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 Container Shipping companies information, parses the JSON file, and extracts the list of container shipping companies, 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.

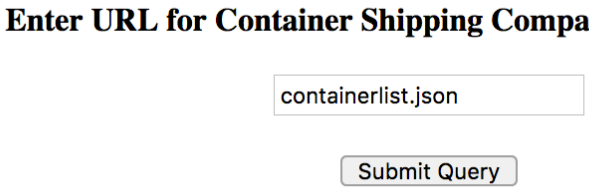
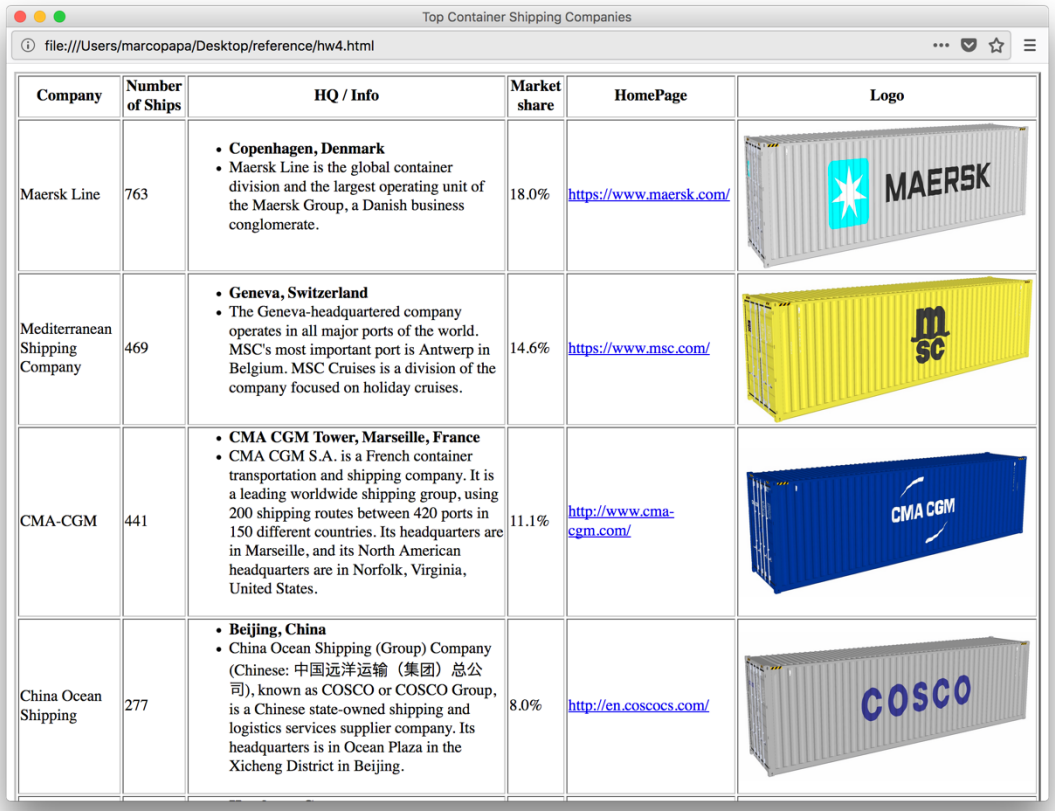


Figure 1: Initial Page

- 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 all container shipping companies that are contained in the JSON file. For example, given the following JSON document:

http://csci571.com/hw/hw4/containerlist.json

the table below should be displayed:



Company	Number of Ships	HQ / Info	Market share	HomePage	Logo
Maersk Line	763	• **Copenhagen, Denmark** • Maersk Line is the global container division and the largest operating unit of the Maersk Group, a Danish business conglomerate.	18.0%	https://www.maersk.com/	MAERSK
Mediterranean Shipping Company	469	• **Geneva, Switzerland** • The Geneva-headquartered company operates in all major ports of the world. MSC's most important port is Antwerp in Belgium. MSC Cruises is a division of the company focused on holiday cruises.	14.6%	https://www.msc.com/	msc
CMA-CGM	441	• **CMA CGM Tower, Marseille, France** • CMA CGM S.A. is a French container transportation and shipping company. It is a leading worldwide shipping group, using 200 shipping routes between 420 ports in 150 different countries. Its headquarters are in Marseille, and its North American headquarters are in Norfolk, Virginia, United States.	11.1%	http://www.cma-cgm.com/	CMA CGM
China Ocean Shipping	277	• **Beijing, China** • China Ocean Shipping (Group) Company (Chinese: 中国远洋运输（集团）总公司), known as COSCO or COSCO Group, is a Chinese state-owned shipping and logistics services supplier company. Its headquarters is in Ocean Plaza in the Xicheng District in Beijing.	8.0%	http://en.coscocs.com/	COSCO

Figure 2: Table containing airlines from airlinelist.json

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


```
{
 "Mainline": {
  "Table": {
   "Header": {
    "Data": [
     "Company",
     "Number of Ships",
     "HQ / Info",
     "Market share",
     "HomePage",
     "Logo"
    ]
   },
   "Row": [
```

```
      {
       "Company": "Maersk Line",
       "Ships": "763",
       "Hubs": {
        "Hub": [
         "Copenhagen, Denmark",
         "Maersk Line is the global container division and the largest operating unit of
    the Maersk Group, a Danish business conglomerate."
        ]
       },
       "Market": "18.0%",
       "HomePage": "https://www.maersk.com/",
       "Logo": "http://csci571.com/hw/hw4/Maersk_shipping_container.jpeg"
      },

      {
       "Company": "Mediterranean Shipping Company",
       "Ships": "469",
       "Hubs": {
        "Hub": [
         "Geneva, Switzerland",
         "The Geneva-headquartered company operates in all major ports of the world.
    MSC's most important port is Antwerp in Belgium. MSC Cruises is a division of the
    company focused on holiday cruises."
        ]
       },
       "Market": "14.6%",
       "HomePage": "https://www.msc.com/",
       "Logo": "http://csci571.com/hw/hw4/MSC_container.jpeg"
      },


[……]

     ]
    }
   }
}
```

3. Error Handling

An error condition that should be checked for a JSON file containing NO container shipping companies. An example of a JSON files which does not contain container company entries:

```
{
 "Mainline": {
  "Table": {
   "Header": {
      "Data": [
```

```
        "Company",
        "Number of Ships",
        "HQ / Info",
        "Market share”,
        "HomePage",
        "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: *Company*, *Ships*, *Hubs*, *Market*, *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>
    "Header": {
     "Data": [
       "Company",
       "Ships #",
       "Location",
       "Market %",
       "Home Page",
       "Container with 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/containerlist.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/COSCO_container.jpeg
http://csci571.com/hw/hw4/CMA_CGM_shipping_container.jpeg
http://csci571.com/hw/hw4/EVERGREEN_container.jpeg
http://csci571.com/hw/hw4/Hapag-Lloyd_shipping_container.jpeg
http://csci571.com/hw/hw4/Maersk_shipping_container.jpeg
http://csci571.com/hw/hw4/MSC_container.jpeg

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.