14/05/25 Technical test - 1

R. GOKULNATH
MERN STACK ①
6379142995

① A code to find no. of occurances of particular number using array concept

```
<!DOCTYPE html>
<html>
<head><title>No. of occurances of number</title></head>
<body><script>
const arr=[1, 2, 1, 3, 2]; let occur=0; let num=0;
for(int i=0; i<arr.length; i++)
{ for(int j=0; j<arr.length-1; j++)
  { if(arr[i]==arr[i+1])
     {
       occur=occur+1; num=arr[i];
       i++;
     }

  }
}
document.write("The number"+arr[i]+" occured in" +
                                   occur+"times
</script></body></html>              in array");
```

② DOM model and document object.

*. DOM is document object model. which can access the elements of a web page like the button and other element.

*. Document object is which can manipulate the content when web page loaded.

*. sample:
```
<script>
    let msg = 'hello';
    document.getElementById('demo').innerHTML
                                        =msg;
</script>
```

In this above sample, after the content loaded we can change the HTML content inside 'demo' Id. also use class, tagname

③ validation:

*. In Javascript, validation means having set of rules to get inner like dont allow empty values, dont have unformatted email not in regular Expression we have also to accept only 10 numbers in mobile field.

*. Eg:

```
<script>
    let phone = document.getElementById('phone').value;
let ph_length = phone.length;
    if (ph_length > 10)
    {
       alert('number should not be greater than
                                          10");
    } </script>.
```

```
    <script>
(*) let name = document.getElementById('name').value;
    if (name == ' ');
    {
      alert('name should not be Empty');
    }
    </script>
```

④ class and object:

*. class is a collection of object. object are instances of class. we can create multiple object as we need with the class.

sample:

```
<script>
class person
{
  display_name (x)
  {
    return 'name' + name;
  }
}
</script>
```

```
<script>
let person = new object();
                 person.
det re = display_name('gokul');
document.write(re);
</script>
```

↳ In this person is class we created object to that class.

⑤ Operator:

*. Operator is used to perform some actions on values of the variables we have.

- *. arithmetic operator
- *. assignment operator
- *. logical operator.
- *. conditional operator.
- *. Bitwise operator.
- *. Ternary operator.

Eg: 1:

```
<script>
 let c = a*b;
 document.write(c);
</script>
```

Eg: 2:

```
 <script>
  let a=10;
  a+=20;
  document.write(a);
 </script>
```

Eg:-(3)

```
<script>
   let a=10; b=20;
   if (a==b)
   { alert('Equal'); } else { alert('not Equal'); }}
```

Eg:(4) `<script> let c=(a>b)? 'greater': 'lesser';`

⑥ Decision making statement:

*. Decision making statement are used to take decision based on the condition. In this we have if else, switch statement. In if else → ladder if else, → nested if else, switch statement we have cases with break.

Eg: 1:

```
<script>
        let a=10;
        let b=20;
        let c;
        if(a==b)
        { alert('Equal')
         }
        else
        { alert('not Equal')
         }
</script>
```

Eg: 2:

```
<script>
let a = parseInt(prompt('Enter
                    value'));
switch(a)
{ case 1:
     alert('hello');
     break
  case 2:
     alert('hai');
     break
 default:
     alert('nothing');
} </script>
```

⑦ looping statement

*. In looping statements, we have while, do while, for loop. while -> executes untill condition fails have increment in braces. do while -> executes one time even if condition fails for has increment, initialisation, condition in same line

①
```
<script>
let i=1;
while (i<=10)

{
   ~~alert~~ document.write
                 (i);

   i++;

}
</script>
```

```
② <script>
  let i=1;
 do
 {
   document.write(i);

     i=i+1;

 }
 while(i<=10);
```

```
③ <script>
 for(let i=1; i<=10; i++)

 {
 document.write(i+"<br>");

 }
```

⑧ Inheritance: In inheritance we have parent class and child class also known as base class, inherited class. we can call the inherited class to use parent class values.

Eg:
```
<script>
class animal
{
   initialisation(a,b)
   {
       let this.a=a;
        let this.b=b;
    }
}
class domestic extends animal
{
   behaves(a,b)
   {
     return this.a+" "+this.b;
   }
}
</script>
<script>
```

```
    let domestic = new object()';
let c = domestic.behaves('hello', 'Animal');

    document.write (c);

</script>
```

↳ In this we can inherit from base class using extends keyword. we can inherit animal base class and extends domestic inherited class from base class.

⑨ constructor:

*. constructor is used to invoke automatically without calling. we can use inside the class like initialisation it get initialised without calling and we can use the initialised value in any other function in same class.

*. only one constructor to be used in class.

Eg:

```
<script>
class animal
{
    constructor(a,b)
    {
        this.a = a;
        this.b = b';
    }
    addition()
    {
        return this.a + this.b;
    }
}
</script>
```

```
<script>
let animal = new object
            (10,20);
let add = animal.addition();
document.write(add)';
</script>
```

↳ In this we can use the initialised value in addition directly with the use of constructor.

(*) (this keyword to use variable in same class)

R.GOKULNATH
MERN STACK (6)
6379442995
16/05/25

(10) constructor with parameters:-

*. constructor is used to invoke automatically without calling

*. we can use constructor inside the class like initialisation get initialised.

*. without calling we can use the value.

*. with parameters we can pass the values what we need. without it we can use the assumed values

without parameter:

```
<script>
  class number
  {
     constructor()
     {
          let a=10;
           let b=20;
     }
     addition()
     {
          let c = this.a +
                          this.b;
          alert(c);
      }
   }
</script>
<script>
    let number= new object();
numb.addition();
</script>
```

with parameters:

```
<script>
   class number_in()
   {
     constructor(1a,1b)
     {
          this.val1 = 1a;
          this.val2 = 1b;
     }
     display()
     {
        return this.val1 + this.val2;
     }
   }
</script>
<script>
   let number_in = new object
                            (10,200);
                  number.
   let inc = display();
   alert( inc);
</script>
```