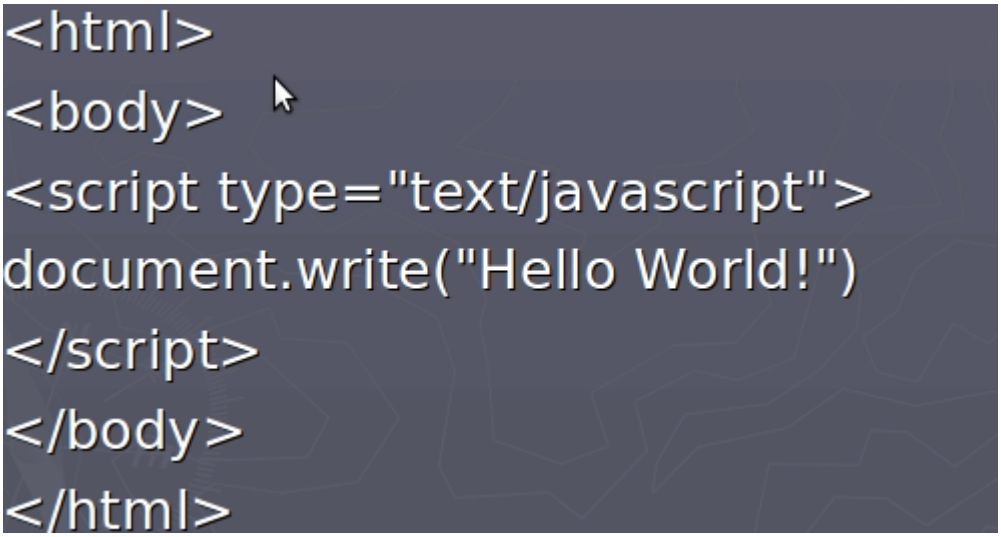
JavaScript

1. JavaScript is used in millions of Web pages to improve the design, validate forms, detect browsers, create cookies, and much more.
2. JavaScript is the most popular scripting language on the internet, and works in all major browsers, such as Internet Explorer, Mozilla, Firefox, Netscape, Opera.
3. JavaScript was designed to add interactivity to HTML pages
4. JavaScript is a scripting language (a scripting language is a lightweight programming language)
5. A JavaScript is usually embedded directly into HTML pages
6. JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
7. Example
    a. 
```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```
8. Syntax
    a. Semicolon optional
    b. Variables
        i. Variables are used to store data.
        ii. A variable is a "container" for information you want to store. A variable's value can change during the script.
        iii. Rules for variable names:
            1. Variable names are case sensitive
            2. They must begin with a letter or the underscore character
                a. strname – STRNAME (not same)
        iv. Variables exercise
            1. var x = 5 + 2 + 3;
            2. var x = "John" + " " + "Doe";
            3. var x = 2 + 3 + "5";?
            4. var x = “5" + 2 + 3;
        v. Scope
            1. GLOBAL
                a. A variable declared outside a function.

b. If you assign a value to a variable that has not been declared, it will automatically become a GLOBAL variable.

2. Local - Variables declared within a JavaScript function

c. Arithmetic Operators

i.

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

d. Assignment Operators

i.

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

e. Comparison Operators

i.

Operator	Description	Example
==	Is equal to	5==8 returns false
===	Is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	Is not equal	5!=8 returns true
>	Is greater than	5>8 returns false
<	Is less than	5<8 returns true
>=	Is greater than or equal to	5>=8 returns false
<=	Is less than or equal to	5<=8 returns true

f. Logical Operators

i.

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
||	or	x=6 y=3 (x==5 || y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

g. Usages
   i. document.getElementById("demo").innerHTML = “test”
   ii. JavaScript Output
      1. innerHTML
      2. document.write()
      3. window.alert()
      4. console.log()

h. Comments:
   i. // Change heading:

ii. /* The code below will change the heading with id = "myH" and the paragraph with id = "myP" in my web page: */

9. Data Types
    a. Strings - var carName = "Volvo XC60";
        i. Functions
            1. str.length;
            2. str.indexOf("locate"); - return position else -1
            3. str.lastIndexOf("locate");
            4. str.search("locate"); - this can accept regular expression
            5. str.replace("Microsoft", "W3Schools");
    b. Numbers - var x1 = 34.00;
    c. Booleans - var x = true
    d. Arrays - var cars = ["Saab", "Volvo", "BMW"];
    e. Objects - var person = {firstName:"John", lastName:"Doe", age:50, Color:"blue"};
10. Functions

```
function name(parameter1, parameter2, parameter3) {
    code to be executed
}
```

    a.
11. Events
    a. <button onclick="displayDate()">The time is?</button>
    b. Onchange
    c. Onclick
    d. Onmousehover
    e. Onmouseout
    f. Onkeydown
    g. Onload
12. Math - The JavaScript Math object allows you to perform mathematical tasks on numbers. Like power, round etc
13. Date - The Date object lets us work with dates.
    a. new Date()
14. Conditional Statements
    a. If
    b. If, else
    c. If, if else
    d. If, else if, else
    e. Switch
15. Loop
    a. For

```
for (i = 0; i < cars.length; i++) {
    text += cars[i] + "<br>";
}
```

        i.
    b. For in

```
for (x in person) {
    text += person[x];
}
```

i.

c. While
d. Do while

16. JavaScript Break and Continue
    a. Break - The break statement can used to jump out of a loop/ switch.
    b. Continue - The continue statement breaks one iteration
17. Error handling
    a. Try catch

```
try {
    Block of code to try
}
catch(err) {
    Block of code to handle errors
}
finally {
    Block of code to be executed regardless of the try / catch result
}
```

i.

18. JavaScript Debugging
    a. The debugger keyword stops the execution of JavaScript
19. JavaScript Coding Conventions
    a. camelCase for identifier names (variables and functions)
    b. Always put spaces around operators (= + - * /), and after commas
    c. Always use 4 spaces for indentation of code blocks
    d. Always end a simple statement with a semicolon.
    e. Global variables written in UPPERCASE
    f. Constants (like PI) written in UPPERCASE
20. Form validation
    a. The purpose of data validation is to ensure correct user input.
        i. Server side validation - using backend language
        ii. Client side validation - using frontend language like javascript

```
function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Name must be filled out");
        return false;
    }
}
```

iii.

1. onsubmit="return validateForm()

21. HTML Dom
    a. With the HTML DOM, JavaScript can access and change all the elements of an HTML document.

b. document.getElementById(id)
c. document.getElementsByTagName(name)
d. document.getElementsByClassName(name)
e. element.innerHTML = new html content
f. element.attribute = new value
g. element.style.property = new style
h. Document.forms
i. document.getElementById("myBtn").addEventListener("click", displayDate);

22. Pop Ups
    a. window.alert("sometext");
    b. window.confirm("sometext"); - returns true or false
    c. window.prompt("sometext","defaultText");
23. JS Ajax

```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
     document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
```

a.

Assignments:

1. Create a form and add javascript validations on submit
    a. Name(Required)
    b. email(Required, should be of type email)
    c. age(Required and should support only positive integers less than 110)
    d. Mobile number(Not required, should only support indian numbers)
2. Create a simple calculator
3. Create a traffic signal light which on/ off in periodic time(Green, Yellow, Red). Also provide below provisions:
    a. Turn off all the lights
    b. Turn on just Green
    c. Turn on just Red
    d. Always blink yellow
    e. Back to normal traffic signal
4. Populate data from the API http://services.groupkt.com/country/get/all to a designed html
5. Create a manual clock
6. Create a form with below details:
    a. Fields:
        i. Name

ii. Email
iii. Country
iv. Region(State)

b. Notes:
i. Country should be populated using the API mentioned here https://battuta.medunes.net
ii. Region(State) should be populated based on the country selected by the user(Use API)
iii. API documentation - https://battuta.medunes.net/#
iv. You can signup and get a token

7. Implement country, region auto populating in the form with JSONP https://www.w3schools.com/js/js_json_jsonp.asp
    a. Use API example - https://battuta.medunes.net/api/country/all/?key=5c41283e72b66936d3f2cb839202b623&callback=country
    b. Mentioned the documentation in the bottom - https://battuta.medunes.net/