Novice Programmers

Lecture Series.

Fahad S. Ahmed, MBBS, MD, ABAIM

Lectures

- Lecture 1: Computer literacy
- **Lecture 2: Programming Languages**
- Lecture 3: Python coding 1: Basics
- Lecture 4: Python coding 2: Files and folders
- Lecture 5: Python coding 3: Statistics and graphs

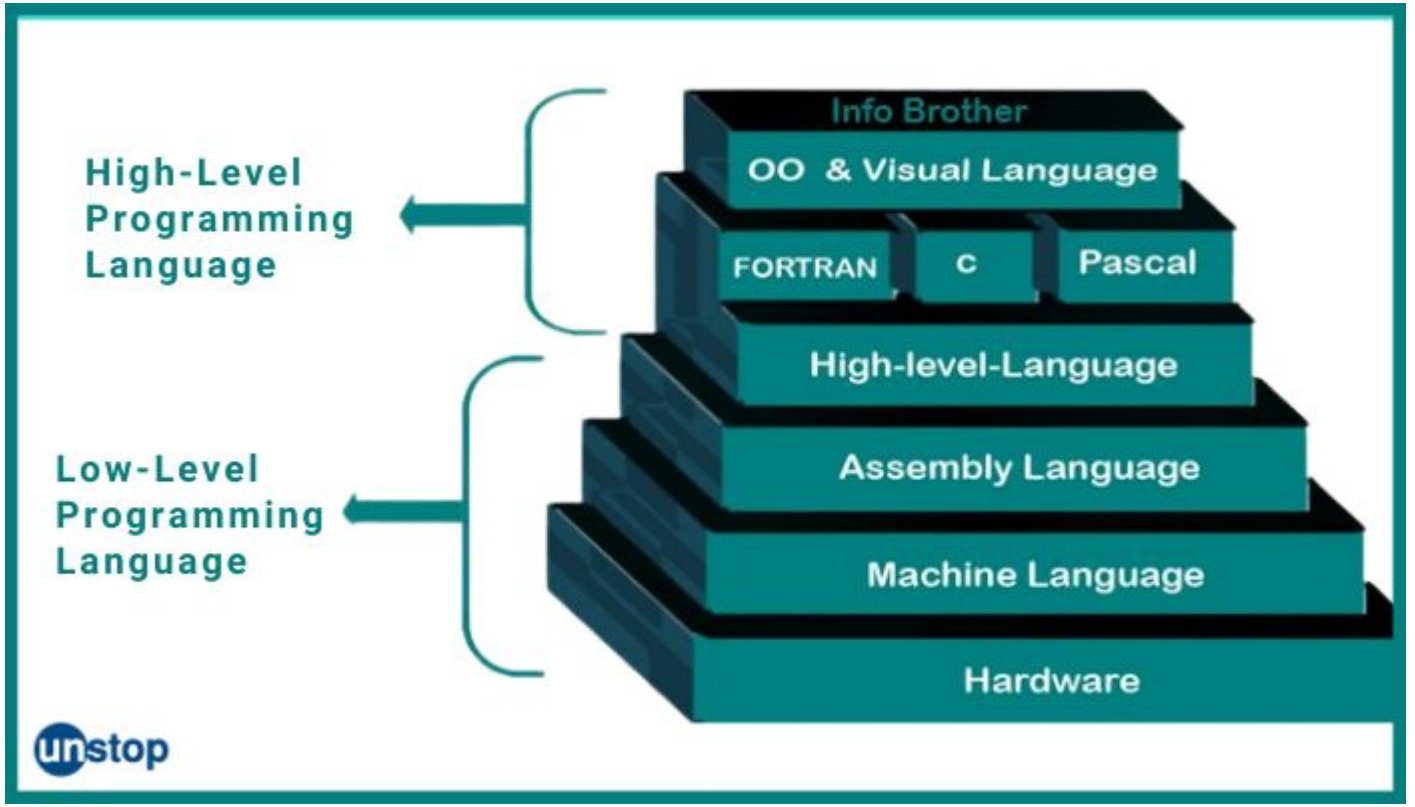
Lecture 2: Programming Languages

Introduction to Programming Languages

- Definition: Programming languages are sets of instructions used to communicate with computers and create software.
- Role: They enable humans to write code that computers can understand and execute.

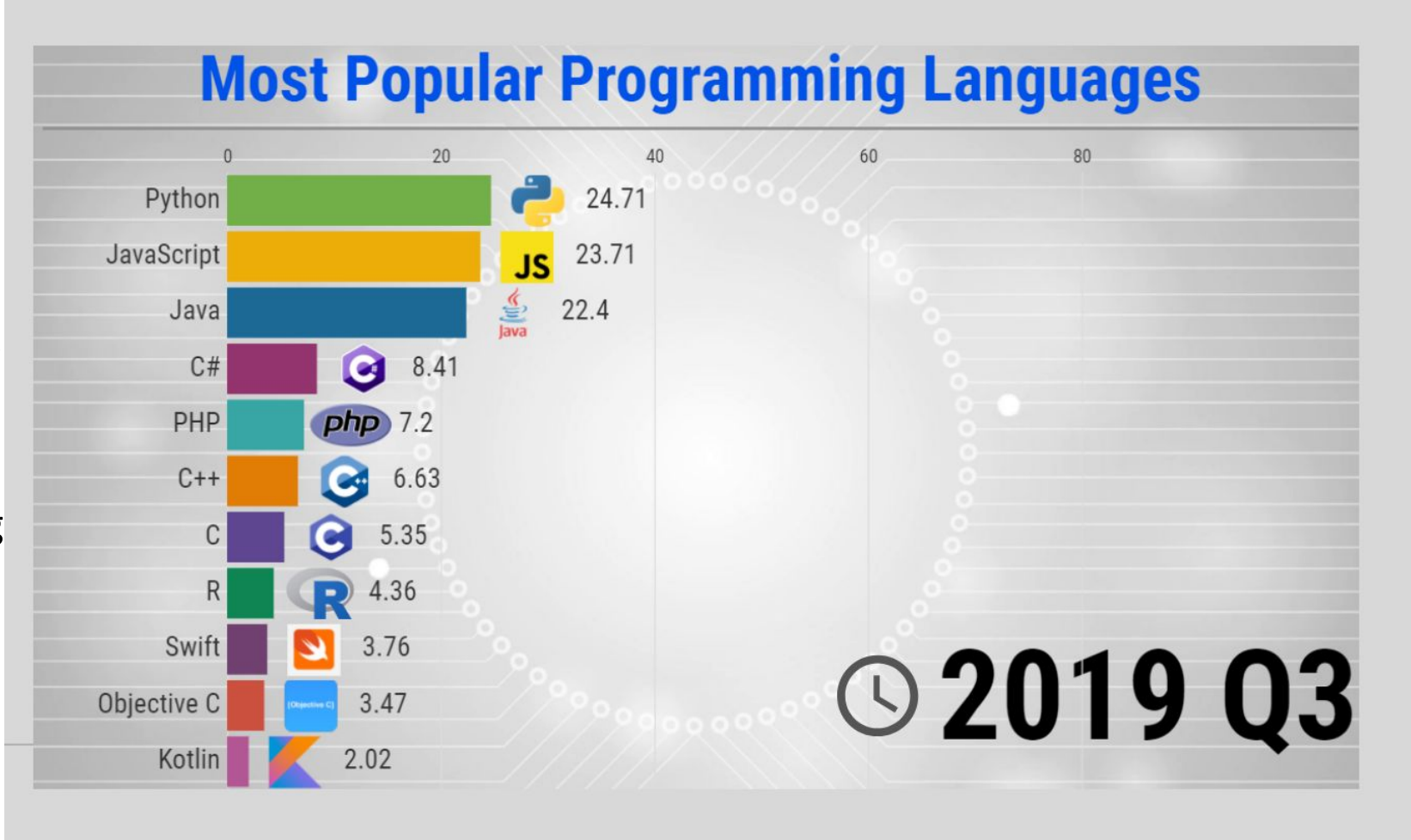
High-Level vs. Low-Level Languages

- High-level languages are more human-readable and abstract, while low-level languages are closer to machine code.
- Examples of high-level languages: Python, Java, C#
- Examples of low-level languages: Assembly language, machine code



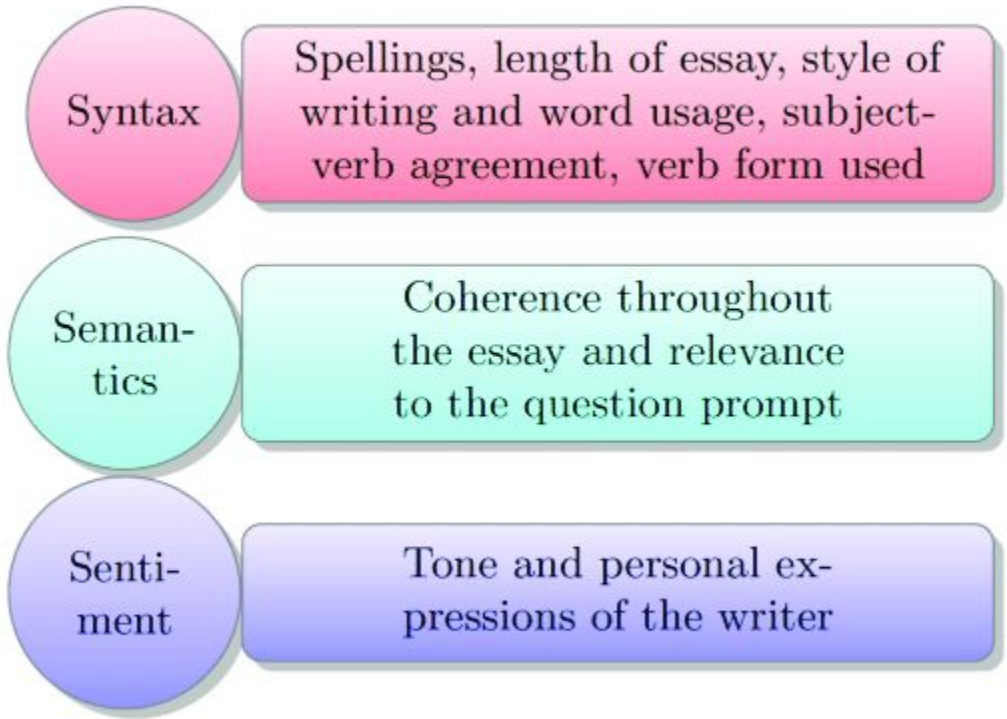
Popular Programming Languages

- Python: Known for its simplicity and versatility.
- JavaScript: Used for web development and adding interactivity to websites.
- Java: Widely used for building cross-platform applications.
- C++: An extension of C, used for system-level programming and game development.
- C#: Developed by Microsoft, used for Windows applications.



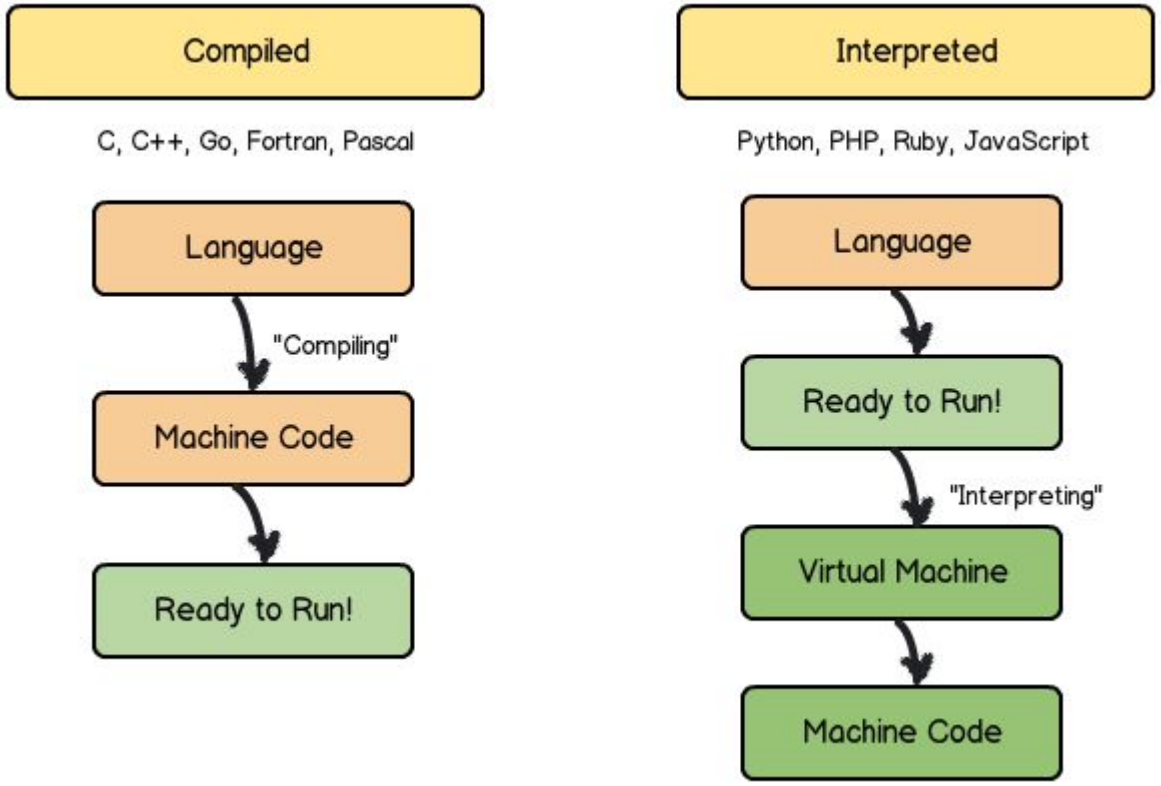
Syntax and Semantics

- Syntax: Rules that dictate how code should be structured.
- Semantics: The meaning behind code instructions and how they affect program behavior.



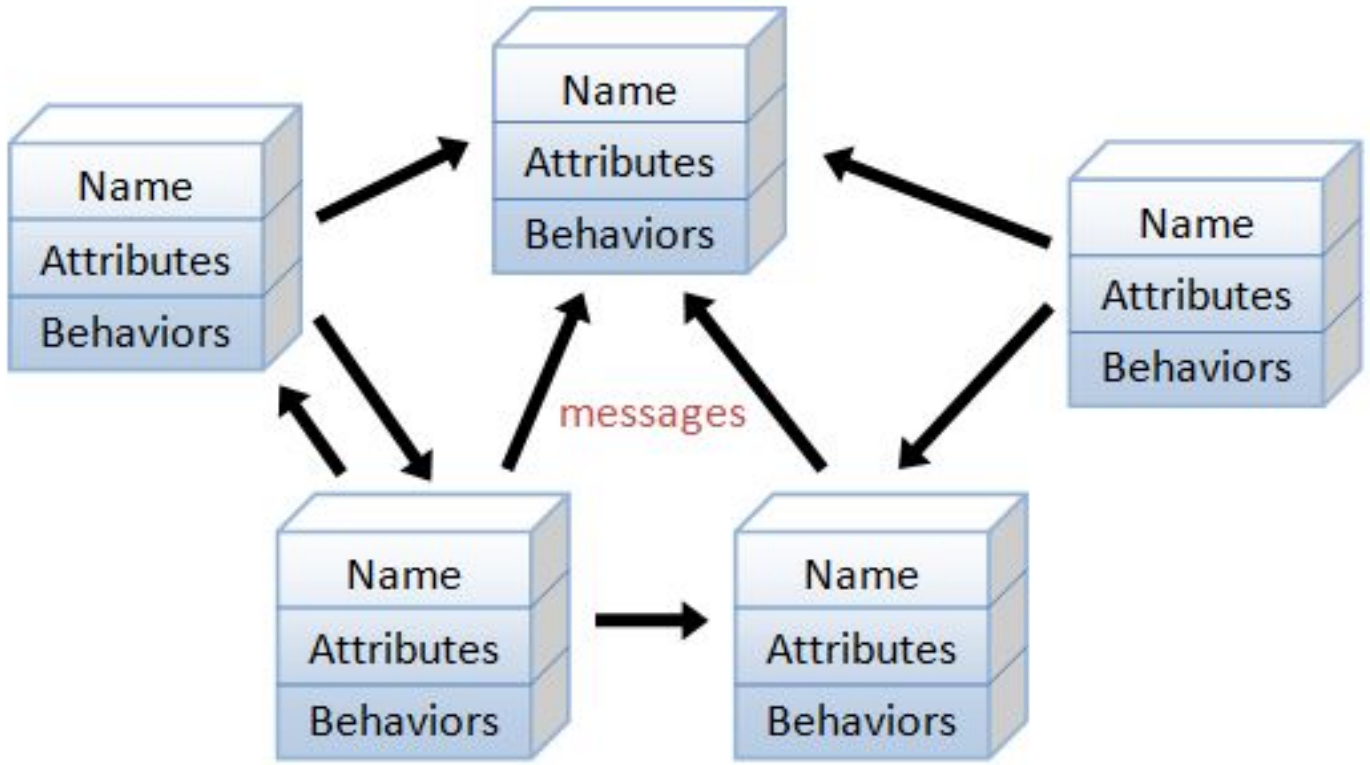
Compiled vs. Interpreted Languages

- Compiled languages are translated into machine code before execution (e.g., C++).
- Interpreted languages are executed line by line by an interpreter (e.g., Python, JavaScript).



Object-Oriented Programming (OOP)

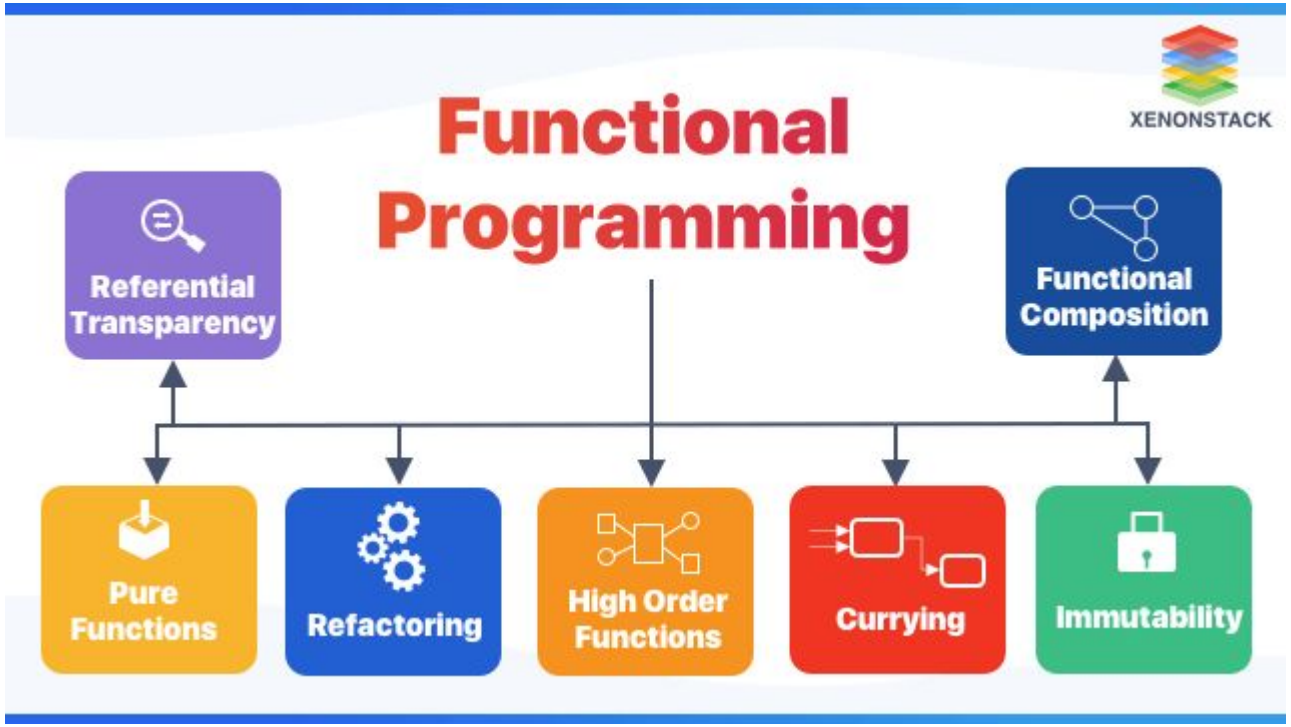
- OOP is a programming paradigm that organizes code around objects and classes.
- Examples of OOP languages: Java, C++, Python.



An object-oriented program consists of many well-encapsulated objects and interacting with each other by sending messages

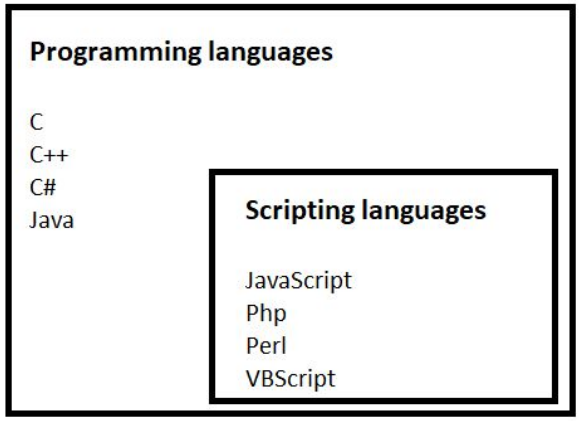
Functional Programming

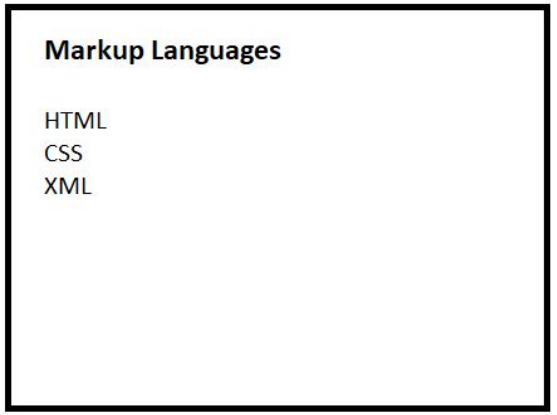
- Functional programming focuses on treating computation as the evaluation of mathematical functions.
- Examples of functional languages: Haskell, Lisp, Erlang.



Scripting Languages

- Scripting languages are used for automating tasks and rapid prototyping.
- Examples: Python, Ruby, Perl.

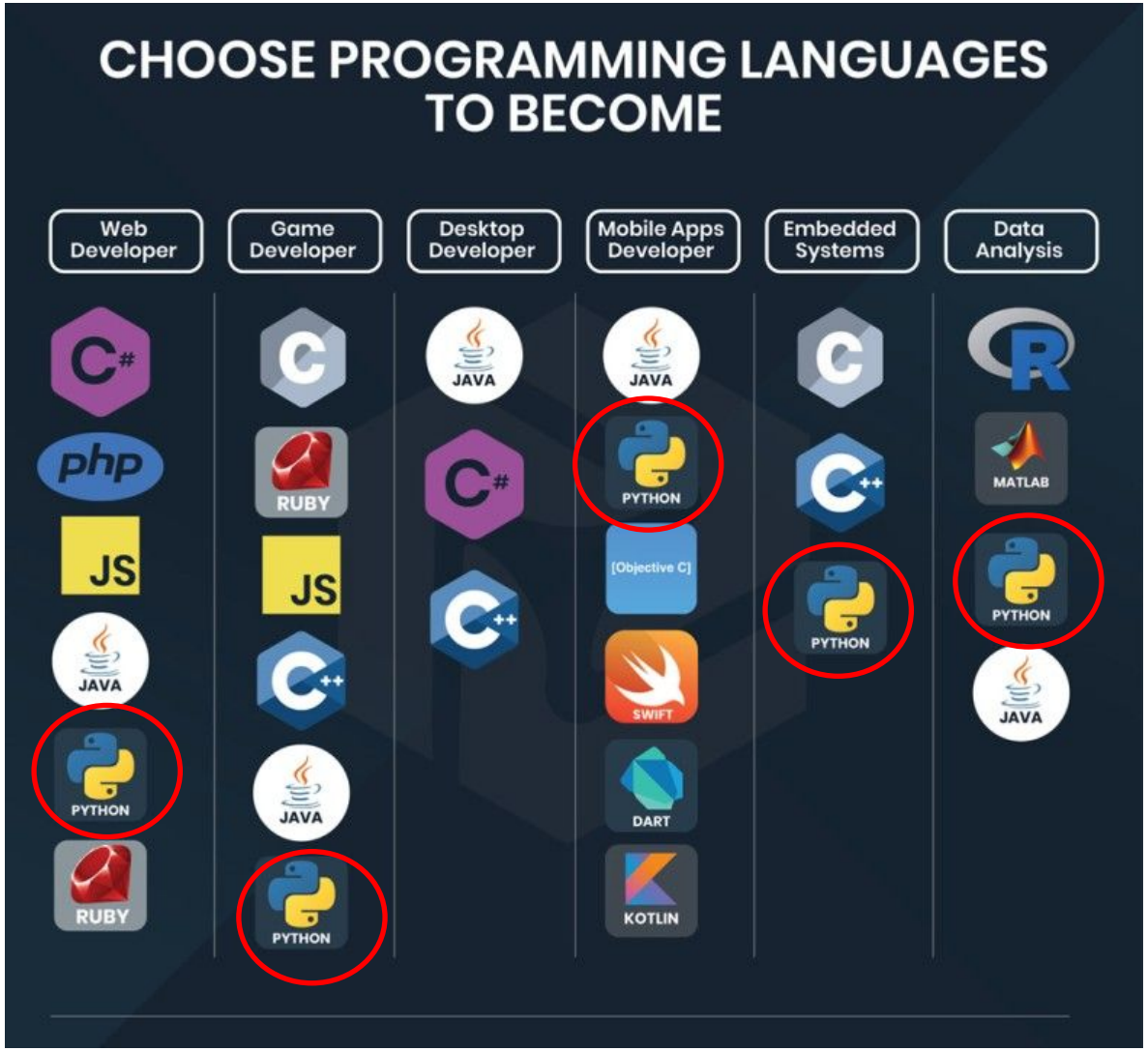




Scripting Languages	Programming Languages
Scripting language is a type of programming language designed for a special runtime environment to automate a specific function execution.	A programming language is a computer language which is used by humans to communicate with computers using a set of instructions.

Choosing a Programming Language

- Consider factors like project requirements, community support, ease of learning, and job market demand.
- No one-size-fits-all language; choose based on your goals and the problem you're solving.



Next time

- Lecture 3: Python Introduction