JAVA SCRIPT

9/07/2020

Scripting Language

- Scripting Language is a form Programming language which is already compiled.
- It only gets interpreted rather than compiled

JAVASCRIPT

- Javascript is a modern programming language use to develop web pages which are interactive.

FEATURES OF JAVASCRIPT

- Js is a lightweight, interpreted programming Lan
- Designed for creating network-centric application
- Complementary to and integrated with Java.
- complementary to and intergrated with HTML
- Open and cross platform
- It is an interpreted programming Language with object-oriented capabilities.

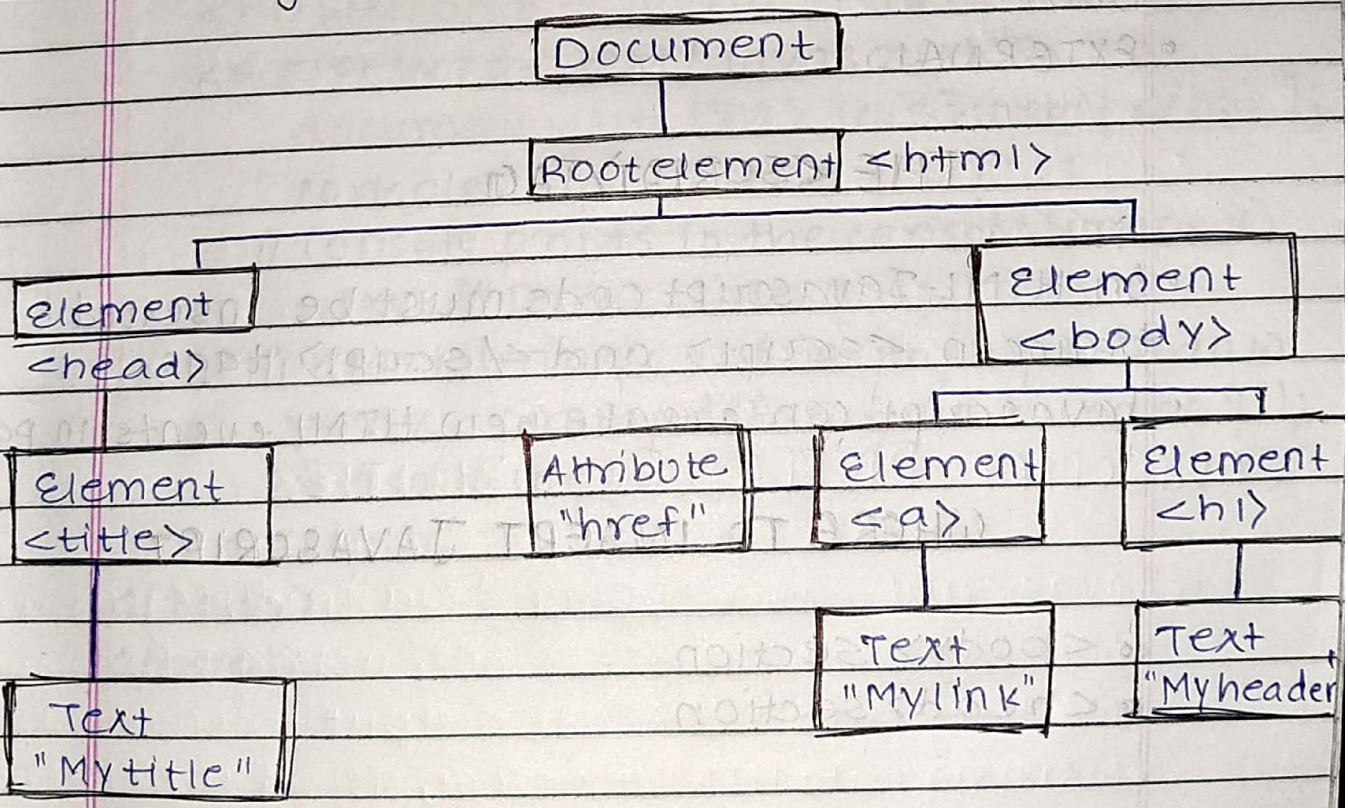
WHAT CAN WE DO WITH JAVASCRIPT

- To create intesactive user interface in web pag
- Eg: menu, pop up alert, windows etc.
- Manipulating web content dynamically.
  - Change the content & style of an element.
  - Replace images on a page without pagereload
  - Hide/show contents.
- Generate HTML contents on the fly.
- form validation.

Javascript can be written within the head tag and also body tag. It can also be written in a different file (external) with a dot js extension and linked back to the main Page

THE DOM

- DOM stands for Document Object Model.
- ~~DOM is the structure is a tree of how your browser~~
- The HTML DOM is a structure / tree of objects which your browser uses to render your webpage



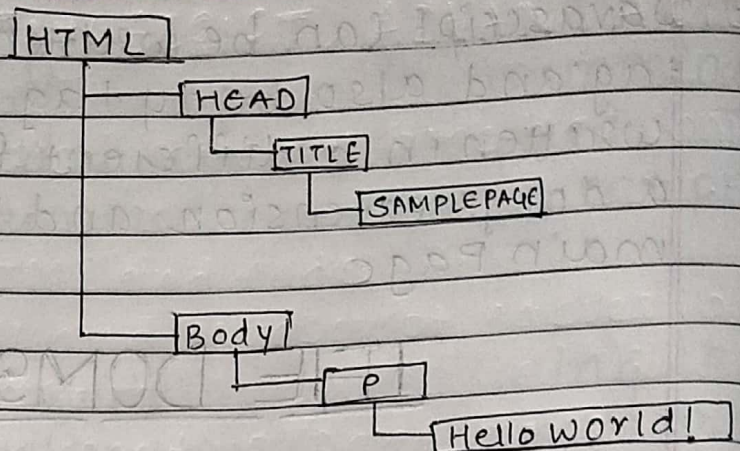
- When a Web-page is loaded, the browser creates a DOM of a page

- With the Object Model, Javascript gets all the power it needs to create Dynamic HTML

PAGE NO.:
DATE: / /

DOM MODEL

```
<html>
<head>
<title>SamplePage
</title>
</head>
<body>
<p>Hello World</p>
</body>
</html>
```



WAYS TO USE JS

- HEAD
- BODY
- EXTERNAL

THE <SCRIPT> TAG

- In HTML, Javascript code must be inserted between <script> and </script> tags.
- Javascript can create new HTML events in page.

WHERE TO INSERT JAVASCRIPT

- <body> section
- <head> section

PAGE NO.:
DATE: / /

EXTERNAL JAVASCRIPT

- Script can also be placed in external files.
- External scripts are practical when the same code is used in many different web-pages.
- Javascript files have the file extension .js.

Index.html

```
<!DOCTYPE html>
<html>
<head>
   <title> My first Javascript </title>
</head>
<body>
   <h1> Welcome to world of JS! </h1>
   <script type="text/javascript">
      document.write("<h2> This work! </h2>");
      console.log("Where am I ?");
     // (console prints in the console (inspect)
      alert("Drink water!");
     // alert brings out a popup window
      var x = prompt("Are you human?");
      console.log(x);
   <script>
</body>
</html>.
```

PAGE NO.:
DATE: / /

EXTERNAL JAVASCRIPT ADVANTAGES

- Placing Javascripts in external files has some advantages:
  - It seperates HTML and code
  - It makes HTML and Javascript easier to read and maintain
  - Cached JS can speed up page loads

INDEX.HTML

```
<body>
  <h1> Welcome to the world of JS </h1>
  <script src="index.js" type="text/javascript">
  </script>
</body>
</html>
```

INDEX.JS

```
alert("This is working, external JS WOW")
```

JAVASCRIPT DISPLAY POSSIBILITIES

- Javascript can display data in different ways
  - Writing into an alert box, using window.alert
  - Writing into the HTML output → document.write
  - Writing into HTML element, using INNER HTML
  - Writing into browser console, using console.log

PAGE NO.:
DATE: / /

INNER HTML

- To access an HTML element, Javascript can use the document.getElementById(id) method.
- The id attribute defines the HTML element.

Example.

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML
                                =5+6;
</script>
```

OUTPUT: 11

CONSOLE.LOG() → console.log()

- On yous browser, you can use the console.log() method to display data.
- Activate the browsers console and select console in the menu.

Example:

```
<script>
console.log(5+6);
</script>
```

Output:

On console section.

```
> 11
```

PAGE NO.:
DATE:

alert()

```
alert ("Text to be displayed");
```

- Display a message in a dialog box.
- The dialog box will block the browser.

confirm()

```
var answer = confirm ("Are you sure?");
```

- Display a message in a dialog box with two buttons: 'OK' or 'Cancel'.
- confirm() returns true if the user clicks "OK", otherwise it returns false

prompt()

```
prompt ("What is your id number?");
```

- Display a message & allows user to enter a value
- The second argument is the "default value" to be displayed in the input textfield.
- Without the default value, "undefined" is shown in the input textfield.
- If the user clicks the 'OK' button, prompt() returns the value in the input textfield as str
- If the user click the "Cancel" button, prom() returns null.

PAGE NO.:
DATE: / /

VARIABLES

- Variables can be thought of as named containers
- We can place data into these containers and then refer to the data simply by naming the containers
- Before we use variable in JS program, we must declare it.
- Variables are declared with var keyword.

Example: index.js

```
var bool = confirm ("Are you sure about this");
if (bool==true) {
    alert ("thanks for the confirmation");
}
else {
    alert ("ok");
}
```

OPERATORS

+, -, *, /, %, ++, --
=, +=, -=, *=, /=, %=

OPERATOR	DESCRIPTION
• typeof	Returns the type of variable
• instanceof	Returns true if an object is an instance of an object type.

DATATYPE

- String
- Number
- Array
- Object.

Eg: index.js (overwrite hota hai)

```
//String character
var myName = "Kishan Chaudhary";
myName = "Sakshi Chudasama";

//Integer datatype
myName = 23;

//Math
myName = myName + 2;

//Boolean datatype
myName = true;

//Printing
document.write(myName);
```

O/P [25]

O/P [True]

PAGE NO.:
DATE: / /

Example → operators

index.html

```
<!DOCTYPE html>
<html>
<head>
<title> My first Javascript </title>
</head>
<body>
   <h1 class="heading"> Welcome </h1>
   <p id="para"> </p>
   <script src=index.js" type=text/js">
   </script>
</body>
</html>
```

index.js

```
document.getElementById("para").innerHTML = "Hi";
var n=10;
var m=5;

//post increment
var opt = m++;
// pre increment
var opt1 = ++n;

// document.write(opt);
// document.write("<br>");
// document.write(m);
document.write("<br>");
document.write(opt1);

document.write("<br>");
document.write(n);
```

PAGE NO.
DATE: / /

increment/decrement sums:

```
var n=10;
var m=5;

var opt=(++m)+(++n)-(n--)-(m--);
document.write(opt);

//++m (opt=6 m=6)
//++n (opt=17 m=6 n=11)
//n-- (opt=6 m=6 n=10)
//m-- (opt=0 m=5 n=10)
```

Ans

(ii)
```
var m=3
var n=2

var opt=(--m)-(m++)-(--n)+(++m)-n+(n++)
```

m=2
m=3
n=1
m=4
n=2

```
:=2-2-1+4-1+1
=3
```

* TYPES OF [typeof() example]

index.js

```
document.getElementById("para").innerHTML="Hi";
var n=2;
var m=3;
document.write(typeof(m));
var m="Rogue code Academy";
document.write("<br>");
document.write(typeof(m));
var m=true;
document.write("<br>");
document.write(typeof(m));
```

O/P
Hi°
number
string
boolean.

DATA-TYPES

1) Primitive datatypes
- Number
- Boolean
- string

2) Composite/complex datatypes
- object
- Array

3) Special datatypes
- Null
- undefined.

STRING

```
var string = "Hello World!" + "Bye";
document.write(string);
```

NUMBER

```
var n = 32;
document.write(n);
```

ARRAY

```
var arr = ["sakshi", "Ankita"];
document.write( arr[0]);
```

OBJECT

```
//object //object Declaration.
var person= {
      name= "Sakshi",
      age= "20",
      favdrink= "bprite" }

//Object calling
document.write(person.name);
```

O/P: Sakshi

JAVASCRIPT

PAGE NO.:
DATE: / /

OBJECT

- Java script objects are written within curly brackets
- Object properties are written as name: value pairs seperated by commas.

```
Eg: var Person = {
        firstName: "John",
        lastName: "Doe",
        age: 50,
        eyeColour: "blue" };
```

Undefined:

- An undefined value is represented by the keyword "undefined".
- It represents the value of an uninitialized variable

Null:

- The keyword 'Null' is used to represent "nothing".
- Declare and define a variable as 'null' if you want the variable to hold nothing.
- Avoid leaving a variable undefined.

FUNCTION

- A Javascript function is a block of code designed to perform a particular task
- function is executed when "something" invokes/calls it