SCRIPTING LANGUAGE

Scripting language is a form Programming language which is already pre compiled. i.e It only get interpreted rather than compiled.

JAVASCRIPT LANGUAGE

Javascript is a modern programming language used to develop webpages which are interactive. It is an interpreted programming language with object-oriented language

FEATURES

- Javascript is a light weight, interpreted progeraming language
- Designed for creating network-centric application
- complementary to and integrated with Java & HTML
- Open and cross platform.

WHAT CAN WE DO WITH JAVASCRIPT

- To create interactive user interface into a web page
  - e.g, menu, pop-up alert, windows etc.

- Manipulating web content dynamically
  - change the content and style of an element
  - Replace images on a page without page reload
  - Hide/Show contents.

inline, internal, external.
↓ ↓
<style> script

PAGE No.
DATE

Javascript can be written within the head tags and also within body tag. It can also be written in a different file (external file) with a .JS (dot JS) extension and linked backed to the main page.

console.log → debugging

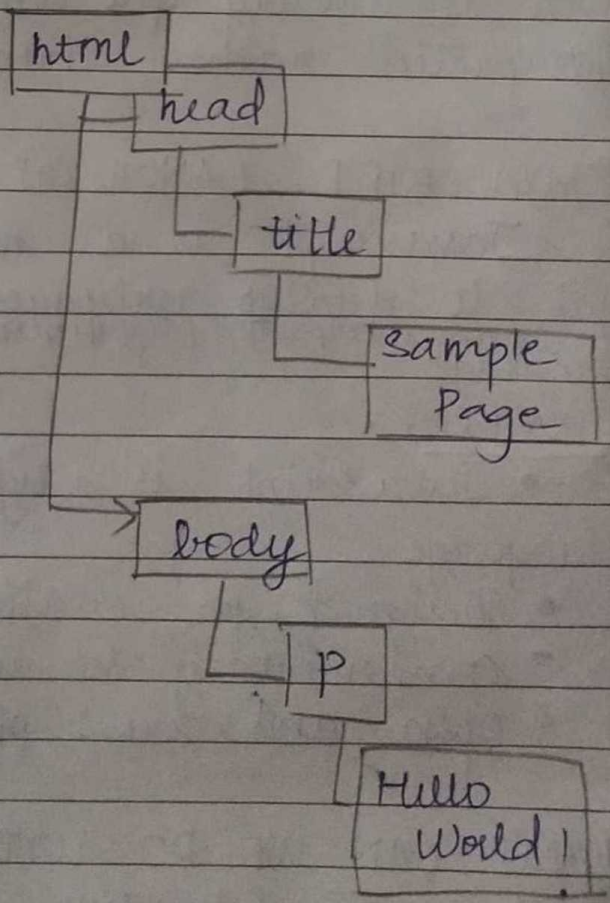
DOCUMENT OBJECT MODEL

The document object model (HTML) is a 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.

```
<html>
<head>
<title> Sample Page </title>
</head>

<body>
<p> Hello World! </p>
</body>
</html>
```



PAGE No.
DATE

WAYS TO USE JAVASCRIPT (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 the page.

EXTERNAL JAVASCRIPT

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

ADVANTAGES

Placing JavaScripts in external files have some advantages:

- It seperates HTML and code.
- It makes HTML and JavaScript easier to read and maintain
- Cached JavaScript

EXAMPLE

```
<!Doctype html>
<html>
<head>
<title> My first Javascript </title>
</head>
<body>
    <h1> Welcome to the world of JS! </h1>
    <script src = "index.js" type = "text/javascript"></
                                                script>
</body>
```

```
var myName = "Kishan chaudary";
document.write();
```

VARIABLES

Every javascript variables starts with var.
ES6 – let / const

EXAMPLE

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

EXAMPLE OF USING JS IN BODY.

```
<!DOCTYPE HTML>
<html>
 <head>
     <title> My first Javascript </title>
 </head>
 <body>
     <h1> Welcome to the world of JS! </h1>
     <script type = "text / javascript">
     document. write ("<h2> This Works ! </h2>");

     console . log ("Where am 99");

     alert ("Drink Water! ");
     var x = prompt ("Are you Human?");
     console . log (x)
     </script>
 </body>
</html>.
```

INNER HTML

```
<body>
  <h1 class = "heading"> Welcome to the world of JS! </h1>
  <p id = "para" </p>

<script src = "index.js" ~~text~~ type = "text/javascript">
</body>
```

```
document.getElementById("para").innerHTML = "Hi";
```

OPERATORS

Operator	Example	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.

Operator	Description
Typeof	Returns the type of a variable
instance of	Returns true if an object is an instance of an object type.

n = 2
m = 3

PAGE No.
DATE

DATATYPES

- PRIMITIVE DATA TYPE
  - NUMBER : integer & | floating-point numbers
  - BOOLEAN : true or false
  - STRING : a sequence of alphanumeric characters

-
  - OBJECT : a named collection of data
  - ARRAY : a sequence of values (an array is actually a predefined object)

.
  - NULL : the only value is "null" - to represent nothing
  - UNDEFINED : the only value is "undefined" - To represent the value of an uninitialized variable

```
var length = 16;                                    // Number
var lastName = "Johnson";                           // String
var cars = ["Saab", "Volvo", "BMW"];                // Array
var x = {firstName : "John", lastName : "Doe"};     // Object
```

STRINGS

- A string is a series of characters and are written with quotes.
- you can use single or double quotes:
  - var answer = "It's alright";
    // Single quote inside double quotes
  - var answer = "He is called "Johnny"";
    // Single quotes inside double quotes
  - var answer = 'He is called "Johnny"';
    // Double quotes inside single quotes

- EXAMPLE 1.

```
// strings
// string concatenation;
var str = "Hello string!" + " Bye";
document.write(str);
```

- EXAMPLE 2

```
// Arrays
var arr = ["Sakshi", "Neha"];
document.write(arr[0]);
```

OBJECT	PROPERTIES	METHODS
	car.name = Fiat	car.start()
car	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

```
var car = {type:"Fiat", model:"500", color:"white"};
```

OBJECTS

- JavaScript objects are written with curly braces.
- Object properties are written as name : value pairs seperated by commas.

```
var person =
  {
    firstName : "John",
    lastName : "Doe,
    age : 50,
    eyeColor : "blue".
  };
```

- EXAMPLE.

```
// Objects
// Object declaration

var person = {
      name :" priana";
      realAge : " 45";
      fakeAge : " 25";
      favDrink : " Vodka";
      favColor : " Black";
      favLanguage : " Java";

  }

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

UNDEFINED AND NULL

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

- 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.

FUNCTIONS

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

FUNCTION INVOCATION

- The code inside the function will execute when "something" invokes (calls) the function:
  - When an event occurs (when a user clicks a button)
  - When it is invoked (called) from JavaScript code
  - Automatically (self invoked)

docker
containizing

hello → Ram

EXAMPLE

1) var a = prompt ("enter a number");
var b = prompt ("enter a number");

function sum(a,b)
{
return a+b;

}// printing onto screen
document.write (sum(3,2));

2) var a = parseInt (prompt (" "));
var b = parseInt (prompt (" "));

EVENTS

- HTML events are "things" that can happen to HTML elements.
- When Javascript is used in HTML pages, Javascript can "react" on this events.
- An HTML event can be something the browser does, or something a user does.
- EXAMPLE:

```
<body>
<p id = "para"></p>
<button onclick = "btnRed()"> Red </button>
<button onclick = "btnGreen()"> green </button>
<button onclick = "btnBlue()"> Blue </button>
<button onclick = "btnYellow()"> yellow </button>
<script src = "index.js" type = "text/javascript">
</script>

</body>
```

```
function btnRed(){
document.body.style.background = 'red';
}

function btnBlue(){
document.body.style.background = 'blue';
function btnYellow(){
document.body.style.background = 'yellow';
}
function btnGreen(){
document.body.style.background = 'green';
}
```