|  |  |
| --- | --- |
| Client-side Language ex. JavaScript | Server-side Language ex. PHP, node.js |
| (+) No additional component is needed and can be accessed via a web browser |  |
| (+) Easy to verify forms |  |
| (-) Bad Security |  |
| (-) Not good for heavy computation or managing other devices |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | JavaScript | PHP | Node.js |
| Pros | (+) No additional component is needed and can be accessed via a web browser  (+) Easy to verify forms  (-) Bad Security  (-) Not good for heavy computation or managing other devices | * Extendable (multiple server) * Interface (Database System, Libraries) * Object -oriented programming * Cross-platform | * Non-blocking: the server never waits for an API to return data   + Very fast(No buffering)   + Single-threaded (PHP와 다름) * Cross-platform |
| when? |  | * Mixed use of HTML, CSS, database * Easy to code(no need for compilers, interpreters) * Database Management * Heavy CPU computation | * I/O bound application * Data streaming application * Data Intensive real-time application * JSON APIs based Application * Single page application |
| reading or writing of files | X | O | O |
| networking applications | X | O |  |
| multi-threading/multiprocessor | single-threaded (never blocks and everything runs concurrently) | multi-threaded(blocks I/O to carry out multiple tasks concurrently) | X (single threaded but highly scalable) |
|  | code can be viewed even after the output is interpreted  combined with HTML, AJAX, XML  used with HTML and CSS | only be viewed after it is interpreted by the server  HTML only  X |  |