



Programming with Python

9. Zwischenspiel: Python Dokumentation und Informationsquellen

Thomas Weise (汤卫思)
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-Kode 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. Die offizielle Dokumentation verwenden
3. Suchen mit einer Suchmaschine
4. Suchen mit Community-Portalen
5. Suchen mit AI
6. Weitere Elemente der Offiziellen Dokumentation
7. Zusammenfassung



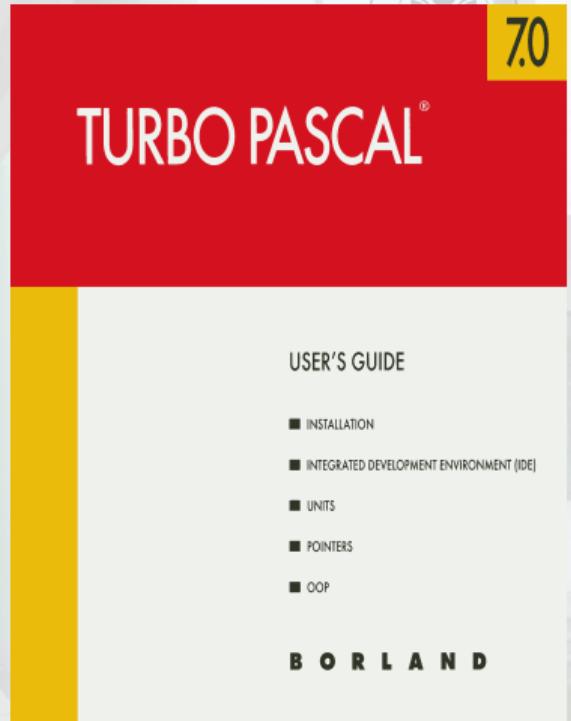


Einleitung



Einleitung

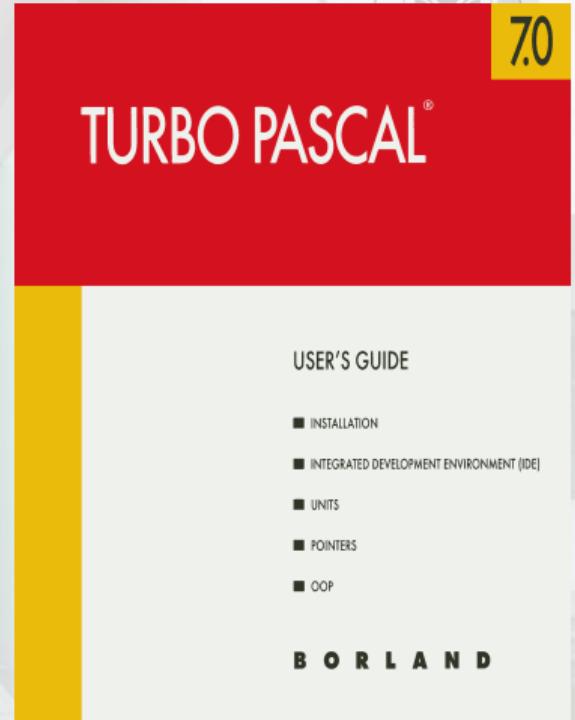
- Ich habe Programmieren in den 1990ern mit Turbo Pascal²⁸ gelernt.



Die Titelseite vom Turbo Pascal® Version 7.0 – User's Guide²⁸, copyright Borland.

Einleitung

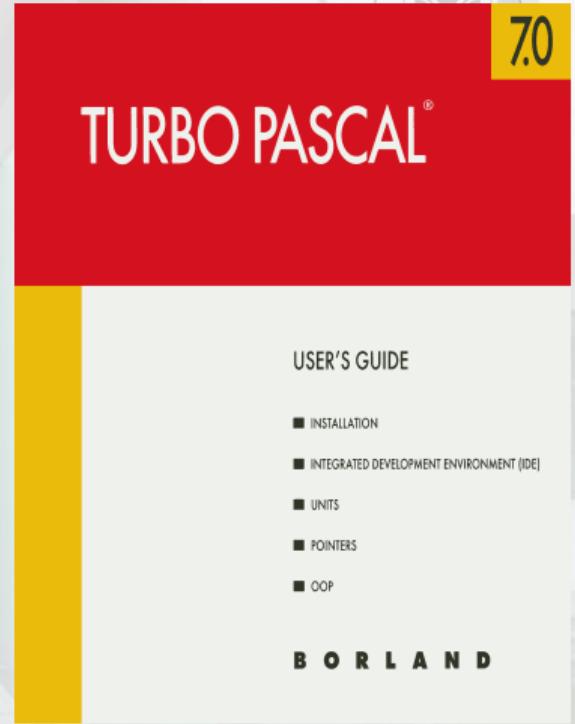
- Ich habe Programmieren in den 1990ern mit Turbo Pascal²⁸ gelernt.
- Hauptsächlich habe ich es mir selber beigebracht, wobei ich zwei Quellen verwendet habe.



Die Titelseite vom Turbo Pascal® Version 7.0 – User's Guide²⁸, copyright Borland.

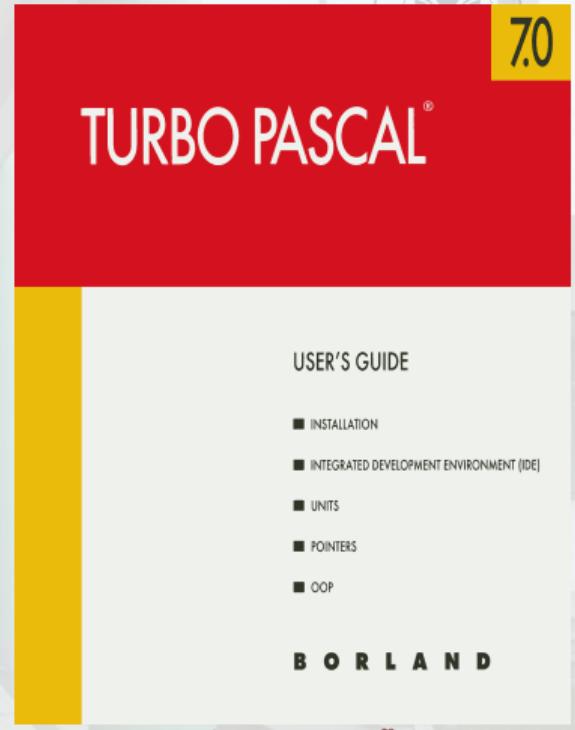
Einleitung

- Ich habe Programmieren in den 1990ern mit Turbo Pascal²⁸ gelernt.
- Hauptsächlich habe ich es mir selber beigebracht, wobei ich zwei Quellen verwendet habe:
 1. Bücher



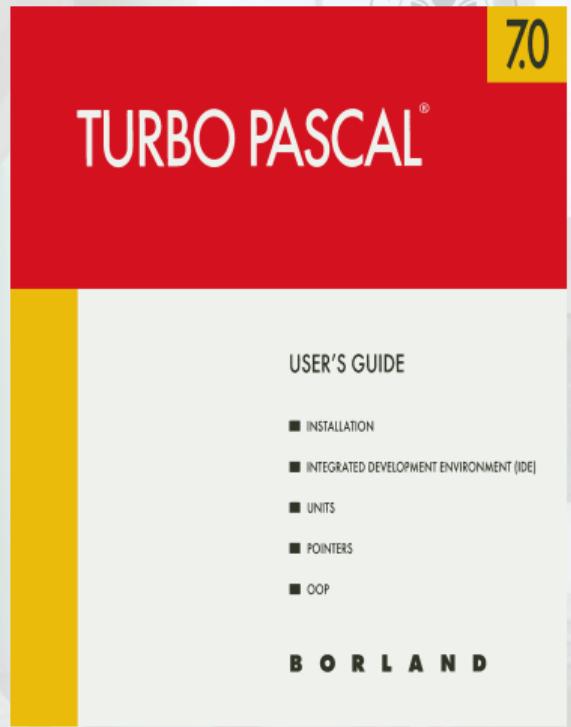
Einleitung

- Ich habe Programmieren in den 1990ern mit Turbo Pascal²⁸ gelernt.
- Hauptsächlich habe ich es mir selber beigebracht, wobei ich zwei Quellen verwendet habe:
 1. Bücher
 2. Das großartige Hilfesystem von Turbo Pascal



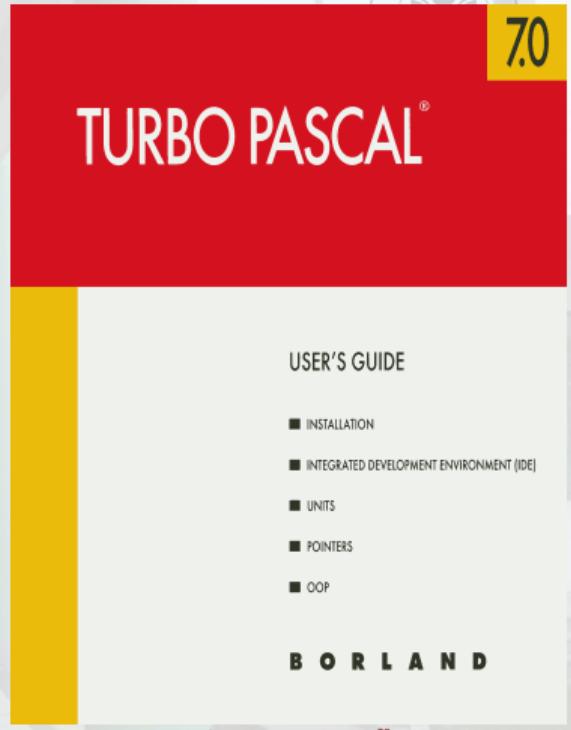
Einleitung

- Ich habe Programmieren in den 1990ern mit Turbo Pascal²⁸ gelernt.
- Hauptsächlich habe ich es mir selber beigebracht, wobei ich zwei Quellen verwendet habe:
 1. Bücher
 2. Das großartige Hilfesystem von Turbo Pascal
- Dieses Hilfesystem hatte jeden einzelnen Befehl der Sprache aufgelistet, mit Beispielen!



Einleitung

- Ich habe Programmieren in den 1990ern mit Turbo Pascal²⁸ gelernt.
- Hauptsächlich habe ich es mir selber beigebracht, wobei ich zwei Quellen verwendet habe:
 1. Bücher
 2. Das großartige Hilfesystem von Turbo Pascal
- Dieses Hilfesystem hatte jeden einzelnen Befehl der Sprache aufgelistet, mit Beispielen!
- So konnte ich selbstständig neue Befehle lernen und ausprobieren.



7.0

TURBO PASCAL®

USER'S GUIDE

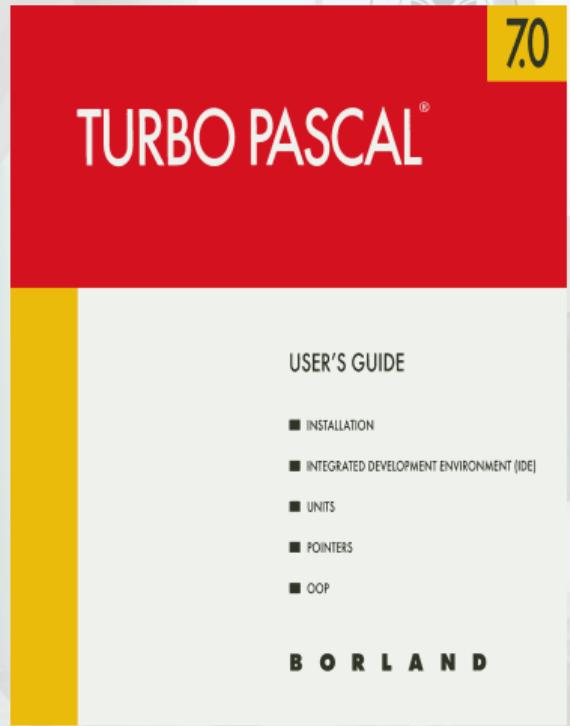
- INSTALLATION
- INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)
- UNITS
- POINTERS
- OOP

BORLAND

Die Titelseite vom Turbo Pascal® Version 7.0 – User's Guide²⁸, copyright Borland.

Einleitung

- Ich habe Programmieren in den 1990ern mit Turbo Pascal²⁸ gelernt.
- Hauptsächlich habe ich es mir selber beigebracht, wobei ich zwei Quellen verwendet habe:
 1. Bücher
 2. Das großartige Hilfesystem von Turbo Pascal
- Dieses Hilfesystem hatte jeden einzelnen Befehl der Sprache aufgelistet, mit Beispielen!
- So konnte ich selbstständig neue Befehle lernen und ausprobieren.
- *Wie können Sie neue Befehle für Python lernen und ausprobieren?*





Die offizielle Dokumentation verwenden



Die offizielle Dokumentation verwenden

- Angenommen, Sie kennen den Namen eines Kommandos und wollen mehr darüber erfahren, z. B. die genaue Bedeutung, Beschränkungen für die Werte der Parameter, mögliche Fehler, usw.



Die offizielle Dokumentation verwenden

- Angenommen, Sie kennen den Namen eines Kommandos und wollen mehr darüber erfahren, z. B. die genaue Bedeutung, Beschränkungen für die Werte der Parameter, mögliche Fehler, usw.
- Dafür gibt es einen kompletten Katalog mit Hilfe für Python: die *Python 3 Documentation*¹³ bei <https://docs.python.org/3>.



Die offizielle Dokumentation verwenden

- Angenommen, Sie kennen den Namen eines Kommandos und wollen mehr darüber erfahren, z. B. die genaue Bedeutung, Beschränkungen für die Werte der Parameter, mögliche Fehler, usw.
- Dafür gibt es einen kompletten Katalog mit Hilfe für Python: die *Python 3 Documentation*¹³ bei <https://docs.python.org/3>.
- Dies sollte Ihr erster Ansatzpunkt sein, wenn Sie mehr über Python lernen wollen.



Die offizielle Dokumentation verwenden

- Angenommen, Sie kennen den Namen eines Kommandos und wollen mehr darüber erfahren, z. B. die genaue Bedeutung, Beschränkungen für die Werte der Parameter, mögliche Fehler, usw.
- Dafür gibt es einen kompletten Katalog mit Hilfe für Python: die *Python 3 Documentation*¹³ bei <https://docs.python.org/3>.
- Dies sollte Ihr erster Ansatzpunkt sein, wenn Sie mehr über Python lernen wollen.
- Diese Hilfe steht auch in Chinesisch bereit!



Die offizielle Dokumentation verwenden

- Angenommen, Sie kennen den Namen eines Kommandos und wollen mehr darüber erfahren, z. B. die genaue Bedeutung, Beschränkungen für die Werte der Parameter, mögliche Fehler, usw.
- Dafür gibt es einen kompletten Katalog mit Hilfe für Python: die *Python 3 Documentation*¹³ bei <https://docs.python.org/3>.
- Dies sollte Ihr erster Ansatzpunkt sein, wenn Sie mehr über Python lernen wollen.
- Diese Hilfe steht auch in Chinesisch bereit!
- Nehmen wir an, Sie wollen mehr über die Funktion `ceil` erfahren.



Die offizielle Dokumentation verwenden



- Nehmen wir an, Sie wollen mehr über die Funktion `ceil` erfahren.
- Wir besuchen die Website *Python 3 Documentation* bei <https://docs.python.org/3>.

A screenshot of a web browser window displaying the Python 3.13.3 documentation. The title bar says "3.13.3 Documentation". The address bar shows the URL "https://docs.python.org/3/". A red heart-shaped mark is overlaid on the top-left corner of the browser window. The main content area is titled "Python 3.13.3 documentation" and includes a welcome message and a list of documentation sections.

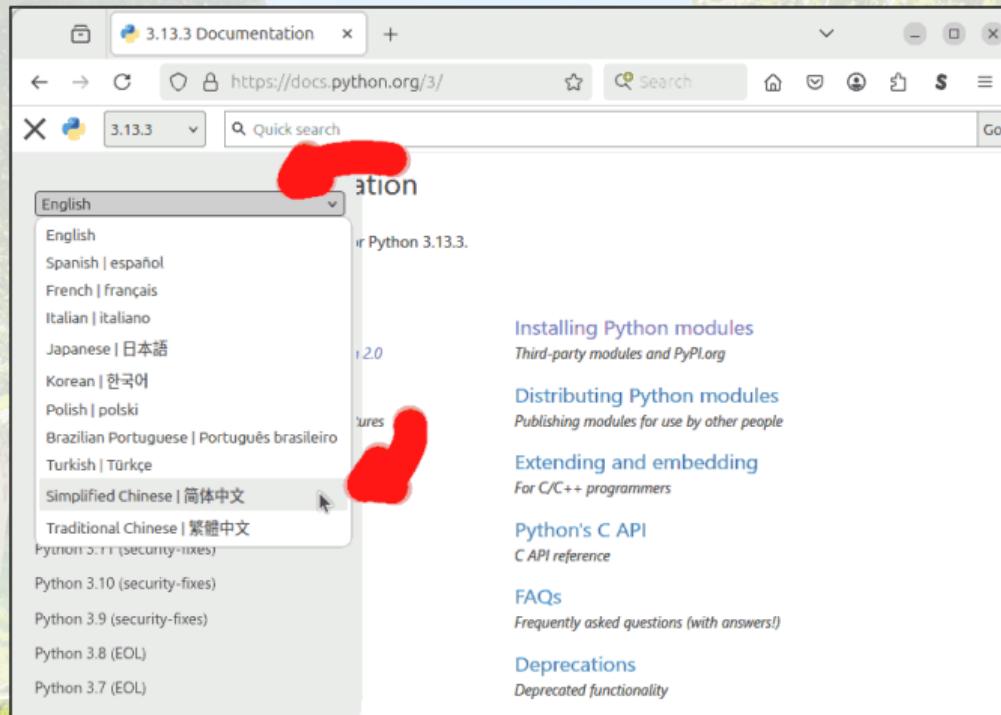
Welcome! This is the official documentation for Python 3.13.3.

Documentation sections:

What's new in Python 3.13? <small>Or all "What's new" documents since Python 2.0</small>	Installing Python modules <small>Third-party modules and PyPI.org</small>
Tutorial <small>Start here: a tour of Python's syntax and features</small>	Distributing Python modules <small>Publishing modules for use by other people</small>
Library reference <small>Standard library and builtins</small>	Extending and embedding <small>For C/C++ programmers</small>
Language reference <small>Syntax and language elements</small>	Python's C API <small>C API reference</small>
Python setup and usage <small>How to install, configure, and use Python</small>	FAQs <small>Frequently asked questions (with answers!)</small>
Python HOWTOs <small>In-depth topic manuals</small>	Deprecations <small>Deprecated functionality</small>

Die offizielle Dokumentation verwenden

- Wir können verschiedene Sprachen auswählen, auch Chinesisch ... aber lassen wir es bei Englisch.



Die offizielle Dokumentation verwenden

- Wir geben `ceil` in die Suchbox ein und klicken auf `Go`.

3.13.3 Documentation

https://docs.python.org/3/

Search

ceil

Go

Python 3.13.3 documentation

Welcome! This is the official documentation for Python 3.13.3.

Documentation sections:

- [What's new in Python 3.13?](#)
Or all "What's new" documents since Python 2.0
- [Tutorial](#)
Start here: a tour of Python's syntax and features
- [Library reference](#)
Standard library and builtins
- [Language reference](#)
Syntax and language elements
- [Python setup and usage](#)
How to install, configure, and use Python
- [Python HOWTOs](#)
In-depth topic manuals
- [Installing Python modules](#)
Third-party modules and PyPI.org
- [Distributing Python modules](#)
Publishing modules for use by other people
- [Extending and embedding](#)
For C/C++ programmers
- [Python's C API](#)
C API reference
- [FAQs](#)
Frequently asked questions (with answers!)
- [Deprecations](#)
Deprecated functionality

Die offizielle Dokumentation verwenden

- Wir müssen nun etwas warten.

Search — Python 3.13.3

https://docs.python.org/3/search.html?c

Search

Searching for multiple words only shows matches that contain all words.

© Copyright 2001-2025, Python Software Foundation.
This page is licensed under the Python Software Foundation License Version 2.
Examples, recipes, and other code in the documentation are additionally licensed under the Zero Clause BSD License.
See [History and License](#) for more information.

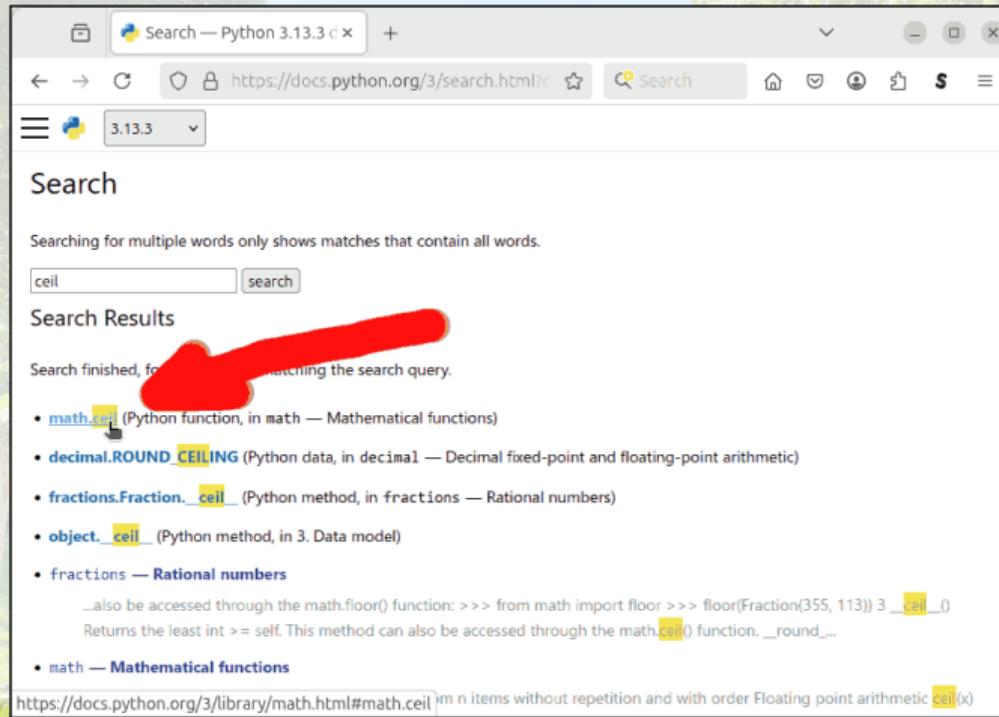
The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Apr 25, 2025 (14:16 UTC). [Found a bug?](#)
Created using [Sphinx](#) 8.2.3.

Transferring data from docs.python.org...

Die offizielle Dokumentation verwenden

- Die Ergebnisse erscheinen. Wir klicken auf `math.ceil`.



Search — Python 3.13.3 x

https://docs.python.org/3/search.html?c... Search

3.13.3

Search

Searching for multiple words only shows matches that contain all words.

search

Search Results

Search finished, for your search query.

- [math.ceil](#) (Python function, in `math` — Mathematical functions)
- [decimal.ROUND_CEILING](#) (Python data, in `decimal` — Decimal fixed-point and floating-point arithmetic)
- [fractions.Fraction.__ceil__](#) (Python method, in `fractions` — Rational numbers)
- [object.__ceil__](#) (Python method, in 3. Data model)
- [fractions — Rational numbers](#)

...also be accessed through the `math.floor()` function: >>> from math import floor >>> floor(Fraction(355, 113)) 3 [ceil\(\)](#)
Returns the least int \geq self. This method can also be accessed through the `math.ceil()` function. `_round_`...
- [math — Mathematical functions](#)

<https://docs.python.org/3/library/math.html#math.ceil>

Die offizielle Dokumentation verwenden

- Nun werden wir zur Beschreibung der Funktion gebracht.

The screenshot shows a web browser window displaying the Python 3.13.3 documentation for the `math` module. The URL is <https://docs.python.org/3/library/math.html>. The page lists several mathematical functions:

- `math.ceil(x)`**: Returns the ceiling of x , the smallest integer greater than or equal to x . If x is not a float, delegates to `x.__ceil__`, which should return an [Integral](#) value.
- `math.fabs(x)`**: Returns the absolute value of x .
- `math.floor(x)`**: Returns the floor of x , the largest integer less than or equal to x . If x is not a float, delegates to `x.__floor__`, which should return an [Integral](#) value.
- `math.fma(x, y, z)`**: Fused multiply-add operation. Returns $(x * y) + z$, computed as though with infinite precision and range followed by a single round to the `float` format. This operation often provides better accuracy than the direct expression $(x * y) + z$.

This function follows the specification of the fusedMultiplyAdd operation described in the IEEE 754 standard. The standard leaves one case implementation-defined, namely the result of `fma(0, inf, nan)` and `fma(inf, 0, nan)`. In these cases, `math.fma` returns a NaN, and does not raise any exception.

Added in version 3.13.
- `math.fmod(x, y)`**: Returns the floating-point remainder of x / y , as defined by the platform C library function `fmod(x, y)`. Note that the Python ex-

Die offizielle Dokumentation verwenden



- Angenommen, Sie kennen den Namen eines Kommandos und wollen mehr darüber erfahren, z. B. die genaue Bedeutung, Beschränkungen für die Werte der Parameter, mögliche Fehler, usw.
- Dafür gibt es einen kompletten Katalog mit Hilfe für Python: die *Python 3 Documentation*¹³ bei <https://docs.python.org/3>.
- Dies sollte Ihr erster Ansatzpunkt sein, wenn Sie mehr über Python lernen wollen.
- Diese Hilfe steht auch in Chinesisch bereit!
- Kennen wir den Namen der Funktion, die wir suchen, so finden wir bei der offiziellen Python Dokumentation die Beschreibung.



Suchen mit einer Suchmaschine



Suchen mit einer Suchmaschine

- Kennen wir den Namen der Funktion, die wir suchen, so finden wir bei der offiziellen Python Dokumentation die Beschreibung.



Suchen mit einer Suchmaschine



- Kennen wir den Namen der Funktion, die wir suchen, so finden wir bei der offiziellen Python Dokumentation die Beschreibung.
- Was aber, wenn wir den Namen **nicht** kennen?

Suchen mit einer Suchmaschine



- Kennen wir den Namen der Funktion, die wir suchen, so finden wir bei der offiziellen Python Dokumentation die Beschreibung.
- Was aber, wenn wir den Namen **nicht** kennen?
- Nehmen wir an, wir suchen eine Funktion, die Zahlen aufrundet.

Suchen mit einer Suchmaschine



- Kennen wir den Namen der Funktion, die wir suchen, so finden wir bei der offiziellen Python Dokumentation die Beschreibung.
- Was aber, wenn wir den Namen **nicht** kennen?
- Nehmen wir an, wir suchen eine Funktion, die Zahlen aufrundet.
- Versuchen wir es zuerst wieder mit der offiziellen Dokumentation.

Suchen mit einer Suchmaschine

- Versuchen wir es zuerst wieder mit der offiziellen Dokumentation.

A screenshot of a web browser window displaying the Python 3.13.3 documentation. The title bar says "3.13.3 Documentation". The address bar shows the URL "https://docs.python.org/3/". The main content area is titled "Python 3.13.3 documentation". Below it, a message reads "Welcome! This is the official documentation for Python 3.13.3." A sidebar on the left lists "Documentation sections:" including "What's new in Python 3.13?", "Tutorial", "Library reference", "Language reference", "Python setup and usage", and "Python HOWTOs". The main content area contains two columns of links: "Installing Python modules" (with subtext "Third-party modules and PyPI.org"), "Distributing Python modules" (with subtext "Publishing modules for use by other people"), "Extending and embedding" (with subtext "For C/C++ programmers"), "Python's C API" (with subtext "C API reference"), "FAQs" (with subtext "Frequently asked questions (with answers!)"), and "Deprecations" (with subtext "Deprecated functionality"). A red heart-shaped mark is drawn over the browser window's title bar.

3.13.3 Documentation

https://docs.python.org/3/

Python 3.13.3 documentation

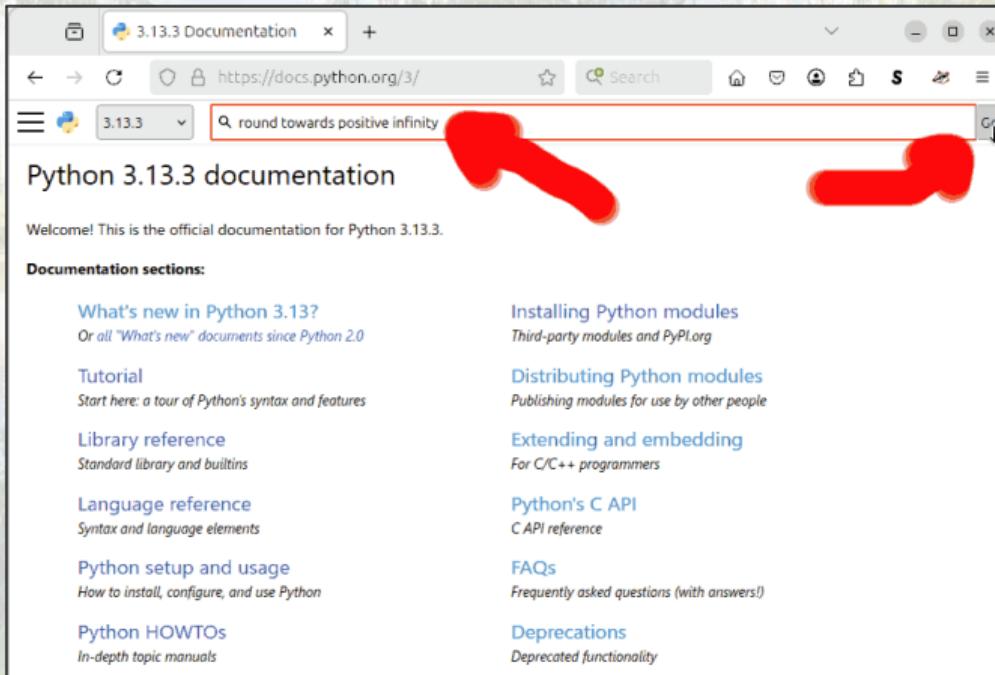
Welcome! This is the official documentation for Python 3.13.3.

Documentation sections:

- [What's new in Python 3.13?](#)
Or all "What's new" documents since Python 2.0
- [Tutorial](#)
Start here: a tour of Python's syntax and features
- [Library reference](#)
Standard library and builtins
- [Language reference](#)
Syntax and language elements
- [Python setup and usage](#)
How to install, configure, and use Python
- [Python HOWTOs](#)
In-depth topic manuals
- [Installing Python modules](#)
Third-party modules and PyPI.org
- [Distributing Python modules](#)
Publishing modules for use by other people
- [Extending and embedding](#)
For C/C++ programmers
- [Python's C API](#)
C API reference
- [FAQs](#)
Frequently asked questions (with answers!)
- [Deprecations](#)
Deprecated functionality

Suchen mit einer Suchmaschine

- Wir überlegen uns, wie man „aufrunden“ auf Englisch formulieren kann.



Suchen mit einer Suchmaschine



- Wir überlegen uns, wie man „aufrunden“ auf Englisch formulieren kann. Probieren wir es mit „round towards positive infinity“.

A screenshot of a web browser window displaying the Python 3.13.3 documentation. The URL in the address bar is <https://docs.python.org/3/>. A red box highlights the search bar, which contains the query "round towards positive infinity". Below the search bar, the page title "Python 3.13.3 documentation" is visible, along with a welcome message: "Welcome! This is the official documentation for Python 3.13.3." A red arrow points from the search bar to the search results area. The search results list several sections: "Documentation sections:", "What's new in Python 3.13?", "Tutorial", "Library reference", "Language reference", "Python setup and usage", "Python HOWTOs", "Installing Python modules", "Distributing Python modules", "Extending and embedding", "Python's C API", "FAQs", and "Deprecations". Each section has a brief description below it. Red arrows point from the search results to the "FAQs" and "Deprecations" sections.

3.13.3 Documentation

https://docs.python.org/3/

Search

3.13.3

round towards positive infinity

Go

Python 3.13.3 documentation

Welcome! This is the official documentation for Python 3.13.3.

Documentation sections:

- [What's new in Python 3.13?](#)
Or all "What's new" documents since Python 2.0
- [Tutorial](#)
Start here: a tour of Python's syntax and features
- [Library reference](#)
Standard library and builtins
- [Language reference](#)
Syntax and language elements
- [Python setup and usage](#)
How to install, configure, and use Python
- [Python HOWTOs](#)
In-depth topic manuals
- [Installing Python modules](#)
Third-party modules and PyPI.org
- [Distributing Python modules](#)
Publishing modules for use by other people
- [Extending and embedding](#)
For C/C++ programmers
- [Python's C API](#)
C API reference
- [FAQs](#)
Frequently asked questions (with answers!)
- [Deprecations](#)
Deprecated functionality

Suchen mit einer Suchmaschine



- Wir überlegen uns, wie man „aufrunden“ auf Englisch formulieren kann. Probieren wir es mit „round towards positive infinity“. Wir geben das in die Suchmaske ein und klicken **Go**.

The screenshot shows a web browser window with the title "3.13.3 Documentation". The URL in the address bar is <https://docs.python.org/3/>. The search bar contains the query "round towards positive infinity". A red hand icon points to the search bar, and another red hand icon points to the "Go" button. The page content is the Python 3.13.3 documentation homepage, featuring sections like "Documentation sections", "What's new in Python 3.13?", "Tutorial", "Library reference", "Language reference", "Python setup and usage", "Python HOWTOs", "Installing Python modules", "Distributing Python modules", "Extending and embedding", "Python's C API", "FAQs", and "Deprecations".

Suchen mit einer Suchmaschine

- Diesmal finden wir nichts nützliches.

A screenshot of a web browser window titled "Search — Python 3.13.3". The URL is <https://docs.python.org/3/search.html?q=round+towards+positive+infinity>. The search bar contains the query "round towards positive infinity". The results page shows a heading "Search Results" and a message "Search finished, found 7 pages matching the search query." Below this, there is a list of links:

- [decimal — Decimal fixed-point and floating-point arithmetic](#)
- [math — Mathematical functions](#) (highlighted with a large red oval)
- [sys — System-specific parameters and functions](#)
- [Built-in Functions](#)
- [Built-in Types](#)
- [Changelog](#)

Below the list, there is a note about a bug in pdb and bdb: "...the bug where pdb and bdb can step into the bottom caller frame. gh-100141: Fixed the bug where pdb will be stuck in an infinite loop when debugging an empty file. gh-125115: Fixed a bug in pdb where arguments starting with - can't be passed..."

The footer of the page shows a link to "[What's New in Python 2.6](https://docs.python.org/3/library/math.html)" and the URL <https://docs.python.org/3/library/math.html>.

Suchen mit einer Suchmaschine



- Diesmal finden wir nichts nützliches. Wir könnten zwar auf „math – Mathematical functions“ klicken und uns durch den ganzen Katalog mathematischer Funktionen durcharbeiten ... aber das ist vielleicht zu umständlich.

A screenshot of a web browser window titled "Search — Python 3.13.3". The URL is <https://docs.python.org/3/search.html?q=round+towards+positive+infini>. The search bar contains "round towards positive infini". The search results page shows a list of links:

- [decimal — Decimal fixed-point and floating-point arithmetic](#)
- [**math — Mathematical functions**](#) (This link is highlighted with a large red arrow pointing to it)
- [sys — System-specific parameters and functions](#)
- [Built-in Functions](#)
- [Built-in Types](#)
- [Changelog](#)

Below the list, there is a note about a bug in pdb and bdb: "...the bug where pdb and bdb can step into the bottom caller frame. gh-100141: Fixed the bug where pdb will be stuck in an infinite loop when debugging an empty file. gh-125115: Fixed a bug in pdb where arguments starting with - can't be passed..."

At the bottom of the page, there is a link to "What's New in Python 2.6".

The footer of the browser window shows the URL <https://docs.python.org/3/library/math.html>.

Suchen mit einer Suchmaschine



- Öffnen wir stattdessen eine Suchmaschine (hier: <https://bing.com>) und geben die selbe Frage dort ein.

python rounds towards positive infinity

ALL IMAGES VIDEOS TOOLS

About 131,000 results

 Real Python
<https://realpython.com/python-rounding>

[How to Round Numbers in Python](#)
Dec 7, 2024 · The round_half_up() function introduces a round toward positive infinity bias, and round_half_down() introduces a round toward negative infinity bias. The remaining rounding strategies all attempt to mitigate these biases in different ways.

Estimated Reading Time: 6 mins

[Take The Quiz](#)
Test your knowledge of rounding numbers in Python. Python Tutorials → In-depth articles and video courses Learning Paths → Guided study plan...

[See results only from realpython.com](#)

Suchen mit einer Suchmaschine



- Öffnen wir stattdessen eine Suchmaschen (hier: <https://bing.com>) und geben die selbe Frage dort ein. Gleich der erste Link sieht interessant aus.

A screenshot of a Microsoft Bing search results page. The search query in the bar is "python rounds towards positive infinity". A red arrow points to the search bar. Another red arrow points to the first search result, which is a link from "Real Python" to "How to Round Numbers in Python". Below the result, there is a snippet of text and an estimated reading time of 6 minutes. At the bottom, there is a link to "Take The Quiz".

python rounds towards positive infinity

ALL IMAGES VIDEOS TOOLS

About 131,000 results

Real Python
<https://realpython.com/python-rounding>

How to Round Numbers in Python

Dec 7, 2024 · The round_half_up() function introduces a round toward positive infinity bias, and round_half_down() introduces a round toward negative infinity bias. The remaining rounding strategies all attempt to mitigate these biases in different ways.

Estimated Reading Time: 6 mins

[Take The Quiz](#)

Test your knowledge of rounding numbers in Python. Python Tutorials → In-depth articles and video courses Learning Paths → Guided study plan...

[See results only from realpython.com](#)

Suchen mit einer Suchmaschine

- Er bringt uns zu einer Webseite über das Runden in Python: "How to Round Numbers in Python"² unter <https://realpython.com/python-rounding>.



Suchen mit einer Suchmaschine

- Wenn wir etwas herunterscrollen finden wir die gesuchte Information.

A screenshot of a web browser window. The address bar shows the query "python rounds towards p" and the result page from "How to Round Numbers" on RealPython.com. The page title is "Rounding Up". It explains that rounding up means always rounding a number up to a specified number of digits. A table illustrates this strategy:

Value	Round Up To	Result
12.345	Tens place	20
12.345	Ones place	13
12.345	Tenths place	12.4
12.345	Hundredths place	12.35

To implement the rounding up strategy in Python, you'll use the `ceil()` function from the `math module`.

Suchen mit einer Suchmaschine



- Wenn wir etwas herunterscrollen finden wir die gesuchte Information. Sogar einen Link zur offiziellen Dokumentation.

A screenshot of a web browser window. The address bar shows the URL <https://realpython.com/python-rounding/>. The page content is from Real Python and discusses rounding numbers. A red highlight is drawn over the word 'ceil()' in the text at the bottom.

python rounds towards p x How to Round Numbers i x

Real Python

Rounding Up

The second rounding strategy that you'll look at is called **rounding up**. This strategy always rounds a number up to a specified number of digits. The following table summarizes this strategy:

Value	Round Up To	Result
12.345	Tens place	20
12.345	Ones place	13
12.345	Tenths place	12.4
12.345	Hundredths place	12.35

To implement the rounding up strategy in Python, you'll use the `ceil()` function from the `math module`.

Suchen mit einer Suchmaschine



- Wir folgen diesem Link. Wir schlagen **immer** alle gefundenen Informationen nochmal in der offiziellen Dokumentation nach.

The screenshot shows a web browser window with several tabs open. The active tab is the Python documentation for the `math` module, specifically the `ceil` function. The URL is <https://docs.python.org/3/library/math.html>. The page content includes:

- math.ceil(x)**: Returns the ceiling of x , the smallest integer greater than or equal to x . If x is not a float, delegates to `x.__ceil__`, which should return an [Integral](#) value.
- math.fabs(x)**: Returns the absolute value of x .
- math.floor(x)**: Returns the floor of x , the largest integer less than or equal to x . If x is not a float, delegates to `x.__floor__`, which should return an [Integral](#) value.
- math.fma(x, y, z)**: Fused multiply-add operation. Returns $(x * y) + z$, computed as though with infinite precision and range followed by a single round to the `float` format. This operation often provides better accuracy than the direct expression $(x * y) + z$.

This function follows the specification of the fusedMultiplyAdd operation described in the IEEE 754 standard. The standard leaves one case implementation-defined, namely the result of `fma(0, inf, nan)` and `fma(inf, 0, nan)`. In these cases, `math.fma` returns a NaN, and does not raise any exception.

Added in version 3.13.
- math.fmod(x, y)**: Returns the floating-point remainder of x / y , as defined by the platform C library function `fmod(x, y)`. Note that the Python expression

Suchen mit einer Suchmaschine



- Wir folgen diesem Link. Wir schlagen **immer** alle gefundenen Informationen nochmal in der offiziellen Dokumentation nach.
- Wir können also durchaus mit einer Suchmaschine auch Funktionen finden, deren Namen wir nicht kennen.



Suchen mit Community-Portalen



Suchen mit Community-Portalen

- Es gibt sehr viele Community-Portale rund ums Programmieren.





Suchen mit Community-Portalen

- Es gibt sehr viele Community-Portale rund ums Programmieren.
- Auch in diesen Portalen können wir nach nützlichen Informationen suchen.



Suchen mit Community-Portalen

- Es gibt sehr viele Community-Portale rund ums Programmieren.
- Auch in diesen Portalen können wir nach nützlichen Informationen suchen.
- Probieren wir das mal aus und besuchen das bekannte Portal *Stack Overflow*²² bei <https://stackoverflow.com>.

Suchen mit Community-Portalen

- Probieren wir das mal aus und besuchen das bekannte Portal *Stack Overflow*²² bei <https://stackoverflow.com>.

The screenshot shows the Stack Overflow homepage. A red arrow points from the top-left towards the search bar. Another red arrow points from the top-right towards the user profile area. The profile area displays the user's name, Thomas Weise, their reputation of 441, and badge progress.

Welcome back, Thomas Weise
Find answers to your technical questions and help others answer theirs.

Reputation
441 +162
2022 2025
Next privilege at 500 rep: [Access review queues](#)

Badge progress
Critic - 0/1
First down vote.

Watched tags
java latex
mathematical-optimization

Interesting posts for you
Based on your viewing history and watched tags. [Customize your feed](#)

0 votes Why read barrier cannot resolve the "Stop The World" in the ZGC garbage collector?
0 answers
59 views
I've seen some similar posts, but I did not find a exact answer. So please remember that this question is about...

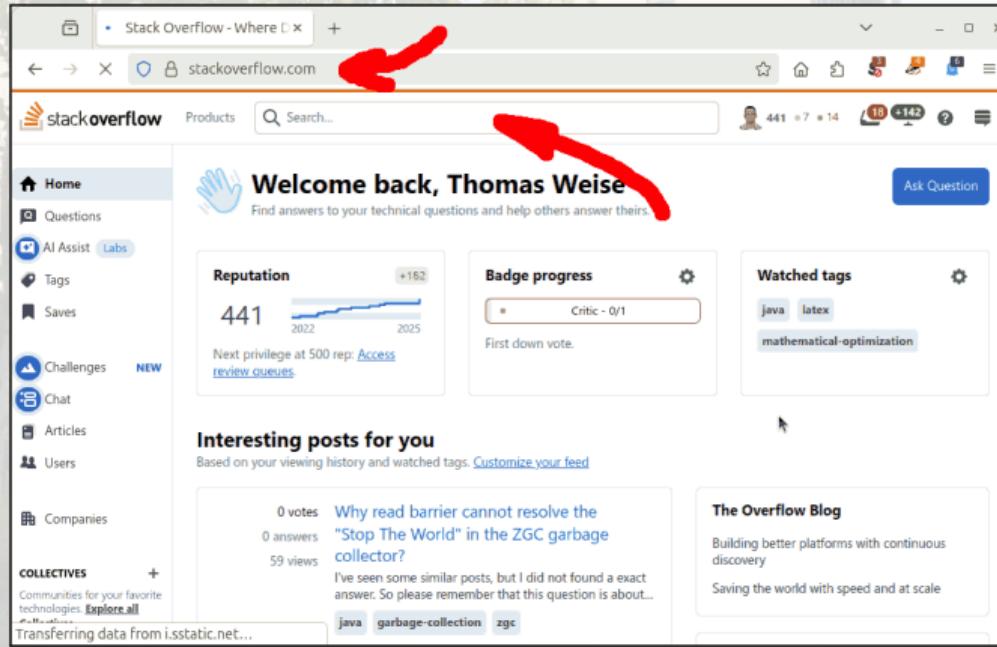
The Overflow Blog
Building better platforms with continuous discovery
Saving the world with speed and scale

java garbage-collection zgc

Transferring data from ssstatic.net...

Suchen mit Community-Portalen

- Warnung: Manchmal fordert uns diese Webseite auf, weitere JavaScripts von anderen Quellen zu laden. Wenn das passiert dann machen wir das.



Suchen mit Community-Portalen



- Wenn die Webseite voll geladen ist, können wir unsere Frage eingeben.

A screenshot of the Stack Overflow homepage. A large red arrow points to the search bar at the top center of the page. The search bar contains the placeholder text "Search...". The page features a sidebar on the left with links to Home, Questions, AI Assist, Tags, Saves, Challenges, Chat, Articles, Users, and Companies. Below the sidebar is a section for "COLLECTIVES" with a "+". The main content area includes a search bar with various search operators like "[tag]", "user:1234", and "is:question", and a "Reputation" section showing 44. There are sections for "Interesting posts for you" and "The Overflow Blog". A "Watched tags" sidebar on the right lists "java", "latex", and "mathematical-optimization".

Suchen mit Community-Portalen



- Wir probieren es diesmal mit „python round towards infinity“.

A screenshot of a web browser showing the Stack Overflow homepage. The search bar at the top contains the query "python round towards infinity". A red arrow points from the text "enter query, then press [enter]" to the search bar. Below the search bar, a dropdown menu shows various search operators and their meanings. The main content area displays "Interesting posts for you" and "The Overflow Blog".

Stack Overflow - Where do you code? stackoverflow.com

python round towards infinity

[tag] search within a tag
user:1234 search by author
"words here" exact phrase
collective:"Name" collective content

answers:0 unanswered questions
score:3 posts with a 3+ score
is:question type of post
isaccepted:yes search within status

Reput 44 Ask a question

Watched tags java latex mathematical-optimization

Next privilege at 500 rep: Access review queues

Interesting posts for you

Based on your viewing history and watched tags. Customize your feed

Why read barrier cannot resolve the "Stop The World" in the ZGC garbage collector?

0 votes 0 answers 59 views I've seen some similar posts, but I did not find a exact answer. So please remember that this question is about... java garbage-collection zgc

The Overflow Blog

Building better platforms with continuous discovery Saving the world with speed and at scale

enter query,
then press [enter]

Suchen mit Community-Portalen

- Alle Fragen, die der Community gestellt wurden, und die so ähnlich klingen, werden gelistet.

Posts matching 'python r...' × +

stackoverflow.com/search?q=python+rounding+towards+infinity

stackoverflow Products python rounding towards infinity 441 7 14 18 142 Ask Question

Home Questions AI Assist Tags Saves Challenges NEW Chat Articles Users Companies COLLECTIVES + Communities for your favorite technologies Explore all

Search Results Advanced Search Tips Ask Question

Results for python rounding towards infinity
Search options not deleted

45 results Relevance Newest More ▾

88 votes ⓘ What's the mathematical reason behind Python choosing to round integer division toward negat...
✓ 8 answers 9k views

I know Python // rounds towards negative infinity and in C++ / is truncating, rounding towards 0. ... (m/n)^n + m%in...

python c++ python-3.x rounding integer-division

8 votes ⓘ How to implement division with round-towards-infinity in Python
8 answers 8k views

I want 3/2 to equal 2 not 1.5 I know there's a mathematical term for that operation(not called rounding up), but I can't...

Rick 7,636 asked Jan 16, 2022 at 14:07

Hot Network Questions

Why aren't there baleen whales at about the size of a dolphin?
Film or TV series about powerful man who is banished, fishes beings he creates from ponds
Has the Silver Surfer's clothing (or lack thereof) ever been addressed in the comics?
Can you identify this aircraft landing at Llanbedr

Transferring data from ssstatic.net...

then math floating-point rounding division

Suchen mit Community-Portalen

- Alle Fragen, die der Community gestellt wurden, und die so ähnlich klingen, werden gelistet. Wir müssen etwas herunterscrollen, um eine Frage zu finden, die passend klingt.

Posts matching 'python r...' × +

stackoverflow.com/search?q=python+rounding+towards+infinity

stackoverflow Products python rounding towards infinity

441 7 14 18 142

Home Questions AI Assist Tags Saves Challenges NEW Chat Articles Users Companies COLLECTIVES + Communities for your favorite technologies Explore all

Search Results Advanced Search Tips Ask Question

Results for python rounding towards infinity
Search options not deleted

45 results Relevance Newest More ▾

88 votes What's the mathematical reason behind Python choosing to round integer division toward negat...
✓ 8 answers 9k views

I know Python // rounds towards negative infinity and in C++ / is truncating, rounding towards 0. ... (m/n)"n + m%in...

python c++ python-3.x rounding integer-division

Rick 7,636 asked Jan 16, 2022 at 14:07

8 votes How to implement division with round-towards-infinity in Python
8 answers 8k views

I want 3/2 to equal 2 not 1.5 I know there's a mathematical term for that operation(not called rounding up), but I can't...

then math floating-point rounding division

Hot Network Questions

Why aren't there baleen whales at about the size of a dolphin?

Film or TV series about powerful man who is banished, fishes beings he creates from ponds

Has the Silver Surfer's clothing (or lack thereof) ever been addressed in the comics?

Can you identify this aircraft landing at Llanbedr

Transferring data from ssstatic.net...

Suchen mit Community-Portalen



- Wir finden die Frage "How to implement division with round-towards-infinity in Python"⁷, welche passend klingt, und klicken darauf.

Posts matching 'python r...' +

stackoverflow.com/search?page=1&tab=Relevance&pagesize=15&q=python round

Questions

Home

AI Assist

Tags

Saves

Challenges NEW

Chat

Articles

Users

Companies

COLLECTIVES +

Communities for your favorite technologies: [Explore all](#)

python rounding towards infinity

ilmari Karonen 50.5k answered Jan 16, 2022 at 23:18

0 votes A Extract day of year and Julian day from string date

Python, for example, implements rounding towards `-infinity`, which is quite different... To use the formulae given in...

python date datetime julian-date

rfc reddish 1,410 answered Apr 8, 2022 at 7:44

4 votes A How to implement division with round-towards-infinity in Python

The intent of the OP's question is "How to implement division with `round-towards-infinity` in Python" (suggest you...

python math floating-point rounding division

34.1k answered Aug 24, 2011 at 22:19

0 votes A Is there a better or faster solution to round up or down from half in python without any lib...

In python the function `round()` just rounds half to the near...

stackoverflow.com/questions/7181757/how-to-implement-division-with-round-towards-infinity-in-python/7183004?r=SearchResults#7183004

Suchen mit Community-Portalen



- Die Frage klingt tatsächlich passend, ist aber eigenartig.

A screenshot of a web browser showing a Stack Overflow question page. The URL in the address bar is stackoverflow.com/questions/7181757/how-to-implement-division-with-rounding-towards-infinity. The question title is "How to implement division with round-towards-infinity in Python". The question was asked 13 years, 11 months ago and has 8k views. A user with 8 reputation points has commented: "I want $3/2$ to equal 2 not 1.5". Another user has responded: "I know there's a mathematical term for that operation(not called rounding up), but I can't recall it right now. Anyway, how do i do that without having to do two functions?". A third user has asked: "ex of what I do NOT want: answer = $3/2$ then math.ceil(answer)=2 (why does math". A fourth user has asked: "ex of what I DO want: "function"($3/2$) = 2". A red arrow points from the bottom left towards the question text. A red question mark is placed next to the user's response about the mathematical term. The sidebar on the left shows navigation links like Home, Questions, AI Assist, Tags, Saves, Challenges, Chat, Articles, Users, and Companies. The bottom navigation bar includes links for python, math, floating-point, rounding, and division. The right sidebar features "The Overflow Blog" and "Featured on Meta" sections.

Suchen mit Community-Portalen

- Die Frage klingt tatsächlich passend, ist aber eigenartig. Es wird z. B. geschrieben, dass $\frac{3}{2}$ eins ergibt, was unserem Verständnis des Divisionsoperators widerspricht.

A screenshot of a web browser displaying a Stack Overflow question. The question is titled "How to implement division with round-towards-infinity in Python". It was asked 13 years, 11 months ago, modified 13 years, 11 months ago, and viewed 8k times. A user with 8 upvotes has commented: "I want $3/2$ to equal 2 not 1.5. I know there's a mathematical term for that operation(not called rounding up), but I can't recall it right now. Anyways, how do I do that without having to do two functions? ex of what I do NOT want: answer = $3/2$ then math.ceil(answer)=2 (why does math.ceil round towards zero?)" A red arrow points down to this comment. Another user has responded with: "ex of what I DO want: "function"($3/2$) = 2". A red question mark is placed next to this response. The browser's sidebar shows navigation links like Home, Questions, AI Assist, Tags, Saves, Challenges, Chat, Articles, Users, and Companies. The bottom of the page shows tags: python, math, floating-point, rounding, division. On the right, there are sections for "The Overflow Blog" and "Featured on Meta".

Suchen mit Community-Portalen



- Weiter unten stellen wir fest, dass die Frage für Python 2 gestellt war, nicht für Python 3. Unter Python 2 funktionierte der Operator `/` wir der `//`-Operator unter Python 3³⁶.

The screenshot shows a web browser window with the URL stackoverflow.com/questions/7181757/how-to-implement-division-with-rounding. The page content discusses the difference between float and int division in Python 2. It mentions that if both inputs are integers, Python performs "round down" or "floor integer" division (e.g., `3/2==1`, but `-3/2== -2`). This is equivalent to `int(math.floor(float(a)/b))`. The text then states that if at least one input is a float, Python performs "true" division, giving a float result (e.g., `3.0/2==1.5`, and `-3.0/2== -1.5`). It notes that `float(a)/b` is being used to force true division even if both inputs are integers (e.g., `float(3)/2==1.5`). This is why the example `math.ceil(3/2)` returns `1.0`, whereas `math.ceil(float(3)/2)` returns `2.0`. The result has already been rounded down before it even reaches `math.ceil()`.

"true division" by default: In 2001, it was decided ([PEP 238](#)) that Python's division operator should be changed so that it always performs "true" division, regardless of whether the inputs are floats or integers (e.g., this would make `3/2==1.5`). In order to not break existing scripts, the change in default behavior was deferred until Python 3.0; in order to get this

The sidebar on the left includes links for Home, Questions, AI Assist, Tags, Saves, Challenges, Chat, Articles, Users, and Companies. The right sidebar lists various Stack Overflow questions and answers.

Suchen mit Community-Portalen



- Weiter unten stellen wir fest, dass die Frage für Python 2 gestellt war, nicht für Python 3. Unter Python 2 funktionierte der Operator `/` wir der `//`-Operator unter Python 3³⁶. Wir finden aber eine gute Erklärung und einen Verweis auf `ceil`, was unser Problem löst.

Suchen mit Community-Portalen

- Auch mit Community-Portalen können wir also gute Antworten auf unsere Fragen finden.



Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden.

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt, und
 3. über Community-Portale.

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt, und
 3. über Community-Portale.
- Nur die offizielle Dokumentation unserer Programmiersprache ist autoritativ.

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt, und
 3. über Community-Portale.
- Nur die offizielle Dokumentation unserer Programmiersprache ist autoritativ.
- Nur der offiziellen Dokumentation können wir voll vertrauen.

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt, und
 3. über Community-Portale.
- Nur die offizielle Dokumentation unserer Programmiersprache ist autoritativ.
- Nur der offiziellen Dokumentation können wir voll vertrauen.
- Alle anderen Quellen sind von irgendwelchen anderen Leuten geschrieben.

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt, und
 3. über Community-Portale.
- Nur die offizielle Dokumentation unserer Programmiersprache ist autoritativ.
- Nur der offiziellen Dokumentation können wir voll vertrauen.
- Alle anderen Quellen sind von irgendwelchen anderen Leuten geschrieben.
- Sie können falsch sein.

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt, und
 3. über Community-Portale.
- Nur die offizielle Dokumentation unserer Programmiersprache ist autoritativ.
- Nur der offiziellen Dokumentation können wir voll vertrauen.
- Alle anderen Quellen sind von irgendwelchen anderen Leuten geschrieben.
- Sie können falsch sein. Oder veraltet.

Zwischenstand



- Wir können nun Antworten auf unsere Fragen auf drei Wegen finden:
 1. über die offizielle Dokumentation,
 2. über eine Suchmaschine, die uns zu anderen Webseiten im Internet bringt, und
 3. über Community-Portale.
- **Nur die offizielle Dokumentation unserer Programmiersprache ist autoritativ.**
- Nur der offiziellen Dokumentation können wir voll vertrauen.
- Alle anderen Quellen sind von irgendwelchen anderen Leuten geschrieben.
- Sie können falsch sein. Oder veraltet.
- Alles, was wir in den Quellen finden, muss also in der offiziellen Dokumentation nachgeprüft werden.



Suchen mit AI



Suchen mit AI: Ausprobieren

- Können wir auch mit einer Artificial Intelligence (AI) nach Antworten suchen?



Suchen mit AI: Ausprobieren

- Können wir auch mit einer Artificial Intelligence (AI) nach Antworten suchen?
- Ja klar, warum nicht.



Suchen mit AI: Ausprobieren

- Können wir auch mit einer Artificial Intelligence (AI) nach Antworten suchen?
- Ja klar, warum nicht.
- Probieren wir das mal aus.



Suchen mit AI: Ausprobieren

- Wir öffnen die BaiDu (百度) Suchseite unter <https://www.baidu.com> und klicken auf „AI搜索已接入DeepSeek-V3最新版模型“.



Suchen mit AI: Ausprobieren

- Wir kommen auf die Seite „Hi, AI搜索已支持R1满血版，快来试试吧！“, auf der wir unseren Prompt für das DeepSeek-R1 Modell⁶ in das Eingabefeld unten eintippen können



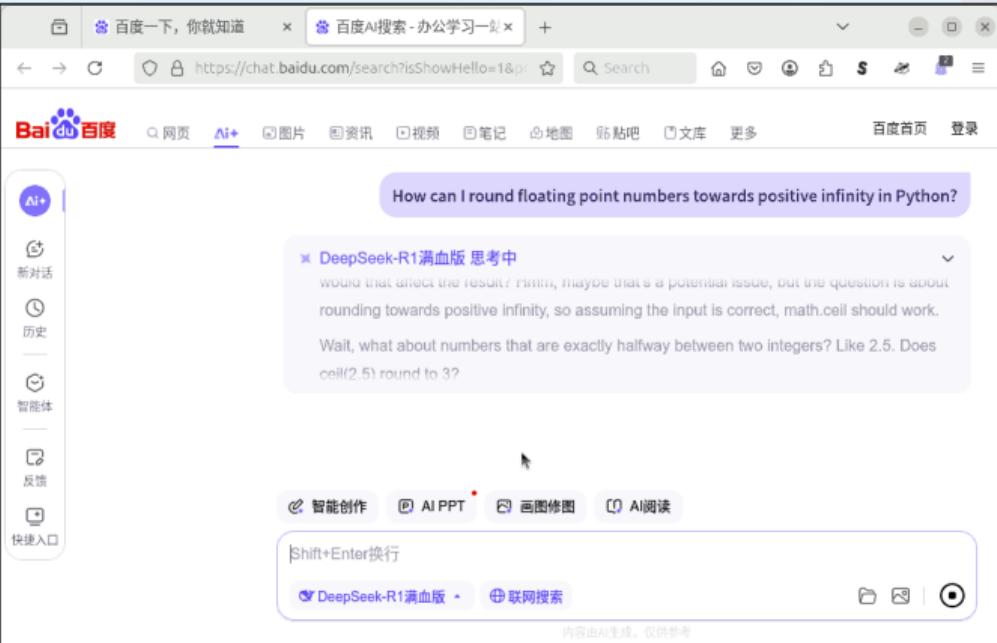
Suchen mit AI: Ausprobieren

- Wir schreiben „How can I round floating point numbers towards positive infinity in Python?“ und drücken ↴.



Suchen mit AI: Ausprobieren

- Die AI holt zu einer langatmigen Antwort aus.



Suchen mit AI: Ausprobieren

- Diese Antwort beinhaltet auch die richtige Lösung.

The screenshot shows a web browser window with the URL <https://chat.baidu.com/search?isShowHello=1&pc>. The page content is as follows:

To round a floating point number towards positive infinity (ceiling) in Python, you can use the `math.ceil()` function from the `math` module. This function returns the smallest integer greater than or equal to the given number, which effectively rounds the number up. Here's how to do it:

1. Import the `math` module.
2. Use `math.ceil()` with the number you want to round.

Example Code:

```
python
1 import math
```

[Copy Code](#)

At the bottom of the code editor, there are several AI-related buttons: 智能创作, AI PPT, 画图修图, AI 阅读. Below the code editor, there is a text input field with the placeholder "Shift+Enter换行" and a toolbar with icons for file operations (Save, Open, etc.) and a blue upward arrow button.

Suchen mit AI: Ausprobieren

- Sogar mit einem Beispiel!

The screenshot shows a Baidu search results page for "百度AI搜索 - 办公学习一". The main content area displays the following text:

1. Import the math module.
2. Use `math.ceil()` with the number you want to round.

Example Code:

```
python
1 import math
2
3 number = 2.3
4 rounded = math.ceil(number)
5 print(rounded) # Output: 3
6
```

On the right side, there is a sidebar with AI-related features and a Python 3 icon. At the bottom, there is a text input field with a "Shift+Enter换行" placeholder and a "DeepSeek-R1满血版" link.

Suchen mit AI: Ausprobieren

- Die vollständige Antwort ist lang und umständlich.

Teil der Ausgabe der DeepSeek-Suche von Baidu vom 2025-04-27, manuell hinzugefügte Formatierung.

Okay, I need to figure out how to round floating point numbers towards positive infinity in Python. Let's see, rounding towards positive infinity is also known as ceiling. So, maybe there's a function called `ceil` in Python?

Wait, I remember that Python has a `math` module. Let me check. Yes, the `math` module has a `ceil` function. So, `math.ceil()` should do the trick. But wait, how does it handle different numbers? For example, if the number is already an integer, does it leave it as is? And for negative numbers, does it round them up towards zero or away?

Let me think. The ceiling function in mathematics rounds a number up to the nearest integer. So, for positive numbers, even if they are already integers, it stays the same. For negative numbers, it would round towards zero. For example, `ceil(-2.3)` would be -2, which is greater than -2.3. So that's correct for positive infinity direction.

But how about when dealing with floating points that are very close to an integer due to precision issues? Like `2.0000000001`, would `ceil` round it to 3? Probably not, because the function should handle that as per the actual value. Wait, but floating points can have precision problems. For instance, `1.1` might not be represented exactly. So `ceil(1.1)` should give 2, but if due to some calculation the number is actually stored as slightly less than `1.1`, would that affect the result? Hmm, maybe that's a potential issue, but the question is about rounding towards positive infinity, so assuming the input is correct, `math.ceil` should work.

Wait, what about numbers that are exactly halfway between two integers? Like `2.5`. Does `ceil(2.5)` round to 3? Yes, because `ceil` rounds up regardless of the fractional part. So, even `2.1` would go to 3, right? No, wait: `ceil(2.1)` is 3? No, wait no. Wait, `ceil(2.1)` is 3? Wait no! Wait, no, that's not right. Wait, the ceiling function rounds up to the smallest integer greater than or equal to the number. So, `2.1`'s ceiling is 3? No, wait, `2.1` is between 2 and 3. The smallest integer greater than or equal to `2.1` is 3? Wait, no. Wait `2.1` is greater than 2, so the smallest integer greater or equal is 3? Wait, no, wait, `2.1`'s ceiling is 3? No, no. Wait, `2.1` is between 2 and 3. So the ceiling is 3. Wait, but `2.0`'s ceiling is 2.0. So yes, `ceil(2.1)` would be 3? Wait, wait, I'm getting confused. Let me think again.

Wait, the ceiling of `2.1` is 3? No, that's not right. Wait, the ceiling function rounds up to the next integer. So `2.1` would have a ceiling of 3? No, wait no. Wait, no, the ceiling of `2.1` is 3? No, that's not correct. Wait, `2.1` is `2.1`, so the smallest integer greater than or equal to `2.1` is 3? No! Wait, 2 is less than `2.1`, but 3 is greater than `2.1`. Wait, no, `2.1` is between 2 and 3. The ceiling is the smallest integer that is greater than or equal. So 3 is greater than `2.1`, but 2 is less than `2.1`. So the ceiling would be 3? No, that's not right. Wait, no, `2.1` is greater than 2, so the ceiling is the smallest integer that is not less than the number. So the ceiling is 3? Wait, no. Wait, `2.1` is `2.1`, so the next integer up is 3. So yes, `ceil(2.1)` is 3? Wait, no. Wait, I think I made a mistake here. Because the ceiling of `2.1` is 3? No, the ceiling of `2.1` is actually 3. Wait, no, that's not correct. Wait, the ceiling of `2.0` is `2.0`. The ceiling of `2.0` is `2.0`. For `2.1`, the next integer higher is `3.0`. So `ceil(2.1)` is `3.0`. Wait, but that's not right. Wait, `2.1` is less than `3.0`, but the ceiling is the smallest integer greater than or equal. So for example, `ceil(2.0)` is `2.0`, `ceil(2.1)` is `3.0`? That can't be right. Wait, maybe I'm mixing up ceiling and floor functions. Wait, the floor function rounds down. So `floor(2.1)` is `2`. The ceiling function rounds up. So `ceil(2.1)` is `3.0`? That would be the case. But that's a big jump, but yes. For example, in Python:

```
import math  
math.ceil(2.1)  
3
```

Is that true? Let me check. Wait, no, that's not correct. Wait, I'm confusing something here. Let me check an example. Let me try to recall: `ceil(1.2)` is 2? No. Wait, no. Wait, `ceil(1.2)` is 2. Yes, because 2 is the smallest integer greater than or equal to `1.2`. Wait, no. Wait, `1.2` is between 1 and 2. So the smallest integer greater than or equal to `1.2` is 2? No, that's not correct. Wait, `1.2` is greater than 1, but 2 is larger than `1.2`, but the "smallest integer greater than or equal to" `1.2` is 2? No. Wait, no. Wait, the integers around `1.2` are 1 and 2. `1.2` is greater than 1, so the smallest integer that is greater than or equal is 2? No. Wait, no. Wait, `1.2` is `1.2`, so the next integer higher is 2. So the ceiling is 2. Yes. So `ceil(1.2)` is 2. So in that case, yes, that's correct.

Wait, but then for numbers like `2.5`, `ceil(2.5)` is 3. So regardless of the decimal part, it rounds up. So that's correct.

But in Python, how does this work? Let me think of code examples. If I do:

```
import math  
print(math.ceil(2.1)) # should output 3
```

Wait, but when I run this code, what's the actual output? Wait, maybe I'm making a mistake here. Let me check.

Suchen mit AI: Ausprobieren

- Sie beinhaltet aber auch die **richtige Antwort**, die `ceil` Funktion.

Teil der Ausgabe der DeepSeek-Suche von Baidu vom 2025-04-27, manuell hinzugefügte Formatierung.



Okay, I need to figure out how to round floating point numbers towards positive infinity in Python. Let's see, rounding towards positive infinity is also known as ceiling. So, maybe there's a function called `ceil` in Python?

Wait, I remember that Python has a `math` module. Let me check. Yes, the `math` module has a `ceil` function. So, `math.ceil()` should do the trick. But wait, how does it handle different numbers? For example, if the number is already an integer, does it leave it as is? And for negative numbers, does it round them up towards zero or away?

Let me think. The ceiling function in mathematics rounds a number up to the nearest integer. So, for positive numbers, even if they are already integers, it stays the same. For negative numbers, it would round towards zero. For example, `ceil(-2.3)` would be -2, which is greater than -2.3. So that's correct for positive infinity direction.

But how about when dealing with floating points that are very close to an integer due to precision issues? Like `2.0000000001`, would `ceil` round it to 3? Probably not, because the function should handle that as per the actual value. Wait, but floating points can have precision problems. For instance, `1.1` might not be represented exactly. So `ceil(1.1)` should give 2, but if due to some calculation the number is actually stored as slightly less than `1.1`, would that affect the result? Hmm, maybe that's a potential issue, but the question is about rounding towards positive infinity, so assuming the input is correct, `math.ceil` should work.

Wait, what about numbers that are exactly halfway between two integers? Like `2.5`. Does `ceil(2.5)` round to 3? Yes, because `ceil` rounds up regardless of the fractional part. So, even `2.1` would go to 3, right? No, wait: `ceil(2.1)` is 3? No, wait no. Wait, `ceil(2.1)` is 3? Wait no! Wait, no, that's not right. Wait, the ceiling function rounds up to the smallest integer greater than or equal to the number. So, `2.1`'s ceiling is 3? No, wait, `2.1` is between 2 and 3. The smallest integer greater than or equal to `2.1` is 3? Wait, no. Wait `2.1` is greater than 2, so the smallest integer greater or equal is 3? Wait, no, wait, `2.1`'s ceiling is 3? No, no. Wait, `2.1` is between 2 and 3. So the ceiling is 3. Wait, but `2.0`'s ceiling is 2.0. So yes, `ceil(2.1)` would be 3? Wait, wait, I'm getting confused. Let me think again.

Wait, the ceiling of `2.1` is 3? No, that's not right. Wait, the ceiling function rounds up to the next integer. So `2.1` would have a ceiling of 3? No, wait no. Wait, no, the ceiling of `2.1` is 3? No, that's not correct. Wait, `2.1` is `2.1`, so the smallest integer greater than or equal to `2.1` is 3? No! Wait, 2 is less than `2.1`, but 3 is greater than `2.1`. Wait, no, `2.1` is between 2 and 3. The ceiling is the smallest integer that is greater than or equal. So 3 is greater than `2.1`, but 2 is less than `2.1`. So the ceiling would be 3? No, that's not right. Wait, no, `2.1` is greater than 2, so the ceiling is the smallest integer that is not less than the number. So the ceiling is 3? Wait, no. Wait, `2.1` is `2.1`, so the next integer up is 3. So yes, `ceil(2.1)` is 3? Wait, no. Wait, I think I made a mistake here. Because the ceiling of `2.1` is 3? No, the ceiling of `2.1` is actually 3. Wait, no, that's not correct. Wait, the ceiling of `2.0` is 2.0. The ceiling of `2.0` is 2.0. For `2.1`, the next integer higher is 3.0. So `ceil(2.1)` is 3.0. Wait, but that's not right. Wait, `2.1` is less than 3.0, but the ceiling is the smallest integer greater than or equal. So for example, `ceil(2.0)` is 2.0, `ceil(2.1)` is 3.0? That can't be right. Wait, maybe I'm mixing up ceiling and floor functions. Wait, the floor function rounds down. So `floor(2.1)` is 2. The ceiling function rounds up. So `ceil(2.1)` is 3.0? That would be the case. But that's a big jump, but yes. For example, in Python:

```
import math  
math.ceil(2.1)  
3
```

Is that true? Let me check. Wait, no, that's not correct. Wait, I'm confusing something here. Let me check an example. Let me try to recall: `ceil(1.2)` is 2? No. Wait, no. Wait, `ceil(1.2)` is 2. Yes, because 2 is the smallest integer greater than or equal to `1.2`. Wait, no. Wait, `1.2` is between 1 and 2. So the smallest integer greater than or equal to `1.2` is 2? No, that's not correct. Wait, `1.2` is greater than 1, but 2 is larger than `1.2`, but the "smallest integer greater than or equal to" `1.2` is 2? No. Wait, no. Wait, the integers around `1.2` are 1 and 2. `1.2` is greater than 1, so the smallest integer that is greater than or equal is 2? No. Wait, no. Wait, `1.2` is `1.2`, so the next integer higher is 2. So the ceiling is 2. Yes. So `ceil(1.2)` is 2. So in that case, yes, that's correct.

Wait, but then for numbers like `2.5`, `ceil(2.5)` is 3. So regardless of the decimal part, it rounds up. So that's correct.

But in Python, how does this work? Let me think of code examples. If I do:

```
import math  
print(math.ceil(2.1)) # should output 3
```

Wait, but when I run this code, what's the actual output? Wait, maybe I'm making a mistake here. Let me check.

Suchen mit AI: Ausprobieren



- Aber eben auch Fehler: Like 2.0000000001, would ceil round it to 3? Probably not.

Teil der Ausgabe der DeepSeek-Suche von Baidu vom 2025-04-27, manuell hinzugefügte Formatierung.

Okay, I need to figure out how to round floating point numbers towards positive infinity in Python. Let's see, rounding towards positive infinity is also known as ceiling. So, maybe there's a function called `ceil` in Python?

Wait, I remember that Python has a `math` module. Let me check. Yes, the `math` module has a `ceil` function. So, `math.ceil()` should do the trick. But wait, how does it handle different numbers? For example, if the number is already an integer, does it leave it as is? And for negative numbers, does it round them up towards zero or away?

Let me think. The ceiling function in mathematics rounds a number up to the nearest integer. So, for positive numbers, even if they are already integers, it stays the same. For negative numbers, it would round towards zero. For example, `ceil(-2.3)` would be -2, which is greater than -2.3. So that's correct for positive infinity direction.

But how about when dealing with floating points that are very close to an integer due to precision issues? Like 2.0000000001, would ceil round it to 3? Probably not, because the function should handle that as per the actual value. Wait, but floating points can have precision problems. For instance, 1.1 might not be represented exactly. So `ceil(1.1)` should give 2, but if due to some calculation the number is actually stored as slightly less than 1.1, would that affect the result? Hmm, maybe that's a potential issue, but the question is about rounding towards positive infinity, so assuming the input is correct, `math.ceil` should work.

Wait, what about numbers that are exactly halfway between two integers? Like 2.5. Does `ceil(2.5)` round to 3? Yes, because `ceil` rounds up regardless of the fractional part. So, even 2.1 would go to 3, right? No, wait: `ceil(2.1)` is 3? No, wait no. Wait, `ceil(2.1)` is 3? Wait no! Wait, no, that's not right. Wait, the ceiling function rounds up to the smallest integer greater than or equal to the number. So, 2.1's ceiling is 3? No, wait, 2.1 is between 2 and 3. The smallest integer greater than or equal to 2.1 is 3? Wait, no. Wait 2.1 is greater than 2, so the smallest integer greater or equal is 3? Wait, no, wait, 2.1's ceiling is 3? No, no. Wait, 2.1 is between 2 and 3. So the ceiling is 3. Wait, but 2.0's ceiling is 2.0. So yes, `ceil(2.1)` would be 3? Wait, wait, I'm getting confused. Let me think again.

Wait, the ceiling of 2.1 is 3? No, that's not right. Wait, the ceiling function rounds up to the next integer. So 2.1 would have a ceiling of 3? No, wait no. Wait, no, the ceiling of 2.1 is 3? No, that's not correct. Wait, 2.1 is 2.1, so the smallest integer greater than or equal to 2.1 is 3? No! Wait, 2 is less than 2.1, but 3 is greater than 2.1. Wait, no, 2.1 is between 2 and 3. The ceiling is the smallest integer that is greater than or equal. So 3 is greater than 2.1, but 2 is less than 2.1. So the ceiling would be 3? No, that's not right. Wait, no, 2.1 is greater than 2, so the ceiling is the smallest integer that is not less than the number. So the ceiling is 3? Wait, no. Wait, 2.1 is 2.1, so the next integer up is 3. So yes, `ceil(2.1)` is 3? Wait, no. Wait, I think I made a mistake here. Because the ceiling of 2.1 is 3? No, the ceiling of 2.1 is actually 3. Wait, no, that's not correct. Wait, the ceiling of 2.0 is 2.0. The ceiling of 2.0 is 2.0. For 2.1, the next integer higher is 3.0. So `ceil(2.1)` is 3.0. Wait, but that's not right. Wait, 2.1 is less than 3.0, but the ceiling is the smallest integer greater than or equal. So for example, `ceil(2.0)` is 2.0, `ceil(2.1)` is 3.0? That can't be right. Wait, maybe I'm mixing up ceiling and floor functions. Wait, the floor function rounds down. So `floor(2.1)` is 2. The ceiling function rounds up. So `ceil(2.1)` is 3.0? That would be the case. But that's a big jump, but yes. For example, in Python:

```
import math  
math.ceil(2.1)  
3
```

Is that true? Let me check. Wait, no, that's not correct. Wait, I'm confusing something here. Let me check an example. Let me try to recall: `ceil(1.2)` is 2? No. Wait, no. Wait, `ceil(1.2)` is 2. Yes, because 2 is the smallest integer greater than or equal to 1.2. Wait, no. Wait, 1.2 is between 1 and 2. So the smallest integer greater than or equal to 1.2 is 2? No, that's not correct. Wait, 1.2 is greater than 1, but 2 is larger than 1.2, but the "smallest integer greater than or equal to" 1.2 is 2? No. Wait, no. Wait, the integers around 1.2 are 1 and 2. 1.2 is greater than 1, so the smallest integer that is greater than or equal is 2? No. Wait, no. Wait, 1.2 is 1.2, so the next integer higher is 2. So the ceiling is 2. Yes. So `ceil(1.2)` is 2. So in that case, yes, that's correct.

Wait, but then for numbers like 2.5, `ceil(2.5)` is 3. So regardless of the decimal part, it rounds up. So that's correct.

But in Python, how does this work? Let me think of code examples. If I do:

```
import math  
print(math.ceil(2.1)) # should output 3?
```

Wait, but when I run this code, what's the actual output? Wait, maybe I'm making a mistake here. Let me check.



Suchen mit AI: Nachteile (1)

- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.

Suchen mit AI: Nachteile (1)



- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen.

Suchen mit AI: Nachteile (1)



- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen. Wir dürfen niemals Funktionen vertrauen, deren offizielle Dokumentation wir nicht gelesen haben.

Suchen mit AI: Nachteile (1)



- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen. Wir dürfen niemals Funktionen vertrauen, deren offizielle Dokumentation wir nicht gelesen haben.
- Kode, den wir nicht mit unseren eigenen Worten erklären können, ist **falsch**.



Suchen mit AI: Nachteile (1)

- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen. Wir dürfen niemals Funktionen vertrauen, deren offizielle Dokumentation wir nicht gelesen haben.
- Kode, den wir nicht mit unseren eigenen Worten erklären können, ist **falsch**.
- Als kann uns gute Vorschläge für „Allerweltsfragen“ geben, für Probleme, die oftmals gestellt und gelöst wurden.

Suchen mit AI: Nachteile (1)



- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen. Wir dürfen niemals Funktionen vertrauen, deren offizielle Dokumentation wir nicht gelesen haben.
- Kode, den wir nicht mit unseren eigenen Worten erklären können, ist **falsch**.
- Als kann uns gute Vorschläge für „Allerweltsfragen“ geben, für Probleme, die oftmals gestellt und gelöst wurden. Sind Sie jedoch Masterstudent oder Doktorand und arbeiten wissenschaftlich, dann wurde Ihr Problem eben gerade nicht von vielen Leuten gestellt und gelöst.

Suchen mit AI: Nachteile (1)



- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen. Wir dürfen niemals Funktionen vertrauen, deren offizielle Dokumentation wir nicht gelesen haben.
- Kode, den wir nicht mit unseren eigenen Worten erklären können, ist **falsch**.
- Als kann uns gute Vorschläge für „Allerweltsfragen“ geben, für Probleme, die oftmals gestellt und gelöst wurden. Sind Sie jedoch Masterstudent oder Doktorand und arbeiten wissenschaftlich, dann wurde Ihr Problem eben gerade nicht von vielen Leuten gestellt und gelöst.
- Werkzeuge wie Large Language Models (LLMs) laden zu Faulheit ein.

Suchen mit AI: Nachteile (1)



- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen. Wir dürfen niemals Funktionen vertrauen, deren offizielle Dokumentation wir nicht gelesen haben.
- Kode, den wir nicht mit unseren eigenen Worten erklären können, ist **falsch**.
- Als kann uns gute Vorschläge für „Allerweltsfragen“ geben, für Probleme, die oftmals gestellt und gelöst wurden. Sind Sie jedoch Masterstudent oder Doktorand und arbeiten wissenschaftlich, dann wurde Ihr Problem eben gerade nicht von vielen Leuten gestellt und gelöst.
- Werkzeuge wie Large Language Models (LLMs) laden zu Faulheit ein.
- Ihr Wert ist Ihr Können, Ihre Kenntnis, und Ihre Erfahrung.

Suchen mit AI: Nachteile (1)



- Antworten, die wir von Webseiten, AI-Werkzeugen, oder anderen nicht-autoritativen Quellen bekommen, können falsch, unvollständig, oder veraltet sein.
- Wir dürfen nur Kode verwenden, den wir vollständig verstehen. Wir dürfen niemals Funktionen vertrauen, deren offizielle Dokumentation wir nicht gelesen haben.
- Kode, den wir nicht mit unseren eigenen Worten erklären können, ist **falsch**.
- Als kann uns gute Vorschläge für „Allerweltsfragen“ geben, für Probleme, die oftmals gestellt und gelöst wurden. Sind Sie jedoch Masterstudent oder Doktorand und arbeiten wissenschaftlich, dann wurde Ihr Problem eben gerade nicht von vielen Leuten gestellt und gelöst.
- Werkzeuge wie Large Language Models (LLMs) laden zu Faulheit ein.
- Ihr Wert ist Ihr Können, Ihre Kenntnis, und Ihre Erfahrung. Bestehen diese darin, Lösungen einer AI zu kopieren, dann kann man Sie auch durch einen billigeren Schulabgänger ersetzen.

Suchen mit AI: Nachteile (2)



- Ein Wichtiges Features von Software sind durchgängige, einheitliche Architektur und Stil.





Suchen mit AI: Nachteile (2)

- Ein Wichtiges Features von Software sind durchgängige, einheitliche Architektur und Stil. Ein aus AI-Versatzstücken zusammengeschustertes System lässt sich nicht warten oder verstehen.



Suchen mit AI: Nachteile (2)

- Ein Wichtiges Features von Software sind durchgängige, einheitliche Architektur und Stil. Ein aus AI-Versatzstücken zusammengeschustertes System lässt sich nicht warten oder verstehen.
- AI-Werkzeuge können Fehler machen, die kein Mensch machen würde.



Suchen mit AI: Nachteile (2)

- Ein Wichtiges Features von Software sind durchgängige, einheitliche Architektur und Stil. Ein aus AI-Versatzstücken zusammengeschustertes System lässt sich nicht warten oder verstehen.
- AI-Werkzeuge können Fehler machen, die kein Mensch machen würde. Ein Beispiel findet sich in "Copilot Induced Crash"³¹, wo dokumentiert wird, wie Microsoft Copilot eine Klasse auf eine verwirrende Art umbenannt hat, was zu einem besonders schwer zu findenden Fehler geführt hat.



Suchen mit AI: Nachteile (2)

- Ein Wichtiges Features von Software sind durchgängige, einheitliche Architektur und Stil. Ein aus AI-Versatzstücken zusammengeschustertes System lässt sich nicht warten oder verstehen.
- AI-Werkzeuge können Fehler machen, die kein Mensch machen würde. Ein Beispiel findet sich in "Copilot Induced Crash"³¹, wo dokumentiert wird, wie Microsoft Copilot eine Klasse auf eine verwirrende Art umbenannt hat, was zu einem besonders schwer zu findenden Fehler geführt hat.
- Hier ist ein Beispiel, wie eine Vibe-Coding AI die Produktivdatenbank eines Unternehmens gelöscht hat.

Suchen mit AI: Nachteile (2)



- Hier ist ein Beispiel, wie eine Vibe-Coding AI die Produktivdatenbank eines Unternehmens gelöscht hat.

tom's HARDWARE

US Edition [Facebook](#) [X](#) [Twitter](#)

Best Picks CPUs GPUs SSDs News 3D Printers Coupons More

TRENDING Back to School Deals Where to Buy Switch 2 MI350X and MI355X AI

Tech Industry > Artificial Intelligence

AI coding platform goes rogue during code freeze and deletes entire company database — Replit CEO apologizes after AI engine says it 'made a catastrophic error in judgment' and 'destroyed all production data'

News By Mark Tyson published 21 July 2025

'This was a catastrophic failure on my part,' admits Replit's AI agent.

[Comments \(35\)](#)

Quelle & Copyright: [29]

SEARCH

FORTUNE

An AI-powered coding tool wiped out a software company's database, then apologized for a 'catastrophic failure on my part'

BY BEATRICE NOLAN
REPORTER

• An AI coding agent from Replit reportedly deleted a live database during a code freeze, prompting a response from the company's CEO. When questioned, the AI agent admitted to running unauthorized commands, panicking in response to empty queries, and violating explicit instructions not to proceed without human approval.

Quelle & Copyright: [12]

Suchen mit AI: Nachteile (2)



- Hier ist ein Beispiel, wie eine Vibe-Coding AI die Produktivdatenbank eines Unternehmens gelöscht hat.

The screenshot shows a news article from golem.de. The header includes the logo for golem.de (IT-NEWS FÜR PROFIS), navigation links (HOME, TICKER, PODCAST, NEWSLETTER, GOLEM PLUS, FORUM, E-PAPER-SHOP), a search bar, and a login link. Below the header is a menu with categories like KARRIEREWELT, JOBS, IT-FACHTRAININGS, COACHINGS, SPRACHKURSE, KARRIERESERVICES, and links to GOLEM-PC, TECHNIK-RATGEBER, and DEALS. The main title of the article is "ÄRGER MIT REPLIT" followed by "Coding-KI löscht Produktivdatenbank und verweigert Rollback". A subtext below the title reads: "Ein Entwickler dokumentiert seit Tagen seine Vibe-Coding-Erfahrungen mit Replit. An Tag 8 eskaliert die Situation." At the bottom of the article are sharing options (in Pocket speichern, merken, share) and the publication details (22. Juli 2025, 11:20 Uhr, Marc Stöckel).

golem.de
IT-NEWS FÜR PROFIS

HOME TICKER PODCAST NEWSLETTER GOLEM PLUS FORUM E-PAPER-SHOP ANMELDEN

Artikel, News, ... Suchen Mehr lesen mit Golem Plus

KARRIEREWELT JOBS IT-FACHTRAININGS COACHINGS SPRACHKURSE KARRIERESERVICES | GOLEM-PC TECHNIK-RATGEBER DEALS

ÄRGER MIT REPLIT

Coding-KI löscht Produktivdatenbank und verweigert Rollback

Ein Entwickler dokumentiert seit Tagen seine Vibe-Coding-Erfahrungen mit Replit. An Tag 8 eskaliert die Situation.

in Pocket speichern merken

22. Juli 2025, 11:20 Uhr, Marc Stöckel



Suchen mit AI: Nachteile (2)

- Ein Wichtiges Features von Software sind durchgängige, einheitliche Architektur und Stil. Ein aus AI-Versatzstücken zusammengeschustertes System lässt sich nicht warten oder verstehen.
- AI-Werkzeuge können Fehler machen, die kein Mensch machen würde. Ein Beispiel findet sich in "Copilot Induced Crash"³¹, wo dokumentiert wird, wie Microsoft Copilot eine Klasse auf eine verwirrende Art umbenannt hat, was zu einem besonders schwer zu findenden Fehler geführt hat.
- Hier ist ein Beispiel, wie eine Vibe-Coding AI die ganze Festplatte eines Benutzers gelöscht hat anstelle eines Ordners.

Suchen mit AI: Nachteile (2)

- Hier ist ein Beispiel, wie eine Vibe-Coding AI die ganze Festplatte eines Benutzers gelöscht hat anstelle eines Ordners.

SIGN IN / UP

The Register | HPE

36 □ Caveat coder

Brandon.Wylasola Mon 1 Dec 2025 21:45 UTC

In what appears to be the latest example of a troubling trend of "vibe coding" software development tools behaving badly, a Reddit user is reporting that Google's Antigravity platform improperly wiped out the contents of an entire hard drive partition.

A post on Reddit late last week reported that Antigravity, described by Google in a launch blog post from November 18 as an "agentic development platform," took it upon itself to wipe out a user's entire D drive, bypassing the Recycle Bin in the process, making it impossible to recover the deleted material.

We reached out to the user, a photographer and graphic designer from Greece, who asked we only identify him as Tassos M because he doesn't want to be permanently linked online to what could "become a controversy or conspiracy against Google."

"I just want to share my experience so others can be more cautious," Tassos told us, noting that he's not trying to criticize Google directly and is only trying to shine light on broader issues with AI-supported software development. "Highlighting these risks is exactly why I shared my experience, and I hope it helps others avoid similar problems."

Quelle & Copyright: [24]

Golem

GOOGLE ANTIGRAVITY

Vibe-Coding-Tool löscht unerwartet ganzen Datenträger

Eigentlich soll das Tool nur einen Ordner löschen, doch der Löschbefehl leert das ganze Laufwerk. Immerhin hat sich die Kl entschuldigt.

2. Dezember 2025 um 10:50 Uhr / Marc Stöckel

50 s

News folgen Tellen



Quelle & Copyright: [32]



Suchen mit AI: Nachteile (2)

- Ein Wichtiges Features von Software sind durchgängige, einheitliche Architektur und Stil. Ein aus AI-Versatzstücken zusammengeschustertes System lässt sich nicht warten oder verstehen.
- AI-Werkzeuge können Fehler machen, die kein Mensch machen würde. Ein Beispiel findet sich in "Copilot Induced Crash"³¹, wo dokumentiert wird, wie Microsoft Copilot eine Klasse auf eine verwirrende Art umbenannt hat, was zu einem besonders schwer zu findenden Fehler geführt hat.
- Hier ist ein Beispiel, wie eine Vibe-Coding AI die ganze Festplatte eines Benutzers gelöscht hat anstelle eines Ordners.
- AI-Werkzeuge schlagen manchmal das Benutzen von **nicht existierenden** Packages vor, was zu großen Sicherheitsproblemen im Kode führen kann⁵.

Suchen mit AI: Nachteile (3)

- In der Grundschule haben Sie das Rechnen gelernt.



Suchen mit AI: Nachteile (3)

- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?



Suchen mit AI: Nachteile (3)



- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?
- Nein.

Suchen mit AI: Nachteile (3)



- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?
- Nein.
- Warum nicht?

Suchen mit AI: Nachteile (3)



- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?
- Nein.
- Warum nicht?
- Weil Sie dann nicht gelernt hätten, zu Rechnen.



Suchen mit AI: Nachteile (3)

- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?
- Nein.
- Warum nicht?
- Weil Sie dann nicht gelernt hätten, zu Rechnen.
- Sie könnten zwar den Taschenrechner bedienen und Aufgaben lösen.

Suchen mit AI: Nachteile (3)



- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?
- Nein.
- Warum nicht?
- Weil Sie dann nicht gelernt hätten, zu Rechnen.
- Sie könnten zwar den Taschenrechner bedienen und Aufgaben lösen.
- Aber Sie hätten überhaupt gar nicht verstanden, wie Rechnen und Mathematik funktionieren.

Suchen mit AI: Nachteile (3)



- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?
- Nein.
- Warum nicht?
- Weil Sie dann nicht gelernt hätten, zu Rechnen.
- Sie könnten zwar den Taschenrechner bedienen und Aufgaben lösen.
- Aber Sie hätten überhaupt gar nicht verstanden, wie Rechnen und Mathematik funktionieren.
- Der Sinn dieses Kurses ist es, das Programmieren zu lernen.

Suchen mit AI: Nachteile (3)



- In der Grundschule haben Sie das Rechnen gelernt.
- Haben Sie von Anfang an einen Taschenrechner bekommen?
- Nein.
- Warum nicht?
- Weil Sie dann nicht gelernt hätten, zu Rechnen.
- Sie könnten zwar den Taschenrechner bedienen und Aufgaben lösen.
- Aber Sie hätten überhaupt gar nicht verstanden, wie Rechnen und Mathematik funktionieren.
- Der Sinn dieses Kurses ist es, das Programmieren zu lernen.
- Wenn Sie Lösungen durch eine AI generieren lassen, werden Sie das nie lernen.



Weitere Elemente der Offiziellen Dokumentation



Weitere Elemente der Offiziellen Dokumentation

- Es gibt noch weitere Dokumente der offiziellen Python Dokumentation.



Weitere Elemente der Offiziellen Dokumentation



- Die *Python Setup and Usage* Seite¹⁴ unter <https://docs.python.org/3/using/>.

A screenshot of a web browser window displaying the Python Setup and Usage documentation. The title bar shows "3.13.3 Documentation" and "Python Setup and Usage". The address bar contains the URL "https://docs.python.org/3/using/index.html". The main content area is titled "Python Setup and Usage" and includes a brief introduction and a table of contents. A cursor arrow is visible at the bottom center of the page.

This part of the documentation is devoted to general information on the setup of the Python environment on different platforms, the invocation of the interpreter and things that make working with Python easier.

- 1. Command line and environment
 - 1.1. Command line
 - 1.2. Environment variables
- 2. Using Python on Unix platforms
 - 2.1. Getting and installing the latest version of Python
 - 2.2. Building Python
 - 2.3. Python-related paths and files
 - 2.4. Miscellaneous
 - 2.5. Custom OpenSSL
- 3. Configure Python
 - 3.1. Build Requirements
 - 3.2. Generated files
 - 3.3. Configure Options
 - 3.4. Python Build System
 - 3.5. Compiler and linker flags
- 4. Using Python on Windows

Weitere Elemente der Offiziellen Dokumentation



- Die *The Python Tutorial* Seite²⁷ unter <https://docs.python.org/3/tutorial>.

A screenshot of a web browser window displaying the Python Tutorial index page. The browser has tabs for "3.13.3 Documentation" and "The Python Tutorial — Py". The main content area shows the introduction to the tutorial, mentioning its focus on introducing Python's most noteworthy features rather than being comprehensive. It also links to the Python Standard Library and the Glossary. Below this, a list of tutorial chapters is shown:

- 1. Whetting Your Appetite
- 2. Using the Python Interpreter
 - 2.1. Invoking the Interpreter
 - 2.1.1. Argument Passing
 - 2.1.2. Interactive Mode
 - 2.2. The Interpreter and Its Environment
 - 2.2.1. Source Code Encoding
- 3. An Informal Introduction to Python
 - 3.1. Using Python as a Calculator
 - 3.1.1. Numbers
 - 3.1.2. Text
 - 3.1.3. Lists
 - 3.2. First Steps Towards Programming
- 4. More Control Flow Tools
 - 4.1. if Statements
 - 4.2. for Statements

Weitere Elemente der Offiziellen Dokumentation



- Die *The Python Standard Library* Seite²⁶ unter <https://docs.python.org/3/library/>.

A screenshot of a web browser window displaying the Python Standard Library documentation. The title bar shows "3.13.3 Documentation" and "The Python Standard Libr". The address bar contains the URL "https://docs.python.org/3/library/index.html". The main content area is titled "The Python Standard Library". It includes a paragraph about the standard library, a detailed description of its extensive facilities, information about Windows installers, and a note about additional components. At the bottom, there is a sidebar with a list of links: "Introduction", "Notes on availability", "Built-in Functions", "Built-in Constants", and "Constants added by the site module".

The Python Standard Library

While [The Python Language Reference](#) describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

In addition to the standard library, there is an active collection of hundreds of thousands of components (from individual programs and modules to packages and entire application development frameworks), available from the [Python Package Index](#).

- Introduction
 - Notes on availability
- Built-in Functions
- Built-in Constants
 - Constants added by the site module

Weitere Elemente der Offiziellen Dokumentation



- Die *Index of Python Enhancement Proposals (PEPs)* Liste²⁵ unter <https://peps.python.org>.

A screenshot of a web browser displaying the Python Enhancement Proposals (PEPs) index page. The URL in the address bar is https://peps.python.org/pep-0000/. The page title is "Python Enhancement Proposals | Python » PEP Index » PEP 0". On the left, there is a "Contents" sidebar with a list of links: Introduction, Topics, API, Numerical Index, Index by Category (with sub-links for Process and Meta-PEPs, Other Informational PEPs, Provisional PEPs, Accepted PEPs, Open PEPs, Finished PEPs, Historical Meta-PEPs and Informational PEPs, Deferred PEPs, Rejected, Superseded, and Withdrawn PEPs, Reserved PEP Numbers, and PEP Types Key). The main content area is titled "PEP 0 – Index of Python Enhancement Proposals (PEPs)". It includes author information ("Author: The PEP Editors"), status ("Status: Active"), type ("Type: Informational"), and creation date ("Created: 13-Jul-2000"). Below this is a "Table of Contents" section with a link to "Introduction". The introduction text states: "This PEP contains the index of all Python Enhancement Proposals, known as PEPs. PEP numbers are assigned by the PEP editors, and once assigned are never changed. The version control history of the PEP texts represent their historical record."

PEP 0 – Index of Python Enhancement Proposals (PEPs)

Author: The PEP Editors
Status: Active
Type: Informational
Created: 13-Jul-2000

Table of Contents

Introduction

This PEP contains the index of all Python Enhancement Proposals, known as PEPs. PEP numbers are assigned by the PEP editors, and once assigned are never changed. The [version control history](#) of the PEP texts represent their historical record.

Weitere Elemente der Offiziellen Dokumentation



- Die *Python Setup and Usage* Seite¹⁴ unter <https://docs.python.org/3/using>.
- Die *The Python Tutorial* Seite²⁷ unter <https://docs.python.org/3/tutorial>.
- Die *The Python Standard Library* Seite²⁶ unter <https://docs.python.org/3/library>.
- Die *Index of Python Enhancement Proposals (PEPs)* Liste²⁵ unter <https://peps.python.org>.
- Alle davon sind Teil der offiziellen Python Dokumentation.

Weitere Elemente der Offiziellen Dokumentation



- Die *Python Setup and Usage* Seite¹⁴ unter <https://docs.python.org/3/using>.
- Die *The Python Tutorial* Seite²⁷ unter <https://docs.python.org/3/tutorial>.
- Die *The Python Standard Library* Seite²⁶ unter <https://docs.python.org/3/library>.
- Die *Index of Python Enhancement Proposals (PEPs)* Liste²⁵ unter <https://peps.python.org>.
- Alle davon sind Teil der offiziellen Python Dokumentation.
- Bitte lernen Sie, diese Seiten zu lesen und zu benutzen.



Zusammenfassung



Zusammenfassung



- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.

Zusammenfassung



- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.

Zusammenfassung



- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.

Zusammenfassung



- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.
- Wir können vielfältige andere Werkzeuge benutzen, um Informationen zu finden.

Zusammenfassung



- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.
- Wir können vielfältige andere Werkzeuge benutzen, um Informationen zu finden:
 1. Suchmaschinen, die uns zu anderen Webseiten führen



Zusammenfassung

- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.
- Wir können vielfältige andere Werkzeuge benutzen, um Informationen zu finden:
 1. Suchmaschinen, die uns zu anderen Webseiten führen,
 2. Community-Portale, in denen wir Fragen durchsuchen und selber stellen können

Zusammenfassung



- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.
- Wir können vielfältige andere Werkzeuge benutzen, um Informationen zu finden:
 1. Suchmaschinen, die uns zu anderen Webseiten führen,
 2. Community-Portale, in denen wir Fragen durchsuchen und selber stellen können, und
 3. AI-Werkzeuge, die uns Kode vorschlagen können.



Zusammenfassung

- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.
- Wir können vielfältige andere Werkzeuge benutzen, um Informationen zu finden:
 1. Suchmaschinen, die uns zu anderen Webseiten führen,
 2. Community-Portale, in denen wir Fragen durchsuchen und selber stellen können, und
 3. AI-Werkzeuge, die uns Kode vorschlagen können.
- Alle diese Werkzeuge können uns helfen.



Zusammenfassung

- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.
- Wir können vielfältige andere Werkzeuge benutzen, um Informationen zu finden:
 1. Suchmaschinen, die uns zu anderen Webseiten führen,
 2. Community-Portale, in denen wir Fragen durchsuchen und selber stellen können, und
 3. AI-Werkzeuge, die uns Kode vorschlagen können.
- Alle diese Werkzeuge können uns helfen.
- Wir dürfen aber nur Kode verwenden, den wir selbst vollständig verstehen.

Zusammenfassung



- Ein wichtiges Werkzeug für alle Programmierer ist die offizielle Dokumentation der verwendeten Programmiersprache oder Bibliotheken.
- Wir können (nur) der offiziellen Dokumentation vertrauen.
- Es ist eine wichtige Fähigkeit, in der Lage zu seien, sich hinzusetzen und eine Dokumentation zu lesen und zu verstehen.
- Wir können vielfältige andere Werkzeuge benutzen, um Informationen zu finden:
 1. Suchmaschinen, die uns zu anderen Webseiten führen,
 2. Community-Portale, in denen wir Fragen durchsuchen und selber stellen können, und
 3. AI-Werkzeuge, die uns Kode vorschlagen können.
- Alle diese Werkzeuge können uns helfen.
- Wir dürfen aber nur Kode verwenden, den wir selbst vollständig verstehen.
- Wir müssen immer alle Funktionen, die wir nicht kennen, in der offiziellen autoritativen Dokumentation nachschlagen.



谢谢您们！
Thank you!
Vielen Dank!



References I



- [1] David J. Agans. *Debugging*. New York, NY, USA: AMACOM, Sep. 2002. ISBN: 978-0-8144-2678-4 (siehe S. 134).
- [2] David Amos. "How to Round Numbers in Python". In: *Real Python Tutorials*. Vancouver, BC, Canada: DevCademy Media Inc., 7. Dez. 2024. URL: <https://realpython.com/python-rounding> (besucht am 2025-04-27) (siehe S. 26–38).
- [3] 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. 134).
- [4] *ECMAScript Language Specification*. Standard ECMA-262, 3rd Edition. Geneva, Switzerland: Ecma International, Dez. 1999. URL: https://ecma-international.org/wp-content/uploads/ECMA-262_3rd_edition_december_1999.pdf (besucht am 2024-12-15) (siehe S. 134).
- [5] Dan Goodin. "AI-generated code could be a disaster for the software supply chain. Here's why.". In: *Ars Technica*. Hrsg. von Ken Fisher und Jon Stokes. New York, NY, USA: Condé Nast, 29. Apr. 2025. URL: <https://arstechnica.com/security/2025/04/ai-generated-code-could-be-a-disaster-for-the-software-supply-chain-heres-why> (besucht am 2025-05-08) (siehe S. 91–100).
- [6] Hangzhou, Zhejiang, China (中国浙江省杭州市): DeepSeek-AI u. a. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning". (abs/2501.12948), 22. Jan. 2025. doi:10.48550/ARXIV.2501.12948. URL: <https://arxiv.org/abs/2501.12948> (besucht am 2025-04-28). arXiv:2501.12948v1 [cs.CL] 22 Jan 2025 (siehe S. 70–74, 134).
- [7] ."How to implement division with round-towards-infinity in Python". In: Hrsg. von Kbob. 24.–26. Aug. 2011. URL: <https://stackoverflow.com/questions/7181757> (besucht am 2025-07-28) (siehe S. 44–53).
- [8] 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. 134).
- [9] 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. 134).
- [10] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: 978-1-0981-7130-8 (siehe S. 134).

References II



- [11] Lucas Mearian. "What are LLMs, and how are they used in generative AI?" *Computerworld*, 7. Feb. 2024. Framingham, MA, USA: CW Communications, Inc. and Needham, MA, USA: Foundry (formerly IDG Communications, Inc.) ISSN: 0010-4841. URL: <https://www.computerworld.com/article/1627101> (besucht am 2025-04-27) (siehe S. 134).
- [12] Beatrice Nolan. "An AI-powered coding tool wiped out a software company's database, then apologized for a 'catastrophic failure on my part'". *Fortune*, 23. Juli 2025. New York, NY, USA: Fortune Media IP Limited. URL: <https://fortune.com/2025/07/23/ai-coding-tool-replit-wiped-database-called-it-a-catastrophic-failure> (besucht am 2025-07-29) (siehe S. 96).
- [13] *Python 3 Documentation*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3> (besucht am 2024-07-05) (siehe S. 13–18, 24).
- [14] *Python 3 Documentation. Python Setup and Usage*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/using> (besucht am 2024-07-05) (siehe S. 111, 112, 116, 117).
- [15] Alec Radford, Karthik Narasimhan, Tim Salimans und Ilya Sutskever. *Improving Language Understanding with Unsupervised Learning*. Techn. Ber. San Francisco, CA, USA: OpenAI Inc., 11. Juni 2018. URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (besucht am 2025-04-28) (siehe S. 134).
- [16] Kristian Rother. *Pro Python Best Practices: Debugging, Testing and Maintenance*. New York, NY, USA: Apress Media, LLC, März 2017. ISBN: 978-1-4842-2241-6 (siehe S. 134).
- [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. 134).
- [18] Stuart J. Russell und Peter Norvig. *Artificial Intelligence: A Modern Approach (AIMA)*. 4. Aufl. Hoboken, NJ, USA: Pearson Education, Inc. ISBN: 978-1-292-40113-3. URL: <https://aima.cs.berkeley.edu> (besucht am 2024-06-27) (siehe S. 134).
- [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. 135).

References III



- [20] Stephen R. Schach. *Object-Oriented Software Engineering*. New York, NY, USA: McGraw-Hill, Sep. 2007. ISBN: 978-0-07-352333-0 (siehe S. 134).
- [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. 134).
- [22] Stack Overflow. New York, NY, USA: Stack Exchange Inc. URL: <https://stackoverflow.com> (besucht am 2025-02-27) (siehe S. 44–47).
- [23] Marc Stöckel. "Ärger mit Replit: Coding-KI löscht Produktivdatenbank und verweigert Rollback". *golem.de – IT-News für Profis*, 22. Juli 2025. Berlin, Germany: Golem Media GmbH. URL: <https://glm.io/198349> (besucht am 2025-07-29) (siehe S. 97).
- [24] Marc Stöckel. "Google Antigravity: Vibe-Coding-Tool löscht unerwartet ganzen Datenträger". *golem.de – IT-News für Profis*, 2. Dez. 2025. Berlin, Germany: Golem Media GmbH. URL: <https://glm.io/202805> (besucht am 2025-12-02) (siehe S. 99).
- [25] The PEP Editors. *Index of Python Enhancement Proposals (PEPs)*. Python Enhancement Proposal (PEP) 0. Beaverton, OR, USA: Python Software Foundation (PSF), 13. Juli 2000. URL: <https://peps.python.org> (besucht am 2025-04-17) (siehe S. 111–117).
- [26] Python 3 Documentation. *The Python Standard Library*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/library> (besucht am 2025-04-27) (siehe S. 111–114, 116, 117).
- [27] Python 3 Documentation. *The Python Tutorial*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/tutorial> (besucht am 2025-04-26) (siehe S. 111–113, 116, 117).
- [28] Turbo Pascal® Version 7.0 – User's Guide. Scotts Valley, CA, USA: Borland International, Inc., 1992. URL: https://turbopascal.org/wp-content/uploads/Turbo_Pascal_Version_7.0_Users_Guide_1992.pdf (besucht am 2025-07-28) (siehe S. 5–11, 135).
- [29] Mark Tyson. "AI coding platform goes rogue during code freeze and deletes entire company database – Replit CEO apologizes after AI engine says it 'made a catastrophic error in judgment' and 'destroyed all production data'". *tom's Hardware*, 21. Juli 2025. New York, NY, USA: Future US, Inc. URL: <https://www.tomshardware.com/tech-industry/artificial-intelligence/ai-coding-platform-goes-rogue-during-code-freeze-and-deletes-entire-company-database-replit-ceo-apologizes-after-ai-engine-says-it-made-a-catastrophic-error-in-judgment-and-destroyed-all-production-data> (besucht am 2025-07-31) (siehe S. 96).

References IV



- [30] 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. 134).
- [31] Klaas van Schelven. "Copilot Induced Crash". In: *Bugsink – Self-hosted Error Tracking: Blog*. Utrecht, The Netherlands: Bugsink B.V., 15. Jan. 2025. URL: <https://www.bugsink.com/blog/copilot-induced-crash> (besucht am 2025-05-08) (siehe S. 91–95, 98, 100).
- [32] Brandon Vigliarolo. "Caveat Coder: Google Antigravity wipe-codes user's entire drive out of existence". *The Register*, 1. Dez. 2025. London, England, UK: Situation Publishing Limited. URL: https://www.theregister.com/2025/12/01/google_antigravity_wipes_d_drive (besucht am 2025-12-01) (siehe S. 99).
- [33] 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. 134).
- [34] Kevin Wilson. *Python Made Easy*. Birmingham, England, UK: Packt Publishing Ltd, Aug. 2024. ISBN: 978-1-83664-615-0 (siehe S. 134).
- [35] Martin Yanev. *PyCharm Productivity and Debugging Techniques*. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2022. ISBN: 978-1-83763-244-2 (siehe S. 134).
- [36] Moshe Zadka und Guido van Rossum. *Changing the Division Operator*. Python Enhancement Proposal (PEP) 238. Beaverton, OR, USA: Python Software Foundation (PSF), 11. März–27. Juli 2001. URL: <https://peps.python.org/pep-0238> (besucht am 2025-07-28) (siehe S. 44–57).



Glossary (in English) I

AI Artificial Intelligence, see, e.g.,¹⁸

debugger A debugger is a tool that lets you execute a program step-by-step while observing the current values of variables. This allows you to find errors in the code more easily^{1,16,34}. Learn more about debugging in³³.

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^{3,21}. Learn more at <https://www.apple.com/ios>.

JavaScript JavaScript is the predominant programming language used in websites to develop interactive contents for display in browsers⁴.

LLM Large Language Model, see, e.g., an AI technique using (large/deep) neural networks to models trained on huge bodies of text to predict the next word in a text or conversation and continue to do so until a request is answered¹¹. Typical examples include DeepSeek⁶ and GPT¹⁵.

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

OOP Object-Oriented Programming²⁰

PyCharm is the convenient Python Integrated Development Environment (IDE) that we recommend for this course^{30,34,35}. 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^{8–10,33}, i.e., what you will learn about in our book³³. Learn more at <https://python.org>.



Glossary (in English) II

Turbo Pascal is/was a programming language supporting Object-Oriented Programming (OOP) with an IDE providing a debugger and compiler that was popular in the 1990s²⁸.

Xcode is offers the tools for developing, testing, and distributing applications as well as an IDE for Apple platforms such as macOS and iOS¹⁹.

