JAVASCRIPT

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 used to develop webpages which are interactive

Features of Javascript

→ JS is a lightweight, interpreted programming language
→ Designed for creating network-centric app.
→ complementary to and integrated with JAVA.
→ complementary ——n—— with HTML.
→ Open & cross platform.
→ It is an interpreted prog lang. with object-oriented capabilities.

What can we Do with Javascript?

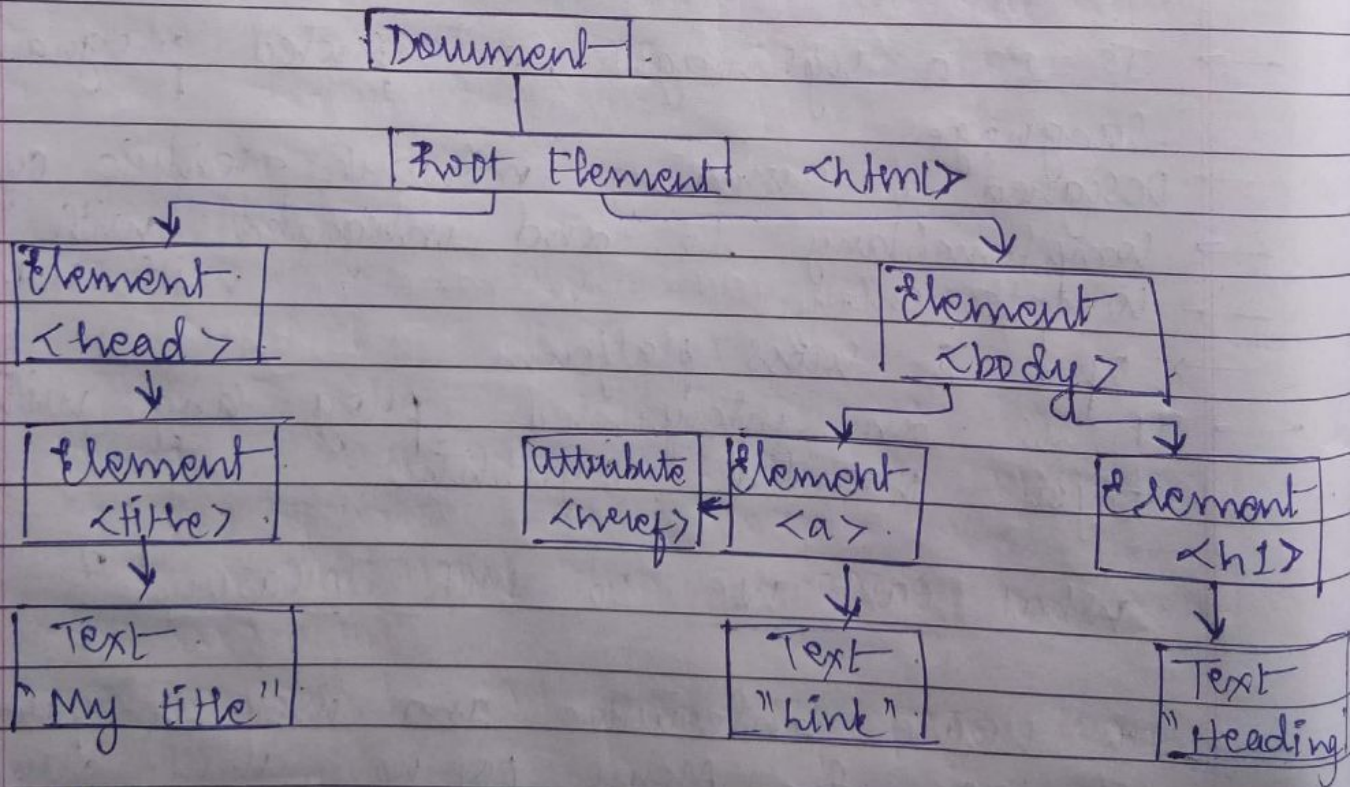
→ To create interactive ~~and~~ user interface in web page Eg: Menu, pop up, alert, windows etc
→ Manipulate web content dynamically.
↳ Change the content & style of an element.
↳ Replace images on a page without page reload
↳ Hide/Show contents
→ Generate HTML contents on the fly.
→ form validation

Scanned with CamScanner

Javascript can be written within the <head> and also <body>. It can also be written in a different file (external) with a dot js (.js) ext. and linked back to the main page.

THE DOM

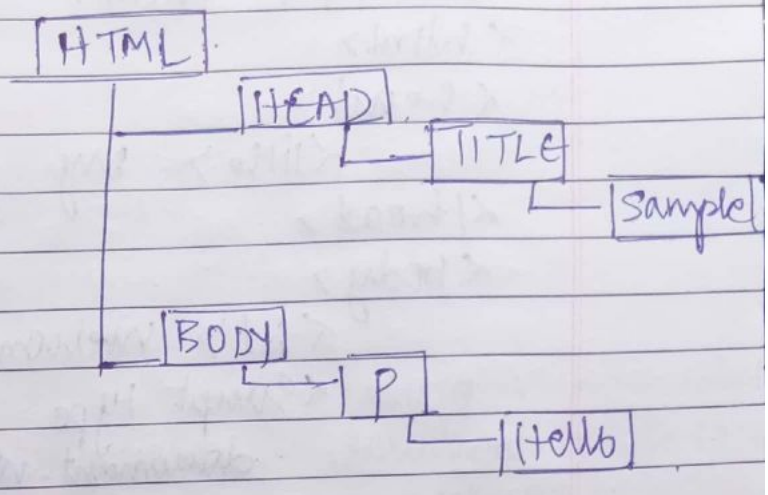
- DOM stands for Document Object Model
- The HTML DOM is a structure/tree of objects which your browser uses to render your webpage.



→ When a webpage is loaded, the browser creates a DOM of a page.
→ With the object model, Javascript gets all the power it needs to creates Dynamic HTML

DOM model

```
<html>
<head>
<title> Sample
</title>
</head>
<body>
  <p> Hello </p>
</body>
</html>
```



WAYS TO USE JS

- Head
- Body
- External

The <script>

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

Where to insert JS

- <body>
- <head>

External JS

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

Scanned with CamScanner

```
<!DOCTYPE html>
<html>
<head>
    <title> My first JS </title>
</head>
<body>
    <h1> welcome </h1>
    <script type = "text/javascript">
    document.write ("<h2> This would </h2>");
    console.log ("Where am I?");
    //console prints in the console (inspect)
    alert ("Drink Water");
    var x = prompt ("Are you human?");
    console.log (x);
    </script>
</body>
</html>
```

EXTERNAL JS ADVTS

→ Placing JS in external files has some advtges:
- → It seperates HTML & code.
- → It makes HTML and JS easier to read & 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>
```

Scanned with CamScanner

index.js

```
alert ("This is working");
```

JS Display possibilities:

→ Writing into an alert box, using window.alert
→ Writing into the HTML output → document.write()
→ Writing into HTML elements, using INNERHTML
→ Writing into browser console, using console.log()

innerHTML

→ To access an HTML element, JS can use document.getElementById(id)

Eg:

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

output → 11

console.log()

→ In your browser, you can use the console.log() method to display data.
→ Activate the browser console and select console in the menu.

Scanned with CamScanner

Example:

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

output:

In console section (Inspect):

>11

alert()

```
alert ("Message");
```

→ Display 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 button: "OK" or "Cancel".
→ confirm() returns true if the user click "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

Scanned with CamScanner

input text field.

→ If the user clicks the 'OK' button, prompt() returns the value in the input textfield as string.

→ If the user click the 'Cancel' button, prompt() returns null.

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 container.

Operators:

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

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

Operator	Description
① typeof	Returns the type of variable
② instanceof	Returns 'true' if an obj is an instance of an Obj type

DATATYPE

- String
- Number
- Array
- Object

Scanned with CamScanner

(Overwrite hota hai)

```
var myName = "Kishan chaudary";
myName = "Aparna Nain" ;  // String
myName = 23 ;             // Number
myName = true ;           // Boolean
```

Eg:

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

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

opt = 5 m = 6.

opt1 = 11 n = 11.

```
var m = 3.
var n = 2.

var opt = (--m) - (m++) - (--n) + (++m) - n + (n++)
        = 2 - 2 - 1 + 1 - 1 + 1
        = 3.
```

Scanned with CamScanner

Datatype:

1) Primitive datatype
   - Number
   - Boolean
   - String

2) composite / complex datatypes
   - Object
   - Array

3) Special datatypes
   - Null
   - undefined

STRING

```
Var String = "Hello " + "Bye" ;
document.Write (String);
```

NUMBER

```
Var n = 32;
```

ARRAY

```
Var arr = ["Sakshi", "Aparna"];
document.Write (arr [0]);
```

Object

```
Var person = {
        name = "Aparna",
        age = "20",
        favdrink = "Chai"
}
```

Scanned with CamScanner

//object calling
document.write(person.name);

Object:

→ Javascript objects are written within curly brackets.
→ Objects properties are written as name: value pa
seperated by commas.

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

Undefined:

→ An undefined value is represented by the ~~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.

Scanned with CamScanner

Function:

→ A JS. funct is a block of code designed to perform a particular task.

→ function is expected when "something" invokes / calls it.

Scanned with CamScanner