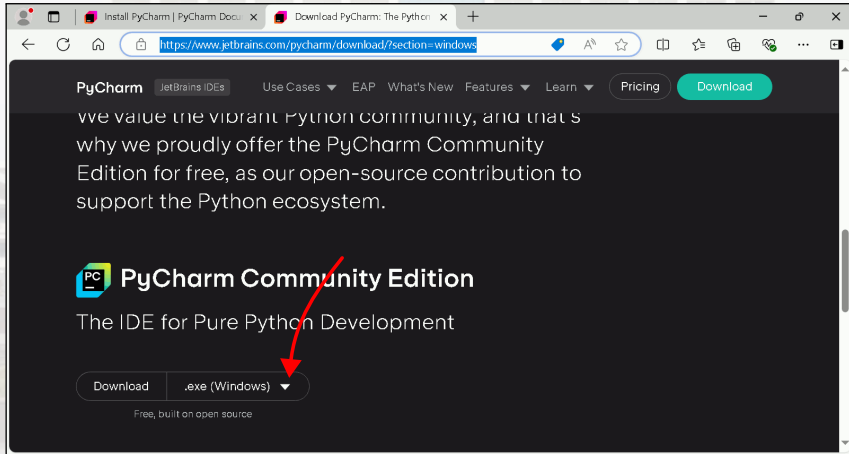
- Nun wollen wir PyCharm unter Microsoft Windows herunterladen und installieren.



# PyCharm unter Microsoft Windows installieren



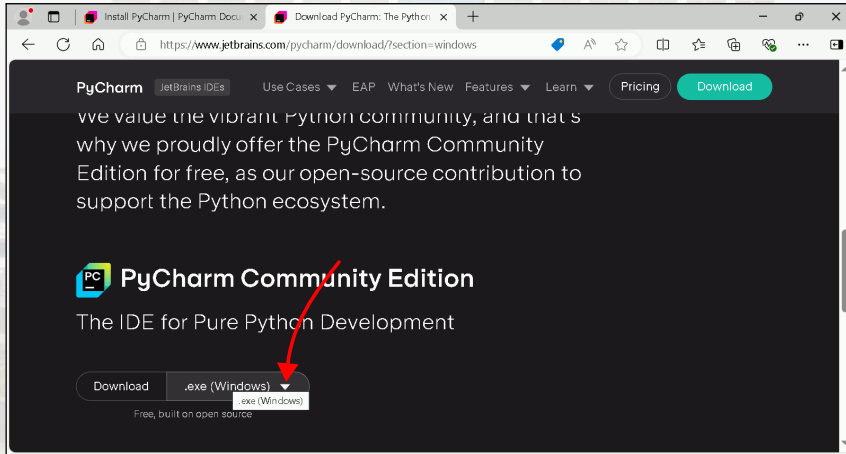
- Die Download-Webseite für PyCharm ist <https://www.jetbrains.com/pycharm/download> for PyCharm.



# PyCharm unter Microsoft Windows installieren



- Wir klicken auf `.exe (Windows)` um den PyCharm Community Edition-Download zu starten.

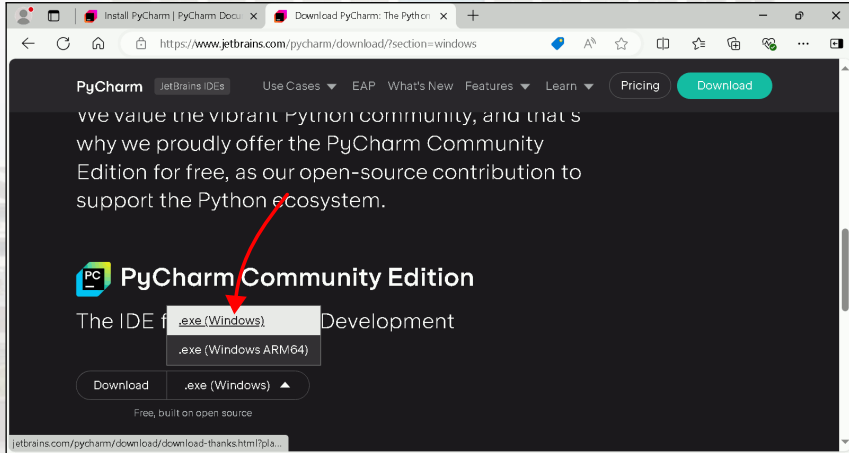




# PyCharm unter Microsoft Windows installieren



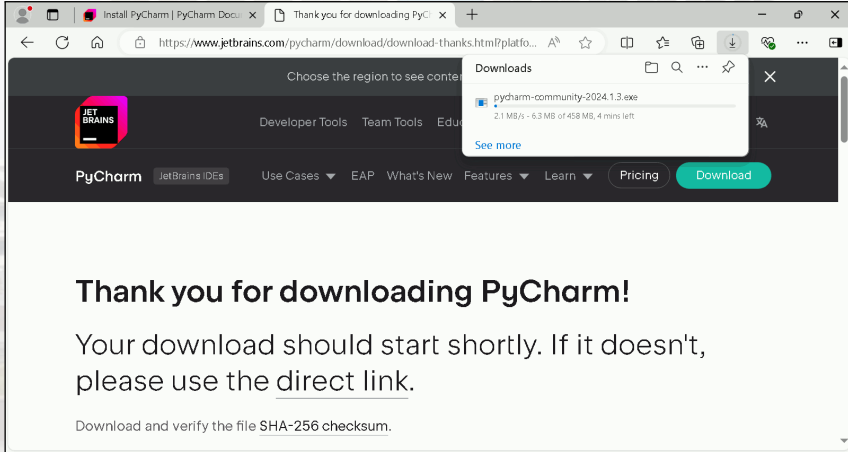
- In dem aufklappenden PopUp-Menü klicken wie ebenfalls auf `.exe (Windows)`.



# PyCharm unter Microsoft Windows installieren



- Der Download beginnt.



Choose the region to see content

pycharm-community-2024.1.3.exe  
2.1 MB/s - 6.3 MB of 458 MB, 4 mins left

See more

PyCharm JetBrains IDEs Use Cases ▼ EAP What's New Features ▼ Learn ▼ Pricing [Download](#)

## Thank you for downloading PyCharm!

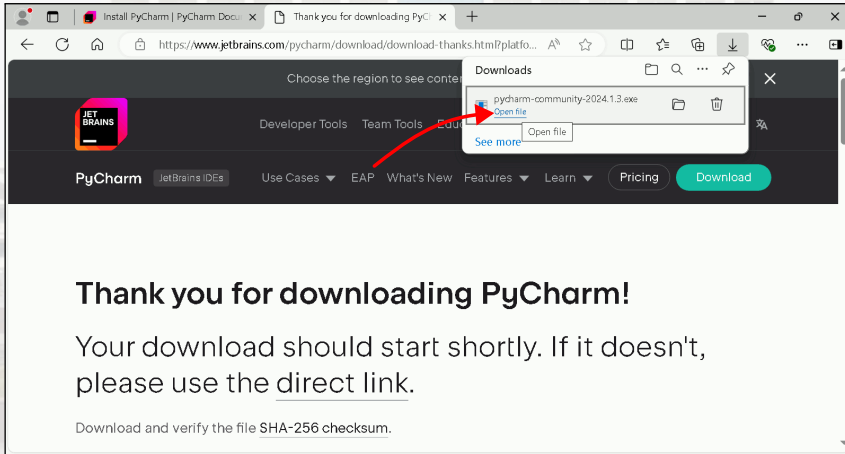
Your download should start shortly. If it doesn't,  
please use the [direct link](#).

Download and verify the file [SHA-256 checksum](#).

# PyCharm unter Microsoft Windows installieren



- Nach dem der Download abgeschlossen ist, führen wir das heruntergeladene Programm aus – z. B. durch Klick auf **Open file**.

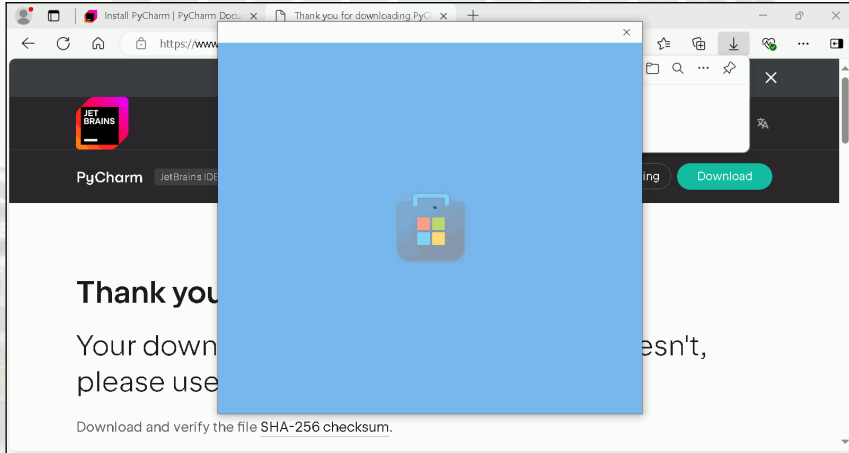


The screenshot shows a web browser window with the URL <https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows&version=2024.1.3>. The page displays the JetBrains logo and navigation links. A 'Downloads' dropdown menu is open, showing the file 'pycharm-community-2024.1.3.exe' with an 'Open file' button highlighted by a red arrow. Below the browser window, the text reads: 'Thank you for downloading PyCharm! Your download should start shortly. If it doesn't, please use the direct link.' At the bottom, it says: 'Download and verify the file [SHA-256 checksum](#).'

# PyCharm unter Microsoft Windows installieren



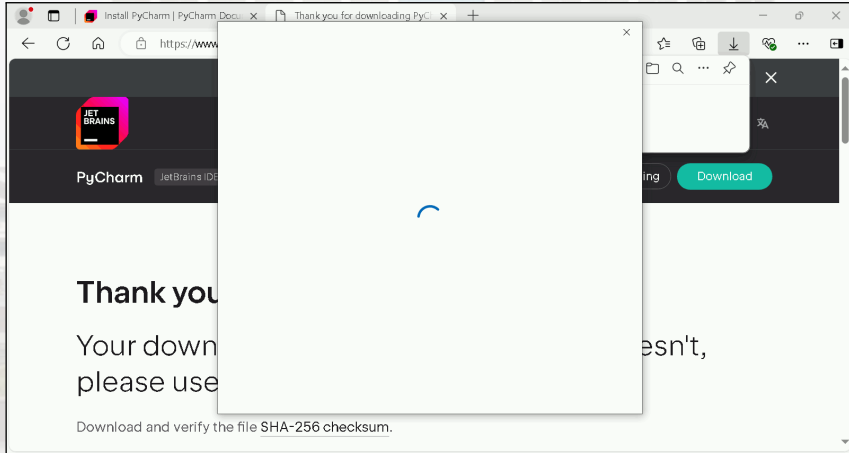
- Der Installer startet.



# PyCharm unter Microsoft Windows installieren



- Der Installer tut und macht.

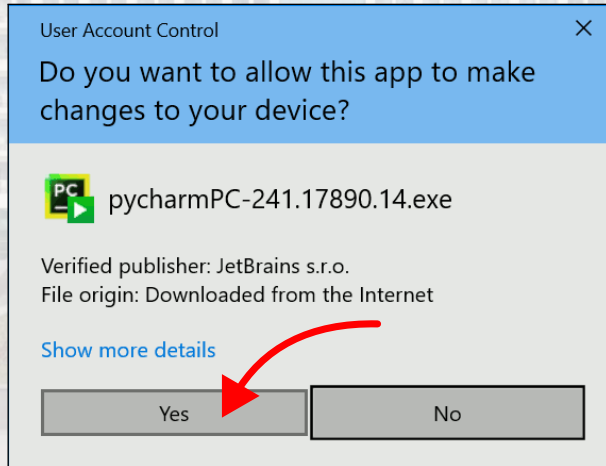




# PyCharm unter Microsoft Windows installieren



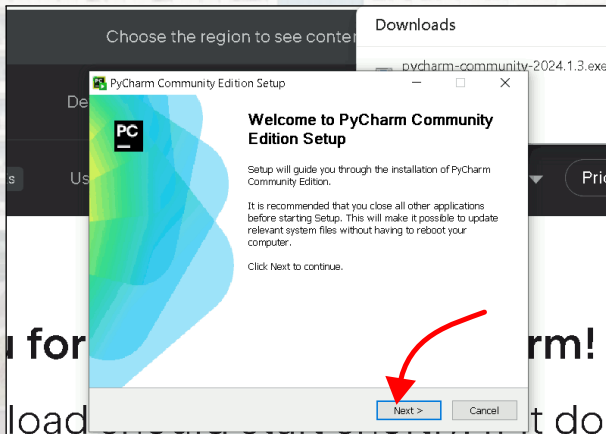
- Da wir PyCharm installieren wollen, klicken wir auf **Yes** und erlauben dem Programm, unser System zu verändern.



# PyCharm unter Microsoft Windows installieren



- Wir sind im Willkommensbildschirm des Installers angekommen. Wir klicken Next.

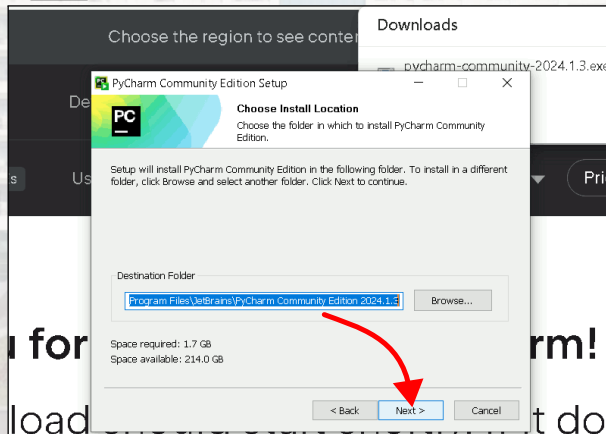


the direct link.

# PyCharm unter Microsoft Windows installieren



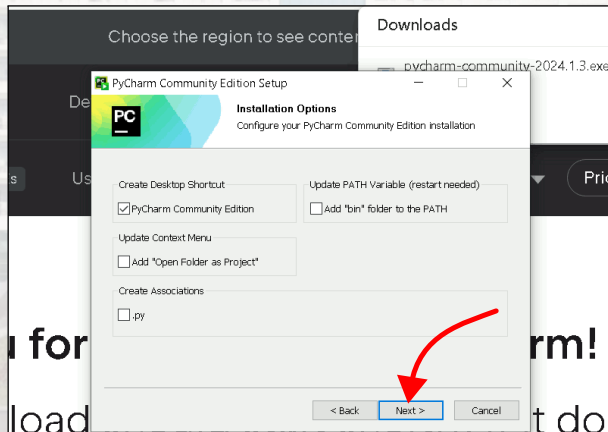
- Wir können den Installationsordner auswählen (oder einfach auf der Standareinstellung lassen) und klicken **Next**.



# PyCharm unter Microsoft Windows installieren



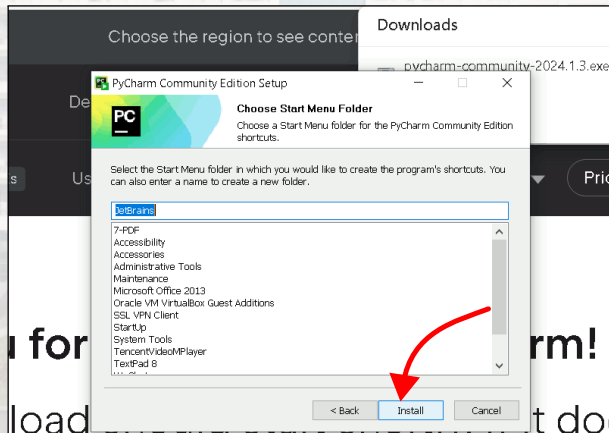
- Wir lassen alle Optionen auf den Standardeinstellungen und klicken **Next**.



# PyCharm unter Microsoft Windows installieren



- Wir lassen alle Optionen auf den Standardeinstellungen und klicken **Install**.

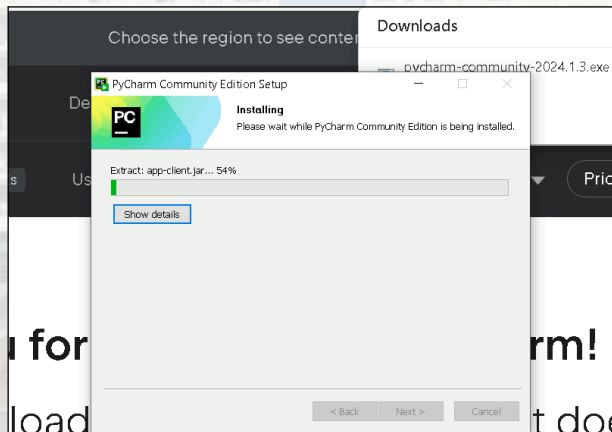




# PyCharm unter Microsoft Windows installieren



- Die Installation beginnt.

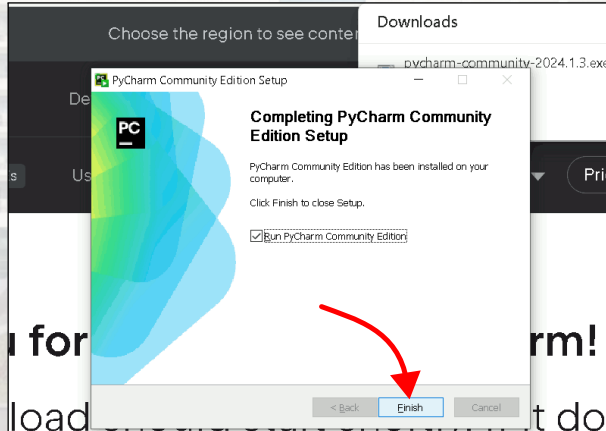


the direct link.

# PyCharm unter Microsoft Windows installieren



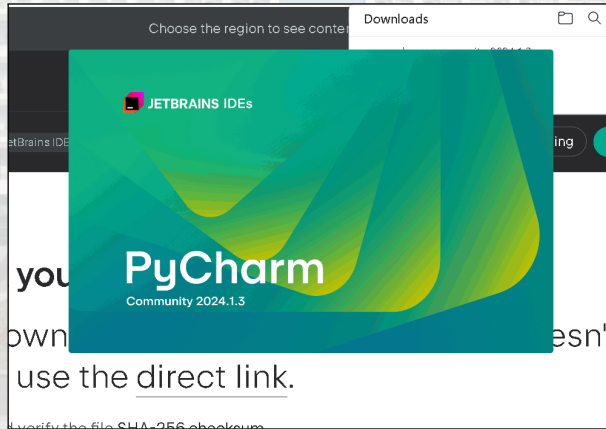
- Die Installation ist abgeschlossen. Wir wählen „Run PyCharm Community Edition“ aus und klicken **Finish**.



# PyCharm unter Microsoft Windows installieren



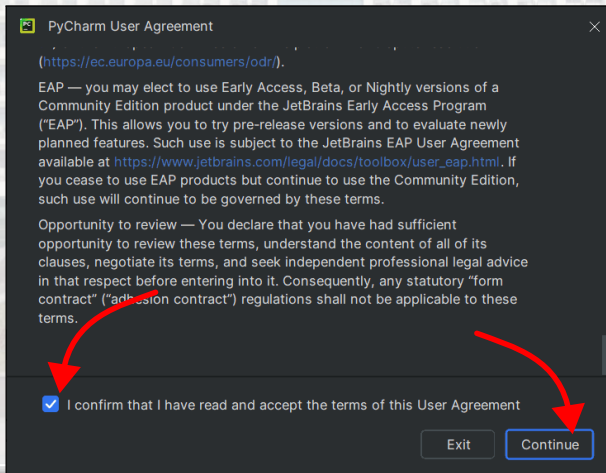
- Der Willkommensbildschirm von PyCharm.



# PyCharm unter Microsoft Windows installieren



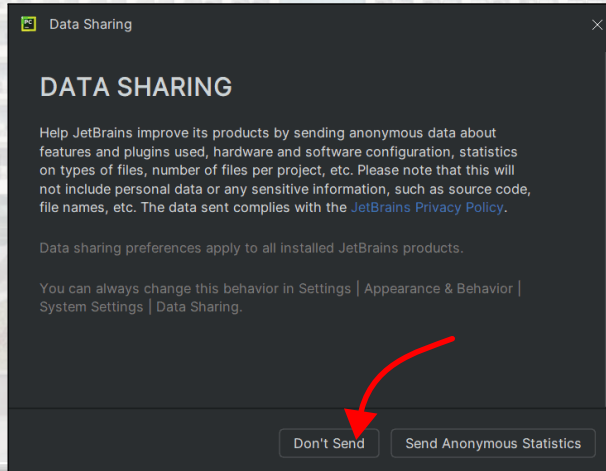
- Wir können entscheiden, ob wir den Nutzungsbedingungen zustimmen und klicken auf Continue.



# PyCharm unter Microsoft Windows installieren



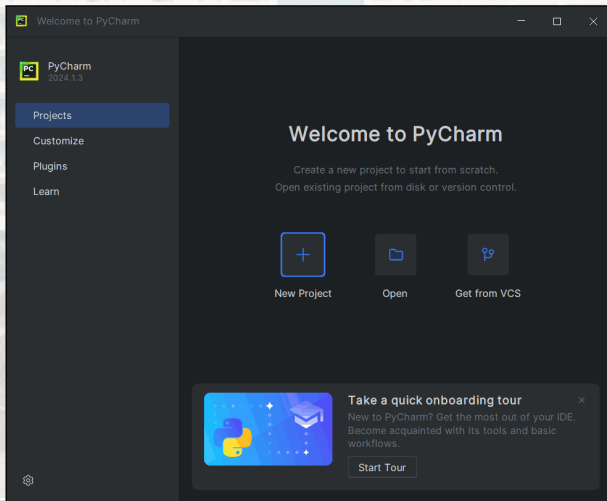
- Wir können entscheiden, ob wir Nutzerstatistiken an JetBrains schicken wollen. Ich entscheide mich für .





# PyCharm unter Microsoft Windows installieren

- PyCharm läuft!





# Zusammenfassung



# Zusammenfassung



- Nun haben wir PyCharm installiert.

# Zusammenfassung



- Nun haben wir PyCharm installiert.
- Damit haben wir eine komfortables Integrated Development Environment (IDE) für die Programmiersprache Python.

# Zusammenfassung



- Nun haben wir PyCharm installiert.
- Damit haben wir eine komfortables Integrated Development Environment (IDE) für die Programmiersprache Python.
- Cool.





谢谢你们！  
Thank you!  
Vielen Dank!





# References I



- [1] Daniel J. Barrett. *Efficient Linux at the Command Line*. Sebastopol, CA, USA: O'Reilly Media, Inc., Feb. 2022. ISBN: 978-1-0981-1340-7 (siehe S. 58, 59).
- [2] Kent L. Beck. *JUnit Pocket Guide*. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2004. ISBN: 978-0-596-00743-0 (siehe S. 59).
- [3] Ed Bott. *Windows 11 Inside Out*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Feb. 2023. ISBN: 978-0-13-769132-6 (siehe S. 58).
- [4] Ron Brash und Ganesh Naik. *Bash Cookbook*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2018. ISBN: 978-1-78862-936-2 (siehe S. 58).
- [5] 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. 58).
- [6] David Clinton und Christopher Negus. *Ubuntu Linux Bible*. 10. Aufl. Bible Series. Chichester, West Sussex, England, UK: John Wiley and Sons Ltd., 10. Nov. 2020. ISBN: 978-1-119-72233-5 (siehe S. 59).
- [7] Michael Hausenblas. *Learning Modern Linux*. Sebastopol, CA, USA: O'Reilly Media, Inc., Apr. 2022. ISBN: 978-1-0981-0894-6 (siehe S. 58).
- [8] Matthew Helmke. *Ubuntu Linux Unleashed 2021 Edition*. 14. Aufl. Reading, MA, USA: Addison-Wesley Professional, Aug. 2020. ISBN: 978-0-13-668539-5 (siehe S. 59).
- [9] 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 (siehe S. 58).
- [10] 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 (siehe S. 58).
- [11] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: 978-1-0981-7130-8 (siehe S. 58).
- [12] Cameron Newham und Bill Rosenblatt. *Learning the Bash Shell – Unix Shell Programming: Covers Bash 3.0*. 3. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., 2005. ISBN: 978-0-596-00965-6 (siehe S. 58).

# References II



- [13] 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. 59).
- [14] 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. 59).
- [15] 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. 59).
- [16] Prague, Czech Republic: JetBrains. *pycharm-community: PyCharm Community Edition*. London, England, UK: Canonical Ltd., 12. Dez. 2024. URL: <https://snapcraft.io/pycharm-community> (besucht am 2025-01-01) (siehe S. 16, 17).
- [17] 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. 58).
- [18] 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. 59).
- [19] 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. 59).
- [20] Ellen Siever, Stephen Figgins, Robert Love und Arnold Robbins. *Linux in a Nutshell*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2009. ISBN: **978-0-596-15448-6** (siehe S. 58).
- [21] 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. 58).
- [22] *Snap Documentation*. London, England, UK: Canonical Ltd., 2025. URL: <https://snapcraft.io/docs> (besucht am 2025-01-01) (siehe S. 16, 17).

# References III



- [23] 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. 59).
- [24] Linus Torvalds. "The Linux Edge". *Communications of the ACM (CACM)* 42(4):38–39, Apr. 1999. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: **0001-0782**. doi:[10.1145/299157.299165](https://doi.org/10.1145/299157.299165) (siehe S. 58).
- [25] 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. 5–12, 58).
- [26] Sander van Vugt. *Linux Fundamentals*. 2. Aufl. Hoboken, NJ, USA: Pearson IT Certification, Juni 2022. ISBN: **978-0-13-792931-3** (siehe S. 58).
- [27] 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), Institute of Applied Optimization (应用优化研究所, IAO), 2024–2025. URL: <https://thomasweise.github.io/programmingWithPython> (besucht am 2025-01-05) (siehe S. 58).
- [28] Kevin Wilson. *Python Made Easy*. Birmingham, England, UK: Packt Publishing Ltd, Aug. 2024. ISBN: **978-1-83664-615-0** (siehe S. 5–12, 58).
- [29] Martin Yanev. *PyCharm Productivity and Debugging Techniques*. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2022. ISBN: **978-1-83763-244-2** (siehe S. 5–12, 58).
- [30] Giorgio Zarrelli. *Mastering Bash*. Birmingham, England, UK: Packt Publishing Ltd, Juni 2017. ISBN: **978-1-78439-687-9** (siehe S. 58).

# Glossary (in English) I



**Bash** is a the shell used under Ubuntu Linux, i.e., the program that „runs“ in the terminal and interprets your commands, allowing you to start and interact with other programs<sup>4,12,30</sup>. Learn more at <https://www.gnu.org/software/bash>.

**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<sup>5,21</sup>. Learn more at <https://www.apple.com/ios>.

**iPadOS** is the operating system that powers Apple iPads<sup>5</sup>. Learn more at <https://www.apple.com/ipados>.

**IT** information technology

**Linux** is the leading open source operating system, i.e., a free alternative for Microsoft Windows<sup>1,7,20,24,26</sup>. We recommend using it for this course, for software development, and for research. Learn more at <https://www.linux.org>. Its variant Ubuntu is particularly easy to use and install.

**macOS** or Mac OS is the operating system that powers Apple Mac(intosh) computers<sup>17,21</sup>. Learn more at <https://www.apple.com/macOS>.

**Microsoft Windows** is a commercial proprietary operating system<sup>3</sup>. It is widely spread, but we recommend using a Linux variant such as Ubuntu for software development and for our course. Learn more at <https://www.microsoft.com/windows>.

**PyCharm** is the convenient Python IDE that we recommend for this course<sup>25,28,29</sup>. It comes in a free community edition, so it can be downloaded and used at no cost. Learn more at <https://www.jetbrains.com/pycharm>.

**Python** The Python programming language<sup>9–11,27</sup>, i.e., what you will learn about in our book<sup>27</sup>. Learn more at <https://python.org>.



# Glossary (in English) II



**sudo** In order to perform administrative tasks such as installing new software under Linux, root (or “super”) user privileges are needed<sup>6</sup>. A normal user can execute a program in the terminal as super user by pre-pending `sudo`, often referred to as “super user do.” This requires the root password.

**terminal** A terminal is a text-based window where you can enter commands and execute them<sup>1,6</sup>. Knowing what a terminal is and how to use it is very essential in any programming- or system administration-related task. If you want to open a terminal under Microsoft Windows, you can Druck auf `Win`+`R`, dann Schreiben von `cmd`, dann Druck auf `↵`. Under Ubuntu Linux, `Ctrl`+`Alt`+`T` opens a terminal, which then runs a Bash shell inside.

**Ubuntu** is a variant of the open source operating system Linux<sup>6,8</sup>. We recommend that you use this operating system to follow this class, for software development, and for research. Learn more at <https://ubuntu.com>. If you are in China, you can download it from <https://mirrors.ustc.edu.cn/ubuntu-releases>.

**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<sup>2,13–15,18,23</sup>. 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.

**Xcode** is offers the tools for developing, testing, and distributing applications as well as an IDE for Apple platforms such as macOS and iOS<sup>19</sup>.