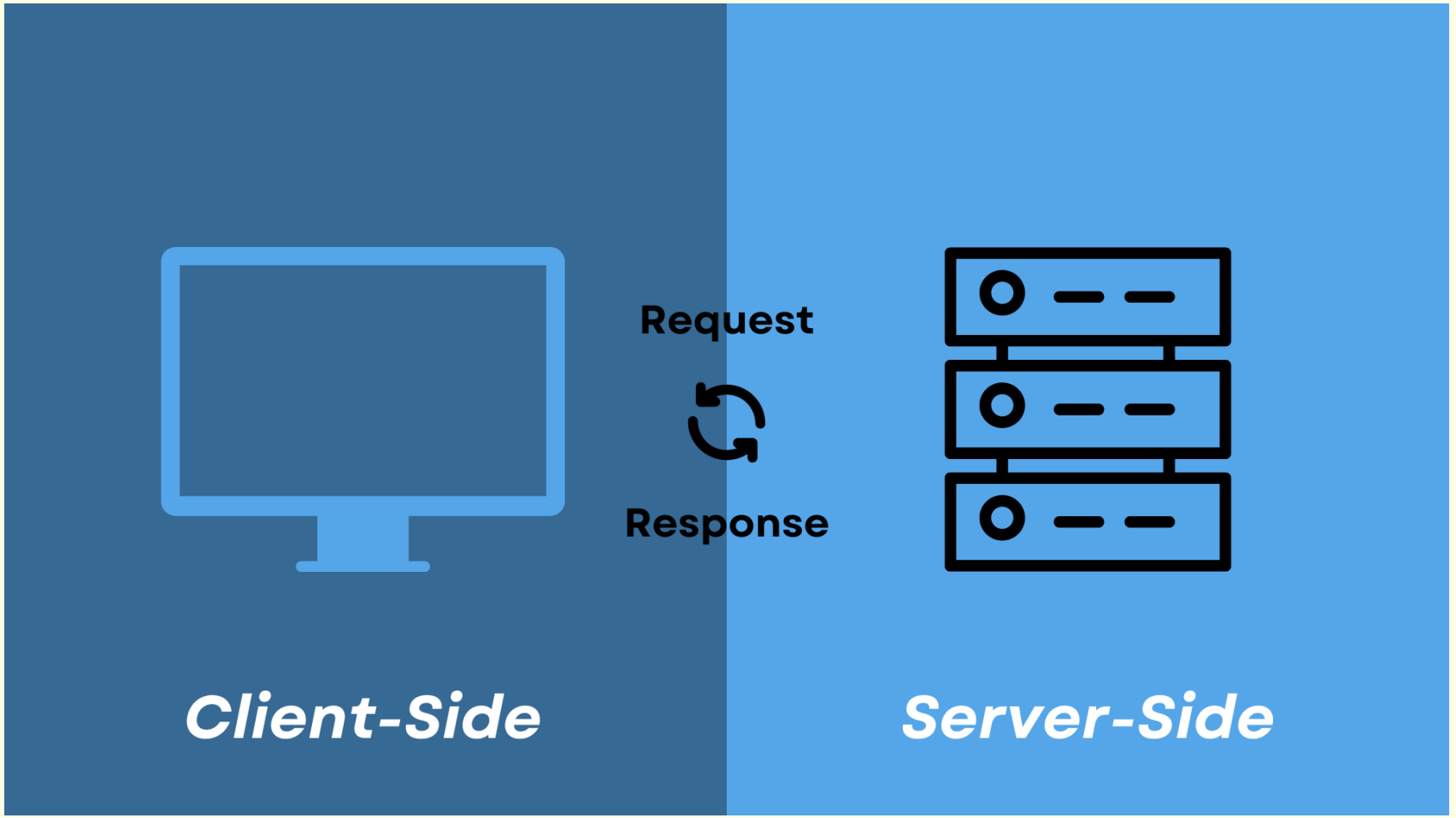
Learning Aims

- Server and client side processing

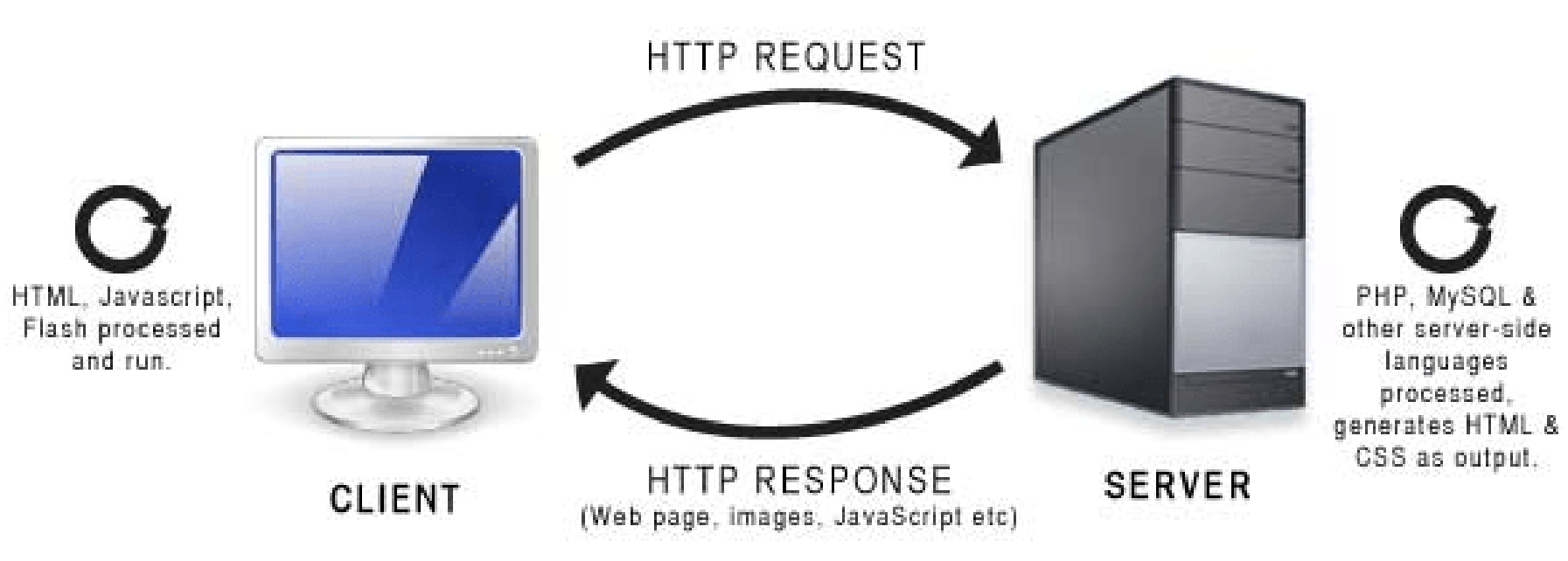


Key terms

Keyword	Definition
Client-Side Processing	When code runs in the user's browser (e.g., JavaScript) after a web page is loaded. It controls interactivity, animations, and form validation.
Server-Side Processing	When code runs on the web server (e.g., PHP, Python, or ASP) before the page is sent to the browser. It handles databases, authentication, and dynamic content creation.
Advantages of Client-Side	Reduces server load, faster user interactions, and allows offline functionality.
Disadvantages of Client-Side	Depends on the user's device performance; less secure (code visible to users).
Advantages of Server-Side	More secure; consistent processing regardless of client device; can handle databases and large computations.
Disadvantages of Server-Side	Can be slower for users; increases server workload and hosting costs.
Dynamic Web Page	A page that changes content based on user input or other factors, often using scripts on the client or server side.
Static Web Page	A fixed page that looks the same for every visitor, with no interactivity or dynamic updates.

Server and client side processing

- Web pages are usually stored on servers and viewed through the Internet.
- Some pages have fixed, unchanging ("static") information.
- Dynamic web pages can show changing or personalised content.
- Adding dynamic features carefully can make the site more engaging.
- Scripts make pages dynamic — **server-side scripts** run on the server, while **client-side scripts** run in the user's browser.



Describe the steps of requesting a web page

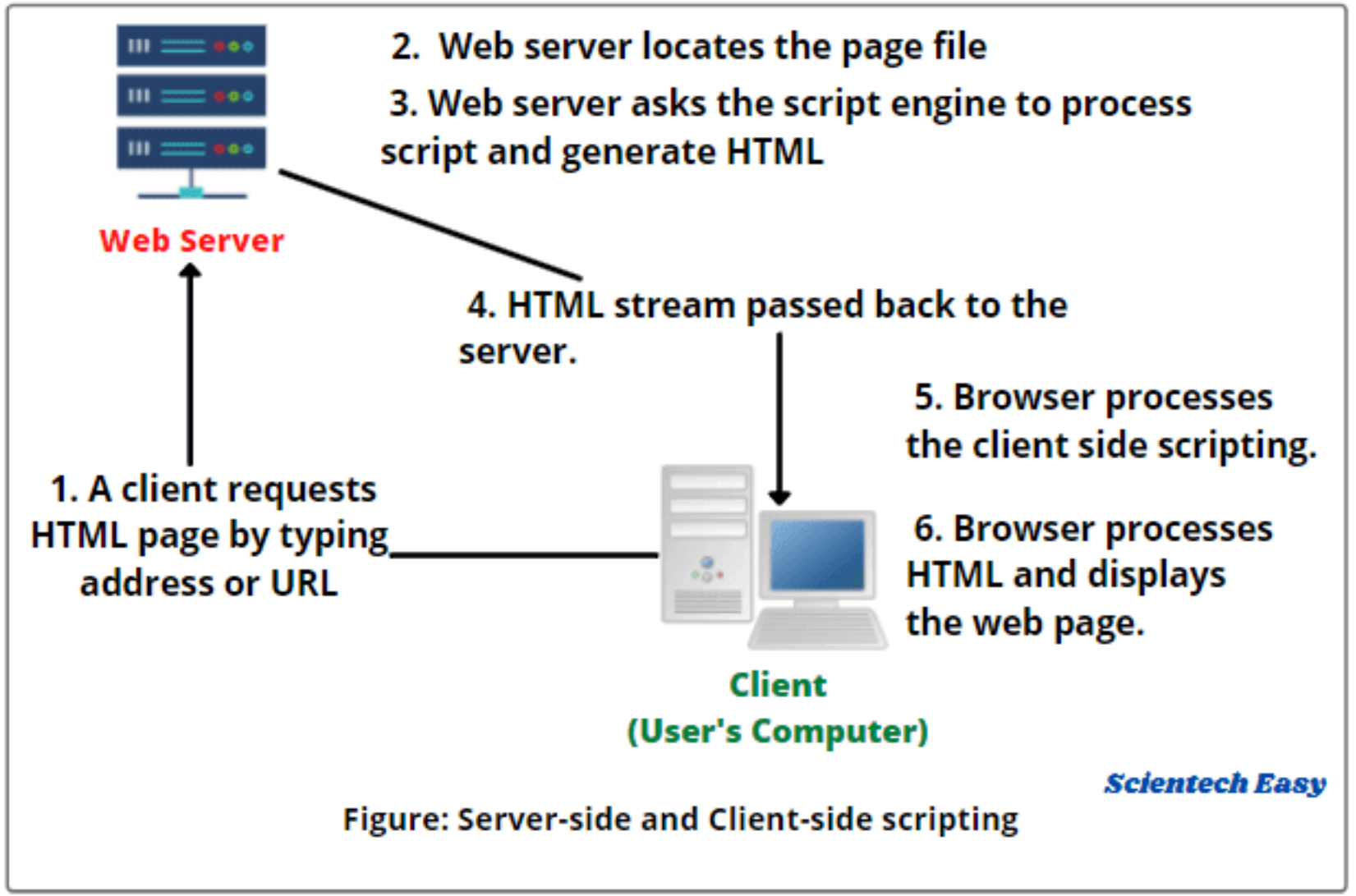
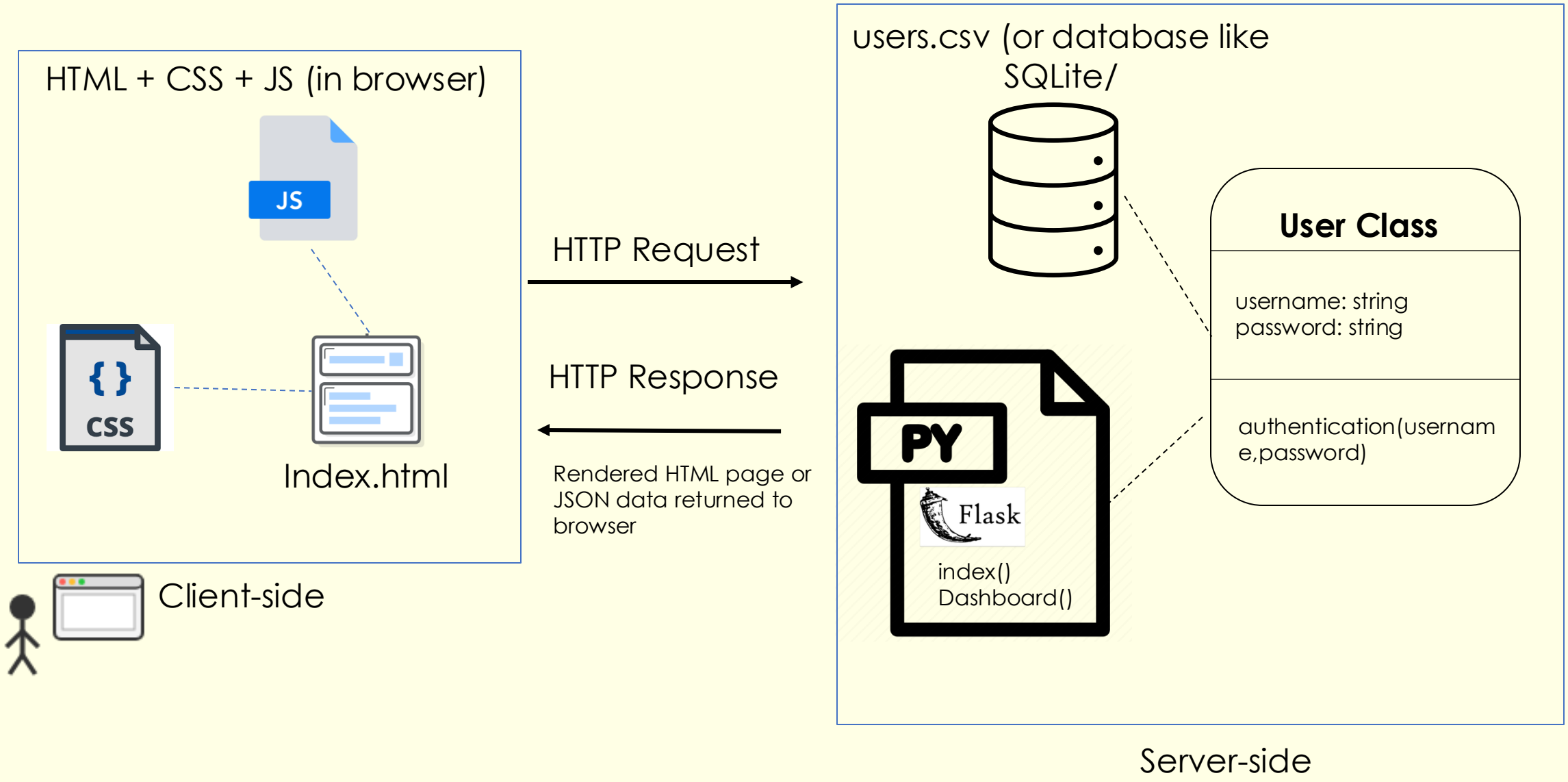


Figure: Server-side and Client-side scripting

Server-side scripting

- Server-side scripting stores **scripts** on the server.
- When a page is requested, the server runs the scripts and sends the finished HTML to the browser.
- All data and processing happen on the server, often using a **database**.
- It's secure and efficient because the user only receives the needed information.
- Common server-side languages include Flask (python), Perl, ASP, PHP.



User input → HTTP Request → Flask handles request → Database query → HTTP Response

Client-side scripting

- Many web hosts limit or block server-side scripts, so designers often use client-side scripts instead.
- The browser (client) runs these scripts and displays the results.
- The main language for client-side scripting is JavaScript.
- JavaScript code is placed inside HTML and can:
  - show dialogue boxes
  - accept user input
  - animate images
  - change how pages look
  - redirect pages or change pictures

You can see an example of JavaScript code and its effects from the www.w3schools.com website here:

JavaScript example: https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_lightbulb

Pros and Cons

Both methods are important and often used together.

Client-side is best for speed and responsiveness.

Server-side is best for security and data management.

Good practice: validate input both on the client (for speed) and server (for security).

Client-side processing	**Server-side processing**
Advantages: • Faster response as data doesn't need to be sent to the server. • Reduces server load. • Useful for interactive features like games or form validation.	**Advantages:** • Secure – data is processed on the server, keeping it hidden from users. • Can access databases and handle sensitive information safely. • Doesn't depend on browser capabilities.
Drawbacks: • Code is visible to the user – can be copied or tampered with. • May not run if the browser doesn't support it or if JavaScript is disabled. • Cannot securely access or process sensitive data.	**Drawbacks:** • Slower – data has to travel between client and server. • Increases load on the server, which can raise hosting costs.