



SCA

SHARMA COMPUTER ACADEMY



PYTHON

LECTURE 2



Today's Agenda



Version History , Downloading and Installing Python

- Version History
- Python 2 v/s Python 3
- Different Python Implementations
- Downloading And Installing Python
- Testing Python Installation



Python Version History



- **First released on Feb-20th -1991 (ver 0.9.0)**
- Python 1.0 launched in Jan-1994
- Python 2.0 launched in Oct-2000
- **Python 3.0 launched in Dec-2008**
- **Python 2.7 launched in July 2010**
- Python 3.6.5 launched on March-28th-2018
- Python 3.7 launched on June-27th -2018
- **Current latest is Python 3.7.3 launched on Mar-25th -2019**



The Two Versions Of Python



- As you can observe from the previous slide , there are 2 major versions of Python , called **Python 2** and **Python 3**
- **Python 3** came in **2008** and **it is not backward compatible with Python 2**
- This means that a project which uses **Python 2** will not run on **Python 3**.
- This means that we have to **rewrite the entire project** to migrate it from **Python 2** to **Python 3**



Some Important Differences



- **In Python 2**
`print "Hello Bhopal"`
- **In Python 3**
`print("Hello Bhopal")`
- **In Python 2**
`5/2 → 2`
`5/2.0 → 2.5`
- **In Python 3**
`5/2 → 2.5`
- The way of accepting input has also changed and like this there are many changes



The Two Versions Of Python



- So to prevent this overhead of programmers , **PSF** decided to support **Python 2** also.
- But this support will only be till **Jan-1-2020**
- You can visit <https://pythonclock.org/> to see exactly how much time is left before Python 2 retires



Which Version Should I Use ?



- For beginners , it is a point of confusion as to **which Python version they should learn ?**



- The obvious answer is **Python 3**



Why Python 3 ?



- We should go with **Python 3** as it brings lot of new features and new tricks compared to **Python 2**
- **Moreover as per PSF, *Python 2.x is legacy, Python 3.x is the present and future of the language***
- **All major future upgrades will be to Python 3 and , Python 2.7 will never move ahead to even Python 2.8**



Various Implementations Of Python



- The **Python language** has many popular **implementations**
- The word **implementation** means **the tools/software** which are used for the execution of programs written in the **Python** language.
- As of now **Python** has around **26 implementations** , but the most common are: **Cpython** , **Jython** , **IronPython** and **PyPy**



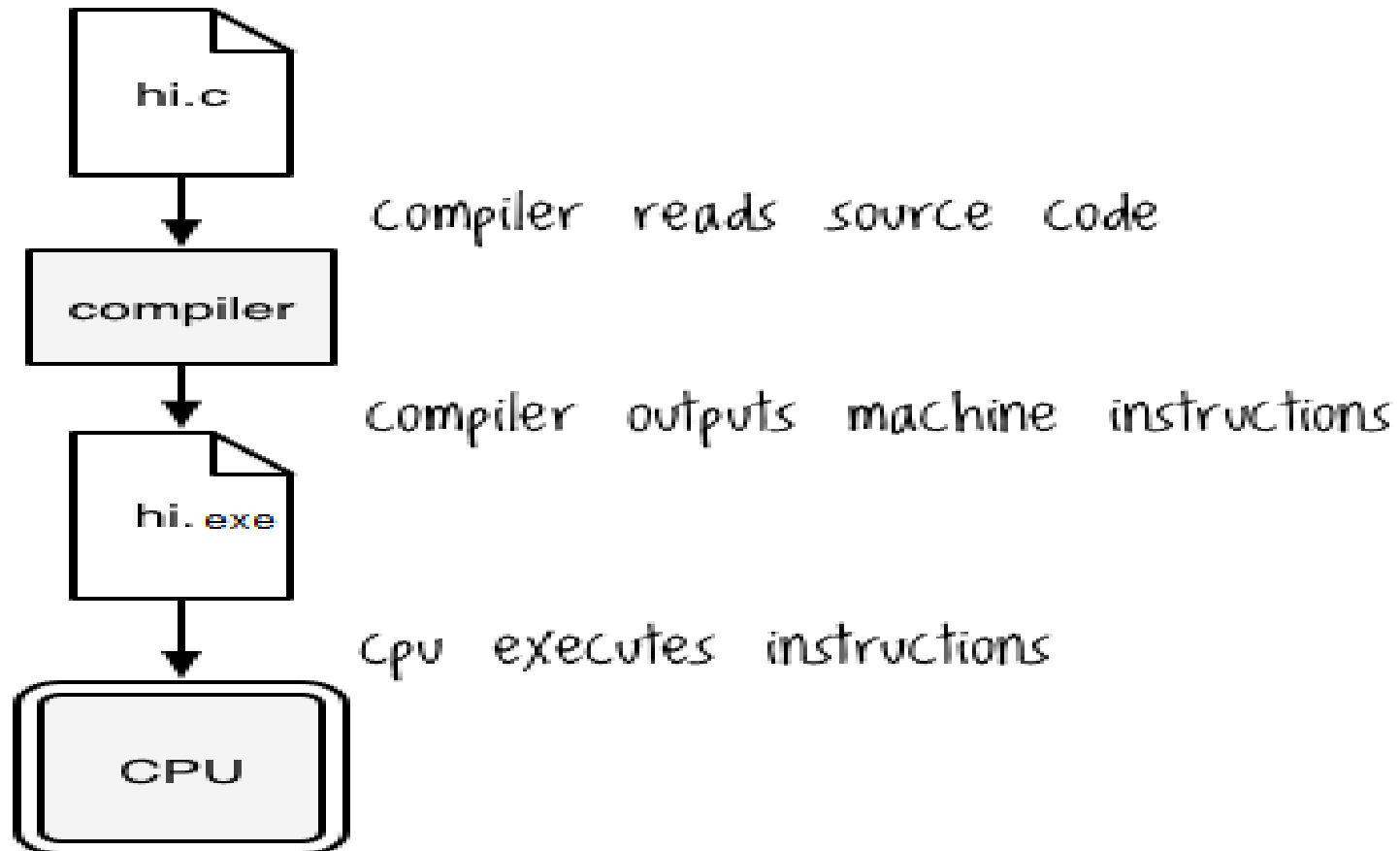
Difference Between Machine Code And ByteCode



- Before proceeding further let us understand the difference between **bytecode** and **machine code**(native code).
- **Machine Code**(*aka native code*)
 - **Machine code** is set of instructions that **directly gets executed** by the **CPU**.
 - Almost all the high level languages such as **C**, **C++** translate the **source code** into **executable machine code** with the help of **Compilers**
 - This **Machine code** is then directly executed by the underlying **Machine** (OS+CPU)



Difference Between Machine Code And ByteCode





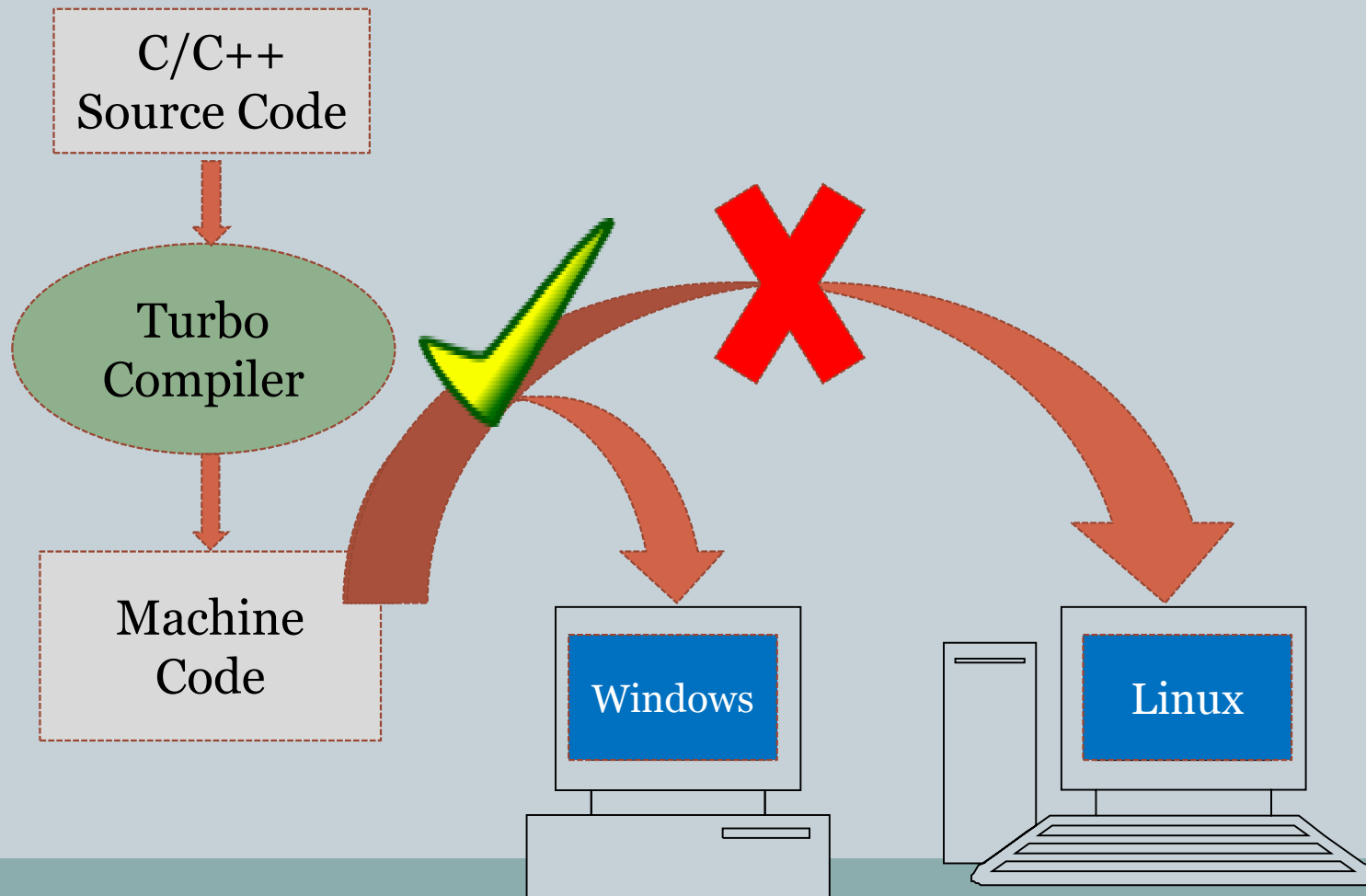
Benefits And Drawbacks Of Machine Code



- The benefit of machine code is that **it runs very fast** because it is in the **native form** i.e a form which is directly understandable to the CPU.
- However the drawback is that it cannot run on another platform which is different than the platform on which the code was compiled.
- In simple words , the **.exe** file of a C program compiled in Windows cannot run on Linux or Mac because every platform (OS+CPU) has it's own **machine code instruction set**.



Benefits And Drawbacks Of Machine Code





Difference Between Machine Code And ByteCode



Bytecode

- **Bytecode** is an **intermediate code** but it is different than **machine code** because it cannot be directly executed by the CPU .
- So whenever the compiler of a language which supports **bytecode** compiles a program , the compiler never generates machine code.
- Rather it generates a machine independent code called the “**bytecode**”.



Difference Between Machine Code And ByteCode



- Now since this bytecode is not directly understandable to the platform(OS & CPU) , *so another special layer of software* is required to convert these **bytecode** instructions to **machine dependent form** .
- This special layer is called **VM** or **Virtual Machine**.



Difference Between Machine Code And ByteCode



- One such language which works on the concept of **VM** is **Java**.
- Thus any such platform for which a **VM** (called **JVM** in java) is available can be used to execute a Java program irrespective of where it has been compiled.



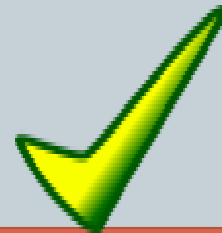
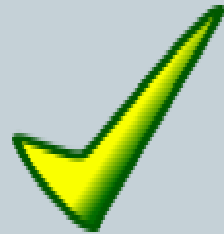
Program Execution in JAVA



Source Code

Compiler

Byte Code



JVM

Windows

JVM

Ubuntu

JVM

MacOS



Benefits And Drawbacks Of ByteCode



- The benefit of **bytecode** is that it makes our program **platform independent** i.e. we only have to write the program once and we can run it any platform provided there is a **VM** available on that platform
- However the **drawback** is that **it runs at a slower pace** because the interpreter inside the **VM** has to translate each **bytecode** instruction to native form and then send it for execution to the CPU.



CPython



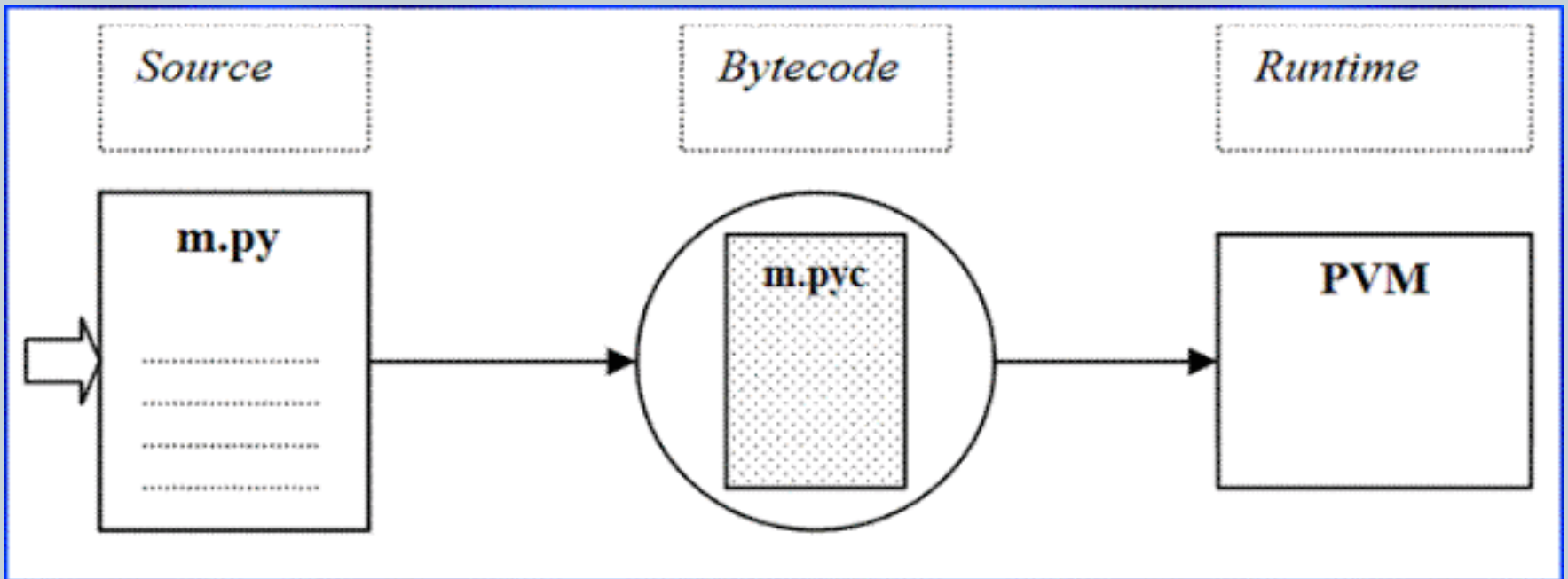
- The default implementation of the Python programming language is **CPython** which is written in **C language**.
- **CPython** is the original **Python** implementation and it is the implementation we will download from [Python.org](https://python.org).
- People call it **CPython** to distinguish it from other Python implementations
- Also we must understand that **Python** is the language and **CPython** is its compiler/interpreter written in **C language** to run the Python code.



CPython



- **CPython** compiles the python source code into intermediate **bytecode**, which is executed by the **CPython virtual machine** also called as the **PVM**.





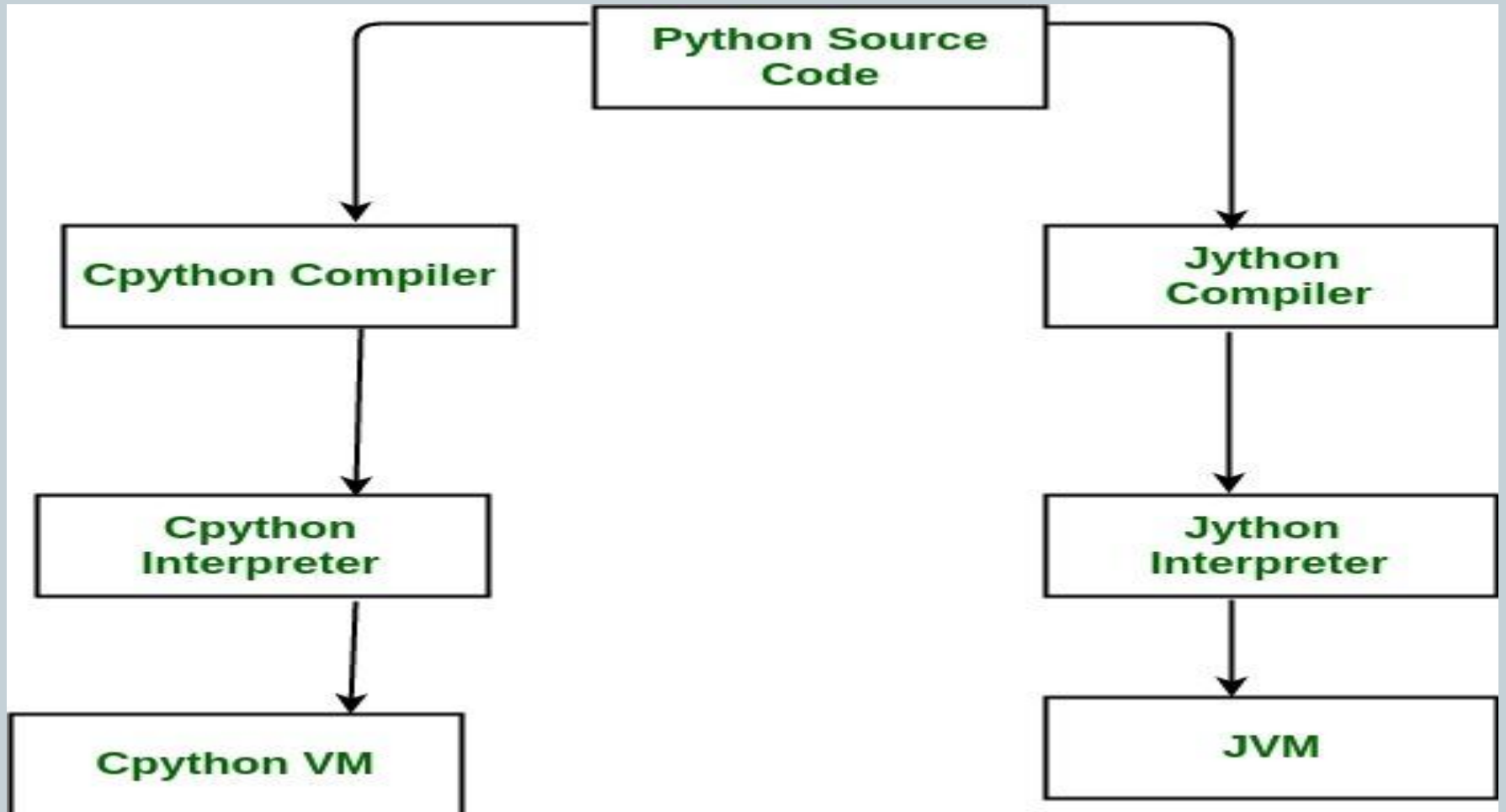
Jython



- The **Jython** system (originally known as JPython) is an alternative implementation of the **Python** language, targeted for integration with the Java programming language.
- **Jython** compiles **Python** source code to Java bytecode and then sends this bytecode to the Java Virtual Machine (JVM).
- Because **Python** code is translated to Java byte code, it looks and feels like a true Java program at runtime.



Jython





IronPython



- A third implementation of **Python**, and newer than both **CPython** and **Jython** is **IronPython**
- **IronPython** is designed to allow **Python** programs to integrate with applications coded to work with Microsoft's .NET Framework for Windows.
- Similar to **Jython**, it uses .Net Virtual Machine which is called as **Common Language Runtime**



- **PyPy** is an implementation of the **Python** programming language written in **Python**.
- It uses a special compiler called JITC ([just-in-time compilation](#)).
- **PyPy** adds **JITC** to **PVM** which makes the **PVM** more efficient and fast by converting bytecode into machine code in much more efficient way than the normal interpreter.



Downloading And Installing Python



- We can use **Python** in 2 ways:
 - Without any IDE , i.e. by simply using notepad for writing the code and running it on command prompt
 - With an IDE like **PyCharm** , **Spyder** , **Visual Studio Code** etc
- Initially we will learn and practice **Python** programs without any IDE and later on we will use **PYCHARM**



Downloading And Installing Python



- **Python's** downloading and installation is fairly easy and is almost same as any other software.
- We can download everything we need to get started with **Python** from the **Python** website called <http://www.python.org/downloads>.
- The website should automatically detect that we're using **Windows** and present the links to the **Windows installer**.



Downloading And Installing Python



If you have **Windows 32 bit** then download the installer by clicking on the button **Download Python 3.7.3**

<https://www.python.org/downloads/>

Python

PSF

Docs

PyPI



Donate



Sea

About

Downloads

Documentation

Community

Success Stories

Download the latest version for Windows

Download Python 3.7.3

Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#),
[Docker images](#)

Looking for Python 2.7? See below for specific releases





Downloading And Installing Python



But if you have windows 64 bit then scroll down and select **python 3.7.3** from the list

https://www.python.org/downloads/



Python releases by version number.

Release version	Release date	Click for more	
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes
Python 3.6.8	Dec. 24, 2018	Download	Release Notes
Python 3.7.1	Oct. 20, 2018	Download	Release Notes
Python 3.6.7	Oct. 20, 2018	Download	Release Notes



Downloading And Installing Python



Now go to the **Files** section and select **windows x86-64 executable installer**
This will download the installer

Files

Version	Operating System	Description	MD5 Sum	File Size	PGP
Gzipped source tarball	Source release		2ee10f25e3d1b14215d56c3882486cf	22973527	SIG
XZ compressed source tarball	Source release		93df27aec0cd18d6d42173e601ffbdfd	17108364	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	5a95572715e0d600de28d6232c656954	34479513	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	4ca0e30f48be690bfe80111daee9509a	27839889	SIG
Windows help file	Windows		7740b11d249bca16364f4a45b40c5676	8090273	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	854ac011983b4c799379a3baa3a040ec	7018568	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a2b79563476e9aa47f11899a53349383	26190920	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	047d19d2569c963b8253a9b2e52395ef	1362888	SIG
Windows x86 embeddable zip file	Windows		70df01e7b0c1b7042aabb5a3c1e2fbd5	6526486	SIG
Windows x86 executable installer	Windows		ebf1644cdc1e000bacc92afa949cfc01	25424128	SIG
Windows x86 web-based installer	Windows		d3944e218a45d982f0abcd93b151273a	1324632	SIG



Downloading And Installing Python



Open the downloads folder and run the file **python-3.7.3.exe** (if you are on 32 bit) or **python-3.7.3-amd64** (if you are on 64bit) by **right clicking** it and selecting **run as administrator**



Downloading And Installing Python



Python 3.6.5 (64-bit) Setup



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the [online tutorial](#) and [documentation](#).

See [what's new](#) in this release.



Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Close

Once the installation is over you will get SETUP WAS SUCCESSFUL message



Testing Python Installation



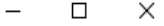
- To verify that Python is installed and working correctly, do the following:
 - Open the command prompt
 - Type the command **python --version**
- In the output we should see the python version number as shown in the next slide



Testing Python Installation



Command Prompt



```
C:\Users\PC>python --version
```

```
Python 3.6.5
```



```
>: \Users\PC>
```