



JSON Y AJAX


FORMATO JSON



JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos.

Está constituido por dos estructuras:

- Una colección de pares de nombre/valor. (*Objeto*, registro, estructura, diccionario, etc..)
- Una lista ordenada de valores. (Arreglos, vectores, listas o secuencias).



```
{
  "squadName" : "Super Hero Squad",
  "homeTown" : "Metro City",
  "formed" : 2016,
  "secretBase" : "Super tower",
  "active" : true
}
```

JSON



USO.

Comúnmente es utilizado para transmitir datos en aplicaciones web.

Ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o vice versa.

Notas:

- JSON es sólo un formato de datos
- Una coma o dos puntos mal ubicados pueden producir que un archivo JSON no funcione.

AJAX



Es una técnica de JavaScript que nos permite consultar información o enviar información desde y hacia un servidor externo.

La información que descargamos puede ser recibida usualmente como texto o como objeto JSON. Lo más usual es JSON, para así poder manipularlo como un objeto de JavaScript.



AJAX Y JSON

Hay dos formas en la que nosotros podemos hacer las peticiones Ajax. Una es la forma antigua de hacerlo y la otra es la forma actual y más moderna. En esta clase aprenderemos ambas.

Empecemos por la antigua: `XMLHttpRequest`.

XMLHttpRequest- OBTENIENDO UN JSON

Para obtener un JSON utilizaremos la url a la que le pediremos la información, a través de una API(Application Programming Interface) llamado XMLHttpRequest.

<https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json>

¡EMPECEMOS!

PASO 1: Lo primero que vamos a hacer es almacenar la URL del JSON que se quiere recuperar en una variable.

```
var URL = 'https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json'; |
```

AJAX - XMLHttpRequest

PASO 2: Para crear una solicitud, se necesita crear una nueva instancia de objeto, desde el Constructor XMLHttpRequest, utilizando la palabra clave new.


```
var request = new XMLHttpRequest();
```

PASO 3: Ahora es necesario abrir una nueva solicitud utilizando el método [open\(\)](#).

```
request.open( 'GET', URL ); |
```

AJAX - XMLHttpRequest

PASO 4: Se envía la solicitud con el método send():



```
request.responseType = 'json';  
request.send();
```

PASO 5: Para poder esperar por la respuesta a retornar desde el servidor y luego, manejarla.



```
request.onload = function() {  
    var respuesta = request.response;  
    console.log(respuesta);  
}
```




¡Manos a la práctica!

Hagamos nuestra primera solicitud AJAX... Ve a la carpeta **01. Tareas > Ejercicios - Clase 9**

AJAX CON FETCH



El API fetch es un nuevo estándar, con un diseño moderno, **basado en promesas**, con mayor flexibilidad y capacidad de control a la hora de realizar llamadas al servidor. También está disponible en Node, por lo que **podemos utilizarlo de forma isomórfica**, es decir, tanto en cliente como en servidor.



```
fetch('https://httpbin.org/ip')
  .then(function(respuesta) {
    return respuesta.json(); // le decimos que queremos un json
  })
  .then(function(respuesta) {
    console.log(respuesta);
  })
  .catch(function(err) {
    console.error(err);
  });
```