

# **Which Programming Language Should a Mechanical Engineer Learn**

# Computer Programming

$\pi$

Computer programming is the process of designing and building an executable computer program to accomplish a specific **computing result** or to perform a **specific task**.

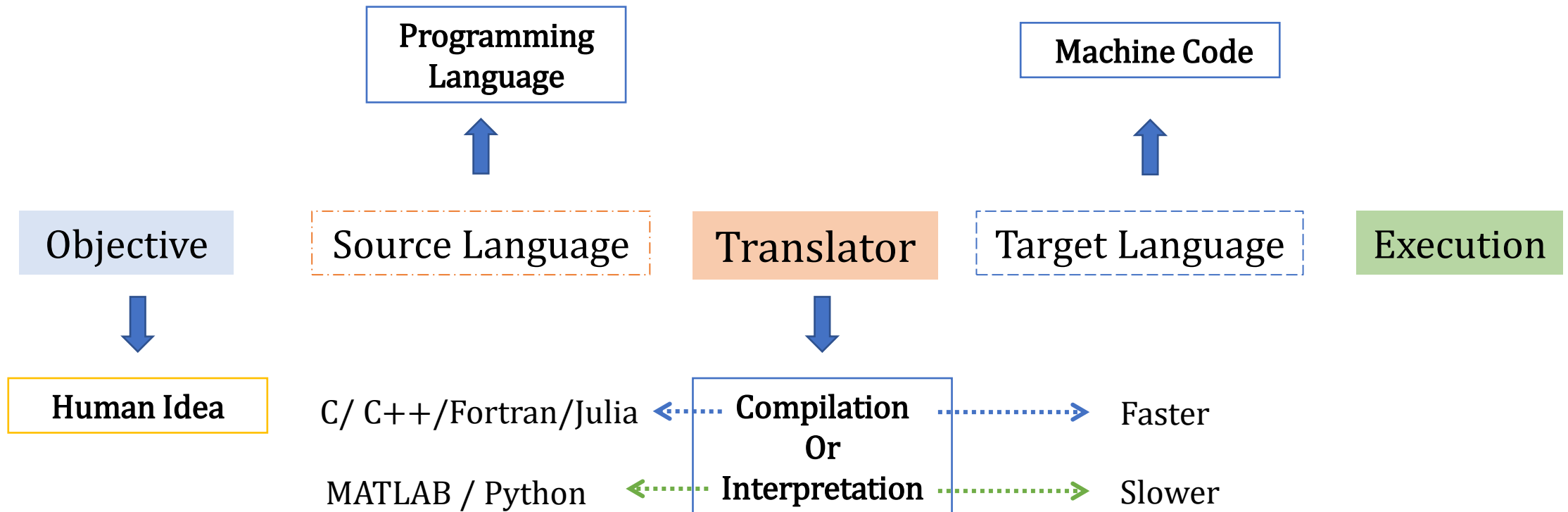
Programming involves tasks such as:

analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms in a chosen **programming language** (commonly referred to as coding)

A programming language

is a formal language comprising a set of instructions that produce various kinds of output. Programming languages are used in computer programming to implement algorithms.

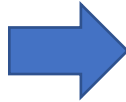
# Translation



# Programming language implementation

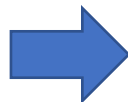
$\pi$

**Interpretation**



In computer science, an **interpreter** is a computer program that directly executes instructions written in a programming or scripting language, without requiring them previously to have been compiled into a machine language program

**Compilation**



In computing, a **compiler** is a computer program that translates computer code written in one programming language (the source language) into another language (the target language). The name "**compiler**" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program.

# High-level vs Low-level programming languages

$\pi$

## What are high-level languages?

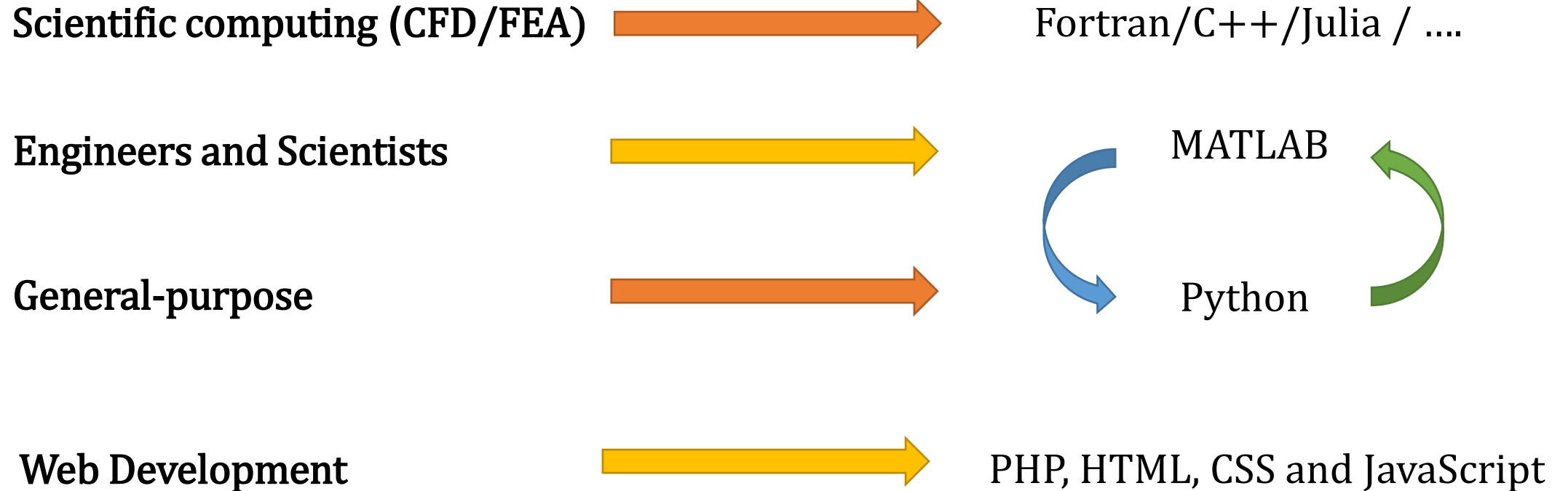
These are words and/or symbols that are used by programmers to write a 'source code.' They are readable and easy to comprehend. Some of the high-level languages include Java, PHP, and C++.

## What are low-level languages?

These are categorized into two – the **assembly language** and the **machine language**. The assembly language is harder to read than the high-level language. Although it is difficult to read, it contains basic instructions for the programmer. This language is then used to interpret and/or translate the code and turn it into a machine language – a language that contains a group of binary codes known to a computer's processing unit. This language cannot be read by humans; it wasn't designed that way.

# Prominent Programming Languages

$\pi$



If you are new to programming, then you should consider learning an "easier" language like **MATLAB** or **Python** before delving into something else.



# Fortran

Fortran is a **general-purpose**, compiled imperative programming language that is especially suited to **numeric computation** and **scientific computing**. Originally developed by IBM in the 1950s for **scientific** and **engineering** applications.

## High performance

Fortran has been designed from the ground-up for computationally intensive applications in science and engineering.

## Statically and strongly typed

Fortran is statically and strongly typed, which allows the compiler to catch many programming errors early on for you. This also allows the compiler to generate efficient binary code.

## Easy to learn and use

Fortran is a relatively small language that is surprisingly easy to learn and use. Expressing most mathematical and arithmetic operations over large arrays is as simple as you'd write them as equations on a whiteboard.

## Versatile

Fortran allows you to write code in a style that best fits your problem: Imperative, procedural, array-oriented, object-oriented, or functional.

## Natively parallel

Fortran is a natively parallel programming language with intuitive array-like syntax to communicate data between CPUs. You can run almost the same code on a single CPU, on a shared-memory multicore system, or on a distributed-memory HPC or cloud-based system. Coarrays, teams, events, and collective subroutines allow you to express different parallel programming patterns that best fit your problem at hand.



# C++

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes".

The language has expanded significantly over time, and modern C++ now has **object-oriented**, **generic**, and **functional** features in addition to facilities for low-level memory manipulation

Fast

Academic





# Julia

Julia is a high-level, high-performance, dynamic and compiled programming language. While it is a **general-purpose language** and can be used to write any application, many of its features are **well suited for numerical analysis and computational science**.

Julia has foreign function interfaces for C, Fortran, C++, Python, R, Java, and many other languages. Julia can also be embedded in other programs through its embedding API. Specifically, Python programs can call Julia using PyJulia.

General

Open source

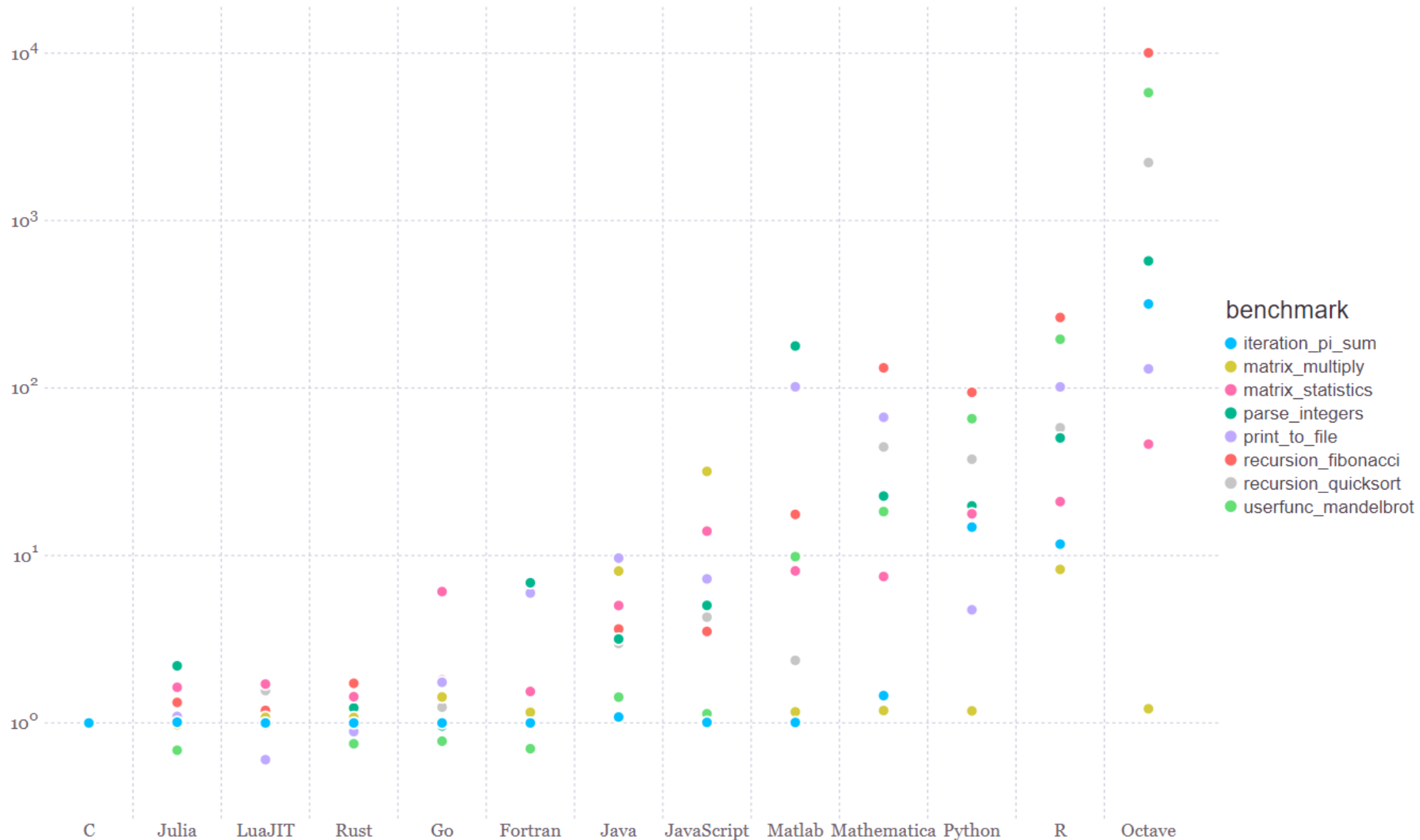
Fast

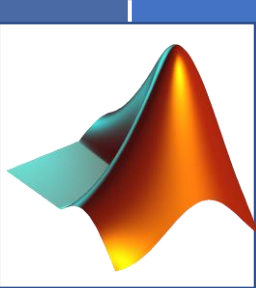
Dynamic

Composable

Reproducible

# Julia





# MATLAB

MATLAB is a programming and numeric computing platform used by millions of **engineers** and **scientists** to analyze data, develop algorithms, and create models.

MATLAB is probably the **most common language** that you will find mechanical engineers using

MATLAB is the programming language/environment of choice among mechanical engineers. MATLAB is essential for doing more complex numerical analysis where simple math doesn't cut it and a programming solution is required. MATLAB is especially well suited for calculations with **matrices** and **vectors**. It has an excellent plotting capability, so it is easy to run a simulation and then quickly display results. There are many additional libraries available for purchase for doing more complicated things such as optimization and control system design, among others.



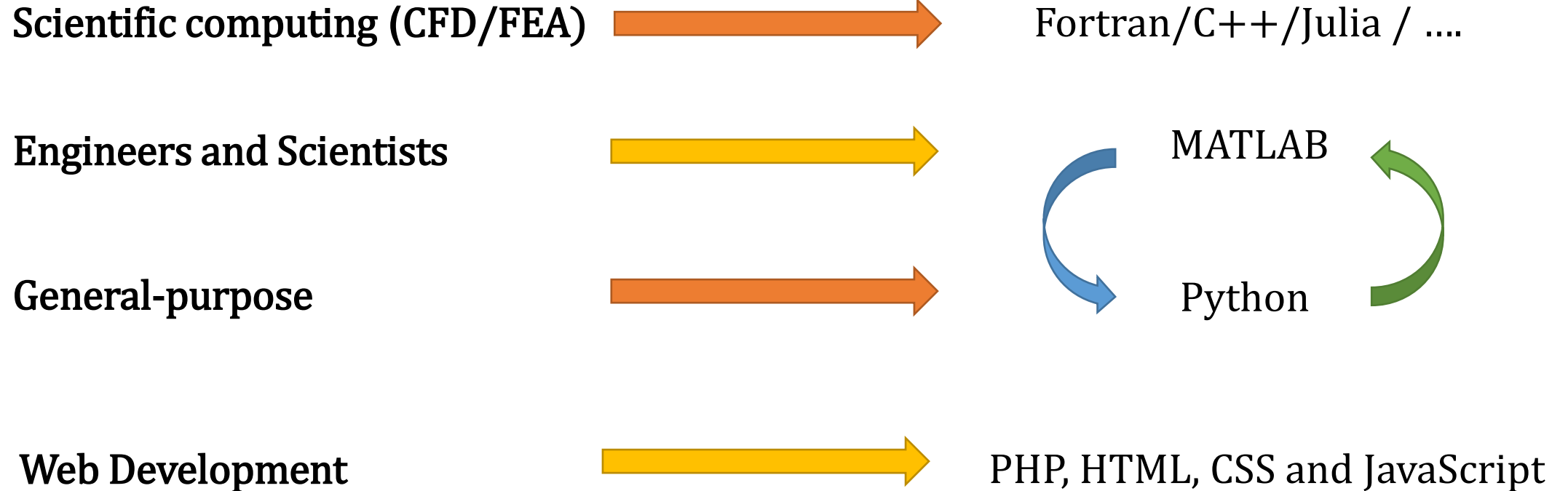
# Python

Python is a **general-purpose** programming language with a strong capability for **scientific programming**. Because of this, Python is an excellent alternative to MATLAB. Python is open-source, and many useful libraries are actively developed and maintained by the widespread Python community. Comparable functionality to MATLAB can be achieved by installing the NumPy, SciPy, and Matplotlib libraries. There are many other scientific libraries available such as Pandas that are worth looking into as well.

One major advantage of Python (besides the price of free) is that it is a general purpose programming language -- it is not specifically tailored to scientific programming, although it is excellent for that. This means that you can use it to do many other things besides numerical analysis, such as work with your file system, manage a database, or build a website.

# Programming Languages

$\pi$



If you are new to programming, then you should consider learning an "easier" language like **MATLAB** or **Python** before delving into something else.

# Summary

$\pi$

