|  |  |  |
| --- | --- | --- |
| 6) | JSON supports array.  JSON types: Number, Array, Boolean, String | XML doesn't support array.  XML data are all string |
| 7) | JSON is less secured than XML. | XML is more secured. |

JSON Object

**To convert JSON text into an object “eval” function is used.**

JSON object holds key/value pair. Each key is represented as a string in JSON and value can be of any type.

1**. The keys and values are separated by colon.**

**2. Each key/value pair is separated by comma.**

**3. The string value must be enclosed within double quote.**

**4. The curly brace { represents JSON object.**

Example:

{

    "employee": {

        "name":       "sonoo",

        "salary":      56000,

        "married":    **true**

    }

}

employee is an object in which "name", "salary" and "married" are the key.

## **JSON Object with Strings**

The string value must be enclosed within double quote.

{

        "name":       "sonoo",

        "email":      "sonoojaiswal1987@gmail.com"

}

## **JSON Object with Numbers**

{

"integer": 34,

"fraction": .2145,

"exponent": 6.61789e+0

}

## **JSON Object with Booleans**

{

"first": **true**,

"second": **false**

}

# JSON Array

JSON array represents ordered list of values. JSON array can store multiple values. It can store string, number, boolean or object in JSON array.

In JSON array, values must be separated by comma.

The **[** (square bracket) represents JSON array.

## **JSON Array of Strings**

["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

## **JSON Array of Objects**

{"employees":[

    {"name":"Ram", "email":"ram@gmail.com", "age":23},

    {"name":"Shyam", "email":"shyam23@gmail.com", "age":28},

    {"name":"John", "email":"john@gmail.com", "age":33},

    {"name":"Bob", "email":"bob32@gmail.com", "age":41}

]}

PHP Object can be converted into JSON by using the PHP function **json\_encode():**

Example:

<?php  
$myObj->name = "John";  
$myObj->age = 30;  
$myObj->city = "New York";  
  
$myJSON = json\_encode($myObj);  
  
echo $myJSON;  
?>

Use **JSON.parse()** to convert the resul(Json file)t into a **JavaScript object**:

Convert a JavaScript object into a string with **JSON.stringify().**