

Python

Module No.	Unit No.	Details	Hrs.	Lab Hrs.	CO
1	Introduction to Python 3.x		01	01	CO 1
	1.1	Features and Applications of Python, Installation of IDE for python		01	
	1.2	Spyder and Jupyter Notebook			
2	Programming with python: Basic Concepts		03	08	CO1
	2.1	Data Types in Python, Strings , format(), print(), Code Block Indentation, Comments, Variables and assignment, Operators in Python, Basic built-in Math functions, Copying Data: Shallow Copy and Deep Copy		02	
	2.2	Data Structures: Tuples, List, Dictionaries, Set, Arrays, Conversion of data structures		06	
3	Programming with python: Decision Making and Functions		03	06	CO2
	3.1	If statement: if, if-else, elif, Repetition using While loop, for loop, break statement		02	
	3.2	What is Regular Expression, Special Symbols and Characters for Regular Expressions, RE Module and functions		02	
	3.3	Functions- Defining a Function, Checking & Setting Parameters, Nested Functions, Lambda and Filter, Map & range functions.		02	
4	Object Oriented Programming using Python programming		05	12	CO3

Module 1

Module No.	Unit No.	Details	Hrs.	Lab Hrs.	CO
1		Introduction to Python 3.x	01	01	CO 1
	1.1	Features and Applications of Python, Installation of IDE for python		01	
	1.2	Spyder and Jupyter Notebook			

Books-

Practical and Oral examination based on laboratory experiments and entire syllabus.

Text Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with country	Edition and Year of Publication
1.	Reema Thareja	<i>Python Programming: Using Problem Solving Approach</i>	Oxford University Press	First Edition 2017, India
2.	Dr. R. Nageswara Rao	<i>Core Python Programming</i>	Wiley Publication.	Second Edition 2018, India
3.	Sheetal Taneja and Naveen Kumar	<i>Python Programing: A Modular Approach</i>	Pearson India	Second Edition 2018, India

Reference Books:

Sr. No.	Name/s of Author/s	Title of Book	Name of Publisher with	Edition and Year of
20-02-2020				

Marking Scheme

- Total Marks =75
- 25 Experiments – Term work
- 20 On screen test
- 15 Mini Project
- 15 Oral

Python Tentative schedule of exam

	Syllabus	Pattern	Tentative Dates
On screen test	Module1 – 3	8+12 (Any 1 out of 2)	24 Apr– 28 Apr
<u>Mini Project</u>		Rubrics	13 Mar – 31 Mar 5 June – 16 June
Oral	Entire syllabus	Rubrics	5 June – 16 June

Is Python trending

bi Binary Informatics

Most Popular In-Demand Programming languages 2019



Developers stay among the most in-demand roles in the tech community, and those skilled in the right programming languages can order pay rates of over \$100,000 as per a report from Tech Republic.



Most Popular In-Demand Programming languages

1. Python

Python is straightforward and has intelligible code to empower software developers to express the idea of utilizing lesser lines of code.



Java

2. Java

Java is the official language of Android development, and 90% of Fortune 500 organizations use Java as a server-side language for back-end development.

3. C++

Is Python trending

- **Top Programming Languages 2020**
- **Python rules the roost, but Cobol gets a pandemic bump**
- Python has held onto its comfortable lead, with Java and C once again coming in second and third place
- **Source: IEEE Spectrum**

Rank	Language	Type	Score
1	Python ▾	🌐💻⚙️	100.0
2	Java ▾	🌐📱💻	95.3
3	C ▾	📱💻⚙️	94.6
4	C++ ▾	📱💻⚙️	87.0
5	JavaScript ▾	🌐	79.5
6	R ▾	💻	78.6
7	Arduino ▾	⚙️	73.2
8	Go ▾	🌐💻	73.1
9	Swift ▾	📱💻	70.5
10	Matlab ▾	💻	68.4

Courtesy:<https://spectrum.ieee.org/at-work/tech-careers/top-programming-language-2020>

Companies using Python

COMPANIES USING PYTHON



Organizations Use Python

- **Web Development** :Google,Yahoo
- **Games** :Battlefield 2, CrystalSpace
- **Graphics** :Walt Disney Feature Animation, Blender3D
- **Science** :National Weather Service, NASA, AppliedMaths
- **Software Development** :Nokia, Red Hat,IBM
- **Education** :University of California-Irvine,SchoolTool
- **Government** :The USA Central Intelligence Agency(CIA)

What is Scripting Language

- A scripting language is a “wrapper” language that integrates OS functions.
 - The interpreter is a layer of software logic between your code and the computer hardware on your machine.
 - scripting languages doesn’t require any compilation and are directly interpreted
- **Scripting Language Features:**
 - Automation of the required process into a program.
 - Fetching information from the provided data sets.
 - Requires less code than modern programming languages
- **Python is scripting language, fast and dynamic.**

History of Python



- Python was conceptualized by **Guido Van Rossum** in the late **1980s**.
- Rossum published the first version of Python code (0.9.0) in February **1991** at the CWI (Centrum Wiskunde& Informatica) in Netherlands , Amsterdam.

History of Python



- Rossum chose the name "**Python**", since he was a big fan of Monty Python's Flying Circus, a BBC comedy series from the 1970s.
- Van Rossum thought he needed a name that was short, unique, and slightly mysterious, so he decided to call the language Python.

History of Python



- Guido van Rossum is a Dutch programmer best known as the author of the Python programming language
- Python is now maintained by a core development team at the institute, although Rossum still holds a vital role in directing its progress.

Releases

- **Python 1.0** -January 1994
 - Python 1.2 -April 10, 1995
 - Python 1.3 -October 12, 1995
 - Python 1.4 -October 25, 1996
 - Python 1.5 -December 31, 1997
 - Python 1.6 -September 5, 2000
 - **Python 2.0** -October 16, 2000
 - Python 2.1 -April 15, 2001
 - Python 2.2 -December 21, 2001
 - Python 2.3 -July 29, 2003
 - Python 2.4 -November 30, 2004
 - Python 2.5 -September 19, 2006
 - Python 2.6 -October 1, 2008
 - Python 2.7 -July 3, 2010
 - **Python 3.0**- December 3, 2008
 - Python 3.1 -June 27, 2009
 - Python 3.2 -February 20, 2011
 - Python 3.3 -September 29, 2012
 - Python 3.4 -March 16, 2014
 - Python 3.5 -September 13, 2015
 - Python 3.6 -December 23, 2016
 - Python 3.7 -June 27, 2018
 - Python 3.8 -14 October 2019
 - Python 3.9 -5 October 2020
 - Python 3.10-Dec 6, 2021
- **Python 1: 1994**
 - **Python 2: 2000**
 - **Python 3: 2008**

Releases

Python releases by version number:

Release version	Release date	
Python 3.10.10	Feb. 8, 2023	 Download
Python 3.11.2	Feb. 8, 2023	 Download
Python 3.11.1	Dec. 6, 2022	 Download
Python 3.10.9	Dec. 6, 2022	 Download
Python 3.9.16	Dec. 6, 2022	 Download
Python 3.8.16	Dec. 6, 2022	 Download
Python 3.7.16	Dec. 6, 2022	 Download
... more releases		

[View older releases](#)

Courtesy:

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

What is Python?

- **Python is a high-level programming language which is:**
- **Interpreted:** Python is processed at runtime by the interpreter.
- **Interactive:** You can use a Python prompt and interact with the interpreter directly to write your programs.
- **Object-Oriented:** Python supports Object-Oriented technique of programming.
- **Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications.

Interpreter vs Compiler

Interpreter translates just one statement of the program at a time into machine code.	Compiler scans the entire program and translates the whole of it into machine code at once.
An interpreter takes very less time to analyze the source code. However, the overall time to execute the process is much slower.	A compiler takes a lot of time to analyze the source code. However, the overall time taken to execute the process is much faster.
An interpreter does not generate an intermediary code. Hence, an interpreter is highly efficient in terms of its memory.	A compiler always generates an intermediary object code. It will need further linking. Hence more memory is needed.

Python Features

- Easy to learn, easy to read and easy to maintain.
- **Portable:** It can run on various hardware platforms and has the same interface on all platforms.
- **Extendable:** You can add low-level modules to the Python interpreter.
- **Scalable:** Python provides a good structure and support for large programs.

Python Features

- Python has support for an **interactive mode of testing and debugging**.
- Python has a broad standard **library cross-platform**.
- Everything in **Python is an Object**: variables, functions, even code. Every object has an ID, a type, and a value.

Python Features

- Python supports **GUI applications**
- Python supports **automatic garbage collection.**
- Python can be easily **integrated** with C, C++, and Java.

Python IDEs and Code Editors

- Firstly understand that IDE and code editor both are different things.
- **Text/Code Editor:**
 - Code editors are the lightweight tool that allows you to
 - write and edit the code with some features such as syntax highlighting and code formatting.
 - It provided fewer features than IDE.

Python IDEs and Code Editors

- An integrated development environment(IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
- An IDE normally consists of a source code editor, build automation tools, and a debugger.
- Most of the modern IDEs have intelligent code completion.

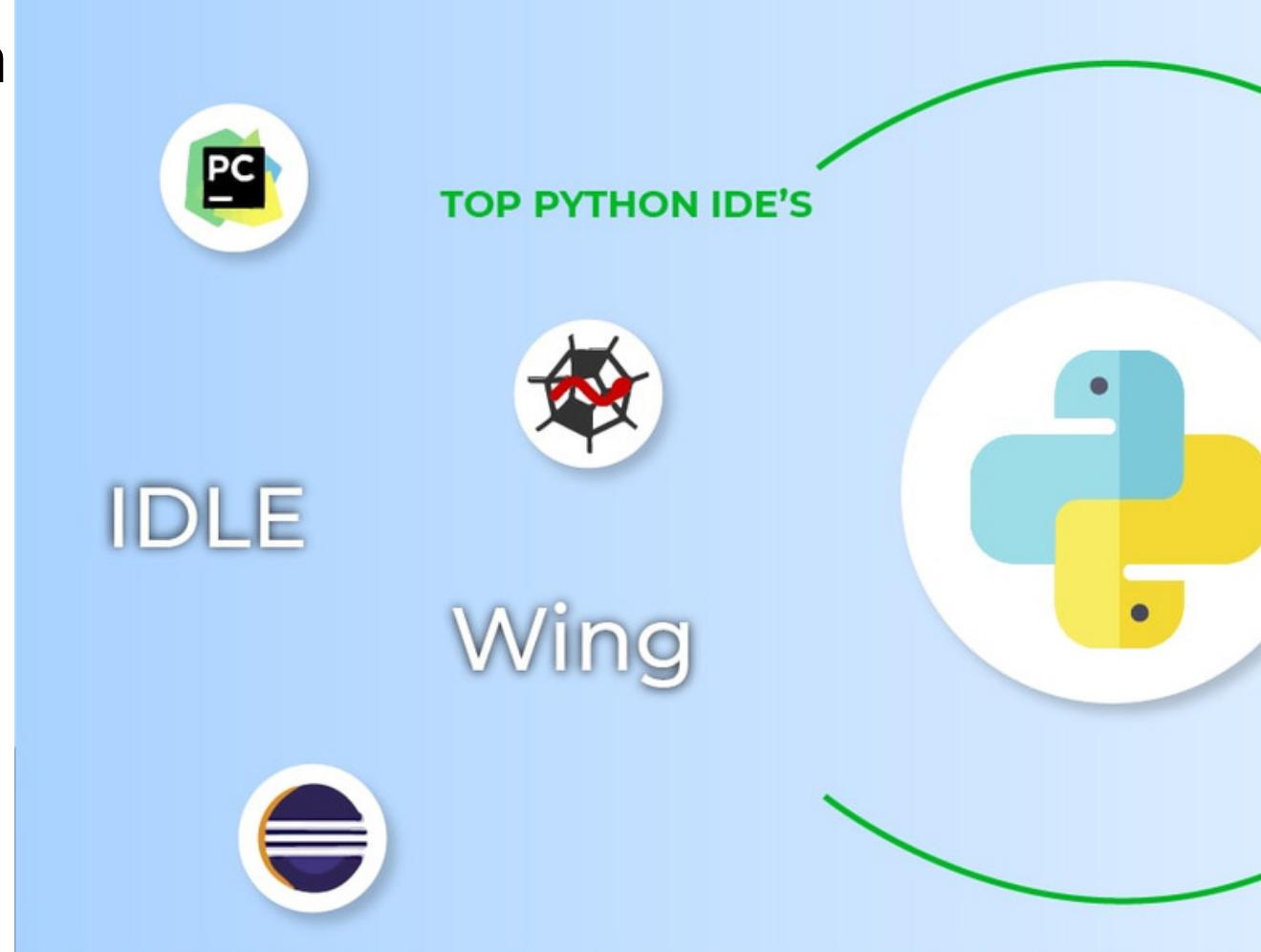
Python IDEs and Code Editors

- **Integrated Development Environment (IDE):**
 - IDEs are full-fledged environment which provide all the essential tools needed for software development.
 - It just doesn't handle the code (for example, write, edit, syntax highlighting and auto-completion) but also provides other features such as debugging, execution, testing, and code formatting that helps programmers.

Python IDEs and Code Editors

List of Best Python IDE

- PyCharm
- Spyder
- Eclipse PyDev
- IDLE
- Wing



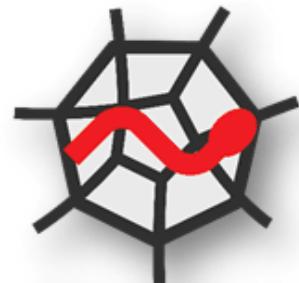
Python IDEs and Code Editors

List of Best Python Code Editor

- Emacs
- Visual Studio Code
- Sublime Text
- Atom
- Vim



Python IDEs and Code Editors



SPYDER



Anaconda



Python IDEs and Code Editors

PyCharm-

- In industries most of the professional developers use PyCharm and it has been considered the **best IDE for python developers.**
- It gives **daily tips to improve your knowledge of how you can use it more efficiently which is a very good feature.**
- It comes in two versions **community version and a professional version where community version is free but the professional version is paid.**

Python IDEs and Code Editors

Spyder-

- Spyder is another good open-source and cross-platform IDE written in Python.
- **Scientific Python Development IDE**
- Most lightweight IDE for Python.
- It is mainly used by data scientists who can integrate with Matplotlib, SciPy, NumPy, Pandas, Cython, IPython, SymPy, and other open-source software.

Python IDEs and Code Editors

Eclipse PyDev-

- Eclipse is one of the most popular IDE among developers which is written in Java but you can install Pydev plugin in eclipse and use it for Python as well.
- The primary focus of this IDE is the analysis of code, debugging in the graphical pattern, refactoring of python code, etc.

Python IDEs and Code Editors

IDLE-

- IDLE is a cross-platform open-source IDE that comes by default with Python so you don't need to worry about the installation or setup.
- IDLE is written in Python
- Suitable for beginner level developers who want to practice on python development.

Python IDEs and Code Editors

IDLE-

- IDLE is lightweight and simple to use
- Can be Used to build simple projects such as web browser game automation, basic web scraping applications, and office automation.
- **This IDE is not good for larger projects so move to some advance IDEs after learning the basics from IDLE.**

Python IDEs and Code Editors

Wing-

- Wing IDE is created by Wingware and it is faster, stable and extremely lightweight cross-platform Python IDE.
- It comes in three editions:
 - **Wing Pro (Free Trial)**: A full-featured commercial version, for professional programmers.
 - **Wing Personal (Paid)**: Free version that omits some features, for students and hobbyists.
 - **Wing 101 (Paid)**: A very simplified free version, for beginners in programming.

Python Shell

Python Shell –running 'python' from the Command Line opens this interactive shell

Jupyter Notebook

- Popular web application or tool which is mainly used for data science projects
- It's well known in the data science community for analyzing, sharing and presenting the information.
- Has Integrated data science libraries (matplotlib, NumPy, Pandas).

Jupyter Notebook

- Jupyter Notebook is a web-based interactive development environment;
- It is easy to use, open-source software that allows you to create and share live code, visualizations, etc.
- It offers you to see and edit your code to create powerful presentations.
- You can also convert your complete work into PDF and HTML files, or you can just export it as a .py file.

Jupyter Notebook

- Project Jupyter is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages.
- Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are
 - Julia,
 - Python and
 - R.

Jupyter Notebook

- Its name and logo are an homage to Galileo's discovery of the moons of Jupiter, as documented in notebooks attributed to Galileo.

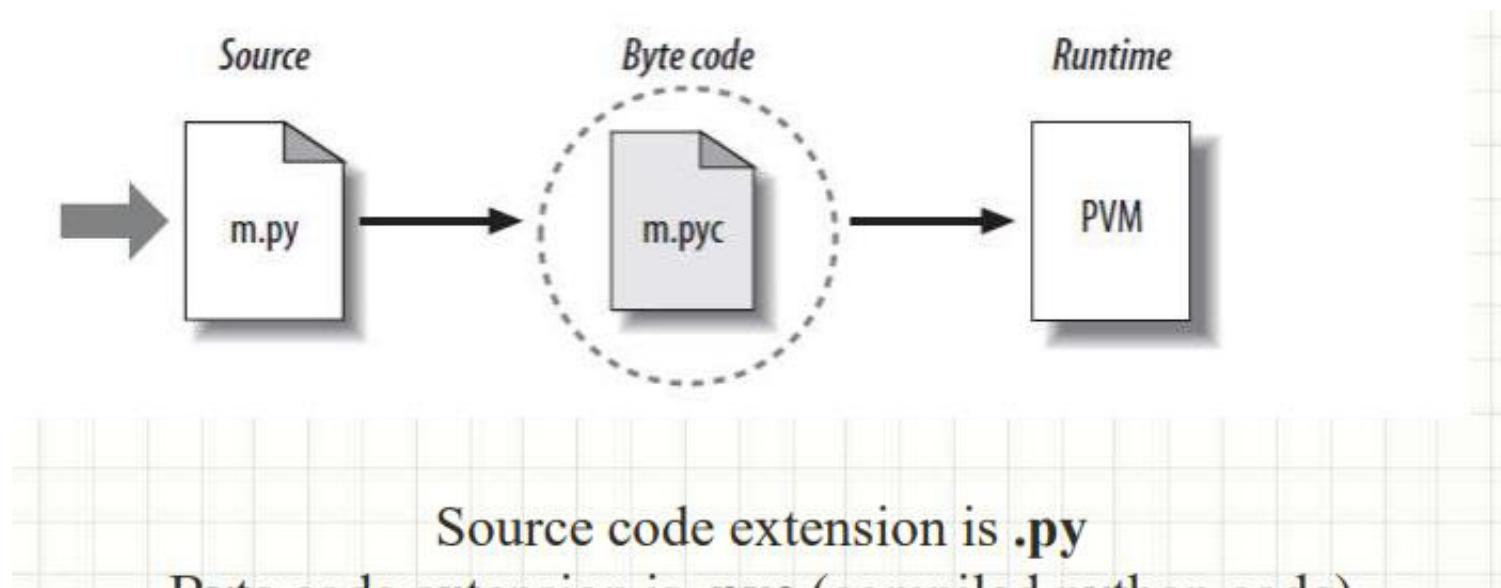


Jupyter Notebook

- Project Jupyter has developed and supported the interactive computing products
 - **Jupyter Notebook,**
 - **JupyterHub, and**
 - **JupyterLab.**
- Jupyter is financially sponsored by NumFOCUS

Python Code Execution

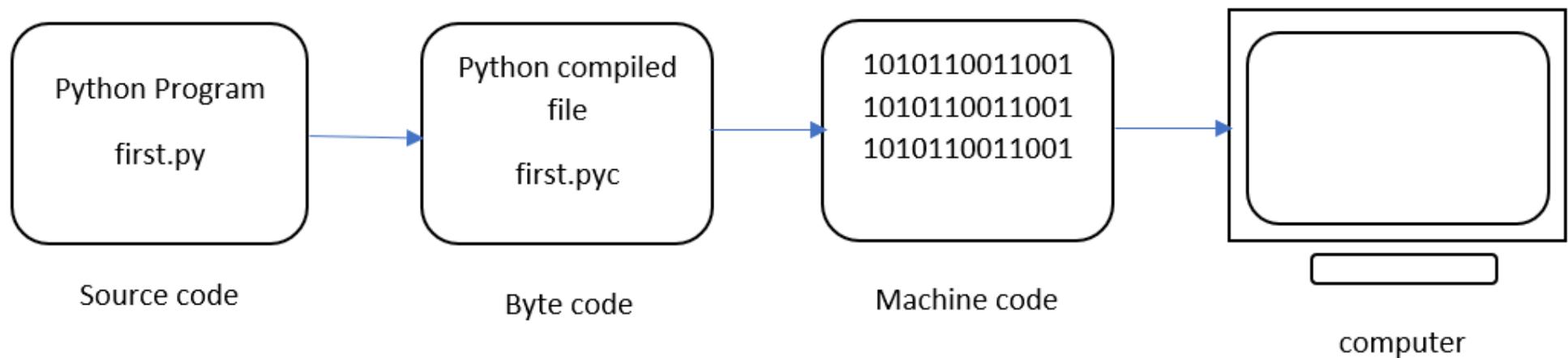
- Python's traditional runtime execution model:
 - source code you type is translated to byte code,
 - which is then run by the Python Virtual Machine.
 - Your code is automatically compiled, but then it is interpreted.



Python Code Execution

- Suppose the a python program is saved as **first.py**.
- Here first is the name and .py is the extension.
- The execution of the Python program involves 2 Steps:
- Compilation
- Interpreter

Python Code Execution



Python Code Execution

Compilation

- The program is converted into **byte code**.
- Byte code is a fixed set of instructions that represent arithmetic, comparison, memory operations, etc.
- It can run on any operating system and hardware.
- The byte code instructions are created in the **.pyc** file.
- The **.pyc** file is not explicitly created as Python handles it internally

Python Code Execution

Interpreter

- The next step involves converting the byte code (.pyc file) into machine code.
- This step is necessary as the computer can understand only machine code (binary code).
- Python Virtual Machine (PVM) first understands the operating system and processor in the computer and then converts it into machine code.
- Further, these machine code instructions are executed by processor and the results are displayed.

Python Code Execution

Interpreter

- However, the interpreter inside the PVM translates the program line by line thereby consuming a lot of time.

Python Installation

To install Python on a Windows machine, follow these steps:

- 1.** Open a web browser and go to <http://www.python.org>.
- 2.** Click the Download link.
- 3.** You should see several links here, with names such as Python 2.5.x and Python 2.5.x Windows installer. Click the Windows installer link to download the installer file. (If you're running on an Itanium or AMD machine, you need to choose the appropriate installer)
- 4.** Store the Windows Installer file somewhere on your computer, such as C:\download\python-2.5.x.msi. (Just create a directory where you can find it later.)
- 5.** Run the downloaded file by double-clicking it in Windows Explorer. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the installation is finished, and you're ready to roll!

Python Installation

- To install Python on a Windows machine, follow these steps:
- 1. Open a web browser and go to <http://www.python.org>.

The screenshot shows the official Python website at <http://www.python.org>. The header features the Python logo and navigation links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the header is a search bar with a 'GO' button and a 'Donate' button. The main content area has a dark blue background. On the left, there's a code snippet demonstrating a Fibonacci series generator:

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
    >>>     a, b = 0, 1
    >>>     while a < n:
    >>>         print(a, end=' ')
    >>>         a, b = b, a+b
    >>>     print()
    >>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

To the right of the code, a yellow callout box highlights the word 'Functions'. Below it, a section titled 'Functions Defined' explains the core of extensible programming. It states: 'The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists.' It includes a link to 'More about defining functions in Python 3'. At the bottom of the page, there are five numbered buttons (1, 2, 3, 4, 5) and a summary statement: 'Python is a programming language that lets you work quickly and integrate systems more effectively. [» Learn More](#)'.

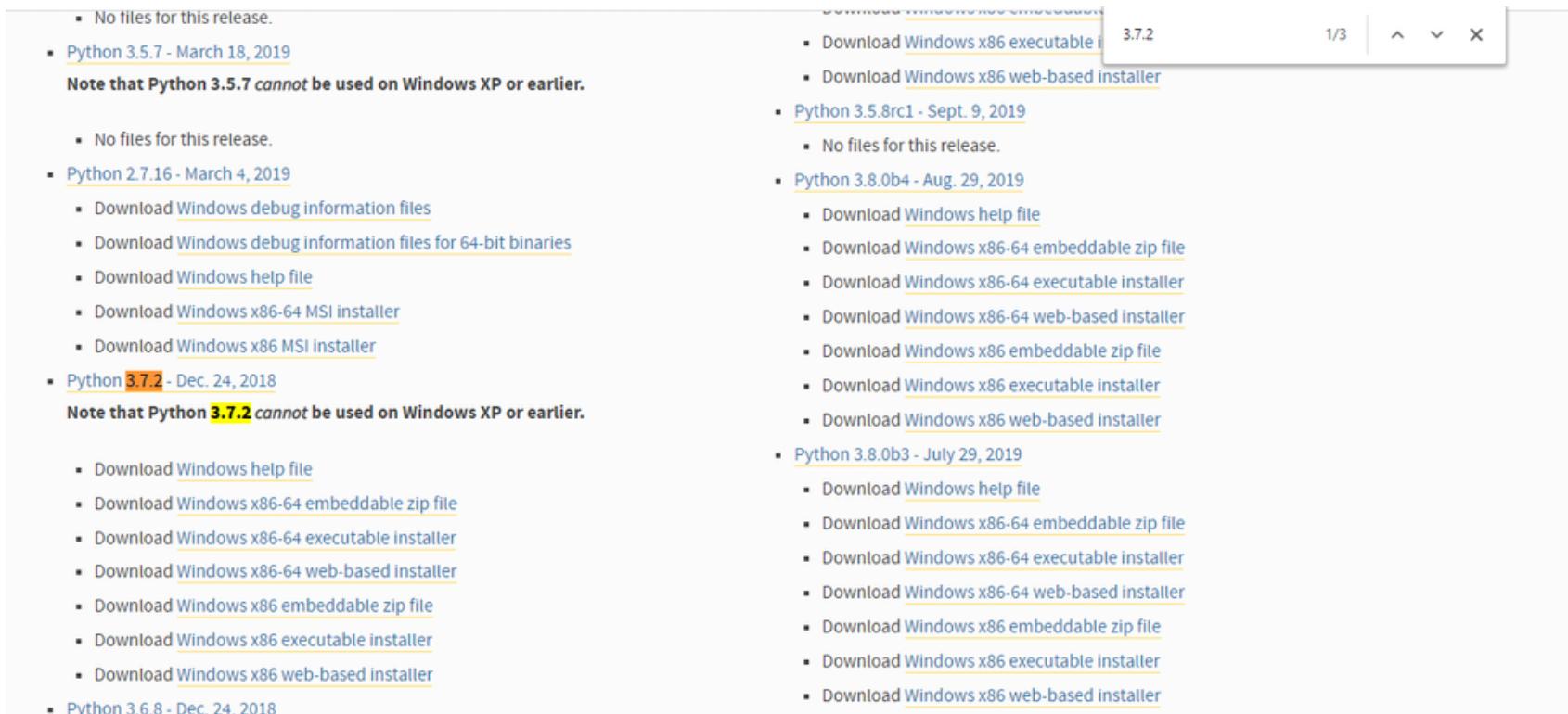
Python Installation

2. Click the Download link.



Python Installation

3. You should see several links here, with names such as Python 2.5.x and Python 2.5.x Windows installer. Click the Windows installer link to download the installer file. (If you're running on an Itanium or AMD machine, you need to choose the appropriate installer)

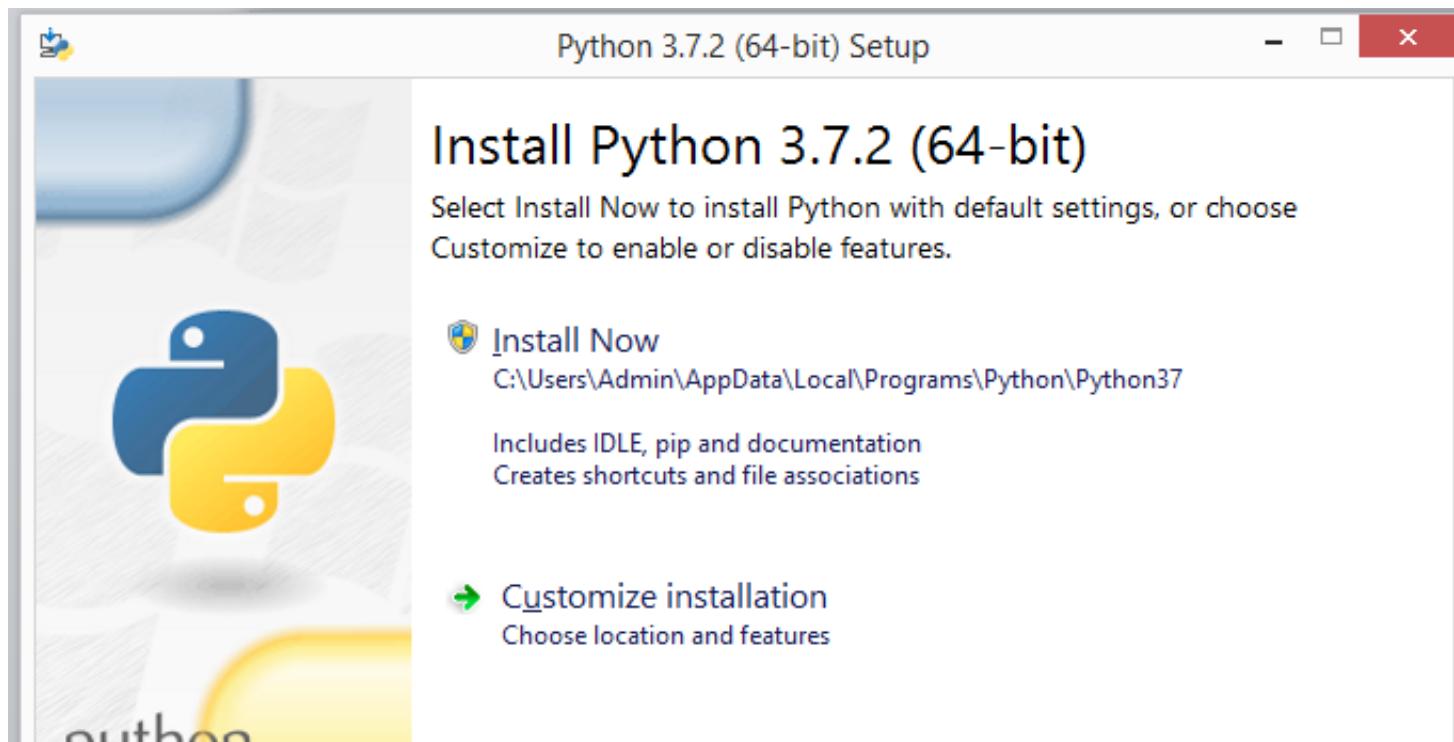


The screenshot shows the Python Download page with a focus on the Windows installer options for Python 3.5.7. The page lists releases from March 18, 2019, down to December 24, 2018. The Python 3.5.7 release is highlighted, showing two download links: "Download Windows x86 executable" and "Download Windows x86 web-based installer". A tooltip-like overlay is shown over the "Windows x86 executable" link, containing the text "3.7.2". The page also includes notes about compatibility with Windows XP and links to Windows help files and debug information files.

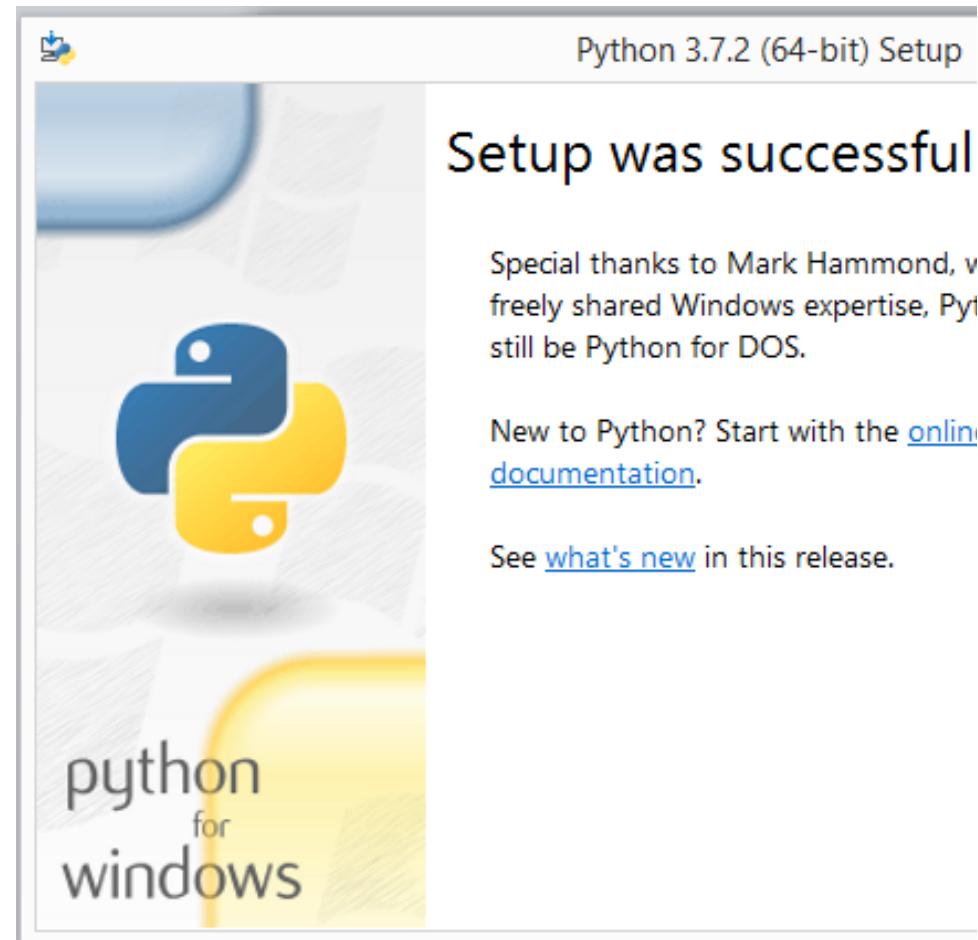
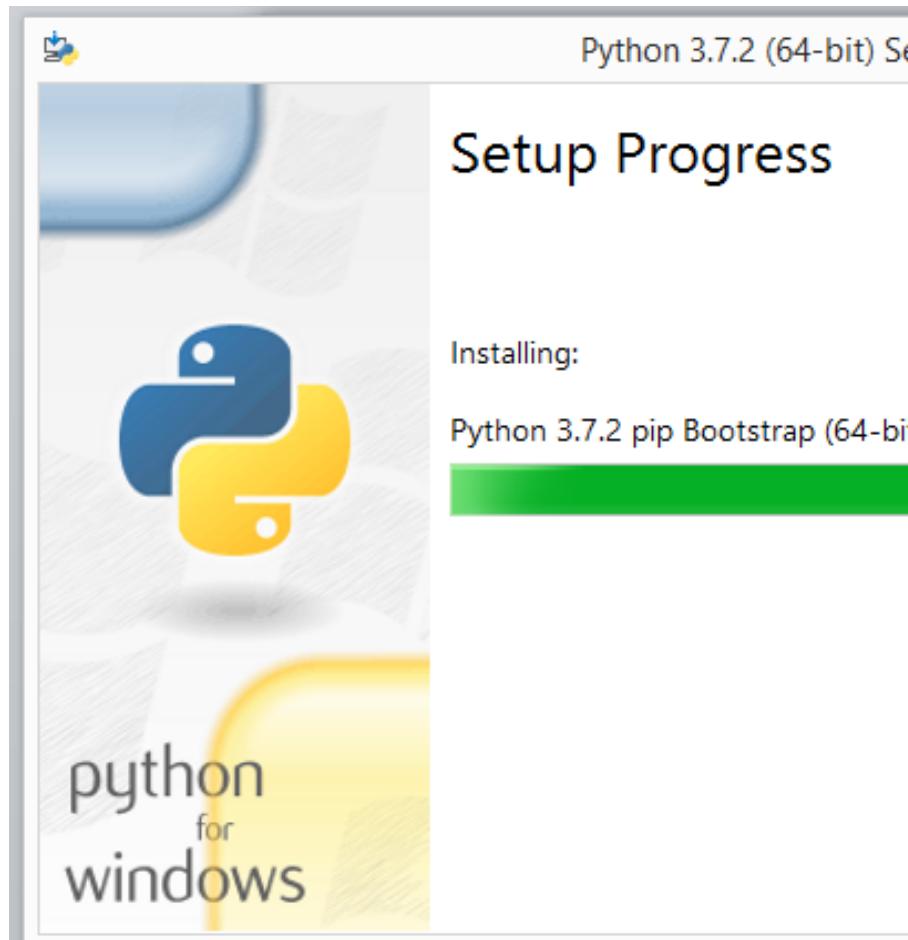
- No files for this release.
- [Python 3.5.7 - March 18, 2019](#)
Note that Python 3.5.7 cannot be used on Windows XP or earlier.
 - No files for this release.
- [Python 2.7.16 - March 4, 2019](#)
 - Download [Windows debug information files](#)
 - Download [Windows debug information files for 64-bit binaries](#)
 - Download [Windows help file](#)
 - Download [Windows x86-64 MSI installer](#)
 - Download [Windows x86 MSI installer](#)
- [Python 3.7.2 - Dec. 24, 2018](#)
Note that Python 3.7.2 cannot be used on Windows XP or earlier.
 - Download [Windows help file](#)
 - Download [Windows x86-64 embeddable zip file](#)
 - Download [Windows x86-64 executable installer](#)
 - Download [Windows x86-64 web-based installer](#)
 - Download [Windows x86 embeddable zip file](#)
 - Download [Windows x86 executable installer](#)
 - Download [Windows x86 web-based installer](#)
- [Python 3.6.8 - Dec. 24, 2018](#)
 - Download [Windows help file](#)
 - Download [Windows x86-64 embeddable zip file](#)
 - Download [Windows x86-64 executable installer](#)
 - Download [Windows x86-64 web-based installer](#)
 - Download [Windows x86 embeddable zip file](#)
 - Download [Windows x86 executable installer](#)
 - Download [Windows x86 web-based installer](#)

Python Installation

- You can easily add Python to Windows path by downloading a recent version of Python, and then checking the box to **Add Python to PATH** during the installation.

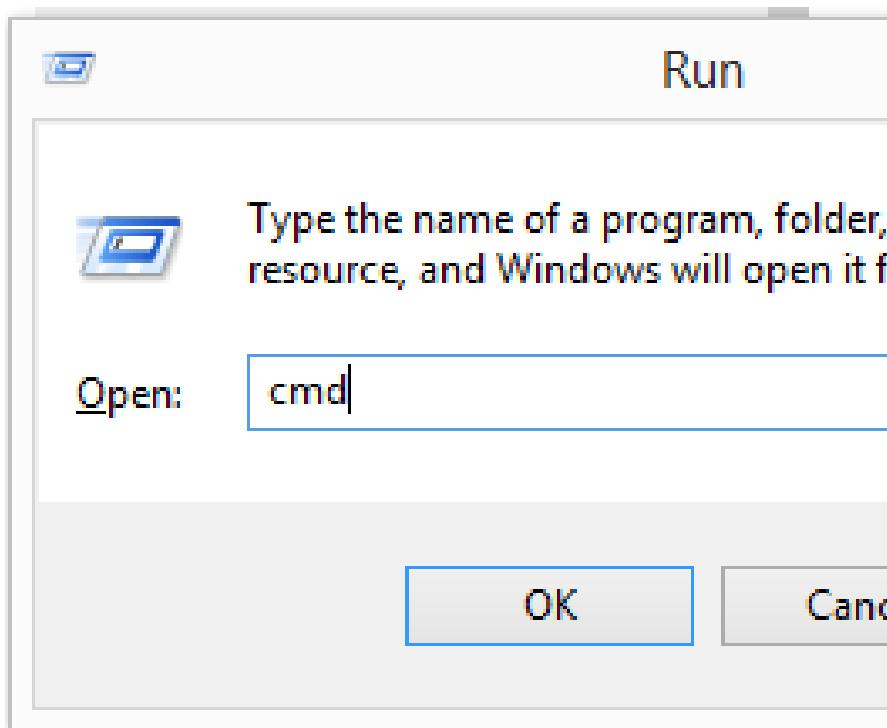


Python Installation



Running Python Shell

- To run the Python Shell, open the command prompt or power shell on Windows and terminal window on mac, **write python and press enter.**
- A Python Prompt comprising of three greater-than symbols >>> appears, as shown below.

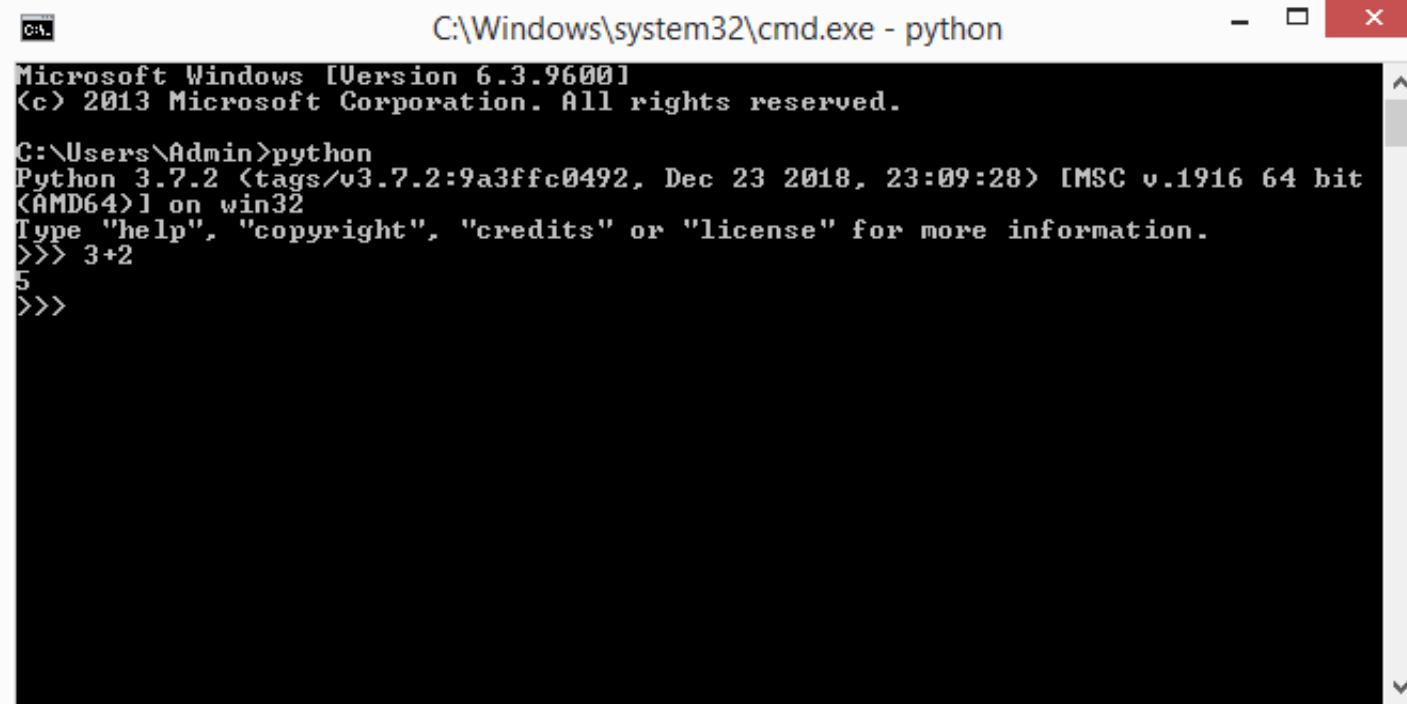


```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Admin>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28)
[AMD64] on win32
Type "help", "copyright", "credits" or "license" for more information
>>>
```

Running Python Shell

- Now, you can enter a single statement and get the result.
- For example, enter a simple expression like $3 + 2$, press enter and it will display the result in the next line, as shown below.



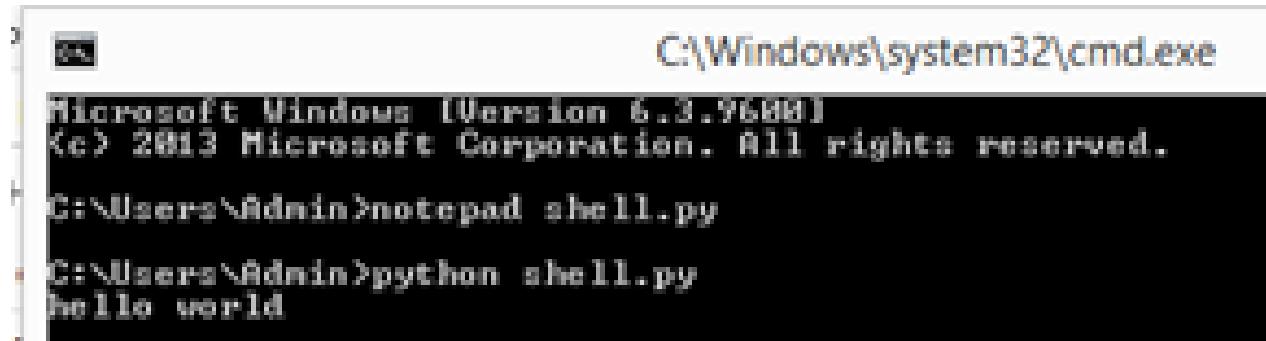
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - python". The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Admin>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
AMD64] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 3+2
5
>>>
```

Executing a .py program from command prompt

- Open cmd prompt
- Open notepad, Create a .py file
- execute using python filename



A screenshot of a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window shows the following text:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Admin>notepad shell.py
C:\Users\Admin>python shell.py
hello world

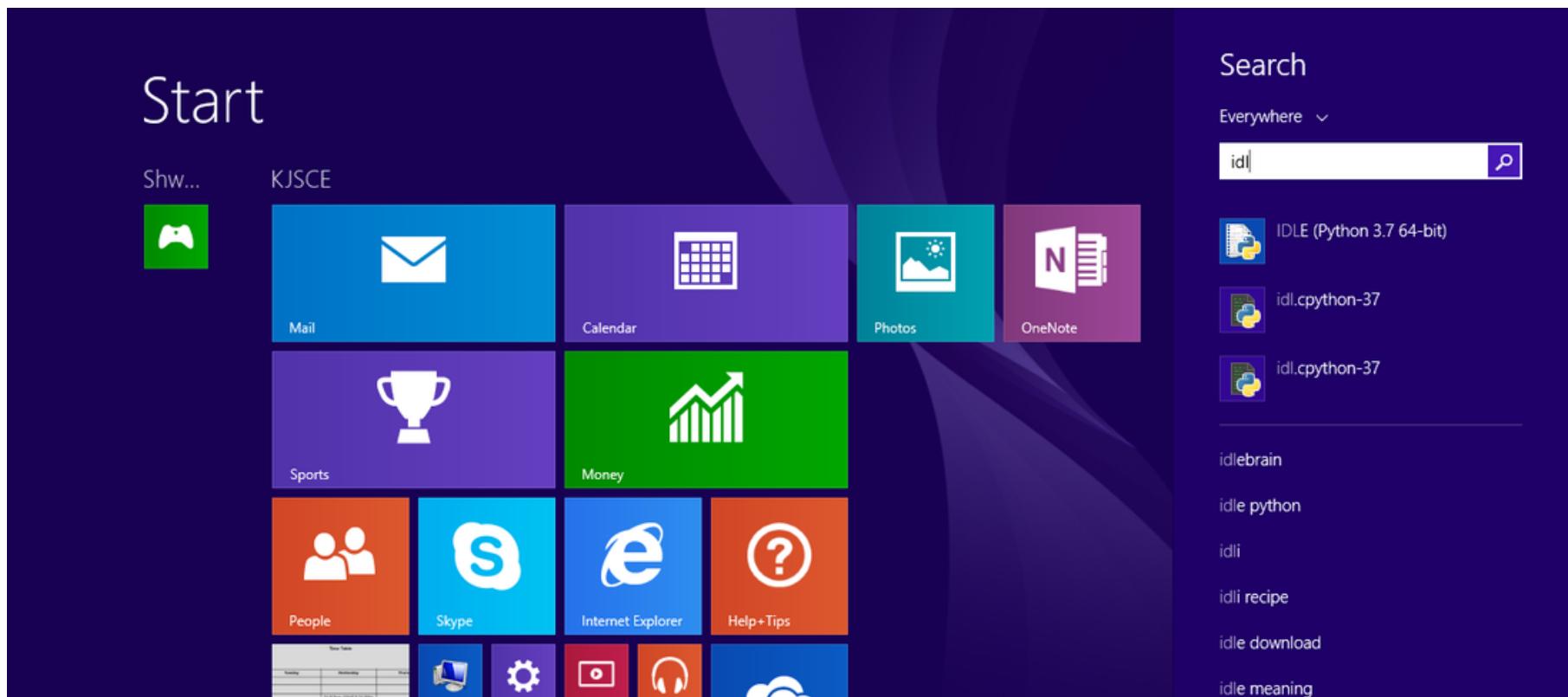
Getting started with Python IDLE

- If you've recently downloaded Python onto your computer, then you may have noticed a new program on your machine called **IDLE**.
- You might be wondering, "What is this program doing on my computer? I didn't download that!" While you may not have downloaded this program on your own, IDLE comes bundled with every Python installation.
- It's there to help you get started with the language right out of the box.

Getting started with Python IDLE

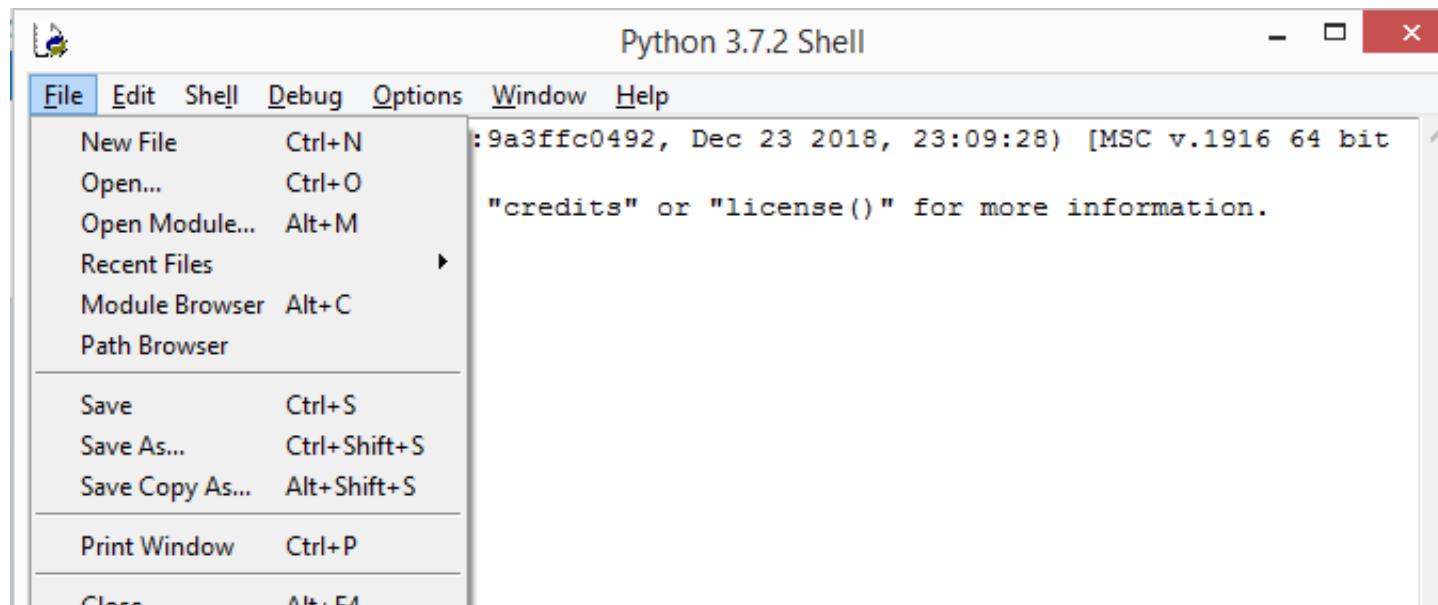
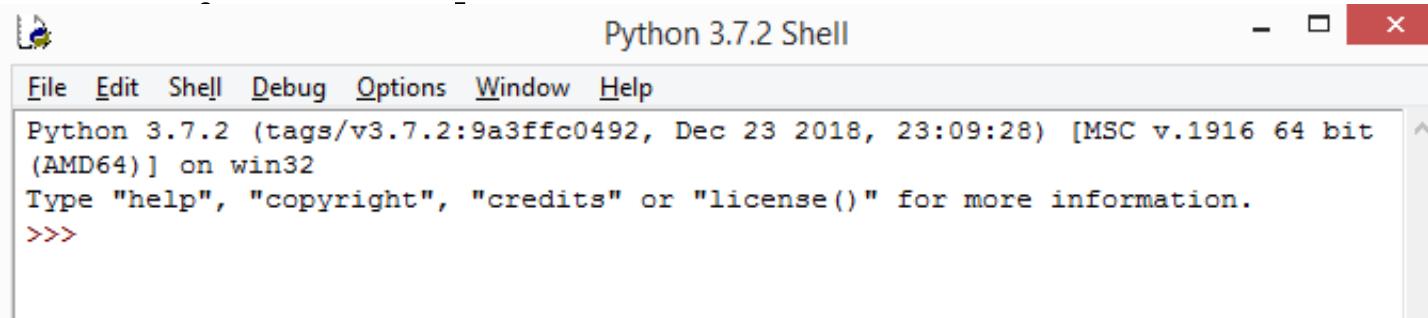
What Is Python IDLE?

Every Python installation comes with an **Integrated Development and Learning Environment**, which you'll see shortened to IDLE or even IDE.



Getting started with Python IDLE

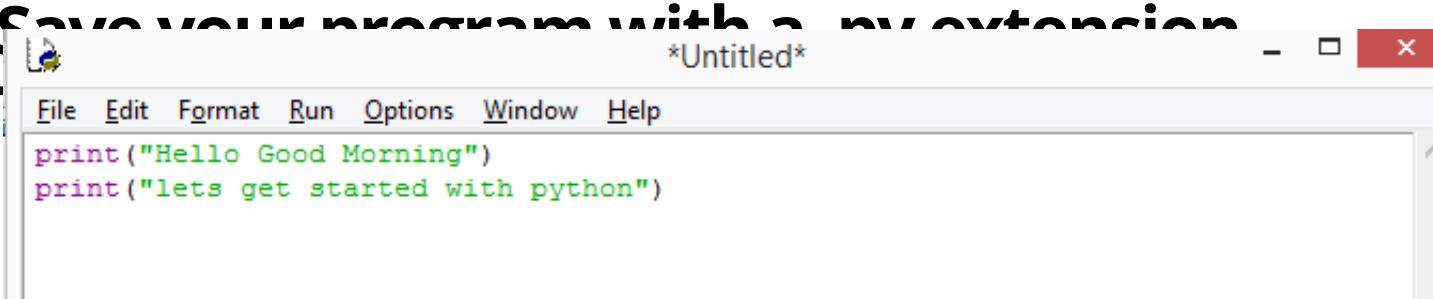
This is how IDLE looks



Getting started with Python IDLE

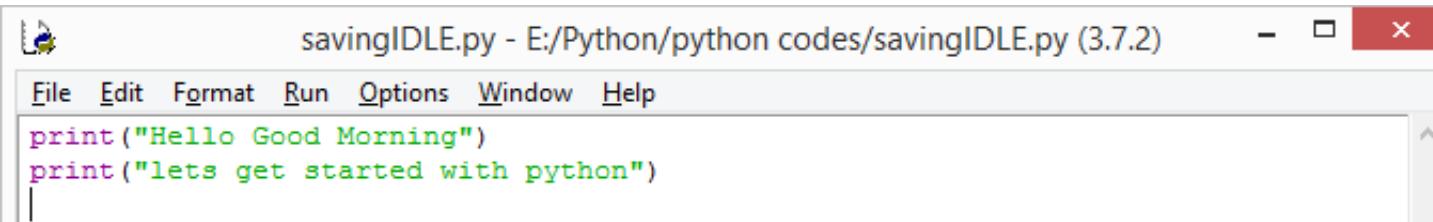
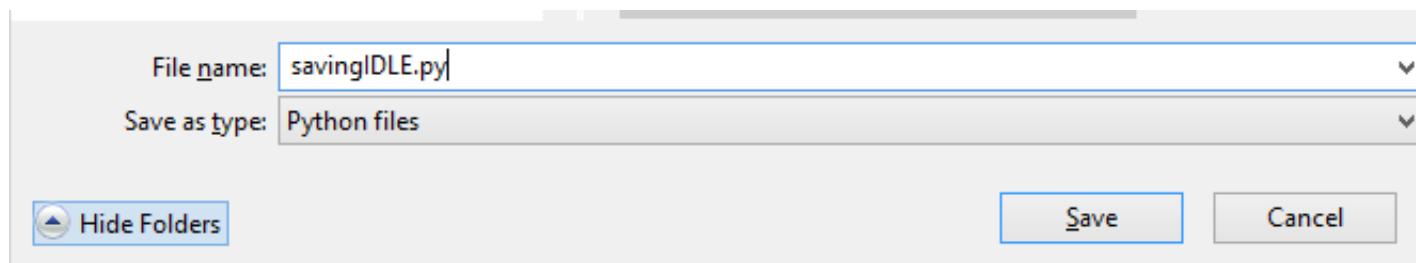
New File will be opened
Write your code

Save your program with a .py extension



The screenshot shows the Python IDLE interface. The title bar says "Untitled". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following Python code:

```
print("Hello Good Morning")
print("lets get started with python")
```

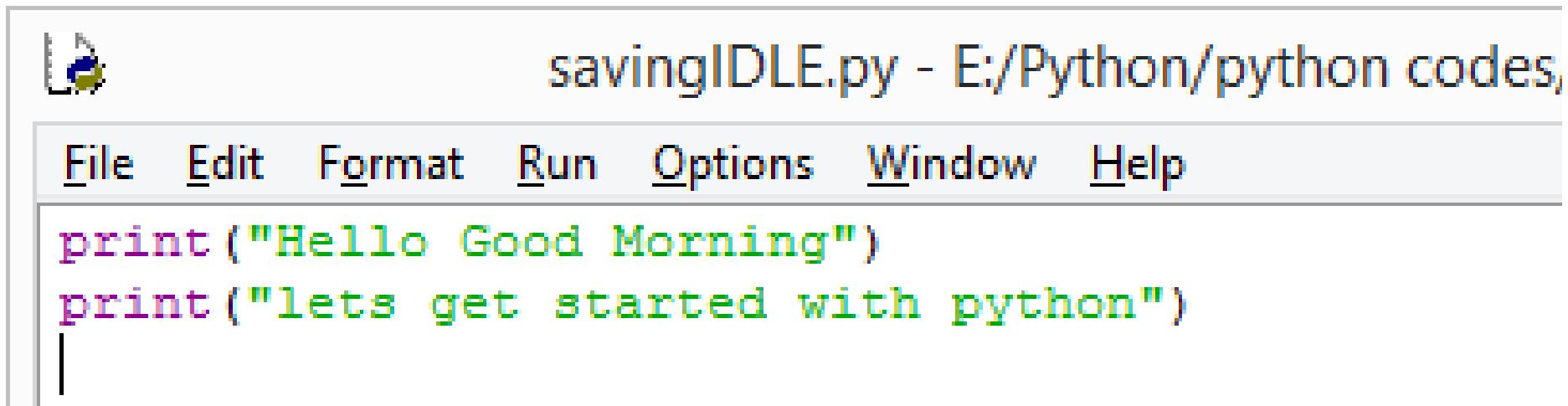


The screenshot shows the Python IDLE interface again. The title bar now says "savingIDLE.py - E:/Python/python codes/savingIDLE.py (3.7.2)". The menu bar is identical to the previous screen. The code editor contains the same Python code as before:

```
print("Hello Good Morning")
print("lets get started with python")
```

Getting started with Python IDLE

- The bar along the top of the program window contains three pieces of important information:
 - **The name** of the file that you're editing
 - **The full path** to the folder where you can find this file on your computer
 - **The version** of Python that IDLE is using

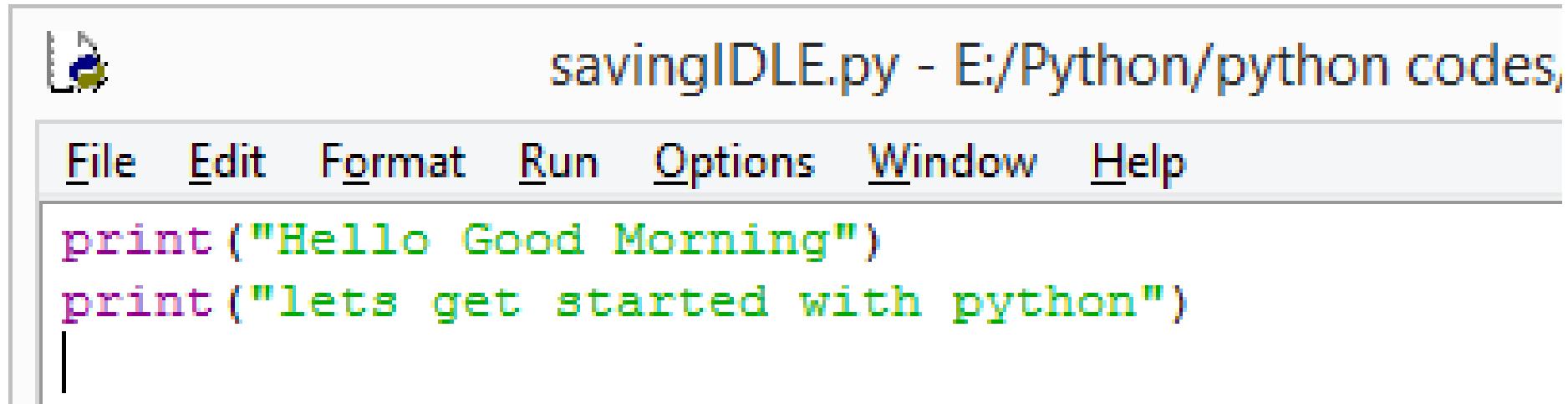


A screenshot of the Python IDLE application. The title bar shows the file name "savingIDLE.py - E:/Python/python codes,". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor displays the following Python script:

```
print ("Hello Good Morning")
print ("lets get started with python")
```

Getting started with Python IDLE

- Here,
- You're editing the file savingIDLE.py,
- Located in the pythoncodes folder.
- The Python version is 3.7.2, which you can see in parentheses.



The screenshot shows the Python IDLE interface. The title bar reads "savingIDLE.py - E:/Python/python codes,". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following Python script:

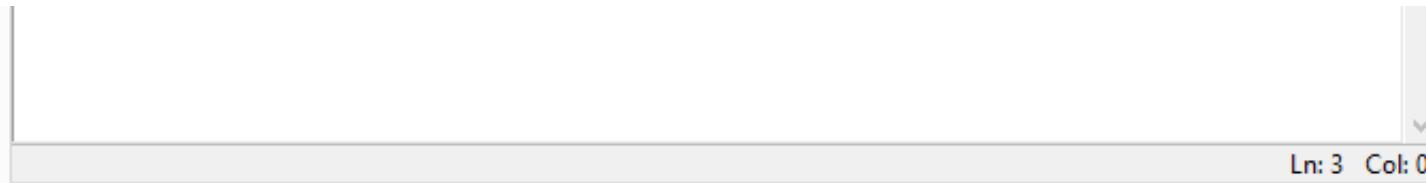
```
print ("Hello Good Morning")
print ("lets get started with python")
```

Getting started with Python IDLE

There are also two numbers in the bottom right corner of the window:

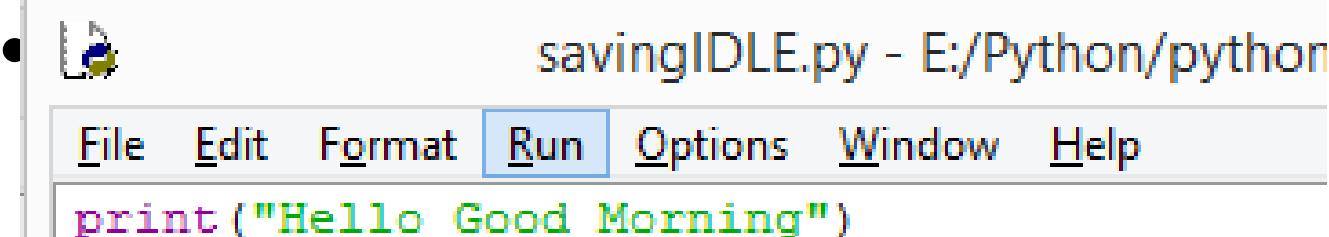
Ln: shows the line number that your cursor is on.

Col: shows the column number that your cursor is on.



Getting started with Python IDLE

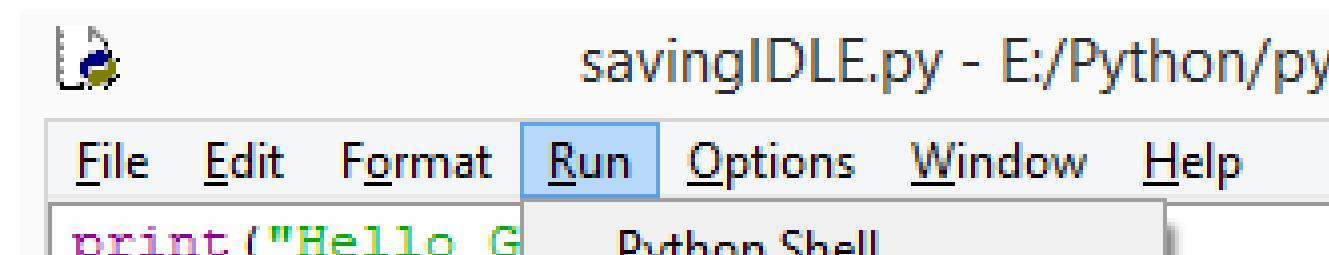
- For execution-Click on Run
- Select Run Module



savingIDLE.py - E:/Python/python

File Edit Format Run Options Window Help

```
print ("Hello Good Morning")
```



savingIDLE.py - E:/Python/pyt

File Edit Format Run Options Window Help

```
print ("Hello G Python Shell
```

```
>>>
=====
RESTART: E:/Python/python codes/savingIDLE.py =====
Hello Good Morning
lets get started with python
>>> |
```

Running using Python Shell

- Running the same program saved via IDLE on command prompt
- Is it possible?
- Yes
- Navigate to that particular path
- Type python filename.py to execute the code

```
C:\Users\Admin>e:
```

```
E:\>cd /Python/python codes
```

```
E:\Python\python codes>python savingIDLE.py
Hello Good Morning
lets get started with python
```

```
E:\Python\python codes>
```

Kick starting Jupyter Notebook

- Type jupyter online on google

The screenshot shows a Google search results page with the query "jupyter online" entered in the search bar. The results are filtered by "All". There are approximately 82,80,000 results. The top result is a link to "Project Jupyter | Try Jupyter - Jupyter Notebook" from jupyter.org. Below it is another link to "Project Jupyter | Home" from jupyter.org. A third result is "Online Jupyter Notebooks - CoCalc" from cocalc.com. A "People also ask" section is visible at the bottom.

Python Releases for Windows | P x | Search results - shweta.chachra@... x | jupyter online - Google Search x +

google.com/search?q=jupyter+online&oq=jupyter+online&aqs=chrome..69i57j0l3j0i395l6.5867j1j7&sourceid=chrome&ie=UTF-8

Google jupyter online

All News Images Books Videos More Settings Tools SafeSearch on

About 82,80,000 results (0.59 seconds)

jupyter.org > try ▾

[Project Jupyter | Try Jupyter - Jupyter Notebook](#)

You can try Jupyter out right now, without installing anything. Select an example below and you will get a temporary Jupyter server just for you, running on ...

jupyter.org ▾

[Project Jupyter | Home](#)

The Jupyter Notebook - The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, ...

cocalc.com > doc > jupyter-notebook ▾

[Online Jupyter Notebooks - CoCalc](#)

CoCalc is an online web service where you can run Jupyter notebooks right inside your browser. You can privately share your notebook with your project ...

People also ask

Jupyter Notebook

- Select Jupyter Lab icon

Help the Jupyter Community by participating in the 2021 Survey

Try Jupyter

You can try Jupyter out right now, without installing anything. Select an example below and you will get a temporary Jupyter server just for you, running on mybinder.org. If you like it, you can install Jupyter yourself.

Try Classic Notebook



A tutorial introducing basic features of Jupyter notebooks and the

Try JupyterLab



JupyterLab is the new interface for Jupyter notebooks and is ready for

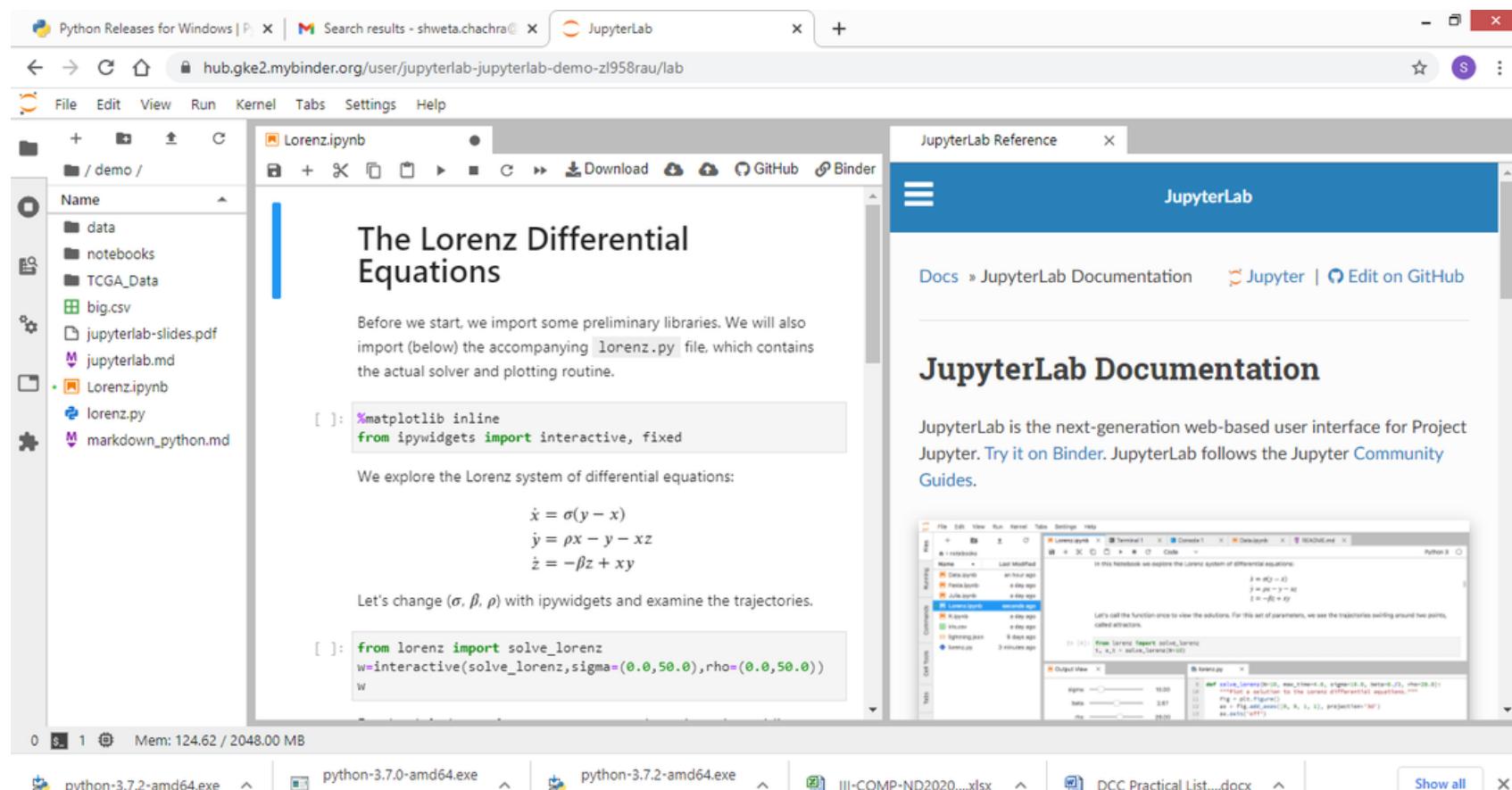
Try Jupyter with Julia



A basic example of using Jupyter with Julia.

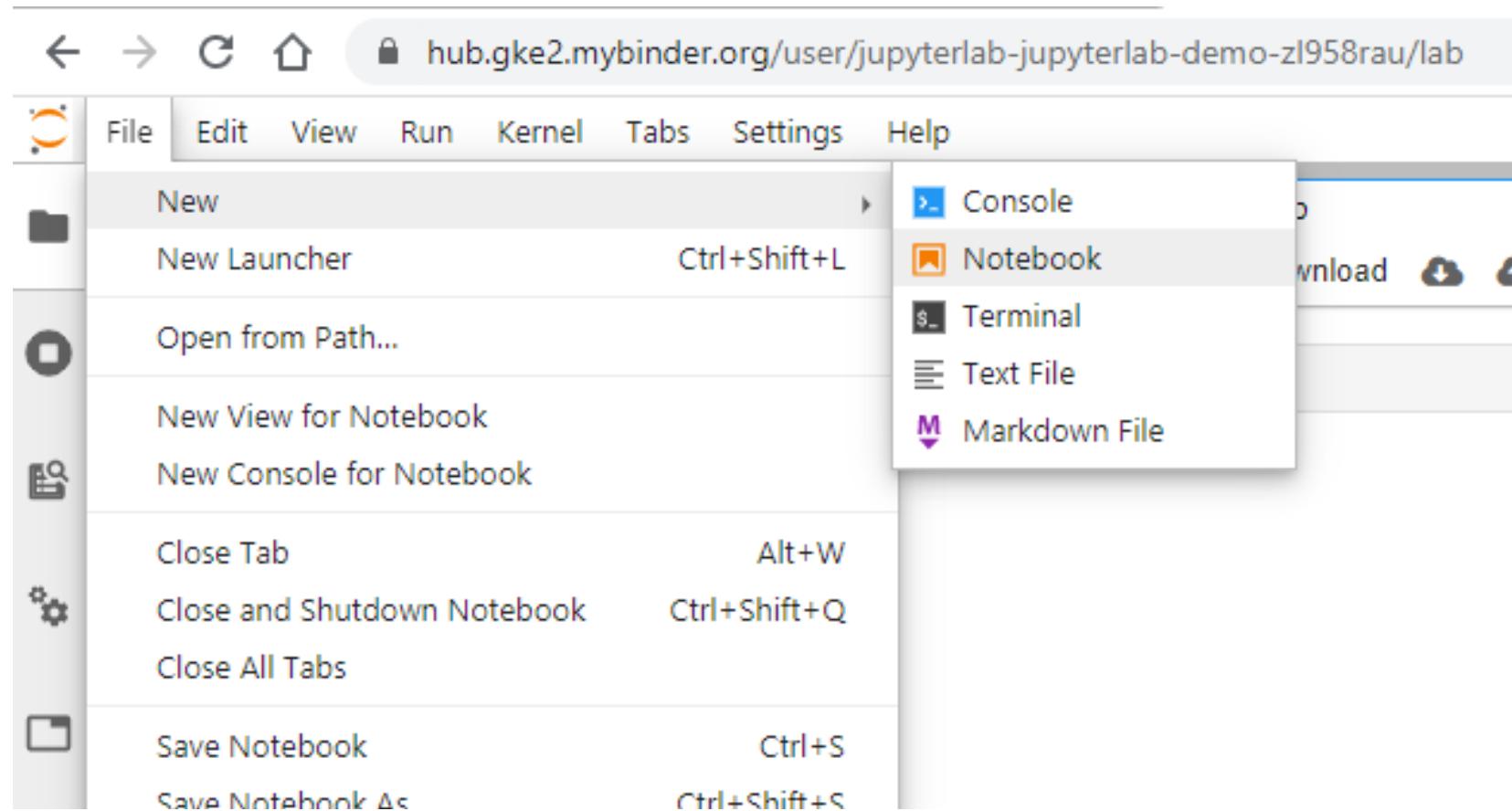
Jupyter Notebook

- On Selecting Jupyter Lab icon



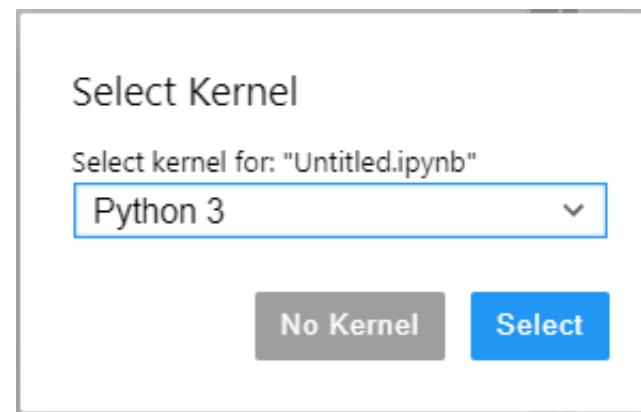
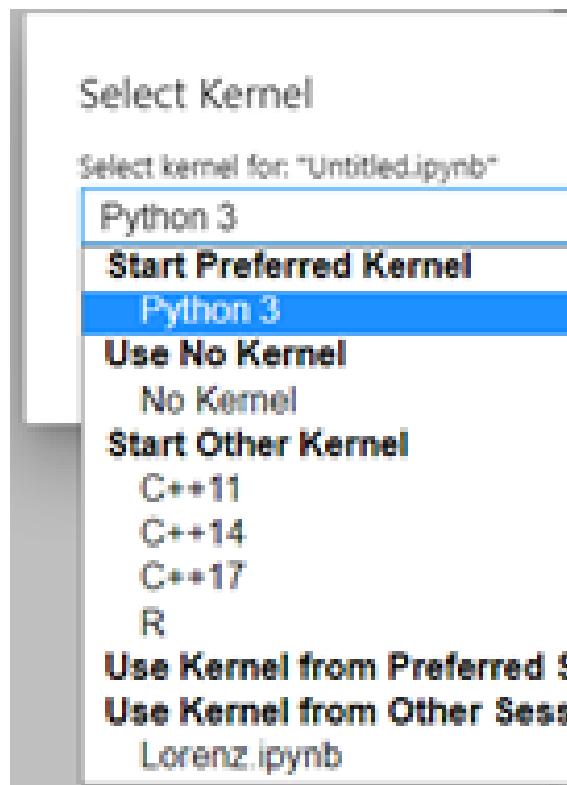
Jupyter Notebook

- Open Jupyter notebook



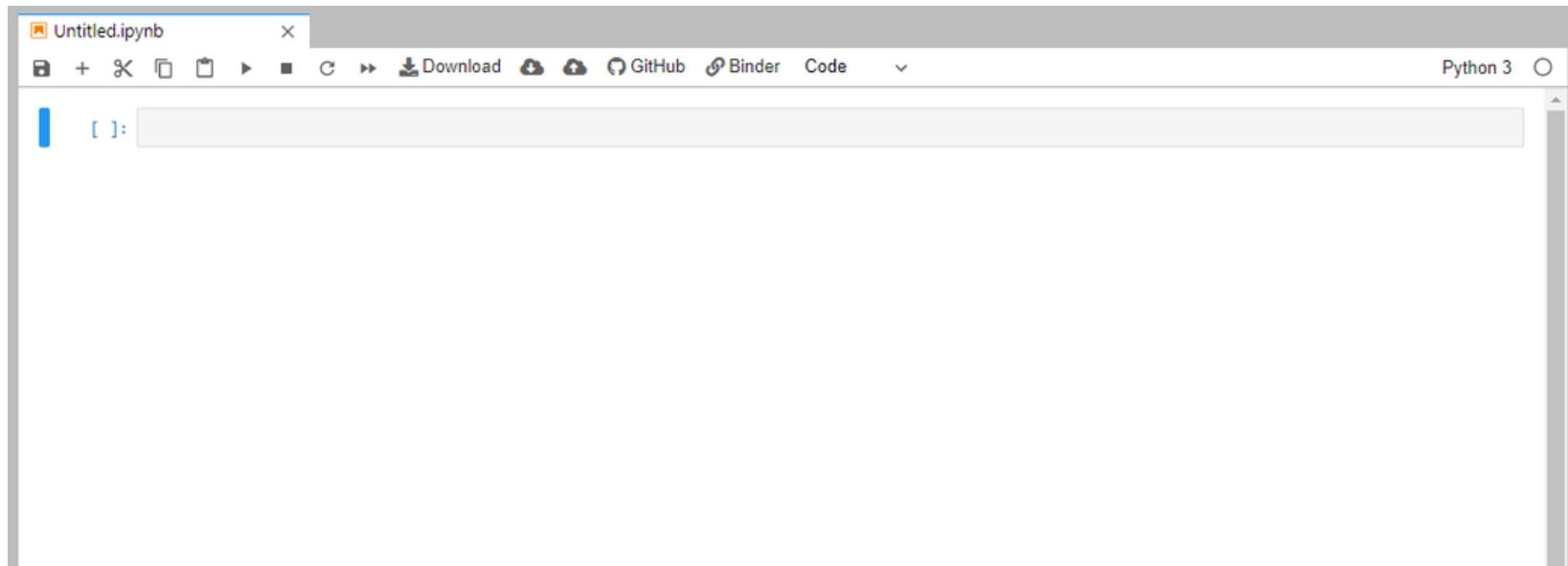
Jupyter Notebook

- Select Kernel



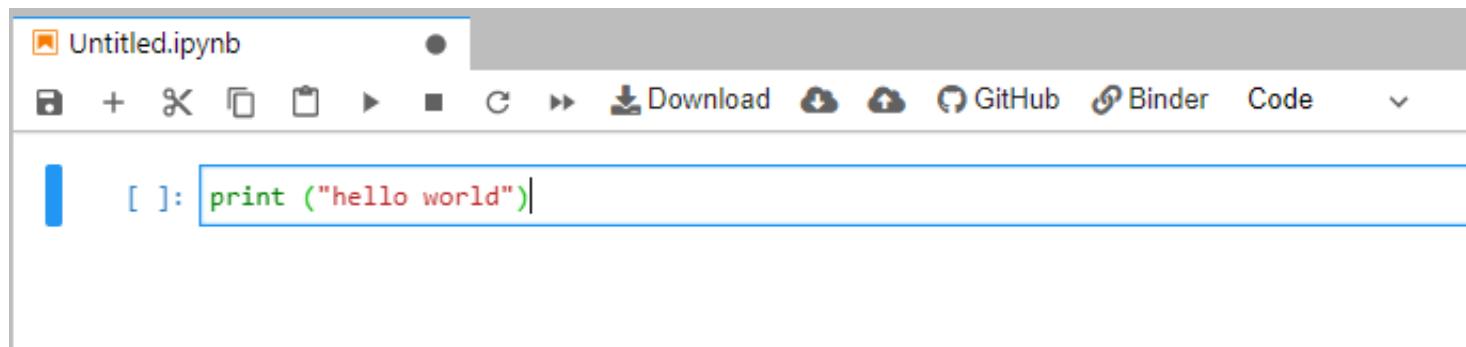
Jupyter Notebook

- The Cell in which we write code



Jupyter Notebook

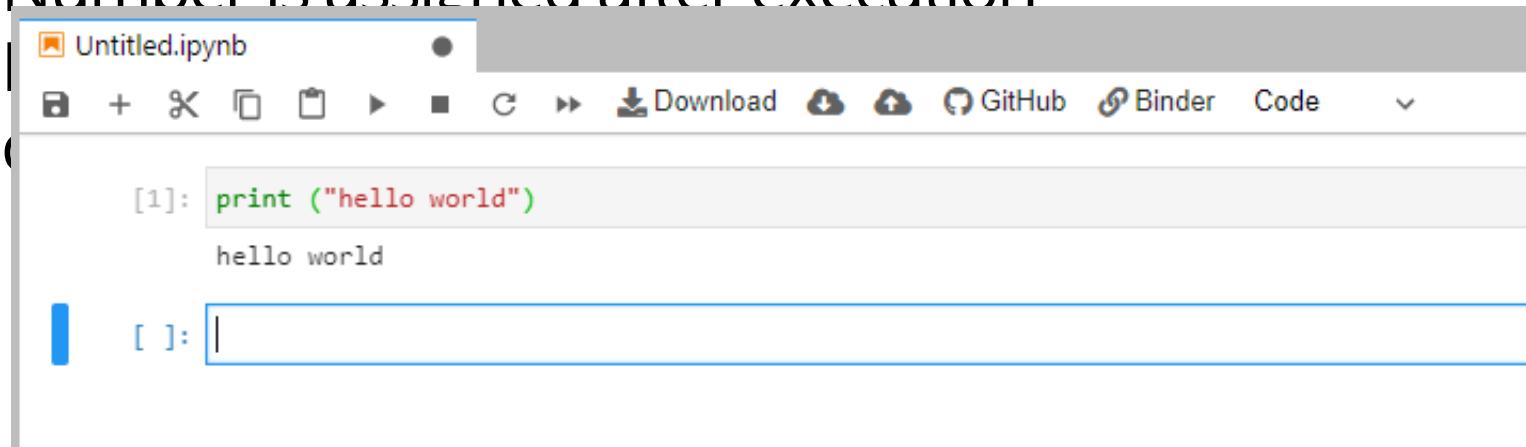
- One line code or multi line code written in the cell will be executed at a time



A screenshot of a Jupyter Notebook interface. The title bar shows "Untitled.ipynb". The toolbar includes standard file operations like New, Open, Save, and Delete, along with Download, GitHub, and Binder buttons. Below the toolbar, a code cell is active, indicated by a blue vertical bar on the left. The cell contains the Python code: []: print ("hello world")

Jupyter Notebook

- Click on Play icon
- Number is assigned after execution



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Untitled.ipynb
- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), a play button (highlighted in blue), a cell type selector, and other tools like Download, GitHub, and Binder.
- Cell 1:** [1]: `print ("hello world")`
Output: hello world
- Cell 2:** []: (An empty cell with a cursor)

- On Jupyter, File gets saved by ipynb extension by default
-



Anaconda

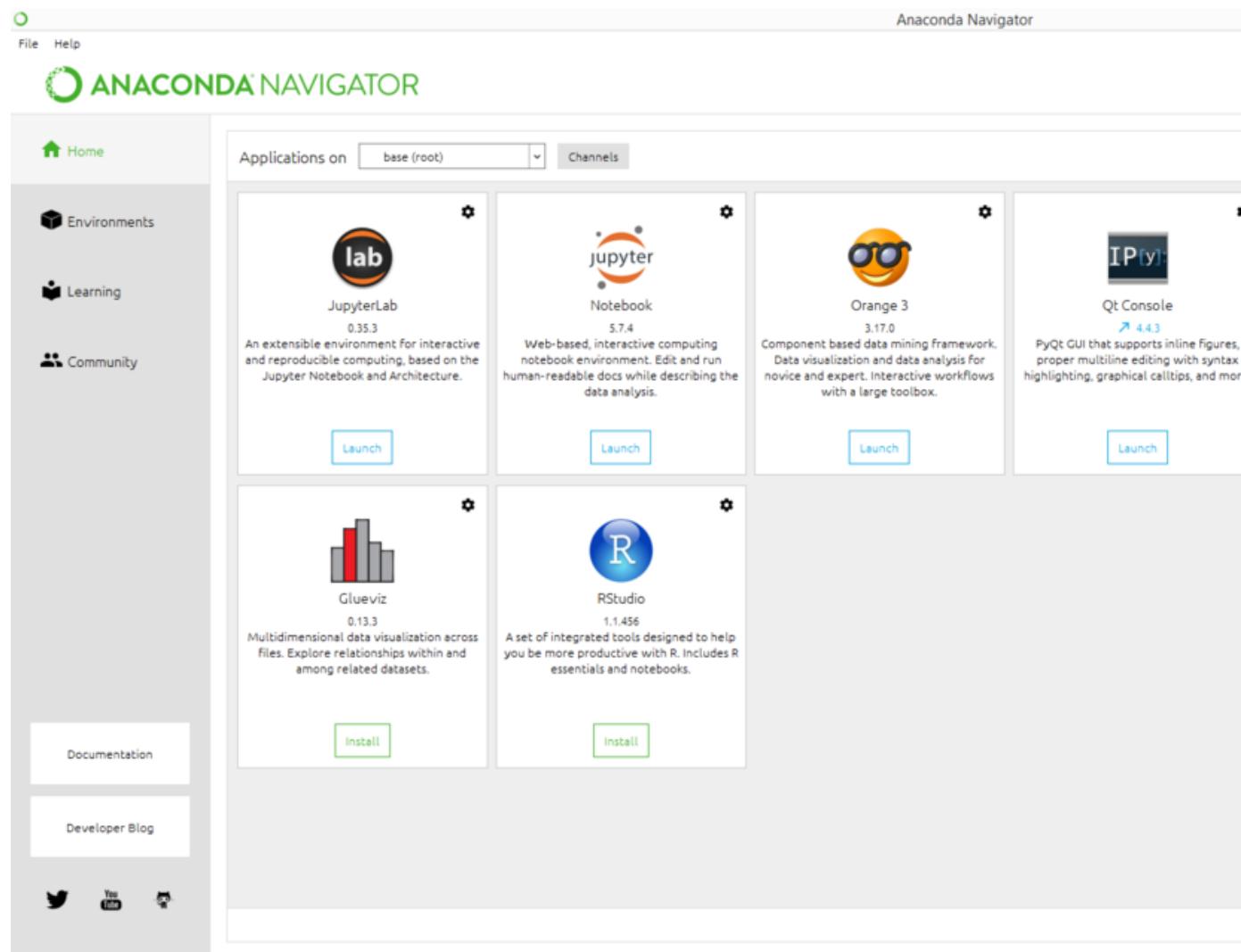
- Anaconda is a distribution of the Python and R programming languages
 - for scientific computing
 - data science,
 - machine learning applications,
 - large-scale data processing,
 - predictive analytics, etc.
 - that aims to simplify package management and deployment.
 - The distribution includes data-science packages suitable for Windows, Linux, and macOS.

Anaconda Navigator

- Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution
 - that allows users to launch applications and manage conda packages, environments and channels without using command-line commands

Anaconda Navigator

- The following applications are available by default in Navigator:
 - JupyterLab
 - Jupyter Notebook
 - QtConsole
 - Spyder
 - Glue
 - Orange
 - RStudio
 - Visual Studio Code



Implementations of Python

- **Cpython**
- **Jython**
- **IronPython**
- **PyPy**

Cpython

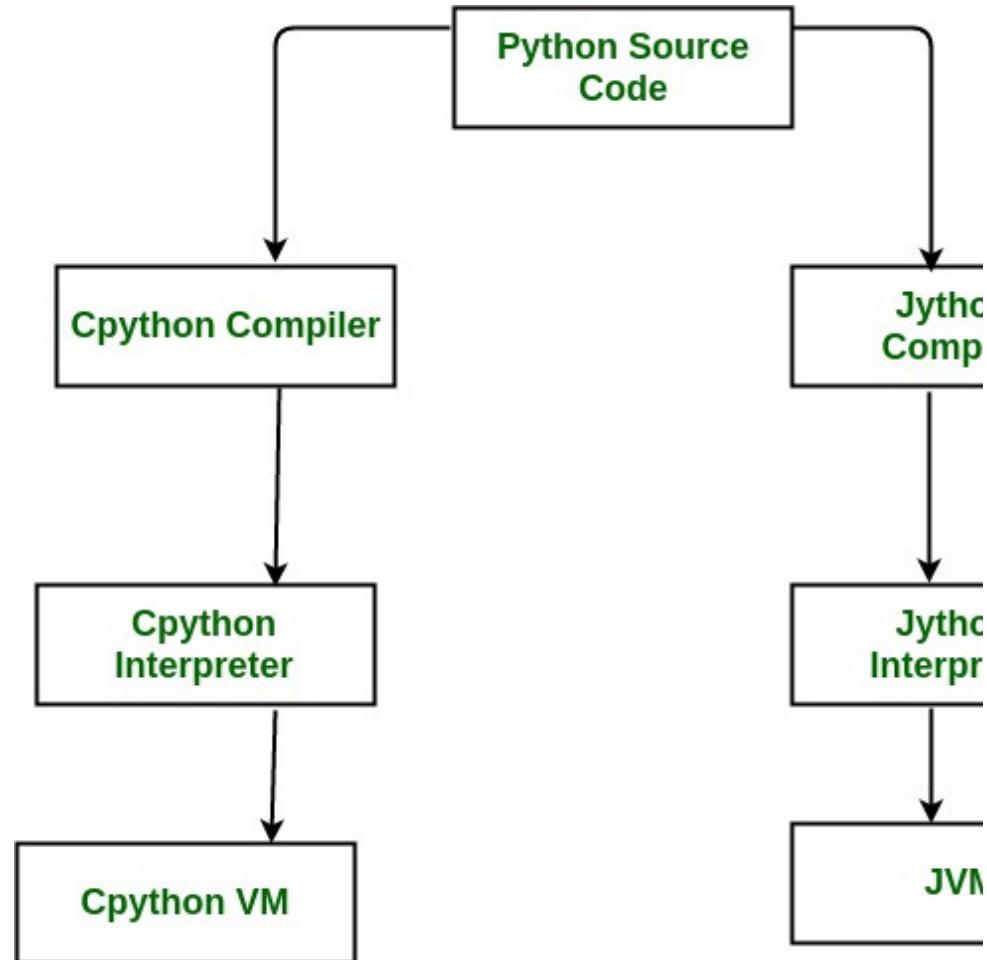
- The default implementation of the Python programming language is Cpython.
- As the name suggests Cpython is written in C language.
- Cpython compiles the python source code into intermediate bytecode, which is executed by the Cpython virtual machine.

Cpython

- CPython is distributed with a large standard library written in a mixture of C and Python.
- CPython provides the highest level of compatibility with Python packages and C extension modules.
- **All versions of the Python language are implemented in C because CPython is the reference implementation.**

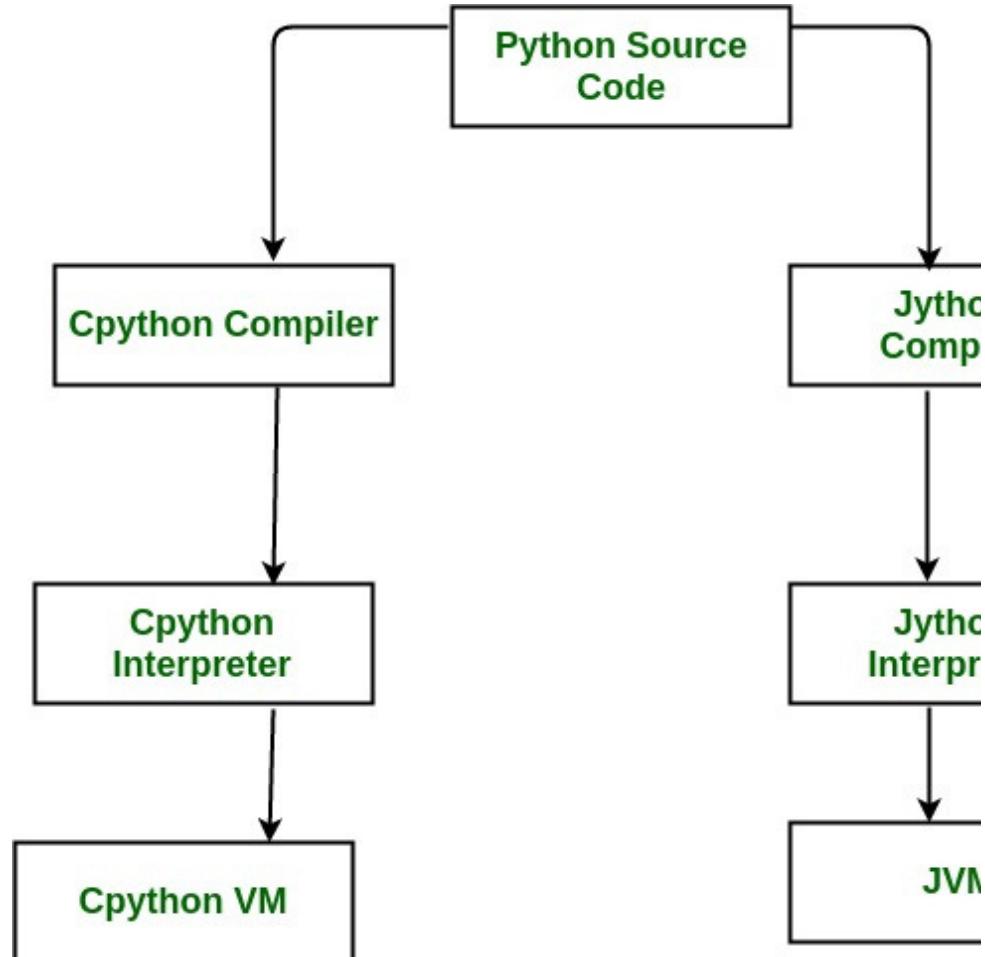
JPython

- Jython is an implementation of the Python programming language that can run on the Java platform.
- Jython programs use Java classes instead of Python modules



JPython

- Jython compiles into Java byte code, which can then be run by Java virtual machine.
- Jython enables the use of Java class library functions from the Python program.
- Jython is slow as compared to Cpython and lacks compatibility with CPython libraries.

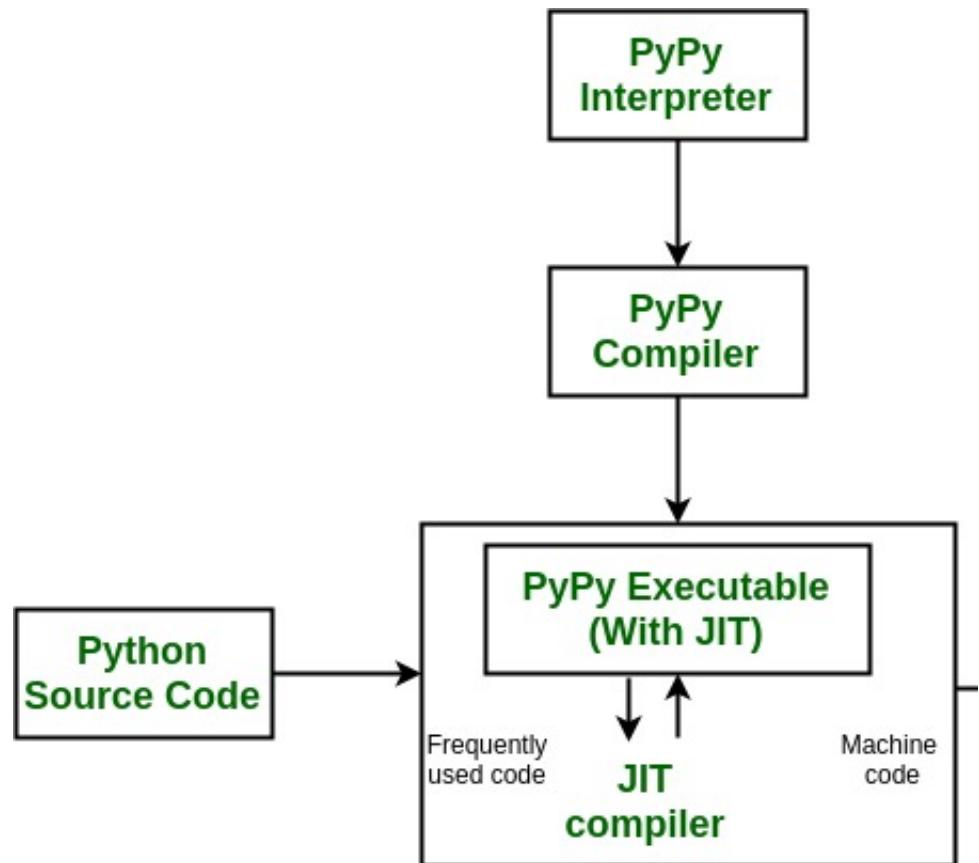


IronPython

- A Python implementation written in C# targeting Microsoft's .NET framework.
- Similar to Jython, it uses .Net Virtual Machine i.e Common Language Runtime.
- IronPython can use the .NET Framework and Python libraries, and other .NET languages can use Python code very efficiently.

PyPy

- “If you want your code to run faster, you should probably just use PyPy.” — Guido van Rossum (creator of Python)
- Python is dynamic programming language. Python is said to be slow as the default CPython implementation compiles the python source code in bytecode which is slow as compared to machine code(native code).
- Here PyPy comes in.



PyPy

- PyPy is an implementation of the Python programming language written in Python.
- The Interpreter is written in RPython (a subset of Python).
- PyPy uses (just-in-time compilation).
 - In simple terms JIT uses compilation methods to make interpreter system more efficient and fast.
 - So basically JIT makes it possible to compile the source code into native machine code which makes it very fast.

*

- You can execute any cell any kernel