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

For loop

var employee = [

{

id: 101,

name: "Joy",

Designation: "Manager",

year: 10,

},

{

id: 102,

name: "Raj",

Designation: "Developer",

year: 8,

},

{

id: 103,

name: "Raju",

Designation: "Linux Administrator",

year: 6,

},

{

id: 104,

name: "Rajesh",

Designation: "Senior Developer",

year: 12,

},

{

id: 105,

name: "Jackey",

Designation: "Junior Developer",

year: 1,

}

];

For(let i=0; i<employee.length-2 ;i++){

Console.log(employee[i]

}

Output:

101

Joy

Manager

10

102

Raj

Developer

8

103

Raju

Linux Administrator

6

for in loop

var employee = [

{

id: 101,

name: "Joy",

Designation: "Manager",

year: 10,

},

{

id: 102,

name: "Raj",

Designation: "Developer",

year: 8,

},

{

id: 103,

name: "Raju",

Designation: "Linux Administrator",

year: 6,

},

{

id: 104,

name: "Rajesh",

Designation: "Senior Developer",

year: 12,

},

{

id: 105,

name: "Jackey",

Designation: "Junior Developer",

year: 1,

},

{

id: 106,

name: "Vignesh",

Designation: "Junior Developer",

year: 1,

},

];

for(var nan in employee){

console.log(employee[nan])

}

Output:

id:101

name:Joy

designation:undefined

year:10

id:102

name:Raj

designation:undefined

year:8

id:103

name:Raju

designation:undefined

year:6

id:104

name:Rajesh

designation:undefined

year:12

id:105

name:Jackey

designation:undefined

year:1

id:106

name:Vignesh

designation:undefined

year:1

for of loop

let vlue ={

id: 107,

name: "ganesh",

Designation: "Junior Developer",

year: 1,

}

for(let vl of vlue){

console.log(v1)

}

Output: 107

ganesh

Junior Developer

1

forEach

var employee = [

{

id: 101,

name: "Joy",

Designation: "Manager",

year: 10,

},

{

id: 102,

name: "Raj",

Designation: "Developer",

year: 8,

},

{

id: 103,

name: "Raju",

Designation: "Linux Administrator",

year: 6,

},

{

id: 104,

name: "Rajesh",

Designation: "Senior Developer",

year: 12,

},

{

id: 105,

name: "Jackey",

Designation: "Junior Developer",

year: 1,

},

{

id: 106,

name: "Vignesh",

Designation: "Junior Developer",

year: 1,

},

];

employee.forEach(function(emp){

console.log(emp.id, emp.name, emp.year)

})

Output:

101 'Joy' 10

102 'Raj' 8

103 'Raju' 6

104 'Rajesh' 12

105 'Jackey' 1

106 'Vignesh' 1

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

**let resume=**

**{**

**name:"ganesh kalyan",**

**mobile:9966775533,**

**email:"ganeshkalyan506@gmail.com",**

**degree:{**

**name\_of\_the\_college:"vignanasudha degree college",**

**department:"BSC",**

**gap:7.6,**

**passout\_year:2021,**

**intermidiate:{**

**name\_of\_the\_college:"sri chaitanya junior college",**

**department:"MEC",**

**gap:7.9,**

**passout\_year:2018,**

**SCC: {**

**name\_of\_the\_school:"mathakonda palli model school",**

**gap:8,**

**passout\_year:2016,**

**},**

**},**

**},**

**hobbies:["reading books","football","music","sports"],**

**languagesknown:["tamil","telugu","english"],**

**technicalskills:["c","c++","java"],**

**adress:{**

**door\_no:9-262,**

**street\_name:"ghandhi road",**

**city:"chittoor",**

**state:"Andhra Pradesh",**

**}**

**}**

**console.log(`name: ${resume.name}**

**mobile: ${resume.mobile}**

**email: ${resume.email}**

**EDUCATION\_DETAILS:**

**degree:**

**NAME\_OF\_THE\_UNIVERSITY: ${resume.degree.name\_of\_the\_college}**

**DEPARTMENT: ${resume.degree.department}**

**GAP: ${resume.degree.gap}**

**PASSOUT: ${resume.degree.passout\_year}**

**intermidiate:**

**NAME\_OF\_THE\_UNIVERSITY: ${resume.degree.intermidiate.name\_of\_the\_college}**

**DEPARTMENT: ${resume.degree.intermidiate.department}**

**GAP: ${resume.degree.intermidiate.gap}**

**PASSOUT: ${resume.degree.intermidiate.passout\_year}**

**SCC:**

**NAME\_OF\_THE\_SCHOOL: ${resume.degree.intermidiate.SCC. name\_of\_the\_school}**

**GAP: ${resume.degree.intermidiate.SCC. gap}**

**PASSOUT: ${resume.degree.intermidiate.SCC. passout\_year}**

**hobbies: ${resume.hobbies}**

**technicalskills: ${resume.technicalskills}**

**languagesknown: ${resume.languagesknown}**

**`)**

**Output:**

name: ganesh kalyan

mobile: 9966775533

email: ganeshkalyan506@gmail.com

EDUCATION\_DETAILS:

degree:

NAME\_OF\_THE\_UNIVERSITY: vignanasudha degree college

DEPARTMENT: BSC

GAP: 7.6

PASSOUT: 2021

intermidiate:

NAME\_OF\_THE\_UNIVERSITY: sri chaitanya junior college

DEPARTMENT: MEC

GAP: 7.9

PASSOUT: 2018

SCC:

NAME\_OF\_THE\_SCHOOL: mathakonda palli model school

GAP: 8

PASSOUT: 2016

hobbies: reading books,football,music,sports

technicalskills: c,c++,java

languagesknown:tamil,telugu,english

3.Read about the difference between window, screen and document in javascript.

Window.

Window is the main JavaScript object root, via the global object in a browser, and it can also be treated as the root of the document object model. You can access it as window.

Each browser tab has its own top-level window object. Each <iframe> (and deprecated <frame>) element has its own window object too, nested within a parent window. 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. In addition to document and screen described below, window properties include

* setTimeout() and setInterval() binding event handlers to a timer
* location giving the current URL
* history with methods back() and forward() giving the tab's mutable history
* navigator describing the browser software.

Screen

a SCREEN  is a small information object about physical screen dimensions.

The window object also has a screen object with properties describing the physical display:

* screen properties width and height are the full screen
* screen properties availWidth and availHeight omit the toolbar

The portion of a screen displaying the rendered document is the **viewport** in JavaScript, which is potentially confusing because we call an application's portion of the screen a window when talking about interactions with the operating system. The getBoundingClientRect() method of any document element will return an object with top, left, bottom, and right properties describing the location of the element in the viewport.

Document

document is the main object of the potentially visible (or better yet: rendered) document object model/DOM.

Each window object has a document object to be rendered. These objects get confused in part because HTML elements are added to the global object when assigned a unique id. E.g., in the HTML snippet

<body>

<p id="holyCow"> This is the first paragraph.</p>

</body>

the paragraph element can be referenced by any of the following:

* window.holyCow or window["holyCow"]
* document.getElementById("holyCow")
* document.querySelector("#holyCow")
* document.body.firstChild
* document.body.children[0].