**Julian’s Personal Notes**: JSON

**JSON** = **J**ava**S**cript **O**bject **N**otation

You should know HTML, CSS, and basic JavaScript before heading into JSONs.

You can convert a JSON file into native JavaScript objects with JSON.parse()

**Syntax**: JSON is written as name/value pairs. Think of maps in C++.

**Example** 🡪 “name”:”Julian”

**Unlike JS**, the name (key) section requires double quotes as well.

The value section also requires double quotes, regardless of its type.

**Types**: Values can be any of the following: string, number, JSON object, array, boolean, and null. That’s a total of six different types of values!

You **cannot** use dates and functions as values for JSON files.

**Features**: JSON is like XML, but has a number of advantages.

+ JSON can be parsed by a standard JavaScript function.

+ There are no end tags. XML would use <name>Julian</name>

+ Arrays can be used, whereas XML does not allow for arrays.

+ Honestly, JSON is easier and quicker to read.

* JSON text files can be fetched with an **XMLHttpRequest**. Details later.
* JSON objects are surrounded by curly braces { }.
* JSON object values are accessible with dot notation.

**JSON Example 1**

{ “students”: [

{ “firstName”:”Julian”, “id”:”111” },

{ “firstName”:”Goomba”, “id”:”999” },

{ “firstName”:”Kuribo”, “id”: 727 }

]}

**Stringify a JavaScript Object**: This means you are going to take a JavaScript object (like a person, student, or class) and turn it into JSON text. You can do this with JSON.stringify(object)

**Parsing JSON Text**: This is the opposite, which can be done with the JSON.parse(object) function. Example code follows below:

<script>

var text = ‘{“name”:”Julian”, “id”:”111”}’

var object = JSON.parse(txt);

document.getElementByID(“myID”).innerHTML = object.name + “ “ + object.id;

</script>

Both stringifying a JavaScript object and parsing JSON text applies to arrays.

**Looping**: You can loop through object keys (or values) with a for loop in JavaScript

//Looping and printing object keys

for (x in MyJSON)

document.getElementByID(“myID”).innerHTML += x + “<br>”;

//Looping and printing object values

for (x in MyJSON)

document.getElementByID(“myID”).innerHTML += myJSON[x];

**JSON Example 2**: Dynamic HTML Table (Used from w3schools)

<select id="myselect" onchange="change\_myselect(this.value)">  
    <option value="">Choose an option:</option>  
    <option value="customers">Customers</option>  
    <option value="products">Products</option>  
    <option value="suppliers">Suppliers</option>  
</select>  
  
<script>  
function change\_myselect(sel) {  
    var obj, dbParam, xmlhttp, myObj, x, txt = "";  
    obj = { table: sel, limit: 20 };  
    dbParam = JSON.stringify(obj);  
    xmlhttp = new XMLHttpRequest();  
    xmlhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            myObj = JSON.parse(this.responseText);  
            txt += "<table border='1'>"  
            for (x in myObj) {  
                txt += "<tr><td>"+ myObj[x].name + "</td></tr>";  
            }  
            txt += "</table>"   
            document.getElementById("demo").innerHTML = txt;  
        }  
    };  
    xmlhttp.open("POST", "json\_demo\_db\_post.php", true);  
    xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
    xmlhttp.send("x=" + dbParam);  
}  
</script>

**More information about JSON and PHP will be added at a later date**. 9/21/2018