PRIME INTUIT
Finishing School

Introduction to programing

General – Session – 1

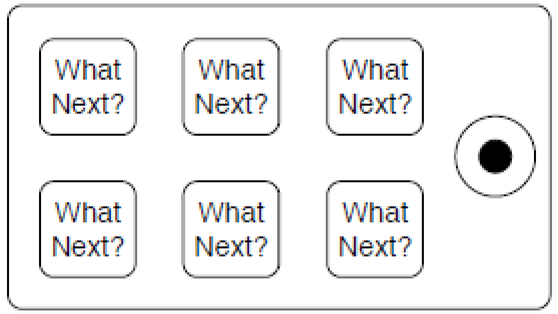
PRIME INTUIT
Finishing School

Agenda

- What is Programing
- Computer Hardware architecture
- Understanding programing
- Why Python
- How is Python different from other languages
- Little History
- Installing Pycharm, Learning what is an IDE

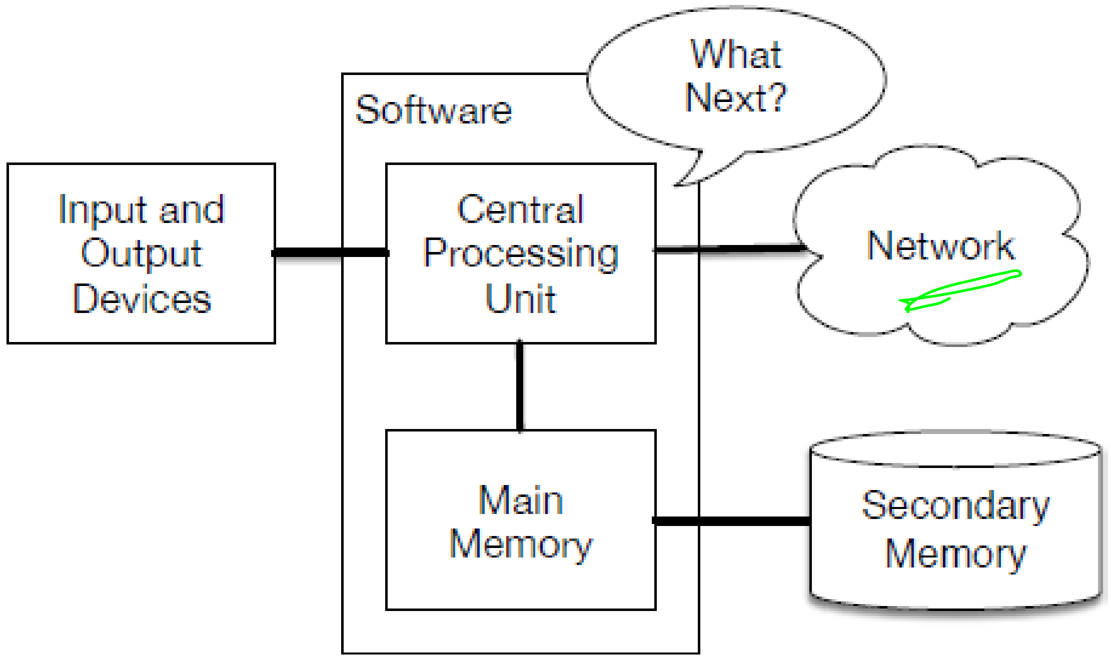
What is programing ?

We can think of today’s computers as our “personal assistants” who can take care of many things on our behalf. The hardware in our current-day computers is essentially built to continuously ask us the question, “What would you like me to do next?”



Our computers are fast and have vast amounts of memory and could be very helpful to us if we only knew the language to speak to explain to the computer what we would like it to “do next”.

Computer hardware architecture



PRIME INTUIT
Finishing School

Understanding Programing

Two skills to be a programmer:

Programming language (Python)

You need to know the vocabulary and the grammar. You need to be able to spell the words in this new language properly and know how to construct well-formed "sentences" in this new language.

Story Telling :

Combine words and sentences to convey an idea to reader

For a programmer program is the "story" and the problem you are trying to solve is the "idea".

Story writing is improved by doing some writing and getting some feedback.

PRIME INTUIT
Finishing School

Why Python?

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects

- **Easy to Use**
- **Very efficient**
- **Has multiple Libraries and Frameworks**
- **Community and Corporate Support**
- **Portable and Extensible**

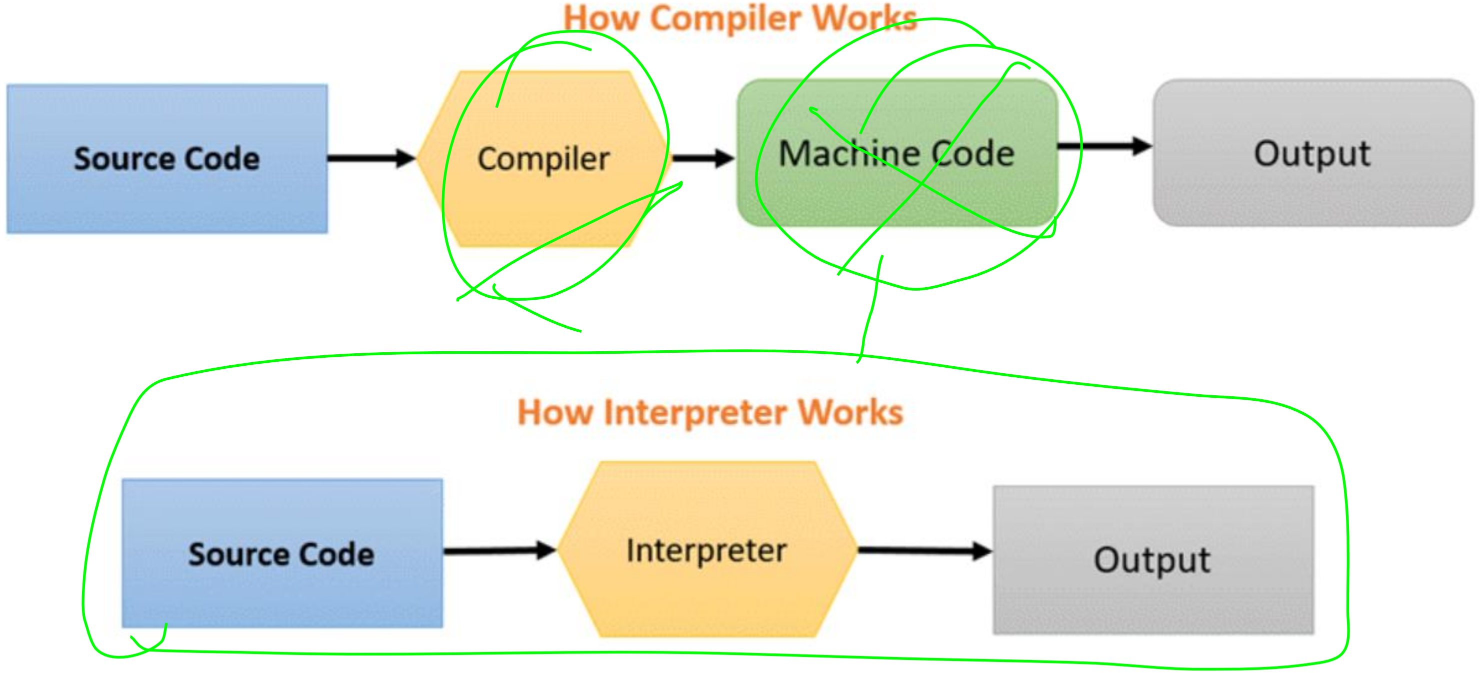
PRIME INTUIT
Finishing School

What is an interpreter?

interpreter reads the source code of the program as written by the programmer, parses the source code, and interprets the instructions on the fly. Python is an interpreter and when we are running Python interactively, we can type a line of Python (a sentence) and Python processes it immediately and is ready for us to type another line of Python.

PRIME INTUIT
Finishing School

An interpreter is a computer program, which coverts each high-level program statement into the machine code. This includes source code, pre-compiled code, and scripts. Both compiler and interpreters do the same job which is converting higher level programming language to machine code. However, a compiler will convert the code into machine code (create an exe) before program run. Interpreters convert code into machine code when the program is run.

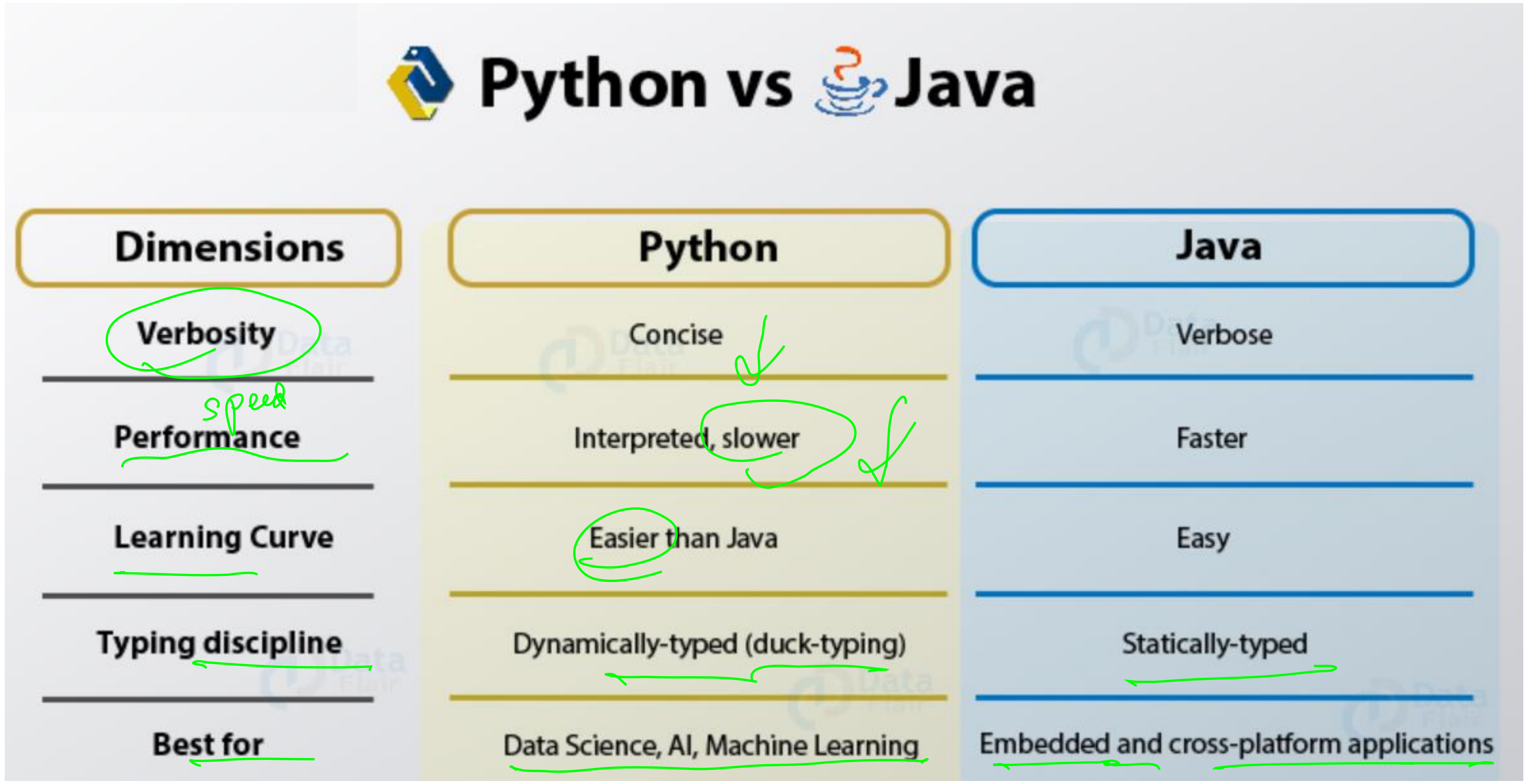


PRIME INTUIT
Finishing School

Compiler	Interpreter
• A compiler takes the entire program in one go.	• An interpreter takes a single line of code at a time.
• The compiler generates an intermediate machine code.	• The interpreter never produces any intermediate machine code.
• The compiler is best suited for the production environment.	• An interpreter is best suited for a software development environment.
• The compiler is used by programming languages such as C, C ++, C #, Scala, Java, etc.	• An interpreter is used by programming languages such as Python, PHP, Perl, Ruby, etc.

PRIME INTUIT
Finishing School

How is Python different from other programing languages?



Python vs Java

Dimensions	Python	Java
Verbosity	Concise ✓	Verbose
Performance (speed)	Interpreted, slower ✓	Faster
Learning Curve	Easier than Java	Easy
Typing discipline	Dynamically-typed (duck-typing)	Statically-typed
Best for	Data Science, AI, Machine Learning	Embedded and cross-platform applications

Little History

Guido van Rossum

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language.

First release was in 1991 as Python 0.9.0.

Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting.

Python 3.0 was released in 2008 and was a major revision of the language that is ~~not completely backward-compatible~~ and much Python 2 code does not run unmodified on Python 3.

Python 2 was discontinued with version 2.7.18 in 2020.

Installing Python & Pycharm

www.python.org/downloads/

Difference between Python 2 and Python 3

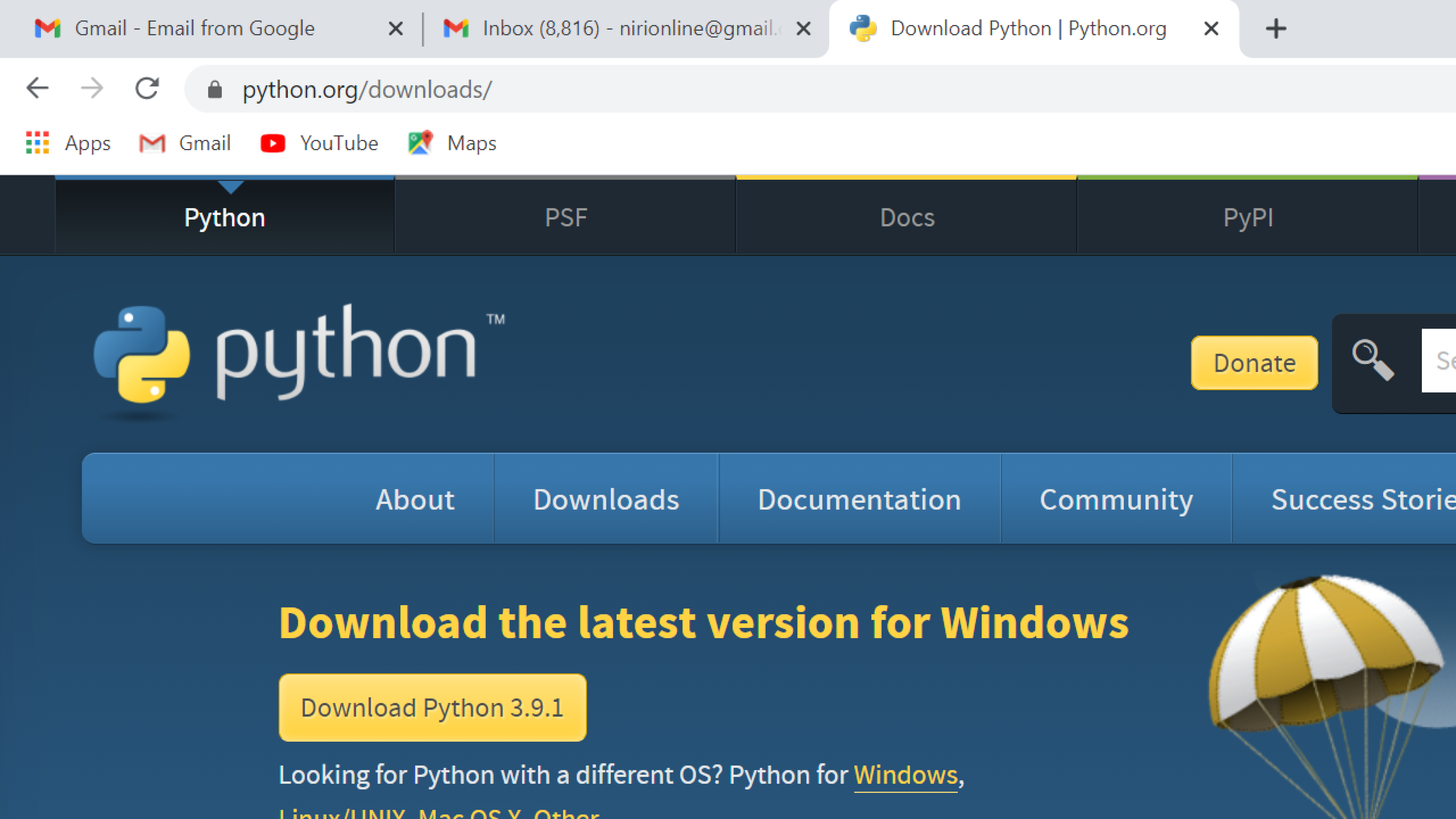
www.jetbrains.com/pycharm/

Wifi details

Prime Intuit 4G

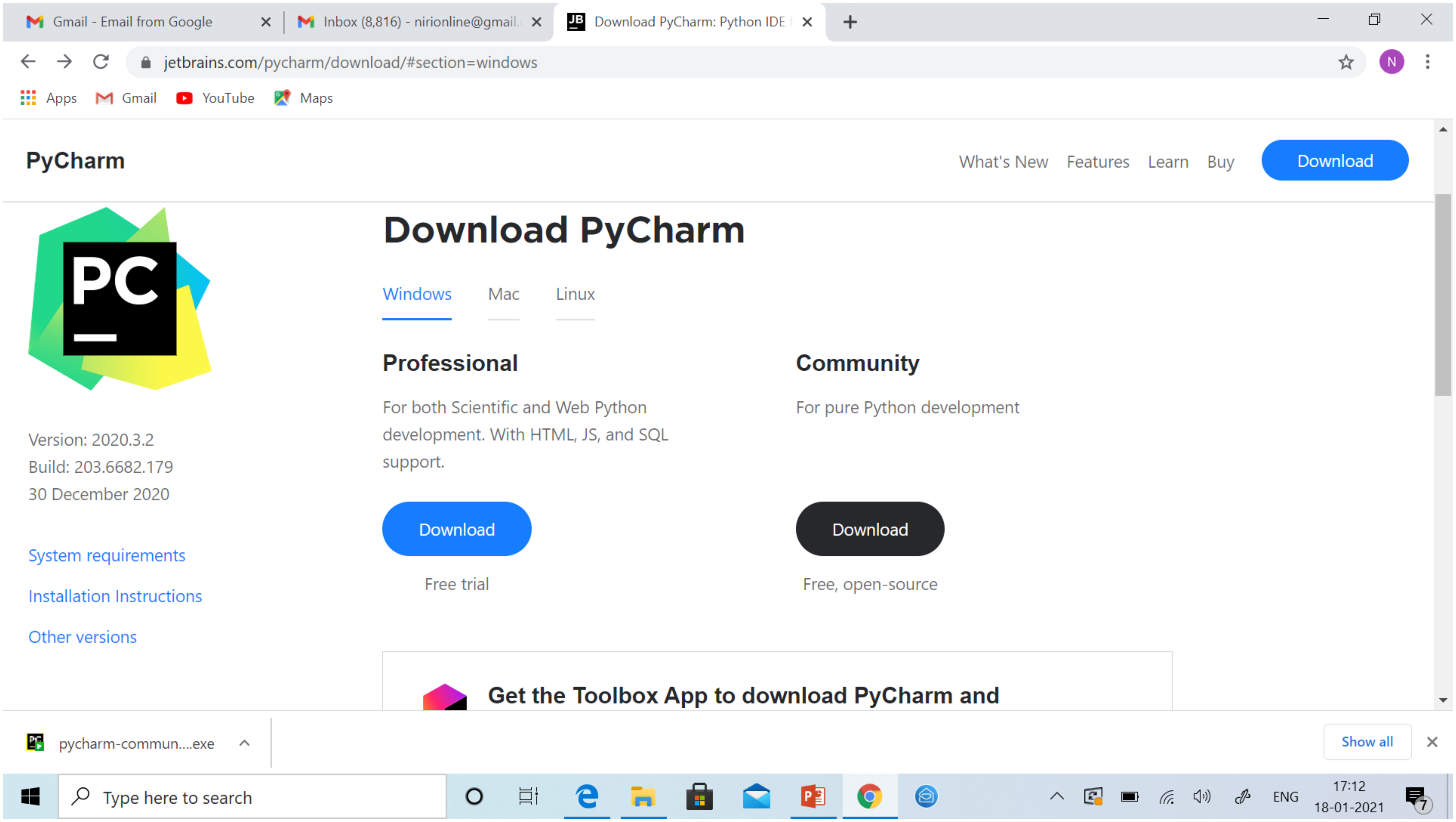
Prime Intuit 5G

Password: kbmnk99@



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows,



Gmail - Email from Google
Inbox (8,816) - nirionline@gmail.
Download PyCharm: Python IDE f
jetbrains.com/pycharm/download/#section=windows
Apps
Gmail
YouTube
Maps

PyCharm

Download PyCharm

Version: 2020.3.2
Build: 203.6682.179
30 December 2020

System requirements
Installation Instructions
Other versions

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Free trial

Community

For pure Python development

Free, open-source

Get the Toolbox App to download PyCharm and

PRIME INTUIT
Finishing School

Your First Python Program

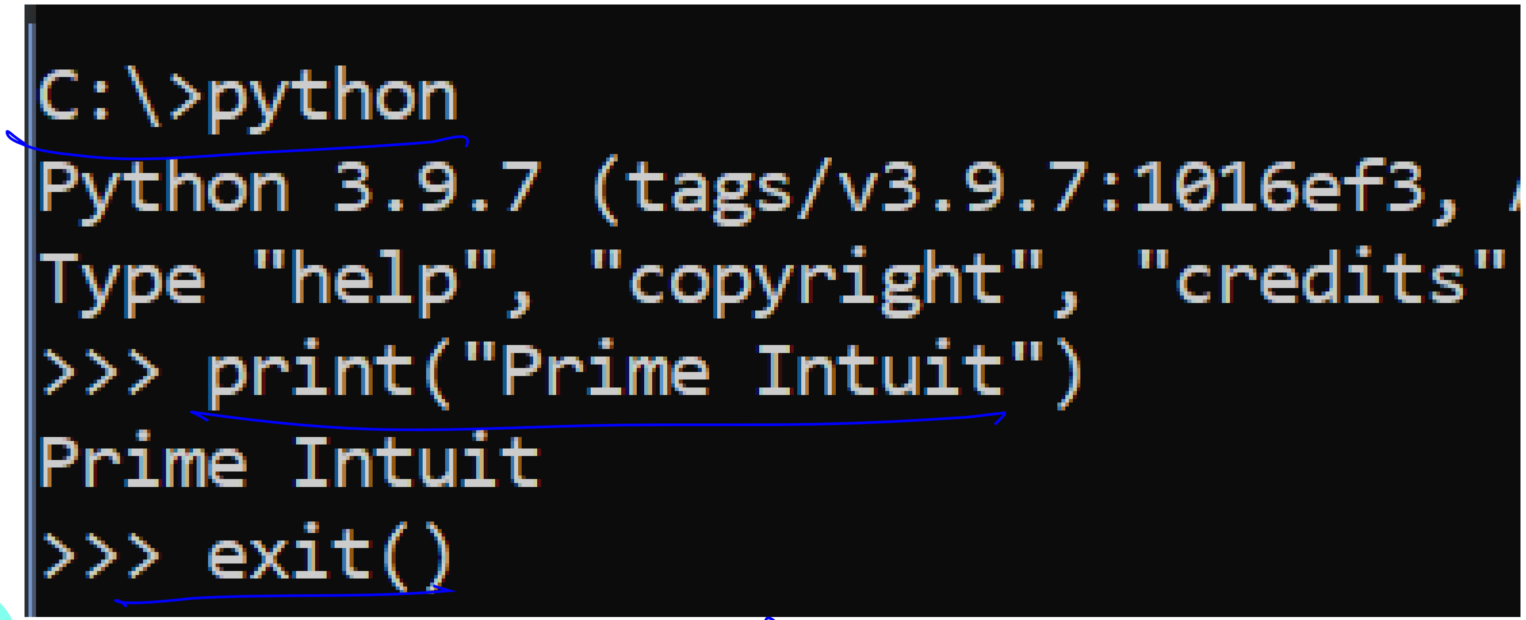
Create New Project: PrimeIntuit

Interpreter: Python V3

New Python file

1 Print(“Hello World”)

2 Menu / Run



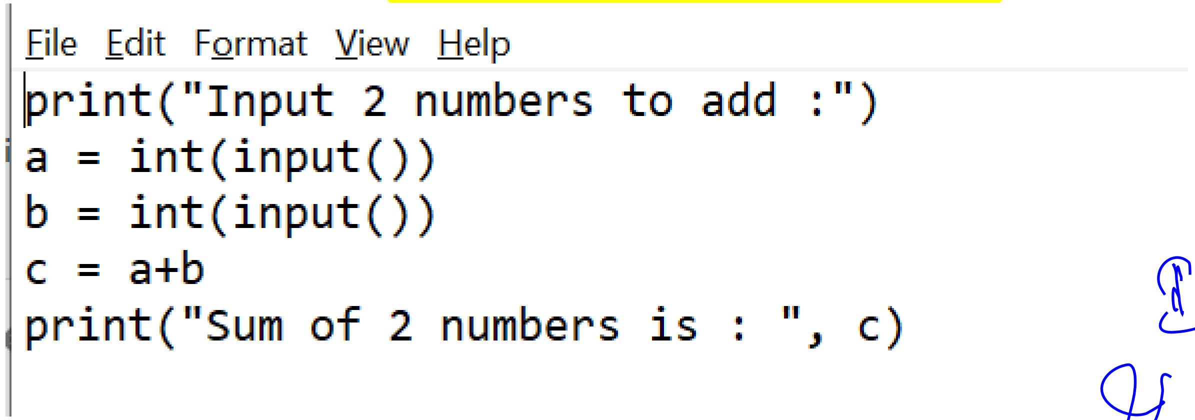
```
C:\>python
Python 3.9.7 (tags/v3.9.7:1016ef3,
Type "help", "copyright", "credits"
>>> print("Prime Intuit")
Prime Intuit
>>> exit()
```

Interactive Mode

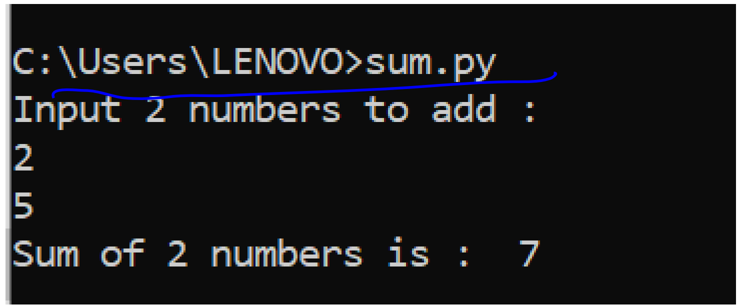
enter command , get result

PRIME INTUIT
Finishing School

Script Mode



```
print("Input 2 numbers to add :")
a = int(input())
b = int(input())
c = a+b
print("Sum of 2 numbers is : ", c)
```



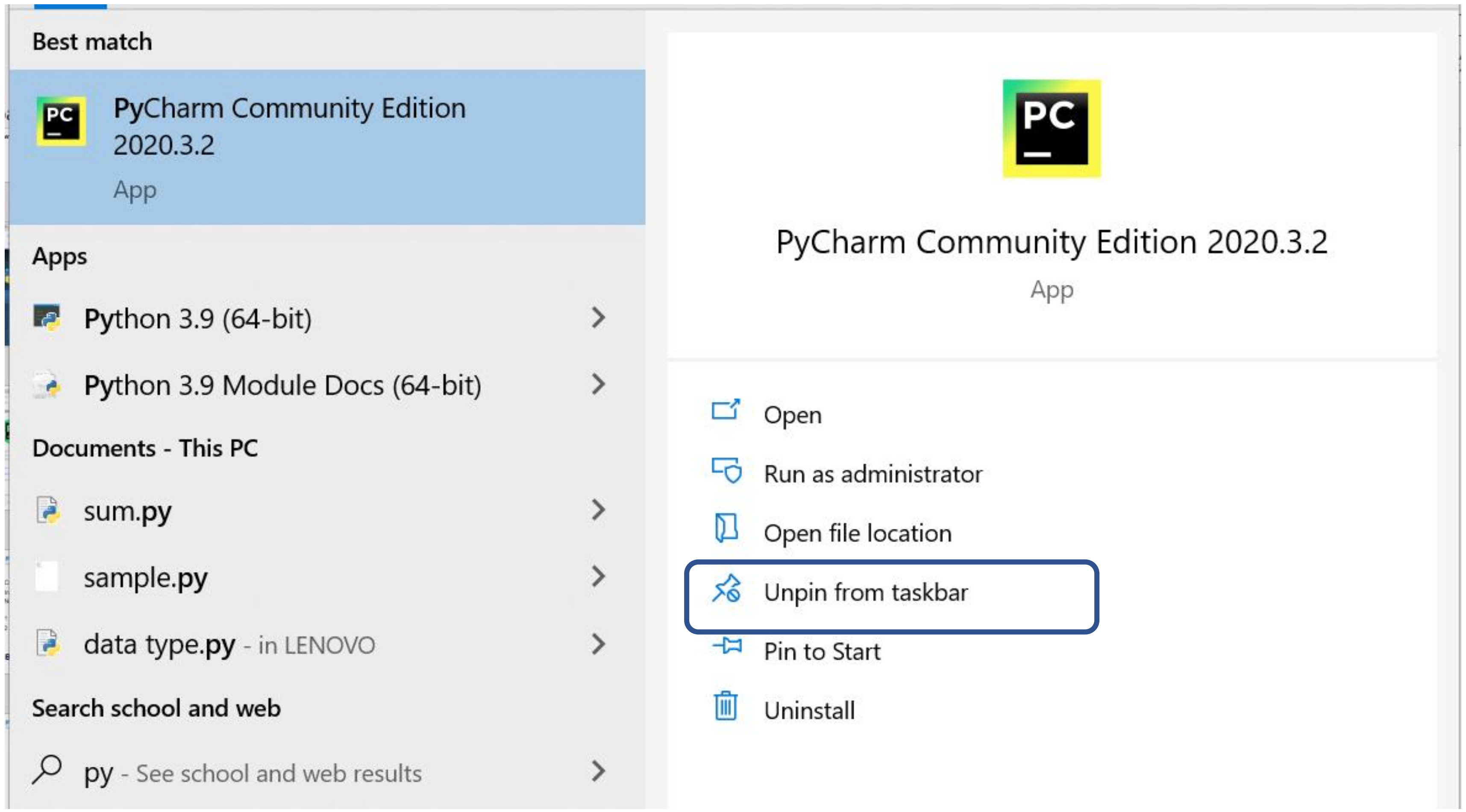
```
C:\Users\LENOVO>sum.py
Input 2 numbers to add :
2
5
Sum of 2 numbers is :  7
```

result command

Save a program in IDE & can run result in command prompt

PRIME INTUIT
Finishing School

Bringing PyCharm to task bar


Best match

PyCharm Community Edition 2020.3.2
App

Apps

Python 3.9 (64-bit)

Python 3.9 Module Docs (64-bit)

Documents - This PC

sum.py

sample.py

data type.py - in LENOVO

Search school and web

py - See school and web results

PyCharm Community Edition 2020.3.2
App

Open
Run as administrator
Open file location
Unpin from taskbar
Pin to Start
Uninstall

What are IDE s and Code Editors

vs

different

What Are IDEs and Code Editors?

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

- An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
- Build, execution, and debugging tools
- Some form of source control

Most IDEs support many different programming languages and contain many more features.

Code Editors

In contrast, a dedicated code editor can be as simple as a text editor with syntax highlighting and code formatting capabilities. Most good code editors can execute code and control a debugger. The very best ones interact with source control systems as well. Compared to an IDE, a good dedicated code editor is usually smaller and quicker, but often less feature rich.

Requirements for a Good Python Coding Environment

Below core set of features that makes coding easier:

- **Save and reload code files**

If an IDE or editor won’t let you save your work and reopen everything later, in the same state it was in when you left, it’s not much of an IDE.

- **Run code from within the environment**
  Similarly, if you have to drop out of the editor to run your Python code, then it’s not much more than a simple text editor.
- **Debugging support**
  Being able to step through your code as it runs is a core feature of all IDEs and most good code editors.
- **Syntax highlighting**
  Being able to quickly spot keywords, variables, and symbols in your code makes reading and understanding code much easier.
- **Automatic code formatting**
  Any editor or IDE worth it’s salt will recognize the colon at the end of a while or for statement, and know the next line should be indented.
- There are lots of other features you might want, like source code control, an extension model, build and test tools, language help, and so on. But the above list is what I’d see as “core features” that a good editing environment should support.

General Editors and IDEs with Python Support

Eclipse + PyDev

Category: IDE
Website: www.eclipse.org
Python tools: PyDev, www.pydev.org

If you've spent any amount of time in the open-source community, you've heard about Eclipse. Available for Linux, Windows, and OS X at, Eclipse is the de-facto open-source IDE for Java development. It has a rich marketplace of extensions and add-ons, which makes Eclipse useful for a wide range of development activities.

One such extension is PyDev, which enables Python debugging, code completion, and an interactive Python console.

- **Pros:** If you've already got Eclipse installed, adding PyDev will be quicker and easier. PyDev is very accessible for the experienced Eclipse developer.
- **Cons:** If you're just starting out with Python, or with software development in general, Eclipse can be a lot to handle. Remember when I said IDEs are larger and require more knowledge to use properly? Eclipse is all that and a bag of (micro)chips.

General Editors and IDEs with Python Support

Sublime Text

Category: Code Editor
Website: http://www.sublimetext.com

Written by a Google engineer with a dream for a better text editor, Sublime Text is an extremely popular code editor. Supported on all platforms, Sublime Text has built-in support for Python code editing and a rich set of extensions (called packages) that extend the syntax and editing features.

Installing additional Python packages can be tricky: all Sublime Text packages are written in Python itself, and installing community packages often requires you to execute Python scripts directly in Sublime Text.

- **Pros:** Sublime Text has a great following in the community. As a code editor, alone, Sublime Text is fast, small, and well supported.
- **Cons:** Sublime Text is not free, although you can use the evaluation version for an indefinite period of time. Installing extensions can be tricky, and there's no direct support for executing or debugging code from within the editor.

General Editors and IDEs with Python Support

Atom

Category: Code Editor
Website: https://atom.io/

Available on all platforms, Atom is billed as the "hackable text editor for the 21st Century." With a sleek interface, file system browser, and marketplace for extensions, open-source Atom is built using Electron, a framework for creating desktop applications using JavaScript, HTML, and CSS. Python language support is provided by an extension that can be installed when Atom is running.

- **Pros:** It has broad support on all platforms, thanks to Electron. Atom is small, so it downloads and loads fast.
- **Cons:** ~~Build and debugging support~~ aren't ~~built-in~~ but are community provided add-ons. Because Atom is built on Electron, it's always running in a JavaScript process and not as a native application.

General Editors and IDEs with Python Support

Visual Studio

Category: IDE
Website: https://www.visualstudio.com/vs/
Python tools: Python Tools for Visual Studio, aka PTVS

Built by Microsoft, Visual Studio is a full-featured IDE, in many ways comparable to Eclipse. Built for Windows and Mac OS only, VS comes in both free (Community) and paid (Professional and Enterprise) versions. Visual Studio enables development for a variety of platforms and comes with its own marketplace for extensions.

Python Tools for Visual Studio (aka PTVS) enables Python coding in Visual Studio, as well as Intellisense for Python, debugging, and other tools.

- **Pros:** If you already have Visual Studio installed for other development activities, adding PTVS is quicker and easier.
- **Cons:** Visual Studio is a big download for just Python. Plus, if you're on Linux, you're out of luck: there's no Visual Studio install for that platform.

Python-Specific Editors and IDEs

- PyCharm
- **Category:** IDE
  Website: https://www.jetbrains.com/pycharm/
- One of the best (and only) full-featured, dedicated IDEs for Python is PyCharm. Available in both paid (Professional) and free open-source (Community) editions, PyCharm installs quickly and easily on Windows, Mac OS X, and Linux platforms.
- Out of the box, PyCharm supports Python development directly. You can just open a new file and start writing code. You can run and debug Python directly inside PyCharm, and it has support for source control and projects.

Python-Specific Editors and IDEs

copied

- Anaconda
- ~~**Category:** IDE~~
  ~~**Website:** https://www.jetbrains.com/pycharm/~~
- ~~One of the best (and only) full-featured, dedicated IDEs for Python is PyCharm. Available in both paid (Professional) and free open-source (Community) editions, PyCharm installs quickly and easily on Windows, Mac OS X, and Linux platforms.~~
- ~~Out of the box, PyCharm supports Python development directly. You can just open a new file and start writing code. You can run and debug Python directly inside PyCharm, and it has support for source control and projects.~~

Python-Specific Editors and IDEs

- Colab
- **Category:** IDE
  Website: https://www.jetbrains.com/pycharm/
- One of the best (and only) full-featured, dedicated IDEs for Python is PyCharm. Available in both paid (Professional) and free open-source (Community) editions, PyCharm installs quickly and easily on Windows, Mac OS X, and Linux platforms.
- Out of the box, PyCharm supports Python development directly. You can just open a new file and start writing code. You can run and debug Python directly inside PyCharm, and it has support for source control and projects.

repeated

Which Python IDE is Right for You?

Only you can decide that, but here are some basic recommendations:

New Python developers should try solutions with as few customizations as possible. The less gets in the way, the better.

If you use text editors for other tasks (like web pages or documentation), look for code editor solutions.

If you're already developing other software, you may find it easier to add Python capabilities to your existing toolset.

PRIME INTUIT
Finishing School

Thank You !