**Interview Questions**

**1.** **How to define a class with properties and methods in JavaScript?**

 <script>

    // class is defined by using class keyword

    class nameOfClass{

        // constructor is an inbuilt method to class having properties

        constructor(property1,property2,property3){

            this.property1 = property1;

            this.property2 = property2;

            this.property3 = property3;

        }

        // in classes we can create different methods too

        newMethod(){

            // this method is used for printing the property2 in console

            let local = this.property2;

            console.log(local);

        }

    }

    // new syntax is used to create a new class

    let bmw = new nameOfClass("BMW","5 Series",2014);

    let hyundai = new nameOfClass("HYUNDAI","Venue",2018);

    console.log(bmw);

    console.log(hyundai);

    //different ways to call the same properties

    console.log(bmw.property2);

    console.log(bmw.newMethod());

    </script>

**2. How to implement class inheritance in JavaScript?**

<script>

    class parentClass{

        constructor(name,age){

            this.name = name;

            this.age = age;

        }

        methodName(){

            // name variable is local

            let name = this.name;

            console.log(name);

        }

        methodAge(){

            // age variable is local

            let age = this.age;

            console.log(age);

        }

    }

    // child class contains both its and parent methods

    // to make another class as child to this parentClass we have to use "extends" keyword

    class childclass extends parentClass{

        childMethod(){

            let college = "ABC college";

            console.log(college);

        }

        method(){

            // we can access the parent methods by using "super" keyword

            super.methodName();

            super.methodAge();

            // to access the same class method we use "this" keyword

            this.childMethod();

        }

    }

    // when we use childclass then it inherit the properties and methods of parentclass

    let prudhvi = new childclass("prudhvi","26");

    prudhvi.method()//console name,age and college

    // in this way we give inheritence of parentclass to childclass

    prudhvi.methodAge(); //we can call the parent method too

    </script>

**3. how to find duplicate elements in a given array?**

<script>

    const array = [52,61,57,25,52,59,5,57,100,5];

    var duplicates = [];

    for(i=0;i<=array.length;i++){

        let check = array[i];

        for(j=i + 1;j<=array.length;j++){

            if(check == array[j]){

                duplicates.push(array[j]);

            }

        }

    }

    console.log(duplicates);

    // filter is an inbuilt property for an array to filter the duplicate elements

    var shortcut = array.filter((ele,index,arr)=>arr.indexOf(ele) != index);

    console.log(shortcut);

    </script>

4. Enumerate the differences between Java and JavaScript?

|  |  |
| --- | --- |
| Java | Javascript |
| It is a Programming language. | It is a scripting language. |
| Java is a pure Object oriented programming | it is an object based programming |
| Java program should be compiled before execution. | JavaScript needs to be integrated into the HTML program for the execution. |

### 5. Can we display a web page inside a web page or is nesting of webpages possible?

Yes, we can display a web page inside another HTML web page. HTML provides a tag <iframe> using which we can achieve this functionality.

<iframe src=”url of the page to be embedded”></iframe>

### 6. Describe HTML layout structure.

Every web page has different components to display the intended content and a specific UI. But still, there are few things which are templated and are globally accepted way to structure the web page, such as:

* **<header>**: Stores the starting information about the web page.
* **<footer>**: Represents the last section of the page.
* **<nav>**: The navigation menu of the HTML page.
* **<article>**: It is a set of information.
* **<section>**: It is used inside the article block to define the basic structure of a page.
* **<aside>**: Sidebar content of the page

### 7. What is the difference between the ‘id’ attribute and the ‘class’ attribute of HTML elements?

Multiple elements in HTML can have the same class value, whereas a value of id attribute of one element cannot be associated with another HTML element.

### 8. What are HTML Entities?

In HTML some characters are reserved like ‘<’, ‘>’, ‘/’, etc. To use these characters in our webpage we need to use the character entities called HTML Entities.

### 9.  What are the various formatting tags in HTML?

 HTML has various formatting tags:

* **<b>** - makes text bold
* **<i>** - makes text italic
* **<em>** - makes text italic but with added semantics importance
* **<big>** - increases the font size of the text by one unit
* **<small>** - decreases the font size of the text by one unit
* **<sub>** - makes the text a subscript
* **<sup>** - makes the text a superscript
* **<del>** - displays as strike out text
* **<strong>** - marks the text as important
* **<mark>** - highlights the text
* **<ins>** - displays as added text

### 10.  When to use scripts in the head and when to use scripts in the body?

If the scripts contain some event-triggered functions or jquery library then we should use them in the head section. If the script writes the content on the page or is not inside a function then it should be placed inside the body section at the bottom. In short, follow below three points:

1. Place library scripts or event scripts in the head section.
2. Place normal scripts that do not write anything on the page, in the head section until there is any performance issue.
3. Place scripts that render something on the web page at the bottom of the body section