Manejadores de Bases de Datos

DOCUMENTOS JSON

Bases de datos no estructuradas

JSON significa: **J**ava**S**cript **O**bject **N**otation.

- Es empleado para almacenar e intercambiar información.
- Es texto escrito en sintaxis de Javascript.

¿Qué es Javascript?

Bases de datos no estructuradas

- **¿Qué es Javascript?**

JavaScript es un robusto lenguaje de programación que puede ser aplicado a un documento HTML y usado para crear interactividad dinámica en los sitios web

Su nombre oficial es **ECMAScript.**

Bases de datos no estructuradas

- Su primera versión fue lanzada en 1995, junto con Netscape Navigator 2.0.
- En 1997 se envió la especificación JavaScript 1.1 al organismo ECMA (European Computer Manufacturers Association).
- JavaScript es la implementación que realizó la empresa Netscape del estándar ECMAScript.

Bases de datos no estructuradas

- **¿Cómo se agrega a una página Web?**

- Dentro de la página Web, el código debe ser insertado entre las etiquetas <script> y </script> y puede colocarse:

En la etiqueta <Body>

Indica página HTML

Indica el cuerpo de la página HTML

Bloque de javascript

```
<!DOCTYPE html>
<html>

<body>
    <h1>Ejemplo 1</h1>
    <p id="prueba">Erika</p>
    <button type="button" onclick="miFuncion()">Cambiar
    nombre</button>

    <script>
        function miFuncion() {
            document.getElementById("prueba").innerHTML =
            "Erika Meneses Rico.";
        }
    </script>

</body>

</html>
```

Bases de datos no estructuradas

Continuando con JSON…

Bases de datos no estructuradas

Ejemplo: XML vs JSON

XML

```xml
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON

```json
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

- Self-describing
- Jerárquico

Bases de datos no estructuradas

JSON

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

- Los datos se encuentran en pares nombre/valor, clave/valor, llave/valor.
- Los datos son separados por comas (,).
- Los objetos son encerrados entre llaves ({ }).
- Los paréntesis cuadrados almacenan arreglos o arrays.

Bases de datos no estructuradas

“name”: “Jhon”

- Un par nombre/valor consiste en un campo nombre encerrado entre comillas dobles, seguido de dos puntos y un valor.

Bases de datos no estructuradas

Los valores pueden ser del siguiente tipo:

- Una cadena.
- Un número.
- Un objeto.
- Un arreglo.
- Un booleano.
- Nulo

Bases de datos no estructuradas

Los valores pueden ser del siguiente tipo:

- Una cadena.
  - { "name":"John" }
- Un número.
  - { "age":30 }
- Un objeto.

```
{
"employee":{ "name":"John", "age":30, "city":"New York" }
}
```

Bases de datos no estructuradas

Los valores pueden ser del siguiente tipo:

- Un arreglo.

```
{
"employees":[ "John", "Anna", "Peter" ]
}
```

- Un booleano.
  - { ”car":true }
- Nulo
  - { "middlename":null }

Bases de datos no estructuradas

Objetos en JSON

```
{ "name":"John", "age":30, "car":null }
```

- Los objetos son escritos entre llaves { }.
- Son escritos en pares llave/valor.
- Las llaves son cadenas y los valores deben ser un tipo de dato válido en JSON (cadena, número, objeto, arreglo, booleano o nulo).
- Las llaves y valores se separan por dos puntos :
- Cada par llave/valor se separa con una coma ,

Bases de datos no estructuradas

Los valores de un objeto pueden ser accedidos a través de la notación de punto (.):

```
myObj = { "name":"John", "age":30, "car":null };
x = myObj.name;
```

Ejemplo: https://www.w3schools.com/js/tryit.asp?filename=tryjson_object_dot

Bases de datos no estructuradas

Con la notación de paréntesis cuadrados ([ ]):

```
myObj = { "name":"John",  "age":30, "car":null  };
x = myObj["name"];
```

Ejemplo:

https://www.w3schools.com/js/tryit.asp?filename=tryjson_object_bracket

Bases de datos no estructuradas

Los valores pueden ser otro objeto:

```
myObj = {
  "name":"John",
  "age":30,
  "cars":  {
     "car1":"Ford",
      "car2":"BMW",
      "car3":"Fiat"
   }
 }
```

Bases de datos no estructuradas

Se pueden acceder a ellos a través de la siguiente notación:

```
x = myObj.cars.car2;
// o:
x = myObj.cars["car2"];
```

Ver ejemplo:
https://www.w3schools.com/js/tryit.asp?filename=tryjson_object_nested

Bases de datos no estructuradas

Arreglos en JSON

```
[ "Ford", "BMW", "Fiat" ]
```

Los arreglos pueden ser valores:

```
{
"name":"John",
"age":30,
"cars":[ "Ford", "BMW", "Fiat" ]
}
```

Bases de datos no estructuradas

Para acceder a los valores de un arreglo:

```
x = myObj.cars[0];
```

Ver ejemplo:

https://www.w3schools.com/js/tryit.asp?filename=tryjson_array_access

Bases de datos no estructuradas

Para acceder a los valores de un arreglo:

x = myObj.cars[0];

- Los arreglos pueden anidarse:

```
myObj = {
  "name":"John",
  "age":30,
  "cars": [
    { "name":"Ford", "models":[ "Fiesta", "Focus", "Mustang" ] },
    { "name":"BMW", "models":[ "320", "X3", "X5" ] },
    { "name":"Fiat", "models":[ "500", "Panda" ] }
  ]
}
```

Manejadores de Bases de Datos

Gracias por su atención