PROGRAMMING LANGUAGES IN DATA ANALYSIS

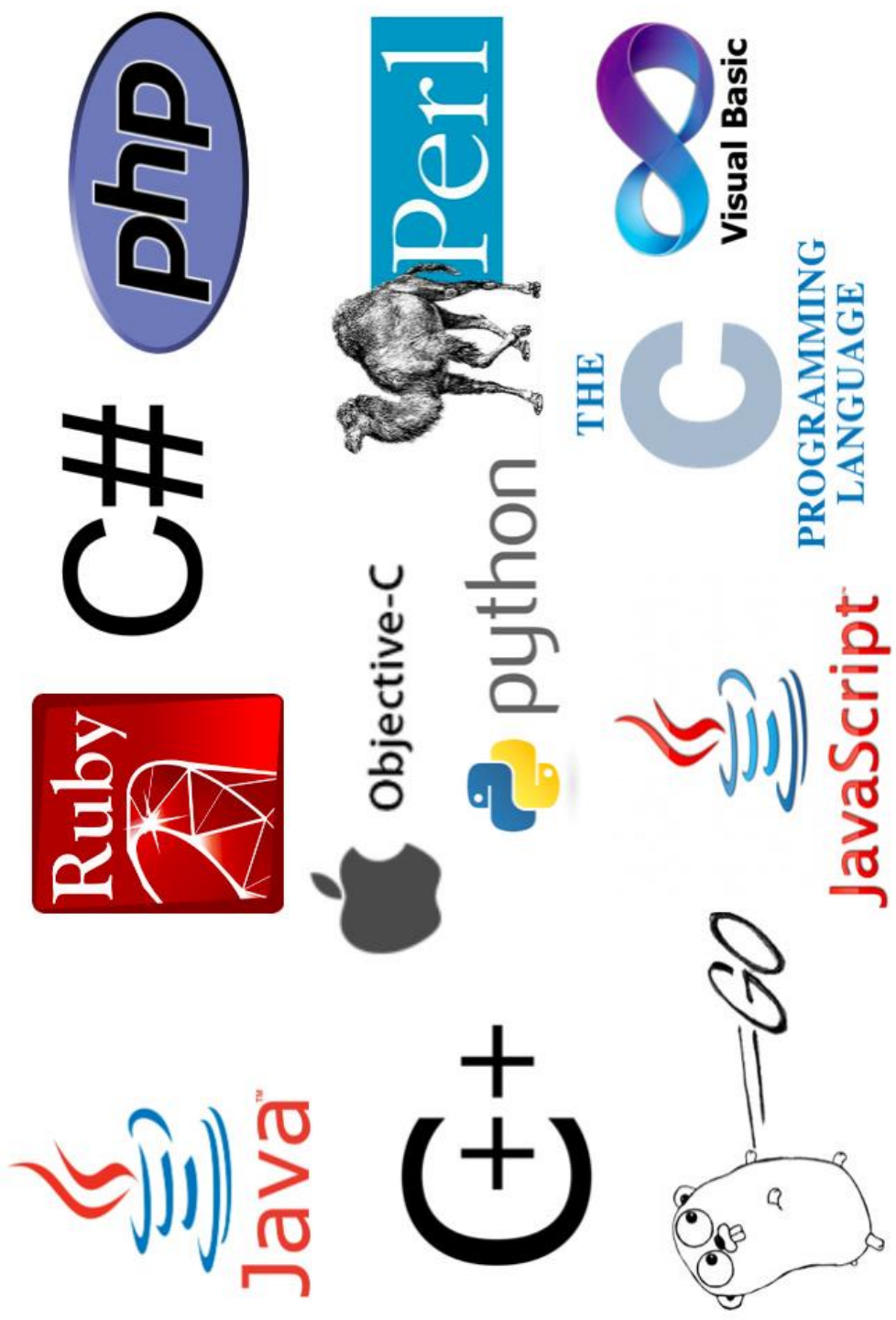
MEMORY EFFICIENCY: COMPILED VS INTERPRETED

AGENDA

- Introduction
- Common Programming Languages In Data Analysis
- Which Type Is More Efficient In Memory Usage During Running: Compiler Or Interpreter

INTRODUCTION

Data Analysis Relies On Programming Languages To Process Information Efficiently, And Understanding The Differences Between Compiled And Interpreted Languages Helps In Choosing The Right Tool For Better Performance And Memory Usage.

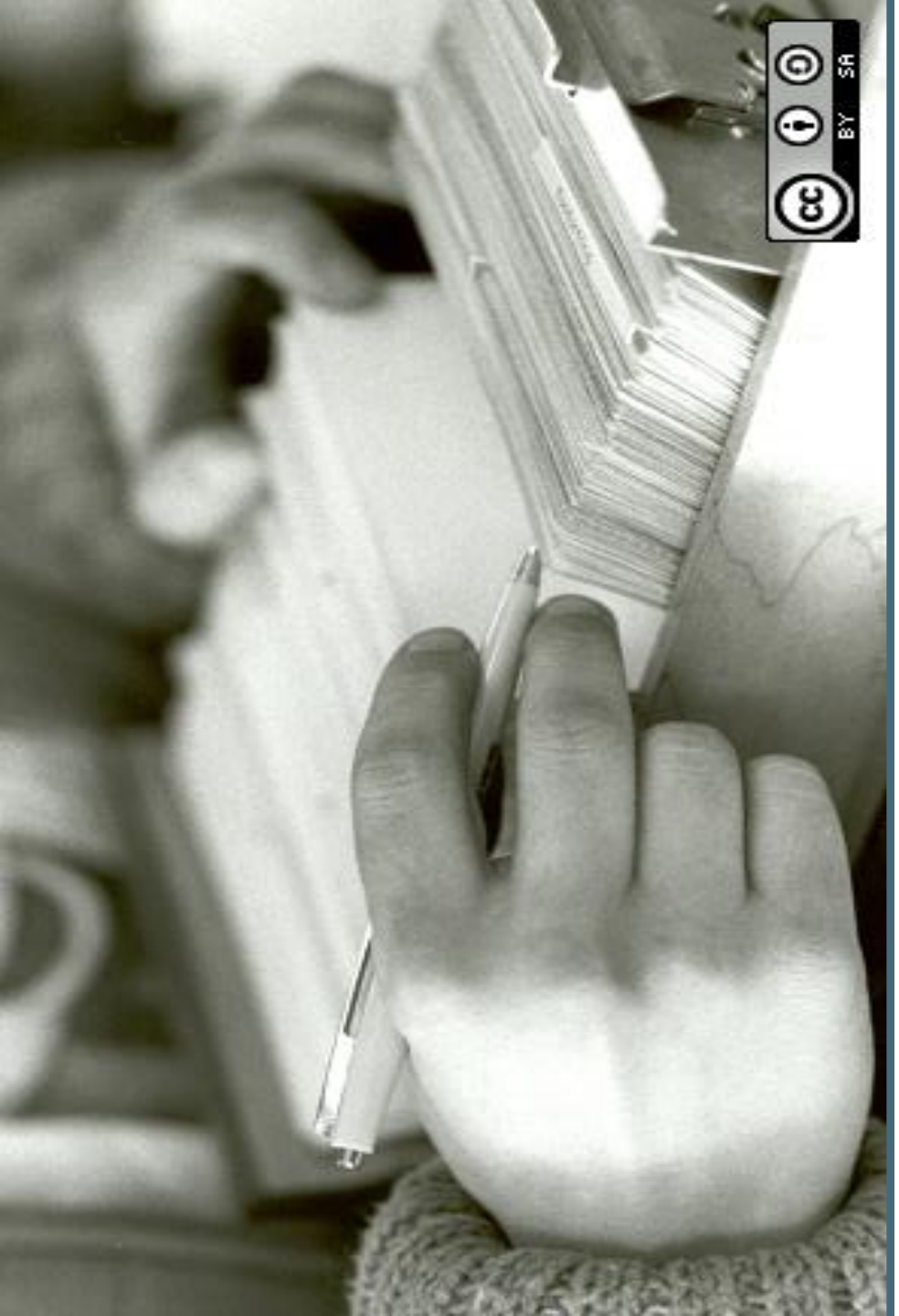


A data analyst's primary responsibility is to study, evaluate, review, and refine data to make informed, strategic decisions for business growth. To efficiently carry out these responsibilities, you must be proficient in various programming languages, alongside other tools and skills needed to thrive as a data analyst. While many programming languages are available in this field, I'll discuss the top eight options to learn .

COMMON PROGRAMMING LANGUAGES IN DATA ANALYSIS

Guido van Rossum's Python programming language is a high-level (easy to read), multi-purpose programming language. It is also one of the most popularly used programming languages in data analytics, according to the TIOBE index. It has a massive community of users and is mainly known for its robust use cases beyond data analytics. For example, you can use it for web development, software development, and artificial intelligence.

Furthermore, Python excels in automation applications. Task automation is a key part of every proficient data analyst's skillset, as it will save you time. It also increases the feasibility of studies during data analysis because of its extensive list of libraries. A few of the notable libraries used are:



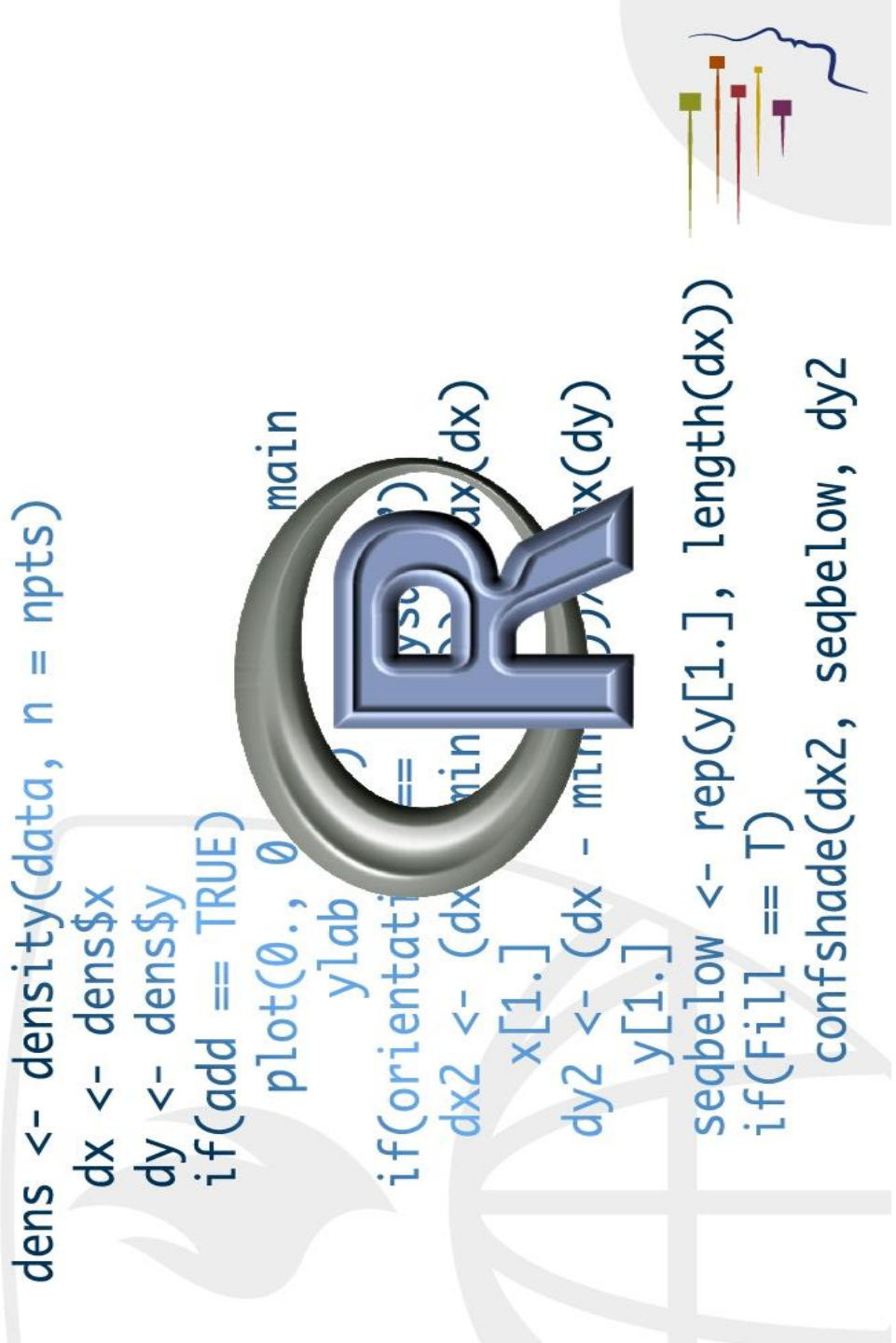
- NumPy stands for Numerical Python: a popular library with an expansive list of advanced mathematical functions like basic linear algebra functions, advanced random number capabilities, and tools for low-level language integration. Examples of such low-level languages are FORTRAN and C++. The most notable feature of the NumPy is the n-dimensional array object.
- SciPy is another notable library built on NumPy, which stands for Scientific Python. SciPy is especially useful for various advanced science and engineering subjects.
- Pandas is a Python library mainly used for analyzing, cleaning, and manipulating data sets. It is one of the most relevant libraries you need when performing data analysis with Python.
- Matplotlib is invaluable for plotting different types of graphs (histograms, line plots, pie plots, and heat plots).
- Seaborn is a python library for making informative and aesthetically pleasing statistical graphics.
- Scikit-learn is one of the Python libraries built on NumPy, SciPy, and Matplotlib. It contains a lot of tools for machine learning and modeling.

Martin Odersky, in 2004, designed the programming language — Scala — as a general-purpose programming language that supports both object-oriented and functional programming. Scala's main advantage is that it offers new solutions to some of the drawbacks of the Java programming language.

Some of the key applications of Scala in data analytics include:

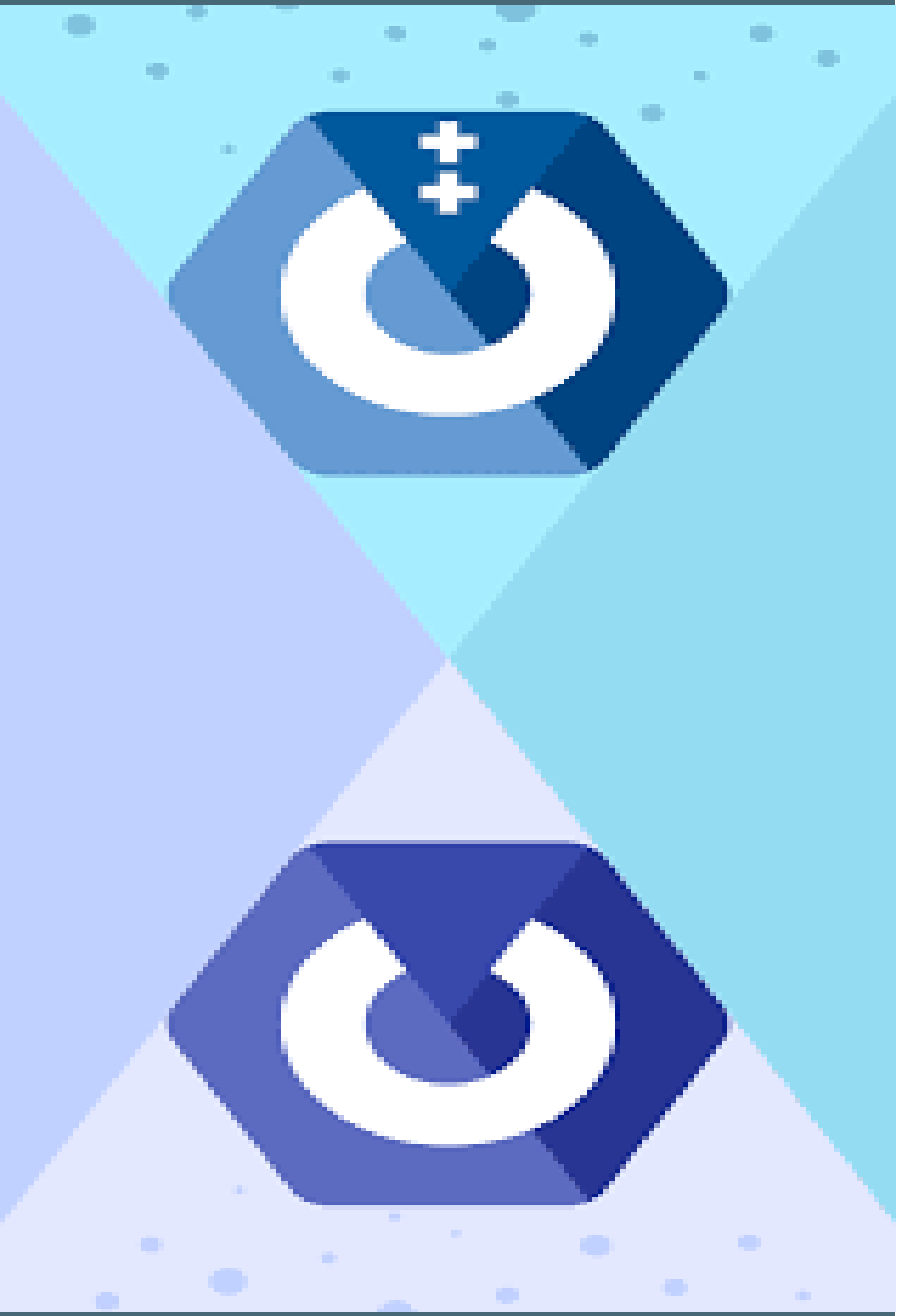
- **Big Data processing**: Scala is used for distributed computing frameworks such as Apache Spark and Flink for processing massive datasets in parallel across multiple nodes.
- **Machine Learning**: Scala's type safety and functional features make it an ideal choice for building scalable and high-performance machine learning applications.
- **Data Processing**: Scala contains numerous rich libraries and APIs that enable efficient data processing and manipulation, making it well-suited for data-driven applications like ETL pipelines.
- **Data Visualization:** Scala's interactive and reactive nature enables the building of real-time data visualization applications, and it has several libraries, such as Apache Zeppelin and Jupyter notebooks.

Moreover, Scala can be run on the Java Virtual Machine and is a perfect choice when working with large datasets and siloed data. It has over 170,000 libraries, making it incredibly valuable for exploring and handling various tasks. Finally, Scala is compatible with IDEs like Sublime Text, IntelliJ IDEA, Atom, VS Code, and your browser.



- R is a solid rival to the Python programming language. While both are immensely popular among data professionals and offer a wide range of similar functions, there are several reasons to learn R in addition to Python.
- The inventors of the R programming language created it for computational statistics and graphical representation of data, also known as data visualization. R is also an open-source language, which makes it possible for the vast community of users to offer quality contributions to improve its performance over time.
- Like the Python programming language, R has an extended library list offering data analytic functions and machine learning operations. Examples include **dplyr**, **tidyr**, **readr**, and **ggplot2**.
- Learning R can be difficult if you are a beginner. You'll need prior knowledge of other programming languages to get a quick hang of it. However, with determination and resilience, you can start working with it quickly. Further, reviews from the community of users online show that it is an exciting language.

- MatLab, or Matrix Laboratory (developed by MathWorks), is a multi-paradigm programming language and a numeric computing platform for high-level mathematical and statistical operations. It is an excellent tool for data visualization and is essential in machine learning, deep learning, robotics, and other emerging technologies.
- Although MatLab has various functions that make it suitable for data analytics, it is a proprietary programming language. Unlike free and open-source programming languages like Java and Python, MatLab requires a certain fee to get a license and gain access.



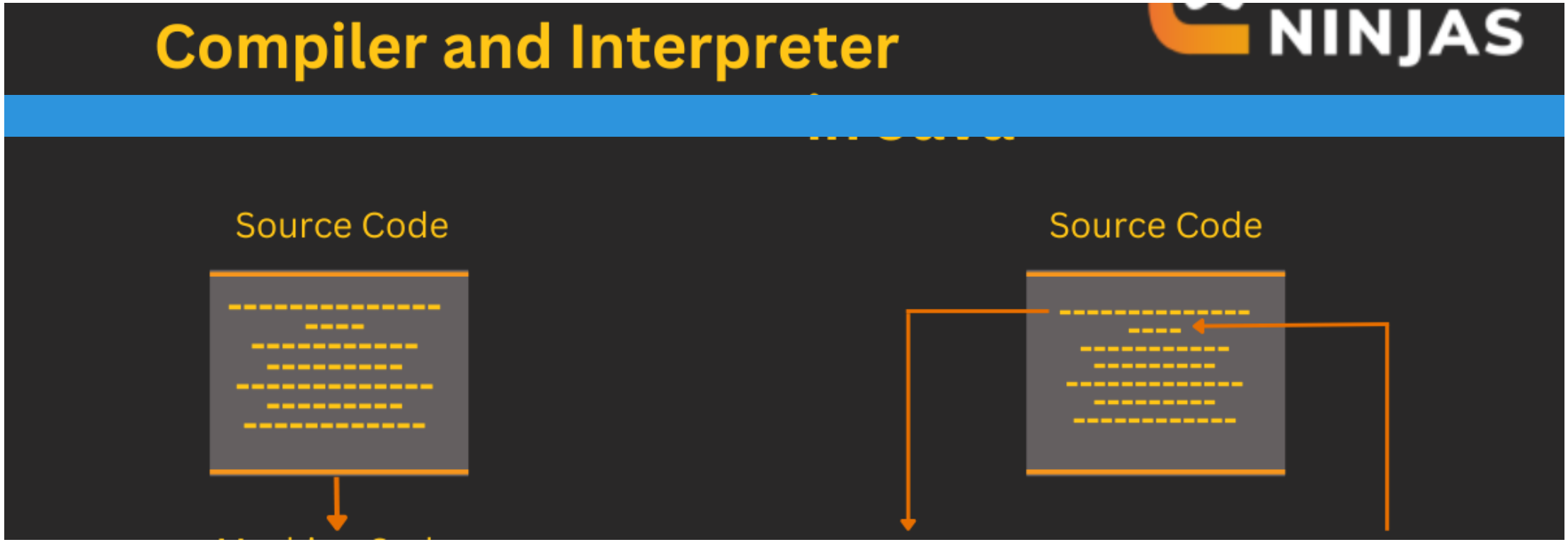
- C is a general-purpose programming language first developed in the early 1970s. It is a low-level language known for its efficiency, performance, and ability to interface with hardware. While C is not typically used as a primary language for data analysis, you can use it with other languages and tools to perform specific data analysis tasks.
- For example, you can use C to optimize performance-critical parts of a data analysis program, such as matrix operations or numerical simulations.
- **C++**
- C++ is an extension of C that was first released in 1985. It is a high-level language known for its versatility, efficiency, and object-oriented programming features. Additionally, it is commonly used in data analysis applications that require high-performance computing and real-time processing, such as financial analysis and scientific simulations.
- C++ also has several libraries and frameworks specifically designed for data analysis, such as the Boost and Armadillo libraries. Overall, as a beginner in data analytics, you should save learning C and C++ programming for last, as they ideally require prior knowledge of data analysis and other programming languages.

- Java is a high-level, general-purpose programming language and computing platform ranked as one of the most popular languages for data analytics.
- One of the unique things about Java is that you can use it to build complex applications from scratch. It also offers a faster computation rate than other programming languages. Most importantly, its “Write once, Run anywhere” function makes it interoperable and easier to integrate operations during data analysis.
- Java is widely used in data analytics due to its portability, scalability, and performance. It provides a robust platform for developing big data applications, machine learning algorithms, and data visualization tools. Some of the popular applications of Java in data analytics include:
- **Hadoop**: Java is the primary language for the Hadoop MapReduce framework, which is used for processing and analyzing large datasets.
- **Spark**: Java is one of the most widely used programming languages for Apache Spark, a distributed computing engine for processing large-scale data.
- **Machine Learning:** Java provides several libraries and frameworks for developing machine learning models, including Weka, Deeplearning4j, and Apache Mahout.
- **Data Visualization**: Java provides several libraries for data visualization, including JavaFX, JFreeChart, and Java 3D.
- **Web Development:** Java provides several web frameworks for developing web-based data analytics applications, including Spring, Struts, and JavaServer Faces.
- Overall, Java offers a comprehensive suite of tools for data analytics, making it a popular choice for developers in this field.

- SQL is a scripting and Domain-Specific Language (DSL) that you can't use for general-purpose programming. It was specifically built to enable users to communicate with, manage, and extract data held in a relational database — all of which are critical stages of data analysis.
- SQL software can be broadly classified into four categories:
- **Relational Database Management Systems (RDBMS):** These are the most common type of SQL software designed to manage structured data. Examples include Oracle, MySQL, Microsoft SQL Server, and PostgreSQL.
- **NoSQL Databases:** These databases do not use the traditional relational data model and are designed to handle unstructured data. Examples include MongoDB and Cassandra.
- **Data Warehousing Tools:** These tools collect and analyze large volumes of data from various sources. Examples include Apache Hadoop and Apache Spark.
- **Cloud Databases:** These databases are hosted on cloud-based platforms and can be accessed from anywhere with an internet connection. Examples include Amazon Aurora, Google Cloud SQL, and Microsoft Azure SQL.
- It has since become instrumental for data queries, management, and other processes used to interact with data. And because data analysis requires database interactions and manipulation, you must learn SQL, no matter your programming language knowledge.

- Lastly, we have Julia: a fast-rising high-level, dynamic programming language explicitly designed for numerical and scientific computing, data analysis, and machine learning. The language was first released in 2012 — it has since gained much popularity among data analysts and researchers due to its speed and versatility.
- Furthermore, Julia's simple syntax makes learning easy for people without prior coding knowledge. It also has an extensive set of built-in functions and libraries, making performing standard data analysis tasks easy.
- Finally, Julia has excellent interoperability with other programming languages, such as Python, R, and C, making integrating its code with existing data analysis workflows easy.

WHICH TYPE IS MORE EFFICIENT IN MEMORY USAGE DURING RUNNING: COMPILER OR INTERPRETER



EXECUTION PROCESS
COPILER VS INTERPRETER

- **Compiler**:
- Scans the entire source code first, checks for errors, and then compiles it all at once.
- Once compiled, the program can be run multiple times without needing the source code or compiler again.
- Errors are reported after the entire code is compiled.
- **Interpreter:**
- Translates and executes the program **line by line.**
- It needs to be run every time the program is executed, as it *doesn't produce a separate executable file.*
- Errors are reported immediately, as they are encountered during the...

SPEED
COPILER VS INTERPRETER

- **Compiler**:
- The compilation process is slower because it analyzes the entire code and produces a complete executable file.
- Once compiled, execution is faster because the code is already translated into machine code.
- **Interpreter**:
- The execution process is slower because the code is translated line by line during runtime.
- It does not require a separate compilation step, but repeated interpretation of the code leads to slower execution.

CONCLUSION

Choosing the right programming language depends on the project requirements; compiled languages offer speed and memory efficiency, while interpreted languages provide flexibility and ease of use.

THANK YOU

Ahmed Gweda Ezzedin

My GitHub

My Linkedin