11-09-2020

1.Concatenation

```
<!DOCTYPEhtml>
<html>
<body>

<h2>JavaScript concatenation</h2>


<script>           //we can use concat() method can be used instead of the plus operator
var text1 = "Hello";                          //var text = "Hello" + "" + "World!";
                                           // var text = "Hello".concat("", "World!");
var text2 = "World!";
var text3 = text1.concat("",text2);
document.write(text3);
</script>

</body>
</html>
```

2.Difference betweengetdate() and getday() for cases

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var day;                             //getDate() will Get the day as a number (1-31)
var daysname = new Date().getDate()//getDay() will Get the weekday as a number (0-6)
switch (daysname) {
case 6:
day = "Sunday";
break;
case 7:
day = "Monday";
break;
case 8:
day = "Tuesday";
break;
```

```
case 9:
day = "Wednesday";
break;
case 10:
day = "Thursday";
break;
case 11:
day = "Friday";
break;
case 12:
day = "Saturday";
}
document.getElementById("demo").innerHTML = "Today is " + day;
</script>

</body>
</html>
```

3.Properties

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Properties</h2>

<p id="demo"></p>

<script>
var person = {
firstname:"sandeep",
lastname:"m",                //There are two different ways to access an object property:
age:22,                        //You can use .property or ["property"]
};

document.getElementById("demo").innerHTML = person["firstname"] + " is " + person["age"]
+ " years old.";
</script>

</body>
</html>
```

4.Methods

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
 var person = {                                    //A method is a block of code which only
                                                                    runs when it is called
                  //in a function definition, this refers    to the owner of the function.

              //In this example this is the person object that owns the fullName function.

             //In other words, this.firstName means the firstName property of this object.
firstName: "sandeep",
lastName : "m",
};
person.name = function() {
return this.firstName + "" + this.lastName;
};

document.getElementById("demo").innerHTML =
"My name is " + person.name();
</script>

</body>
</html>
```

5.Difference between document.getElementById(id) and console.log().

To access an HTML element, JavaScript can use the document.getElementById(id) method.

For debugging purposes, we can call the console.log() method in the browser to display data.

debugging is the process of identifying and removing errors from computer hardware or software.

For example1,

```
<!DOCTYPE html>

<html>
<body>



<scriptlangugae="javascript">

function Combine() {

document.getElementById('txt3').value = document.getElementById('txt1').value + "" +

document.getElementById('txt2').value;

}

</script>
```

// document.getElementById is used almost every time we want to manipulate, or get info from, an element on your document.

```
<inputtype="text"id="txt1"name="txt1"/> +

<inputtype="text"id="txt2"name="txt2"onchange="Combine();"/> =

<inputtype="text"id="txt3"name="txt3"/>

</body>

</html>
```

For example 2

// document.getElementById(). innerHTML

This is also used to write something on webpage,

But it is very specific with location where content has to be written unlike document. write() method which is written at the end of webpage.

This doesn't overwrite whole content of webpage but overwrite content of element it is targeted.

```
<!DOCTYPE html>

<html>

<body>


   <h2>JavaScript Data Types</h2>


   <script>

      var x,y,z;                    // Now x,y,z is undefined

      x = 8;                       // Now  x is a number

      y="sandeep"                   // Now y is a string

      z = x+y;


      document.write(z);

   </script>


</body>

</html>
```