19 - Qué es JSON?

JSON es el acrónimo de JavaScript Object Notation.

JSON es un formato alternativo de envío y recepción de datos, es decir remplaza a XML o el envío de texto plano.

Este formato de datos es más liviano que XML.

Veremos que hace el código más sencillo ya que utiliza el código JavaScript como modelo de datos.

Tenemos que repasar un poco como se definen los array literales y objetos literales en JavaScript, ya que serán las estructuras para la transmisión de datos:

```
var usuario=["juan","26"];
```

Como vemos los elementos de un array literal se encierran entre corchetes y los valores contenidos van separados por coma.

Cuando definimos un array literal no le indicamos a cada elemento de que tipo se trata, podemos almacenar cadenas (entre comillas), números, valores lógicos (true,false) y el valor null.

Para acceder a los elementos de un array literal lo hacemos por su nombre y entre corchetes indicamos que elementos queremos acceder:

```
alert(usuario[0]); //Imprimimos el primer elemento del array
alert(usuario[1]); //Imprimimos el segundo elemento del array
```

Veamos como definimos los objetos literales en JavaScript:

```
var persona={
  "nombre":"juan",
  "clave":"xyz",
  "edad":"26"
};
```

Los objetos literales se definen por medio de pares "nombre":"valor"

Todo objeto literal tiene un nombre, en el ejemplo le llamé persona. Un objeto literal contiene una serie de propiedades con sus respectivos valores. Todas las propiedades del objetos se encuentran encerradas entre llaves. Luego cada propiedad y valor se las separa por dos puntos. Y por último cada propiedad se las separa de las otras propiedades por medio de la coma.

Para acceder a las propiedades del objeto literal lo podemos hacer de dos formas:

```
alert(persona.nombre);  //Imprime el valor de la propiedad nombre del objeto persona.
```

La segunda forma es indicando la propiedad entre corchetes:

```
alert(persona['nombre']);
```

Luego se pueden combinar objetos literales y array literales:

```
var persona={
          "nombre":"juan",
          "edad":"22",
          "estudios":["primario","secundario"]
         };
```

Como podemos ver podemos crear estructuras de datos complejas combinando objetos literales y array literales.

Luego para acceder a los distintos elementos tenemos:

```
alert(persona.nombre);
alert(persona.estudios[0]);
alert(persona.estudios[1]);
```

La sintaxis de JSON difiere levemente de lo visto anteriormente:

```
{
  "nombre":"juan",
  "edad":"22",
  "estudios":["primario","secundario"]
}
```

Como podemos ver en la sintaxis JSON no aparecen variables, sino directamente indicamos entre llaves las propiedades y sus valores.

También hemos eliminado el punto y como luego de la llave final. El resto es igual.

Ahora veamos la diferencia entre JSON y XML utilizando este ejemplo:

XML:

```
<persona>
<nombre>juan</nombre>
<edad>22</edad>
<estudios>
<estudio>primario</estudio>
<estudio>secundario</estudio>
</estudios>
</persona>
```

Y como vimos en JSON:

```
{
  "nombre":"juan",
  "edad":"22",
  "estudios":["primario","secundario"]
}
```

Podemos ver que es mucho más directa la definición de esta estructura.

Como podemos ver si tenemos que transmitir estas estructuras por internet la notación JSON es más liviana.

Otra ventaja es como recuperamos los datos en el navegador, como vimos si se trata de un archivo XML llamamos al método requestXML y luego accedemos por medio del DOM

En cambio con JSON al llegar el archivo procedemos a generar una variable en JavaScript que recree el

objeto literal, esto mediante la el método parse del objeto JSON:

```
var persona=JSON.parse(conexion1.responseText);
```

Ya veremos en el próximo concepto como recuperar los datos del servidor mediante el objeto XMLHttpRequest

Para probar y generar un objeto a partir de una notación JSON haremos el siguiente problema:

Confeccionar una página que contenga un botón. Al ser presionado evaluar un string que almacena un objeto literal con notación JSON. El objeto literal debe representar las características de una computadora (procesador, memoria ram, capacidad de cada disco duro)

Mostrar los datos mediante el método alert

Hay que tener bien en cuenta que en este problema no hay nada de AJAX ya que no nos comunicaremos con el servidor para el envío de datos.

pagina1.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Problema</title>
  <script src="funciones.js"></script>
</head>
<body>
  <h1>Evaluar una variable que contiene notación JSON.</h1>
  <input type="button" id="boton1" value="Ver">
</body>
</html>
```

funciones.js

```
addEventListener('load',inicializarEventos,false);

function inicializarEventos()
{
  var ob=document.getElementById('boton1');
  ob.addEventListener('click',presionBoton,false);
}

function presionBoton(e)
{
  var cadena='{"microprocesador":"pentium",' +
             '"memoria":1024,' +
             '"discos":[80,250]' +
             '}';
  var maquina=JSON.parse(cadena);
  alert('microprocesador:'+maquina.microprocesador);
  alert('Memoria ram:'+maquina.memoria);
  alert('Capacidad disco 1:'+maquina.discos[0]);
  alert('Capacidad disco 2:'+maquina.discos[1]);
}
```

Cuando se presiona el botón se ejecuta la función presionBoton. En esta lo primero que hacemos definimos un string que contiene un objeto con notación JSON:

```
var cadena='{"microprocesador":"pentium",' +
           '"memoria":1024,' +
           '"discos":[80,250]' +
           '}';
```

Seguidamente pasamos a evaluar este string:

```
var maquina=JSON.parse(cadena);
```

Ahora si tenemos un objeto JavaScript. Esto se logra utilizando la función parse del objeto JSON.

Una vez que tenemos el objeto en JavaScript procedemos a acceder a sus atributos:

```
alert('microprocesador:'+maquina.microprocesador);
alert('Memoria ram:'+maquina.memoria);
alert('Capacidad disco 1:'+maquina.discos[0]);
alert('Capacidad disco 2:'+maquina.discos[1]);
```

Problema resuelto.

Confeccionar una página que contenga un botón. Al ser presionado evaluar un string que almacena un objeto literal con notación JSON. El objeto literal debe representar las características de una computadora (procesador, memoria ram, capacidad de cada disco duro)

Mostrar los datos mediante el método alert

pagina1.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Problema</title>
  <script src="funciones.js"></script>
</head>
<body>
  <h1>Evaluar una variable que contiene notación JSON.</h1>
  <input type="button" id="boton1" value="Ver">
</body>
</html>
```

Ejecutar ejemplo

funciones.js

```
addEventListener('load',inicializarEventos,false);

function inicializarEventos()
{
  var ob=document.getElementById('boton1');
  ob.addEventListener('click',presionBoton,false);
}

function presionBoton(e)
{
  var cadena='{"microprocesador":"pentium",' +
             '"memoria":1024,' +
             '"discos":[80,250]' +
             '}';
  var maquina=JSON.parse(cadena);
```

Problema propuesto.

Definir con JSON una estructura que represente a un empleado, almacenar el nombre,fecha de ingreso, número de documento y los importes de sus últimos tres sueldos.

Imprimir los datos luego en pantalla.

Ver solución

Retornar