Software engineering

json Tutorial 2021.

reference: w3shools & tutorialspoint

1. What is json?

JSON stands for **JavaScript Object Notation**. When exchanging data between a browser and a server, the data can only be text. JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server. We can also convert any JSON received from the server into JavaScript objects. This way we can work with the data as JavaScript objects, with no complicated parsing and translations.

JSON is written with attributes and their values pair. The JSON syntax is a subset of the JavaScript syntax. JSON syntax is derived from JavaScript object notation syntax: Data is in name/value pairs, Data is separated by commas, Curly braces hold objects, Square brackets hold arrays. The JSON format is almost identical to JavaScript objects.

Example:

```
{       firstName:"John",
        lastName:"Doe",
        fullName: function(){return this.firstName + this.lastName}
};
```

Why use JSON? Since the JSON format is text only, it can easily be sent to and from a server, and used as a data format by any programming language.

JavaScript has a built in function to convert a string, written in JSON format, into native JavaScript objects: **JSON.parse()** So, if you receive data from a server, in JSON format, you can use it like any

other JavaScript object. If you have data stored in a JavaScript object, you can convert the object into JSON, and send it to a server:

```
<!DOCTYPE html>
<html>
<body>
<h2>Convert a string written in JSON format, into a JavaScript object.</h2>
<p id="demo"></p>
<script>
var myJSON = '{ "name":"John", "age":31, "city":"New York" }';
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML ="name: " + myObj.name + "
age: " + myObj.age + "   city: " + myObj.city;
</script>
</body>
</html>
```

Practice #1) Try the above example and capture the result. Change John Doe with your first and last name.

2. json and Javascript

```
<html>
<head>
<title>JSON example</title>

 <script language="javascript" >

        var object1 = { "language" : "JavaScript", "class" : "software engineering" };
        document.write("<h1>JSON with JavaScript example</h1>");
        document.write("<br>");
        document.write("<h3>Language = " + object1.language+"</h3>");
        document.write("<h3>Class = " + object1.class+"</h3>");

        var object2 = { "language" : "Ruby", "professor" : "woongsup Kim" };
        document.write("<br>");
        document.write("<h3>Language = " + object2.language+"</h3>");
        document.write("<h3>Professor = " + object2.professor+"</h3>");
        document.write("<hr />");
        document.write(object2.language +  " will be taught by professor " +
object2.author);
        document.write("<hr />");

        </script>

         </head>
         <body>

         </body>
</html>
```

3. Data Types

JSON format supports the following data types –

S. No	Type & Description
1. Number	double- precision floating-point format in JavaScript ex) { "age" : 30 }

2. String	double-quoted Unicode with backslash escaping ex) { "name" : "Andy" }
3. Boolean	true or false ex) { "sale" : true }
4. Array	an ordered sequence of values ex) { "student" : ["Tom", "Anna", "Peter"] }
5. Value	it can be a string, a number, true or false, null etc ex) var k = "tim"
6. Object	an unordered collection of key:value pairs ex) { "name" : "John", "age" : 30 , "city" : "Seoul" }
7. null	Empty ex) { "middlename" : null }

Example showing array containing multiple objects

```
{
    "books": [
        { "language":"Java" , "edition":"second" },
        { "language":"C++" , "lastName":"fifth" },
        { "language":"C" , "lastName":"third" }
    ]
}
```

4. Array Object

JSON objects are surrounded by curly braces {}. JSON objects is written in key/value pairs. Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null). Keys and values are separated by a colon. Each key/value pair is separated by a comma.JSON objects can be created with JavaScript. The following example shows creation of an array object in javascript using JSON, save the below code as **array_object.htm**

Practice #2) Try the above example and capture the result. Put two more books with your favorite choice.

5. JSON Parse

A common use of JSON is to exchange data to/from a web server. When receiving data from a web server, the data is always a string. Parse the data with **JSON.parse()**, and the data becomes a JavaScript object.

Example - Parsing JSON Imagine we received this text from a web server:

```
<!DOCTYPE html>
<html>
<body>

<h2>Create Object from JSON String</h2>

<p id="demo"></p>

<script>

var obj = JSON.parse('{ "name":"John", "age":30, "city":"Seoul"}');
document.getElementById("demo").innerHTML = obj.name + ", " + obj.age + ", " +
obj.city;

</script>

</body>
</html>
```

6. AJAX

You can request JSON from the server by using an AJAX(Asynchronous Javascript And XML) request As long as the response from the server is written in JSON format, you can parse the string into a JavaScript object. Don't forget to make test.txt on your desktop

```
<!DOCTYPE html>
<html>
<body>

<h2>Create Object from JSON String</h2>

<p id="demo"></p>
<!DOCTYPE html>
<html>
<body>

<h2>Use the XMLHttpRequest to get the content of a file.</h2>
<p id="demo"></p>

<script>

var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        myObj = JSON.parse(this.responseText);
        document.getElementById("demo").innerHTML = myObj.name;
    }
};
xmlhttp.open("GET", "test.txt", true);
xmlhttp.send();
</script>
<p>Take a look at <a href="test.txt" target="_blank">test.txt</a></p>
</body>
</html>
<script>

var obj = JSON.parse('{ "name":"John", "age":30, "city":"Seoul"}');
document.getElementById("demo").innerHTML = obj.name + ", " + obj.age +
", " + obj.city;

</script>

</body>
</html>
```

Practice #3) Try the above example and capture the result.

7. JSON with Python

This chapter covers how to encode and decode JSON objects using Python programming language. Let's start with preparing the environment to start our programming with Python for JSON.

Environment

Before you start with encoding and decoding JSON using Python, you need to install any of the JSON modules available for Python. But you don't need to install something other. Python has a built in package called json to work with JSON data.

To parse JSON data, first import json library.

```
import json
```

Parsing JSON using Python

If you have json string then you can parse it by using json.load() method.

```
import json

# some JSON string
x =  '{ "name":"John", "age":30, "city":"New York"}'

# parse x:
y = json.loads(x)

# the result is a Python dictionary:
print(y["age"])
```

Practice #4) Try the above example and capture the result.

If you have a python object, you can convert it into JSON string by using json.dump() method.

```
import json

# a Python object: dictionary
x = {
  "name": "John",
  "age": 30,
  "city": "New York"
}

# convert into JSON object:
y = json.dumps(x)

# the result is a JSON string:
print(y)
```

You can convert a Python object containing all data types.

```
import json

x = {
  "name": "John",
  "age": 30,
  "married": True,
  "divorced": False,
  "children": ("Ann","Billy"),
  "pets": None,
  "cars": [
    {"model": "BMW 230", "mpg": 27.5},
    {"model": "Ford Edge", "mpg": 24.1}
  ]
}

print(json.dumps(x))
```

You can read a JSON file and parse it through with statement

Example.json

```
{
  "name": "John",
  "age": 30,
  "married": True,
  "divorced": False,
  "children": ("Ann","Billy"),
  "pets": None,
  "cars": [
    {"model": "BMW 230", "mpg": 27.5},
    {"model": "Ford Edge", "mpg": 24.1}
  ]
}
```

example.py

```
import json

with open('example.json') as json_file:
      json_data = json_load(json_file)

json_string = json_data["cars"]
print(json_string)
```

Practice #5) Try the above example and capture the result.