

INSTITUTO FEDERAL DE MATO GROSSO DO SUL
CAMPUS NOVA ANDRADINA

NOTAS DE AULA

LINGUAGEM E TÉCNICAS DE
PROGRAMAÇÃO

Prof. Me. Luiz F. Picolo

NOVA ANDRADINA - MS

Atualizado em 10 de maio de 2022

1 Definição sobre JavaScript Object Notation (JSON)

*Um ladrão rouba um tesouro, mas não furta a inteligência.
Uma crise destrói um herança, mas não uma profissão.
Não importa se você não tem dinheiro, você é uma pessoa rica,
pois possui o maior de todos os capitais: a sua inteligência.
Invista nela. **Estude!**
Augusto Cury*

JSON (JavaScript Object Notation) é um formato de intercâmbio de dados leve. É fácil para que seres humanos possam ler e escrever. É fácil para que as máquinas possam analisar e gerar. É baseado em um subconjunto do **Padrão de Linguagem de Programação JavaScript ECMA-262 3ª Edição** - Dezembro de 1999. JSON é um formato de texto completamente independente da linguagem, mas usa convenções que são familiares aos programadores da família C de linguagens, incluindo C, C ++, Java, JavaScript, Perl, Python e muitos outros. Essas propriedades tornam o JSON uma linguagem de intercâmbio de dados ideal¹.

Json também é um formato baseado em texto padrão para representar dados estruturados com base na sintaxe do objeto JavaScript. É comumente usado para transmitir dados em aplicativos da Web (por exemplo, enviar alguns dados do servidor para o cliente, para que possam ser exibidos em uma página da Web ou vice-versa). Você se deparará com isso com bastante frequência, portanto, neste artigo, oferecemos tudo o que você precisa para trabalhar com o JSON usando JavaScript, incluindo a análise do JSON para que você possa acessar os dados dentro dele e criar o JSON².

1.1 Estrutura JSON

Um JSON é uma string cujo formato se parece muito com o formato literal do objeto JavaScript. Você pode incluir os mesmos tipos de dados básicos dentro do JSON, como em um objeto JavaScript padrão — strings, números, matrizes, booleanos e outros literais de objeto. Isso permite que você construa uma hierarquia de dados.

¹ Para mais detalhes acesse: <https://www.json.org/json-pt.html>

² Para mais detalhes acesse: <<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>>

```
1 let Retangulo = {
2   "largura": 100,
3   "altura": 100
4 }
5
6 let Retangulos = [
7   {"largura": 100, "altura": 100},
8   {"largura": 200, "altura": 200},
9 ]
```

Assim, para realizar o acesso aos dados de um objeto Json, usamos a mesma notação **dot** / **bracket** usada em objetos literais.

```
1 let Retangulo = {
2   "largura": 100,
3   "altura": 100
4 }
5
6 let Retangulos = [
7   {"largura": 100, "altura": 100},
8   {"largura": 200, "altura": 200},
9 ]
10
11 console.log(Retangulo.largura)
12 console.log(Retangulo['largura'])
13
14 console.log(Retangulos[0].largura)
15 console.log(Retangulos[0]['largura'])
```

Contudo, o Json recebido venha em formato de String, ele deverá ser convertido para um objeto JavaScript. Assim, usando a função **JSON.parse()** podemos converter a string em um objeto, da seguinte forma:

```
1 let retangulo = Json.parse('{ "largura": 100, "altura": 100 }')
2 console.log(Retangulo.largura)
```

1.2 Exercícios de Fixação

1. No exercício 2.4 criamos uma classe **Noticia** que continha alguns atributos (titulo, dataDaPublicacao, resumo e texto) e uma outra classe **NoticiaDestaque** que continha, além dos atributos citados, um atributo imagem. Assim, baseando-se nos atributos listados, crie um arquivo **json** com nome **noticias.json** e atribua a ele uma lista de notícias no formato estudado. O arquivo deve conter no mínimo 5 notícias.

1.3 Obtendo JSON de um servidor web (Requisição Ajax)

Para obter o JSON, vamos usar uma API chamada **XMLHttpRequest** (geralmente chamada de XHR). Esse é um objeto JavaScript muito útil que nos permite fazer solicitações de rede para recuperar recursos de um servidor via JavaScript (por exemplo, imagens, texto, JSON e até trechos de código HTML), o que significa que podemos atualizar pequenas seções de conteúdo sem ter que recarregar toda página. Isso levou a páginas da Web mais responsivas e parece empolgante, mas está além do escopo deste artigo ensinar isso com muito mais detalhes³.

Para começar, vamos armazenar a URL do JSON que queremos recuperar em uma variável. Adicione o seguinte na parte inferior do seu código JavaScript:

```
1  let requestURL =  
    'https://raw.githubusercontent.com/luizpicolo/json-for-testing  
2  /main/dragonball.json';
```

Para criar uma solicitação, precisamos criar uma nova instância de objeto de solicitação a partir do construtor **XMLHttpRequest** usando a palavra-chave **new**. Adicione o seguinte abaixo sua última linha:

```
1  let request = new XMLHttpRequest();
```

Agora precisamos abrir uma nova solicitação usando o método **open()**. Adicione a seguinte linha:

```
1  request.open('GET', requestURL);
```

³ Para mais detalhes acesse: <<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>>

Isso leva pelo menos dois parâmetros — existem outros parâmetros opcionais disponíveis. Nós só precisamos dos dois obrigatórios para este exemplo simples:

- O método HTTP a ser usado ao fazer a solicitação de rede. Neste caso, GET é bom, pois estamos apenas recuperando alguns dados simples.
- O URL para fazer a solicitação — esta é a URL do arquivo JSON que armazenamos anteriormente.

Em seguida, adicione as duas linhas a seguir — aqui estamos definindo o `responseType` como JSON, para que o XHR saiba que o servidor retornará o JSON e que isso deve ser convertido nos bastidores em um objeto JavaScript. Em seguida, enviamos a solicitação com o método `send()`:

```
1 request.responseType = 'json';
2 request.send();
```

A última parte desta seção envolve aguardar a resposta retornar do servidor e, em seguida, lidar com ela. Adicione o seguinte código abaixo do seu código anterior:

```
1 request.onload = function() {
2     let personagens = request.response;
3
4     const elemento = document.getElementById('list');
5     let p1 = personagens.characters[0].name;
6     elemento.insertAdjacentHTML('afterbegin', p1);
7 }
```

1.4 Exercícios de Fixação

- A partir do arquivo **noticias.json** criado no exercício anterior, e utilizando também as classes **Noticia** e **NoticiaDestaque** implementadas, leia e apresente no navegador as notícias contidas no arquivo **noticias.json**. Para tanto, você deve utilizar implementar a leitura do arquivo e, a partir dos dados, criar objetos e apresentar os mesmos no navegador. Abaixo segue o modelo base para o desenvolvimento do trabalho

Referências

FARINELLI, F. Conceitos basicos de programação orientada a objetos. *Instituto Federal Sudeste de Minas Gerais*, 2007. Nenhuma citação no texto.

HAVERBEKE, M. *Eloquent JavaScript: a modern introduction to programming*. [S.l.]: No Starch Press, 2018. Nenhuma citação no texto.

PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Nenhuma citação no texto.

ROY, P. V. Programming paradigms for dummies: What every programmer should know. 04 2012. Nenhuma citação no texto.

RUIZ, E. E. S. *IBm1030 Programação Orientada a Objetos*. 2008. Disponível em: <http://dcm.ffclrp.usp.br/~evandro/ibm1030/constru/heranca.html>. Nenhuma citação no texto.

SILVA, D. Lucas da et al. Ontologias e unified modeling language: uma abordagem para representação de domínios de conhecimento. v. 10, 10 2009. Nenhuma citação no texto.

SOMMERVILLE, I. *Engenharia de Software*. 6ª. [S.l.: s.n.], 2003. Nenhuma citação no texto.

TRASVIÑA, F. *Introdução ao JavaScript Orientado a Objeto*. 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/conflicting/Learn/JavaScript/Objects>. Nenhuma citação no texto.