

# PROGRAMAÇÃO DE SCRIPTS

## 5 Variáveis e Operadores

### 5.1 Tipos em JavaScript

“Os programas de computador funcionam manipulando valores” (FLANAGAN, 2013, p. 28). “[...] Uma das características mais fundamentais de uma linguagem de programação é o conjunto de tipos que ela aceita” (FLANAGAN, 2013, p. 28).

Os tipos de JavaScript podem ser divididos em duas categorias: tipos primitivos e tipos de objeto. Os tipos primitivos de JavaScript incluem números, sequências de texto (conhecidas como strings) e valores de verdade (conhecidos como booleanos). [...] Os valores especiais null e undefined de JavaScript são valores primitivos, mas não são números, nem strings e nem booleanos. [...] Qualquer valor em JavaScript que não seja número, string, booleano, null ou undefined é um objeto. Um objeto (isto é, um membro do tipo objeto) é um conjunto de propriedades, em que cada propriedade tem um nome e um valor (FLANAGAN, 2013, p. 28).

“Ao contrário de algumas linguagens de programação, o JavaScript não permite definir explicitamente o tipo de uma constante ou variável. Em vez disso, o tipo fica implícito quando” se “[...] define o valor dos dados” (MORRISON, 2008, p. 49). Essa característica “[...] permite bastante flexibilidade às variáveis do JavaScript” (MORRISON, 2008, p. 49).

### 5.2 Variáveis em JavaScript

“Variável é um local de armazenamento na memória com um nome específico, como uma etiqueta em uma caixa usada para armazenar coisas” (MORRISON, 2008, p. 44).

“Ao criar uma variável usando a palavra-chave *var*, essa variável se encontra inicialmente vazia” (MORRISON, 2008, p. 44). Uma variável pode ser inicializada com “=” (MORRISON, 2008, p. 45). Exemplo: `var nome = “nome”`.

Na visão de Flanagan (2013, p. 28), “uma variável define um nome simbólico para um valor e permite que o valor seja referido pelo nome”.

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).

Em termos de regras para nomes, uma variável “não pode começar com um número; [...] não pode começar com um caractere especial diferente de `_` ou `$`” (MORRISON, 2008, p. 52). A comunidade JavaScript possui alguns padrões não oficiais” para a definição de nomes de variáveis (MORRISON, 2008, p. 52). “Variáveis normalmente usam letras minúsculas (camelCase), onde a primeira palavra é toda composta de letras minúsculas, mas as palavras adicionais possuem letras maiúsculas e minúsculas misturadas” (MORRISON, 2008, p. 52).

### 5.3 Constantes em JavaScript

“Constantes são resistentes à mudança”. Elas são utilizadas para “[...] armazenar um fragmento de dados que não muda. [...] O valor ‘inicial’ se tornará um valor permanente” (MORRISON, 2008, p. 46).

“Constantes são criadas” com “[...] a palavra-chave *const*” (MORRISON, 2008, p. 46). Exemplo: `const numero = 9`.

### 5.4 Operadores em JavaScript

“Os programas JavaScript trabalham com números usando os operadores aritméticos fornecidos pela linguagem. Isso inclui `+` para adição, `-` para subtração, `*` para multiplicação, `/` para divisão e `%` para módulo (resto da divisão)” (FLANAGAN, 2013, p. 32).

### 5.5 Depuração

“A depuração é parte integrante do trabalho de um auditor de código” (DOWD; MCDONALD; SCHUH, 2006, p. 153).

“O nível de sofisticação nos depuradores varia muito, assim como seus conjuntos de recursos” (DOWD; MCDONALD; SCHUH, 2006, p. 166).

[...] Todos os navegadores Web atuais contêm ferramentas para desenvolvedores Web que são indispensáveis para depuração, experimentação e aprendizado. Normalmente, essas ferramentas podem ser encontradas no menu Ferramentas do navegador, sob nomes como “Desenvolvedor Web” ou “Console da Web”. O Firefox contém um “Console da Web” interno[...]. Frequentemente, é possível ativar um console com um toque de tecla, como F12 ou Ctrl-Shift-J. Muitas vezes, essas ferramentas de console aparecem como painéis na parte superior ou inferior da janela do navegador, mas

alguns permitem abri-las como janelas separadas (FLANAGAN, 2013, p. 3).

## 5.6 console.log

“Existe uma API de console simples, implementada de forma portátil pelos navegadores modernos”, na qual o desenvolvedor “[...] pode usar a função console.log() para exibir texto no console” (FLANAGAN, 2013, p. 3). O uso desse recurso,

[...] muitas vezes é surpreendentemente útil ao se fazer depuração [...] para produzir saída simples. Uma maneira semelhante, porém mais invasiva, de exibir saída ou mensagens de depuração é passar uma string de texto para a função alert(), a qual as exibe em uma caixa de diálogo modal (FLANAGAN, 2013, p. 3).

\* 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.