2

How Programming Languages Work

Objectives

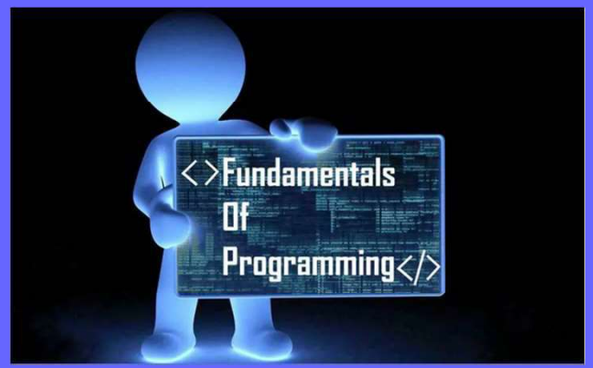
After completing this lesson, you should be able to:

- Definition of Program, Computer Programming, and Computer Programmer.
- Generations of Programming Language
- Types of Programming Language

Course Roadmap

Fundamentals of Programming

- Lesson 1: Introduction to Fundamentals of Programming
- **Lesson 2: How Programming Languages Work** — You are here!
- Lesson 3: Data and Types
- Lesson 4: Control Flow Statement Errors & Debugging
- Lesson 5: : Structured Modular & Object Oriented



Introduction

Computer Program

- A program is a set of instructions following the rules of the chosen language.
- Without programs, computers are useless.

- A program is like a recipe.

- It contains a list of ingredients (called variables) and a list of directions (called statements) that tell the computer what to do with the variables.

Elements of a program

- Knowing how a program runs and what data it relies on is an important first step toward understanding how to create your own programs.
- A single instruction in a program is called a ***Statement***. A statement usually has a character or line spacing that marks where the instruction ends, or terminates. How a program terminates varies with each language.

Programs are Data Driven

- Most programs rely on using data that's obtained from a user or another source, where statements might rely on such data to carry out instructions. Data can change how a program behaves, so programming languages come with a way to temporarily store data for later use.
- The data is stored in a statement called a ***Variable***. Variables are statements that instruct a device to save data in its memory. Variables in programs are similar to those in algebra, where they have a unique name and their value might change over time.

Flow control

- Some statements might not be executed by a device. This usually happens either by design, as written by the developer, or by accident, as the result of an unexpected error.

- Controlling the flow of an application makes it more robust and maintainable. Changes in control ordinarily occur when certain conditions are met. A common statement in modern programming languages, to control how a program is run, is the if...else statement.

Programming languages

- The principal purpose of programming languages is for developers to build instructions to send to a device.
- Programming languages are a vehicle for communication between humans and computers. Devices can understand only the binary characters 1 and 0. For most developers, using only binary characters isn't an efficient way to communicate.
- Programming languages come in a variety of formats and can serve different purposes. For example, JavaScript is used primarily for web applications, and Bash is used primarily for operating systems.

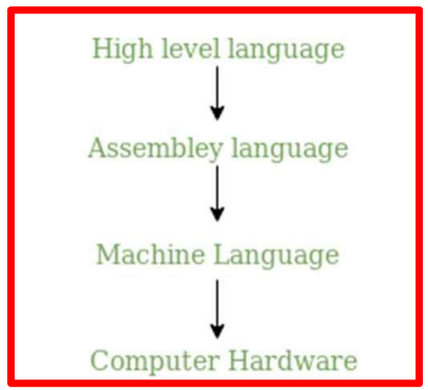
What is Programming - The language of Code

- Programming languages serve as a middle-man of sorts
- Translate your instructions into machine code
- Very useful for programmers

Low-level and high-level languages

- To be interpreted by a device, low-level languages typically require fewer steps than do high-level languages. However, what makes high-level languages popular is their readability and support.

- Each language also has an attribute known as ***Power Or Level***
- Basically how similar it is to machine code



Programming Language

- A vocabulary and set of grammatical rules (syntax) for instructing a computer to perform specific tasks.
- Programming languages can be used to create computer programs.
- The term programming language usually refers to high-level languages, such as BASIC, C, C++, COBOL, FORTRAN, Ada, Pascal, Java, C#, R, Ruby, Go, Swift, JavaScript and More.

How languages vary ?

- Each language is unqiue in how they operate
  - Java and Python - General Purpose languages
  - HTML/ CSS – Primarly used for designing Web UI.
- Each language also varies in how powerful it is

- You eventually need to convert your program into machine language so that the computer can understand it.
- There are two ways to do this:
  - Compile the program
  - Interpret the program

- **Compile** is to transform a program written in a high-level programming language from source code into object code.

- This can be done by using a tool called compiler.

- A compiler reads the whole source code and translates it into a complete machine code program to perform the required tasks which is output as a new file

- **Interpreter** is a program that executes instructions written in a high-level language.
- An interpreter reads the source code one instruction or line at a time, converts this line into machine code and executes it.

Computer Programming

- Computer programming is the process of writing, testing, debugging/troubleshooting, and maintaining the source code of computer programs.
- This source code is written in a programming language like C++, JAVA, Perl etc

Computer Programmer

- A programmer is someone who writes computer program.
- Computer programmers write, test, and maintain programs or software that tell the computer what to do.

What Skills are Required to Become a Programmer?

- **Programming -** Writing computer programs for various purposes.
- **Writing -** Communicating effectively with others in writing as indicated by the needs of the audience.
- **Reading Comprehension -** Understanding written sentences and paragraphs in work-related documents.
- **Critical Thinking -** Using logic and analysis to identify the strengths and weaknesses of different approaches.

What Skills are Required to Become a Programmer?

- **Computers and Electronics -** Knowledge of electric circuit boards, processors, chips, and computer hardware and software, including applications and programming.

- **Mathematics -** Knowledge of numbers, their operations, and interrelationships including arithmetic, algebra, geometry, calculus, statistics, and their applications.

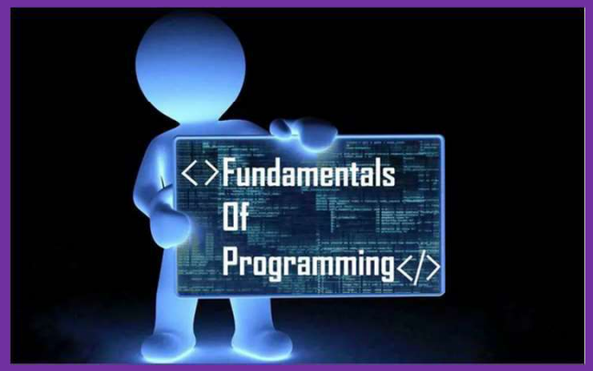
- **Oral Expression -** The ability to communicate information and ideas in speaking so others will understand.

Cont …

- **Oral Comprehension -** The ability to listen to and understand information and ideas presented through spoken words and sentences.

- **Written Expression -** The ability to communicate information and ideas in writing so others will understand.

- **Written Comprehension -** The ability to read and understand information and ideas presented in writing.



Generations of Programming Languges

Generations of Programming Language

- **The first generation** languages, or 1GL, are low-level languages that are machine language.
- **The second generation** languages, or 2GL, are also low-level languages that generally consist of assembly languages.
- **The third generation** languages, or 3GL, are high-level languages such as C

- **The fourth generation** languages, or 4GL, are languages that consist of statements similar to statements in a human language. Fourth generation languages are commonly used in database programming and scripts.

- **The fifth generation** languages, or 5GL, are programming languages that contain visual tools to help develop a program. A good example of a fifth generation language is Visual Basic.

Types of Programming Language

- There are three types of programming language:
  - **Machine language** (Low-level language)
  - **Assembly language** (Low-level language)
  - **High-level language**
- Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.

Machine Language

- Machine language is a collection of binary digits or bits that the computer reads and interprets.
- Machine languages are the only languages understood by computers.
- While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.
  - **Machine Language**
    169 1 160 0 153 0 128 153 0 129 153 130 153 0 131 200 208 241 96
  - **High level language**
    5 FOR I=1 TO 1000: PRINT "A";: NEXT I

Machine Language

Example:

- Let us say that an electric toothbrush has a processor and main memory.
- The processor can rotate the bristles left and right, and can check the on/off switch.
- The machine instructions are one byte long, and correspond to the following machine operations:

Machine Instruction	Machine Operation
0000 0000	Stop
0000 0001	Rotate bristles left
0000 0010	Rotate bristles right
0000 0100	Go back to start of program
0000 1000	Skip next instruction if switch is off

Assembly Language

- A program written in assembly language consists of a series of instructions mnemonics that correspond to a stream of executable instructions, when translated by an assembler, that can be loaded into memory and executed.

- **Assembly languages** use keywords and symbols, much like English, to form a programming language but at the same time introduce a new problem.

- The problem is that the computer doesn't understand the assembly code, so we need a way to convert it to machine code, which the computer does understand.
- Assembly language programs are translated into machine language by a program called an **Assembler**.

Assembly Language

Example:

- Machine language :
  - 10110000 01100001
- Assembly language :
  - mov a1, #061h
- Meaning:

  Move the hexadecimal value 61 (97 decimal) into the processor register named "a1".

High Level Language

- High-level languages allow us to write computer code using instructions resembling everyday spoken language (for example: print, if, while) which are then translated into machine language to be executed.

- Programs written in a high-level language need to be translated into machine language before they can be executed.

- Some programming languages use a compiler to perform this translation and others use an interpreter.

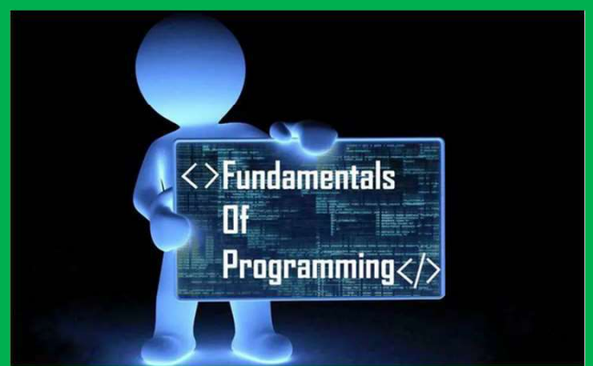
High-Level Language

Examples of High-level Language:

1. ADA
2. BASIC
3. COBOL
4. PASCAL
5. C
6. C++
7. JAVA
8. C#
9. PHYTON …

Comparisson

	Machine Language	Assembly Language	High-level Languages
Time to execute	Since it is the basic language of the computer, it does not require any translation, and hence ensures better machine efficiency. This means the programs run faster.	A program called an 'assembler' is required to convert the program into machine language. Thus, it takes longer to execute than a machine language program.	A program called a compiler or interpreter is required to convert the program into machine language. Thus, it takes more time for a computer to execute.
Time to develop	Needs a lot of skill, as instructions are very lengthy and complex. Thus, it takes more time to program.	Simpler to use than machine language, though instruction codes must be memorized. It takes less time to develop programs as compared to machine language.	Easiest to use. Takes less time to develop programs and, hence, ensures better program efficiency.



Programming

Whats the syntax?

- Syntax
  - Rules you must follow if you want your program to run correctly
  1. How you type out certain functions
  2. What you put at the end of each line of code
  3. How you set up certain functions
- Syntax for each programming language is unique
- Breaking programming rules will result in an error.

How to write code: Variable Example

- As an example lets initialize a variable in Java , Python, and Javascript

```
int variable = 3;
```

java

```
x = 3;
```

python

```
var x = 3;
```

javascript

Built in Error checks

- Another cool thing about IDE's is that they will let you know if and when there are syntax errors in your code

Tools of the trade

- The ability to develop code fast is crucial. Having a tool to do so helps you not only with speed, but it can usually help you ***With Formatting And Correctness As Well.***
- **Editors**
  - One of the most crucial tools for software development is your editing environment. An editor is where you write your code and sometimes where you run your code.
  - Developers rely on editors for their helpful features
    - Debugging
    - Syntax Highlighting
    - Extensions & Integrations
    - Customizations

Command-line tools

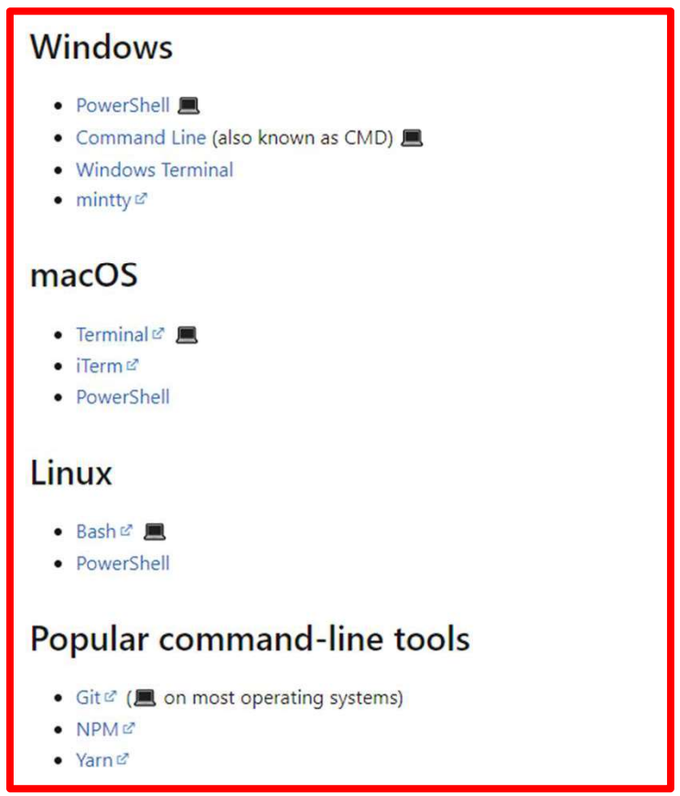
- As a developer, you're likely to use command-line tools to accomplish some or all of your programming tasks.
- Because development environments are unique to each developer, some avoid using the command line, some rely on it exclusively, and some prefer a mix of the two.

Why command-line tools?

- The command line, compared to a graphical user interface, has no graphical elements and is primarily text based. The reasons for using the command line are many:
  - **Preference**: Some developers prefer a less graphical view for their daily programming tasks.
  - **Better workflow**: Developing code requires a significant amount of typing, and some developers prefer not to disrupt their flow on the keyboard. They use keyboard shortcuts to swap between desktop windows, work on various files, and open tools.
  - **Avoiding mouse arm syndrome**: Most tasks can be completed with a mouse, but one benefit of using the command line is that a lot can be done with command-line tools without having to constantly switch from mouse to keyboard and back again.
  - **Configurability**: With command-line tools, you can save your custom configuration, change it later, and import it to other development machines.

Command line options

- Your command-line tool options differ depending on the operating system you use. The computer icon (💻) indicates that the command-line tools come preinstalled on the operating system.



Windows

- PowerShell 💻
- Command Line (also known as CMD) 💻
- Windows Terminal
- mintty

macOS

- Terminal 💻
- iTerm
- PowerShell

Linux

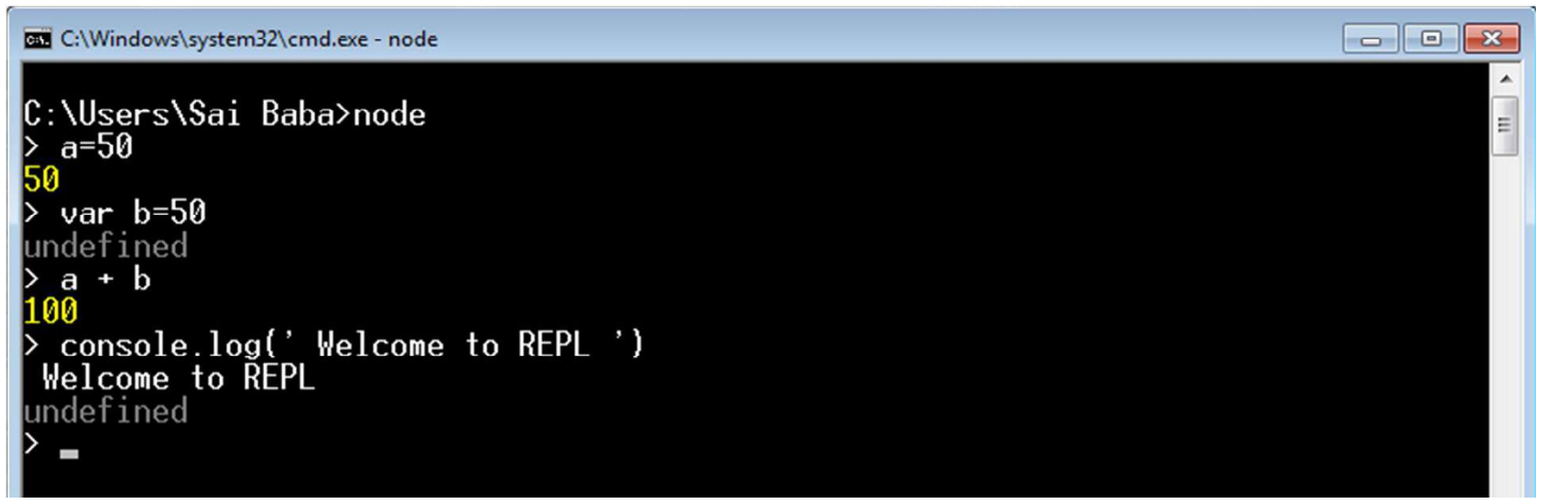
- Bash 💻
- PowerShell

Popular command-line tools

- Git (💻 on most operating systems)
- NPM
- Yarn

How do we get information from a computer: The Console

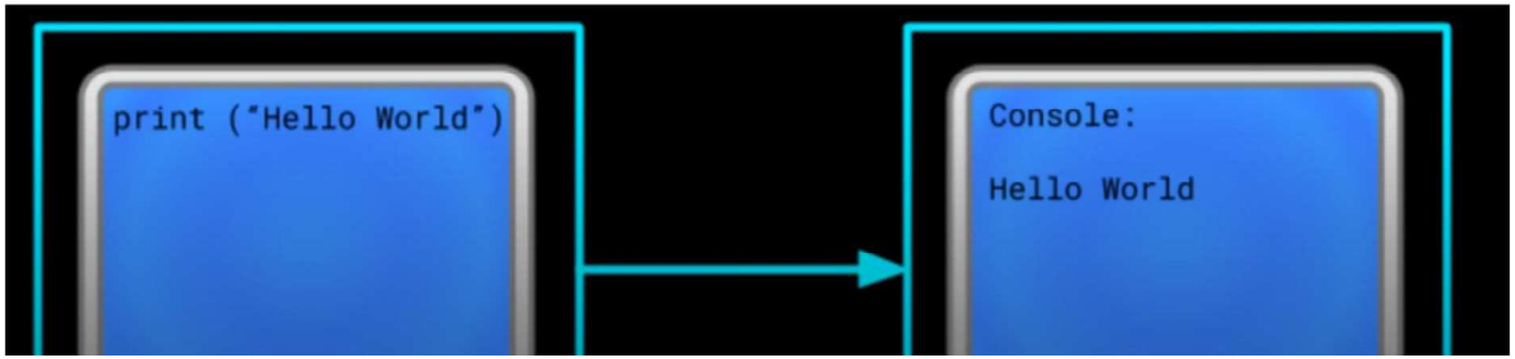
- Programmers keep track of their progress by looking at the console



- The main use of the console is to output text from the program using the code
- More specifically a print statement

How do we get information from a computer: Using the print statement

- To use the print statement, simply instruct the console to print, and then include whatever you want to print inside the parenthesis



How do we get information from a computer: Using the print statement

- print statement is also used for viewing and interpreting the computer's output of a program
  - Example print (4+3), will display 7 output in the console

Need of Console

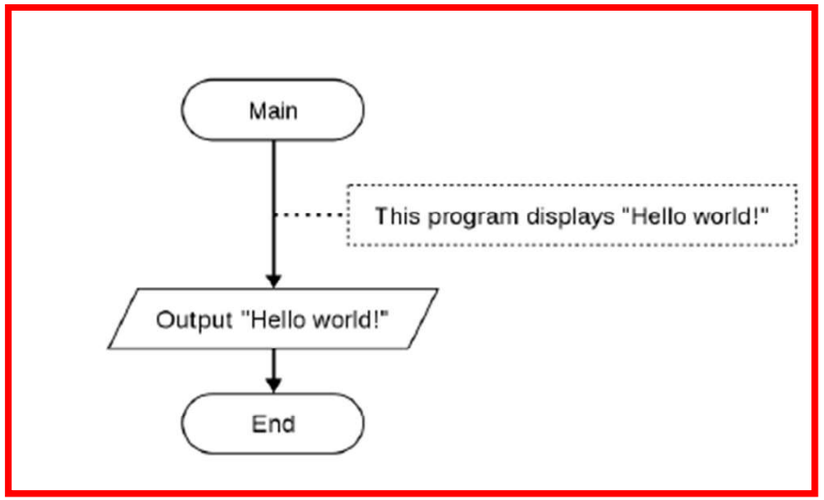
- The console is mainly a ***Developer tool***
- Not usually meant to be used and interpreted with by the ***End User*** but is useful for the ***Developer*** to view the logs

Hello World

- A “**Hello, world**!” program is a computer program that outputs or displays “Hello, world!” to a user.
- Being a very simple program in most programming languages, it is often used to illustrate the basic syntax of a programming language for a working program, and as such is often the very first program people write.

Pseudocode

```
Function Main
    ... This program displays "Hello world!"
    Output "Hello world!"
End
```



BASIC

- Short for **Beginner's All-purpose Symbolic Instruction Code**.
- Developed in the 1950s for teaching University students to program and provided with every self-respecting personal computer in the 1980s,
- BASIC has been the first programming language for many programmers.
- It is also the foundation for Visual Basic.

BASIC

Example:

```
PRINT "Hello world!"
```

Pascal

- A high-level programming language developed by Niklaus Wirth in the late 1960s.
- The language is named after Blaise Pascal, a seventeenth-century French mathematician who constructed one of the first mechanical adding machines.
- It is a popular teaching language.

Pascal

Example:

```
Program HelloWorld(output);
begin
        writeLn('Hello, World!')
end.
```

Visual Basic

- A programming language and environment developed by Microsoft.
- Based on the BASIC language, Visual Basic was one of the first products to provide a graphical programming environment and a paint metaphor for developing user interfaces.

Visual Basic

Example:

```
MsgBox "Hello, World!“
```

C

- Developed by Dennis Ritchie at Bell Labs in the mid 1970s.
- C is much closer to assembly language than are most other high-level languages.
- The first major program written in C was the UNIX operating system.
- The low-level nature of C, however, can make the language difficult to use for some types of applications.

C

Example:

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
    return 0;
}
```

C++

- A high-level programming language developed by Bjarne Stroustrup at Bell Labs.
- C++ adds object-oriented features to its predecessor, C.
- C++ is one of the most popular programming language for graphical applications, such as those that run in Windows and Macintosh environments.

C++

Example:

```
#include <iostream>

int main()
{
        std::cout << "Hello World!" << std::endl;
        return 0;
}
```

Java

- A high-level programming language developed by Sun Microsystems.
- Java was originally called *OAK,*and was designed for handheld devices and set-top boxes.
- Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web.
- Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web.

Java

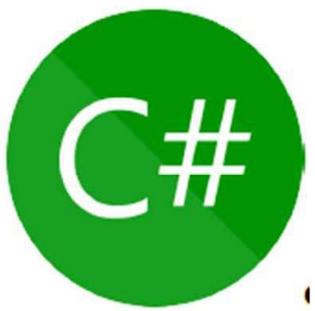
Example:

```
/* * Outputs "Hello, World!" and then exits */

public class HelloWorld {
        public static void main(String[] args) {
                System.out.println("Hello, World!");
        }
}
```

- **JavaScript**, often abbreviated as JS, is a high-level, interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web.
- JavaScript enables interactive web pages and therefore is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it


```
// This script displays "Hello world!".
//
// References:
// https://www.digitalocean.com/community/tutorials/how-to-write-your

console.log("Hello world!")
```

- C# is a general-purpose, object-oriented programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines.
- It was developed around 2000 by Microsoft within its .NET initiative and later approved as a standard by Ecma




```
// This program displays "Hello world!"
//
// References:
// https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/i

public class Hello
{
    public static void Main()
    {
        System.Console.WriteLine("Hello world!");
    }
}
```

- **Python** is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991,
- Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales


```
# This program displays "Hello world!"
#
# References:
# https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Pytho

print("Hello world!")
```

Choosing a Programming Language

- Before you decide on what language to use, you should consider the following:
- Your server platform
- The server software you run
- Your budget
- Previous experience in programming
- The database you have chosen for your backend

Developer documentation

- When developers want to learn something new, they'll most likely turn to expert documentation. Expert documentation can guide them through how to use programming tools and languages properly, and help them gain deeper knowledge about how it all fits together.

Popular Development documentation

- The following resources are just two examples of documentation for developers:
- Mozilla Developer Network
- https://devdocs.io
- Every Language has its own Documentations

Summary

In this lesson, you should have learned that:

- Definition of Program, Computer Programming, and Computer Programmer.
- Generations of Programming Language
- Types of Programming Language