Benjamin Zhuang - Week 1 Day 5

---

1. What is ECMAScript?
    - ECMAScript is the standard for Javascript to ensure compatibility across different browsers.
2. What is the JavaScript engine?
    - A JavaScript engine is simply a software program that optimizes and executes JavaScript code.
3. Explain just-in-time (JIT) compilation. What's the difference between JIT compilation and interpretation?
    - JIT compilation optimizes and compiles the program into machine code at **runtime**, then executes the program.
    - Interpretation uses an interpreter program, which is often written in some programming/scripting languages, and executes the program **line-by-line**.
4. What are primitive data types in JS?
    - number
    - boolean
    - String
    - undefined
    - null
    - symbol
    - bigint
5. What are reference data types in JS?
    - objects
6. What is type coercion, and how does it differ from type conversion?
    - A type coercion is an implicit change of a data type into another data type during some operation.
    - A type conversion is an explicit cast from a data type into another data type.
7. What is dynamic typing?
    - The data type of variables are checked and assigned at runtime.
8. What is immutability?
    - Immutability refers to a value that cannot be changed without creating a new value, thus immutable.
    - What data types are immutable?
        - Primitive types
9. What is the difference between == and ===?
    - == only checks the value but not the data type.
    - === checks the value and the data type.

10. What are some examples of falsey values in JS?
    - false, undefined, null, 0, “”, NaN
11. How do primitive and reference data types differ in where they’re stored in memory?
    - Primitive data stores the value in memory.
    - Reference data stores the reference location of the data in memory.
    - How does this affect them when they are passed as arguments to a function?
        - Because primitive types are passed by value, they will not be affected outside the function scope.
        - Making changes to the reference types inside the function will also change the data itself.
12. What are 3 ways to declare functions? What is their syntax?
    - function name() {} [keyword function]
    - const name = function {} [function expression]
    - const name = () => {} [arrow function]
13. What are 3 ways to iterate an array? What is their syntax?
    - Sequential for-loop
        - for (let i=0; i <arr.length; i++) {}
    - Array.prototype.forEach
        - arr.forEach((item) => {})
    - ES6 for-of syntax
        - for (const item of arr) {}
14. What are the major differences between a set and array?
    - Values in a set are unique and not indexed (unordered).
    - Values in an array are not unique and indexed (ordered).
15. What are the major differences between a map and object?
    - The key of a map object can be anything, while the key of an object can only be string.
    - Map object entries are ordered based on the insertion order, while the object is unordered.
    - You can access the size of a map object by map.size(), while you have to iterate through an object to find the size.
    - Map object is not serializable, while an object can be serialized/de-serialized by JSON.parse() and JSON.dump().
16. What is the DOM?
    - It is a document tree structure to represent nested HTML elements, where these HTML elements are represented as nodes.
17. How can you select an HTML element using JS?
    - document.querySelector()

    - document.querySelectorAll()
    - document.getElementById(), etc.
18. What is a DOM event?
    - A signal that something has occurred/occuring on the web pages. It can be triggered by user interactions or by the browser.
19. What is the Event interface?
    - An object that describes a DOM event.
20. How do we register event handlers for a selected element?
    - element.addEventListener()
21. What is event propagation? How many phases are there? In what order does it occur?
    - Event propagation is a mechanism of how events are traveled through the DOM tree until it reaches the target and what should happen afterwards.
    - Capturing, target, bubbling.
22. Explain event delegation. Why is it important?
    - It is common that multiple siblings have the same event handler behavior. It is difficult to manage and prone to bugs if we assign an event handler to every single sibling. Rather, we lift the event handler to their parents and let it handle the event, which optimizes the code.