

JSON - XML

PROF. ME. HÉLIO
ESPERIDIÃO

JSON - JavaScript Object Notation

É uma formatação leve de troca de dados.

Para seres humanos, é fácil de ler e escrever.

Para máquinas, é fácil de interpretar e gerar.

É baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro - 1999.

JSON

JSON é em formato texto e completamente independente de linguagem

Formato ideal de troca de dados

é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida (parsing) entre sistemas

ESTRUTURA

Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um *object*, record, struct, dicionário, hash table, keyed list, ou arrays associativas.

Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma *array*, vetor, lista ou sequência.

EXEMPLO - JavaScript

```
var myObj = { "name": "John", "age": 31, "city": "New York" };  
var myJSON = JSON.stringify(myObj);  
window.location = "demo_json.php?x=" + myJSON;
```

EXEMPLO - JavaScript

```
var myJSON = '{ "name": "John", "age": 31, "city": "New York" }';  
var myObj = JSON.parse(myJSON);  
document.getElementById("demo").innerHTML = myObj.name;
```

Exemplo Array

```
<script>

var myObj, x,y;
myObj = {
    "name":"John",
    "age":30,
    "cars":[ "Ford", "BMW", "Fiat" ]
};
x = myObj.name;
y=myObj.cars[0];
document.getElementById("demo").innerHTML = x + " - " +y;

</script>
```

```
<script>

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

for (i in myObj.cars) {
  x += "<h2>" + myObj.cars[i].name + "</h2>";
  for (j in myObj.cars[i].models) {
    x += myObj.cars[i].models[j] + "<br>";
  }
}

document.getElementById("demo").innerHTML = x;

</script>
```

Varrendo todos as posições

Exemplo php

```
<?php
$myObj->name = "John";
$myObj->age = 30;
$myObj->city = "New York";

$myJSON = json_encode($myObj);

echo $myJSON;
?>
```

Retorna um object ou um array associativo se o parâmetro opcional assoc é TRUE.

```
<?php
$myArr = array("John", "Mary", "Peter", "Sally");

$myJSON = json_encode($myArr);

echo $myJSON;
?>
```

```
<?php
$json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

var_dump(json_decode($json));
var_dump(json_decode($json, true));

?>
```

A principal característica do XML, de criar uma infraestrutura única para diversas linguagens, é que linguagens desconhecidas e de pouco uso também podem ser definidas sem maior trabalho e sem necessidade de ser submetidas aos comitês de padronização.

O XML é um formato para a criação de documentos com dados organizados de forma hierárquica, como se vê, frequentemente, em documentos de texto formatados, imagens vetoriais ou bancos de dados.

XML (eXtensible Markup Language)

Vantagens

Separação do conteúdo da formatação

Simplicidade e legibilidade, tanto para humanos quanto para computadores

Possibilidade de criação de tags sem limitação

Criação de arquivos para validação de estrutura (chamados DTDs)

Interligação de bancos de dados distintos

Concentração na estrutura da informação, e não na sua aparência

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<receita nome="pão" tempo_de_preparo="5 minutos" tempo_de_cozimento="1 hora">
  <titulo>Pão simples</titulo>
  <ingredientes>
    <ingrediente quantidade="3" unidade="xícaras">Farinha de Trigo</ingrediente>
    <ingrediente quantidade="7" unidade="gramas">Fermento</ingrediente>
    <ingrediente quantidade="1.5" unidade="xícaras" estado="morna">Água</ingrediente>
    <ingrediente quantidade="1" unidade="colheres de chá">Sal</ingrediente>
  </ingredientes>
  <instrucoes>
    <passo>Misture todos os ingredientes, e dissolva bem.</passo>
    <passo>Cubra com um pano e deixe por uma hora em um local morno.</passo>
    <passo>Misture novamente, coloque numa bandeja e asse num forno.</passo>
  </instrucoes>
</receita>
```

Exemplo

```
<script>
var parser, xmlDoc;
var text = "<bookstore><book><title>Everyday Italian</title><author>Giada De Laurentiis</author>" +
"<year>2005</year></book></bookstore>";

parser = new DOMParser();
xmlDoc = parser.parseFromString(text, "text/xml");

document.getElementById("demo").innerHTML =
xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
</script>
```

XML em javascript

JSON VERSUS XML

O JSON pode ser considerado concorrente da XML na área de troca de informações.

Vejam algumas das principais semelhanças e diferenças entre os modelos de marcação das informações:

SEMELHANÇAS

Representam informações no formato texto.;

Ambos podem ser utilizados para transportar informações em aplicações AJAX.

Ambos são independentes de linguagem.

Dados representados em XML e JSON podem ser acessados por qualquer linguagem de programação, através de API's específicas.

DIFERENÇAS

Json Não é uma linguagem de marcação. (Não possui tags de abertura e de fechamento).

Json É tipicamente destinado para a troca de informações, enquanto XML possui mais aplicações.

Por exemplo: existem bancos de dados no formato XML e estruturados em SGBD XML nativo.