PYTHON

1. How is Python? (History)

* Python was conceived in the late 1980s[[31]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-venners-interview-pt-1-31) by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) at [Centrum Wiskunde & Informatica](https://en.wikipedia.org/wiki/Centrum_Wiskunde_%26_Informatica) (CWI) in the [Netherlands](https://en.wikipedia.org/wiki/Netherlands) as a successor to the [ABC language](https://en.wikipedia.org/wiki/ABC_(programming_language)) (itself inspired by [SETL](https://en.wikipedia.org/wiki/SETL))[[32]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-12-32), capable of [exception handling](https://en.wikipedia.org/wiki/Exception_handling) and interfacing with the [Amoeba](https://en.wikipedia.org/wiki/Amoeba_(operating_system)) operating system.[[7]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-faq-created-7) Its implementation began in December 1989.[[33]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-timeline-of-python-33) Van Rossum's long influence on Python is reflected in the title given to him by the Python community: [*Benevolent Dictator For Life*](https://en.wikipedia.org/wiki/Benevolent_Dictator_For_Life) (BDFL) – a post from which he gave himself permanent vacation on July 12, 2018.[[34]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-lj-bdfl-resignation-34)
* Python 2.0 was released on “16 October 2000” with many major new features,

including a [cycle-detecting](https://en.wikipedia.org/wiki/Cycle_detection) [garbage collector](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) and supporfor [Unicode](https://en.wikipedia.org/wiki/Unicode).[[35]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-newin-2.0-35)

* Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility).[[36]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-3.0-release-36) Many of its major features were [backported](https://en.wikipedia.org/wiki/Backporting" \o "Backporting) to Python 2.6.x[[37]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-pep-3000-37) and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.[[38]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-38)

1. What is Python?

* Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.
* Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables,

1. Why is python?

* Python is a general-purpose language, which means it can be used to build just about anything, which will be made easy with the right tools/libraries.
* Professionally, Python is great for backend web development, data analysis, artificial intelligence, and scientific computing. Many developers have also used Python to build productivity tools, games, and desktop apps, so there are plenty of resources to help you learn how to do those as well

4) Where is Python?

* With a few exceptions, Python is used pretty much wherever a programmer who knows Python wants to focus on solving a problem instead of struggling with implementation details. You'll find it [in](http://www.civ4.com/) [games](http://www.eveonline.com/), [web](http://www.djangoproject.com/) [applications](http://code.google.com/appengine/), [network servers](http://www.list.org/), [scientific computing](http://wiki.python.org/moin/NumericAndScientific), [media tools](http://www.pitivi.org/), [application scripting](http://www.autodesk.com/maya), etc. (There's a somewhat old list of some organizations that use it [here](http://wiki.python.org/moin/OrganizationsUsingPython).) People who know it well tend to [love it](http://xkcd.com/409/) because it strikes a very rare balance of conciseness and clarity, and (perhaps to a lesser extent) because it has a rich set of useful libraries.

5)Python Installation.(WINDOWS)

### Step 1: Download the Python 3 Installer

1. Open a browser window and navigate to the [Download page for Windows](https://www.python.org/downloads/windows/) at [python.org](https://www.python.org/).
2. Underneath the heading at the top that says **Python Releases for Windows**, click on the link for the **Latest Python 3 Release - Python 3.x.x**. (As of this writing, the latest is Python 3.6.5.)
3. Scroll to the bottom and select either **Windows x86-64 executable installer** for 64-bit or **Windows x86 executable installer** for 32-bit.

### Step 2: Run the Installer

Once you have chosen and downloaded an installer, simply run it by double-clicking on the downloaded file.

**Important:** You want to be sure to check the box that says **Add Python 3.x to PATH** as shown to ensure that the interpreter will be placed in your execution path.

1. Virtual Environment Setup

## Installing pip

[pip](https://packaging.python.org/key_projects/#pip) is the reference Python package manager. It’s used to install and update packages. You’ll need to make sure you have the latest version of pip installed.

### Windows

The Python installers for Windows include pip. You should be able to access pip using:

py -m pip --version

pip 9.0.1 from c:**\p**ython36**\l**ib**\s**ite-packages (Python 3.x.x)

x belongs to the version you downloaded.

You can make sure that pip is up-to-date by running:

py -m pip install --upgrade pip

## Installing virtualenv

[virtualenv](https://packaging.python.org/key_projects/#virtualenv) is used to manage Python packages for different projects. Using virtualenv allows you to avoid installing Python packages globally which could break system tools or other projects. You can install virtualenv using pip.

On Windows:

py -m pip install --user virtualenv

## Creating a virtualenv

[virtualenv](https://packaging.python.org/key_projects/#virtualenv) allows you to manage separate package installations for different projects. It essentially allows you to create a “virtual” isolated Python installation and install packages into that virtual installation. When you switch projects, you can simply create a new virtual environment and not have to worry about breaking the packages installed in the other environments. It is always recommended to use a virtualenv while developing Python applications.

To create a virtual environment, go to your project’s directory and run virtualenv.

On Windows:

py -m virtualenv env

virtualenv will create a virtual Python installation in the env folder.

## Activating a virtualenv

Before you can start installing or using packages in your virtualenv you’ll need to activate it. Activating a virtualenv will put the virtualenv-specific python and pip executables into your shell’s PATH.

On Windows:

.\env\Scripts\activate

# IDE:

What is IDE:

# integrated development environment (IDE)

An integrated development environment (IDE) is a software suite that consolidates basic tools required to write and test software.

An IDE typically contains a code editor, a compiler or interpreter, and a debugger, accessed through a single [graphical user interface](https://searchwindevelopment.techtarget.com/definition/GUI) (GUI). The user writes and edits [source code](https://searchmicroservices.techtarget.com/definition/source-code) in the code editor. The compiler translates the source code into a readable language that is executable for a computer. And the debugger tests the software to solve any issues or bugs.

An IDE can also contain features such as programmable editors, object and data modeling, [unit testing](https://searchsoftwarequality.techtarget.com/definition/unit-testing), a source code library and build automation tools.

* There are many ide’s supported for Python ; likely pyCharm, Eclipse, Aptana studio etc , of which we are using “pyCharm” For our projects.

# Data types and Data Structures