Short Answer:

Answer the following questions with complete sentences in **your own words**. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers.

1. **What is ECMAScript?**

It is a standard for scripted language

2. **What is the JavaScript engine?**

Software that optimizes and executes JavaScript code

3. **Explain just-in-time (JIT) compilation. What's the difference between JIT compilation and interpretation?**

JIT compilation is a method for improving the performance of interpreted programs.

The difference between interpretation and JIT compilation is that the JIT compilation translates source code into native code, whereas an interpreter runs code directly, runs precompiled code, or turns code into an intermediate language.

4. **What is REPL?**

REPL (Read-Eval-Print-loop) is a programming language environment that where user inputs are read and evaluated, and then the results are returned to the user

5. **What are primitive data types in JS?**

Primitive Data types are datas that is not an object and has no methods or properties. Examples include string, number, bigint, boolean, undefined, symbol, and null. Because of this, they are immutable or data can't be changed.

6. **What are reference data types in JS?**

Reference data types in JS are datas that are objects. Examples include array, object, and function. They can be mutable or data can be changed.

7. **What is type coercion, and how does it differ from type conversion?**

type coercion is implicit data change whereas type conversion can be either implicit or explicit data change. Example of type coercion includes when you add a string a number together. Example of type conversion includes using methods such as String(), Boolean(),Number().

8. **What is dynamic typing?**

Dynamic typing involves not specifying what type the data is when declaring a variable. This is because JavaScript engine infers what type this variable is based on the value assigned to at run time.

9. **What is immutability?**

Immutability means that data can't be changed.

1. **What data types are immutable?**

Primitive data types such as string, number, bigint, boolean, undefined, symbol, and null are considered

immutable

10. **What is the difference between == and ===?**

== means loose equality and it doesn't take in account the type of data when comparing two variables. For example, '5' == 5 will return true.

==means strict equality and it does take into account the type of data when comparing two variables. Therefore, '5' === 5 will equal false.

11. **What are some examples of falsy values in JS?**

Falsy values include 0, empty string (", "",``), undefined, null, and NaN

12. **Explain short-circuit evaluation.**

Short-circuit evaluation is the evaluation of an expression from left to right with || and && operators. If the condition is met and the rest of the conditions won't affect the already evaluated result, then the expression will return that result (value).

13. **How do primitive and reference data types differ in where they're stored in memory?**

Primitive data are fixed data stored in stack and manipulates the actual value in the variable.

Reference data are dynamic data stored in a heap where you manipulate the reference of that data in memory and not the data itself

1. **How does this affect them when they are passed as arguments to a function?**

The affect them when they are passed as arguments to a function because it involves the issue of shallow copy and deep copy.

14. **What are 3 ways to declare functions? What is their syntax?**

Function Declaration- function dog(){}
Function expression- var dog function(){}
Arrow function- const dog=()=>{}

15. **How do first-class functions differ from higher-order functions?**

First-class functions are functions that can be assigned to variables, passed as arguments, and returned from functions.

Higher-order functions are functions that take one or more functions as arguments or return a function as the result.

16. **What are pure functions?**

a pure function always returns the same result if the same arguments are passed.

17. **What are 3 ways to iterate an array? What is their syntax?**

For Loop:

```
let array=[ 1, 2, 3, 4, 5, 6 ];
for (let i= 0; i< array.length; i++) {
        console.log(array[i]);
}
```

While loop:

```
let index = 0;
Let array = [ 1, 2, 3, 4, 5, 6 ];

while (index < array.length) {
   console.log(array[index]);
   index++;
}
```

ForEach:

```
let index = 0;
let array = [ 1, 2, 3, 4, 5, 6 ];

array.forEach(myFunction);
function myFunction(item, index)
{
   console.log(item);
}
```

18. **What are the major differences between a set and array?**

Arrays are ordered through index. It can be accessed through index. It doesn't have unique values. It can use .includes() method to look up elements, it can add elements with methods such as .push(), .unshift(), and .splice()

Set aren't ordered. It cant be accessed iteratively. It does have unique values. It uses .has() method to lookup. It uses .Add() method to add elements

19. **What are the major differences between a map and object?**

A map doesn't have a default key. Its key types can be anything. The ordering is based on insertion. You can find the size of it through .size. It is iterable. However, it doesn't have serilaization and parsing.

An object does have default keys within the protype. Its key type are based off string and symbol. It isn't ordered. To find the size, it must be iterated. It is not iterable. For serilazation and parsing, you can use JSON.stringify and JSON.parse.

20. **What is the DOM?**

DOM(Document Object Model) and it determines how a browser represents the webpage. The DOM model has a tree like structure.

21. **How can you select an HTML element using JS?**

You can use Query selectors and document.getElementBy_ to select HTML with JavaScript

22. **What is a DOM event?**

DOM event allows JavaScript to use event handlers and on the HTML document

23. **What is the Event interface?**

Event interface is the event that happens in DOM.

24. **How do we register event handlers for a selected element?**

One way to register event handlers is by using the EventTarget. addEventListener method

25. **What is event propagation? How many phases are there? In what order does it occur?**

Event propagation is how events travels from DOM. The 3 phrases from order are capturing, target, and then bubbling.

26. **Explain event delegation. Why is it important?**

Event delegation are ways how to handle events efficiently. It is important because it **helps avoid adding event listeners to specific nodes by adding**,the event listener to one parent. That event listener analyzes bubbled events to find a match on child elements.