
Python in a Windows Universe; Survival Guide Documentation

Release 0.1

Code For Tallahassee

Mar 29, 2017

Contents

1	Quick Start	3
1.1	Step 1: Download & Install Python 2 & 3	3
1.2	Step 2: Validate the Default Interpreter	3
1.3	Step 3: Install a Virtual Environment Manager	4
1.4	Step 4: Install Interactive Computing Environments	4
1.5	Step 5: Install a Version Control System	5
1.6	Step 6: Install Unix Utilities	5
1.7	Step 7: Install a Regular Expression Debugger	5
1.8	Step 8: Install a Tool for Getting & Sending Files using URL Syntax	5
1.9	Step 9: Install a Code & Markup Editor	6
1.10	Congratulations!	7
2	Python Installation	9
2.1	Python version 2 or 3?	9
2.2	Python 32 or 64-bit version?	9
2.3	Download Python	10
2.4	Install Python	10
2.5	Congratulations!	13
3	Python Packages	15
3.1	PyPI	15
3.2	pip	15
3.3	Wheels	16
4	Virtual Environments	17
4.1	Virtual Environment Commands	17
4.2	Virtualenvwrapper-win Environment Variables	18
4.3	Summary	18

Up-to-date, easy-to-read guide for building Python developer-friendly ecosystems in Windows 10.

This guide is a community project ...

Brought to you to courtesy of [Code For Tallahassee](#), a non-profit [Code For America Bridgade](#) located in [Florida's](#) beautiful [capital city](#).

The guide is a work in progress ...

The plan is to expand to include other Python in Windows related topics over time. We welcome new contributions, corrections and other feedback from the Python community at large; be respectful, but please don't be shy.

This project is licensed under Creative Commons ...

[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#)

Step 1: Download & Install Python 2 & 3

Python 2.7 (32-bit)

1. Click on the following link to download Python 2.7 ... <https://www.python.org/ftp/python/2.7.12/python-2.7.12.msi>
2. Double-click on the downloaded file to install it and use all of the default settings.

Python 3.5 (32-bit)

Attention: Mark the “**Install Python 3.5 PATH**” checkbox at the bottom of the first dialog box in the Python 3.5 install wizard.

1. Click on the following link to download Python 3.5 ... <https://www.python.org/ftp/python/3.5.2/python-3.5.2.exe>
2. Double-click on the downloaded file to install it. With the exception of the “Install Python 2.5 PATH” checkbox, use all of the default settings.

Step 2: Validate the Default Interpreter

Use **Ctrl-z** to exit

Press **[Ctrl]-z** where you see the **^z** symbol to exit out of the Python Interpreter.

1. Open the command prompt and open the default interpreter:

```
C:\> python
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit_
↳(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^z
```

2. Open the Python 2.7 interpreter:

```
C:\> c:\Python27\python.exe
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit_
↳(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^z
```

Step 3: Install a Virtual Environment Manager

1. Install virtualenvwrapper-win ...

```
C:\> cd \temp
C:\> git clone https://github.com/davidmarble/virtualenvwrapper-win.git
C:\> cd virtualenvwrapper-win
C:\> python setup.py install
C:\> copy scripts\mkproject.bat %USERPROFILE%\AppData\Local\Programs\Python\Python35-
↳32\Scripts
C:\> notepad %USERPROFILE%\AppData\Local\Programs\Python\Python35-32\Scripts\workon.
↳bat
```

2. Add the “doskey workoff=deactivate” line after “@echo off”.

```
@echo off

doskey workoff=deactivate

if not defined WORKON_HOME (
    set "WORKON_HOME=%USERPROFILE%\Envs"
)
```

Step 4: Install Interactive Computing Environments

Install IPython & Jupyter Notebook ...

```
C:\> pip install jupyter

C:\> mkvirtualenv -p c:\Python27\python.exe cp27
C:\> setprojdir c:\%USERNAME%\projects\cp27
C:\> workon cp27
C:\> pip install jupyter

C:\> mkproject cp35
```


Step 5: Install a Version Control System

Note: Download will start automatically once you click on the link

1. Click on the following link to download Git ... <https://git-scm.com/download/win>
2. Double-click on the downloaded file to install it and use all of the default settings.

Step 6: Install Unix Utilities

1. Click on the following link to download UnxUtils ... <https://sourceforge.net/projects/unxutils/files/latest/download>
2. Click on the downloaded file to open it.
3. Right-click anywhere in the whitespace on right pane of file explorer and select **“Extract All”**.
4. When setting the destination folder browse to **“C:Program Files (x86)”**.
5. Press [Win]-s then type **“env”**
6. Select **“Edit the System Environment Variables: Control Panel”**
7. Click on the **“Environment Variables”** button
8. Click on the **“Path”** variable under **“System Variables”**
9. Click on the **“Edit”** button
10. Click on the **“New”** button
11. Type **“C:Program Files (x86)UnxUtilsusrlocalwbin”** and press [Enter]
12. Click the **“Move Up”** button until it is above **“%SystemRoot%system32”**
13. Verify the **sort** command is the one that comes with *unxutils* by typing the following ...

```
C:\>whereis sort
C:\Program Files (x86)\UnxUtils\usr\local\wbin\sort.exe
```

Step 7: Install a Regular Expression Debugger

1. Click on the following link to download Kodos ... <https://sourceforge.net/projects/kodos/files/latest/download?source=files>
2. Double-click on the downloaded file to install it and use all of the default settings.

Step 8: Install a Tool for Getting & Sending Files using URL Syntax

1. Click on the following link to download cURL ... <http://www.confusedbycode.com/curl/#downloads>
2. Check the **“I am not a robot”** checkbox.
3. Click on the **curl-7.46.0-win32.exe** button on the **“Administrator Privileges (free)”** row.

4. Double-click on the downloaded file to install it and use all of the default settings.

Step 9: Install a Code & Markup Editor

1. Click on the following link to download Python 2.7 ... <https://download.sublimetext.com/Sublime%20Text%20Build%203114%20Setup.exe>
2. Double-click on the downloaded file to install it.
3. Set the destination directory to ...

```
c:\Program Files (x86)\Sublime Text 3
```

3. Press and release the [Win] key and right-click on the on the “**Sublime Text 3**” icon and select “**More**” then “**Pin to Taskbar**”. Click on the new icon on your taskbar (brown square with an orange “S” in the middle).
4. From the menu at the top select “View” → “Show Console”.
5. Click in the console at the bottom of the screen and paste in the following code and press [Enter]

```
import urllib.request,os,hashlib; h = '2915d1851351e5ee549c20394736b442' +  
→ '8bc59f460fa1548d1514676163daafc88'; pf = 'Package Control.sublime-package'; ipp =  
→ sublime.installed_packages_path(); urllib.request.install_opener( urllib.request.  
→ build_opener( urllib.request.ProxyHandler()) ); by = urllib.request.urlopen( 'http://  
→ packagecontrol.io/' + pf.replace(' ', '%20')).read(); dh = hashlib.sha256(by).  
→ hexdigest(); print('Error validating download (got %s instead of %s), please try_  
→ manual install' % (dh, h)) if dh != h else open(os.path.join( ipp, pf), 'wb' ).  
→ write(by)
```

6. From the menu select “**Preferences**” → “**Package Control**”
7. Press “**i**” [Enter] to “**Install Package**”
8. Type “**flake8**” and select “**Python Flake8 Lint**”
9. From the menu select “**Preferences**” → “**Package Control**”
10. Press “**i**” [Enter] to “**Install Package**”
11. Type “**rest**” and select “**RestructuredText Improved**”
12. From the menu select “**Preferences**” → “**Package Control**”
13. Press “**i**” [Enter] to “**Install Package**”
14. Type “**sidebar**” and select “**SideBarEnhancements**”
15. From the menu select “**File**” → “**New File**”
16. Press [Ctrl]-s to “**Save As**”
17. Type “**temp.py**”
18. From the menu select “**Preferences**” → “**Settings - More**” → “**Syntax Specific - User**”
19. Copy & Paste the text below into the main editor window

```
{  
    "rulers": [72, 79],  
    "translate_tabs_to_spaces": true,  
    "draw_white_space": "all",
```

```

"tab_size": 4
}

```

17. Press **[Ctrl]-f** then **s** to save the changes.
18. Press **[Ctrl]-w** to close the window.
19. If you have administrator right to your machine ... Right-click **[Windows Start]** and select **“Command Prompt - Admin”** Click the **“Yes”** button on the **“Do you want to allow this application to make changes to your PC?”** Copy and paste the following into the command prompt ...

Attention: If you installed Sublime Text 3 somewhere other than “C:Program Files (x86)Sublime Text 3” then modify the SET statement below accordingly.

```

@echo off
SET st3Path=C:\Program Files (x86)\Sublime Text 3\sublime_text.exe

rem add it for all file types
@reg add "HKEY_CLASSES_ROOT\*\shell\Open with Sublime Text 3" /t REG_
↪SZ /v "" /d "Edit with Sublime Text 3" /f
@reg add "HKEY_CLASSES_ROOT\*\shell\Open with Sublime Text 3" /t REG_
↪EXPAND_SZ /v "Icon" /d "%st3Path%,0" /f
@reg add "HKEY_CLASSES_ROOT\*\shell\Open with Sublime Text 3\command" /t REG_
↪SZ /v "" /d "%st3Path% \"%1\"" /f

rem add it for folders
@reg add "HKEY_CLASSES_ROOT\Folder\shell\Open with Sublime Text 3" /t REG_
↪REG_SZ /v "" /d "Edit with Sublime Text 3" /f
@reg add "HKEY_CLASSES_ROOT\Folder\shell\Open with Sublime Text 3" /t REG_
↪REG_EXPAND_SZ /v "Icon" /d "%st3Path%,0" /f
@reg add "HKEY_CLASSES_ROOT\Folder\shell\Open with Sublime Text 3\command" /t REG_
↪REG_SZ /v "" /d "%st3Path% \"%1\"" /f

```

20. Close the Command window.

Congratulations!

You're Windows 10 Python ecosystem is ready to rumble.

Python version 2 or 3?

Tip: Recommendation: **Both**

- Develop your coding habits in Python 3; it's not going away.
- Keep 2.7 around; you can't kill it, it's immortal.¹
- If your code's going public consider supporting both.

See also:

Should I use Python 2 or Python 3 for my development activity?

Python 32 or 64-bit version?

Tip: Recommendation: **32-bit**, unless ...

- You're willing to trade away memory for performance (64-bit can consume up twice the memory).
- You work with really large data sets and need access to more than 4GB of system memory.

¹ Despite claims that Python 2's scheduled End Of Life date (EOL, sunset date) is in 2020.

Download Python

32-Bit Python

- [Download Python 2.7.12 \(32-bit\)](#)
- [Download Python 3.5.2 \(32-bit\)](#)

64-Bit Python

- [Download Python 2.7.12 \(64-bit\)](#)
- [Download Python 3.5.2 \(64-bit\)](#)

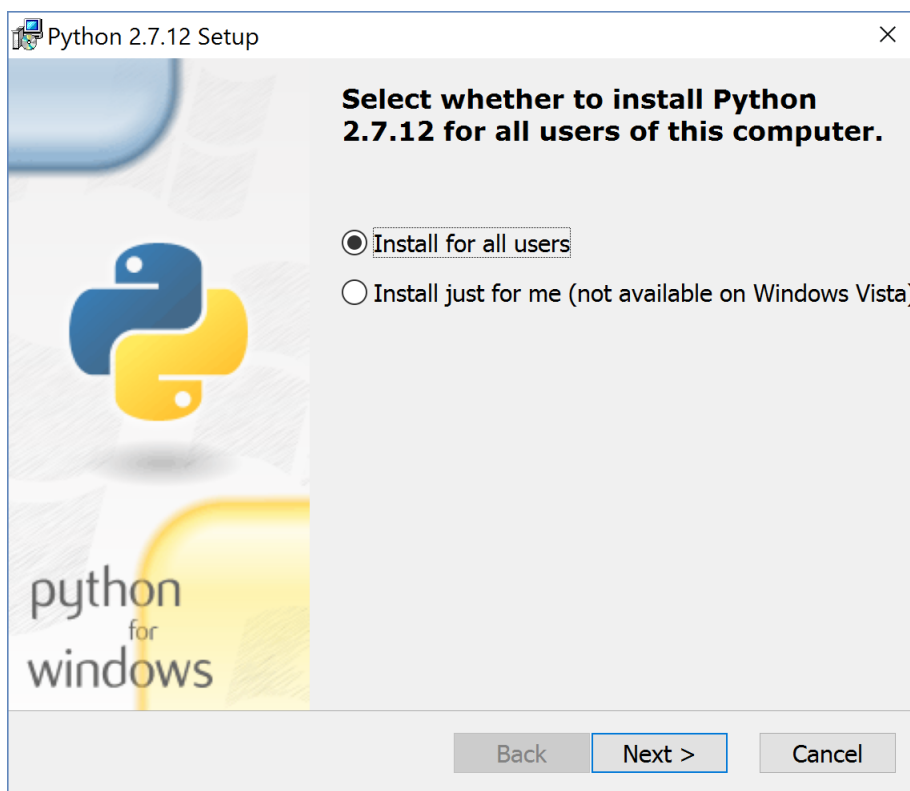
See also:

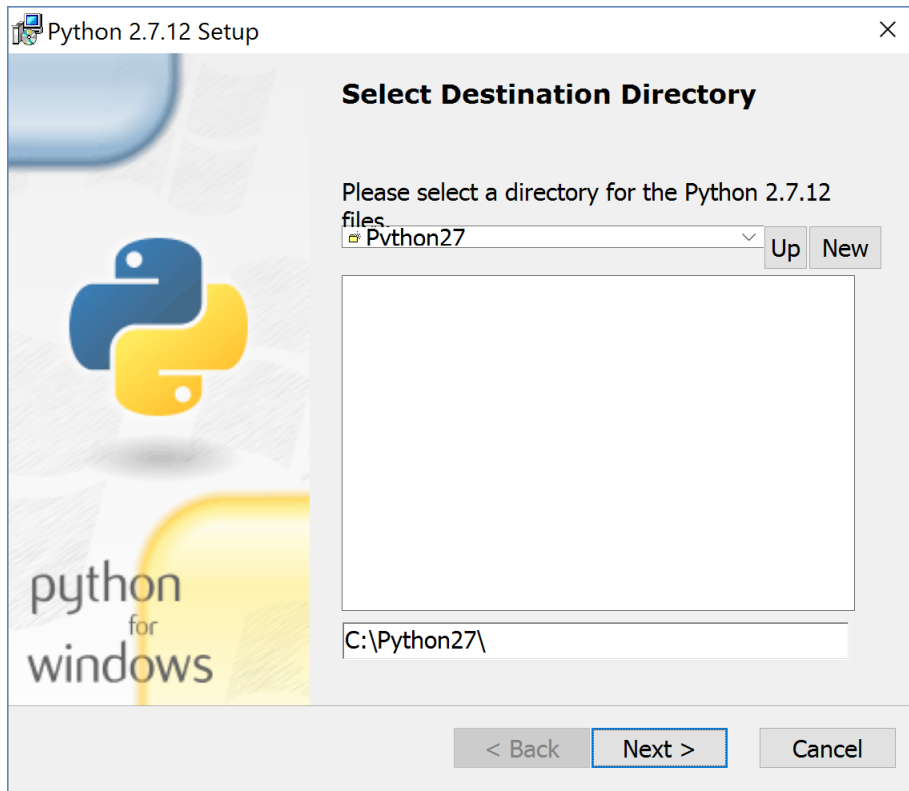
[Python downloads page](#)

Install Python

Install Python 2.7

Double-click on the Python 2.7 MSI that you downloaded to launch the installer, and click the “**Next**” button at each screen to accept the default settings.





Note: At this point you may see a Windows pop-up dialog box, “Do you want to allow this application install software

on your PC?” If the “Verified Publisher” is set to “Python Software Foundation” then click the “Yes” button.

See also:

User Account Control

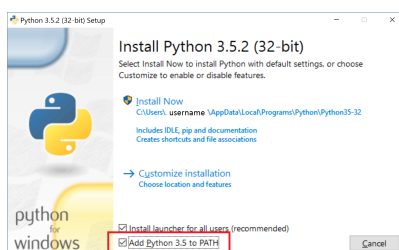


Click “**Finish**” to complete the Python 2.7 installation.

Install Python 3.5

Double-click on the Python 3.5 executable (.exe) that you downloaded to launch the installer.

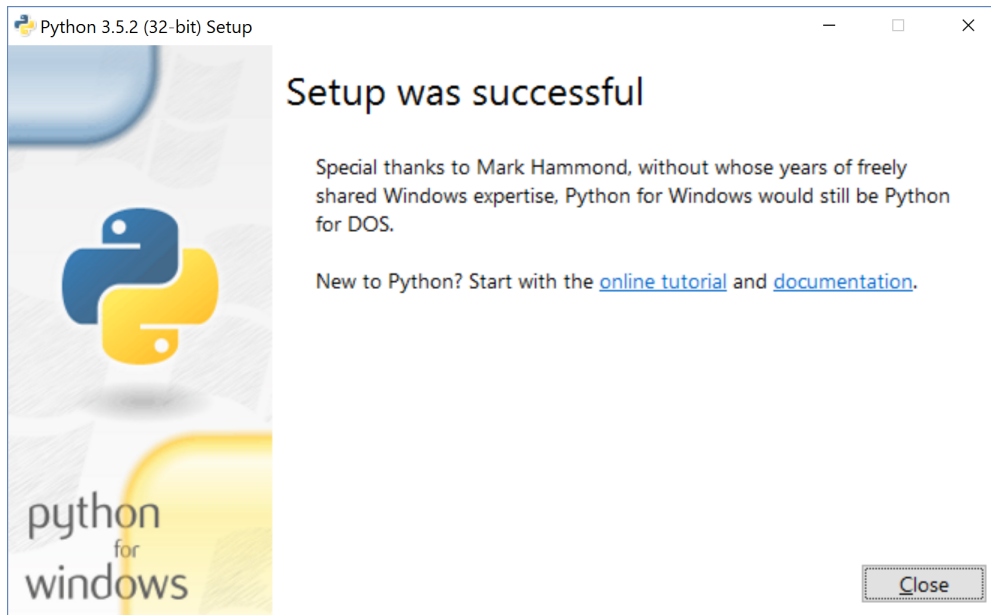
Note: On the first screen make sure to check the “**Add Python 3.5 to PATH**” checkbox.



Click on the “**Install Now**” section to begin.

Note: At this point you may see a Windows pop-up dialog box, “Do you want to make change to your PC?” If the

“Verified Publisher” is set to “Python Software Foundation” then click the “Yes” button.



Click “**Finish**”

Congratulations!

You have now installed both Python 2 & 3.

CHAPTER 3

Python Packages

“Python is known for its **‘batteries included’** philosophy and has a rich standard library. However, being a popular language, the number of third party packages is much larger than the number of standard library packages. So it eventually becomes necessary to discover how packages are used, found ...”

—Hitchiker’s Guide to Packaging

PyPI

The [Python Package Index \(PyPI\)](#), formerly known as the [Cheeseshop](#), is a central repository containing tens of thousands of ready to install Python community projects and distributions. It will become an integral part of your Python ecosystem.

pip

[Pip](#) works with PyPI, it’s used to install and manage Python packages. Pip replaces [easy_install](#).

Action	Command
install package	<code>pip install <pkg-name></code>
uninstall package	<code>pip uninstall <pkg-name></code>
upgrade package	<code>pip install -U <pkg-name></code>

Pip was installed as part of Python, but now we need to upgrade it to make sure we are using the latest version:

```
pip install --upgrade pip
```

Command	Action
<code>pip list</code>	List all packages installed in the current environment.
<code>pip search <text></code>	List all PyPI packages whose name or summary contains <text>.
<code>pip show <pkg-name> ...</code>	Show information about one or more installed packages.

Go head and practice:

```
pip list
pip search web2py
pip show pip
```

Wheels

A **Wheel** is python distribution standard that doesn't require building or compilation; wheels replace **eggs**. Wheel has an official [PEP](#); Egg doesn't.

Use Wheel if you try to install a package from PyPI and you see this ...

Warning: error: Unable to find vcvarsall.bat

The package is probably a C extension and it can't find a compiler in Windows. The easiest way to install these types of packages is to download the Wheel file from the [Unofficial Windows Binaries for Python Extension Packages](#).

For example, locate the **pandas**, a data analysis package. The Wheel files look something like this:

```
pandas-0.18.1-cp27-cp27m-win32.whl
```

The Wheel file format is defined in [PEP 425](#) as:

```
{distribution}-{version}(-{build tag})?-{python tag}-{abi tag}-{platform tag}.whl
```

“When people speak of Python they often mean not just the language but also the CPython implementation. Python is actually a specification for a language that can be implemented in many different ways.”

—Official Python Guide

File Name	Package-Version	Python-Version	32/64-bit?
pandas-0.18.1-cp27-cp27m-win32.whl	0.18.1	Python 2.7	32-bit
pandas-0.18.1-cp27-cp27m-win_amd64.whl	0.18.1	Python 2.7	64-bit
pandas-0.18.1-cp35-cp35m-win32.whl	0.18.1	Python 3.5	32-bit
pandas-0.18.1-cp35-cp35m-win_amd64.whl	0.18.1	Python 3.5	64-bit

Once you download the correct **Wheel** file for your Python version & architecture you can install it with **pip**:

```
pip install <wheel-filename>
```

Make sure to include the path when you specify the Wheel file, for example:

```
pip install c:\users\<username>\downloads\pandas-0.18.1-cp27-cp27m-win32.whl
```

Virtual Environments

“A Virtual Environment is a tool to keep the dependencies required by different projects in separate places, by creating virtual Python environments for them. It solves the ‘Project X depends on version 1.x but, Project Y needs 4.x’ dilemma, and keeps your global site-packages directory clean and manageable.”

—Hitchiker’s Guide to Python, Virtual Environments

Virtual environments not only help manage package versions, but also ensure the correct version of the Python interpreter is associated with a project.

- `virtualenv` is a tool to create these isolated Python environments.
- `virtualenvwrapper` is a set of extensions to Ian Bicking’s `virtualenv` tool to add the following features:
 1. Organize all of your virtual environments in one place.
 2. Manage virtual environments (create, delete and list).
 3. Allows the use a single command to switch between environments.
- `virtualenvwrapper-win` is a port of Doug Hellmann’s `virtualenvwrapper` to Windows

Let’s install `virtualenvwrapper-win`:

```
pip install virtualenvwrapper-win
```

Virtual Environment Commands

Command	Description
<code>mkvirtualenv <project-name></code>	Create a new virtual environment
<code>lsvirtualenv</code>	List all existing virtual environments
<code>setprojectdir <path></code>	Associate a project folder with a virtual environment
<code>cdproject</code>	Change to the project folder
<code>workon <project-name></code>	Start up a virtual environment
<code>deactivate</code>	Shown down the virtual environment

Virtualenvwrapper-win Environment Variables

Default Environment Variables (preset by virtualenvwrapper-win):

```
WORKON_HOME=%USERPROFILE%\Envs  
PROJECT_HOME=%USERPROFILE%\projects
```

Variable	Description
WORKON_HOME	The directory where the virtual environments will be stored.
PROJECT_HOME	The project directory associated with a virtualenv.

We recommend creating a generic sandbox environment for both Python 2.7 and the latest Python 3 release. The gives you a environment to play, experiment and learn in, but we suggest creating a separate virtual environment for each Python project that you work with.

Here's how to setup the Python 3.5 sandbox:

```
C:\> mkvirtualenv cp35  
C:\> setprojectdir %PROJECT_HOME%\cp35  
C:\> workon cp35
```

1. This creates new virtual environment we chose to call **cp35** (stands for CPython ver. 3.5)
2. This sets the project directory for the virtual environment.
3. This restarts the virtual environment and moves you into the project directory.

Now let's setup the Python 2.7 sandbox (remember you can use tab completion to avoid typing in the whole name:

```
C:\> mkvirtualenv -p c:\python\python27\python.exe  
C:\> setprojectdir %PROJECT_HOME%\cp27  
C:\> workon cp27
```

Note: Since our default Python Interpreter is 3.5, we need to use the -p option with mkvirtualenv to indicate which interpreter we want to use for the virtual environment.

Summary

When you want to play in CPython 2.7:

```
C:\> workon cp25
```

When you want to switch to CPython 3.5:

```
C:\> workon cp35
```

When you want to exit the virtual environment:

```
C:\> deactivate
```

When you want to create a virtual environment for a new project:

```
C:\> mkvirtualenv <name>  
C:\> setprojectdir <path>  
C:\> workon <name>
```

Attention: The [Windows logo key](#) will appear as `[win]` in the docs.

Credits ...

- Gas mask image in left pane courtesy of [Pixabay](#).
- Document structure loosely patterned after [Hitchhiker's Guide to Python!](#)