UTTARANCHAL UNIVERSITY

JSON
Unit-II

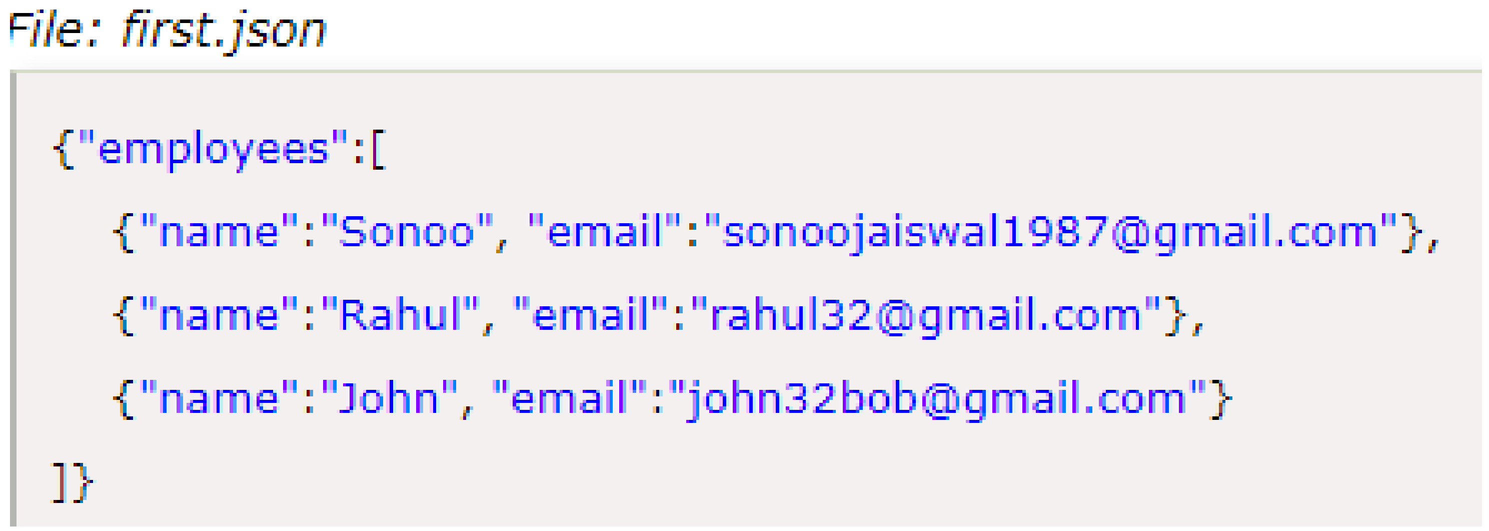
By
Amar Jeet Rawat
Assistant Professor,
Department of Computer Application
UIM

Introduction to JSON

- JSON is an open standard for exchanging data on the web.
- It supports data structures like object and array. So it is easy to write and read data from JSON.
- JSON stands for JavaScript Object Notation.
- JSON is lightweight data-interchange format.
- JSON is easy to read and write than XML.
- JSON is language independent.
- JSON supports array, object, string, number and values.

JSON Example

- The JSON file must be save with .json extension. Let's see a simple JSON example.



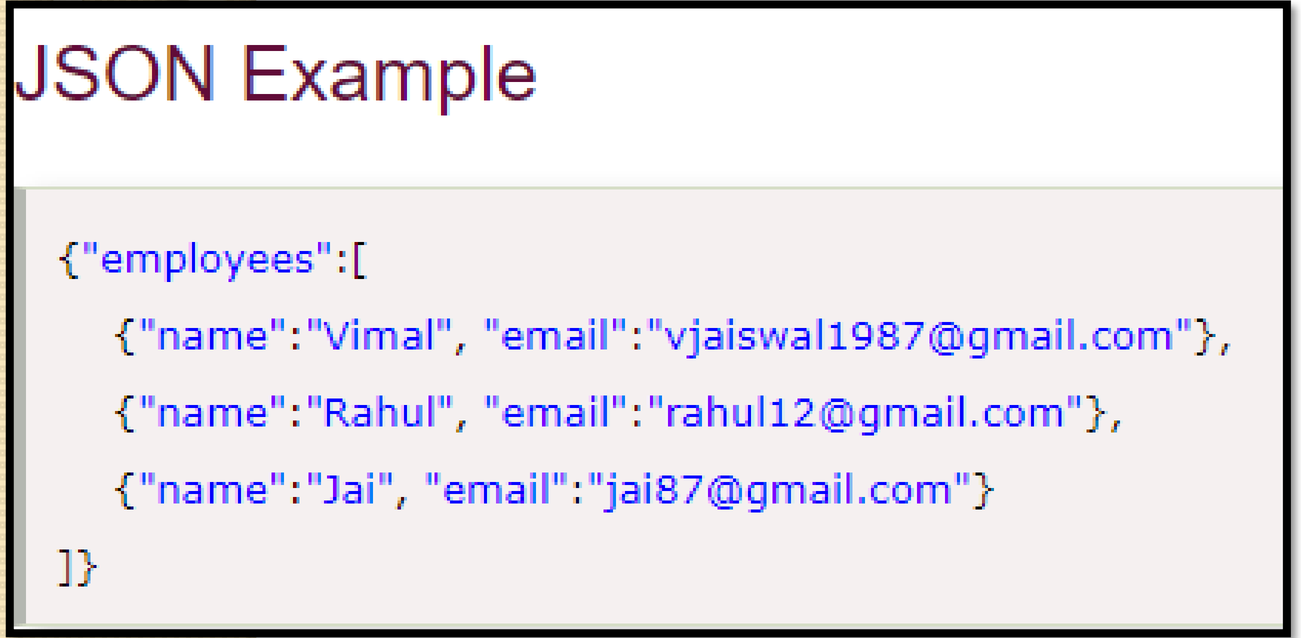
File: first.json

```
{"employees":[
    {"name":"Sonoo", "email":"sonoojaiswal1987@gmail.com"},
    {"name":"Rahul", "email":"rahul32@gmail.com"},
    {"name":"John", "email":"john32bob@gmail.com"}
]}
```

JSON v/s XML

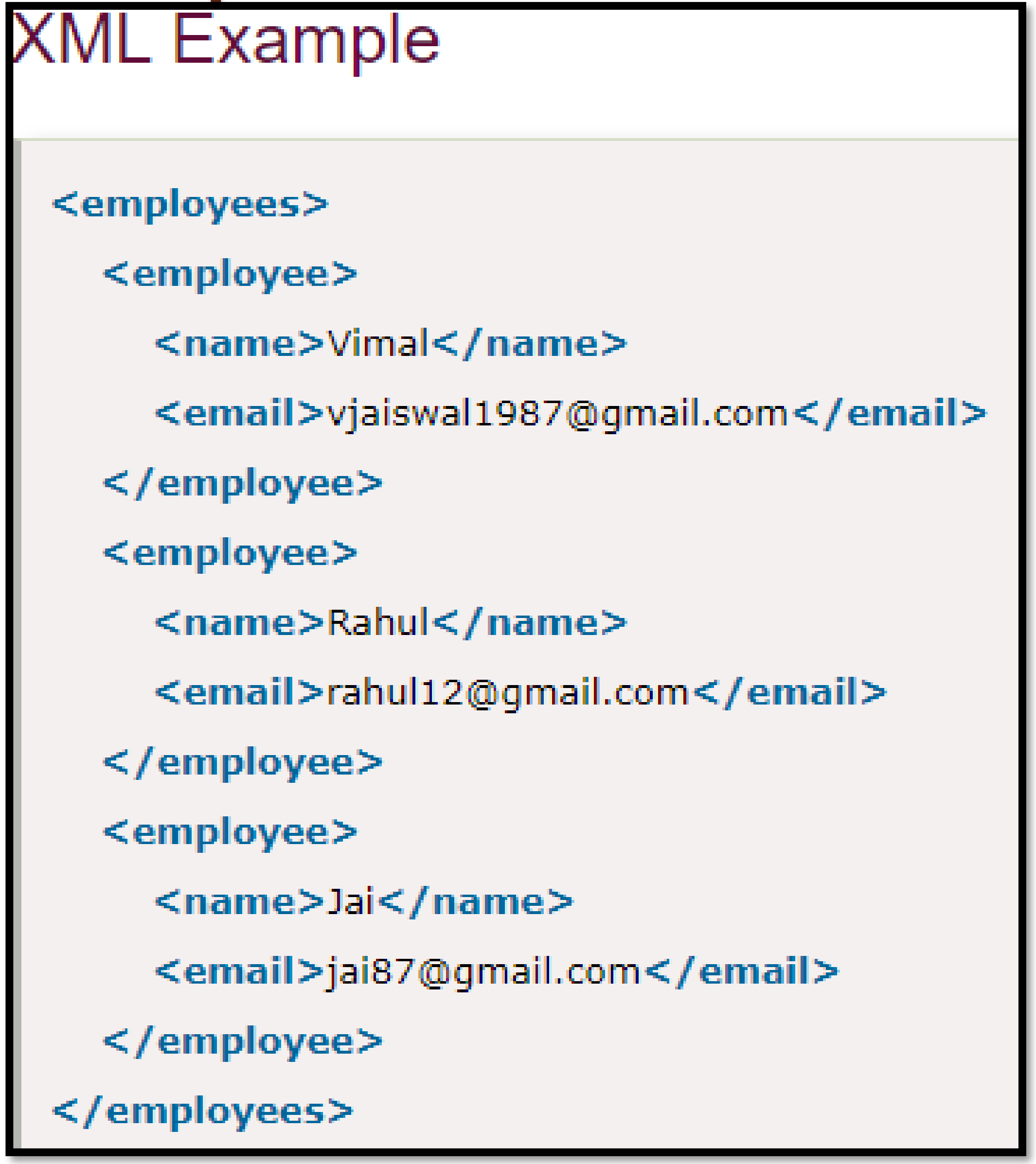
#	JSON	XML
1)	JSON stands for JavaScript Object Notation.	XML stands for eXtensible Markup Language.
2)	JSON is simple to read and write.	XML is less simple than JSON.
3)	JSON is easy to learn.	XML is less easy than JSON.
4)	JSON is data-oriented.	XML is document-oriented.
5)	JSON doesn't provide display capabilities.	XML provides the capability to display data because it is a markup language.
6)	JSON supports array.	XML doesn't support array.
7)	JSON is less secured than XML.	XML is more secured.
8)	JSON files are more human readable than XML.	XML files are less human readable.
9)	JSON supports only text and number data type.	XML support many data types such as text, number, images, charts, graphs etc. Moreover, XML offers options for transferring the format or structure of the data with actual data.

JSON v/s XML : By Example



JSON Example

```
{"employees":[
   {"name":"Vimal", "email":"vjaiswal1987@gmail.com"},
   {"name":"Rahul", "email":"rahul12@gmail.com"},
   {"name":"Jai", "email":"jai87@gmail.com"}
]}
```



XML Example

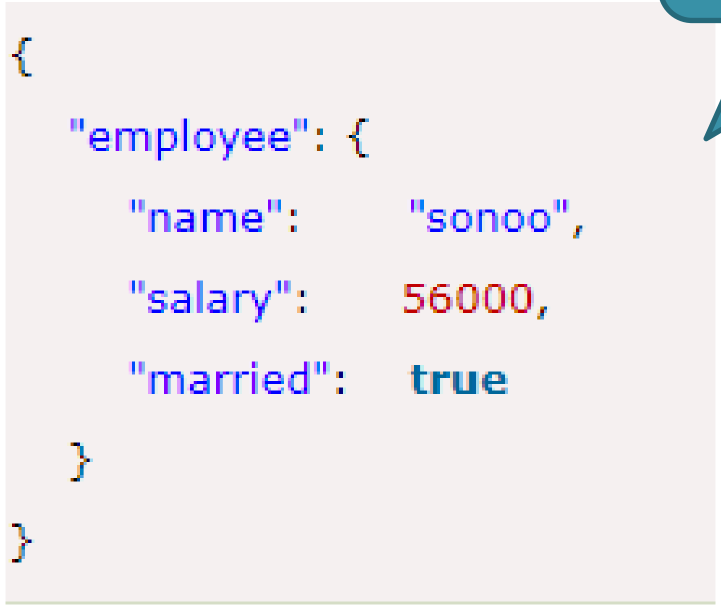
```
<employees>
   <employee>
      <name>Vimal</name>
      <email>vjaiswal1987@gmail.com</email>
   </employee>
   <employee>
      <name>Rahul</name>
      <email>rahul12@gmail.com</email>
   </employee>
   <employee>
      <name>Jai</name>
      <email>jai87@gmail.com</email>
   </employee>
</employees>
```

Creating JSON

- JSON can be created by object and array.
- Each object can have different data such as text, number, boolean etc
  - JSON Object
    - A JSON object contains data in the form of key/value pair.
    - The keys are strings and the values are the JSON types. Keys and values are separated by colon.
    - Each entry (key/value pair) is separated by comma.
  - JSON Array
    - The **[** (square bracket) represents the JSON array. A JSON array can have values and objects.

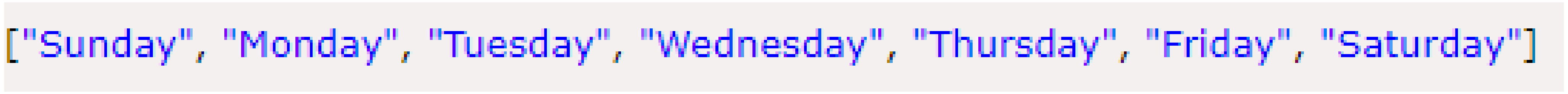
Examples JSON:

JSON Object Example



```
{
   "employee": {
      "name":       "sonoo",
      "salary":     56000,
      "married":    true
   }
}
```

JSON Array Example



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

JSON Functions : parse()

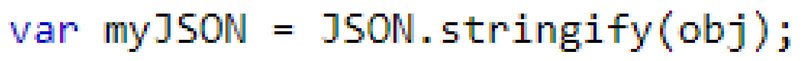
- JSON data, which exchange to/from a web server in always a string.
- To parse the JSON string into a Java Script Object we user JSON.parse() function.
- JSON.parse() convert text into java script object.
- Syntax :
    - Var objName= JSON.parse(JSON Data);
    - Example : var obj=JSON.parse({“Name”:“JOHN”, ”Age”:33});
- You can request JSON from the server by using an AJAX request
- When using the JSON.parse() on a JSON derived from an array, the method will return a JavaScript array, instead of a JavaScript object.

Exception : JSON.parse()

- Parsing date
  - Date objects are not allowed in JSON.
  - If you need to include a date, write it as a string.
  - You can convert it back into a date object later:
- Parsing function
  - Functions are not allowed in JSON.
  - If you need to include a function, write it as a string.
  - You can convert it back into a function later:

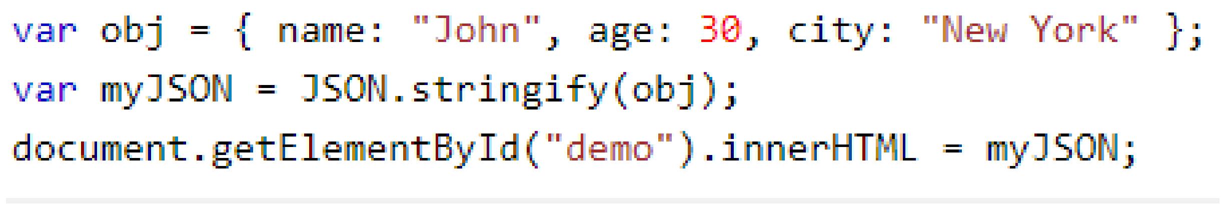
JSON Function: stringify()

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



```
var myJSON = JSON.stringify(obj);
```

- Example :



```
var obj = { name: "John", age: 30, city: "New York" };
var myJSON = JSON.stringify(obj);
document.getElementById("demo").innerHTML = myJSON;
```

Stringify a JavaScript Array and Dates

Java Script Array

```
var arr = [ "John", "Peter", "Sally", "Jane" ];
```

Use the JavaScript function JSON.stringify() to convert it into a string.

Example:

```
var arr = [ "John", "Peter", "Sally", "Jane" ];
var myJSON = JSON.stringify(arr);
document.getElementById("demo").innerHTML = myJSON;
```

Java Script Date can be converted using stringify()

```
var obj = { name: "John", today: new Date(), city : "New York" };
var myJSON = JSON.stringify(obj);

document.getElementById("demo").innerHTML = myJSON;
```