

/* Objetos de JS y Objetos de JSON

Curso practico de JSON - JavaScript Object Notation.

---¿Porque aprender JSON?

- Formato popular para el intercambio de datos entre diferentes sistemas sin importar el lenguaje de programacion.
- Bastante utilizado en REST APIs.
- Es usado en Bases de datos.

---¿Que es JSON?

Abreviado de JavaScript Object Notation. Es un formato ligero de intercambio de datos. Formato basado en JavaScript pero puede ser usado por cualquier lenguaje de programacion. Facil de escribir. Usado en el mundo web como AJAX.

---Tipos de datos de JSON: Porque como un lenguaje de programacion este formato tambien los tiene:

- numbers: Integer y numeros flotantes
- string: Caracteres Unicode, que colocaremos con dobles comillas "".
- boolean: True o False, representa los dos estados.
- arrays: Listas ordenadas que comienzan con sus indeice en cero.
- objetos: Datos representados por pares de clave - valor.
{"nickname": "Alfred"}
- null: Para datos no disponibles o que no existe aun.

---Reglas para poder crear un archivo json:

- Pares clave:valor, JSON esta conformado por datos en clave/valor:
name: "eugenia",
password: "eugenia123",
country: "spain",
age: 27

Representaacion de datos facil de leer y entender.

- Las claves deben ir en doble comillas tambien: Si o si debe ser asi:
"name": "eugenia",
"password": "eugenia123",
"country": "spain",
"age": 27

- Cualquier clave que creemos siempre, siempre debe tener un valor.
Podemos crear una clave con un valor vacio:
"nickname": ""

- Colocar todos estos clave/valor entre llaves: asi agrupamos todos los datos en una sola entidad.

```
{  
  "name": "eugenia",  
  "password": "eugenia123",  
  "country": "spain",  
  "age": 27  
}
```

- Para separar los claves valor unos de otros debemos usar coma, ",", salvo el ultimo par clave valor que no llevara.

- Podemos anidar objetos, uno dentro de otros:

```
{
  "name": "eugenia",
  "password": "eugenia123",
  "address": {
    "city": "london",
    "street": "baker street 123"
  },
  "age": 27
}
```

- Las extensiones de los archivos deben ser .json:

datos.json

mydata.json

users.json

- Cuando enviamos datos de un sistema a otro por lo general le damos informacion al otro sistema que datos le estamos enviando, eso se conoce como mime type, o el tipo de datos de archivos que estoy enviando, en json es comun usar:

application/json

*/

// Tenemos nuestro primer objeto de JS:

```
const user1 = {
  "name": "Rayn",
  "lastname": "Ray",
  "age": 25,
  "nickname": "Rayn123",
  "hobbies": ["run", "code", "eat"],
  "address": {
    "street": "123 Main St",
    "city": "New York"
  },
  "married": true,
  greet() {
    return "Hola"
  }
}
```

// Tenemos un segundo objeto de JS vemos que no usa la clave doble comillas, eso es permitido en los objetos de JS:

```
const user2 = {
  name: "Jose",
  lastname: "Perez"
}
```

```

// console.log(user1);
// console.log(user2);

// Transformamos eso Objetos de JS a Objetos de JSON usando el metodo
JSON.stringify() recibe por parametro el objeto de JS:

const json1 = JSON.stringify(user1);
const json2 = JSON.stringify(user2);
// console.log(json1);
// console.log(json2);

/*Creamos un objeto de amigos, con son varios creamos un array usando los
corchete []:
const friends = [
    {},
    {},
    {},
    {}
]
*/

const friends = [
    {
        name: "Joe",
        nickname: "Joe123"
    },
    {
        name: "Martin",
        nickname: "Martin123"
    },
    {
        name: "Julliet",
        nickname: "Julliet123"
    },
    {
        name: "Dan",
        nickname: "Dan123"
    }
]

// Agregamos al objeto user1 la lista de amigos:
user1.friends = friends;

// Pasamos la lista de amigos de Objeto JS a Objeto JSON:
const friendsJSON = JSON.stringify(friends);
// console.log(friendsJSON);
// console.log(user1);

// Aqui vemos como mostrar la lista de amigos en HTML recorreremos el array
con un for para mostrarla en el dom del archivo index.html:
let output = '';
for (let i = 0; i < friends.length; i++) {
    output = output + `<li>${friends[i].name} -
    ${friends[i].nickname}</li>`;
    console.log(output);
}
document.getElementById('people').innerHTML = output;

```

```
// El metodo para convertir un JSON que nos viene del servidor es el
metodo JSON.parse() y recibe por parametro el objeto JSON.
const userObj = JSON.parse(json1);
console.log(userObj);
```

```
/*
Ahora vamos a meter dentro de un archivo .JSON nuestro user1. Vemos el
archivo que se llama user.JSON
*/
```

```
    console.log(JSON.stringify(user1));
```

```
/*
Tambien podemos ver un archivo JSON en el navegador basta con arrastrar
ese archivo al navegador y este tiene la capacidad de poder mostrarlo.
```

Ahora vamos a ver como consumir datos desde el servidor, esos datos nos vendran en JSON, no nos vamos a conectar haremos un servidor en local y este nos servira esos datos, lo haremos con NodeJS.

El primer programa que instalaremos sera livenesserver. Paa tener un servidor web: Lo haremos todo sobre la carpeta json_course:
npm install -g live-server

Lo instalamos de manera global. ahora vamos a correr la app usando ese servidor de livenesserver: Usaremos la terminal de VScode. Simplemente sobre la carpeta damos live-server y nos abria un servidor y vemos como corre nuestra app en el navegador.

Ahora este servidor sera el encargado de servirnos el archivo user.json

http://127.0.0.1:8080/user.json

```
// 20220511163050
// http://127.0.0.1:8080/user.json
```

```
{
  "name": "Rayn",
  "lastname": "Ray",
  "age": 25,
  "nickname": "Rayn123",
  "hobbies": [
    "run",
    "code",
    "eat"
  ],
  "address": {
    "street": "123 Main St",
    "city": "New York"
  },
  "married": true,
  "friends": [
    {
      "name": "Joe",
      "nickname": "Joe123"
    },
    {
      "name": "Martin",
```

```

        "nickname": "Martin123"
      },
      {
        "name": "Julliet",
        "nickname": "Julliet123"
      },
      {
        "name": "Dan",
        "nickname": "Dan123"
      }
    ]
  }
}

```

Esos son los datos que nos esta mandando el servidor en local a traves del protocolo http.

Tambien existen redes APIS donde podemos obtener datos:

Tenemos JSONPlaceholder:
<https://jsonplaceholder.typicode.com/>

Tenemos varios archivos a los cuales acceder por ejemplo post o user.

<https://jsonplaceholder.typicode.com/posts>

<https://jsonplaceholder.typicode.com/users>

Ahora vamos a usar nuestro archivo user.json como si viniera de un servidor externo.

Para traer esos datos el lenguaje JS nos prevee de un metodo llamado fetch() recibe por parametro la url del servidor y asi podemos traer datos desde otro servidor: Lo vemos a continuacion:

```

*/

fetch('https://jsonplaceholder.typicode.com/posts')
  .then((resp)=>{
    return resp.json();
  })
  .then(data =>{
    console.log(data);
  })

// Ahora ua vez que tenemos los datos los podemos recorrer como hicimos
arriba y mostrarlos en el dom:

let postsElements = '';
fetch('https://jsonplaceholder.typicode.com/posts')
  .then((resp) => {
    return resp.json();
  })
  .then(data => {
    for(let i = 0; i<data.length; i++){
      postsElements += `<li>${data[i].userId} - ${data[i].title}</li>`
    }
    document.getElementById('posts').innerHTML = postsElements;
  })

```

```
/*
Ahora vamos a ver como hacerlo con nuestro arreglo de amigos que es lo
mismo que hemos visto anterior pero eso viene de otro servidor
*/

let amigos = '';
fetch('/user.JSON')
  .then(respuesta =>{
    return respuesta.json();
  })
  .then(data=>{
    console.log(data.friends);
    for (let i = 0; i < data.friends.length; i++) {
      amigos += `<li>${data.friends[i].name} -
${data.friends[i].nickname
}</li>`
    }
    document.getElementById('friends').innerHTML = amigos;
  })
```