



Introdução à Linguagem JavaScript

Paradigmas de Linguagens de Programação

Gabriel Marques de Amaral Gravina

Ausberto S. Castro Vera

20 de setembro de 2021



Copyright © 2021 Gabriel Marques de Amaral Gravina e Ausberto S. Castro Vera

UENF - UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO

CCT - CENTRO DE CIÊNCIA E TECNOLOGIA

LCMAT - LABORATÓRIO DE MATEMÁTICAS

CC - CURSO DE CIÊNCIA DA COMPUTAÇÃO

Primeira edição, Maio 2019



Sumário

1	Introdução	5
1.1	Aspectos históricos da linguagem JavaScript	5
1.2	Áreas de Aplicação da Linguagem	5
1.2.1	NodeJS	6
1.2.2	Orientação a objetos	6
1.2.3	Programação Funcional	6
2	Conceitos básicos da Linguagem JavaScript	7
2.1	Variáveis e constantes	8
2.1.1	Tipos Primitivos	8
2.1.2	Tipos de Objeto	9
2.2	Tipos de Dados Básicos	9
2.2.1	String	9
2.3	Tipos de Dados de Coleção	10
2.3.1	Tipos Sequenciais	10
2.3.2	Tipos Conjunto	10
2.3.3	Tipos Mapeamento	10
2.4	Estrutura de Controle e Funções	10
2.4.1	O comando IF	10
2.4.2	Laço FOR	10
2.4.3	Laço WHILE	10
2.5	Módulos e pacotes	10
2.5.1	Módulos	10
2.5.2	Pacotes	10

Bibliografia	11
Index	13



1. Introdução

A linguagem de programação JavaScript é a “linguagem da web”. Seu uso é dominante na internet e praticamente quase todos os sites a utilizam. Além disso, smartphones, tablets e vários outros dispositivos têm interpretadores de JavaScript embutidos. Isso a torna uma das linguagens mais utilizadas dos dias atuais e uma das linguagens mais usadas por desenvolvedores de software. É importante dizer que, embora o nome sugira, JavaScript é uma linguagem completamente diferente e independente da linguagem Java. Mesmo assim, suas sintaxes tem traços de semelhança, mas nada além disso.

Por ser uma linguagem fácil de ser aprendida e fortemente tolerante, permitiu que usuários pudessem ter suas necessidades atendidas de forma cômoda e eficiente. A linguagem é de alto-nível, dinâmica e interpretada. Além disso, é adequada para orientação de objeto e programação funcional. É uma linguagem não tipada – ou seja, suas variáveis não tem um tipo específico e seus tipos não são importantes para a linguagem. Baseado no livro [Fla20].

1.1 Aspectos históricos da linguagem JavaScript

A linguagem foi criada na NETSCAPE por Brendan Eich. Tecnicamente, JavaScript é uma marca registrada da Sun Microsystems (atualmente Oracle) usada para descrever a implementação da língua pela Netscape (atualmente Mozilla). Na época, a Netscape enviou a linguagem para a padronização da ECMA – European Computer Manufacturer’s Association, sua versão padronizada ficou conhecida como “ECMAScript”. Na prática, todos chamam a linguagem apenas de JavaScript. De acordo com [Fla20].

1.2 Áreas de Aplicação da Linguagem

A linguagem JavaScript é completamente versátil e tem aplicações nos variados ambientes, seja no client-side ou no server-side. Nesta seção falarei de algumas aplicações e paradigmas da programação que podem ser implementados em JavaScript. De acordo com [Fla20].

1.2.1 NodeJS

A linguagem foi criada para ser utilizada em navegadores da web, e esse segue sendo seu ambiente mais comum de execução até hoje. Enfim, o ambiente do navegador permite a linguagem obter a entrada de usuários e fazer requests HTTP. Porém, em 2010 outro ambiente foi criado para executar código em JavaScript. O NodeJS, popularmente conhecido como Node, tinha a ideia de invés de manter a linguagem presa a um navegador, permitir que a linguagem tivesse acesso ao sistema operacional. Isso proporcionou a utilização da linguagem no lado do servidor, invés de se limitar apenas ao navegador. Atualmente, o Node tem grande popularidade na implementação de servidores web. Baseado no livro [Fla20].

1.2.2 Orientação a objetos

A linguagem é orientada a objeto, porém apresenta algumas diferenças que valem ser mencionadas. Na linguagem, as classes são baseadas no mecanismo de herança de protótipos. Se dois objetos herdam do mesmo objeto protótipo, então diz-se que são instâncias de uma mesma classe. Membros, ou instâncias da classe, tem suas propriedades para manter e também métodos que definem seu comportamento. Este comportamento é definido pela classe e compartilhado para todas as instancias. Retirado do artigo da documentação da linguagem, em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes>

1.2.3 Programação Funcional

Basicamente, a programação funcional é um paradigma da programação que visa produzir software através de funções puras, evitando compartilhamento de estados, dados mutáveis e efeitos colaterais. Embora JavaScript não seja uma linguagem de programação funcional como Haskell ou Lisp, o fato da linguagem poder manipular funções como objeto significa que técnicas de programação funcional podem ser implementadas na linguagem. Os métodos de array do ECMAScript 5, como `map()` e `reduce()` satisfazem bem o estilo de programação funcional. Retirado do livro [Pow15].



2. Conceitos básicos da Linguagem JavaScript

Os livros básicos para recomendados o estudo da Linguagem JavaScript são: [?], [?], [?] e [?]

Neste capítulo é serão apresentados os principais conceitos da linguagem JavaScript, sua estrutura léxica e.

Segundo [?], a linguagem JavaScript, . . .

De acordo com [?] e [?], a linguagem JavaScript . . .

[?] afirma que a linguagem JavaScript . . .

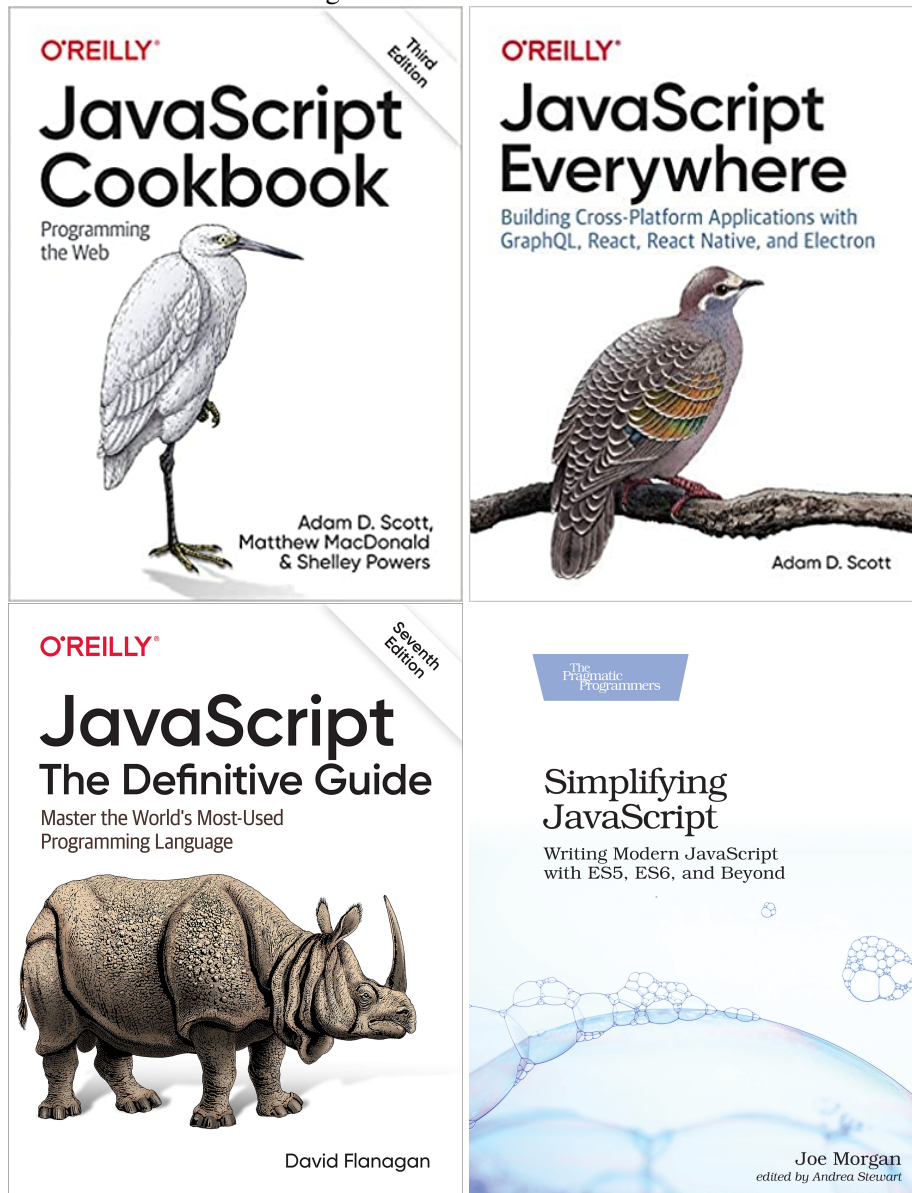
Considerando que a linguagem JavaScript ([?], [?]) é considerada como

A linguagem JavaScript é feita utilizando o set de caracteres Unicode, que dá suporte a praticamente todas as linguagens utilizadas atualmente no mundo. É uma linguagem case sensitive, ou seja, os nomes de variáveis, funções e outros identificadores devem ser sempre utilizados de maneira consistente, ao contrário do que acontece no html, por exemplo. Além disso, o JavaScript ignora os espaços e as quebras de linhas (com algumas exceções) que aparecem nos programas. Isso permite que os programas sejam identados de maneira que façam o código ser legível e fácil de entender. Falando em tornar o código legível, os comentários em JavaScript podem ser feitos de duas formas. Uma delas são os comentários de uma só linha, que utilizam "//" e a outra são os comentários de múltiplas linhas, que ignorarão tudo que está dentro dos caracteres. Exemplo:

```
1  /* Explicacao do codigo
2     0 codigo abaixo realiza... */
3  var helloWorld = function(){
4  console.log('Hello World!');
5  }
6  helloWorld();
```

No JavaScript o uso de vírgulas é opcional.

Figura 2.1: Melhores Livros



Fonte: O autor

2.1 Variáveis e constantes

Uma variável é, de forma resumida, um nome simbólico para um valor armazenado no computador. Quando chamamos uma variável, estamos acessando o valor guardado por ela. Na linguagem JavaScript, existem dois tipos de variáveis: as primitivas e as de objeto.

2.1.1 Tipos Primitivos

Os tipos primitivos do JS incluem números, strings de textos e valores booleanos (true e false). Os tipos especiais null e undefined são valores primitivos, porém não são números, strings ou booleanos. Cada um é considerado membro de um tipo especial.

Números

2.1.2 Tipos de Objeto

Na linguagem, qualquer valor que não seja um número, string, objeto ou null e undefined é um objeto. Um objeto é uma coleção de propriedades onde cda propriedade tem um nome e um valor.

2.2 Tipos de Dados Básicos

2.2.1 String

Um string é uma sequência de caracteres considerado como um item de dado simples. Para JavaScript, um string é um array de caracteres ou qualquer grupo de caracteres escritos entre dobre aspas ou aspas simples. Por exemplo,

```
1  <script type="text/javascript">
2  var rows = prompt("How many rows for your multiplication table?");
3  var cols = prompt("How many columns for your multiplication table?");
4  if(rows == "" || rows == null)
5      rows = 10;
6  if(cols== "" || cols== null)
7      cols = 10;
8  createTable(rows, cols);
9  function createTable(rows, cols)
10 {
11     var j=1;
12     var output = "<table border='1' width='500' cellspacing='0'
13     cellpadding='5'>";
14     for(i=1;i<=rows;i++)
15     {
16         output = output + "<tr>";
17         while(j<=cols)
18         {
19             output = output + "<td>" + i*j + "</td>";
20             j = j+1;
21         }
22         output = output + "</tr>";
23         j = 1;
24     }
25     output = output + "</table>";
26     document.write(output);
27 }
```

2.3 Tipos de Dados de Coleção

2.3.1 Tipos Sequenciais

2.3.2 Tipos Conjunto

2.3.3 Tipos Mapeamento

2.4 Estrutura de Controle e Funções

2.4.1 O comando IF

2.4.2 Laço FOR

2.4.3 Laço WHILE

2.5 Módulos e pacotes

2.5.1 Módulos

2.5.2 Pacotes

Código fonte para a linguagem JavaScript:

```
1      <!DOCTYPE html >
2      <html >
3      <body>
4
5      <h2>JavaScript Objects</h2>
6
7      <p id="demo"></p>
8
9      <script>
10     var person = {
11         firstName : "John",
12         lastName  : "Doe",
13         age       : 50,
14         eyeColor  : "blue"
15     };
16
17     document.getElementById("demo").innerHTML =
18     person.firstName + " is " + person.age + " years old.";
19 </script>
20
21 </body>
22 </html>
23
```



Referências Bibliográficas

- [Fla20] David Flanagan. *JavaScript : the definitive guide : master the world's most-used programming language*. O'Reilly Media, Sebastopol, CA, 2020. Citado 2 vezes nas páginas 5 e 6.
- [Pow15] Shelley Powers. *JavaScript cookbook : [programming the web]*. O'Reilly Media, Sebastopol, CA, 2015. Citado na página 6.

Disciplina: Paradigmas de Linguagens de Programação 2021

Linguagem: Linguagem JavaScript

Aluno: Gabriel Marques de Amaral Gravina

Ficha de avaliação:

Aspectos de avaliação (requisitos mínimos)	Pontos
Elementos básicos da linguagem (Máximo: 01 pontos) <ul style="list-style-type: none">• Sintaxe (variáveis, constantes, comandos, operações, etc.)• Usos e áreas de Aplicação da Linguagem	
Cada elemento da linguagem (definição) com exemplos (Máximo: 02 pontos) <ul style="list-style-type: none">• Exemplos com fonte diferenciada (Courier , 10 pts, azul)	
Mínimo 5 exemplos completos - Aplicações (Máximo : 2 pontos) <ul style="list-style-type: none">• Uso de rotinas-funções-procedimentos, E/S formatadas• Menu de operações, programas gráficos, matrizes, aplicações	
Ferramentas (compiladores, interpretadores, etc.) (Máximo : 2 pontos) <ul style="list-style-type: none">• Ferramentas utilizadas nos exemplos: pelo menos DUAS• Descrição de Ferramentas existentes: máximo 5• Mostrar as telas dos exemplos junto ao compilador-interpretador• Mostrar as telas dos resultados obtidos nas ferramentas• Descrição das ferramentas (autor, versão, homepage, tipo, etc.)	
Organização do trabalho (Máximo: 01 ponto) <ul style="list-style-type: none">• Conteúdo, Historia, Seções, gráficos, exemplos, conclusões, bibliografia	
Uso de Bibliografia (Máximo: 01 ponto) <ul style="list-style-type: none">• Livros: pelo menos 3• Artigos científicos: pelo menos 3 (IEEE Xplore, ACM Library)• Todas as Referências dentro do texto, tipo [ABC 04]• Evite Referências da Internet	
Conceito do Professor (Opcional: 01 ponto)	
Nota Final do trabalho:	

Observação: Requisitos mínimos significa a *metade* dos pontos