JavaScrip 2

1. Why is it important to write clean code?

The answer:

- **it allows you to clearly communicate with the next person who works with what you've written.**

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

The answer:

- **Underestimating the value of good comments is disregarding a very effective way of code documentation. It's easy, fast, and very straight-to-the-point, since it's done right then and there when it's needed.**
- **Comments are also efficient considering the fact that they can be read at the exact moment of doubt.**
- **Comments aren't to be treated lightly. When commenting on code, the current functionality is explained in terms of variables and results. What comments are NOT made for is:**
- **Writing explanatory notes to self (e.g. /* Will finish this later… */).**
- **Blaming stuff on other people (e.g. /* John coded this. Ask him. */).**
- **Writing vague statements (e.g. /* This is another math function. */).**
- **Erasing chunks of code. Sometimes people are not sure of erasing things and it's not absolutely evil to comment that code instead.**
- **What's not right is to just leave it afterwards. It'll be terribly confusing. If the code will be documented via embedded comments, the team members need to make sure those comments are there for a reason.**

3. What is an array?

The answer:

- **An array is a special variable, which can hold more than one value.**

4. What are arrays useful for?

The answer:

- **If you have a list of items (a list of car names, for example).**
- **An array can hold many values under a single name, and you can access the values by referring to an index number.**

5. How do you access an array element?

The answer:

- **You access an array element by referring to the index number:**

6. How do you change an array element?

The answer:

- **This statement changes the value of the first element in cars: cars[0] = "Opel";**

7. What are some useful array properties?

The answer:

- **cars.length // Returns the number of elements**
- **cars.sort() // Sorts the array**
- **fruits.push("Lemon"); // Adds a new element (Lemon) to fruits**

8. What are some useful array methods?

The answer:

- **toString()**
- **join()**
- **pop()**
- **push()**
- **shift()**
- **unshift()**

9. What are loops useful for?

The answer:

- **Loops are all about doing the same thing over and over again. Often, the code will be slightly different each time round the loop, or the same code will run but with different variables.**

10.What is the break statement?

The answer:

- **If you want to exit a loop before all the iterations have been completed, you can use the break statement.**

11. What is the continue statement?

The answer:

- **The continue statement works similarly to break, but instead of breaking out of the loop entirely, it skips to the next iteration of the loop.**

12.What is the DOM?

The answer:

- **The Document Object Model (DOM) is the data representation of the objects that comprise the structure and content of a document on the web.**

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

The answer:

- **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):**
- **getElementById().**
- **getElementsByClassName().**
- **getElementsByTagName()**
- **getElementsByName().**
- **querySelector().**
- **querySelectorAll().**

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

The answer:

- **document.createElement();**

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

The answer:

- **.appendChild(node)**

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

The answer:

- **.remove();**

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

The answer:

- **When you have a reference to an element, you can use that reference to alter the element's own properties. This allows you to do**
- **many useful alterations, like adding/removing and altering attributes, changing classes, adding inline style information and more.**
-

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

The answer:

- **textContent**

19.Where should you include your JavaScript tag in your HTML file when working

with DOM nodes?

The answer:

- **The best practice is to put JavaScript <script> tags just before the closing </body> tag.**

20.How do "events" and "listeners" work?

The answer:

- **An event listener is a function that initiates a predefined process if a specific event occurs.**
- **The addEventListener() is an inbuilt function in JavaScript which takes the event to listen for, and a second argument to be called whenever the described event gets fired.**

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

The answer:

- **The addEventListener Method**
- **Delegating Events to addEventListener**
- **Using the onclick Attribute**

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

The answer:

- **Generally, it is advised to use event listeners over event handlers because you can add multiple event listeners for the same event.**

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

The answer:

- **We can access more information about the event by passing a parameter to the function that we are calling.**

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

The answer:

- **Using a loop**

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

The answer:

- **querySelector() method can 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**

26.What does a “nodelist” contain?

The answer:

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

The answer:

- **Capturing phase : the event moves down towards the element. Bubbling phase: the event bubbles up from the element.**

28.What is the difference between objects and arrays?

The answer:

- **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.

- **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.**

  They are also mutable and data can be modified at any index. Indexes are zero based which means the first item is stored at Oth index, second at first and so on, last item is stored at n-1th index.

29. How do you access object properties?

The answer:

- **Dot property accessor: object.property.**
- **Square brackets property accessor: object['property']**
- **Object destructuring: const { property } = object.**