Name : Ricardo Ferreira

1. Why is it important to write clean code?

Writing clean code is important because it allows you to clearly communicate with the next person who works with what you've written. Being able to return to previously written code and understand what it does is key, especially in the software development world.

2. What is the difference between good comments and bad comments?

Accurate, well-placed, informative, and intention-revealing comments are good;

Good comments describe why the code is doing what it does.

Bad comments describe what the code is doing.

3. What is an array?

In JavaScript, an array is an ordered list of values. Each value is called an element specified by an index: A JavaScript array has the following characteristics: First, an array can hold values of mixed types.

4. What are arrays useful for?

In JavaScript, array is a single variable that is used to store different elements. It is often used when we want to store list of elements and access them by a single variable.

5. How do you access an array element?

Access the values by referring to an index number.

6. How do you change an array element?

7. What are some useful array properties?

splice() Adds/Removes elements from an array

toString() Converts an array to a string, and returns the result

unshift() Adds new elements to the beginning of an array, and returns the new length

valueOf() Returns the primitive value of an array

8. What are some useful array methods?

filter()

forEach()

some()

every()

includes()

map()

reduce()

sort()

9. What are loops useful for?

Loops are used in JavaScript to perform repeated tasks based on a condition. Conditions typically return true or false . A loop will continue running until the defined condition returns false

10.What is the break statement?

The break statement terminates the current loop, switch , or label statement and transfers program control to the statement following the terminated statement.

11. What is the continue statement?

The continue statement terminates execution of the statements in the current iteration of the current or labeled loop, and continues execution of the loop with the next iteration.

12.What is the DOM?

he DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

13.How do you target the nodes you want to work with?

In order to interact with any node in the tree, we first need to target (select) it. We can do this using one of the multiple methods the DOM API offers (notice that all these methods are called on the document object using the dot notation

14.How do you create an element in the DOM?

To create a DOM element, you use the createElement() method

15.How do you add an element to the DOM?

First, create a div section and add some text to it using <p> tags.

Create an element <p> using document. createElement("p").

Create a text, using document. ...

Using appendChild() try to append the created element, along with text, to the existing div tag.

16.How do you remove an element from the DOM?

To remove an element from the DOM, you can also use the remove() method of the element

17.How can you alter an element in the DOM?

Create an HTML element with document. createElement.

Add content to the HTML element by setting the innerHTML .

Add it to the DOM with parentNode. prepend or parentNode. append

18.When adding text to a DOM element, should you use textContent or innerHTML?

You should use textContent.

19.Where should you include your JavaScript tag in your HTML file when working with DOM nodes?

Using the script tag to include an external JavaScript file

The value for the src attribute should be the path to your JavaScript file. This script tag should be included between the <head> tags in your HTML document.

20.How do “events” and “listeners” work?

Often an event listener is registered with the object that generates the event. When the event occurs, the object iterates through all listeners registered with it informing them of the event.

21.What are three ways to use events in your code?

inline, using a property, or using a listener

22.Why are event listeners the preferred way to handle events?

You can only have one event handler for a specific event type, but you can add multiple event listeners for it.

23.What are the benefits of using named functions in your listeners?

A named function helps keep your code more DRY (an acronym for Don't Repeat Yourself). Second, you can remove them later if you want using removeEventListener()

24.How do you attach listeners to groups of nodes?

We can get a nodelist of all of the items matching a specific selector with querySelectorAll('selector')

25.What is the difference between the return values of querySelector and querySelectorAll?

querySelector() methodcan only be used to access a single element while querySelectorAll() method can be used to access all elements which match with a specified CSS selector. To return all matches, querySelectorAll has to be used, while to return a single match, querySelector is used

26.What does a “nodelist” contain?

A NodeList is a collection of document nodes (element nodes, attribute nodes, and text nodes)

27.Explain the difference between “capture” and “bubbling”.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements. With capturing, the event is first captured by the outermost element and propagated to the inner elements

28.What is the difference between objects and arrays?

Arrays are objects only in javascript. The major difference is that they store the data in an ordered collection in which the data can be accessed using a numerical index

Objects are mutable data structure in javascript which is used to represent a ‘Thing’. This could be anything like cars, plants, person, community etc.

It stores the data in key value pair and the key can be anything except for undefined. The keys are iterable and can be accessed in any order.

29.How do you access object properties

Dot property accessor: object. property.

Square brackets property access: object['property']

Object destructuring: const { property } = object