

PROGRAMAÇÃO DE SCRIPTS

4 História e Introdução à Linguagem JavaScript

4.1 JavaScript

O JavaScript foi criado “[...] na Netscape na fase inicial da Web” (1995), “[...] licenciada pela Sun Microsystems (agora Oracle)” (FLANAGAN, 2013, p. 2). “A Netscape enviou a linguagem para a ECMA – European Computer Manufacturer’s Association – para padronização e [...] a versão padronizada manteve o nome [...] ‘ECMAScript’ ” (FLANAGAN, 2013, p. 2).

JavaScript “[...] é a linguagem de programação da Web. A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos - em computadores de mesa, consoles de jogos, tablets e smartphones - incluem interpretadores JavaScript” (FLANAGAN, 2013, p. 1).

“O JavaScript move a chave que transforma uma página Web em uma experiência interativa” (MORRISON, 2008, p. 4). Assim,

[...] o JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas. [...] JavaScript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais. A sintaxe de JavaScript é derivada da linguagem Java, das funções de primeira classe de Scheme e da herança baseada em protótipos de Self (FLANAGAN, 2013, p. 1).

4.2 Estrutura léxica do JavaScript

“A estrutura léxica de uma linguagem de programação é o conjunto de regras elementares que especificam o modo de escrever programas nessa linguagem” (FLANAGAN, 2013, p. 21). Ela “[...] é a sintaxe de nível mais baixo de uma linguagem; especifica detalhes de como são os nomes de variáveis, os caracteres delimitadores para comentários e como uma instrução do programa é separada da seguinte” (FLANAGAN, 2013, p. 21).

A linguagem JavaScript “diferencia letras maiúsculas de minúsculas. Isso significa que palavras-chave, variáveis, nomes de função e outros identificadores da linguagem sempre devem ser digitados com a composição compatível de letras” (FLANAGAN, 2013, p. 21).

O “JavaScript ignora os espaços que aparecem entre sinais em programas. De modo geral, JavaScript também ignora quebras de linha” (FLANAGAN, 2013, p. 22).

O “JavaScript aceita dois estilos de comentários. Qualquer texto entre // e o final de uma linha é tratado como comentário e é ignorado por JavaScript. Qualquer texto entre os caracteres /* e */ também é tratado como comentário” (FLANAGAN, 2013, p. 23).

“Em JavaScript, os identificadores são usados para dar nomes a variáveis e funções e para fornecer rótulos para certos laços no código JavaScript. [...] Um identificador é simplesmente um nome” (FLANAGAN, 2013, p. 23). Ele também “reserva vários identificadores como palavras-chave da própria linguagem” (palavras-reservadas) (FLANAGAN, 2013, p. 23).

“Assim como muitas linguagens de programação, JavaScript usa o ponto e vírgula (;) para separar instruções”. Todavia, há a possibilidade de “omitir o ponto e vírgula entre duas instruções, caso essas instruções sejam escritas em linhas separadas” (FLANAGAN, 2013, p. 23).

4.3 JavaScript e sua relação com o HTML

Inicialmente, “[...] os navegadores Web têm uma parte especial do software dentro deles chamada interpretador JavaScript e seu serviço é executar o código JavaScript assim que aparece em uma página” (MORRISON, 2008, p. 13). Essa característica da linguagem JavaScript, quando está do lado cliente, faz com que ela, muitas vezes, seja referida como “interpretada”, mas há mais possibilidades (MORRISON, 2008, p. 13).

A tag utilizada para adicionar código JavaScript em uma página HTML é “[...] a tag <script>” (MORRISON, 2008, p. 11). Vale à pena destacar que a tag <script> também necessita ser fechada </script>.

* Apenas para fins educacionais.

BIBLIOGRAFIA

BASHAM, B. **Use a cabeça! Servlets e JSP**. Alta Books, 2008.

BORGES JÚNIOR, M. P. **ASP.NET utilizando C#**. Ciência Moderna Ltda., 2006.

FEDOSEJEV, A. **React.js Essentials**: A fast-paced guide to designing and build scalable and maintable web apps with React.js. Birmingham: Packt, 2015.

FERREIRA, S. **Guia prático de HTML5**. São Paulo: Universo dos Livros, 2013.

FLANAGAN, D. **JavaScript**: O guia definitivo. 6. ed. Porto Alegre: Bookman, 2013.

FREEMAN, E; FREEMAN, E. **Use a cabeça! HTML com CSS e XHTML**. 2. ed. Rio de Janeiro: Alta Books, 2008.

GAMMA, E. et al. **Padrões de projeto**. Porto Alegre: Bookman, 2005.

GUEDES, G. T. A. **UML 2**: uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2011.

KRUTZ, R. L. VINES, R. D. **The CISSP Prep Guide**: Mastering the CISSP and ISSEP Exams. 2. ed. Indianapolis: Wiley, 2004.

MACRAE, C. **Vue.js Up & Running**: Building accessible and performant web apps. 1. ed. Boston: O'Reilly, 2018.

MORRISON, M. **Use a cabeça! Javascript**. Alta Books, 2008.

RIORDAN, R. **Use a cabeça! Ajax profissional**. Alta Books, 2009.

SCHMITZ, D; GEORGII P. D. **Vue.js na Prática**. LeanPub, 2016.

SESHADRI, S. GREEN, B. **Desenvolvendo com AngularJS**: Aumento de produtividade com aplicações Web estruturadas. 1. ed. São Paulo: Novatec, 2014.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

WATRALL, E; SIARTO, J. **Use a cabeça! Web design**. Alta Books, 2009.

ZERVAAS, Q. **Aplicações Práticas de Web 2.0 com PHP**. Alta Books, 2009.