HTML & CSS

1. HTML

2. **Basics**
3. **Block element and inline element**
4. Element
    a. Void elements
    b. Container Element
5. Attributes
    a. boolean attributes
    b. lang attribute
6. Nesting
7. <!DOCTYPE html>
8. **head**
    a. **<meta>**
    b. <meta charset="utf-8">
    c. Adding an author and description

9. VS

10. h1 vs title in head
11. vs <i>
12. <b> vs

13.GOOD TO KNOW

14. Whitespace
15. entity references
    a. < <
    b. > >
    c. " "
16. Open Graph Data

17. CSS

18. Anatomy of CSS ruleset
19. Selecters
    a. Element
    b. Id, Class
    c. Attribute
    d. Pseudo
20. Box model

JAVASCRIPT

1. DOM

a. querySelector
b. textContent
c. addEventListener
d. Order of Parsing

2. **event Propagation**
   a. event Bubbling
   b. event Capturing/ Trickling
   c. how to add both on program
3. event.stopPropagation();
4. inst
   a. e.target
      i. id
      ii. tagName
      iii. pros and cons

5. Architecture

a. Execution context
   i. variable environment (memory)
   ii. Thread of execution (code)
   iii. - global & local execution context
   iv. - phases
      1. Memory allocation
      2. Code execution
b. Synchronous single threaded app
c. Call stack
d. Call stack
e. **Event loop**
   i. Callback queue/ task queue
   ii. Microtask queue
      1. mutation observer
   iii. Starvation
   iv. Memory Heap
f. Just In Time Compilation
g. Interpreter vs Compiler
h. Abstract Syntax Tree
i. Concurrency model

6. Theory

7. Data types
   a. wrapper objects
   b. 0 vs new Number(0)
   c. **Numbers**
      i. 1_000_000
      ii. 1e9, 1e-6
      iii. Hex, binary and octal numbers
      iv. toString(base)
      v. Math.trunc
8. Operators
9. enum
   a. how to get enum in javascript
10. **Function**
    a. Function Statement
    b. Function Expression
    c. Function Declaration
    d. Anonymous function
    e. Named Function Expression
    f. Functional Programing
    g. **Higher order function**
    h. First class function
    i. **Decorator function**
       i. use
       ii. - count no of function call
       iii. - valid data of params
    j. **Pure function**
       i. pros and cons
       ii. rules
       iii. pure vs impure
    k. IIFE
       i. pros
11. Advantages and disadvantages of JS
12. **Set Map Flat**
    a. set
       i. add, delete, has, clear, kyes, values, entries
       ii. <setName>.size
    b. map

        i. get, set, has, delete, clear, keys, values, entries, forEach
        ii. iterating
    c. object vs map
    d. weekSet()
        i. features
    e. weekMap()
        i. features
        ii. key is private
    f. Week set and map summary
    g. falt()
    h. flatMap()
    i. reduceRight()
    j. copyWithin()
13. **Operators**
    a. Nullish coalescing operator
    b. Optional chaining
    c. || vs ??
    d. Ternary operator
    e. Type Operators
    f. **Unary operators**
        i. delete
        ii. typeof
        iii. !, ++, -, +
    g. **Bitwise Operators**
        i. bitwise OR
        ii. bitwise AND
        iii. uses
14. **Scope**
    a. Global scope
    b. Module scope
    c. Function scope
    d. Lexical scope
    e. Block scope
15. Shadowing & Illegal shadowing
16. **Prototype**
17. Types of error
    a. syntax, logic
18. **Closure**
    a. Disadvantage
    b. Uses
    c. lexical scope vs closure
    d. IIFE
19. **Garbage collection**
    a. How does it work?
    b. mark-and-sweep
    c. reachability
    d. **Optimizations**
        i. - Generational
        ii. collection
        iii. - Incremental collection
        iv. - Idle-time collection
20. **Hoisting**
    a. TDZlet, const vs var
    b. Function vs arrow function
21. **Call Apply Bind**
    a. function borrowing
    b. call vs apply vs bind
    c. polyfills
22. transpiler
    a. bable
    b. webpack
23. polyfills vs transpiler
24. This Keyword
25. **String Methods**
    a. Length, toUpperCase, LowerCase, Trim, Pad, charAt, Split, Concat, substring, indexOf, lastIndexOf, localeCompare
26. **Array Methods**
    a. Map, Filter, Reduce, Find, Sort, Foreach, Push, Pop, Shift, Unshift, Slice, Splice, concat, indexOf, lastIndexOf, forEach, split, join, reduceRight, iArray, fill, copy, flat
27. **Object Methods**
    a. object constructor, literal
    b. deleting field
    c. Computed properties
    d. __proto__
    e. in
    f. Object.assign
    g. structuredClone
    h. _.cloneDeep(obj)
    i. methods
    j. this keyword

    k. Symbol type
28. Symbol
    a. properties
    b. useail
    c. ongo
    d. global symbol registry
    e. for, keyFor, iterator, toPrimitive
29. **Loop**
    a. for
    b. do while vs while
    c. labelled statements
    d. - break
    e. - continue
    f. for...in
    g. for...of
30. **Callback**
    a. callback hell
    b. inversion of control
31. **Promises**
    a. Promise states
    b. Promise chaining
    c. Promise.all
    d. Promise.allSettled
    e. Promise.any
    f. Promise.race
    g. Promise.resolve
    h. Thenable
    i. Finally
    j. Catch
    k. immutable
    l. promisify
    m. pros and cons
32. **Async await**
    a. async always return a promise
    b. error handling in async await
33. **Debouncing & Throttling**
    a. both are used for optimising performance of a web app
    b. by limiting the rate of API calls
34. Spread and Rest Operator
35. DOM, BOM
36. **ES6 and its features**
    a. Let, Var, Const
    b. Ternary operator
    c. Arrow function
    d. Template literals
    e. Default Parameters
    f. Classes
    g. Modules
    h. Iterators
    i. Object & Array Destructuring
37. **Primitive and non-primitive**
    a. Pass by value and pass by reference
38. Message queue
39. Life
40. Generator
41. **Prototype**
    a. Prototype chain
    b. Prototypal Inheritance
    c. uses?
    d. Circular reference
    e. Object.key
42. **Recursion**
    a. recursive call to function
    b. condition to exit
    c. pros and cons
    d. display the fibonacci sequence
    e. use
43. JavaScript is dynamically types
44. **Currying**
    a. function inside function
45. **Type Casting**
    a. Implicite (Coercion)
    b. Explicit (Conversion)
46. Microtask queue
47. **Shallow copy vs Deep copy**
    a. primitive vs structural
    b. how to make these copies
    c. pros and cons
    d. Mutable vs Immutable
    e. Object.freeze()
48. TCP/IP
49. DNS
50. **IIFE**
    a. pros and cons