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

* For

let student = [ {

    "name" : "Arun",

    "Id" : "2567",

    "dept" : "Computer Science"

  },

                 {

    "name" : "Selva",

    "Id" : "2588",

    "dept" : "Mechanical"

                },

                 {

     "name" : "David",

    "Id" : "25588",

    "dept" : "Civil"

                 }];

  for (let i=0;i<student.length;i++)

    {

      let result = student[i];

      console.log(result.name,result.Id);

    }

\*For Each

let student = [ {

    "name" : "Arun",

    "Id" : "2567",

    "dept" : "Computer Science"

  },

                 {

    "name" : "Selva",

    "Id" : "2588",

    "dept" : "Mechanical"

                },

                 {

     "name" : "David",

    "Id" : "25588",

    "dept" : "Civil"

                 }];

                 student.forEach(function(x) {

                     console.log("NAME: " + x.name);

                     console.log("ID: " + x.Id);

                     console.log("DEPT: " + x.dept);

                 });

* For In

let student = [ {

    "name" : "Arun",

    "Id" : "2567",

    "dept" : "Computer Science"

  },

                 {

    "name" : "Selva",

    "Id" : "2588",

    "dept" : "Mechanical"

                },

                 {

     "name" : "David",

    "Id" : "25588",

    "dept" : "Civil"

                 }];

                 let x = "";

                 for (let key in student) {

                  if (student.hasOwnProperty(key)); {

                    let x = "";

                    console.log(student[key].Id,student[key].name);

                  }

                   }

* For Off
* let student = [ {
* "name" : "Arun",
* "Id" : "2567",
* "dept" : "Computer Science"
* },
* {
* "name" : "Selva",
* "Id" : "2588",
* "dept" : "Mechanical"
* },
* {
* "name" : "David",
* "Id" : "25588",
* "dept" : "Civil"
* }];
* for(let x of student) {
* console.log(x.name,x.Id);
* }
* Create your own resume data in JSON format
* var resume  = [
* {
* "basics": {
* "name": "Rajshankar",
* "label": "Programmer",
* "email": "rajshankar607@gmail.com",
* "phone": "8778087447",
* "location" : {
* "address": "Mettur",
* "postalCode": "636403",
* "city": "Salem",
* "countryCode": "India",
* "state": "Tamilnadu"
* },
* "profiles" : {
* "network": "LinedIn.com",
* "username": "Raj Shankar",
* "url": "https://www.linkedin.com/in/raj-shankar-27a58b15b/"
* },
* "education" : {
* "institution": "Annamalai University",
* "area": "BCA",
* "studyType": "Bachelor",
* "Year": "2020",
* },
* "skills" : {
* "name": "Web Development",
* "level": "Entry Level",
* "keywords": [
* "HTML",
* "CSS",
* "JavaScript"
* ]
* },
* "languages" : {
* "language": "English",
* "fluency": "Native speaker",
* "language": "Tamil",
* "fluency": "Native speaker"
* },
* }}];
* for (var i=0;i<resume.length;i++)
* {
* var obj = resume[i];
* }
* console.log(obj);
* Read about the difference between window, screen and document in javascript

## 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.

**Document**

* The **Document interface** 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.
* HTML documents, served with the **"text/html"** content type, also implement the HTMLDocument interface, whereas XML and SVG documents implement the XMLDocument interface.

## Screen

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