JSON:

JSON stands for Javascript Object Notation

JSON objects are used for transferring data between server and client, XML serves the same purpose. However JSON objects have several advantages over XML and we are going to discuss them in this tutorial along with JSON concepts and its usages.

****Features of JSON****:

* It is light-weight
* It is language independent
* Easy to read and write
* Text based, human readable data exchange format

Structure

JSON structure is very standard and easy to understand

JSON is very lightweight when compared to XML which makes the response time optimal.

JSON is compatible with almost all programming languages.

Storage of data in JSON and XML format

{"students":[

{"name":"John", "age":"23", "city":"Agra"},

{"name":"Steve", "age":"28", "city":"Delhi"},

{"name":"Peter", "age":"32", "city":"Chennai"},

{"name":"Harry", "age":"29", "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>

JSON is lightweight when compared to XML and also supports arrays unlike XML

JSON objects

var student = {

"name" : “Sandeep”,

"age" : 21,

“sport”:”Basketball”

};

The above text creates an object that we can access using the variable chaitanya. Inside an object we can have any number of key-value pairs like we have above. We can access the information out of a JSON object  using objectname.property.

JSON objects in array

var students = [{

"name" : "Steve",

"age" : "29",

"gender" : "male"

},

{

"name" : "Peter",

"age" : "32",

"gender" : "male"

},

{

"name" : "Sophie",

"age" : "27",

"gender" : "female"

}];

It is an array of objects and can be accesed by students[i].property where i is the index of the array starting from zero.

Nesting JSON objects

var students = {

"steve" : {

"name" : "Steve",

"age" : "29",

"gender" : "male"

},

"pete" : {

"name" : "Peter",

"age" : "32",

"gender" : "male"

},

"sop" : {

"name" : "Sophie",

"age" : "27",

"gender" : "female"

}

}

Students.sop.name gives the name Sophie in this way json objects can be nested.

JSON to JavaScript Object

We have two ways to do this.  
1) Using eval function, but this is not suggested due to security reasons (malicious data can be sent from the server to the client and then eval in the client script with harmful effects).  
2) Using JSON parser: No security issues plus it is faster than eval.

JSON.parse(string)

stringify method converts javascript object into json format.

JSON.stringify(object)

In python it can be achieved by using dumps and loads functions.

json.loads(string) converts json to a python dictionary.

json.dumps(object) converts a python dictionary to json format.

Apart from that it also has to additional methods named dumps and loads which are used to load or dump data into a json files.