

A linguagem JavaScript. História e evolução do JavaScript. Ferramentas de desenvolvimento

APRESENTAR A LINGUAGEM JAVASCRIPT, SUA HISTÓRIA E EVOLUÇÃO, ASSIM COMO ALGUMAS FERRAMENTAS DE DESENVOLVIMENTO.

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR



JavaScript

JavaScript é uma linguagem de programação destinada a utilização em arquivos HTML. Sua finalidade é a manipulação, animação ou interação de uma página web. Muito dos elementos que de fato “criam vida” em uma página web é criado em JavaScript. Sua função não é apenas estética ou gráfica, esta linguagem pode realizar acesso e manipulação de dados externos, compondo e enriquecendo o seu HTML. Com JavaScript é possível criar funções para agilizar ou dinamizar a experiência de usabilidade aos usuários em ambiente web. Resumindo, o JavaScript é um assunto muito importante no mundo web.

História

JavaScript foi inventa por Brendan Eich, co-fundador do projeto Mozilla, em 1995, quando ainda trabalhava na Netscape. O nome original da linguagem chamava-se Mocha, alterado para LiveScript e por último para JavaScript.

A primeira versão da linguagem foi utilizada ainda no Netscape 2.0, em uma época que ficou conhecida com a “guerra dos browsers”.

A alteração do nome se deu na época do lançamento e adição da tecnologia Java no navegador Netscape, por meio de applets. Java é uma linguagem de programação robusta e orientada a objetos, desenvolvida pela Sun Microsystems. Esta estratégia de marketing causa até hoje certa confusão aos leigos, pois alguns acreditam que o JavaScript é “parente” ou descendente do Java.

Por suas características de manipulação da interface HTML, JavaScript ganhou reconhecimento com uma linguagem de script client-side, ou seja, que opera no cliente, neste caso no navegador. Diante desta popularidade, uma forte concorrência surge com a Microsoft, que não dominava na época a liderança entre os navegadores, lançando uma versão do JavaScript chamada Jscript. Este foi adicionado já no Internet Explorer 3.0, com a finalidade de prover as funcionalidades do JavaScript, bem melhorar alguns aspectos a fim de se diferenciar da concorrência.

A fim de se proteger e garantir a padronização do desenvolvimento da linguagem, a Netscape submeteu o JavaScript a Ecma (European Computer Manufacturers Association), a fim de se tornar um padrão na área. Após a aceitação a versão padronizada do JavaScript chama-se ECMAScript ou ECMA-262, aceita inicialmente em 1997. Esta ação de padronização trabalhou em aspectos com sintaxe, características da linguagem, performance e portabilidades. Atualmente está na 7ª versão (junho 2016).

O ECMAScript é o padrão e a base aos navegadores para implementação do JavaScript.

As implementações das novas edições do ECMAScript não são imediatas ou uniformes, uma vez que as atualizações nos navegadores não são simultâneas. Observar as atualizações aceitas nos navegadores é importante se você estiver utilizando recursos novos, e descritos inicialmente no ECMAScript.

É possível acompanhar este ritmo de implementação em site que monitoram este desenvolvimento.

Visite: <https://kangax.github.io/compat-table/es6/> (<https://kangax.github.io/compat-table/es6/>)

A seguir, a imagem da capa da especificação da linguagem oficial JavaScript, chamada de ECMA-262.

Standard ECMA-262

7th Edition / June 2016

**ECMAScript® 2016
Language Specification**

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

DICA:

Para saber mais sobre o ECMA-262, acesse:

<https://www.ecma-international.org/publications/standards/Ecma-262.htm> (<https://www.ecma-international.org/publications/standards/Ecma-262.htm>)
<https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
(<https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>)
<https://kangax.github.io/compat-table/es6/> (<https://kangax.github.io/compat-table/es6/>)

Evolução

Em um primeiro momento, o JavaScript estava associado a pequenas animações e validações em formulários na web.

No entanto, tecnologias como o AJAX e JSON impulsionaram o uso e desenvolvimento da linguagem. Somase a isso a criação de novas bibliotecas e frameworks, como o JQuery. Atualmente há diversos frameworks, com os mais diversos propósitos, tais como o Vue.js, Backbone.js, Angular.js, React, Node.js, Meteor, A linguagem cresceu e evoluiu, e está muito acima de ser apenas um script, podemos afirmar que é um linguagem completa e atual.

DICA:

Veja mais sobre frameworks em JavaScript:

<https://www.meteor.com/> (<https://www.meteor.com/>)
<https://nodejs.org/en/> (<https://nodejs.org/en/>)
<https://facebook.github.io/react/> (<https://facebook.github.io/react/>)
<https://angularjs.org/> (<https://angularjs.org/>)
<https://vuejs.org/> (<https://vuejs.org/>)
<https://documentcloud.github.io/backbone/> (<https://documentcloud.github.io/backbone/>)
<https://jquery.com/> (<https://jquery.com/>)

Esta evolução levou o JavaScript para além das animações e validações. Atualmente ele pode ser utilizado como tecnologia base a client-server, server-side, acesso a dados, e atualmente até para aplicações móveis.

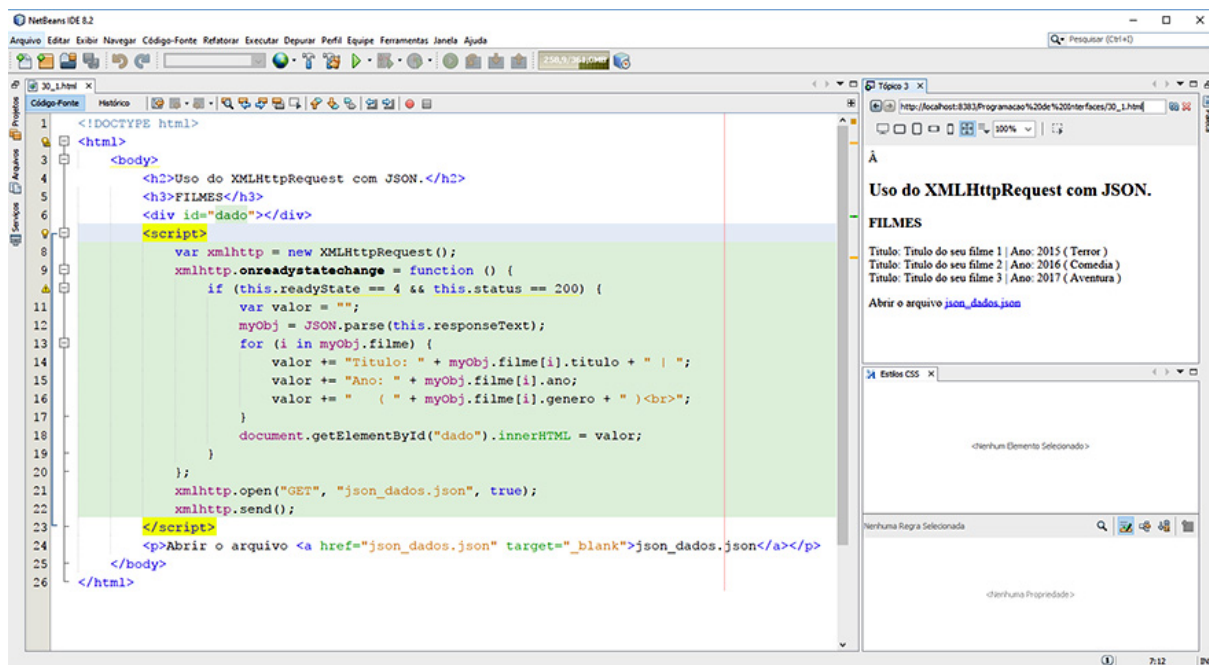
Ferramentas

Uma grande vantagem do JavaScript é que ele está disponível em quase todos os computadores. Pelo fato de sua execução se dar em navegadores, e não necessitar de servidores (ao menos para as funções básicas) a sua disponibilidade é enorme.

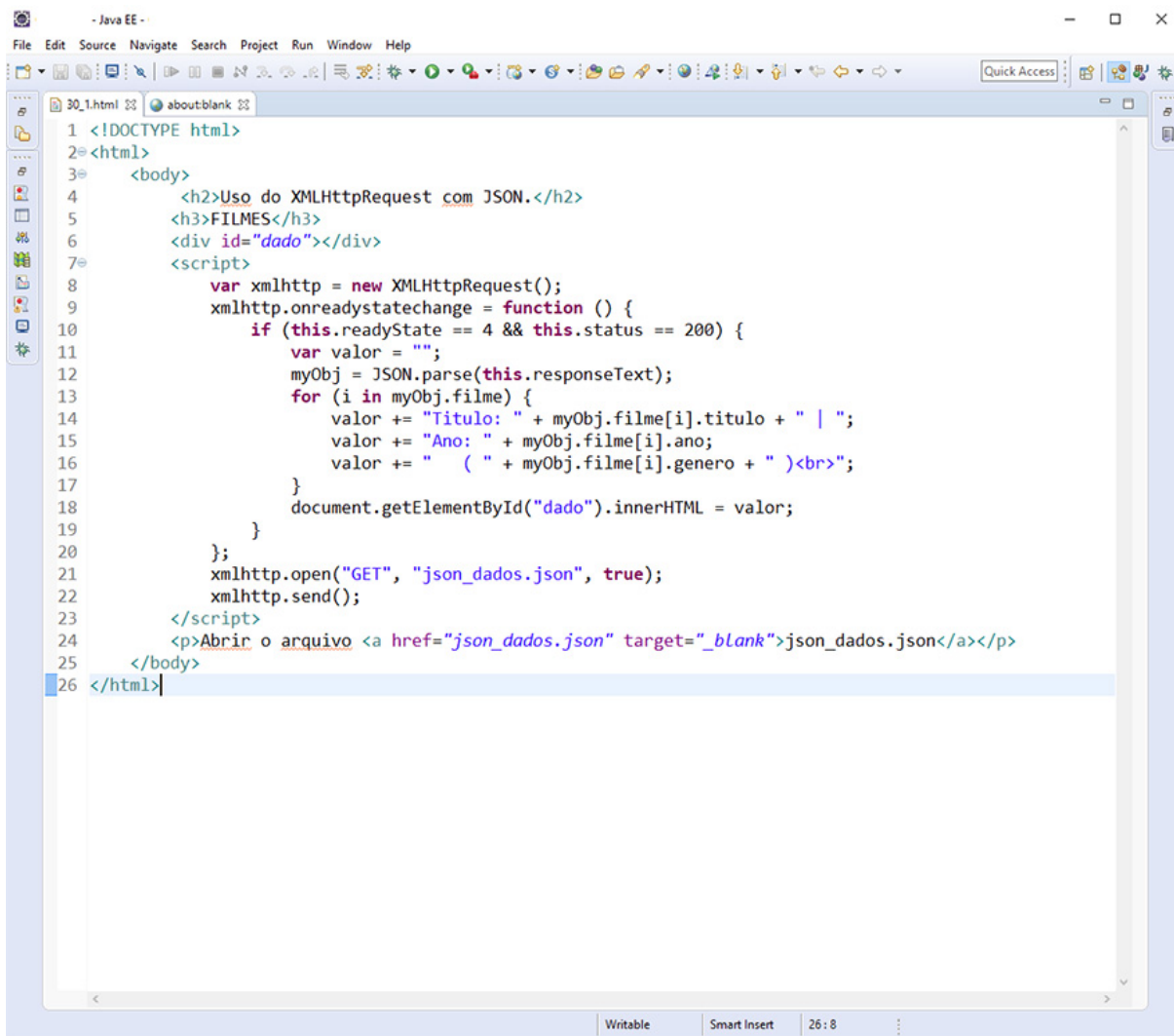
Neste sentido, as ferramentas de edição e programação acompanham esta tendência e já dão suporte a edição ao JavaScript. Até mesmo os navegadores aderiram a debuggers a JavaScript.

Veja algumas ferramentas úteis para o desenvolvimento e estudo no JavaScript:

IDE NetBeans:

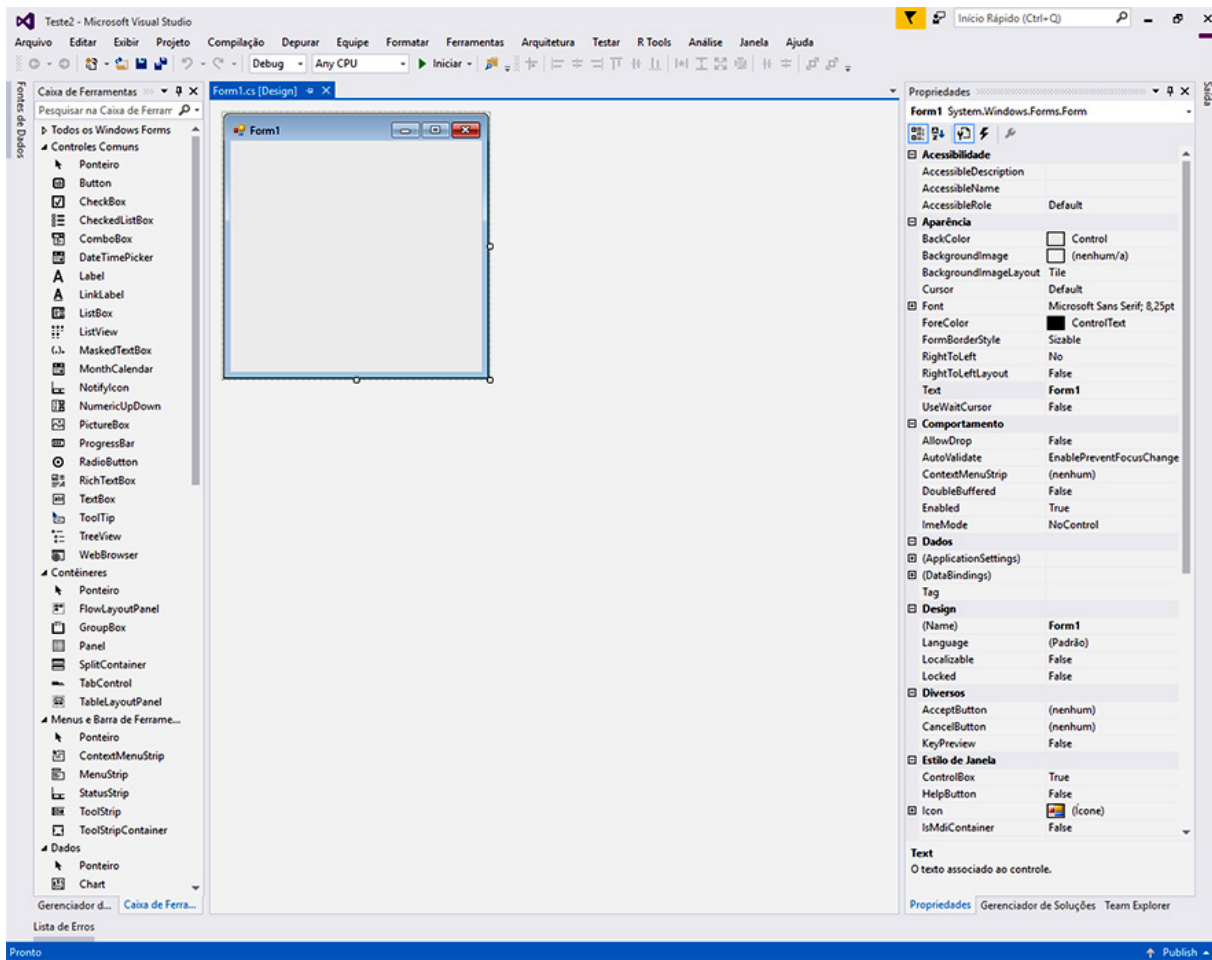


IDE Eclipse:



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Uso do XMLHttpRequest com JSON.</h2>
5     <h3>FILMES</h3>
6     <div id="dado"></div>
7     <script>
8       var xmlhttp = new XMLHttpRequest();
9       xmlhttp.onreadystatechange = function () {
10         if (this.readyState == 4 && this.status == 200) {
11           var valor = "";
12           myObj = JSON.parse(this.responseText);
13           for (i in myObj.filme) {
14             valor += "Titulo: " + myObj.filme[i].titulo + " | ";
15             valor += "Ano: " + myObj.filme[i].ano;
16             valor += " ( " + myObj.filme[i].genero + " )<br>";
17           }
18           document.getElementById("dado").innerHTML = valor;
19         }
20       };
21       xmlhttp.open("GET", "json_dados.json", true);
22       xmlhttp.send();
23     </script>
24     <p>Abrir o arquivo <a href="json_dados.json" target="_blank">json_dados.json</a></p>
25   </body>
26 </html>
```

IDE Visual Studio 2017:



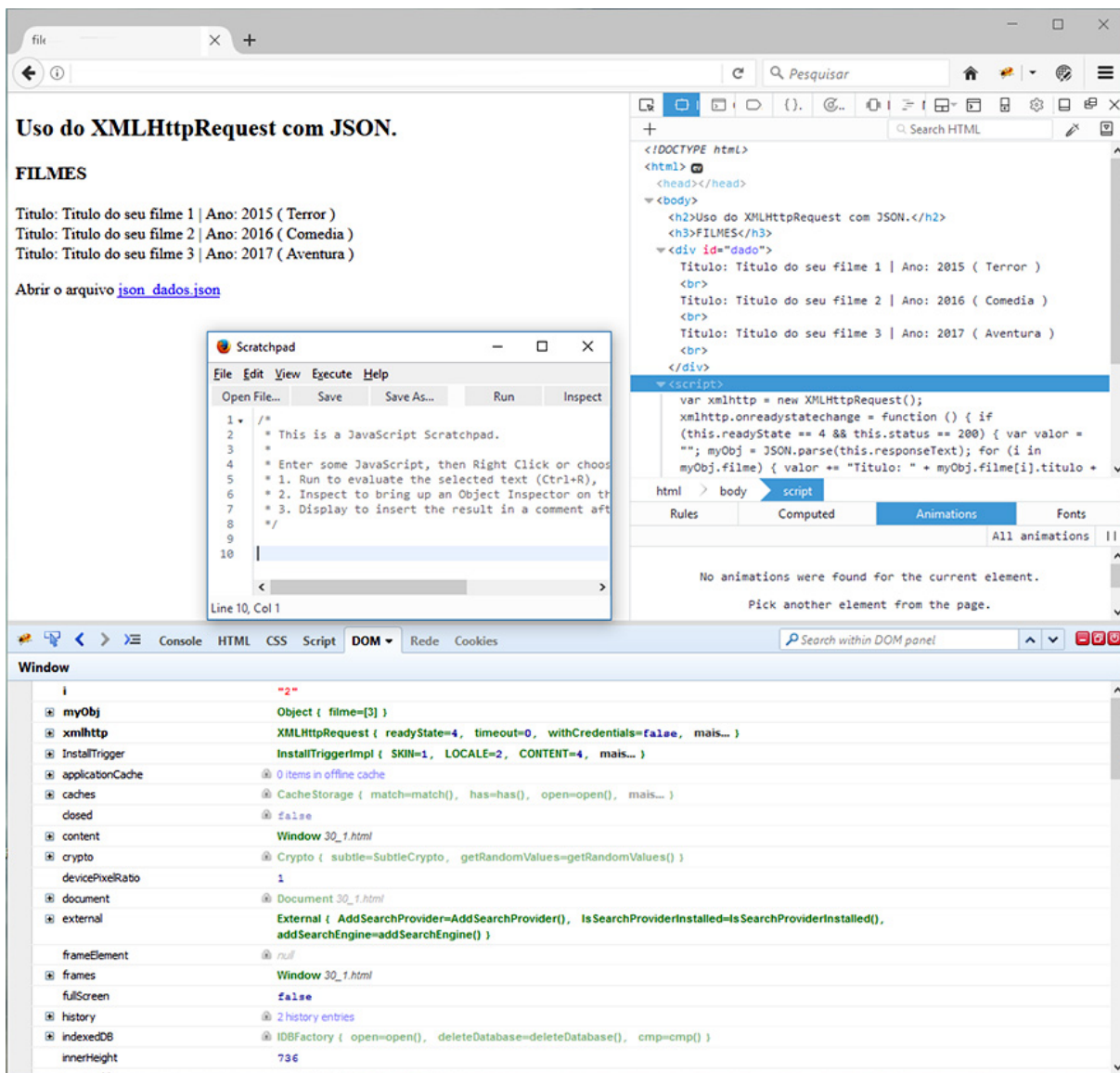
Editor Notepad++ :

```
Arquivo  Editar  Localizar  Visualizar  Formatar  Linguagem  Configurações  Macro  Executar  Plugins  Janela  ?
new 4  new 1  new 2  new 5  30_1.html
1  <!DOCTYPE html>
2  <html>
3  <body>
4  <h2>Uso do XMLHttpRequest com JSON.</h2>
5  <h3>FILMES</h3>
6  <div id="dado"></div>
7  <script>
8  var xmlhttp = new XMLHttpRequest();
9  xmlhttp.onreadystatechange = function () {
10     if (this.readyState == 4 && this.status == 200) {
11         var valor = "";
12         myObj = JSON.parse(this.responseText);
13         for (i in myObj.filme) {
14             valor += "Titulo: " + myObj.filme[i].titulo + " | ";
15             valor += "Ano: " + myObj.filme[i].ano;
16             valor += " ( " + myObj.filme[i].genero + " )<br>";
17         }
18         document.getElementById("dado").innerHTML = valor;
19     }
20 };
21 xmlhttp.open("GET", "json_dados.json", true);
22 xmlhttp.send();
23 </script>
24 <p>Abrir o arquivo <a href="json_dados.json" target="_blank">json_dados.json</a></p>
25 </body>
26 </html>
JavaScript file  length: 1.023  lines: 26  Ln: 23  Col: 13  Sel: 0|0  Unix (LF)  UTF-8  INS
```

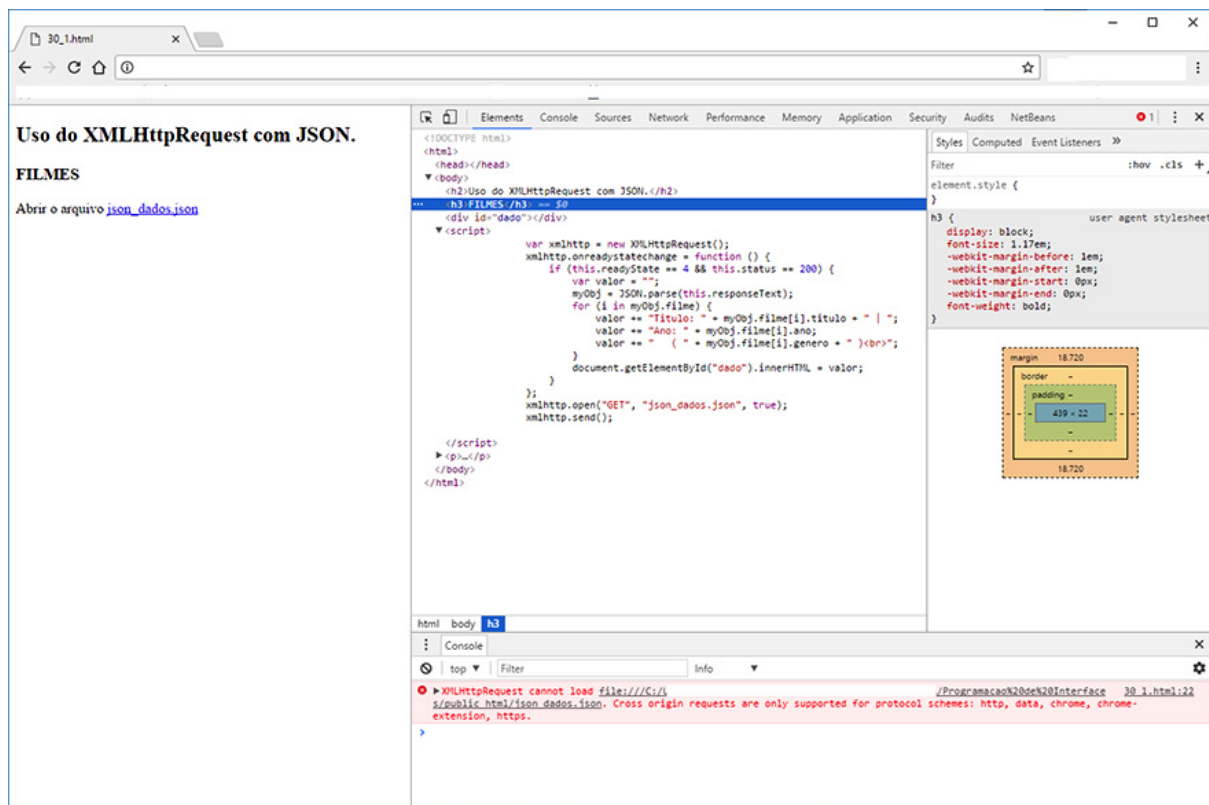
IDE Visual Studio Code :


```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Usado XMLHttpRequest com JSON.</h2>
5     <h3>FILMES</h3>
6     <div id="dado"></div>
7     <script>
8       var xmlhttp = new XMLHttpRequest();
9       xmlhttp.onreadystatechange = function () {
10         if (this.readyState == 4 && this.status == 200) {
11           var valor = "";
12           myObj = JSON.parse(this.responseText);
13           for (i in myObj.filme) {
14             valor += "Título: " + myObj.filme[i].titulo + " | ";
15             valor += "Ano: " + myObj.filme[i].ano;
16             valor += " ( " + myObj.filme[i].genero + " )<br>";
17           }
18           document.getElementById("dado").innerHTML = valor;
19         }
20       };
21       xmlhttp.open("GET", "json_dados.json", true);
22       xmlhttp.send();
23     </script>
24     <p>Abrir o arquivo <a href="json_dados.json" target="_blank">json_dados.json</a></p>
25   </body>
26 </html>
```

Navegador Mozilla Firefox :



Navegador Chrome :



Exemplo

Este é um exemplo simples de um código JavaScript, onde é realizado diversas operações matemáticas:

```

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 1</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 1)</div>
9.         <script type="text/javascript">
10.             var a = 10, b = 20, c = 30;
11.             var somar = a + b;
12.             var subtrair = a - b;
13.             var mult = a * b;
14.             var dividir = a/b;
15.             document.write("<br>a = " + a + " | b = " + b + " | c = " + c);
16.             document.write("<br>Somar (a+b) = " + somar );
17.             document.write("<br>Subtrair (a-b) = " + subtrair );
18.             document.write("<br>Multiplicar (a*b) = " + mult );
19.             document.write("<br>Dividir (a/b) = " + dividir );
20.             document.write("<br>(a/(a+c)) = " + a/(a+c) );
21.             document.write("<br>(b*a) /(a+c) = " + (b*a)/(a+c) );
22.         </script>
23.     </body>
24. </html>

```

Veremos muitos outros exemplos durante os tópicos desta disciplina. Aguarde.

Conclusão

JavaScript é uma linguagem de programação completa, simples, atual, de grande aceitação e utilidade nos dias atuais.

Seu nascimento e evolução se confunde com o próprio desenvolvimento da Web.

ATIVIDADE

Qual é o nome oficial da Linguagem e especificação JavaScript?

A. ECMA-2

- B. ScriptMA-262
- C. ECMA-262
- D. ECMA-262-script

ATIVIDADE

Qual e o nome do inventor do JavaScript?

- A. William Henry Gates III
- B. Steve Paul Jobs
- C. Brendan Eich
- D. Lawrence Joseph Ellison

ATIVIDADE

Escolha a opção abaixo que não representa um framework em JavaScript.

<https://angularjs.org/> (<https://angularjs.org/>)

<https://vuejs.org/> (<https://vuejs.org/>)

<https://documentcloud.github.io/backbone/>
(<https://documentcloud.github.io/backbone/>)