

JSON

# WHAT IS JSON?

- JavaScript Object Notation.
- Format to exchange data.
- Was born as an alternative to XML.
- It's easy use in JS and jQuery has gained lots of followers and most applications use it as its main way to exchange data between server and cliente.
- Language-independent.

# WHY USE JSON?

- Standard structure
  - JSON objects have a standard structure that makes developers job easy to read and write code.
- Light weight: When working with AJAX, it's important to load data quickly .
- Scalable: JSON is language independent, which means it can work well with most of the modern programming language.

# JSON SYNTAX

- Unordered sets of name/value pairs.
- Begins with left brace {
- Ends with right brace }
- Each name is followed by :
- Name / Value pairs are separated by ,

# JSON SYNTAX

```
var employeeData = {  
  "employee_id": 1234567,  
  "name": "Jeff Fox",  
  "hire_date": "1/1/2013",  
  "location": "Norwalk, CT",  
  "consultant": false  
};
```

# JSON VALUES

- The types we find in JSON are:
  - Number (integer or float)
  - String (between simple quotes)
  - Boolean (true or false)
  - Array (between square brackets[])
  - Object (between brackets {})
  - Null

# JSON ARRAYS

- An ordered collection of values
- Begins with [ (left square bracket)
- Ends with ] (right square bracket)
- Name/values pairs are separated by , (comma)

# JSON ARRAYS

```
var employeeData = {  
  "employee_id": 1236937,  
  "name": "Jeff Fox",  
  "hire_date": "1/1/2013",  
  "location": "Norwalk, CT",  
  "consultant": false,  
  "random_nums": [ 24, 65, 12, 94 ]  
};
```



# JSON VS XML

```
{ "students": [
  { "name": "John", "age": "23", "city": "Agra" },
  { "name": "Steve", "age": "28", "city": "Delhi" },
  { "name": "Peter", "age": "32", "city": "Chennai" },
  { "name": "Chaitanya", "age": "28", "city": "Bangalore" }
]}
```

```
<students>
  <student>
    <name>John</name> <age>23</age> <city>Agra</city>
  </student>
  <student>
    <name>Steve</name> <age>28</age> <city>Delhi</city>
  </student>
  <student>
    <name>Peter</name> <age>32</age> <city>Chennai</city>
  </student>
  <student>
    <name>Chaitanya</name> <age>28</age> <city>Bangalore</city>
  </student>
</students>
```

# HOW USE IT IN JS / JQUERY

- <http://api.jquery.com/jquery.parsejson/>
- Modern browsers include their own JSON implementation, becoming part of the native code.
- `JSON.parse(jsonString)` □ deserialize a JSON object into a JS object.

```
var data = '{"name": "mkyong", "age":  
  
var json = JSON.parse(data);  
  
alert(json["name"]); //mkyong  
alert(json.name); //mkyong
```

# HOW USE IT IN JS / JQUERY

- `JSON.stringify(myObject)` □ serialize a JS object into JSON

```
var myObject = new Object();  
myObject.name = "John";  
myObject.age = 12;  
myObject.pets = ["cat", "dog"];
```

```
var myString = JSON.stringify(myObject);
```

- <http://jsonviewer.stack.hu/>