UNIT 9 INTRODUCTION TO PYTHON

Structure

9.0 INTRODUCTION

Python programming is widely used in Artificial Intelligence, Machine Learning, Neural Networks and many other advanced fields of Computer Science. Ideally, It is designed for rapid prototyping of complex applications. Python has interfaces with various Operating system calls and libraries, which are extensible to C, C++ or Java. Many large companies like NASA, Google, YouTube, Bit Torrent, etc. uses the Python programming language for the execution of their valuable projects.

To build the carrier path the skill of programming can be a fun and profitable way, but before starting the learning of this skill, one should be clear about the choice of programming language. Before learning any programming language, one should figure out which language suits best to the learner. As in our case the comparison of C and Python programming languages may help the learners to analyze and generate a lot of opinions about their choice of programming language. In this unit, I have tried to compile a few of them to give you a clear picture.

Metrics	Python	C
Introduction	It is a high-level, general-purpose & interpreted programming language.	C is general-purpose procedural programming language.
Speed	Being Interpreted programming language its execution speed is slower then that of the compiled programming language (i.e. C).	Being compiled programming language its execution speed is faster then that of the interpreted programming language (i.e. Python).
Usage	Number of lines of code written in Python is quite less in comparison to C	Program syntax of C is quite complicated in comparison to Python.

Declaration of variables	Declaration of Variable type is not required, they are un-typed and a given variable can be stuck different types of values at different instances during the execution of any python program.	In C, declaration of variable type is must, and it is done at the time of its creation, and only values of that declared type must be assigned to the variables.
Error Debugging	Error debugging is simple, as it takes only one instruction at a time. Compilation and execution is performed simultaneously. Errors are instantly shown, and execution stops at that instruction only.	Being compiler dependent language, error debugging is difficult in C i.e. it takes the entire source code, compiles it and finally all errors are shown.
Function renaming mechanism	the same function can be used by two different names i.e. it supports the mechanism of function renaming	Function renaming mechanism is not supported by C i.e. the same function cannot be used by two different names.
Complexity	Syntax of Python programs is Quite simple, easy to read, write and learn	The syntax of a C program is comparatively harder than the syntax of Python.
Memory-management	It supports automatic garbage collection for memory management.	In C, the Programmer has to explicitly do the memory management.
Applications	Python is a General-Purpose programming language can be used for Microcontrollers also.	C is generally used for hardware related applications.
Built-in functions	Library of built-in functions is quite large in Python.	Library of built-in functions is quite limited in C
Implementing Data Structures	Gives ease of implementing data structures with built-in insert, append functions.	Explicit implementation of functions is requited for the implementation of datastructures
Pointers	Pointers functionality is not available in Python.	Pointers are frequently used in C.

We learned from the comparisonof C and Python, that Python is an high-level,general-purpose, interpreted programming language. It is dynamically typedandgarbage-collected, and supports multiple programming paradigms likestructured(particularly, procedural,) object-oriented, and functional programming, and due to its comprehensive standard library Python is often described as a "batteries included" language

In this course you are given exposure to both programming languages i.e. C and Python, based on your requirement you can choose your option to build your carrer in programming.

9.1 OBJECTIVES

After going through this unit you will be able to:

- Understand the need of Python as a programming language

- Describe the cross platform applications
- Understand the structure of python program
- Write and execute your first code in python

9.2 HISTORY OF PYTHON

Python was conceived in the late 1980s as a successor to the ABC language, it was created by Guido van Rossum in 1989 and its first release was in 1991. Later, Python 2.0, released in 2000, this version includes features like list comprehensions and a garbage collection system, which was capable of collecting reference cycles.Python 2 implemented the technical details of Python Enhancement Proposal(PEP), and thus simplified the code development complexities of the earlier versions. But due to some stability issues with Python 2.x versions, the development of Python 2.7 (last version in 2.x, released in 2010) will be discontinued in 2020. The developers were resolving the issues with Python 2 in parallel, and in 2008, Python 3.0 was released. It was a major revision of the language, this version i.e. Python 3 was mainly released to fix problems which exist in Python 2. As python 3 was not completely backward-compatible, thus much of the Python 2 code are not running unmodified on Python 3. To make the migration process easy in Python 3, some features of Python 3 have been backported to Python 2.x versions.

Thus, to migrate a project from python 3 to python 2.x, a lot of changes were required, which are not only related to the projects and its applications but also to the libraries that form the part of the Python environment.

You can freely download the latest version of Python from www.python.org, where various Python interpreters are available for many operating systems. A global community of programmers develops and maintains C-Python, which is an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and C-Python development.

So, what is C-python? It is the default or reference implementation of the Python programming language, and as the name suggests C-python is written in C language. Some of the implementations which are based on C-Python runtime core but with extended behavior or features in some aspects are Stack-less Python and Micro-Python; Stack-less Python, relates to C-Python with an emphasis on concurrency using tasklets and channels (used by ds-python for the Nintendo DS), and Micro-Python relates to working with microcontrollers.

C-python compiles the python source code into intermediate bytecode, which is executed by the C-python virtual machine. It is to be noted that Python is dynamic programming language, but it is said to be slow, because the default C-Python implementation compiles the python source code in bytecode which is slow as compared to machine code(native code).C-Python is distributed with a large standard library written in a mixture of C and Python, C-Python provides the highest level of compatibility with Python packages and C extension modules.

Further, Jython, IronPython and PyPy are also the implementation of Python as a programming language,they also providea good level of compatibility with Python packages. These implementations are based on Java, .NET and Python, respectively.We will be briefly discussing about these implementations, starting with Jython.

Jython is actually the Java platform based implementation of the Python programming language, so that you can run Python programs on the Java platform. The programs written in Jython uses Java classes and not the Python modules. The Jython compiler, compiles the Jython programs 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. In comparison to C-Python, Jython is slow, and it is not that much compatible the libraries of CPython. However, the compatibility in Iron Python and PyPy is quite good.

IronPython is the implementation of Python written in C# (Microsoft's .NET framework).As Jython used Java Virtual Machine (JVM), IronPython uses .Net Virtual Machine i.e. Common Language Runtime. The .NET Framework and Python libraries are used efficiently by the IronPython. Further, due to the availability of Just In Time (JIT) compiler and absence of Global Interpreter Lock, the IronPython performs better for the Python programs where use threads or multiple cores is highly required.

Guido van Rossum (creator of Python) said "If you want your code to run faster, you should probably just use PyPy." —PyPy is an implementation of the Python programming language written in Python itself. The PyPy Interpreter is written in RPython, which is a subset of Python. The PyPy interpreter uses JIT compiler to compile the source code into native machine code which makes it very fast and efficient.PyPy also comes with default with support for stackless mode, providing micro-threads for massive concurrency.

☞ Check Your Progress 1

1. Compare between C and Python programming languages

   ………………………………………………………………………………………

   ………………………………………………………………………………………

   ………………………………………………………………………………………

   ………………………………………………………………………………………

2. Discuss different variant of python viz. C-python, Jython, IronPython,PyPy

   ………………………………………………………………………………………

   ………………………………………………………………………………………

   ………………………………………………………………………………………

   ………………………………………………………………………………………

9.3 NEED OF PYTHON

We learned from past sections 9.0 and 9.1, that Python is free and simple to learn. The primary features of Python are, that it is an high-level, Interpreted and dynamically typed programming language, This encourages the rapid development of application prototypes, makes debugging of errors easy and identifying itself as the language to code with.

Now, you might be in the position to appreciate the simplicity and capability of Python as a programming Language, it is a general purpose language and have applicability in almost every domain of software development, be it web development or Scientific or Business or any other application, you can ever think off. It has wider coverage of applications, and it is complying with the current needs of software industry, Thus it assures a promising future to the learners.

The learners will understand the need of Python, once they understand the application areas of Python. You might not be knowing that various applications like YouTube, BitTorrent, DropBox etc. uses Python to achieve their functionality.The reason behind the choice of python, for these applications is the ability of python to be compatible for cross-platform operating systems (a cross-platform application may run on any Operating System, be it Microsoft Windows or Linux, or macOS or any other.), which simplifies the development of applications. Now, its time to start our discussion for the various types of applications that can be developed by using Python, below are few, there may be many more.

- Artificial Intelligence(AI) and Machine Learning(ML) are need of the times, as they yield the most promising careers for the learners. The field of AI and ML relates to the incorporation of intelligence in to machines by the process of self-learning from the data stored in to the system, various algorithms are available for this purpose. But, to bring AI and ML into action, Python is the first choice. Why?, because various well equipped libraries like Pandas, Scikit-Learn, NumPy etc. are available, to facilitate the engineers and scientists.Learn the algorithm, use the library and you have your solution to the problem. It is that simple.
- Data Science and Data Visualization: Data Scientists is just another promising career, if you know how to extract relevant information from the data source then world is yours. But, its not easy, various algorithms and techniques are required to work with. To simplify your work, again Python emerges to help you to study the data you have, perform operations and extract the required information. Libraries such as Pandas, NumPy help you in extracting information. Further, Matplotlib and Seaborn are the data visualization libraries, which are quite helpful in plotting and visualization of data. This is how Python helps you to become a Data Scientist.
- Mobile Applications: In general people think that Android and iOS are for mobile development, but what about using Python for mobile app

development ? Historically, Python didn't have a strong story when it came to writing mobile GUI applications, and that was quite questionable, because Python is projected as one solution for a variety of applications.But, with the passage of time, lot of development in Python has happened and now we are having Python as one of the option for Mobile App development. Now, we'll take a look at two frameworks i.e. Kivy and BeeWare, as options for mobile application development with Python.

- Embedded Applications: We learned from earlier sections of this unit that Python is based on C. Thus, it can be used to create software for embedded applications. The Micro-Python is a software implementation of Python 3, written in C. In fact, Micro-Python is a Python compiler that runs on the hardware of micro-controller, it includes modules to provide access to the low-level hardware. One of the renowned embedded application is Raspberry Pi which uses Python for its computing. It can be used as a computer or like a simple embedded board to perform high-level computations, you can use it for IoT or Mobile applications, thus may produce your smart gadgets.
- *Web Development:* Python is frequently used for web development, the reason behind is the availability of the full-stack frameworks for web development. There are many frameworks but Django, Flask and Pyramid are quite famous, among the frameworks for web development. They it includes common-backend logic and libraries to integrate protocols like HTTPS,FTP,SSL etc., and also supports processing of JSON,XML, E-Mail etc.
- Game Development : Gamming is one of the most upcoming software industry, user can build various interactive games by using various modules of Python, like Pygame and Pyglet, also the Python libraries like PySoy are available, to develop 3D games. Some of the renowned games like Civilization-IV, Disney's Toontown Online, Vega Strike etc. are developed by using Python.

Python has a variety of applications where it can be used. No matter what field you take up, Python is rewarding. So I hope you have understood the Python Applications and what sets Python apart from every other programming language.

9.4 PACKAGES FOR CROSS PLATFORM APPLICATION OF PYTHON

Python is the most popular programming language, and it marked its presence by contributing actively to every emerging field in computer science. It has vast set of libraries for almost all fields such as Machine Learning (Numpy, Pandas, Matplotlib), Artificial intelligence (Pytorch, TensorFlow), and Game development (Pygame,Pyglet), and many more.

After going through the section 9.2 of this unit, you understood the meaning of Cross-platform applications and you are now aware of the potential of Python as a cross-platform programming language, you also came to know that the development of different type of applications require different types of packages, libraries, modules, frameworks etc. Now you might be confused that what is the difference between these terms, are they same or different. Let's Clear your confusion first and then we will briefly discuss about the functionality of different packages, used for the development of various Python applications.

Python: Framework Vs Library Vs Package Vs Module

Framework is a collection of libraries. This is the architecture of the program.

Library is a collection of packages. (*We may understand that, Python library or framework is a pre-written program that is ready to use on common coding tasks.)*

Package is a collection of modules. It must contains an __init__.py file as a flag so that the python interpreter processes it as such. The __init__.py could be an empty file without causing issues. A package, in essence, is like a directory holding subpackages and modules. While we can create our own packages, we can also use one from the Python Package Index (PyPI) to use for our projects.To import a package, we type *import Game.Sound.load* Or We can also import it giving it an alias:*import Game.Sound.load as load game*.

Module is a file which contains various Python functions and global variables. It is simply just .py extension file which has python executable code. We put similar code together in one module. This helps us modularize our code, and make it much easier to deal with. And not only that, a module grants us reusability. With a module, we don't need to write the same code again for a new project that we take up.

let's see how Module and Package differ:

1) A module is a file containing Python code. A package, however, is like a directory that holds sub-packages and modules.

2) A package must hold the file __init__.py. This does not apply to modules.

3) To import everything from a module, we use the wildcard *. But this does not work with packages.

We will learn more about them, as we proceed in this course, don't worry. To start with we will discuss about various Frameworks and Libraries first, you will be learning about their usage and also the usage of methods and packages, later.

TensorFlow: TensorFlow is an end-to-end python machine learning library for performing high-end numerical computations. TensorFlow can handle deep neural networks for image recognition, handwritten digit classification, recurrent neural networks, NLP (Natural Language Processing)

Keras : is a leading open-source Python library used for development of neural networks and machine learning projects. It simplifies the process of designing and development of neural networks for the beginners of machine learning. Keras also deals with convolution neural networks(CNN) and Recurrent Neural Networks (RNN), highly required in the field of Deep Learning. It includes algorithms for normalization, optimizion, and activation layers. Instead of being an end-to-end Python machine learning library, Keras acts as a user-friendly, extensible interface that enhances modularity & total expressiveness.

Theano : is a library for scientific computation, it allows you to define, optimize as well as evaluate the complex mathematical expressions, which deals with multidimensional arrays. The repetitive computation of a tricky mathematical expression is the basis of several ML and AI applications. Theano quickly performs the data-intensive calculation, the rate of execution is almost hundred times faster than when executing on our CPU alone. It aims to reduce the development and execution time of ML apps, particularity in deep learning algorithms. Only one drawback of Theano in front of TensorFlow is that its syntax is quite hard for the beginners.

Scikit-learn : it is a prominent open-source library for machine learning through Python, it includes a wide range of algorithms like DBSCAN, gradient boosting, random forests, vector machines, and k-means etc., which are generally used to implement various concepts of Machine Learning i.e. Classification, Clustering, Regression etc. It can interoperate with numeric and scientific libraries of Python like NumPy and SciPy. Scikit-learn supports both supervised as well as unsupervised ML.

PyTorch : it is a production-ready library for machine-learning, supported with excellent examples, applications and use cases, also supported by a strong community.It supports GPU and CPU computations, thus provides performance optimization and scalablity in research as well as production. The two high-end features of the PyTorch are Deep neural networks and Tensor computation, which are boosted because of the machine learning compiler “Glow”, specially designed to improve the performance of deep learning frameworks.

NumPy : also known as Numerical Python, a library to perform scientific computations. Almost all Python machine-learning tools like Matplotlib, SciPy, Scikit-learn, etc rely on this library to a reasonable extent. It comes with functions for dealing with complex mathematical operations like linear algebra, Fourier transformation, random number and features that work with matrices and n-arrays in Python. It is widely used in handling sound waves, images, and other binary functions.

SciPy : It is library, that works for all type od scientific programming projects. Its main functionality is built upon NumPy, and it includes modules for linear algebra, Integration, optimization, and statistics too. SciPy is supported with extensive documentation, which makes its working really easy.

Pandas : it is an open-source library of python that offers a wide range of tools for data analysis & manipulation, with Pandas, the data from a broad range of sources like CSV, SQL databases, JSON files, and Excel, can be read. The data analysis & manipulation is a pre-requisite for most of the machine learning projects, where a significant amount of time is spent to analyse the trends and patterns hidden in the datasets. Using Pandas one can manage complex data operations with just one or two commands, it comes with several inbuilt methods for data handling, and it also serves as the starting point to create more focused and powerful data tools.

Matplotlib : this library is specifically meant for Data Science, it helps to generate data visualization like 2D diagrams and graphs like histograms, scatter plots, and even graphs for non-cartesian coordinates.It is equipped with the object oriented API to embed plots into the applications, directly. It is because of this library, the Python is able to compete with scientific tools like MATLAB or Mathematica. It is quite compatible with the popular plotting libraries, but developers are required to write more code than usual, while using this library for generating advanced visualizations.

Seaborn : it is an unparalleled visualization library, based on Matplotlib's foundations. Seaborn offers high-level dataset based interface to make amazing statistical graphics. With Seaborn, it is simple to create certain types of plots like time series, heat maps, and violin plots. The functionalities of Seaborn go beyond Pandas and matplotlib with the features to perform statistical estimation at the time of combining data across observations, plotting and visualizing the suitability of statistical models to strengthen dataset patterns.

Scrapy : It is the most widely used library of python, for the purpose of data sciences, specifically data mining. It is used to build crawling programs i.e spider bots, which are used to retrieve structured data (example URLs or Contact Information) from the web. Developers use it for gathering data from APIs.

Pygame : is a python programming language library for making multimedia applications like video games, animations etc. It includes libraries related to computer graphics and sound, which are designed to be used with the Python programming language. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

Django : it a web framework build using Python,it encourages rapid development of web applications.Developers can focus on writing app only, It's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes on code reusability and pluggability of components, thus it supports less code, low coupling, and rapid development of websites.

MicroPython : is a software implementation of Python 3, written in C. Infact, MicroPython is a Python compiler that runs on the hardware of micro-controller, it includes modules to provide access to the low-level hardware. One of the renowned embedded application is Raspberry Pi which uses Python for its computing. It can be used as a computer or like a simple embedded board to perform high-level computations, you can use it for IoT or Mobile applications, thus may produce your smart gadgets.

BeeWare – is a Python GUI and mobile development framework, to develop native Python Mobile Apps. "BeeWare" project, provides a set of tools and an abstraction layer, which can be used to write native-looking mobile and desktop applications using Python.

Kivy – is an opensource Python Framework used as a Cross-platform Python GUI for Mobile App development, it allows you to write pure-Python graphical applications that run on both i.e. the main desktop platforms (Windows, Linux, macOS and Raspberry Pi) and also on on iOS & Android.Both Kivy and BeeWare are worth considering. So far as maturity goes, Kivy seems to be the more mature platform right now

☞ **Check Your Progress 2**

1. Compare Framework, Library, Package and Modules in python
2. Identify the prominent tools of python for following:
   a. Artificial Intelligence application
   b. Mobile application development
   c. Embedded programming
   d. Web development
   e. Game development
3. Discuss the utility of following:
   a. Tensor Flow
   b. Keras
   c. Theano
   d. Pytorch
   e. NumPy
   f. SciPy
   g. Pandas
   h. Matplotlib
   i. Kivy
   j. Django
   k. Pygame

9.6 PROGRAM STRUCTURE IN PYTHON

This section relates to the discussion over the programming structure of Python programming, i.e. the way you divide a program in your source files and mix parts from various libraries or modules or packages in to a single program, i.e. directly or indirectly, a single program includes multiple filesfrom different modules. Modules are in fact the library tools, which are implemented to make a collection of top-level files i.e. the high level files may use tools, defined in modules, which may use files, defined in other modules. Coming to our point, in python a file takes a module to get access to the tools it defines, and also to the tools defined in other modules included in programme. In python high level file has important path of control of your program, where from you can start your application. Just refer to the figure given below

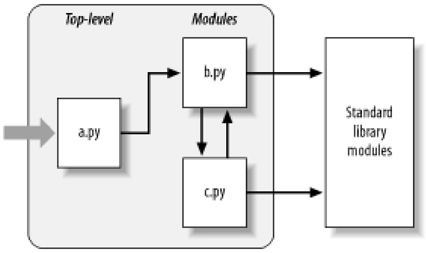


Figure : Structure Python Program

To understand the structure of Python program, say there exist three files a.py,b.py and c.py. The file model a.py is chosen for high level file . it is known as a simple text file of statements. Files b.py and c.py are modules. They are also text files of statements but they are generally not started directly, but they are invoked by a.py i.e high level file.

This is the general discussion over the structure of Python program, although Python includes all the components as they are in any other language such as **data types, conditional statements, looping constructs,** functions, file handling, Classes, Exception handling, Libraries, Modules, packages etc. We will discuss a few of them over here and the rest will be discussed in the subsequent units in this course

9.5 GETTING STARTED WITH PYTHON

Now, we are having a bit of clarity about the Python as a programming language, from the past sections we learned about the various libraries and frameworks of Python. Now, we need to work on the IDEs (Integrated Development Environments) of Python, there are many IDEs like Jupyter Notebook, Spyder, VS Code, R Programming etc., all are collectively available in Anaconda (Anaconda isa free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing,predictiveanalytics, etc.)), or you may also go for the cloud versions Like Google Colab Notebook, where you need not to have high configuration hardwares, only internet is required and through your gmail account you can work on Python using Jupyter notebook.

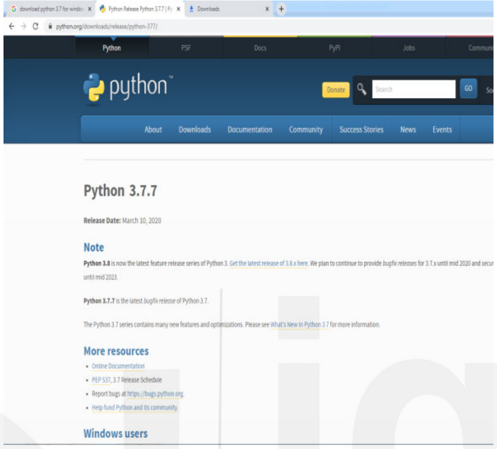
We will discuss in brief, some of the ways to work with Python, you may choose any or try all and other options too.

1) Just browse for https://www.python.org and perform following steps :

a. Download the latest version of Python for the operating system installed on your computer, as in my case its windows 64 bit, so I downloaded python-3.7.7 (python-3.7.7-amd64.exe) from https://www.python.org/downloads/release/python-377/

b. Now Run this exe file and install the Python, just click next and go ahead, till the setup installation is finished

c. Finally you are having an interface for Python programming



Python 3.7.7

Release Date: March 10, 2020

Note

Python 3.8 is now the latest feature release series of Python 3. Get the latest release of 3.8.x here. We plan to continue to provide bugfix releases for 3.7.x until mid 2020 and secu until mid 2023.

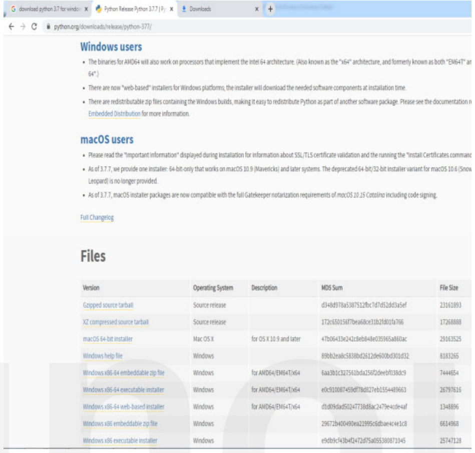
Python 3.7.7 is the latest bugfix release of Python 3.7.

The Python 3.7 series contains many new features and optimizations. Please see What's New in Python 3.7 for more information.

More resources

- Online Documentation
- PEP 537, 3.7 Release Schedule
- Report bugs at https://bugs.python.org.
- Help fund Python and its community.

Windows users


download python 3.7 for windo
Python Release Python 3.7.7 | Py
Downloads
python.org/downloads/release/python-377/

Windows users

- The binaries for AMD64 will also work on processors that implement the Intel 64 architecture. (Also known as the "x64" architecture, and formerly known as both "EM64T" ar 64".)
- There are now "web-based" installers for Windows platforms; the installer will download the needed software components at installation time.
- There are redistributable zip files containing the Windows builds, making it easy to redistribute Python as part of another software package. Please see the documentation r Embedded Distribution for more information.

macOS users

- Please read the "Important Information" displayed during installation for information about SSL/TLS certificate validation and the running the "Install Certificates comman
- As of 3.7.7, we provide one installer: 64-bit-only that works on macOS 10.9 (Mavericks) and later systems. The deprecated 64-bit/32-bit installer variant for macOS 10.6 (Snow Leopard) is no longer provided.
- As of 3.7.7, macOS installer packages are now compatible with the full Gatekeeper notarization requirements of *macOS 10.15 Catalina* including code signing.

Full Changelog

Files

Version	Operating System	Description	MD5 Sum	File Size
Gzipped source tarball	Source release		d348d978a5387512fbc7d7d52dd3a5ef	23161893
XZ compressed source tarball	Source release		172c650156f7bea68ce31b2fd01fa766	17268888
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	47b06433e242c8eb848e035965a860ac	29163525
Windows help file	Windows		89bb2ea8c5838bd2612de600bd301d32	8183265
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	6aa3b1c327561bda256f2deebf038dc9	7444654
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	e0c910087459df78d827eb1554489663	26797616
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	d1d09dad50247738d8ac2479e4cde4af	1348896
Windows x86 embeddable zip file	Windows		29672b400490ea21995c6dbae4c4e1c8	6614968
Windows x86 executable installer	Windows		e9db9cf43b4f2472d75a055380871045	25747128

There is a variety of IDEs for python, Like Jupyter Notebook, Spyder, VS Code etc, and all are available at Anaconda (a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.)). To start with Anaconda Just perform following steps:

2) Just browse https://www.anaconda.com/ and perform following steps:

a. Click the Download button on the webpage of https://www.anaconda.com/

b. the distribution section https://www.anaconda.com/distribution/ will open,

c. click the download option given on this pagehttps://www.anaconda.com/distribution/

d. https://www.anaconda.com/distribution/#download-section will open, the option of 64bit and 32bit graphic installer for Python 3.x (currently 3.7) and 2.x (currently 2.7) are given under Anaconda 2020.02 for windows installer.

e. It is recommended to download 64bit version of Python 3.x (currently 3.7),

f. Anaconda3-2020.02-Windows-x86_64.exe will be downloaded.

g. Now, just run the setup of this Anaconda3-2020.02-Windows-x86_64.exe , and click next-next, till the installation is completed.

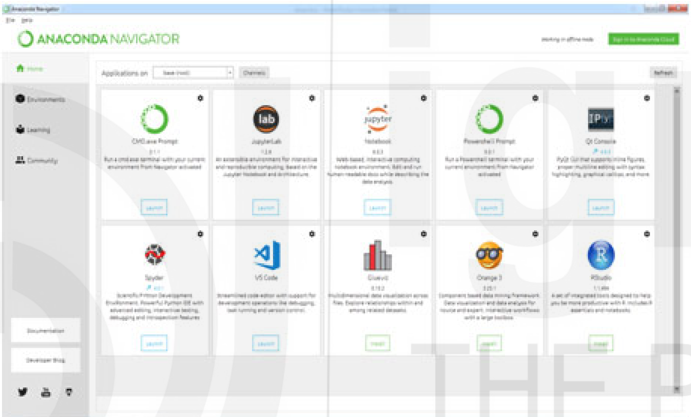
h. Finally, you will find Anaconda Navigators shortcut on your desktop, click on it and you can start working with any IDE be it

Jupyter Notebook, Spyder, VS Code etc., even you can work with R-Programming.

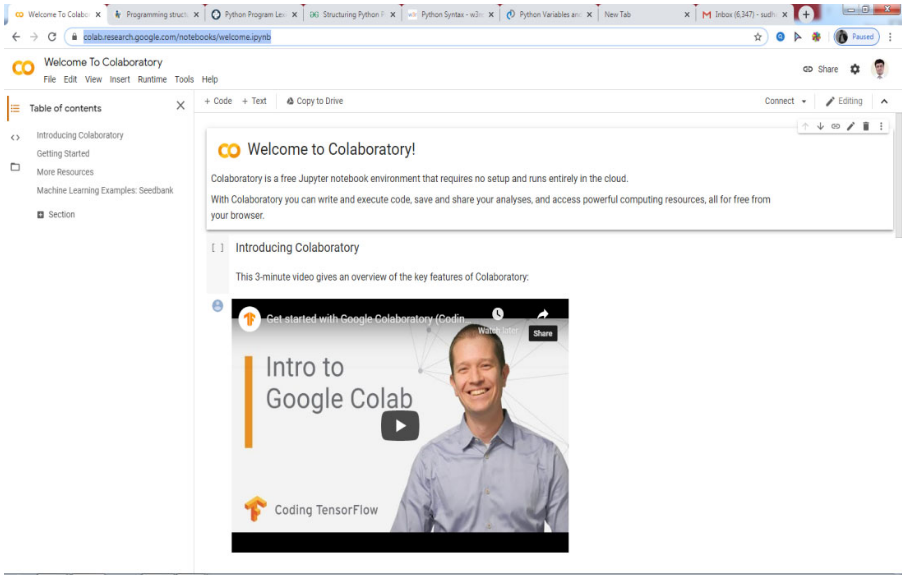
Important: Before working with IDEs you need to understand how to work with them and which ine is suitable, following are observations, currently:

a) When you start Jupyter Notebook on windows (by clicking on the jupyter section, given in the anaconda Navigator), a browser will open in internet explorer, many a times Jupyter Notebook won't work here, then just copy the URL from the Internet Explorer and paste it in the Google Chrome, you will find that Jupyter Notebook starts working, other details regarding the writing, saving, execution of program, will be discussed later.

b) Program exeution in Jupyter is line by line and in VS Code and Spyder it goes sideways, even errors can also be seen sideways. Any ways all are good to work with Python.



3) Many a times the learners may not be equipped with the systems having latest hardware configuarations, as desired for the installation of Python, or their might be compatibility issues with operating syatem or may be due to any reason you are not able to install and start your work with Python. Under such circumstances the solution is Google Colab Notebook (https://colab.research.google.com/notebooks/welcome.ipynb), use this and just login with your gmail account and start your work on Jupyter Notebook, as simple as that.



Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

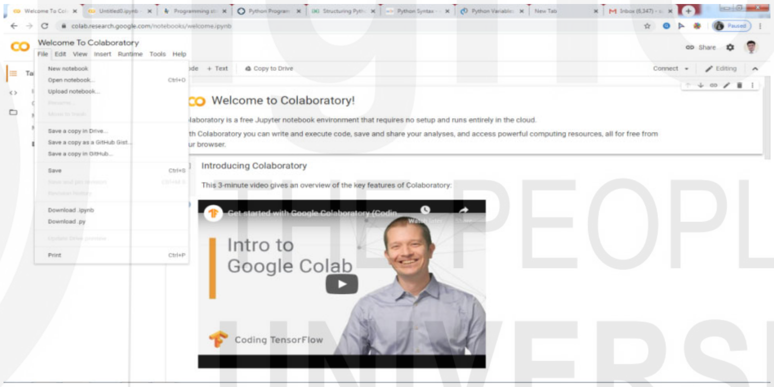
With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

Introducing Colaboratory

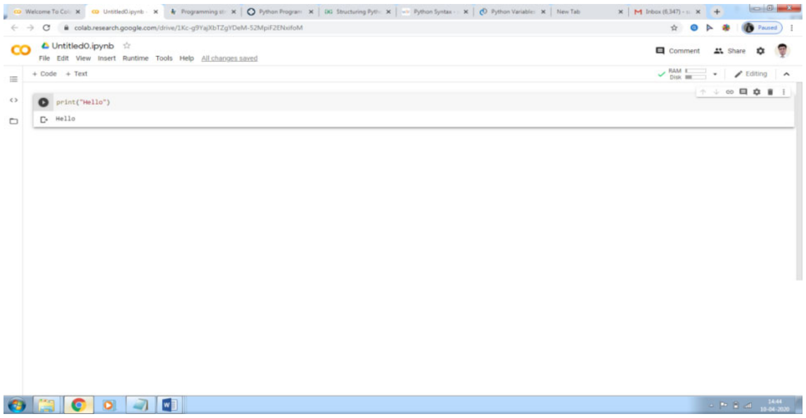
This 3-minute video gives an overview of the key features of Colaboratory:

9.7 RUNNING THE FIRST PROGRAM

Just click file option and select new workbook, and new Jupyter notebook will open in Google Colab, now you may start your work



As here, I wrote my first program (print("hello")) to say "hello" to all of you , and executed it by simply pressing the play button, you may see just before the print command I wrote, and the output comes just beneath the statement print("hello").


colab.research.google.com/drive/1Kc-g9YajXbTZgYDeM-52MpiF2ENxifoM
Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
Editing

```
print("Hello")
```

```
Hello
```

9.8 SUMMARY

The conceptual fundamentals of python programming language were discussed in this unit, after going through this unit the learner will be equipped not only with the with the historical understanding of python, but also with the various application areas and the tools relevant to explore the concerned application area.

SOLUTION TO CHECK YOUR PROGRESS

Check your progress 1

1) refer to section 9.0
2) refer to section 9.2

Check your progress 2

1) refer to section 9.4
2) refer to section 9.3
3) refer to section 9.4