Connection to JSON file

- Introduction to JSON
- JSON vs XML
- Display JSON data in HTML table using JavaScript

JSON

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a **text format** for storing and transporting data
- JSON is "self-describing" and easy to understand

JSON vs XML

JSON Example

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML Example

```
<employees>
 <employee>
 <firstName>John</firstName> <lastName>Doe</lastName>
 </employee>
 <employee>
  <firstName>Anna</firstName> <lastName>Smith</lastName>
 </employee>
 <employee>
  <firstName>Peter</firstName> <lastName>Jones</lastName>
 </employee>
</employees>
```

JSON is Like XML Because

- Both JSON and XML are "self describing"
- Both are hierarchical (values within values)
- Both can be parsed and used by lots of programming languages
- Both can be fetched with an XMLHttpRequest

JSON is Unlike XML Because

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

XML is much more difficult to parse than JSON.
JSON is parsed into a ready-to-use JavaScript object.

Parse : Analiser

JSON – Exemple

N	LastName	FirstName	Email	Photo
1	Ajawin	Sunday Otwel Akat	otwelsunday94@gmail.com	photos/Ajawin-Sunday-Otwel-Akat.jpg
2	Allal El Bakhti	Mouad	allalelbakhti.mouad@gmail.com	photos/Allal-El-Bakhti-Mouad.jpg
3	Azemat	Zainab	zainab.azemat@etu.uae.ac.ma	photos/Azemat-Zainab.jpg
4	Barhoun	Mohamed	mohamedbarhun2020@gmail.com	photos/Barhoun-Mohamed.jpg
5	Benzeyan	Safouan	safouanbenziyane@gmail.com	photos/Benzeyan-Safouan.jpg
6	Bouzarhoun	Ismail	Ismailbouzarhoune09@gmail.com	photos/Bouzarhoun-Ismail.jpg
7	El Gazi	Youssef	ghyyoussef@gmail.com	photos/El-Gazi-Youssef.jpg
8	El yamlahi	Oussama	oussamaelyamlah10@gmail.com	photos/El-yamlahi-Oussama.jpg
9	Ghbalou	Mohammed diyae	meddiyae.ghbalou@gmail.com	photos/Ghbalou-Mohammed-Diyae.jpg
10	Haiddouch	Yassir	haiddouchy5@gmail.com	photos/Haiddouch-Yassir.jpg
11	Jmili	Mouad	mouad.jmili@etu.uae.ac.ma	photos/Jmili-Mouad.jpg
12	Lemnaouar	Doha	duha.lmnr@hotmail.com	photos/Lemnaouar-Doha.jpg
13	Mejdoubi	Mehdi	mehdimejdoubi134@gmail.com	photos/Mejdoubi-Mehdi.jpg
14	Zian	Al Amine	alaminezian@gmail.com	photos/Zian-Al-Amine.jpg
15	Zian	Hicham	zhicham63@gmail.com	photos/Zian-Hicham.jpg
16	Zoumhan	hajar	hajar.zoumhan20@gmail.com	photos/Zoumhan-Hajar.jpg

N	Last Name	First Name	Email	Photo
1	Ajawin	Sunday Otwel Akat	otwelsunday94@gmail.com	
2	Allal El Bakhti	Mouad	allalelbakhti.mouad@gmail.com	
3	Azemat	Zainab	zainab.azemat@etu.uae.ac.ma	
4	Barhoun	Mohamed	mohamedbarhun2020@gmail.com	

JSON file

students.json


```
[
    {
        "N": "1",
        "LastName": "Ajawin",
        "FirstName": "Sunday Otwel Akat",
        "Email": "otwelsunday94@gmail.com",
        "Photo": "photos/Ajawin-Sunday-Otwel-Akat.jpg"
    },
    {
        "N": "2",
        "LastName": "Allal El Bakhti",
        "FirstName": "Mouad",
        "Email": "allalelbakhti.mouad@gmail.com",
        "Photo": "photos/Allal-El-Bakhti-Mouad.jpg"
    },
    {
        "N": "3",
        "LastName": "Azemat",
        "FirstName": "Zainab",
        "Email": "zainab.azemat@etu.uae.ac.ma",
        "Photo": "photos/Azemat-Zainab.jpg"
    },
```


Convert table to JSON file
https://tableconvert.com/excel-to-json

JSON – HTML File – Table with rows from json file

```
<body>
<table>
        <thead>
            <tr>
                <th>N</th>
                <th>Last Name</th>
                <th>First Name</th>
                <th>Email</th>
                <th>Photo</th>
            </tr>
        </thead>
        <tbody id="data-output">
            <!-- Students from javascript file in here. -->
        </tbody>
</table>
    <script src="script.js"></script>
</body>
```

index.hml

JSON –Javascript file

In line 1 we use the fetch() method to get the data from the students.json file.

```
1  fetch("stutents.json")
```

The fetch() method returns a Promise object. So in line 2 we use the .then() method to catch the Response object and to resolve it to a javascript object with the .json() method, in line 3.

```
2  .then(function(response){
3    return response.json();
4  })
```

The .json() method returns also a promise, so we have to use another .then() method to catch our data (in our case the products). That is what we do in line 5.
The products argument inside the function is holding a javascript array of products.

```
5  .then(function(students){
```

In line 6 we are targeting the table-body in the html file and we storing it in the placeholder variable.

```
6  let placeholder = document.querySelector("#data-output");
```

JSON –Javascript file

- In line 7 we initialize a variable named out and we are setting its value to an empty string, so we can use the out variable later in the script.

```
7   let out = "";
```

- Next we are going to loop through the students array in line 8 so we can access every student.

```
8   for(let student of students){
```

Inside the for loop, we use the out variable to append a table-row template which holds the student values. We are using a template literal (the back ticks ``) to write the html code.

```
9   out += `
10   <tr>
11          <td>${student.LastName}</td>
12          <td>${student.FirstName}</td>
13          <td>${student.Email}</td>
14          <td>${student.Email}</td>
15          <td> <img src='${student.photo}'> </td>
16  </tr>
17     `;
18  }
```

JSON –Javascript file

We can insert in the html code, javascript variables using this structure ${product.image}.

And the last thing we have to do is to target the **tbody** element and add the data that the out variable holds.

```
19  placeholder.innerHTML = out;
20  });
```

JSON –Javascript file

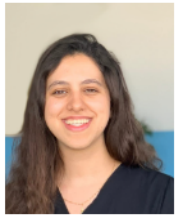
```
fetch("students.json")
.then(function(response){
    return response.json();
})
.then(function(students){
    let placeholder = document.querySelector("#data-output");
    let out = "";
    for(let student of students){
        out += `
            <tr>
                <td>${student.N}</td>
                <td>${student.LastName}</td>
                <td>${student.FirstName}</td>
                <td>${student.Email}</td>
                <td> <img src='${student.Photo}'> </td>
            </tr>
        `;
    }
    placeholder.innerHTML = out;
});
```

script.js

JSON – CSS File

```
/* I always reset all the html elements */
*{margin: 0;padding: 0; box-sizing: border-box;}
body{font-family: sans-serif; min-height: 100vh; color: #555;}
table{width:  1000px; margin: 30px auto;}
table th{padding: 10px 0; background-color: #f4f4f4;    border:  thin solid #d4d4d4;}
table td{padding: 10px; border:  thin solid #d4d4d4; width: 18%; text-align: center;
    background-color: #fff;}
table img{width:  70%;}
table td{font-weight: bold; font-size: 22px;}
table a{text-decoration: none; color: darkred;background-color: #ffecef; padding: 10px
15px;}
table a:active{background-color: #ccf4d6; color:  green;}
table select{padding: 10px; width:  200px; font-size: 16px;
    border: thin solid #d4d4d4; background-color: transparent;}
table select:focus{outline: none;}
table img{  width:  60%;}
```

JSON - Result

N	Last Name	First Name	Email	Photo
1	Ajawin	Sunday Otwel Akat	otwelsunday94@gmail.com	
2	Allal El Bakhti	Mouad	allalelbakhti.mouad@gmail.com	
3	Azemat	Zainab	zainab.azemat@etu.uae.ac.ma	
4	Barhoun	Mohamed	mohamedbarhun2020@gmail.com