**JavaScript Interview Questions**

**1. General**

**1.1. What are Higher Order Functions?**

* JavaScript functions that take in or return a function.
* Eg. map(), filter()

**1.2. What are Arrow Functions?**

* Introduced in ES6 and are more precise way of writing functions.
* Key differences from normal functions:
  + Removal of function keyword.
  + If only consist of a return statement following a single line of code, we can remove the curly braces and remove the return keyword.
  + If function has only one argument, we can remove the parenthesis.
  + Cannot use arrow functions as object constructors.
  + Eg. const Car = (color) => { this.color = color;};

**2. HTML**

**2.1. What are Semantic HTML tags?**

* Tags that define the meaning of the content they contain.
* Eg. <header>, <article>, and <footer>
* On the other hand, tags like <div> and <span> are typical examples of non-semantic HTML elements.

**3. Rendering**

**3.1. Server-Side Rendering (SSR) vs. Client-Side Rendering (CSR).**

* SSR: The server generates the full HTML (+CSS and JS functions) for a web page and sends it to the client's browser. The browser then renders the HTML to display the content to the user.
* CSR: the browser downloads a minimal HTML shell and then uses JavaScript to build and render the content dynamically on the client side.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Server-Side Rendering (SSR)** | **Client-Side Rendering (CSR)** |
| Initial Load Time | Faster | Slower |
| Subsequent Navigation | Slower (full page reloads) | Faster (partial updates, no full reloads) |
| SEO | Better (pre-rendered HTML) | Challenging (requires additional setup like pre-rendering or server-side rendering for critical pages) |
| Server Load | Higher (renders HTML on server) | Lower (renders HTML on client) |
| Interactivity | Less dynamic (requires full reloads) | More dynamic (single-page application feel) |
| Browser Compatibility | Better (less reliance on JavaScript) | Relies heavily on JavaScript |
| Frameworks | Next.js, Ruby on Rails | React.js, Vue.js, Angular |

**3.2. In what scenarios would you prefer Client-Side Rendering over Server-Side Rendering?**

* Web pages requiring dynamic page generation eg. online quiz.
* Advanced UI Components: Applications that utilize sophisticated UI components (e.g., drag-and-drop, animations, real-time data visualization) are well-suited for CSR since it allows for more dynamic and responsive interfaces.

**4. HTTP**

**4.1. List the main HTTP request methods.**

|  |  |  |
| --- | --- | --- |
| **Methods** | **Purpose** | **Characteristics\*** |
| **GET** | Retrieve data from the server. | Safe, Idempotent |
| **POST** | Submit data to the server to create a new resource. | Not Idempotent |
| **PUT** | Update an existing resource or create a new resource if it does not exist. | Idempotent |
| **PATCH** | Partially update an existing resource. | Not Necessarily Idempotent |
| **DELETE** | Remove a resource. | Idempotent |
| **HEAD** | Get the headers of a resource, like GET but without the response body. | Safe, Idempotent |

\*Safe = does not alter the resource

\*Idempotent = same request with the same data will produce the same result