

Homework: JSON Exercise

1. Objectives

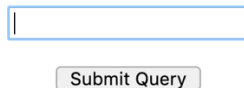
- Become familiar with the navigating JavaScript JSON objects;
- Use of JSON.parse parser and synchronous XMLHttpRequest;
- Transform the content of a JSON document into an HTML page.

2. Description

You are required to write an HTML/JavaScript program, which takes the URL of a JSON document containing top car manufacturers information, parses the JSON file, and extracts the list of top car manufacturers, displaying them in a table. The JavaScript program will be embedded in an HTML file so that it can be executed within a browser.

- Your program should display a text box to enter the JSON file name as shown below in Figure 1. On clicking the “Submit Query” button, your program should display the table as shown below, Figure 2. If the text box is left empty and Submit Query is clicked, an appropriate error message must be shown.

Enter URL for largest manufacturers by production (2017) List JSON File



The image shows a simple web form. At the top, there is a text label: "Enter URL for largest manufacturers by production (2017) List JSON File". Below this label is a rectangular text input field with a blue border. Underneath the input field is a button with the text "Submit Query".

Figure 1: Initial Page

- Once the JSON file is downloaded, a JavaScript function within the HTML file parses the JSON document that was passed as an input in the edit box.
- After parsing the JSON document, a popup window or tab should display a table consisting of the data for largest manufacturers of cars on 2017 that are contained in the JSON file. For example, given the following JSON document:

<http://csci571.com/hw/hw4/carlist.json>

the table below should be displayed:





Rank	Group	Country	Vehicles	Wiki Page	Logo
1	Toyota	<ul style="list-style-type: none"> Japan Toyota Motor Corporation is a Japanese multinational automotive manufacturer headquartered in Toyota, Aichi, Japan. In 2017, Toyota's corporate structure consisted of 364,445 employees worldwide and, as of September 2018, was the sixth-largest company in the world by revenue. 	10,466,051	https://en.wikipedia.org/wiki/Toyota	
2	Volkswagen Group	<ul style="list-style-type: none"> Germany Volkswagen AG, known internationally as the Volkswagen Group, is a German multinational automotive manufacturing company headquartered in Wolfsburg, Lower Saxony, Germany and indirectly majority owned by Porsche SE. It designs, manufactures and distributes passenger and commercial vehicles, motorcycles, engines, and turbomachinery and offers related services including financing, leasing and fleet management. 	10,382,334	https://en.wikipedia.org/wiki/Volkswagen_Group	
3	Hyundai / Kia	<ul style="list-style-type: none"> South Korea The Hyundai Motor Group is a South Korean multinational conglomerate headquartered in Seoul, South Korea, and it is the largest car manufacturer in the country. 	7,218,391	https://en.wikipedia.org/wiki/Hyundai_Motor_Group	
4	General Motors	<ul style="list-style-type: none"> United States General Motors Company, commonly referred to as General Motors (GM), is an American multinational corporation headquartered in Detroit that designs, manufactures, markets, and distributes vehicles and vehicle parts, and sells financial services, with global headquarters in Detroit's Renaissance Center. It was originally founded by William C. Durant on September 16, 1908 as a holding company. 	6,856,880	https://en.wikipedia.org/wiki/General_Motors	
		<ul style="list-style-type: none"> United States 			

Figure 2: Table containing car manufacturers from carlist.json

Here is a version of the *carlist.json* file containing the data that is displayed above:

```
{
  "Mainline": {
    "Table": {
      "Header": {
        "Data": [
          "Rank",
          "Group",
          "Country",
          "Vehicles",
          "Wiki Page",
          "Logo"
        ]
      },
      "Row": [
        {
          "Rank": "1",
          "Group": "Toyota",
          "Hubs": {
            "Hub": [
              "Japan",
              "Toyota Motor Corporation is a Japanese multinational automotive manufacturer headquartered in Toyota, Aichi, Japan. In 2017, Toyota's corporate structure consisted of 364,445 employees worldwide and, as of September 2018, was the sixth-largest company in the world by revenue. "
            ]
          }
        }
      ]
    }
  }
}
```

```

    ],
    },
    "Vehicles": "10,466,051",
    "HomePage": "https://en.wikipedia.org/wiki/Toyota",
    "Logo": "http://csci571.com/hw/hw4/toyota.jpg"
  },

  {
    "Rank": "2",
    "Group": "Volkswagen Group",
    "Hubs": {
      "Hub": [
        "Germany",
        "Volkswagen AG, known internationally as the Volkswagen Group, is a
        German multinational automotive manufacturing company headquartered in
        Wolfsburg, Lower Saxony, Germany and indirectly majority owned by Porsche SE. It
        designs, manufactures and distributes passenger and commercial vehicles,
        motorcycles, engines, and turbomachinery and offers related services including
        financing, leasing and fleet management."
      ]
    },
    "Vehicles": "10,382,334",
    "HomePage": "https://en.wikipedia.org/wiki/Volkswagen_Group",
    "Logo": "http://csci571.com/hw/hw4/volkswagen.jpg"
  },
  [.....]
]
}
}
}

```

3. Error Handling

An error condition that should be checked for a JSON file containing NO car manufacturers. An example of a JSON files which does not contain manufacturers entries:

```

{
  "Mainline": {
    "Table": {
      "Header": {
        "Data": [
          "Rank",
          "Group",
          "Country",
          "Vehicles",
          "Wiki Page",
          "Logo"
        ]
      }
    }
  }
}

```

```
}  
}  
}  
}
```

In addition, your program should handle the case when the JSON file does not exist. A proper message should be displayed.

The “structure” of the input JSON files will not change. We will not test the case where one of the keys is missed. In other words, every *Row* always contains the keys: *Rank*, *Group*, *Hubs*, *Vehicles*, *HomePage*, and *Logo*. The *Hubs* tag contains an array with key *Hub*. Note that The *Hub* array may contain ZERO or more values.

If the value of a tag is empty, you should display a blank cell.

You are required to handle the case where the Header data values are different. Please note that the Data tag values might differ but the JSON structure remains the same. For example, the Header can be like this:

```
"Header": {  
  "Data": [  
    "Rank of Manufacturer",  
    "Car Group",  
    "Company Info",  
    "No. of Vehicles",  
    "Wikipedia Page",  
    "Logo"  
  ]  
},
```

No other error conditions need be checked. In all cases if an error is found your program should show an alert box indicating the error was detected.

4. Hints

- *Step 1: Writing Your HTML/JavaScript program - Using the DOM Parser*

Here's how you could use the Microsoft DOM API and the Mozilla DOM API (used in Firefox) to load and parse a JSON document into a DOM tree, and then use the DOM API to extract information from that document.

```
<script type="text/javascript">
```

```

var jsonDoc;
function loadJSON (url) {
    var xmlhttp=new XMLHttpRequest();
    xmlhttp.open("GET",url,false); //open, send, responseText are
    xmlhttp.send();                //properties of XMLHttpRequest

    jsonDoc=xmlhttp.responseText;
    return jsonDoc;
}
// ..... processing the document goes here
</script>

```

Now you can parse the JSON file with `JSON.parse` and generate the HTML table on the fly by navigating through the JSON object. You can assume that every JSON file will have identical Object, Array and key names.

Your task is to write a program that transforms this type of JSON file into the table as shown above.

- *Step 2: Display the Resulting HTML Document*

You should use the DOM `document.write` method to output the required HTML.

- *Step 3: Use JavaScript control syntax*

The only program control statements that you will need to use for this exercise are the “if” and the “for” statements. The syntax of both statements is practically identical to the syntax of the corresponding statement in the C, C++ and Java languages, as in:

```

if (container_keys[j]=="Image") {
    // do stuff
}
for (j=0; j<container_keys.length; j++) {
    // do more stuff
}

```

Please make a note of the following issue:

Cross-Site Scripting (XSS):

JavaScript cannot call the resources from another domain. This is called cross side scripting which is not allowed in browsers. Therefore, you must put your JSON files and your script in the same domain. Please make sure, when you are testing your implementation, to place both the HTML file and the JSON file on your local machine IN THE SAME FOLDER. A sample file can be copied from here:

<http://csci571.com/hw/hw4/carlist.json>

Window.open() method must be used to pop up a new window which would display the final widget.

Image files can be either local or remote, as these files do not exhibit the cross-site scripting issue.

Scrollable Window:

The popup window should be scrollable so the user can read all records listed in the window.

6. Image Test Files

These image files are provided for testing purpose.

<http://csci571.com/hw/hw4/toyota.jpg>
<http://csci571.com/hw/hw4/volkswagen.jpg>
<http://csci571.com/hw/hw4/hyundai.jpg>
<http://csci571.com/hw/hw4/GM.jpg>
<http://csci571.com/hw/hw4/ford.jpg>
<http://csci571.com/hw/hw4/nissan.jpg>
<http://csci571.com/hw/hw4/honda.jpg>

7. Material You Need to Submit

On your course homework page, your link for this homework should go to a page that looks like the one displayed in the Figure on page 1. **This page should include your entire JavaScript/HTML/CSS program in a single file.** Also, you should upload your source code electronically to the csci571 GitHub Classroom account so that it can be compared to all other students'

code via the MOSS code comparison tool. If your submission is composed of multiple files, 3 points will be deducted.