1. **For the given JSON iterate over all for loops(for, for in, for of, for Each)**

**Ans:**

**For Loop:**

**Input:**

var arr=[{

"name":"john doe",

"age":"23",},

{"name":"mary",

"age": "24"}];

for(var i=0;i<arr.length;i++){

console.log(arr[i].name,arr[i].age);

}

**Output:**

john doe 23

mary 24

**For In Loop:**

**Input:**

var arr=[{

"name":"john doe",

"age":"23",},

{"name":"mary",

"age": "24"}];

for(var i=0;i<arr.length;i++){

for(var a in arr[i]){

console.log(a,arr[i][a]);

}

}

**Output:**

name john doe

age 23

name mary

age 24

1. **Create your own resume data in JSON format**

**Ans:**

**Input:**

var arr=[{

"name": "anirutha",

"age":"26",

"gender": "female",

"nationality": "indian",

“languages known”: “English,tamil”}];

for(var i=0;i<arr.length;i++){

for(var a in arr[i]){

console.log(a,arr[i][a]);

}

}

**Output:**

name anirutha

age 26

gender female

nationality indian

languages known English,tamil

1. **Read about difference between window, screen and document in JavaScript**

**Ans:**

**Window**

The JavaScript **window object** sits at the top of the JavaScript Object hierarchy and represents the browser window. The window object is supported by all browsers. All global **JavaScript objects** , functions, and variables automatically become members of the window object. The window is the first thing that gets loaded into the **browser** . This window object has the majority of the properties like length, innerWidth, innerHeight, name, if it has been closed, its parents, and more.

The window object represents the current **browsing context** . It holds things like window.location, window.history, window.screen, window.status, or the **window.document** . Each browser tab has its own top-level window object. Each of these windows gets its own separate global object. window.window always refers to window, but **window.parent** and window.top might refer to enclosing windows, giving access to other execution contexts. Moreover, the window property of a window object points to the window object itself. So the following statements all return the same window object:

window.window  
window.window.window  
window.window.window.window  
and so on

**Window Properties:**

Window object has two properties to determine the size of the browser window. They are:

***window.innerHeight*:**gives the inner height of the browser window (in pixels)

***window.innerWidth***: gives the inner width of the browser window (in pixels)

**Window methods:**

Some window object methods are:

***window.open()***: open a new window

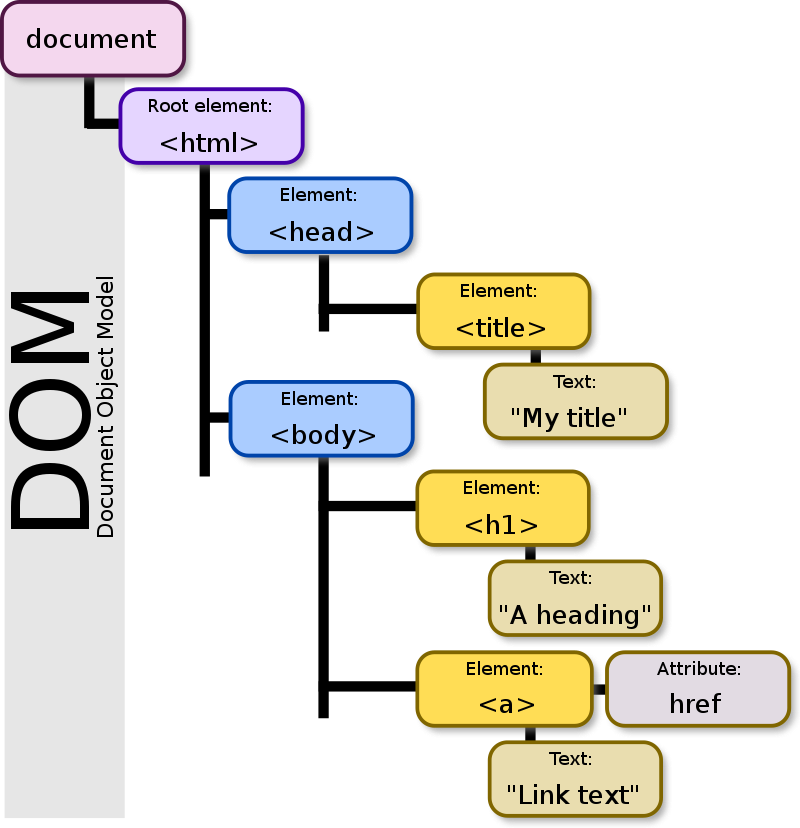
***window.close()***: close the current window

***window.moveTo()***: move the current window

***window.resizeTo()***: resize the current window

**Document**

The **Document object** represents any web page loaded in the browser and serves as an entry point into the web page’s content, which is the DOM tree. When an HTML document is loaded into a **web browser** , it becomes a document object. It is the root node of the HTML document. The document actually gets loaded inside the window object and has properties available to it like title, URL, cookie, etc.



**Finding HTML elements:**

We can find the HTML elements by using the below**document object**methods:

***document.getElementById(id)*** : Find and return an element by element id

***document.getElementsByTagName(name)***: Find and return an element by *tag name*

***document.getElementsByClassName(name)***: Find return an element by class name

**Changing HTML elements:**

We can change the HTML element contents like style, text, attribute using the below properties:

***element.innerHTML = new html content*:**Change the inner HTML of an element

***element.attribute = new value*:**Change the attribute value of an HTML element

***element.style.property = new style* :**Change the style of an HTML element

**Adding and Deleting HTML elements:**

We can create, add, delete and replace HTML elements by using the below methods:

***document.createElement(*element*)*:**Create an HTML element

***document.removeChild(*element*)*:**Remove an HTML element

***document.appendChild(*element*)*:**Add an HTML element

***document.replaceChild(*new, old*)*:**Replace an HTML element

***document.write(*text*)*:**Write into the HTML output stream

**Adding Event Handlers:**

We can also add event handlers when a specific event occurs like onclick, onload, onkeydown etc by using the respective event property:

***document.getElementById(id).onclick = function(){code} :****Adding event handler code to an onclick event*

**Screen**

Screen is a small information object about physical **screen dimensions**of the user device. It can be used to display screen width, height, colorDepth, pixelDepth etc. It is not mandatory to write **window prefix** with screen object like **window.screen**. It can be written without window prefix.

**Properties:**

***screen.width  
screen.height  
screen.availWidth  
screen.availHeight  
screen.colorDepth  
screen.pixelDepth***