Json y Js

JSON

- **El formato JSON es el formato de elección para intercambio de datos cuándo se trabaja con Ajax**.
- Aunque como hemos visto podemos usarlo con muchos tipos de archivos: xml, texto,html,css …

JSON

- JSON son las siglas de *JavaScript Object Notation* (Notación de Objeto JavaScript).

- Los objetos JavaScript tienen una estructura compuesta por pares :
- key : value (clave : valor)
- {key : value, key2 : value2, key3 : value3, ...}

JSON

- var misDatos = {"nombre" : “Maria", "apellidos" : “Valle Rubio", "edad" : 35};

Se accede asi:

- console.log(misDatos.nombre);
- console.log(misDatos.apellidos);
- console.log(misDatos.edad);

JSON

- var misDatos = {

"nombre" : “Maria",
"apellidos" : “Valle Rubio",
"edad" : 35,
"direccion" : { "calle" : "Gran Via, 2", "ciudad" : "Madrid" } };

- console.log(misDatos.nombre);
- console.log(misDatos.apellidos);
- console.log(misDatos.edad);
- console.log(misDatos.direccion.calle);
- console.log(misDatos.direccion.ciudad);

- Ejemplo con subkeys.

JSON

- {
  "name":"John",
  "age":30,
  "cars":["Ford", "BMW", "Fiat"]
  }
- x = myObj.cars[0];
- Console.log(x)

- Arrays en Json

JSON.parse

- Cuando enviamos una petición Ajax , la respuesta JSON es un string, **no es un objeto JavaScript sino una cadena de texto con notación de objeto JavaScript, esto es JSON**.
- Es necesario convertir esta cadena a un objeto antes de intentar trabajar con los datos.
- Esta conversión se realiza con el método **JSON.parse(datos aconvertir).**

Transformar un objeto de JavaScript en formato JSON.

- La función JSON.stringify
- JSON.stringify(value [, replacer] [, space])

- Se utiliza con array y con objetos.
- El segundo parámetro puede ser una función o un array .
- Si *replacer* es una función, **JSON.stringify** llama a la función y pasa la clave y el valor de cada miembro. El valor devuelto se utiliza en lugar del valor original. Si la función devuelve **undefined**, el miembro se excluye. La clave del objeto raíz es una cadena vacía: "
- Si *space* es un número, se aplica al texto del valor devuelto una sangría con el número especificado de espacios en blanco en cada nivel. Si *space* es mayor que 10, la sangría aplicada al texto es de 10 espacios.
- Si *space* es una cadena no vacía, como '\t', al texto del valor devuelto se le aplica una sangría con los caracteres de la cadena en cada nivel.

ejemplo1

Por ejemplo creamos un objeto genérico

- var contact = new Object();
- contact.nombre= “Luisa";
- contact.apellidos = “Garcia";
- contact.tlf = [“942-344567", “942-897898"];

Luego definimos un array , si el parámetro replace es un array actuará de filtro.

- Var filtrocontactos = new Array();
- filtrocontactos[0] = “apellidos";
- filtrocontactos[1] = “tlf";

Realizamos la conversión

- var jsonText = JSON.stringify(contact, filtrocontactos, "\t");
  document.write(jsonText);

Práctica 1

- Creamos un array con datos :
- Ej array razas– caniche,chiguagua,dalmata,pastor aleman,doberman…
- Crea un función que utilizaremos para el parámetro replace que nos ponga todos los nombres a mayúscula.(ya sabeis hay que crearla con dos parámetros key y value)
- Convierte el array original en json teniendo en cuenta que los datos saldrán en mayúscula.

Práctica

- {"Personas":[
  {"nombre":"Jorge","edad":23},
  {"nombre":"Carlos","edad":17}
  {"nombre":“luis","edad":20}
  {"nombre":“Jose","edad":15}
  ]};

Creamos este fichero json y hacemos una petición ajax para después visualizar los datos de las personas menores de 18 años. Para visualizar crear dinámicamente los contenedores que sean necesarios.