

Lecture 12

Learning outcomes:

- *Python library functions*
- *Basic algorithms interpreted on vectors*
- *Structures and basic file management*
- *Computer viruses and protection*

Python library functions

Library functions

- *Normally*, a library is a collection of books or is a room or place where many books are stored to be used later.
- *Similarly*, in the programming world, a library is a collection of precompiled codes that can be used later on in a program for some specific well-defined operations.

Library in Python

- A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs.
- It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs.
- Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc.

Working of Python Library

- As is stated above, a Python library is simply a collection of codes or modules of codes that we can use in a program for specific operations.
- We use libraries so that we don't need to write the code again in our program that is already available. But how it works.
- Actually, in the MS Windows environment, the library files have a DLL extension (Dynamic Load Libraries).
- When we link a library with our program and run that program, the linker automatically searches for that library. It extracts the functionalities of that library and interprets the program accordingly. That's how we use the methods of a library in our program.

Python standard library

- The Python Standard Library contains the exact syntax, semantics, and tokens of Python. It contains built-in modules that provide access to basic system functionality like I/O and some other core modules.
- Most of the Python Libraries are written in the C programming language. The Python standard library consists of more than 200 core modules. All these work together to make Python a high-level programming language.
- Python Standard Library plays a very important role. Without it, the programmers can't have access to the functionalities of Python. But other than this, there are several other libraries in Python that make a programmer's life easier. Let's have a look at some of the commonly used libraries:

1. TensorFlow: This library was developed by Google in collaboration with the Brain Team. It is an open-source library used for high-level computations. It is also used in machine learning and deep learning algorithms. It contains a large number of tensor operations. Researchers also use this Python library to solve complex computations in Mathematics and Physics.

❖ *Common Functions:*

- `tensorflow.constant()`: Creates a constant tensor.
- `tensorflow.Variable()`: Creates a variable tensor.
- `tensorflow.keras.Sequential()`: Creates a sequential model.
- `tensorflow.keras.layers.Dense()`: Adds a dense layer to the model.
- `tensorflow.keras.optimizers.Adam()`: Adam optimizer.
- `tensorflow.keras.losses.MeanSquaredError()`: Computes mean squared error.

2. Matplotlib: This library is responsible for plotting numerical data. And that's why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc.

❖ *Common Functions:*

- `matplotlib.pyplot.plot()`: Plots data on a graph.
- `matplotlib.pyplot.xlabel()`: Sets the label for the x-axis.
- `matplotlib.pyplot.ylabel()`: Sets the label for the y-axis.
- `matplotlib.pyplot.title()`: Sets the title of the graph.
- `matplotlib.pyplot.show()`: Displays the graph.
- `matplotlib.pyplot.hist()`: Plots a histogram.

3. Pandas: Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.

❖ *Common Functions:*

- `pandas.read_csv()`: Reads a CSV file into a DataFrame.
- `pandas.DataFrame()`: Creates a DataFrame.
- `DataFrame.head()`: Returns the first few rows of a DataFrame.
- `DataFrame.describe()`: Provides descriptive statistics.
- `DataFrame.groupby()`: Groups data based on a column.
- `DataFrame.merge()`: Merges two DataFrames.

4. Numpy: The name “Numpy” stands for “Numerical Python”. It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally to perform several operations on tensors. Array Interface is one of the key features of this library.

❖ *Common Functions:*

- `numpy.array()`: Creates an array.
- `numpy.zeros()`: Creates an array filled with zeros.
- `numpy.ones()`: Creates an array filled with ones.
- `numpy.linspace()`: Generates an array of evenly spaced values.
- `numpy.dot()`: Computes the dot product of two arrays.
- `numpy.sum()`: Sums elements of an array.

5. SciPy: The name “SciPy” stands for “Scientific Python”. It is an open-source library used for high-level scientific computations. This library is built over an extension of Numpy. It works with Numpy to handle complex computations. While Numpy allows sorting and indexing of array data, the numerical data code is stored in SciPy. It is also widely used by application developers and engineers.

❖ **Common Functions:**

- `scipy.integrate()`: Performs numerical integration.
- `scipy.optimize()`: Contains functions for optimization.
- `scipy.stats()`: Contains statistical functions.
- `scipy.signal()`: Signal processing functions.
- `scipy.linalg()`: Linear algebra functions.

6. Scikit-learn: It is a famous Python library to work with complex data. Scikit-learn is an open-source library that supports machine learning. It supports variously supervised and unsupervised algorithms like linear regression, classification, clustering, etc. This library works in association with Numpy and SciPy.

❖ *Common Functions:*

`sklearn.model_selection.train_test_split()`: Splits data into training and testing sets.

`sklearn.preprocessing.StandardScaler()`: Standardizes features by removing the mean and scaling to unit variance.

`sklearn.linear_model.LinearRegression()`: Implements linear regression.

`sklearn.ensemble.RandomForestClassifier()`: Implements a random forest classifier.

`sklearn.metrics.accuracy_score()`: Calculates the accuracy of a model.

7. Scrapy: It is an open-source library that is used for extracting data from websites. It provides very fast web crawling and high-level screen scraping. It can also be used for data mining and automated testing of data.

8. PyGame: This library provides an easy interface to the Standard Directmedia Library (SDL) platform-independent graphics, audio, and input libraries. It is used for developing video games using computer graphics and audio libraries along with Python programming language.

9. PyTorch: PyTorch is the largest machine learning library that optimizes tensor computations. It has rich APIs to perform tensor computations with strong GPU acceleration. It also helps to solve application issues related to neural networks.

10. PyBrain: The name “PyBrain” stands for Python Based Reinforcement Learning, Artificial Intelligence, and Neural Networks library. It is an open-source library built for beginners in the field of Machine Learning. It provides fast and easy-to-use algorithms for machine learning tasks. It is so flexible and easily understandable and that’s why is really helpful for developers that are new in research fields.

11. Seaborn: This package makes statistical model visualization possible. The library, which is largely based on Matplotlib, makes statistical graphics possible by:

- Variable examination by means of a Programming interface in view of datasets.
- Make complex representations effortlessly, including multi-plot frameworks.
- To compare data subsets, univariate and bivariate visualizations are utilized.
- A wide range of color schemes are available for pattern displays.
- Direct relapse assessment and plotting are done consequently.

12. Keras: Keras is a Python-based open-source neural network library that enables in-depth research into deep neural networks. Keras emerges as a viable option as deep learning becomes more common because, according to its developers, it is an API (Application Programming Interface) designed for humans rather than machines. Keras has a higher rate of adoption in the research community and industry than TensorFlow or Theano. The TensorFlow backend engine should be downloaded first before Keras can be installed.

Use of Libraries in Python Program

```
# Importing math library
```

```
import math
```

```
A = 16
```

```
print(math.sqrt(A))
```

✓ Output

4.0

- Here in the above code, we imported the math library and used one of its methods i.e. sqrt (square root) without writing the actual code to calculate the square root of a number.
- That's how a library makes the programmers' job easier. But here we needed only the sqrt method of math library, but we imported the whole library.
- Instead of this, we can also import specific items from a library module.

Importing specific items from a library module

- As in the above code, we imported a complete library to use one of its methods. But we could have just imported “sqrt” from the math library. Python allows us to import specific items from a library.

➤ *Let's look at an exemplar code:*

```
# Importing specific items  
from math import sqrt, sin  
A = 16  
B = 3.14  
print(sqrt(A))  
print(sin(B))
```

✓ Output

4.0

0.0015926529164868282

Basic algorithms interpreted on vectors

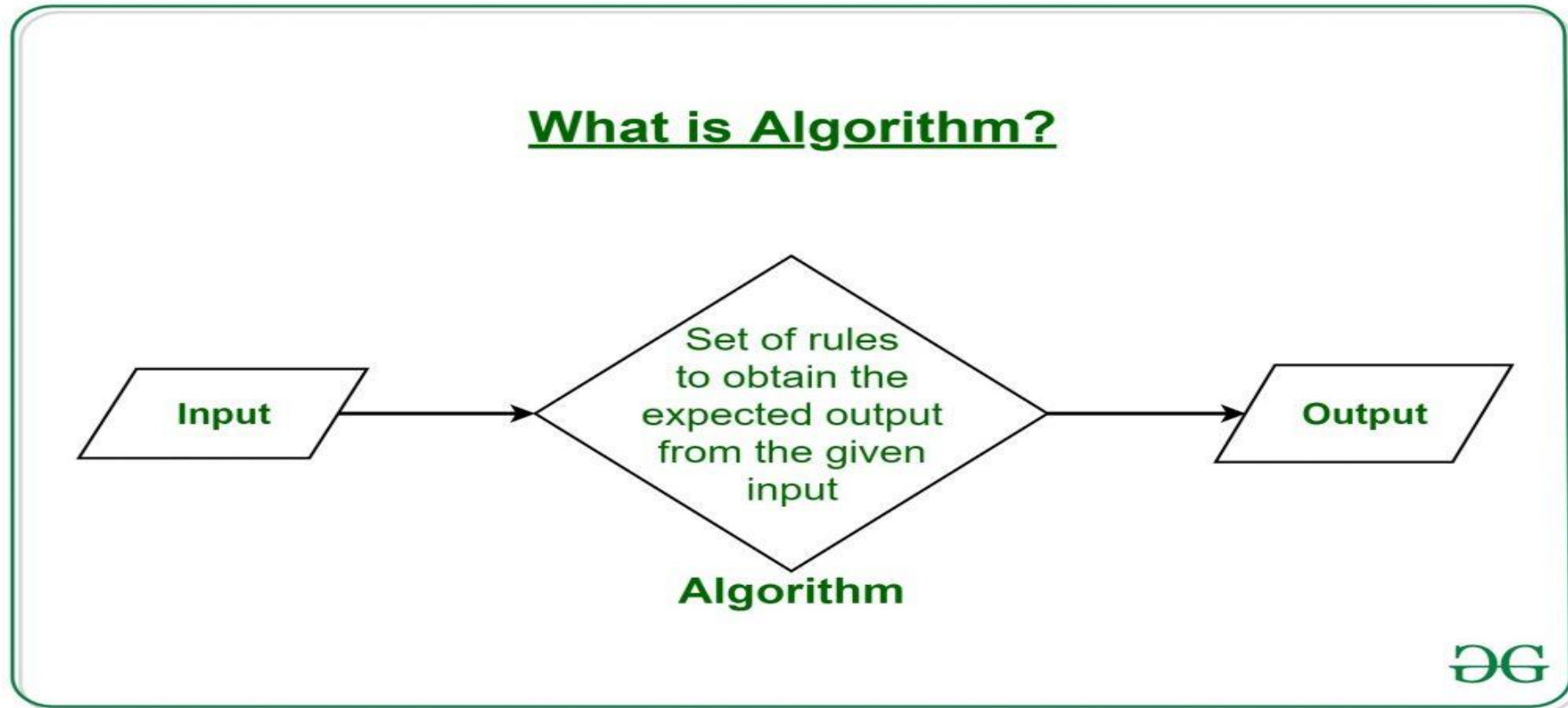
What is Algorithm

❖ The word Algorithm means ” A set of finite rules or instructions to be followed in calculations or other problem-solving operations ”

Or

❖ ” A procedure for solving a mathematical problem in a finite number of steps that frequently involves recursive operations”.

- Therefore Algorithm refers to a sequence of finite steps to solve a particular problem.



What is a vector algorithm?

- A vector algorithms are a type of algorithm designed to operate on vectors to represent data and computations, which are ordered collections of elements (often numbers).
- Vectors are fundamental in various fields, including mathematics, physics, engineering, and computer science, particularly in areas like linear algebra, data science, and machine learning.
- In machine learning, vector algorithms are used to process large datasets and make predictions.

Vector Algorithms in Computing:

- **Vectorization:** The process of converting algorithms from operating on a single value at a time to operating on a set of values (vectors) at one time. This often leverages SIMD (Single Instruction, Multiple Data) instructions in modern CPUs and GPUs.
- **Vector Processing:** Using vector processors or vectorized instructions in general-purpose processors to accelerate the execution of vector operations.

Advantages of Vector Algorithms

- **Efficiency:** They can perform multiple operations simultaneously, taking advantage of parallelism.
- **Simplicity:** Vector operations can often be expressed more succinctly and clearly than their scalar counterparts.
- **Performance:** Utilizing hardware capabilities like SIMD instructions can significantly speed up computations.

Vector Operations:

- Addition: Adding corresponding elements of two vectors.
- Subtraction: Subtracting corresponding elements of two vectors.
- Scalar Multiplication: Multiplying each element of a vector by a scalar.
- Dot Product: Summing the products of corresponding elements of two vectors.
- Cross Product: A binary operation on two vectors in three-dimensional space.

❑ Example Code: Vector Addition in Python

➤ Here's a simple example of vector addition using Python and NumPy:

```
import numpy as np
```

```
# Define two vectors
```

```
a = np.array([1, 2, 3])
```

```
b = np.array([4, 5, 6])
```

```
# Perform vector addition
```

```
c = a + b
```

```
print("Vector a:", a)
```

```
print("Vector b:", b)
```

```
print("Sum vector c:", c)
```

✓ Output

Vector a: [1 2 3]

Vector b: [4 5 6]

Sum vector c: [5 7 9]

□ This output shows the two original vectors and their sum, demonstrating how vector addition works in NumPy.

Structures and basic File management

What is a file structure?

- A file structure is the way files are organized on a computer. It's important because it helps you find the files you need.
- Adopting a file structure makes getting to the files you need as easy as possible. Specifically, it's to minimize the number of trips to the disk to get the data you're looking for.
- The main goal of developing file structures is to minimize the number of trips to the disk in order to get desired information. It ideally corresponds to getting what we need in one disk access or getting it with as little disk access as possible.

What is File Management?

- File management refers to the process of organizing, storing, and maintaining files and documents in a structured manner.
- It involves various components that help individuals and organizations efficiently manage their files through proper storage, organization, naming conventions, file formats, version control, security, backup and recovery, file sharing, collaboration, and automation.

Importance of File Management in Business and Personal Use

- Efficient file management is crucial for businesses and individuals to streamline operations, enhance productivity, and ensure easy accessibility to important files.
- It minimizes the risk of data loss, improves collaboration and information sharing, and safeguards sensitive and confidential information.

The main components of file management include:

- File storage
- Hierarchical structure
- File naming
- File format
- File versioning

❖ **File Storage**

- Effective file storage practices contribute to better organization and accessibility. Files must be stored in designated folders and drives, either on local storage or cloud-based platforms. Proper storage enhances searchability, reduces clutter, and simplifies file retrieval.

❖ **Hierarchical Structure**

- A hierarchical structure is essential for effective file management. Utilizing folders and subfolders allows for logical grouping and easy navigation through files and directories.

❖ **File Naming**

- Consistent and well-defined file naming conventions are essential for efficient file management. Proper file naming improves searchability, reduces confusion, and enables users to understand the content of files without opening them.

❖ **File Format**

- Choosing the appropriate file format is crucial for effective file management. Different file formats such as PDF, DOC, XLS, JPEG, and MP3 serve different purposes, and selecting the right format ensures compatibility, readability, and preservation of file content.

❖ **File Versioning**

- File versioning allows for the tracking and management of different iterations or modifications of files.
- Version control ensures that the latest modifications are saved while retaining previous versions for reference or potential rollback.

□ **File Security:**

- File security is paramount to protect sensitive and confidential information against unauthorized access, modification, or loss.
- Proactive measures must be implemented to secure files and prevent data breaches.
- Implement access controls, encrypt files, regularly update security software, and educate users about safe file management practices. Use reputable tools and applications designed to enhance file security.

❑ **File Backup and Recovery:**

- File backup and recovery processes are vital to ensure the integrity and availability of files. Regular backups protect against accidental deletion, hardware failure, and data corruption.

❑ **File Sharing and Collaboration:**

- File sharing and collaboration play crucial roles in today's digital age, enabling efficient communication, teamwork, and knowledge sharing. Effective tools and techniques are essential for seamless collaboration.

□ **Automation in File Management:**

- Automation in file management processes can significantly streamline workflows, saving time and effort.
- Various tools and applications automate repetitive tasks, file organization, and maintenance.

➤ *Computer viruses, protection*

❖ **What is a Computer Virus?**

- A computer virus is a malicious software program that infects your computer without your knowledge or consent.
- It can replicate itself and spread to other computers through different methods, such as emails, downloads, or removable drives.

❖ **Why do people create Computer Viruses?**

- People create computer viruses for various reasons: some do it for fun or as a challenge; others do it for financial gain by stealing personal information or selling malware; and some use viruses for political purposes as a form of cyber warfare.

❖ **How do Computer Viruses work?**

- Computer viruses work by attaching themselves to legitimate programs or files on your computer and then modifying them to include the virus code. When you open the infected file or program, the virus activates and starts spreading to other files and programs on your system.

❖ **What are the most common types of Computer Viruses?**

- Some of the most common types of computer viruses include Trojan horses, worms, ransomware, spyware, adware, and rootkits. Each type has its own unique characteristics and ways of infecting your system.

Types of virus

- ❖ One of the first malware viruses was “Rabbit”, which locked devices. Then came “Elk Cloner”, which affected Apple devices; and “Brain”, which hit IBM computers. This led to a list of virus types being drawn up:
 - Resident. Found in a computer system’s memory, they infect files as they’re being selected or closed.
 - Direct action. Installed in the root directory of the hard drive, they attack when you run an infected file.
 - Overwrite. Found in files, they erase and replace content.
 - Directory. Also known as “cluster” viruses, they change the location of files when we try to open them.
 - Boot sector. They attack when you start your computer. Other types of secondary viruses

❖ **How can I protect my computer from viruses?**

- To protect your computer from viruses, you should install antivirus software and keep it updated regularly.
- You should also avoid opening suspicious emails or downloading attachments from unknown sources.
- Additionally, you can use a firewall to block unauthorized access to your network.

❖ **How can I prevent infecting others with Computer Viruses?**

- To prevent infecting others with computer viruses, you should avoid sharing files or programs that may contain malware. You should also warn others about suspicious emails or links they may receive from you.

❖ **What should I do if my computer is infected with a virus?**

- If you suspect that your computer is infected with a virus, you should disconnect it from the internet immediately to prevent further damage.
- Then run an antivirus scan to detect and remove any malware on your system.