JAVASCRIPT

Date: Page:

1- Variables and Data Types
- 'var', 'let', 'const': Declare variables with var, let, or const (block-scoped).
- Data Types: Primitive types include number, string, boolean, null, undefined, symbol, objects include object, array, function.

2- Control flow and loops
Conditional statements: Use if, else if, else for conditional branching.
Loops: Use for, while, do-while loops for iteration.

3- Functions
Declaration: Define functions using function keyword or arrow function () => {}.
Parameters and Arguments: Pass parameters to functions and use them as arguments.

4- Arrays and Objects
Arrays: Create arrays using square brackets [], access elements using index.
Objects: Create objects using curly braces {}, access properties using dot notation or bracket notation.

5- Scope and Closures
Scope: Understand global scope, function scope, and block scope.
Closures: Functions that have access to the outer function's variables, even after the outer function has finished executing.

Date: Page:

6- DOM Manipulation
Document Object Model (DOM): Represent HTML elements as objects, manipulate them using javascript.
Method: Use querySelector, getElementById, getElementsByClassName, etc. to select and manipulate DOM elements.

7- Events
Event Handling: Attach event listeners to DOM elements to respond to user actions.
common Events: click, mouseover, mouseout, keyup, keydown, etc.

8- Asynchronous Javascript
Callbacks: Pass functions as arguments to other functions, often used in asynchronous operations.
Promises: Represent a value that may be available now, in the future, or never.
Async/Await: Syntactic sugar for working with promises, making asynchronous code appear as synchronous.

9- Error Handling
Try-Catch: Use try and catch blocks to handle errors gracefully.
Throw: Use throw statement to throw an exception.

10- ES6 features
Arrow functions: Shorter syntax for defining functions.
Template literals: Use backticks to interpolate variables in strings.
Destructing: Extract value from arrays or objects into variables

11- Modules
Import/Export: Split code into multiple files and import/export functions

or variables between them.

12- Classes
Constructor: Define a constructor method to initialize class properties.
Methods: Define methods inside classes to encapsulate functionality.

13- Regular Expressions
Patterns: Define patterns for string matching & manipulation.
Method: Use test, exec, match, replace, etc., to work with Regular expressions.

14- Local Storage
Web Storage: Store data locally in the browser using local Storage and session Storage.
Methods: Use setItem, getItem, removeItem, clear, etc, to manipulate stored data.

15- Ajax and Fetch API
Asynchronous Request: Make asyn. HTTP requests to servers.
Fetch API: Modern Alternative to XMLHttpRequest for fetching resources.

16- ES6+ features
Spread Operator: Spread elements of ~~any~~ an array or object into another array or object.
Rest parameters: Gather remaining arguments into an array within function definitions.

17- Best Practices
Use Strict Mode: Enable strict mode to catch common coding mistakes and improve performance.
Avoid Global Variables: Minimize the use of global variables to prevent naming collisions and unexpected behavior.

18- Debugging
Console: Use console.log, console.error, etc for debugging.
Browser DevTools: Utilize browser developer tools for more advanced debugging and profiling.

19- Libraries and Frameworks
Popular Libraries: jQuery, Lodash, Moment.js, etc.
Frameworks: React, Angular, Vue.js, etc.

20- Deployment
Bundling and Minification: Bundle Javascript files and minify them for production deployment.
CDNs: Use Content delivery Networks to serving libraries and resources efficiently.