August 08, 2022
Instructor: Engr. Atta Ullah Khan

PCAP – Certified Associate in Python Programming

PCAP – Certified Associate in Python Programming

Module 1
Introduction to Python and Computer Programming

Module 2
Data Types, Variables, Basic Input-Output Operations, and Basic Operators

Module 3
Boolean Values, Conditional Execution, Loops, Lists and List Processing, Logical and Bitwise Operations

Module 4
Functions, Tuples, Dictionaries, And Data Processing

Module 5
Object Oriented Programming Concepts

Module 6
NumPy, Pandas, Matplotlib, Linear Regression, Logistic Regression.

PCAP – Module 1

What is Python?

Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming.

Who created Python?

One of the amazing features of Python is the fact that it is actually one person's work. Usually, new programming languages are developed and published by large companies employing lots of professionals, and due to copyright rules, it is very hard to name any of the people involved in the project. Python is an exception.

There are not many languages whose authors are known by name. Python was created by **Guido van Rossum**, born in 1956 in Haarlem, the Netherlands. Of course, Guido van Rossum did not develop and evolve all the Python components himself.

Where do you use Python?

Generally, Python is a great choice for:

- **Web Application Development** (a lot of web development frameworks Django and Flask)
- **Data Science** (Aa a highly demanded skill, Data science is now reaching the top. Python libraries like Pandas, NumPy, SciPy)
- **Artificial intelligence** (Most interesting practical uses for Python is in Artificial Intelligence and Machine Learning. Python collection of libraries for Machine Learning Application. These include Pandas, Keras, TensorFlow, NumPy and many more)
- **Game Development**
- **Internet of Things**

- **Web Scraping**
- **Desktop GUI**
- **Enterprise Application**
- **Image Recognition and Text Processing**

Python the Future Technology:

4th most used programming language in 2021, 44.1% of developers worked with it.

Top companies using Python

- Google
- Instagram
- Facebook
- Quora
- Netflix
- Dropbox
- Reddit

Source: https://www.botreetechnologies.com/blog/top-10-python-use-cases-and-applications/

Get PCAP Certified

Becoming PCAP certified will help the certificate holder stand out from other candidates and get their foot in the door.
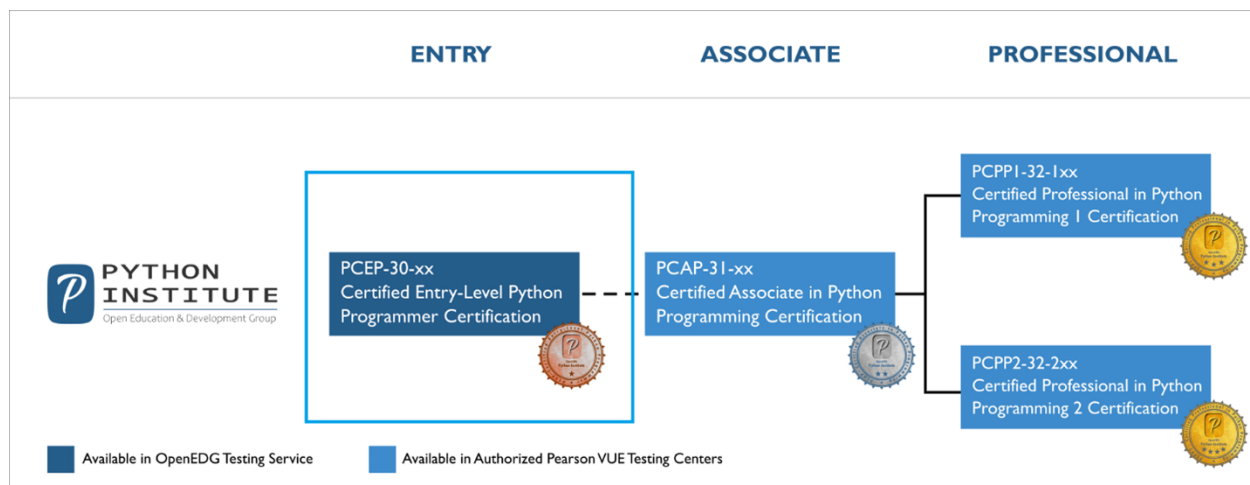
OpenEDG Python Institute Certification Roadmap

Natural Languages vs. Programming Languages

A language is a means (and a tool) for expressing and recording thoughts. There are many languages all around us. Some of them require neither speaking nor writing, such as body language; it's possible to express your deepest feelings very precisely without saying a word.

Another language you use each day is your mother tongue, which you use to manifest your will and to ponder reality. Computers have their own language, too, called **machine** language, which is very rudimentary.

Note: machine languages are developed by humans.

Interpreter and Compiler

- Python is a high-level language intended to be relatively straightforward for humans to read and write and for computers to read and process
- Other high-level languages include Java, C++, PHP, Ruby, Basic, Perl, JavaScript, and many more
- The actual hardware inside the Central Processing Unit (CPU) does not understand any of these high-level languages
- The CPU understands a language we call **Machine Language**

001010001110100100101010000001111

1110011000001110101001010101101101
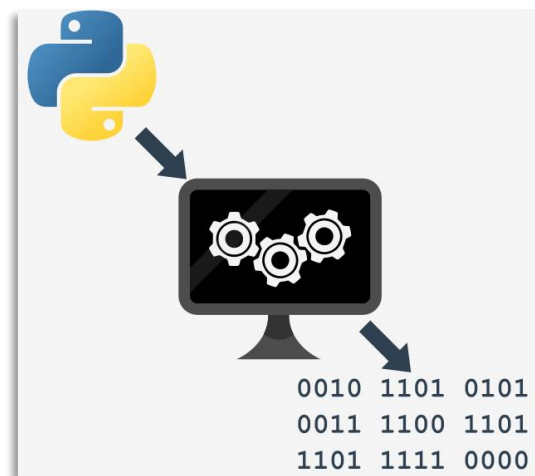
- Machine Language seems quite simple on the surface, given that there are only zeros and ones, but its syntax is even more complex than Python
- We build various translators to allow programmers to write in high-level languages like Python or JavaScript and these translators convert the programs to machine language for actual execution by the CPU
- These programming language translators fall into two general categories:
  - **Interpreters**
  - **Compilers**
- An interpreter reads the source code of the program as written by the programmer, parses the source code, and **interprets the instructions on the fly**
- A compiler needs to be handed the entire program in a file, and then it runs a process to translate the high-level source code into machine language
- Then the compiler puts the resulting machine language into a file for later execution

How to get Python and how to get to use it

There are several ways to get your own copy of Python 3, depending on the operating system you use.

Linux users most probably have Python already installed - this is the most likely scenario, as Python's infrastructure is intensively used by many Linux OS components.

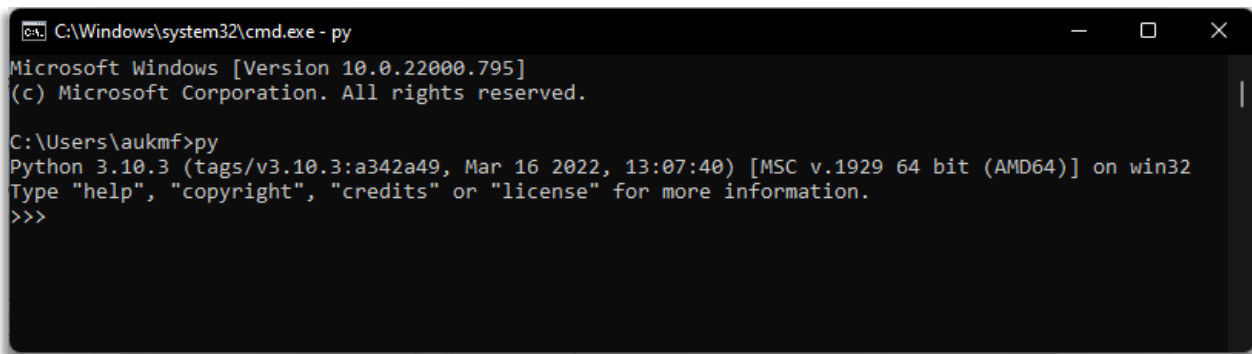**If you're a Linux user, open the terminal/console, and type:**
`python3`
at the shell prompt, press *Enter* and wait.

If you see something like this:
```
Python 3.4.5 (default, Jan 12 2017, 02:28:40)
[GCC 4.2.1 Compatible Clang 3.7.1 (tags/RELEASE_371/final)] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

If you're a Window user, open the CMD and type:

`py` and then press enter. In this case you see something like this:

Setting Python Environment for Coding

Downloading and Installing Python

This is the site to download Python -> https://www.python.org/downloads/

Because the browser tells the site you've entered the OS you use, the only step you have to take is to click the appropriate Python version you want.

In our case, select Python 3. The site always offers you the latest version of it.

If you're a **Windows user**, start the downloaded .exe file and follow all the steps.

Leave the default settings the installer suggests for now, with one exception - look at the checkbox named **Add Python 3.x to PATH** and check it.
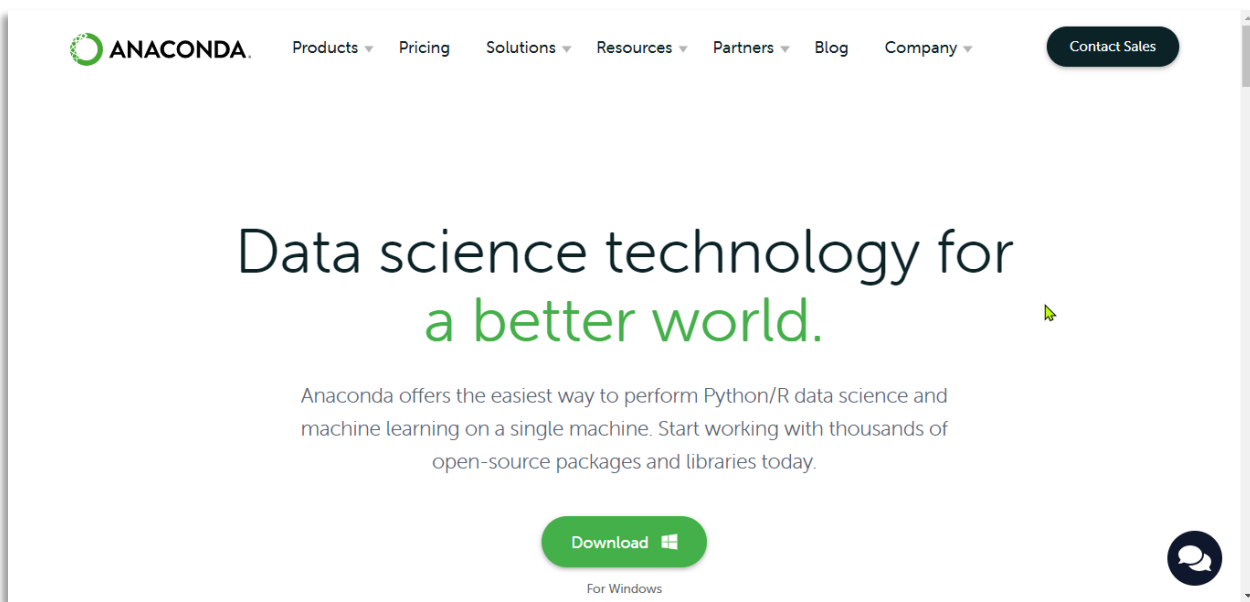
This will make things easier.
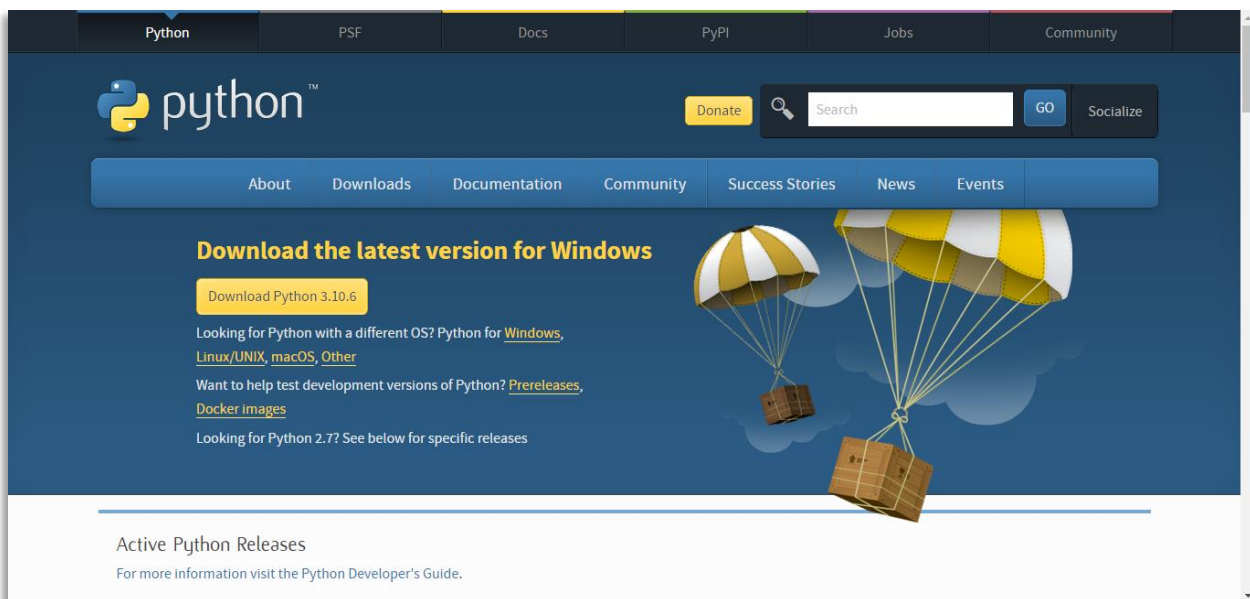
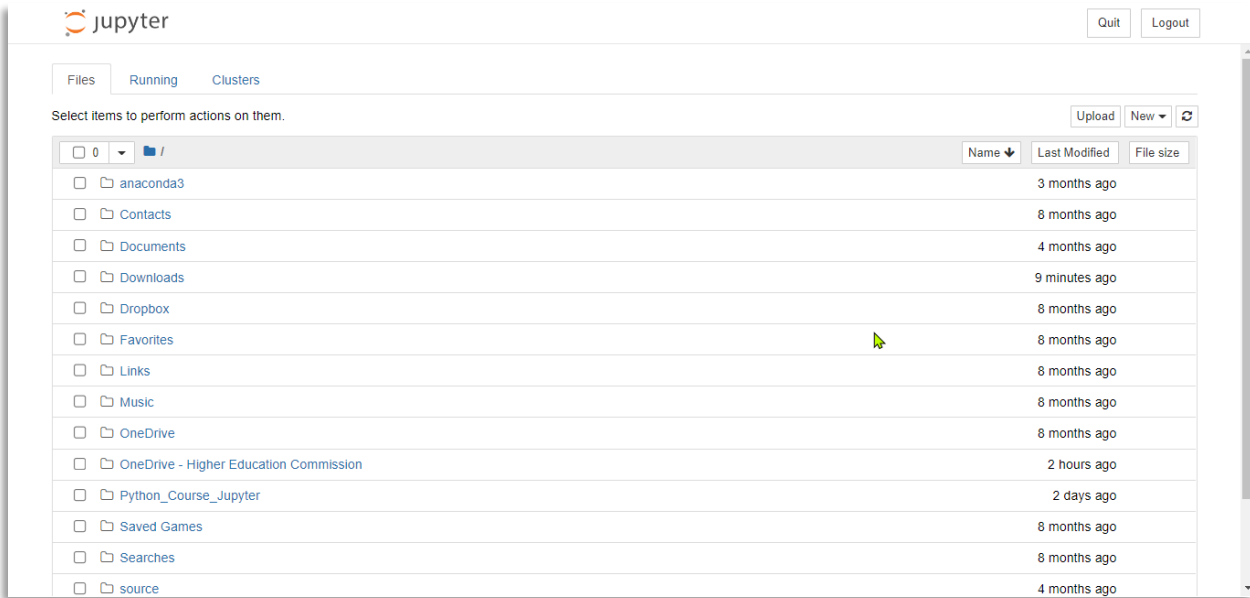# Downloading and Installing PyCharm

This is the site to Install PyCharm -> https://www.jetbrains.com/pycharm/

Installing Anaconda

This is the site to Install Anaconda -> https://anaconda.com/

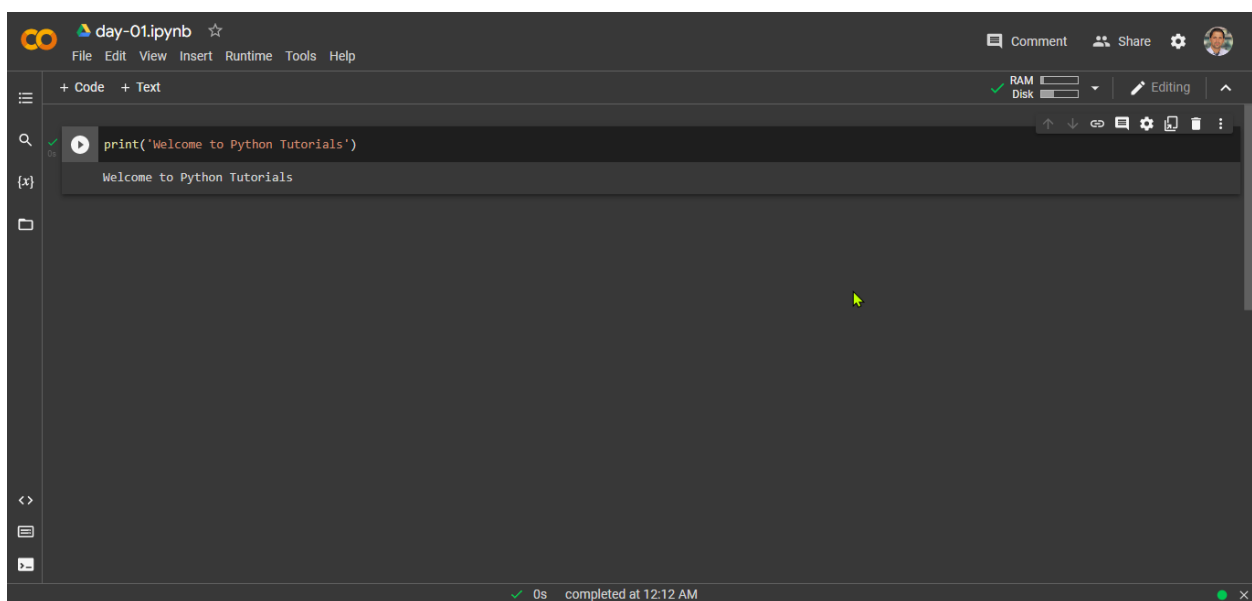# Intro to Jupyter Notebook

Jupyter Notebook is a browser-based interpreter that allows us to interactively work with Python.

Intro to Google Colab

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more.
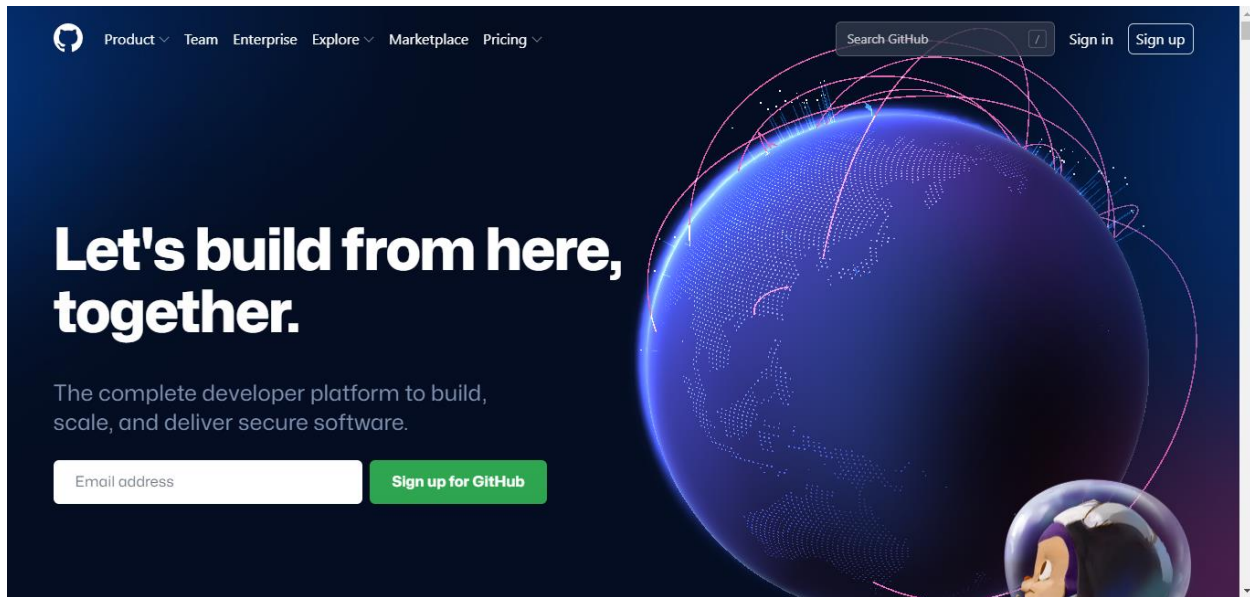
GitHub

GitHub is a code hosting platform for **version control and collaboration.** It lets you and others work together on projects from anywhere.

Version Control: Version control, also known as source control, is the practice of tracking and managing changes to software code.

This is the site to Install GitHub -> https://github.com/

How to write and run your very first program

- It is now time to write and run your first Python 3 program. It will be very simple, for now.
- The first step is to create a new source file and fill it with code. Click *File* in the IDLE menu and choose *New file.*
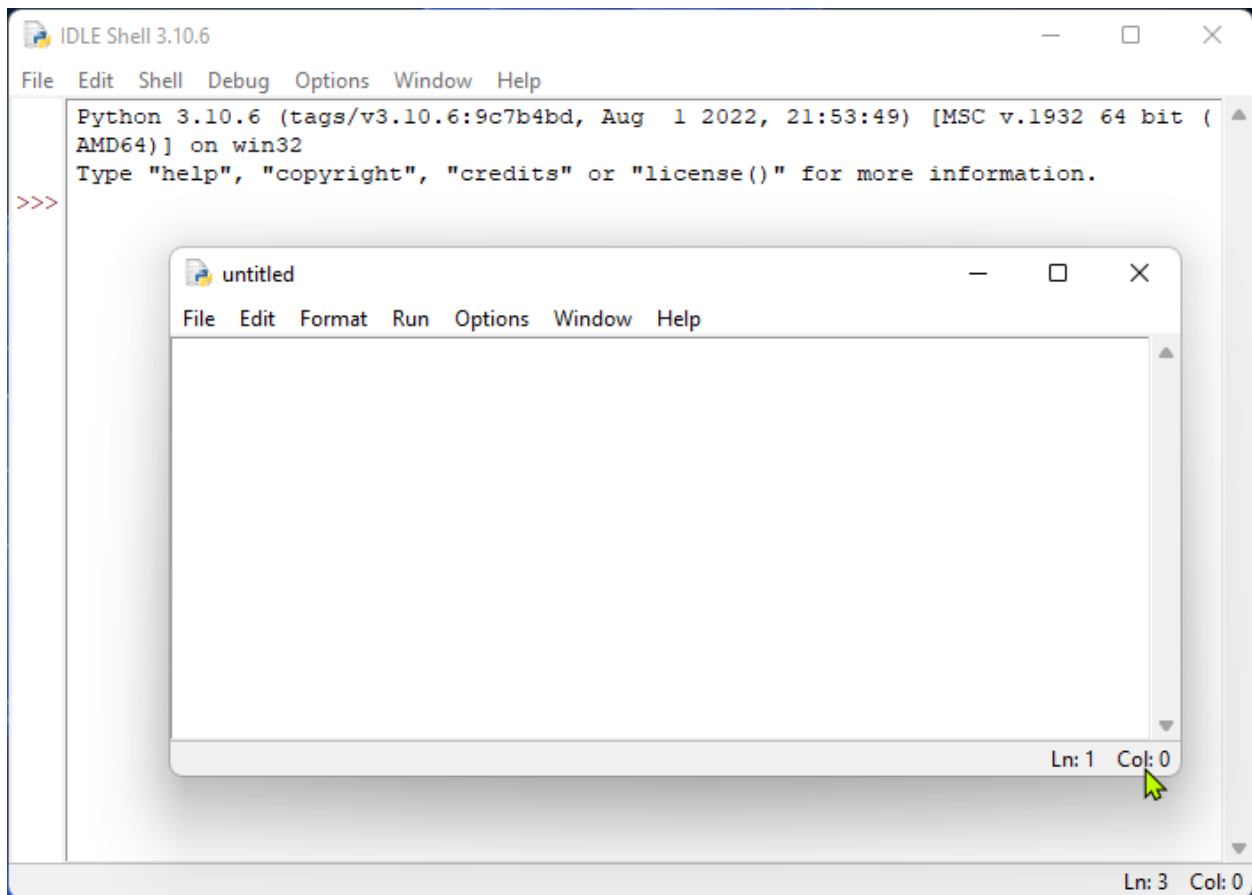
- As you can see, IDLE opens a new window for you. You can use it to write and amend your code.
- This is the **editor window**. Its only purpose is to be a workplace in which your source code is treated. Do not confuse the editor window with the shell window. They perform different functions.
- The editor window is currently untitled, but it's good practice to start work by naming the source file.
- Click File (in the new window), then click Save as..., select a folder for the new file (the desktop is a good place for your first programming attempts) and chose a name for the new file.
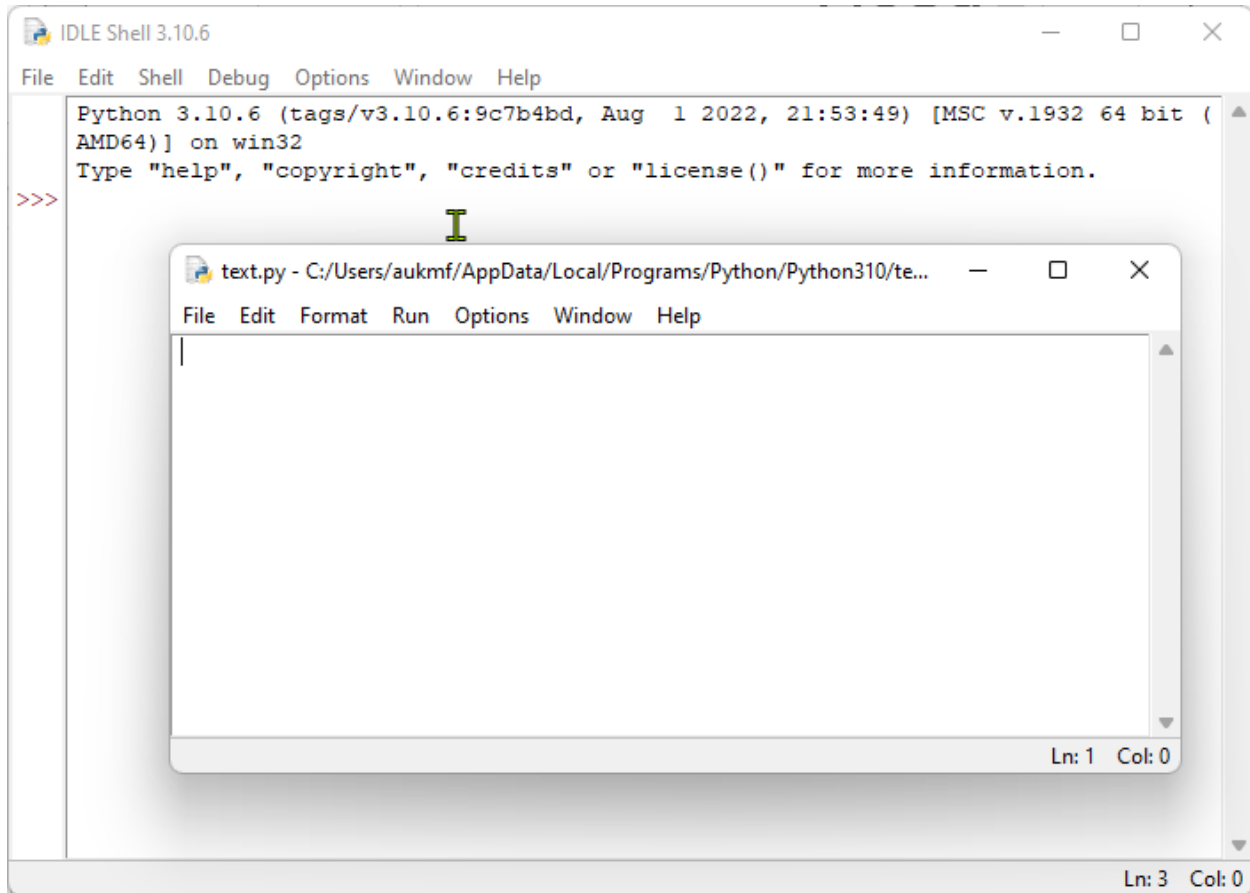
Note: don't set any extension for the file name you are going to use. Python needs its files to have the .py extension, so you should rely on the dialog window's defaults. Using the standard .py extension enables the OS to properly open these files.

How to write and run your very first program

Now put just one line into your newly opened and named editor window.

The line looks like this:

```
print("Welcome to Python...")
```

You can use the clipboard to copy the text into the file.

We're not going to explain the meaning of the program right now. You'll find a detailed discussion in the next chapter.

Take a closer look at the quotation marks. These are the simplest form of quotation marks (neutral, straight, dumb, etc.) commonly used in source files. Do not try to use typographic quotes (curved, curly, smart, etc.), used by advanced text processors, as Python doesn't accept them.
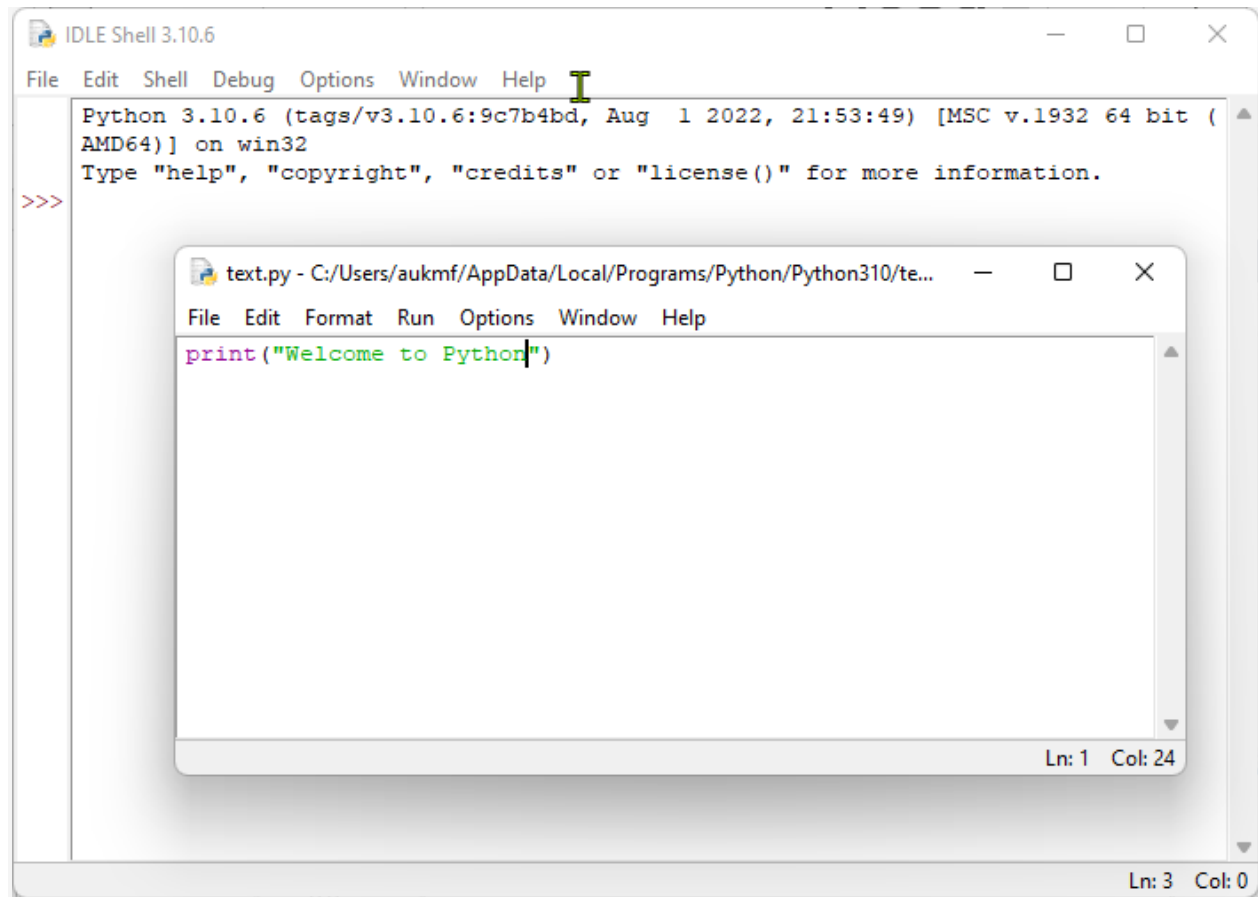
Save the file (*File -> Save*) and run the program (*Run -> Run Module*).

- If everything goes okay and there are no mistakes in the code, the console window will show you the effects caused by running the program.
- In this case, the program **Welcome to Python**.
- Try to run it once again. And once more.
- Now close both windows now and return to the desktop.