

Who put Python in the Windows 10 May 2019 Update?



Steve

May 21st, 2019

Today the Windows team [announced the May 2019 Update for Windows 10](#). In this post we're going to look at what we, Microsoft's Python team, have done to make Python easier to install on Windows by helping the community publish to the Microsoft Store and, in collaboration with Windows, adding a default "python.exe" command to help find it. You may have already heard about these on the [Python Bytes](#) podcast, at PyCon US, or [through Twitter](#).

As software moves from the PC to the cloud, the browser, and the Internet of Things, development workflows are changing. While Visual Studio remains a great starting point for any workload on Windows, many developers now prefer to acquire tools individually and on-demand.

For other operating systems, the platform-endorsed package manager is the traditional place to find individual tools that have been customized, reviewed, and tested for your system. On Windows we are exploring ways to provide a similar experience for developers without impacting non-developer users or infringing publishers' ability to manage their own releases. The [Windows Subsystem for Linux](#) is one approach, offering developers consistency between their build and deployment environments. But there are other developer tools that also matter.

One such tool is Python. Microsoft has been involved with the Python community for over twelve years, and currently employ four of the key contributors to the language and primary runtime. The



growth of Python has been incredible, as it finds homes among data scientists, web developers, system administrators, and students, and roughly half of this work is [already happening on Windows](#). And yet, Python developers on Windows find themselves facing more friction than on other platforms.

Installing Python on Windows

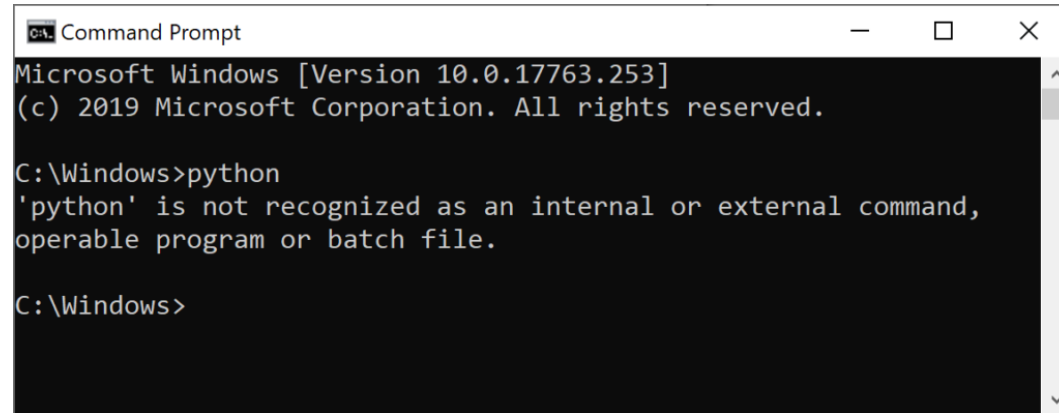
It's been widely known for many years that Windows is the only mainstream operating system that does not include a Python interpreter out of the box. For many

users who are never going to need it, this helps reduce the size and improve the security of the operating system. But for those of us who do need it, Python's absence has been keenly felt.

Once you discover that you need to get Python, you are quickly faced with many choices. Will you download an installer from python.org? Or perhaps a distribution such as Anaconda? The Visual Studio installer is also an option. And which version? How will you access it after it's been installed? You quickly find more answers than you need, and depending on your situation, any of them might be correct.

We spent time figuring out why someone would hit the error above and what help they need. If you're already a Python expert with complex needs, you probably know how to install and use it. It's much more likely that someone will hit this problem the first time they are trying to use Python. Many of the teachers we spoke to confirmed this hypothesis – students encounter this far more often than experienced developers.

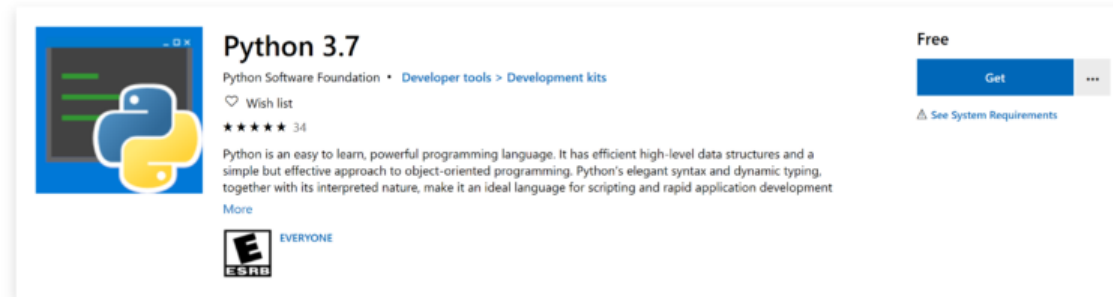
So we made things easier.

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the following text: "Microsoft Windows [Version 10.0.17763.253] (c) 2019 Microsoft Corporation. All rights reserved. C:\Windows>python 'python' is not recognized as an internal or external command, operable program or batch file. C:\Windows>". The text is displayed in a monospaced font on a black background with white text. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Command Prompt
Microsoft Windows [Version 10.0.17763.253]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows>python
'python' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows>
```



First, we helped the community release [their distribution of Python](#) to the Microsoft Store. This version of Python is fully maintained by the community, installs easily on Windows 10, and automatically makes common commands such as `python`, `pip` and `idle` available (as well as equivalents with version numbers `python3` and `python3.7`, for all the commands, just like on Linux).

```
Command Prompt
Microsoft Windows [Version 10.0.18362.113]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows>python3.7
Python 3.7.3 (tags/v3.7.3:ef4ec6ed12, Mar 25 2019, 22:05:12) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Windows>pip3 list
Package Version
-----
certifi 2019.3.9
chardet 3.0.4
idna 2.8
requests 2.21.0
urllib3 1.24.2

C:\Windows>_
```

Finally, with the May 2019 Windows Update, we are completing the picture. While Python continues to remain completely independent from the operating system, every install of Windows will include `python` and `python3` commands that take you directly to the [Python store page](#). We believe that the Microsoft Store package is perfect for users starting out with Python, and given our experience with and participation in the Python community we are pleased to endorse it as the default choice

with and participation in the Python community we are pleased to endorse it as the default choice.

We hope everyone will be as excited as [Scott Hanselman was when he discovered it](#). Over time, we plan to extend similar integration to other developer tools and reduce the getting started friction. We'd love to hear your thoughts, and suggestions, so feel free to post comments here or use the Windows Feedback app.



Scott Hanselman
@shanselman

Follow



WHOA. I'm on a new copy of Windows (Insiders? newish) and I typed Python - on a machine where I don't have it - and it launched the Windows Store into an official distribution I can install in a click. WHEN did this happen. I love this.



[Steve Dower](#)

Software Engineer, Python

Follow



Posted in [Python](#)

Tagged [cpython](#) [python](#) [windows](#)

Read next

Python in Visual Studio Code – May 2019 Release

We are pleased to announce that the May 2019 release of the Python Extension for Visual Studio Code is now available. In this release we

Python in Visual Studio Code – June 2019 Release

We are pleased to announce that the June 2019 release of the Python Extension for Visual Studio Code is now available. In this release we

closed a total of 42 issues including IntelliSense in the Python Interactive Window and additional improvements to the Python Language Server. Check out the full list of improvements in our changelog.

 **Luciana Abud** May 28, 2019

 [5 comments](#)

closed a total of 70 issues including a plot viewer with the Python Interactive window, parallel tests with pytest, and indentation of run selection in the terminal.

 **Luciana Abud** June 25, 2019

 [3 comments](#)

70 comments

Comments are closed.

[Newer Comments →](#)



Andrew Lucius May 21, 2019 1:54 pm



Microsoft R Open + RStudio Desktop would be great additions to the store.

 [Log in](#)



Niko Strijbol May 21, 2019 4:31 pm



I know this is probably too much work for one release of Windows, so it's a nice interim solution, but wouldn't it make more sense to create a decent official package manager for Windows instead of having to include shims for who knows how many programs?

 [Log in](#)



Steve Dower  May 22, 2019 3:20 pm



Yep, we're definitely not going to get into the "let's have hundreds of shims" game. But it was worth it for this one to see whether the workflow actually works.

And the challenge with an official package manager is that someone needs to produce the official packages 😊 The advantages are that we know that users aren't going to be stuck with broken products or installs if they all go through the store, but of course it takes more effort for the package publishers. Getting one example out there should help us see which approach to take in the future.

↩ [Log in](#)



Andrew Marlow

May 18, 2020 5:41 am



I welcome this relatively new awareness and acceptance of python by Microsoft. IMO it's a step in the right direction but sadly falls short of simply providing it by default. I had this argument with a Microsoft representative at QCON Lon in 2019. My argument was that there needs to be a decent portable scripting language. By default Windows doesn't have one. He gave the reply "yes, but which language should we pick (there are so many to choose from)". Well, I think this move shows that the answer to that question is "python".

It's interesting to note that making the python command available by default but by default "python" does not invoke the python interpreter seems to hurt people that require their systems to have python the most, judging by the comments below. I think these issues would just go away if the actual interpreter was installed by default, set up so it can be upgraded as and when via the Microsoft Store.

Another reason to have python installed as default is for use in corporations. Many corporations disable access to the Windows Store so that makes python off-limits.

↩ [Log in](#)



Steve Dower

June 2, 2020 2:51 pm



You make some good points here, sorry I didn't see your post when you made it.

Unfortunately, a pre-installed Python would have caused just as many issues as the current stub command. Overriding existing installs was unintentional (and we discovered/fixed a number of bugs here after release!) but a real install of a different version would have likely been more disruptive than the fast-failure case that was shipped.

The default install is very complicated, both legally (who becomes liable for the impact of bugs?), ethically/security (enables hackers to run complex scripts on any Windows machine... yes, like PowerShell... which is why we knew we didn't want to do that again 😊), and technically (when do we upgrade? Like, what time of day is safe? And can we go from 3.7->3.8 silently?). Not to mention that 90+% of Windows users do not want or need Python, and would rather store a few extra photos on their device.

I've actually been strongly against a default install for the entire time I've been responsible for Python's Windows releases, so people actually had to convince *me* that this approach wasn't a terrible idea. And despite the (very real and totally not users' fault) problems that have come up, the vast majority of installs appear to work fine, which is always the way when you add a somewhat niche feature to a very popular operating system.

As for the corporations that disable access, I agree that makes things more complicated. But it's also kind of the point – presumably they're blocking other installs too. The [Microsoft Store for Business](#) is, as I understand, the way for corporations to control the set of available apps for their users. It should be easy to allow installing Python through that (though I'm not an admin on any, so have never seen the interface).

↩ [Log in](#)



Darren Woodford

June 2, 2020 2:18 pm



Rather than putting in shims (that can confuse existing developers and power users) could Microsoft back "chocolatey" instead?

↩ [Log in](#)



Steve Dower

June 2, 2020 2:36 pm



As [winget](#) becomes more integrated, we'll be looking for ways to migrate the current stub file to use that instead. Which was always the plan, but winget hadn't been announced at the time we started this approach, so we couldn't say anything 😊

Unfortunately, using the standard installer isn't a satisfactory approach for us, due to reliability and "undo"-ability, so we want to keep pulling the package from the Microsoft Store. That rules out most existing options, unfortunately, but winget will have it before their official release.

[↩ Log in](#)



Darren Woodford June 3, 2020 12:05 am



Ah cool, I wasn't aware of winget. It's a shame there couldn't have been a tie-up with an existing package manager but I can see you have at least looked into it. It will be good to have an 'official' package manager at last anyway.



Charles Milette June 8, 2019 3:29 pm



I believe that the idea is that the Store is the package manager.

[↩ Log in](#)



Jackson, Dan May 22, 2019 2:32 am



What about LTSC versions of Windows that don't have the Microsoft Store? Should we just continue to use python.org or Visual Studio installers, or will there be a way to sideload this official version onto 1809 LTSC (for example)?

[↩ Log in](#)



Steve Dower  May 22, 2019 3:24 pm



If the python.org installers are fine, I'd say keep using those (they are also the ones that the Visual Studio installer includes, so it doesn't matter which of these you use).

But if you have a use case for getting the MSIX file directly so you can sideload it, the best place to share it would be by filing a bug at <https://bugs.python.org/> asking for them to be published as a separate download. They could be made available on python.org as their own download, but the need would have to outweigh the confusion when people download them by mistake and have to figure out what to do with it.

[↩ Log in](#)



Ian Hanschen May 5, 2020 10:19 am



You're going to need to remove this "helpful" thing.

[↩ Log in](#)



Shawn O. Eary May 23, 2019 5:12 am



I know I'm a bit odd, but I would personally rather see the Glasgow Haskell Compiler and Interpreter installed into MS Windows by default than something named after an ugly snake. I've used Python before and honestly I don't care for scripting languages that lend themselves to tons of runtime errors. I tend to prefer more strict languages like F# and now Haskell for "automation" tasks. While it's disappointing that F# seems to have "failed to thrive", I would be interested if Microsoft reattempted to create a *pure* Haskell.NET that wasn't polluted by Microsoft's attempt to appeal to the masses.

DISCLAIMER: My comment does not represent the opinions of any of my current or former employers.

[↩ Log in](#)



Rui Rei May 28, 2019 5:44 am



You are odd indeed sir. I would say more than a "bit" though... 😊

[↩ Log in](#)



Radosław Rusiniak September 11, 2019 12:23 am



1. Haskell is cool, I agree.
2. AFAIK Python was named after Monty Python's Flying Circus, not snake 😊

↩ [Log in](#)



Vitali Supruniuk May 30, 2019 10:43 am



So how do we fix existing installations now? When I run python via command line it always opens Microsoft Store, this new feature affected hundreds of our vms.

↩ [Log in](#)



Steve Dower  May 30, 2019 12:27 pm



For that many machines, you probably want to put your own Python install higher in PATH (at least higher than the WindowsApps folder, possibly even into the per-machine section if that makes more sense). The Python installer normally puts it high enough that this isn't an issue.

If the machines are being used interactively, users can use the "Manage App Execution Aliases" page to disable the alias. Then your regular python.exe will be found even when it is later on PATH.

↩ [Log in](#)



Albin Gustavsson June 1, 2019 11:47 pm



Yeah, I'm not really sure how you didn't realize that adding applications that is not installed to the PATH is an absolutely terrible idea. Having a store with a proper package management backend is Great! Just don't implement obviously bad things like installing shortcuts before the app is installed potentially breaking existing installs.

↩ [Log in](#)



Jeff Bowes June 13, 2019 5:57 am



Broke my current install too. Very annoying

[↩ Log in](#)



Steve Dower  June 13, 2019 7:16 am

✓ [🔗](#)

Was your current install from the Store? Or using the installer from python.org and with manually modified PATH?



Chris Goldsmith November 15, 2019 1:12 am

✓ [🔗](#)

Not a great Idea guys – breaking peoples development environments by putting something in the path that hijacks something that is already installed – via the store or from python.org should have caused a little head scratching.

:thumbs down: for this approach

[↩ Log in](#)



Lennart Borgman June 6, 2019 10:22 am

✓ [🔗](#)

Could you please tell the version number of Windows 10 where this way to setup python is included? (I can't see the "python.exe" in my path. And I have just checked Windows Update again. I have the latest update of Windows 10, the latest I can get right now, at least.)

[↩ Log in](#)



Steve Dower  June 7, 2019 2:08 pm

✓ [🔗](#)

It is part of version 1903, which started rolling out the same day this blog was posted. Apparently you should be able to get it by checking for updates, but it's possible that you have some piece of hardware that isn't compatible yet and so it's being blocked. You'll have to contact support for help with that, I'm afraid. It's more than we can do via blog comments

[↩ Log in](#)



Mayeul Cantan June 7, 2019 1:45 pm



That's an interesting and welcome development, I am curious to see it unfold.
But please, do tell me that "Python" and "pip" launch Python3 and pip3 by default, respectively. Python2 is EOL at the end of the year, so if you really need to include it, that's the one to postfix with a "2" 😊
Archlinux does it right, in my opinion 😊

[↩ Log in](#)



Steve Dower  June 7, 2019 2:05 pm



What's Python 2? 🐼

Yes, we're only recommending Python 3 releases from the core Python team, not any of the legacy ones. People who know they need old versions know how to get them, while this is targeted at people who aren't quite sure where to begin and most of the online docs aren't helping them.

[↩ Log in](#)



Chris June 8, 2019 12:04 pm



I wonder about how visual studio copes with installing and developing for python and it's interoperabilities with code in java, powershell, etc...

[↩ Log in](#)



Steve Dower  June 10, 2019 7:32 am



When you install the Python workload, Visual Studio 2019 handles Python very well.

Interoperating with other languages is up to the languages though, so you'd need to get in touch with those communities to ask about that. Visual Studio is really just an editor, not a runtime.

[↩ Log in](#)



Jan Gazda June 10, 2019 12:43 am



Hi, this is exciting news! But I suspect something is missing (or am I missing something?) *"every install of Windows will include `python` and `python3` commands that take you directly to the [Python store page](#)."* I just did fresh installation of win10 with May update and executables are not in PATH or missing entirely 😞

[↩ Log in](#)



Steve Dower  June 10, 2019 7:33 am



The link itself comes through a default store package called App Installer, so that may still need to be updated. Depending on how quickly you tried it, you may have been faster.

If you open the Microsoft Store app and get all updates, that ought to set it up.

[↩ Log in](#)



Jamal eddine Naamani November 30, 2019 3:53 am



Hi by default and if `!python` is in PATH the command will be `py` to verify if python is installed and how to update to python 3.8 i suggest this tuto "[How to install python 3.8](#)"

[↩ Log in](#)



Morten242 . June 15, 2019 3:41 am



While I appreciate it a lot I'm seeing some problems actually using this thing from programs like cmake. Having the shim in path isn't enough for it to be actually picked up and used, I need to manually specify the path to the actual installation. Just picking the shim ends up with an error about the system not being allowed to access the file.

[↩ Log in](#)



Steve Dower 

June 17, 2019 7:19 am



This is likely something to do with how cmake is launching Python, rather than Python itself.

Do you have an issue on their repository you could link for me so I can help them look into it?

[↩ Log in](#)



Long Li (MSRA) 

June 20, 2019 10:12 pm



I met the exact same problem with cmake and filed an issue, I think someone already @ you in one issue.

<https://gitlab.kitware.com/cmake/cmake/issues/19324>

<https://github.com/dotnet/coreclr/pull/24722>

[↩ Log in](#)



Steve Dower 

August 21, 2019 8:47 am



(Sorry for the delay – didn't realise this was held up by our spam filter because of the two links.)

I can't post to the cmake thread, but my suggestion would be to check "python -V" for a successful exit code (0) and if that fails, treat Python as if it's not available on PATH. Users who then interactively run "python" will get taken to the Store.

Those who have manually modified their own PATH to put a Python install *at the end* will need to update their configurations to move it up in priority (or use Manage App Execution Aliases to disable the global commands). Directly deleting the files is not going to be reliable, as Windows Update will just put them back.

[↩ Log in](#)



Morten242 .

July 7, 2019 1:02 pm



Sorry, I don't know of an issue in their repo. But it could be a problem in their end, indeed.

[← Log in](#)

[Newer Comments →](#)

Relevant Links

Python Resources at Microsoft
Python in Visual Studio
Python in VS Code
Azure Notebooks
Python on Azure
Microsoft on PyPI

Top Bloggers



Luciana Abud
Program Manager



Jeffrey Mew



Dan Taylor
Principal Program Manager



Steve Dower
Software Engineer



Claudia Regio
Program Manager

Archive

July 2020
June 2020
May 2020
April 2020
March 2020
February 2020
January 2020
November 2019
October 2019
September 2019
August 2019

Stay informed



What's new

Surface Go 2

Surface Book 3

Microsoft 365

Surface Pro X

Windows 10 apps

Microsoft Store

Account profile

Download Center

Microsoft Store support

Returns

Order tracking

Store locations

Virtual workshops and training

Microsoft Store Promise

Education

Microsoft in education

Office for students

Office 365 for schools

Deals for students & parents

Microsoft Azure in education

Enterprise

Azure

AppSource

Automotive

Government

Healthcare

Manufacturing

Financial services

Retail

Developer

Microsoft Visual Studio

Windows Dev Center

Developer Center

Microsoft developer program

Channel 9

Office Dev Center

Microsoft Garage

Company

Careers

About Microsoft

Company news

Privacy at Microsoft

Investors

Diversity and inclusion

Accessibility

Security



English (United States)

[Sitemap](#)

[Contact Microsoft](#)

[Privacy & cookies](#)

[Terms of use](#)

[Trademarks](#)

[Safety & eco](#)

[About our ads](#)

© Microsoft 2020