

Sintaxis JSON

- Los elementos en un objeto JSON son: claves (Keys: cadenas de caracteres (strings)) y valores (Values: tipos de datos JSON como la forma de un array, objeto, cadena (string), booleano, número o nulo.).
- Puede tener dos o más pares de claves/valor, con una coma para separarlos.

{“nombre1”:“valor”, “nombre2”:“valor”}

- los objetos están definidos por corchetes { },

“Programacion”: {“Gteorico”:“01T”, “Glaboratorio”:“01L”}

Programación es la clave y lo demás el objeto

- los arrays están definidos por corchetes [].

**“Programacion”: [{“Gteorico”:“01T”, “Glaboratorio”:“01L”}
, {“Gteorico”:“02T”, “Glaboratorio”:“02L”}]**

Los tipos de datos JSON son:

String	"cadena"	"Hola"
Number	Numero	10 3.5 -10
Object	{"clave":valor, Wclave":valor}	"Programacion": { "Gteorico": "01T", "Glaboratorio": "01L" }
Array	[valor]	[{ "Gteorico": "01T", "Glaboratorio": "01L" } , { "Gteorico": "02T", "Glaboratorio": "02L" }]
Null	Nulo	Null
Booleans	Lógicos	true false

Métodos que facilitan trabajar con objetos JSON y JavaScript

- Método que analiza una cadena escrita en formato JSON y devuelve un objeto JavaScript: `JSON.parse()`
- Un uso común de JSON es intercambiar datos hacia / desde un servidor web. Al enviar datos a un servidor web, los datos deben ser string. Para convertir un objeto JavaScript en una cadena se usa el método `JSON.stringify ()`.

EJEMPLO 1

JSON.parse()

```
<!DOCTYPE html>
<html>
<body>

<h2>Convierte una cadena JSON en un objeto de JavaScript</h2>

<p id="demo"></p>

<script>
var myJSON = '{"GTeorico":"01T", "GLaboratorio":5, "Universidad":"UDB"}';
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = "Grupo de teoria " + myObj.GTeorico +
" " + myObj.Universidad;
</script>

</body>
</html>
|
```

Convierte una cadena JSON en un objeto de JavaScript

Grupo de teoria 01T UDB

EJEMPLO 2

JSON.parse()

```
<!DOCTYPE html>
<html>
<body>

<h2>Creando un objeto a partir de una cadena JSON</h2>

<p id="demo"></p>

<script>
var text = '{"empleado":[" +
'{"firstName":"Javier","lastName":"Ramos" },' +
'{"firstName":"Rosa","lastName":"Rios" },' +
'{"firstName":"José","lastName":"López" }]}';

obj = JSON.parse(text);
document.getElementById("demo").innerHTML =
obj.empleado[1].firstName + " " + obj.empleado[1].lastName;
</script>

</body>
</html>
```

Creando un objeto a partir de una cadena JSON

Rosa Rios

EJEMPLO 3

JSON.stringify ().

```
<!DOCTYPE html>
<html>
<body>

<h2>Crear una cadena JSON a partir de una matriz de JavaScript.</h2>

<p id="demo"></p>

<script>

var arr = [ "David", "Fher", "Carlos", "Roberto" ];
var myJSON = JSON.stringify(arr);
document.getElementById("demo").innerHTML = myJSON;

</script>

</body>
</html>
```

Crear una cadena JSON a partir de una matriz de JavaScript.

["David","Fher","Carlos","Roberto"]

Objetos LocalStorage y SesiónStorage

Nueva característica que soportan los navegadores actuales para guardar información en el lado del cliente. Los objetos de almacenamiento web `localStorage` y `sessionStorage` permiten guardar pares clave / valor en el navegador. Lo interesante de ellos es que los datos sobreviven una actualización de la página (para `sessionStorage`) e incluso un reinicio completo del navegador (para `localStorage`).

DIFERENCIA

- `LocalStorage` no tiene fecha de expiración, los datos no se eliminan cuando se cierra el navegador, es de sólo lectura y `SesionStorage` es válida para la ventana en la que estamos navegando, almacena datos para una sesión, los datos se pierden cuando se cierra el navegador.
- Ambas pueden ser eliminadas si se limpia la información guardada en el navegador.

Sintaxys

window.localStorage

- **Para guardar datos:**
- `localStorage.setItem("key", "value");`
- **Para leer datos:**
- `var lastname = localStorage.getItem("key");`
- **Para eliminar datos:**
- `localStorage.removeItem("key");`

LocalStorage

```
<!DOCTYPE html>
<html>
<body>

<div id="resultado"></div>

<script>
//Verifique el soporte del navegador
if (typeof(Storage) !== "undefined") {

    localStorage.setItem("apellido", "Majano");
    // Recuperar
    document.getElementById("result").innerHTML = localStorage.getItem("apellido");
} else {
    document.getElementById("result").innerHTML = "Lo sentimos, su navegador no es
compatible con Web Storage ...";
}
</script>

</body>
</html>
```

SesionStorage

```
<!DOCTYPE html>
<html>
<body>

<div id="resultado"></div>

<script>
// Verificación del soporte del navegador
if (typeof(Storage) !== "undefined") {

    sessionStorage.setItem("apellido", "Majano");
    // Recuperar
    document.getElementById("resultado").innerHTML =
sessionStorage.getItem("apellido");
} else {
    document.getElementById("resultado").innerHTML = "Lo sentimos, su navegador no
es compatible con Web Storage ...";
}
</script>

</body>
</html>
```

Objetos LocalStorage y SesiónStorage

Ambos objetos proporcionan los mismos métodos y propiedades:

- `setItem(key, value)` – almacena el par clave/valor.
- `getItem(key)` –obtiene el valor por clave.
- `removeItem(key)` – elimina la clave con su valor
- `clear()` – borra todo.
- `key(index)` – obtiene la clave en una posición determinada
- `length` – número de elementos almacenados.