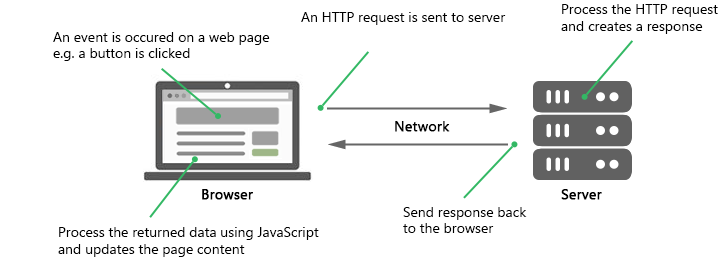
+.3

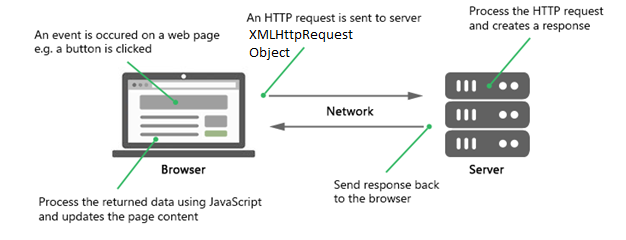
The browser makes HTTP request and send to server when event occurs on webpage. The server receives Http request . The server process the Http request and server makes a response for the request. The response is sent to browser by server.

The browser receives response. After received response,the browser updates complete part[s] of web page.

Drawback: The HttpRequest is synchronous request. By synchronous request the following program happens.

Problem: After sending request, The browser immeidately halt execution of JS code and browser continously reload/ refresh the web page till getting response from server. By the time of refreshing, webpage is unable to interact with user.

**1.AJAX(Asynchronous Java Script And XML):** The ajax is not technology at all. Ajax is just means of getting the data from webserver asynchronously though java script and selectively updating the part[s] of web page with out reloading the entire web page.

****

This AJAX concept is performed by XMLHttpRequest object. This is built-in object and it is available in all browsers. The browser makes the XMLHttpRequest object. It send object to server. After sent,Browser does not halt the JS code execution And webpage is also able to interact with user. When response from server is ready, browser takes it and browser updates part[s] of webpage.

The following online applications are using ajax concept.

Example: google MAP, gmail , facebook, youtube ,..etc.

**2. XMLHttpRequest object:-**

**2.1 . Properties:**

a.onload:- It hold the function. This function is invoked when response is ready.

b.readystate:- It holds status of XMLHttpRequest.

0-request is not initialized.

1-server connection established.

2- request received by server.

3- request is being processed by server.

4- request finished and response ready.

c.onreadystatechange:- It holds function. This function is invoked when readystate value is changed.

d. status:- It returns the status number of request.

200:”ok”

404:”File not found”.

…etc.

e. statusText: -It returns the status text of request. Ex:-( “OK”, “File not Found”).

f.responseText:- It returns response data as Text.

g.responseXML:- It returns response data as XML data.

h.onprogress:- It stores the callback function. It triggers periodically while response is being downloaded,reports how much has been downloaded.

**2.2.Methods:**

a.new XMLHttpRequest():- create a new XML HttpRequest object.

b.abort():- It cancel the XMLHttpRequest.

c.open():- It specifies the request information.

Syntax:

Open(methodname,url,username,pwd);

1. Method name is get or post. The get is used for taking the data from server. Put is used for sending data from program to server.
2. url is address of source which contains the data.

d. send():- send the request to server.

f. send(string):- send the request to server. Where string is data which has to be given to server.

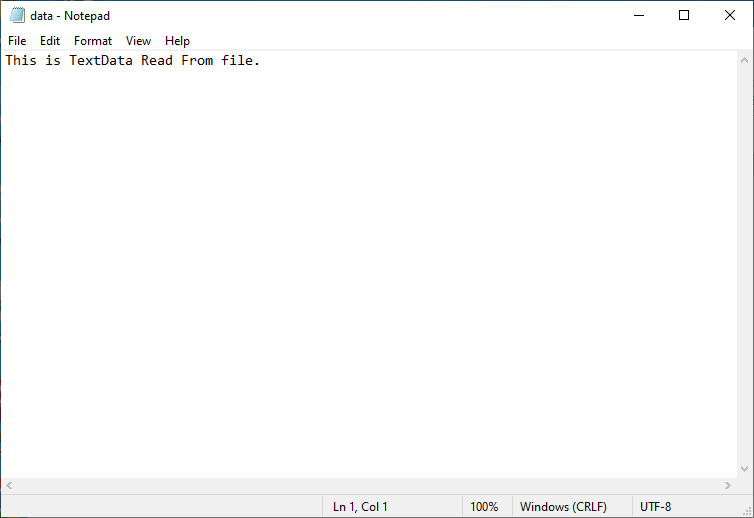
g.setRequest Header:- add a label/value to request header.

h.getAllResponseHeaders():- It returns complete header information.

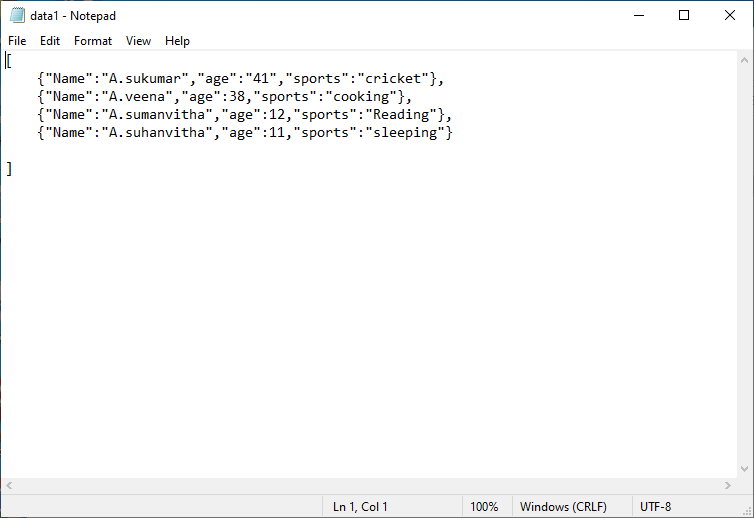
i. getResponseHeader():- It returns specific header information.

Example:

1.data.txt



2.Data1.json



3.sample.json

<!DOCTYPEhtml>

<htmllang="en">

<head>

    <metacharset="UTF-8">

    <metahttp-equiv="X-UA-Compatible"content="IE=edge">

    <metaname="viewport"content="width=device-width, initial-scale=1.0">

    <title>Document</title>

<style>

    body{

        background-color: blue;

    }

    button{

        width:15vw;

        position:relative;

        left:25vw;

        margin:5px2.5px;

        background-color: blueviolet;

        color: azure;

        font-family: 'Times New Roman', Times, serif;

        font-size: medium;

        cursor:pointer;

    }

    div{

        margin-left:10px;

        border:0.5pxsolidblack;

        width:20vw;

        padding:2vh,2vw;

        font-size: 1.2em;

        background-color: brown;

        color:white;

        position: relative;

        left:20vw;

        top:5vh;

        float:left;

    }

</style>

</head>

<body>

    <buttononclick="displayData(event)">TextData</button>

    <buttononclick="displayData(event)">JSONData</button>

    <buttononclick="displayData(event)">APIData</button><br>

    <div>TextData</div>

    <div>JSONData</div>

    <div>APIData</div>

    <script>

        letdbtn=document.querySelectorAll('button');

        letdCon=document.querySelectorAll('div') ;

        functiondisplayData(event)

        {

           if(event.target.innerHTML=='TextData')

           {

               dbtn[1].disabled=true;

               dbtn[2].disabled=true;

               putData(event.target,'data.txt',dCon[0]);

           }

        elseif(event.target.innerHTML=='JSONData')

        {

               dbtn[0].disabled=true;

               dbtn[2].disabled=true;

            putData(event.target,'data1.JSON',dCon[1]);

        }

        else{

            dbtn[1].disabled=true;

            dbtn[0].disabled=true;

            putData(event.target,'data.txt',dCon[2]);

            }

       }

       functionputData(a,b,c)

       {

        letxObj=newXMLHttpRequest();

        xObj.onload=function(){

            if(xObj.status==200)

            {

              if(a.innerHTML=='TextData')

              {

                lety1=document.createElement('hr');

                c.append(y1);

                lety=document.createElement('p');

                y.innerHTML=xObj.responseText;

                y.style='font-size:0.75em';

                c.append(y);

                setTimeout(()=>{dbtn[1].disabled=false;dbtn[2].disabled=false;c.removeChild(y1); y.innerHTML='';},2000);

              }

              else

              {

                letdata=xObj.responseText;

                data=JSON.parse(data);

                if(a.innerHTML=='JSONData')

                {

                 lety1=document.createElement('hr');

                c.append(y1);

                for(letiofdata)

                {

                    for(letjini)

                    {

                     letx= document.createElement('p');

                     x.innerHTML=j+':'+i[j];

                     x.style='text-align:left;font-size:0.75em';

                     c.append(x) ;

                    }

                    c.append(document.createElement('hr'));

                }

                setTimeout(()=>{dbtn[0].disabled=false;dbtn[2].disabled=false;c.removeChild(y1); dCon[1].innerHTML='TextData';},2000);

               }

               else

               {

                lety1=document.createElement('hr');

                c.append(y1);

                for(letiofdata)

                {

                    for(letjini)

                    {

                     letx= document.createElement('p');

                     x.innerHTML=j+':'+[j];

                     x.style='text-align:left;font-size:0.75em';

                     c.append(x) ;

                    }

                    c.append(document.createElement('hr'));

                }

                setTimeout(()=>{dbtn[1].disabled=false;dbtn[0].disabled=false;c.removeChild(y1); dCon[2].innerHTML='APIData';},2000);

               }

              }

            }

        }

        try{

            xObj.open('get',b);

            xObj.send();

        }

        catch(e){

            console.log(e.message);

        }

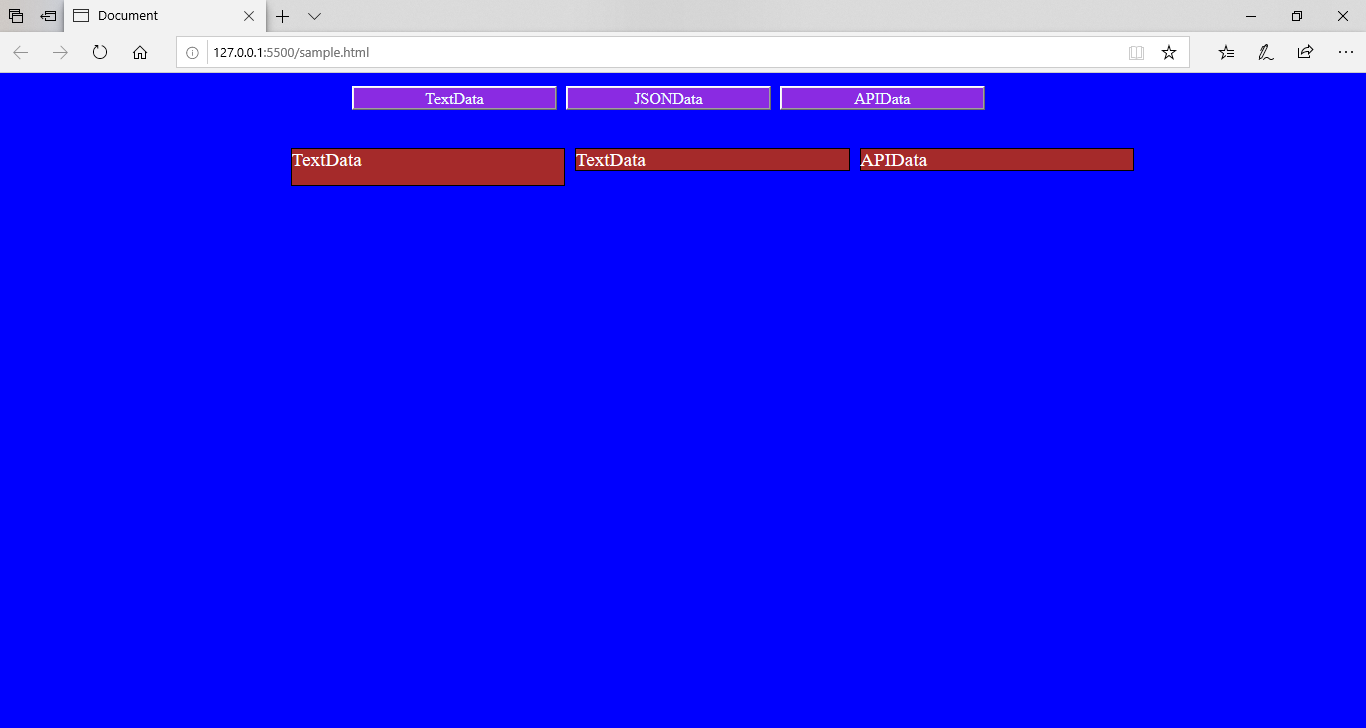
       }

    </script>

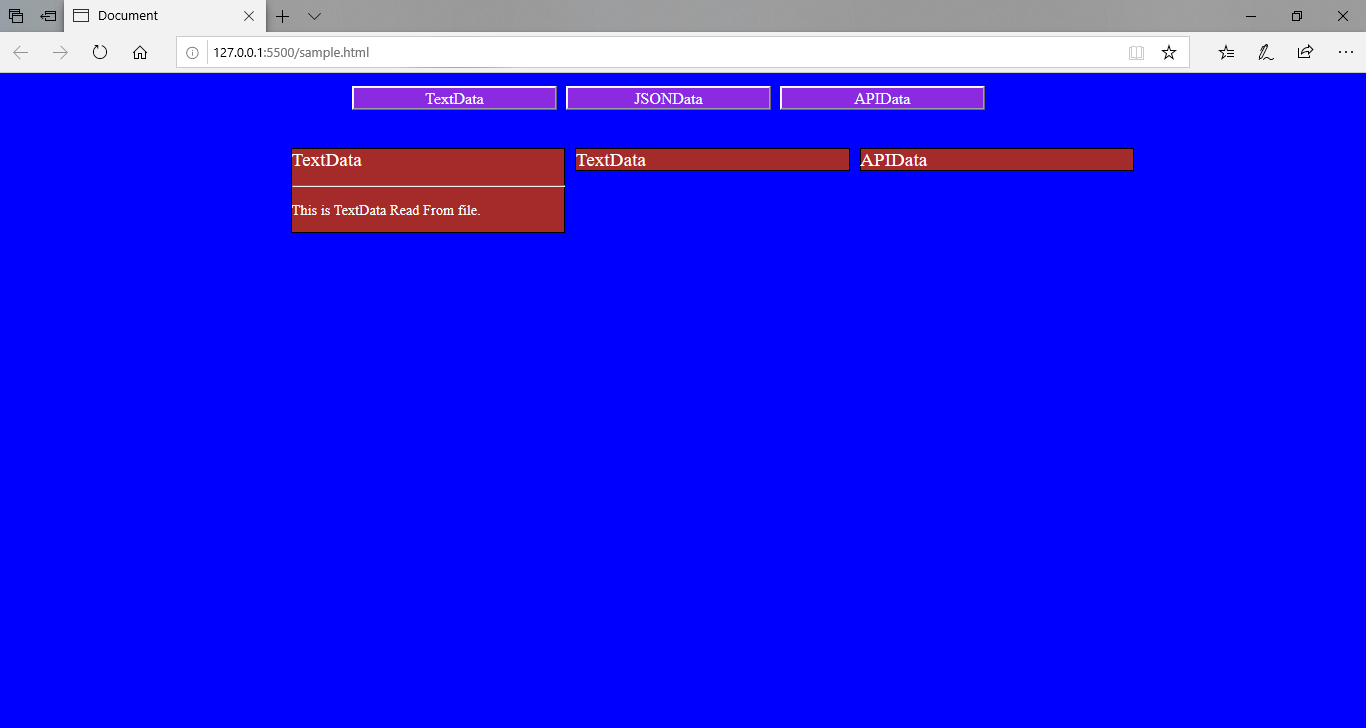
</body>

</html>

Output;

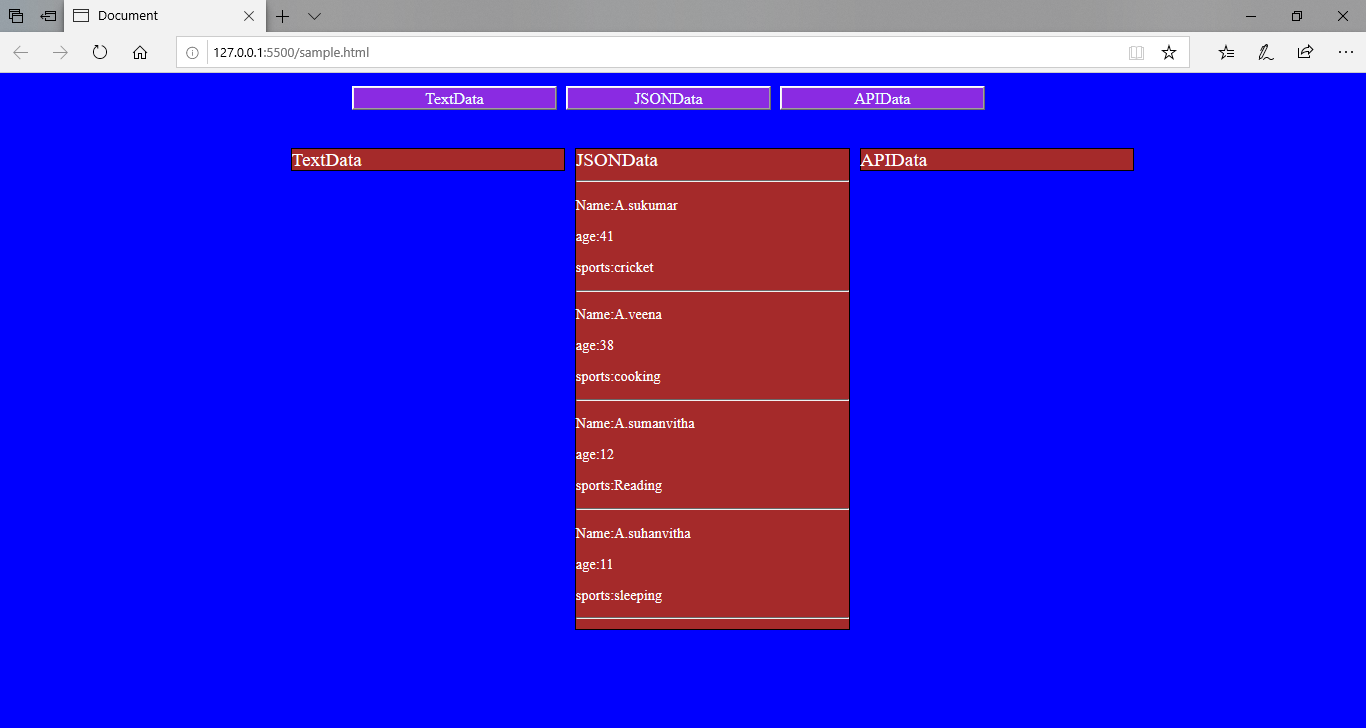


After clicking the TextData button.



After 2 secons textdata is cleared and Remaining two button enabled.

After pressing JSONData button.



Drawback in above Program:

1.we created the template(web page).

2. The JS code has red JSON data from JSON Data or APIData.

3. Especially, we have written logic to post the JSON Data on webpage. If data is simple, logic is also simple. If data is complex and more, programmer has to write logic. Writing takes more time. This is drawback.

This drawback could be overcame by External API’s like **handlebars mustache js**,..etc. If we give template and json object external api’s, then external api’s automaticlly post json data on web page.

Step1: link **handlebars min.js file** to our current html program.

https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.7.7/handlebars.min.js

Step2.handle bar Compile template(webpage) using following syntax.

Var var-name=Handlebars.compile(Template file Name)

Step3. Give the JsonData to handlebar, which post data on web page.

Example:

<!DOCTYPEHTML>

<html>

   <head>

      <scriptsrc="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.7.7/handlebars.min.js"></script>

      <style>

      </style>

   </head>

   <body  >

      <scriptid='one'type="text/x-handlebars-template">

         <div>

           I am{{Name}}. I am {{age}} old. I like {{sports}} playing.

         </div>

     </script>

      <div></div>

      <buttononclick="drawGraphic()"> press</button>

      <script>

         vardata;

         functiondrawGraphic(){

            vara=newXMLHttpRequest();

            a.onload=function(){

              data=a.responseText;

              data=JSON.parse(data);

              vartemp1=Handlebars.compile(document.querySelector('#one').innerHTML);

              document.querySelector('div').innerHTML=temp1(data);

            }

            a.open('get','data1.json');

            a.send();

         }

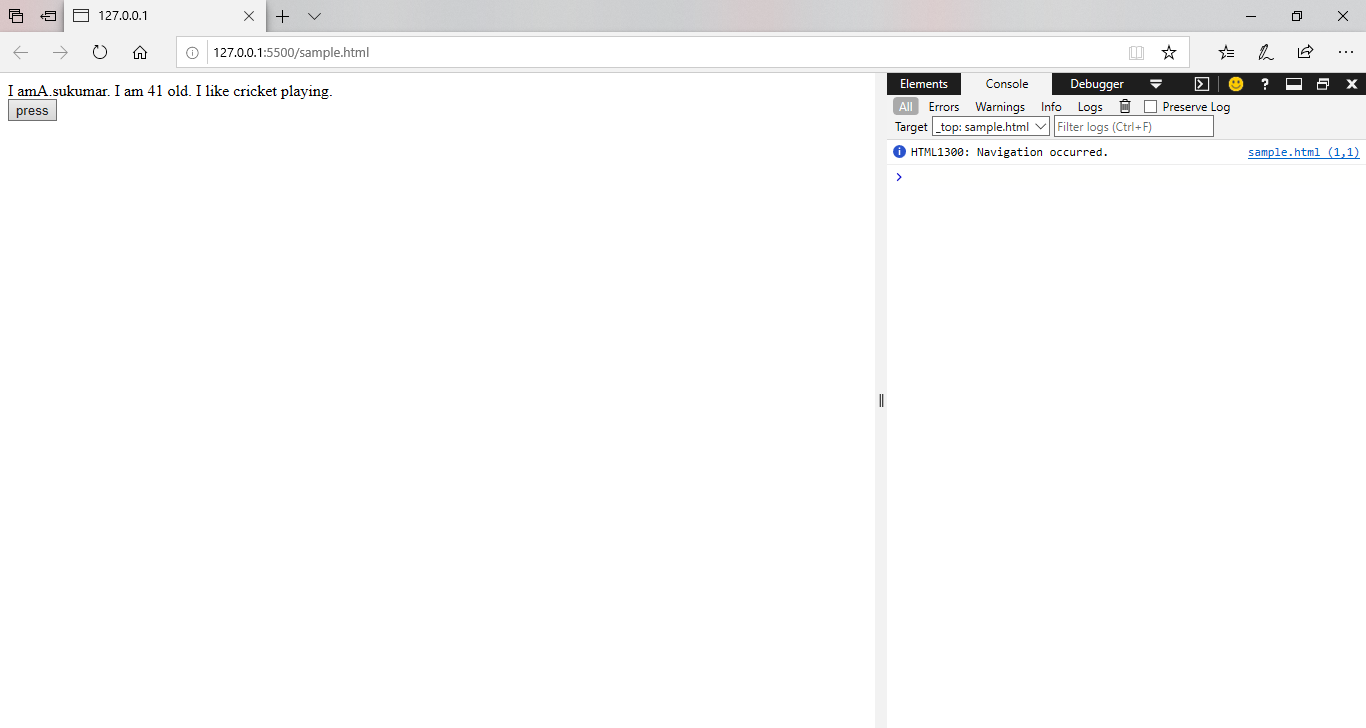
      </script>

    </body>

 </html>

Filename:data1.json

{"Name":"A.sukumar","age":"41","sports":"cricket"}



Example:2

<!DOCTYPEHTML>

<html>

   <head>

      <scriptsrc="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/4.7.7/handlebars.min.js"></script>

      <style>

      </style>

   </head>

   <body  >

      <scriptid='one'type="text/x-handlebars-template">

         <div>

            {{#each ab}}

             {{../front}}<p>I am{{Name}}. I am {{age}} old. I like {{sports}} playing.</p>

            {{/each}}

         </div>

     </script>

      <div></div>

      <buttononclick="drawGraphic()"> press</button>

      <script>

         vardata;

         functiondrawGraphic(){

            vara=newXMLHttpRequest();

            a.onload=function(){

              data=a.responseText;

              data=JSON.parse(data);

              vartemp1=Handlebars.compile(document.querySelector('#one').innerHTML);

              document.querySelector('div').innerHTML=temp1(data);

            }

            a.open('get','data1.json');

            a.send();

         }

      </script>

    </body>

 </html>

Data1.json

{"ab":[{"Name":"A.sukumar","age":"42","sports":"cricket"}

,{"Name":"A.veena","age":"38","sports":"cricket"}],

  "front":"hai"}

