

Find the current directory and file's directory [duplicate]

Ask Question
Asked 14 years, 1 month ago
Modified 2 years, 1 month ago
Viewed 5.3m times

3102

This question already has answers here:
What's the working directory when using IDLE? (6 answers)
How do I get the path and name of the python file that is currently executing? (26 answers)
Closed 7 years ago.
How do I determine:


1. the current directory (where I was in the shell when I ran the Python script), and
2. where the Python file I am executing is?

[python:directory](#)

Share
Improve this question
Follow
edited Nov 21, 2022 at 14:25


Andy Lester
93.9k • 16 16 gold 10 10 silver 16 16 bronze badges

asked Feb 28, 2011 at 1:51


John Howard
64.3k • 23 23 gold 30 30 silver 16 16 bronze badges

31

import os
cwd = os.getcwd() # to pwd within python
– Charlie Parker
Dec 14, 2012 at 19:14

This question is **blatantly** two questions in one and should have been closed as needing more focus. Both questions are simple reference questions, and thus ought to each have separate canonicals that this can be dupe-hammered with. However, I have been absolutely tearing my hair out trying to find a **proper** canonical for **only** the first question. I am turning up **countless** duplicates for the second question, most of which involve OP **not realizing there is a difference**.
– Karl Knechtel
Mar 14, 2013 at 8:03

I have added the best I could find for "Q. How do I determine the current directory? A. Use os.getcwd()" after **literally hours** of searching. Ugh.
– Karl Knechtel
Mar 14, 2013 at 8:40

if you are just trying to get the current folder name without full path then you can try this: os.path.basename(<path>)
– sarawgeck
Feb 19, 2014 at 2:50

[Add a comment](#)

13 Answers

Sorted by:
Highest score (default)

▲

4794

▼

🔄

To get the full path to the directory a Python file is contained in, write this in that file:

```
import os  
dir_path = os.path.dirname(os.path.realpath(__file__))
```

(Note that the incantation above won't work if you've already used os.chdir() to change your current working directory, since the value of the `__file__` constant is relative to the current working directory and is not changed by an os.chdir() call.)

To get the current working directory use

```
import os  
cwd = os.getcwd()
```

Documentation references for the modules, constants and functions used above:

- The [os](#) and [os.path](#) modules.
- The `__file__` constant
- [os.path.realpath\(path\)](#) (returns "the canonical path of the specified filename, eliminating any symbolic links encountered in the path")
- [os.path.dirname\(path\)](#) (returns "the directory name of pathname path")
- [os.getcwd\(\)](#) (returns "a string representing the current working directory")
- [os.chdir\(path\)](#) ("change the current working directory to path")

Share
Improve this answer
Follow
edited Jul 31, 2016 at 12:34


Mark Amery
156k • 90 90 gold 140 140 silver 47 47 bronze badges

answered Feb 28, 2011 at 1:54


Russell Dias
73.5k • 5 5 gold 136 136 silver 32 32 bronze badges

206

I hate it when I use this to append to sys.path. I feel so dirty right now.

– FlipMcF
Sep 16, 2015 at 21:52

18

file will not work if invoked from an IDE (say IDLE). Suggest os.path.realpath("/") or os.getcwd(). Best anser in here: stackoverflow.com/questions/2632199/.
– Neon22
Dec 20, 2013 at 11:12

5

@Neon22 might suit some needs, but I feel it should be noted that those things aren't the same at all - files can be outside the working directory.

– Mark

Sep 15, 2018 at 17:31

3

@Moberg Often the paths will be the same when reversing `realpath` with `dirname`, but it will differ when the file (or its directory) is actually a symbolic link.

– Lekensteyn

Mar 12, 2015 at 17:00

9

It gets an error `NameError: name '__file__' is not defined`. How to solve this?

– Mohammad ElNesr

Sep 26, 2016 at 13:15

Show 10 more comments



392



Current working directory: `os.getcwd()`

And the `__file__` attribute can help you find out where the file you are executing is located. This Stack Overflow post explains everything: [How do I get the path of the current executed file in Python?](#)

Share

Improve this answer

Follow

edited Sep 6, 2021 at 20:05



Peter Mortensen

31.6k • 22 140 140 137 badges

answered Feb 28, 2011 at 1:53



Nix

58.7k • 31 31 150 150 207 badges

Add a comment



370



You may find this useful as a reference:

```
import os

print("Path at terminal when executing this file")
print(os.getcwd() + "\n")

print("This file path, relative to os.getcwd()")
print(__file__ + "\n")

print("This file full path (following symlinks)")
full_path = os.path.realpath(__file__)
print(full_path + "\n")

print("This file directory and name")
path, filename = os.path.split(full_path)
print(path + " -> " + filename + "\n")

print("This file directory only")
print(os.path.dirname(full_path))
```

Share

Improve this answer

Follow

edited Nov 13, 2015 at 14:16



CommunityBot

1 • 1 1 silver badge

answered Dec 5, 2012 at 10:18



Daniel Reis

13.4k • 6 6 40 40 134 badges

13

what does `__file__` signifies here? It does not work for me.

– Alex Raj Kaliamoorthy

Mar 17, 2016 at 15:22

15

The `__file__` is an attribute of the module object. You need run the code inside a Python file, not on the REPL.

– Daniel Reis

Mar 20, 2016 at 13:07

Add a comment



339



The `pathlib` module, introduced in Python 3.4 ([PEP 428](#) & “The `pathlib` module & “object-oriented filesystem paths”), makes the path-related experience much much better.

pwd

/home/skovorodkin/stack

tree

```
â””â””â””â””â”” scripts
  â””â””â””â””â”” 1.py
  â””â””â””â””â”” 2.py
```

In order to get the current working directory, use `Path.cwd()` :

```
from pathlib import Path
```

```
print(Path.cwd()) # /home/skovorodkin/stack
```

To get an absolute path to your script file, use the `Path.resolve()` method

```
print(Path( file ).resolve()) # /home/skovorodkin/stack/scripts/1.py
```

And to get the path of a directory where your script is located, access `.parent` (it is recommended to call `.resolve()` before `.parent`):

```
print(Path( file ).resolve().parent) # /home/skovorodkin/stack/scripts
```

Remember that `__file__` is not reliable in some situations: [How do I get the path of the current executed file in Python?](#).

Please note, that `Path.cwd()` , `Path.resolve()` and other `Path` methods return path objects (`PosixPath` in my case), not strings. In Python 3.4 and 3.5 that caused some pain, because `open` built-in function could only work with strings or bytes objects, and did not support `Path` objects, so you had to convert `Path` objects to strings or use the `Path.open()` method, but the latter option required you to change old code:

File `scripts/2.py`

```
from pathlib import Path
p = Path(__file__).resolve()

with p.open() as f: pass
with open(str(p)) as f: pass
with open(p) as f: pass

print('OK')
```

Output

```
python3.5 scripts/2.py
```

```
Traceback (most recent call last):
  File "scripts/2.py", line 11, in <module>
    with open(p) as f:
TypeError: invalid file: PosixPath('/home/skovorodkin/stack/scripts/2.py')
```

As you can see, `open(p)` does not work with Python 3.5.

[PEP 519](#) *"Adding a file system path protocol"*, implemented in Python 3.6, adds support of [PathLike](#) objects to the `open` function, so now you can pass `Path` objects to the `open` function directly:

```
python3.6 scripts/2.py
```

OK

[Share](#)

[Improve this answer](#)

[Follow](#)

edited Sep 6, 2021 at 20:32



Peter Mortensen

31.6k • 22 22 gold 22 gold badges 140 140 silver 140 silver badges 433 433 bronze badges

answered Sep 5, 2017 at 19:14



skovorodkin

10.3k • 1 1 gold 140 140 silver 140 silver badges 339 339 bronze badges

8

Note also that these methods are chainable, so you can use `app_path = Path(__file__).resolve().parent.parent.parent` as a parallel to `../././` if you need to.

– shacker

Commented Apr 19 at 7:10

What system has executables (or the equivalent) by the name "python3.5" and "python3.6"? Ubuntu [Ubuntu MATE 20.04](#) (Focal Fossa) doesn't (at least not by default). It has executables by the name "python3" and "python2" (but not "python" - which causes [some things to break](#))

– Peter Mortensen

Commented Apr 20 at 20:42

@PeterMortensen, thanks for the corrections. I don't remember if I actually had `python3.x` symlinks that time. Maybe I thought it would make snippets a bit clearer to the reader.

– skovorodkin

Commented Apr 20 at 16:06

Add a comment



85



1. To get the current directory full path


```
>>import os
>>print os.getcwd()

Output: "C:\Users\admin\myfolder"
```
2. To get the current directory folder name alone


```
>>import os
>>str1=os.getcwd()
>>str2=str1.split("\")
>>n=len(str2)
>>print str2[n-1]

Output: "myfolder"
```

[Share](#)

[Improve this answer](#)

[Follow](#)

edited Sep 6, 2021 at 20:09



Peter Mortensen

31.6k • 22 22 gold 22 gold badges 140 140 silver 140 silver badges 433 433 bronze badges

answered Apr 24, 2012 at 7:00



vinitravat

883 • 6 6 gold 6 gold badges 33 33 silver 33 silver badges

14

better do it in one line, i think: `os.getcwd().split("\")[-1]`

– imkost

Commented Apr 22 at 16:24

61

better to use `os.sep` rather than hardcode for Windows: `os.getcwd().split(os.sep)[-1]`

– kkurian

Commented Apr 2 at 8:24

7

the problem with this approach is that if you execute the script from a different directory, you will get that directory's name instead of the script's, which may not be what you want.

– airstrike

Commented Apr 23 at 16:28

2

Right, the current directory which hosts your file may not be your CWD

– foster

Commented Apr 16 at 3:56

Add a comment



64



[Pathlib](#) can be used this way to get the directory containing the current script:

```
import pathlib
filepath = pathlib.Path( file ).resolve().parent
```

[Share](#)

[Improve this answer](#)

[Follow](#)

edited Sep 6, 2021 at 20:15



Peter Mortensen

31.6k • 22 22 gold 22 gold badges 140 140 silver 140 silver badges 433 433 bronze badges

answered Sep 29, 2016 at 13:07



Mathieu C.

869 ● 6 ● 6 silver badges

I like this solution. However can cause some Python 2.X issues.

– Kimmo Hintikka

edited Jul 7, 2017 at 9:32

2

For python 3.3 and earlier pathlib has to be installed

– A. Romeu

answered Oct 11 at 6:43

8

@Kimmo The only reason you should be working in Python 2 code is to convert it to Python 3.

– kagornick

edited Oct 8 at 18:42

@kagornick agreed, but there are still people who don't. I write all my new stuff with formatted string literals (PEP 498) using Python 3.6 so that someone doesn't go and push them to Python2.

– Kimmo Hintikka

edited Sep 8 at 13:21

Note also that these methods are chainable, so you can use `app_path = Path(__file__).resolve().parent.parent.parent` as a parallel to `../../../../` if you need to.

– shacker

edited Oct 9 at 7:11

Show 1 more comment



44



If you are trying to find the current directory of the file you are currently in:

OS agnostic way:

```
dirname, filename = os.path.split(os.path.abspath(__file__))
```

Share

Improve this answer

Follow

edited Jul 7, 2013 at 21:10



Ashwini Chaudhary

251k ● 60 ● 60 gold ● 476 silver badges

answered Oct 7, 2012 at 9:10



Kurt

473 ● 4 ● 4 silver badges

Add a comment



43



To get the current directory full path:

```
os.path.realpath('.')
```

Share

Improve this answer

Follow

edited May 29, 2020 at 20:12



Logovskii Dmitrii

3,013 ● 4 ● 4 gold ● 11 silver badges

answered Sep 22, 2015 at 8:12



Ilia S.

887 ● 8 ● 8 silver ● 23 bronze badges

11

This one works from inside a jupyter iPython notebook ('__file__' and getcwd won't)

– Oliver Zendt

edited Apr 16 at 10:26

5

I'm working remotely with a Jupyter Notebook: `os.getcwd()` and `os.path.realpath('.')` return exactly the same string path.

– Leevo

edited Oct 9 at 8:56

@Leevo: Point being?

– Peter Mortensen

edited Apr 20 at 20:13

This returns the jupyter root directory, not the directory holding the file.

– Scott

edited Sep 23 at 23:20

Add a comment



40



If you're using Python 3.4, there is the brand new higher-level `pathlib` module which allows you to conveniently call `pathlib.Path.cwd()` to get a `Path` object representing your current working directory, along with many other new features.

More info on this new API can be found [here](#).

Share

Improve this answer

Follow

edited Feb 23, 2017 at 15:12

answered Feb 20, 2015 at 20:32



[Jazzer](#)
3,053 • 1 • 1 gold badge 24 badges

4

For Python version < 3.4 you can use `pathlib2` : pypi.python.org/pypi/pathlib2
– Eyal Levin
Contributed at 14:54

Add a comment



38



Answer to #1:

If you want the current directory, do this:

```
import os
os.getcwd()
```

If you want just any folder name and you have the path to that folder, do this:

```
def get_folder_name(folder):
    """Returns the folder name, given a full folder path
    return folder.split(os.sep)[-1]
```

Answer to #2:

```
import os
print os.path.abspath( file )
```

[Share](#)
[Improve this answer](#)
[Follow](#)

edited Dec 15, 2016 at 0:08

answered Nov 5, 2015 at 9:31



[Blair23](#)

12.1k • 7 • 7 gold badges 25 badges

Add a comment



32



I think the most succinct way to find just the name of your current execution context would be:

```
current_folder_path, current_folder_name = os.path.split(os.getcwd())
```

[Share](#)
[Improve this answer](#)
[Follow](#)

edited Sep 6, 2021 at 20:11



[Peter Mortensen](#)

31.6k • 22 • 22 gold badges 133 badges

answered Oct 9, 2013 at 10:31



[StormShadow](#)

1,627 • 4 • 4 gold badges 13 badges

Add a comment



22



For question 1, use `os.getcwd()` # Get working directory and `os.chdir(r'D:\Steam\steamapps\common')` # Set working directory

I recommend using `sys.argv[0]` for question 2 because `sys.argv` is immutable and therefore always returns the current file (module object path) and not affected by `os.chdir()`. Also you can do like this:

```
import os
this_py_file = os.path.realpath(__file__)
```

vvv Below comes your code vvv

But that snippet and `sys.argv[0]` will not work or will work weird when compiled by PyInstaller, because magic properties are not set in `__main__` level and `sys.argv[0]` is the way your executable was called (it means that it becomes affected by the working directory).

[Share](#)
[Improve this answer](#)
[Follow](#)

edited Sep 6, 2021 at 20:18



[Peter Mortensen](#)

31.6k • 22 • 22 gold badges 133 badges

answered Jun 15, 2017 at 13:45



[Kotauskas](#)

1,402 • 16 • 16 gold badges 34 badges

Add a comment



18



If you're searching for the location of the currently executed script, you can use `sys.argv[0]` to get the full path.

[Share](#)
[Improve this answer](#)
[Follow](#)

edited Oct 18, 2015 at 3:03



[Eric Leschinski](#)

154k • 96 • 96 gold badges 337 badges

answered Aug 12, 2013 at 11:27



[Grochni](#)

1,861 • 21 • 21 gold badges 38 badges

12

This is wrong. `sys.argv[0]` needn't contain the *full* path to the executing script.

– Mark Amery
Contributed at 10:23

Add a comment



Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.
Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

[pythondirectory](#)

See similar questions with these tags.

- [The Overflow Blog](#)
- [Generating components, not tokens](#)
- [Visually orchestrating data diagnostics but platform agnostic](#)
- [Featured on Meta](#)
- [Changes to reporting for the \[status-review\] escalation process](#)
- [Results of the March 2025 Community Asks Sprint](#)
- [A discussion about closed \(and potentially useful\) posts on Stack Overflow](#)
- [Policy: Generative AI \(e.g., ChatGPT\) is banned](#)

Linked

How do I get the path and name of the python file that is currently executing?

What's the working directory when using IDLE?

How to indicate the current directory of the script not me?

0

Python3 can't read or at least find file

0

print the folder name of a python script

Is there a way to view the directory that a file is in using python?

0

Python : Howto locate current Path?

0

Hi. I'd like to ask a question about directory in python

How do I get the full path of the current file's directory?

How do I change the working directory in Python?

See more linked questions

Related

What does the "yield" keyword do in Python?

How do I get the directory where a Bash script is located from within the script itself?

What does if `__name__ == "__main__":` do?

How do I create a directory, and any missing parent directories?

3464

How do I list all files of a directory?

How do I get the current time in Python?

Find all files containing a specific text (string) on Linux?

How do I add an empty directory to a Git repository?

"Least Astonishment" and the Mutable Default Argument

Hot Network Questions



What is the word or phrase that expresses a person is unhappy and stops talking because someone upsets them?



What is the actus reus of an automated crime and when does it occur?



Capacitor replacement inside hot gun



fit 17 blocks into a 5x5x5 box



Did any stateless societies have writing?

more hot questions

STACK OVERFLOW

Questions
Help
Chat
PRODUCTS
Teams
Advertising
Talent
COMPANY
About
Press
Work Here
Legal
Privacy Policy
Terms of Service
Contact Us
Cookie Settings
Cookie Policy
STACK EXCHANGE NETWORK
Technology
Culture & recreation
Life & arts
Science
Professional
Business

[API](#)
[Data](#)
[Blog](#)
[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)

Site design / logo © 2025 Stack Exchange Inc; user contributions licensed under [CC BY-SA](#) . rev 2025.4.21.25386