

Tevin Bristol

Front-End Languages

HTML (Hypertext Markup Language):

Functionality: HTML is used for creating the structure and content of web pages.

Advantages: Simple, widely supported, easy to learn, semantic markup for content.

Shortcomings: Limited interactivity, lacks dynamic behavior.

JavaScript:

Functionality: JavaScript is a versatile scripting language for adding interactivity, manipulating DOM, and controlling behavior on the client side.

Advantages: High interactivity, asynchronous operations, large ecosystem (libraries, frameworks).

Shortcomings: It can have browser compatibility issues, and its single-threaded nature can lead to performance bottlenecks.

Back-end Languages

Python:

Functionality: Python is a versatile language used for back-end development, known for its readability and wide range of libraries.

Advantages: Clean syntax, extensive standard library, used in data science and AI.

Shortcomings: Not as fast as compiled languages, not ideal for low-level system programming.

Kotlin:

Functionality: Kotlin is a modern JVM-based language suitable for front-end (via Kotlin/JS) and back-end development.

Advantages: Concise, interoperable with Java, null safety, modern features.

Shortcomings: Smaller community compared to Java.

Java:

Functionality: Java is a widely used, robust, and platform-independent language, often used for enterprise-level applications.

Advantages: Strong type system, large ecosystem, excellent for large-scale applications.

Shortcomings: Verbosity is slower to develop compared to some other languages.

PHP:

Functionality: PHP is a server-side scripting language primarily used to develop and create dynamic web pages.

Advantages: Designed for web, extensive built-in functions, large community.

Shortcomings: Historically criticized for inconsistent function names, not as suitable for non-web tasks.

New Language (Rust for Back-end Development):

Functionality: Rust is a systems programming language emphasizing safety, performance, and low-level control.

Advantages: Memory safety without garbage collection, high performance, strong typing, suitable for system-level programming.

Shortcomings: Steeper learning curve, less mature ecosystem than languages like C++.

Language	Functionality (Front-End)	Advantages (Front-End)	Shortcomings (Front-End)	Functionality (Back-end)	Advantages (Back-end)	Shortcomings (Back-end)
HTML	Structure & Content	Simplicity, Widely Supported	Limited Interactivity	N/A	N/A	N/A
JavaScript	Interactivity	High Interactivity	Browser Compatibility, Performance	N/A	N/A	N/A
Python	Back-end Development	Clean Syntax, Libraries	Slower than Compiled Languages	Back-end Development	Readability, Extensive Standard Library	Not Ideal for Low-Level System Programming
Kotlin	Back-end Development	Conciseness, Interoperability	Smaller Community	Front-End/Back-end Development	Modern Features, Null Safety	Smaller Community
Java	N/A	N/A	N/A	Back-end Development	Robustness, Platform Independence	Verbosity, Slower Development
PHP	N/A	N/A	N/A	Back-end Development	Web Focus, Extensive Built-in Functions	Historically Criticized Syntax
Rust	N/A	N/A	N/A	Back-end Development	Memory Safety, High Performance	Steeper Learning Curve, Less Mature Ecosystem