**W02 Prepare – Q/A AI**

**JavaScript Variables**

**Q: How do you declare and use variables in JavaScript?**

A: You can declare variables using let, const, or var. It's generally recommended to use let and const for better scoping and to avoid var due to potential pitfalls. Variables act like named storage containers for data in your program.

**Q: What are scoping issues with JavaScript variables?**

A: Scoping issues arise from how variables are accessible in different parts of your code. var variables are scoped to the function they are declared in, which can lead to unexpected behavior. let and const introduce block scope, making variables more predictable.

**Q: When to use let, const, and var?**

A: Use const for fixed values like pi (3.14159). Use let for variables that need to be reassigned, like a player's score in a game. Avoid var in new code due to scoping issues.

**Q: Why do variables need to be declared?**

A: Declaring variables allocates memory and improves code clarity. It tells the program what kind of data to expect and helps prevent errors.

**Q: Best ways to organize JavaScript code for variables?**

A: Use meaningful names, leverage scoping with let and const, group related variables, and consider modules/namespaces for larger projects. Utilize destructuring (optional) and comments effectively.

**JavaScript Functions**

**Q: What are the purposes of functions in JavaScript?**

A: Functions in JavaScript serve several purposes:

* **Code Reusability:** Reuse code throughout your program by calling the function.
* **Modularity:** Break down code into smaller, manageable pieces for better understanding and maintenance.
* **Data Abstraction:** Hide implementation details of a task for cleaner code and easier modification.
* **Parameterization:** Allow functions to accept data as input, making them adaptable.

**Q: What are some examples of poorly written functions in JavaScript?**

A: Poorly written functions might have:

* **Side effects:** Unexpectedly modify data outside their scope.
* **Magic numbers:** Use hardcoded values without clear meaning.
* **Unclear naming:** Don't clearly indicate their purpose.
* **Unnecessary parameters:** Take arguments not always needed.
* **Do too much:** Try to perform multiple unrelated tasks.

**Q: What are some hallmarks of well-written functions in JavaScript?**

A: Well-written functions are:

* **Clear and Readable:** Easy to understand with descriptive names and proper formatting.
* **Single Responsibility:** Perform a single, well-defined task.
* **Pure Functions (when possible):** Avoid side effects and rely only on input for output.
* **Proper Input Validation:** Check input parameters for expected data types.
* **Efficient Use of Parameters and Return Values:** Take only necessary parameters and provide clear return values.

**Q: What are the basic steps to writing a proper function?**

A: Steps to writing a proper function include:

1. Define the purpose of the function.
2. Choose a meaningful name that reflects its purpose.
3. Identify any parameters (arguments) the function needs.
4. Write the function body with clear logic using the parameters.
5. Use return to specify the output value (if applicable).
6. Test the function with different inputs to verify expected results.

**Q: How do you write and format "if statements" inside JavaScript functions?**

A: An if statement has:

* **Condition:** An expression in parentheses that evaluates to true or false.
* **Code Block:** Code to execute if the condition is true (enclosed in curly braces {}).
* **Optional else Block:** Code to execute if false (also in curly braces {}).
* Use proper indentation (usually 2 or 4 spaces) for readability, especially with nested statements.
* Brackets are only around the condition itself, not the entire statement or code block.

**DOM Manipulation with JavaScript**

**Q: What is the Document Object Model (DOM)?**

A: The DOM is a programming interface that represents a web page as a tree structure. It allows scripts to access and manipulate the content, structure, and style of the page.

**Q: How is DOM manipulation different from changing website code?**

A: DOM manipulation is temporary and client-side. It modifies how your browser interprets the existing website code for you only. Changes disappear on refresh. Website code changes are permanent and server-side, modifying the actual code everyone sees.

**Q: How do I manipulate an HTML document using JavaScript?**

A: You can use JavaScript to:

* Select elements using methods like getElementById, getElementsByClassName, or getElementsByTagName.
* Modify element content with properties like innerHTML or textContent.
* Add or remove elements using methods like createElement, appendChild, and removeChild.
* Change element styles using the style property.

**Q: What are the purposes of DOM manipulation?**

A: Common purposes include:

* Creating dynamic and interactive web pages.
* Enhancing user experience with features like personalization and real-time updates.
* Simplifying complex user interactions.
* Building Single Page Applications (SPAs).
* Enabling asynchronous data loading for improved performance.

**Q: How can I add or remove attributes from elements?**

A: There are two main methods:

1. **setAttribute** and **removeAttribute**: These methods provide explicit control over adding or removing attributes and their values.
2. **Property Access**: This approach allows direct access and modification of attributes as properties. Note the subtle difference in how non-existent attributes and boolean attributes are handled.