ENERGY GeO Hackathon
Society of Petroleum Engineers

SPE International®
SPE Europe Region

2023 SPE EUROPE ENERGY GEOHACKATHON

Introduction to Python

Vikas Kooneti

16 October 2023

#DatafyingEnergy

About me

I have currently working as an Energy Economics & Finance Advisor an Innovate Energy Tech Company **“eCERTO”** in the United Kingdom, with over 2 years of experience in Energy Economics & Finance. My expertise lies at the intersection of Engineering, Economics, and Data Science, focusing towards Economic Modeling and Analysis, predominantly employing Excel & Python as tool of choice.

I have a Bachelor's degree in Petroleum Engineering from the University of Portsmouth, followed by a Master's degree in Petroleum Energy Economics and Finance from the University of Aberdeen. I have also undergone various Continuing Professional Development (CPD) programs and holds Certifications in Python Programming and Data Science, underscoring my commitment to staying at the forefront of this field.

I have actively participated as a member of both SPE YP London and the Energy Institute, engaging in professional networks that further enrich his knowledge and expertise.

Content

1. **About Speaker**
2. **Introduction to Python**
3. **Installation of Python**
4. **Python Basics**
5. **Loading Data in Python and Libraries**
6. **Regression Analysis & Machine Learning Example**
7. **Q&A**

Introduction to Python

- Python is a General Purpose high level programming Language known for its readability and clean syntax, making it a better friendly and easy to learn language. It was designed by Guido van Rossum and was first appeared in 1991. The name python came from Monty Python

- Python is an interpreted language, where the code is executed line by line.

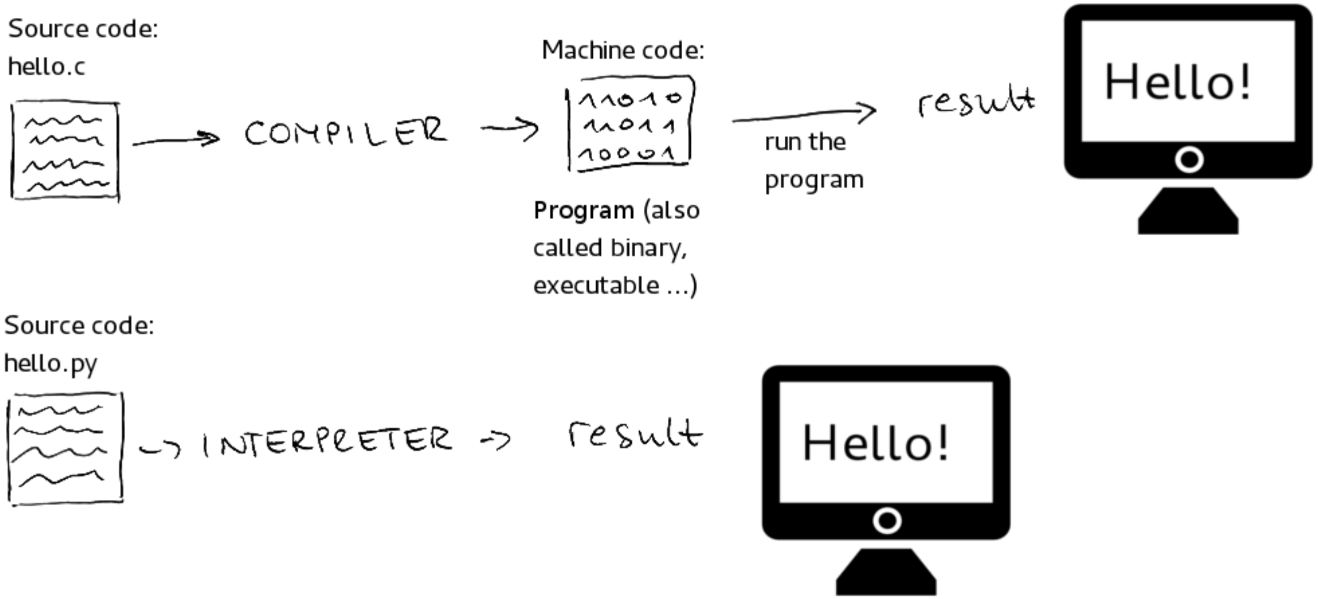
- It is Built in high level data structures simplifying coding.

- Python is scalable and can be used for both small-scale projects and large-scale enterprise applications.

- It supports object-oriented programming, enabling developers to structure code using classes and objects.

- Python encourages collaboration and allows users to view, modify, and distribute the source code freely.

Introduction to Python



Source code:
hello.c

Machine code:

Program (also called binary, executable ...)

run the program

Source code:
hello.py

Multi purpose

Libraries

Free

Dynamic Semantics

Integrated

Multiple Applications

Data

Open Source

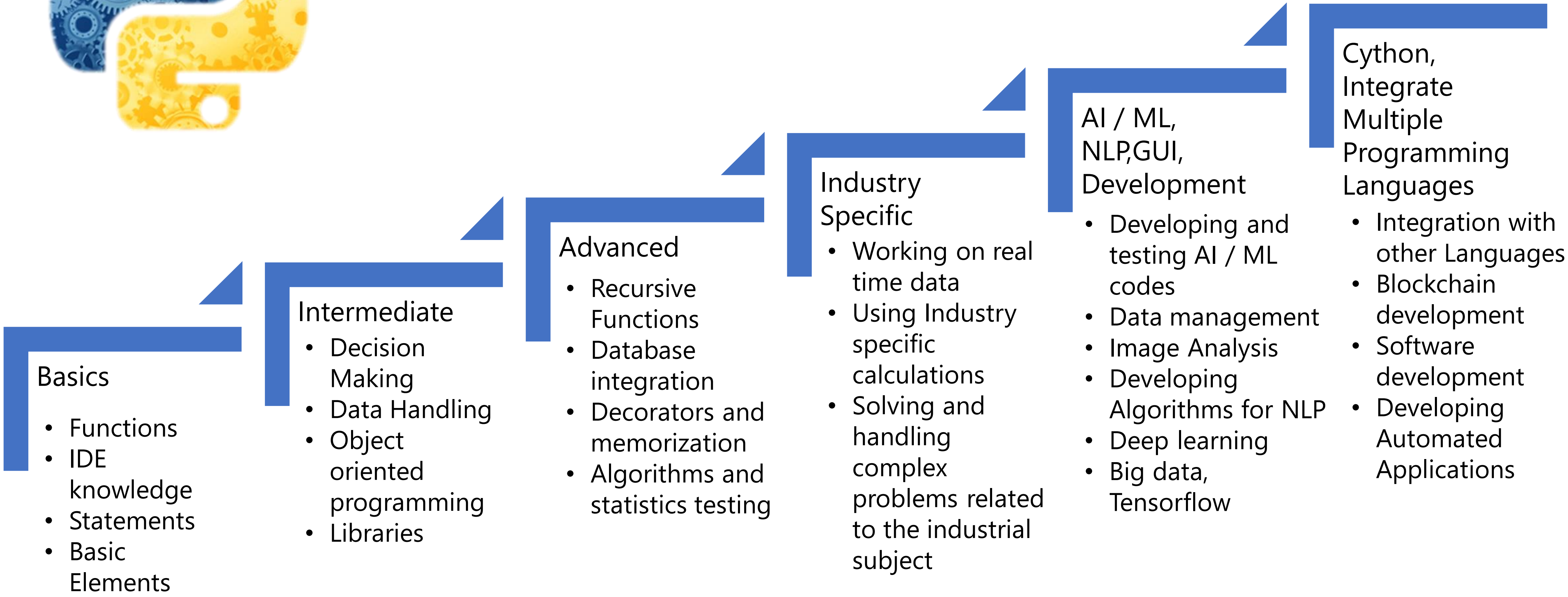
Simple

Easy Syntax

Simplified Syntax

Embedding

Introduction to Python (Python Journey)



Basics

- Functions
- IDE knowledge
- Statements
- Basic Elements

Intermediate

- Decision Making
- Data Handling
- Object oriented programming
- Libraries

Advanced

- Recursive Functions
- Database integration
- Decorators and memorization
- Algorithms and statistics testing

Industry Specific

- Working on real time data
- Using Industry specific calculations
- Solving and handling complex problems related to the industrial subject

AI / ML, NLP,GUI, Development

- Developing and testing AI / ML codes
- Data management
- Image Analysis
- Developing Algorithms for NLP
- Deep learning
- Big data, Tensorflow

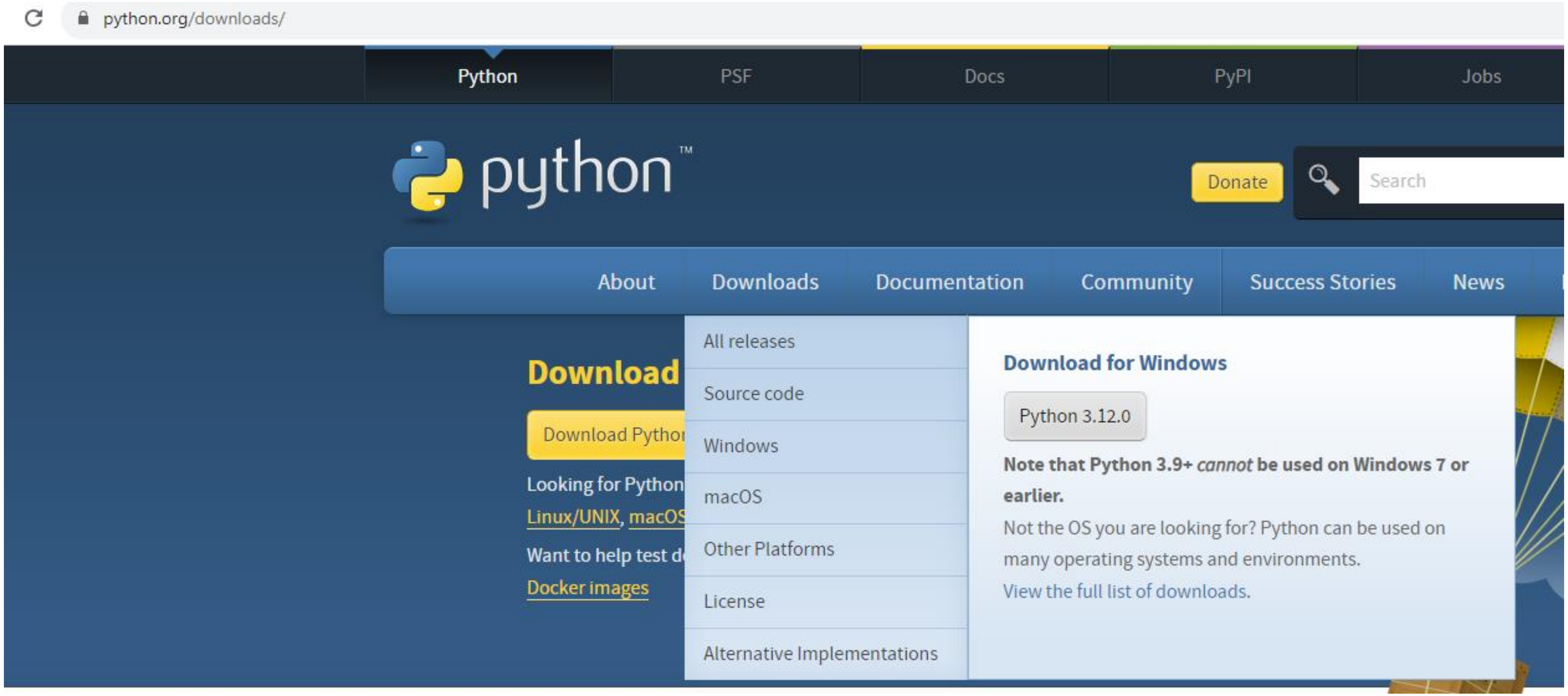
Cython, Integrate Multiple Programming Languages

- Integration with other Languages
- Blockchain development
- Software development
- Developing Automated Applications

Introduction to Python

- Python can be programmed on different IDEs and Operating Systems
- Python Shell and Idle are basic Python IDEs for basic use.
- For Data Science and Analytics most commonly used IDE is Jupyter Notebook.
- Google Colab is also another interesting IDE which can execute arbitrary Python code through browser for learning and analysts purpose.

Installing Python



Looking for Python

Linux/UNIX, macOS

Want to help test d

Download for Windows

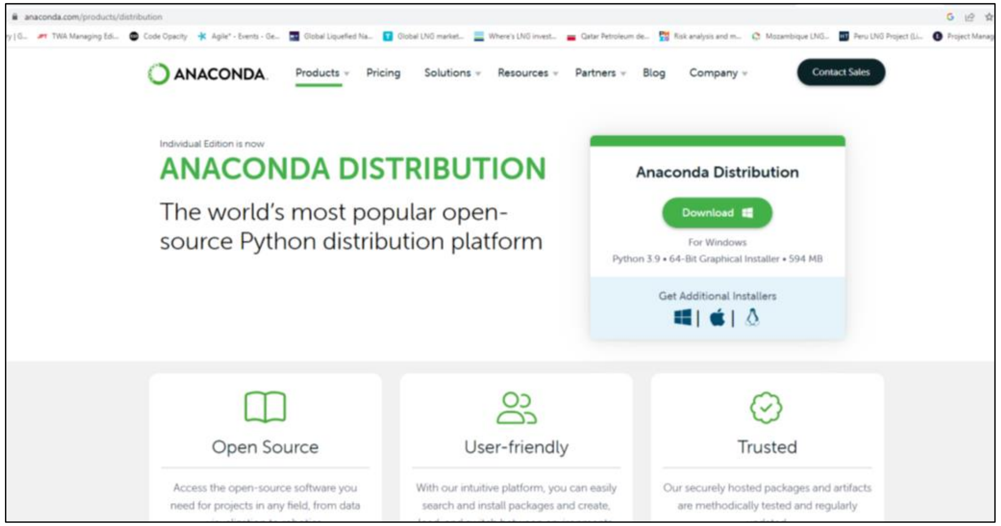
Note that Python 3.9+ *cannot* be used on Windows 7 or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

View the full list of downloads.

Installing Python (IDE & Enviornment)

1



Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

Anaconda Distribution

For Windows

Python 3.9 • 64-Bit Graphical Installer • 594 MB

Open Source

Access the open-source software you need for projects in any field, from data

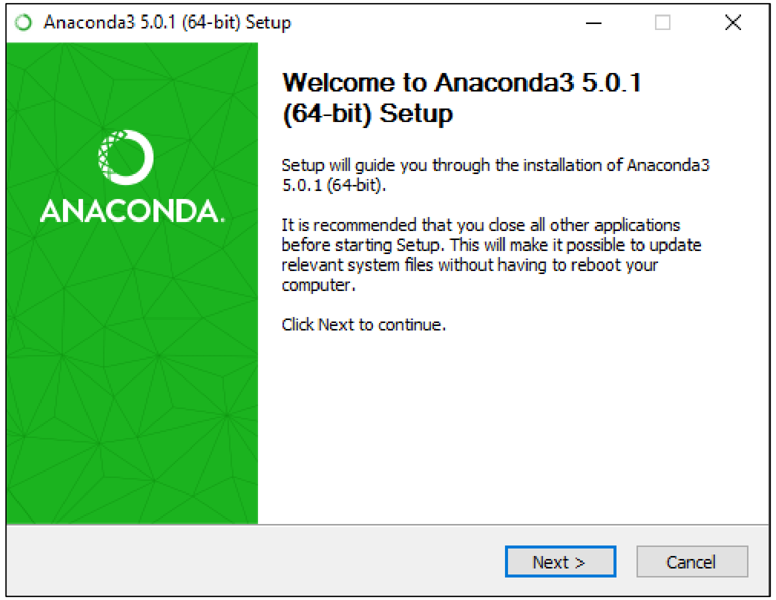
User-friendly

With our intuitive platform, you can easily search and install packages and create,

Trusted

Our securely hosted packages and artifacts are methodically tested and regularly

2



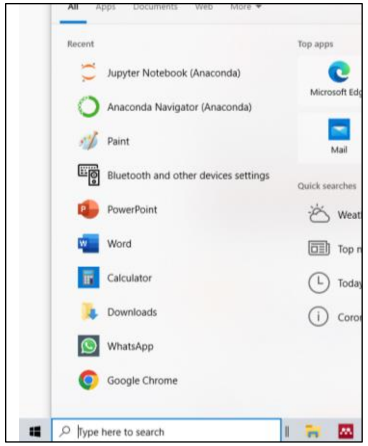
Welcome to Anaconda3 5.0.1 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 5.0.1 (64-bit).

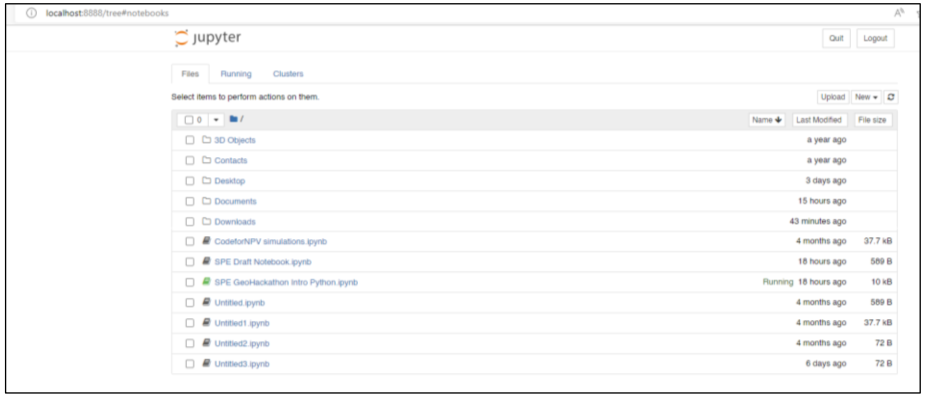
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

3



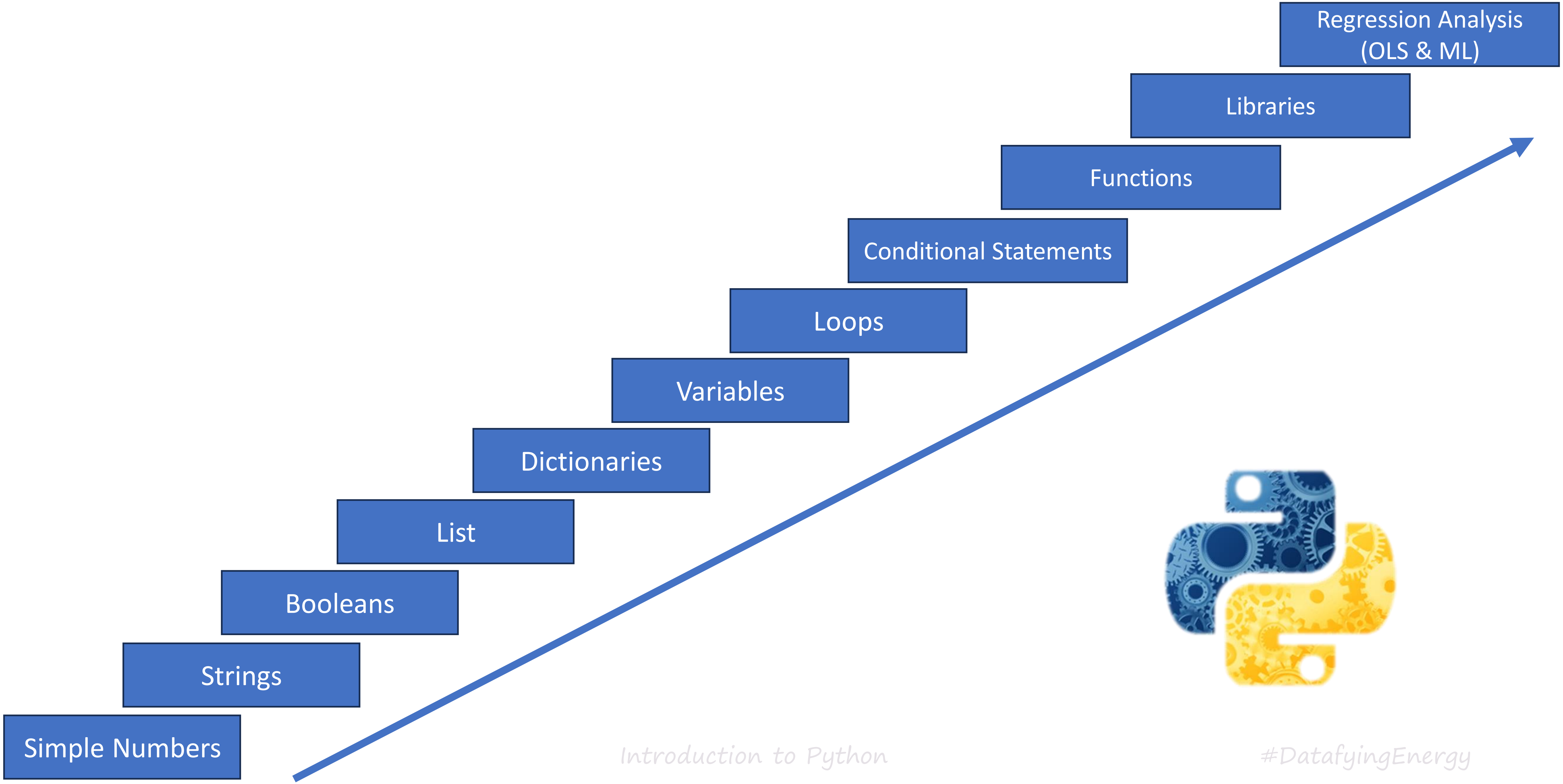
4



Select items to perform actions on them.

Name	Last Modified	File size
3D Objects	a year ago	
Contacts	a year ago	
Desktop	3 days ago	
Documents	15 hours ago	
Downloads	43 minutes ago	
CodeforNPV simulations.ipynb	4 months ago	37.7 kB
SPE Draft Notebook.ipynb	18 hours ago	589 B
SPE GeoHackathon Intro Python.ipynb	Running 18 hours ago	10 kB
Untitled.ipynb	4 months ago	589 B
Untitled1.ipynb	4 months ago	37.7 kB
Untitled2.ipynb	4 months ago	72 B
Untitled3.ipynb	6 days ago	72 B

Python Basics



Simple Numbers

Strings

Booleans

List

Dictionaries

Variables

Loops

Conditional Statements

Functions

Libraries

Regression Analysis (OLS & ML)

Python Basics

Strings

- **Character Sequence:** Python strings represent sequences of characters, encompassing letters, numbers, and symbols.
- **Immutability:** Once a string is instantiated, it remains unalterable. Any operation seeming to modify a string results in the creation of a new string.
- **Quotation Enclosure:** Strings are delineated by enclosing character sequences within either single (' ') or double (" ") quotation marks.
- **Index Accessibility:** Accessing characters within a string is achieved through indexing, starting at 0 for the initial character and progressing accordingly.

Booleans

- **Binary Logic Representation:** Booleans in Python are a data type that represents binary logic, where values can be either True or False.
- **Logical Operations:** Booleans are commonly used in logical operations and expressions, helping in decision-making and control flow in a program.
- **Comparison Results:** Boolean values often result from comparisons (e.g., equality or inequality) and serve as indicators of the truth or falsity of a statement.
- **Fundamental for Control Flow:** Booleans play a crucial role in conditional statements (if, else, elif) and looping constructs, enabling the creation of dynamic and responsive program behavior.

Python Basics

Lists

•**Ordered Collection:** Lists in Python are ordered collections of items, allowing for the storage of elements in a specific sequence.
•**Mutable:** Lists can be modified after creation; elements can be added, removed, or altered within a list.
•**Heterogeneous Elements:** Lists can contain a mix of different data types, including numbers, strings, or even other lists.
•**Indexed Access:** Elements in a list can be accessed using indices, starting from 0 for the first element. This allows for efficient retrieval and manipulation of individual elements within the list.

Dictionaries

•**Key-Value Pairs:** Dictionaries in Python are collections of key-value pairs, where each key is associated with a corresponding value.
•**Unordered:** Unlike lists, dictionaries are unordered collections, meaning that the order of key-value pairs is not guaranteed.
•**Mutable:** Dictionaries can be modified after creation. This includes adding new key-value pairs, removing existing ones, and updating values associated with keys.
•**Key-Based Access:** Instead of using indices as in lists, dictionaries allow access to values based on their keys. This enables efficient retrieval of values using descriptive keys rather than numeric indices.

Python Basics

Variables:

•**Storage for Data:** Variables in Python are used to store and manage data. They act as named containers that hold values or references to objects.

•**Dynamic Typing:** Python is dynamically typed, meaning you don't need to explicitly declare the type of a variable. The interpreter infers the type based on the assigned value.

•**Mutable Values:** Variables can be reassigned to different values during the program's execution. This flexibility contributes to the dynamic nature of Python programs.

•**Symbolic Names:** Variables are given names that serve as symbolic references to memory locations. These names make the code more readable and maintainable.

Loops:

•**Repetitive Execution:** Loops in Python, such as for and while loops, enable the repetitive execution of a block of code. This is valuable for automating tasks or processing data iteratively.

•**Iteration Control:** Loops provide a way to iterate over elements in a sequence (e.g., a list) or execute a block of code until a certain condition is met (e.g., while loop).

•**Enhanced Efficiency:** Loops help in writing more concise and efficient code, especially when dealing with repetitive tasks, as they eliminate the need for duplicating code.

•**Flow Control:** Loops contribute to the flow control of a program, allowing developers to create structures that handle repetitive tasks and respond dynamically to changing conditions.

Python Basics

Conditional Statements:

- **Decision Making**: Conditional statements in Python, like `if`, `else`, and `elif`, allow the execution of different code blocks based on certain conditions. They facilitate decision-making within a program.
- **Branching Logic**: These statements create branches in the code, where different paths are followed depending on whether a given condition evaluates to True or False.
- **Control Flow**: Conditional statements determine the flow of the program by enabling it to take different routes based on the truth value of specified conditions.
- **Logical Evaluation Conditions**: in these statements are evaluated as either true or false, and the corresponding block of code is executed based on the result.

Functions:

- **Modular Code:** Functions in Python encapsulate a set of instructions, allowing the grouping of code into reusable and modular units. This enhances code organization and maintainability.
- **Parameterized:** Functions can take parameters (inputs) that influence their behavior. These parameters make functions adaptable to different use cases and scenarios.
- **Return Values:** Functions can return values, providing a way to obtain results or outputs from the executed code. This enhances the flexibility of functions in various contexts.
- **Code Abstraction Functions:** allow developers to abstract details, enabling them to focus on high-level logic without delving into the implementation details of a specific functionality.

Python Basics

Libraries in Python:

•**Reusable Code Modules:** Libraries in Python are collections of pre-written code modules that provide a set of functionalities. They contain functions, classes, and constants that can be reused in different programs.

•**Code Abstraction:** Libraries abstract complex operations, algorithms, or functionalities, allowing developers to use high-level functions without needing to understand the underlying implementation details. This promotes code simplicity and readability.

•**Time and Effort Savings:** By leveraging existing libraries, developers can save time and effort. Instead of writing code from scratch, they can use well-tested and optimized functions, accelerating the development process.

•**Specialized Functionality:** Libraries often focus on specialized areas such as data manipulation, machine learning, web development, etc. This specialization allows developers to use the right tools for specific tasks, contributing to efficient and robust software development.

NumPy

pandas

SciPy

scikit learn

matplotlib

Regression Analysis:

- **Definition:** Regression analysis is a statistical technique used in data analysis to model the relationship between a dependent variable and one or more independent variables. It is commonly employed for predicting the value of the dependent variable based on the values of the independent variables.

OLS Regression (Ordinary Least Squares Regression):

- **Definition:** OLS is a method used in linear regression analysis to estimate the parameters of the linear relationship between the dependent variable and the independent variables. It aims to minimize the sum of the squared differences between the observed and predicted values.

Python's extensive ecosystem of libraries provides a comprehensive toolkit for regression analysis, offering capabilities for data handling, numerical operations, visualization, machine learning, statistical modeling, and more. This flexibility and abundance of resources make Python a preferred language for analysts, and data scientists working on regression-related tasks.

ENERGY GeO Hackathon
Society of Petroleum Engineers

Q&A

SPE International Italian Section

SPE International Netherlands Section

SPE International London Section

SPE International Romanian Section

SPE International Croatian Section

SPE International Central Ukraine Section

SPE International Geothermal Technical Section

SPE International Data Science and Engineering Analytics Technical Section

ENERGY Geo Hackathon

Society of Petroleum Engineers

www.spehackathon-eu.com

#DatafyingEnergy

SPE International®

SPE Europe Region