JSON

Quando começamos a estudar programação back-end, um termo que sempre aparece logo de cara é **JSON**. Mas o que é isso?

JSON é um acrônimo de ***JavaScript Object Notation*** ou "notação de objeto JavaScript". Como o próprio nome já sugere, JSON é um formato que utiliza a sintaxe de objetos e arrays do JavaScript. É muito versátil e se tornou a forma mais comum de estrutura para transferência de dados entre cliente/servidor, e tem sido utilizado mesmo em programas que não utilizam JavaScript. A estrutura também é mais fácil de compreender, comparada com outro formato de transferência de dados, o XML:

Formato JSON:

```
{
  "id": 59,
  "titulo": "ECMAScript 6",
  "autor": "Diego Martins de Pinho",
  "categoria": "programação"
}
```

Formato XML:

```
<livro id="59">
 <titulo>ECMAScript 6</titulo>
 <autor>Diego Martins de Pinho</autor>
 <categoria>programação</categoria>
</livro>
```

Você pode conferir mais sobre o XML aqui.

À primeira vista, um objeto **JSON** não parece muito diferente de um objeto literal em JavaScript:

```
{
 "editora": "Casa do Código",
 "catalogo": [
 {
  "id": 50,
  "titulo": "Primeiros Passos com NodeJS",
  "autor": "João Rubens",
  "categoria": "programação",
  "versoes": ["ebook", "impresso"]
```

```
},
{
  "id": 59,
  "titulo": "ECMAScript 6",
  "autor": "Diego Martins de Pinho",
  "categoria": "programação",
  "versoes": ["ebook"]
},
{
"id": 39,
"titulo": "Orientação a Objetos",
"autor": "Thiago Leite",
"categoria": "programação",
"versoes": ["ebook", "impresso"]
}
]}
```

O código acima mostra um JSON com dois conjuntos de propriedade/valor: um tem valor de **string (editora)** e **catálogo** é um array de objetos, cada um representando um livro.

As diferenças de sintaxe entre JSON e um objeto JavaScript são:

- No JSON, as chaves sempre devem estar entre aspas duplas **” ”**, no formato **string**. Já no objeto JavaScript, as aspas não são obrigatórias;
- O JSON aceita como valores apenas dados primitivos (**string**, **number** para números finitos, **true**, **false** e **null**), **objetos** e **arrays**. Não é possível declarar funções/métodos;
- Não são permitidas vírgulas após o último conjunto de chave/valor do objeto.

JSON é um formato criado para transferência de dados, sendo assim é necessário convertê-lo para um objeto JavaScript para que os dados possam ser utilizados em um programa. Para isso, existem dois métodos nativos:

- **JSON.parse():** converte JSON para um objeto JavaScript;
- **JSON.stringify():** converte um objeto JavaScript para o formato JSON.

Por exemplo, podemos converter um **objeto de livro** para o **JSON**:

```
const jsonLivro = JSON.stringify({
 id: 50,
 titulo: "Primeiros Passos com NodeJS",
 autor: "João Rubens",
```

```
 categoria: "programação",
 versoes: ["ebook", "impresso"]
})

console.log(jsonLivro)
```

O resultado é um string JSON:

```
{"id":50,"titulo":"Primeiros Passos com NodeJS","autor":"João Rubens","categoria":"programação","versoes":["ebook","impresso"]}
```

Para fazer o processo inverso:

```
const objLivro = JSON.parse(jsonLivro)
console.log(objLivro)
```

O resultado é um objeto JavaScript:

```
{
 id: 50,
 titulo: 'Primeiros Passos com NodeJS',
 autor: 'João Rubens',
 categoria: 'programação',
 versoes: [ 'ebook', 'impresso' ]
}
```

O processo de converter estruturas de dados em sequências de bytes ou caracteres. como no caso do JSON, é chamado de **serialização** (ou *marshaling* em algumas linguagens como Go).

O JSON é uma ferramenta imprescindível para o desenvolvimento web. Agora que você já sabe o que é e também sabe os métodos para acessar e percorrer objetos, já pode começar a praticar.