

Linguagem de programação: Instruções e Objetos

Resumo

Referências Bibliográficas

- FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6ª Ed. Porto Alegre: Bookman, 2013.
- FREEMAN, Eric. **Use a cabeça!: programação JavaScript**. 1ª Ed. São Paulo: Alta Books, 2016.

Operadores

Vejamos a expressão simples $2 + 3$ é igual a 5. Aqui 2 e 3 são chamados de **operandos** e '+' é chamado de **operador**. JavaScript oferece suporte aos seguintes tipos de operadores.

- Operadores aritméticos
- Operadores de comparação
- Operadores lógicos (ou relacionais)
- Operadores de atribuição
- Operadores condicionais (ou ternários).

Operadores aritméticos

JavaScript suporta os seguintes operadores aritméticos:

Operador	Descrição	Exemplo
+	adição	$2 + 3$ Resulta em 5
-	subtração	$5 - 3$ Resulta em 2
*	multiplicação	$2 * 3$ Resulta em 6
/	divisão	$10 / 2$ Resulta em 5
%	módulo (resto da divisão)	$5 \% 2$ Resulta em 1, o resto da divisão de 5 por 2
++	incremento	$++5$ Resulta em 6

--	decremento	-5 Resulta em 4
----	------------	--------------------

Operadores de comparação

Operador	Descrição	Exemplo
<code>==</code>	igual a	<code>2 == 2</code> Resulta em true (verdadeiro) <code>1 == 2</code> Resulta em false (falso)
<code>!=</code>	diferente de	<code>2 != 1</code> Resulta em true (verdadeiro) <code>2 != 2</code> Resulta em false (falso)
<code>></code>	maior que	<code>3 > 2</code> Resulta em true (verdadeiro)
<code><</code>	menor que	<code>2 < 3</code> Resulta em true (verdadeiro)
<code>>=</code>	maior ou igual a	<code>3 >= 3</code> Resulta em true (verdadeiro)
<code><=</code>	menor ou igual a	<code>2 <= 3</code> Resulta em true (verdadeiro)

Operadores lógicos

Utilizados na criação de expressões lógicas. JavaScript suporta os seguintes operadores lógicos:

Operador	Descrição	Exemplo
<code>&&</code>	E	<code>(a > b) && (a > c)</code> Resulta true se ambos os teste forem verdadeiros
<code> </code>	OU	<code>(a > b) (a > c)</code> Resulta true se um dos testes for verdadeiro.
<code>!</code>	NÃO (negação)	<code>!(3 == 3)</code> Resulta em false, pois está negando a expressão que é verdadeira, invertendo o valor.

Operador `typeof`

O operador `typeof` é um operador unário colocado antes de seu único operando, que pode ser de qualquer tipo. Seu valor é uma string que indica o tipo de dados do operando.

O operador `typeof` é avaliado como "número", "string" ou "booleano" se seu operando for um número, string ou valor booleano e retorna verdadeiro ou falso com base na avaliação.

Aqui está uma lista dos valores de retorno para o operador `typeof`:

Tipo	Resultado
Undefined	"undefined"
Null	"object"
Boolean	"boolean"
Number	"number"
String	"string"
Host object (provided by the JS environment)	Implementation-dependent
Function object (implements <code>[[Call]]</code> in ECMA-262 terms)	"function"
E4X XML object	"xml"
E4X XMLList object	"xml"
Qualquer outro objeto	"object"

Objetos em JavaScript

JavaScript é uma linguagem de Programação Orientada a Objetos (POO). Uma linguagem de programação pode ser chamada de orientada a objetos se fornecer quatro recursos básicos para os desenvolvedores:

- **Encapsulamento** - a capacidade de armazenar informações relacionadas, sejam dados ou métodos, juntas em um objeto.
- **Agregação** - a capacidade de armazenar um objeto dentro de outro objeto.
- **Herança** - a capacidade de uma classe confiar em outra classe (ou número de classes) para algumas de suas propriedades e métodos.
- **Polimorfismo** - a capacidade de escrever uma função ou método que funciona de várias maneiras diferentes.

Os objetos são compostos de atributos. Se um atributo contém uma função, é considerado um método do objeto, caso contrário, o atributo é considerado uma propriedade.

Propriedades do objeto

As propriedades do objeto podem ser qualquer um dos três tipos de dados primitivos ou qualquer um dos tipos de dados abstratos, como outro objeto. As propriedades do objeto geralmente são variáveis usadas internamente nos métodos do objeto, mas também podem ser variáveis globalmente visíveis que são usadas em toda a página.

A sintaxe para adicionar uma propriedade a um objeto é:

```
nomedoObjeto.propriedadedoObjeto = valordapropriedade;
```

Exemplo:

```
var str = documento.titulo;
```

Métodos de Objeto

Métodos são as funções que permitem que o objeto faça algo ou que algo seja feito a ele. Há uma pequena diferença entre uma função e um método – uma função é uma unidade autônoma de instruções e um método é anexado a um objeto e pode ser referenciado pela palavra-chave **this**.

Os métodos são úteis para tudo, desde a exibição do conteúdo do objeto na tela até a execução de operações matemáticas complexas em um grupo de propriedades e parâmetros locais.

Exemplo: a seguir está um exemplo simples para mostrar como usar o método **write()** do objeto **document** para escrever qualquer conteúdo no documento.

```
document.write("Teste de escrita");
```

Objetos definidos pelo usuário

Todos os objetos definidos pelo usuário e objetos embutidos são descendentes de um objeto chamado **Object**.

O operador new

O operador **new** é usado para criar uma instância de um objeto. Para criar um objeto, o operador **new** é seguido pelo método construtor. No exemplo a seguir, os métodos do construtor são **Object()**, **Array()** e **Date()**. Esses construtores são funções JavaScript integradas:

```
1  var funcionario = new Object();
2  var livros = new Array("C++", "Perl", "Java");
3  var dia = new Date("15 de novembro, 2021");
4
```

O Construtor Object()

Um construtor é uma função que cria e inicializa um objeto. JavaScript fornece uma função construtora especial chamada **Object()** para construir o objeto. O valor de retorno do construtor **Object()** é atribuído a uma variável.

A variável contém uma referência ao novo objeto. As propriedades atribuídas ao objeto não são variáveis e não são definidas com a palavra-chave **var**.

```
1  var livro = new Object(); // Cria o objeto
2  livro.assunto = "Perl"; // Atribui as propriedades do objeto
3  livro.autor = "Moran";
4
```


Exercícios

1. Sobre os operadores em JavaScript, assinale o item correto:
 - a) O símbolo '+' é um operador de atribuição.
 - b) São divididos em aritméticos, de string, de atribuição e de comparação.
 - c) O operador ternário é um operador de atribuição, retornando sempre um valor numérico.
 - d) Operadores lógicos retornam os valores verdadeiro ou falso.
 - e) A expressão $5 - 2$ resulta em false.

2. Observe as expressões a seguir e assinale o item correto:
 - a) $10 \% 2$ retorna 1
 - b) $10 \% 5$ retorna 2
 - c) $++5 + 6$ retorna 11
 - d) $5 / 2$ retorna 2.5
 - e) $A = 2; A / 2$ retorna 1

3. Sobre POO em JavaScript, assinale o item correto:
 - a) Não possui a propriedade de encapsulamento.
 - b) A agregação é a capacidade de uma classe armazenar informações relacionadas, dados ou métodos, em um objeto.
 - c) Os objetos possuem somente atributos e métodos.
 - d) `salariofunc.valor = 2000` atribui o valor 2000 a propriedade `salariofunc` do objeto `valor`.
 - e) O comando `var funcionário = new method()` cria um novo objeto funcionário.

4. Qual das seguintes funções do objeto String retorna o valor primitivo do objeto especificado? Assinale o item correto:
 - a) `toLocaleUpperCase ()`
 - b) `toUpperCase ()`
 - c) `toString ()`
 - d) `valueOf ()`
 - e) `typeOf()`

5. Qual das afirmações a seguir é verdadeira sobre o operador `typeof` em JavaScript?
 - a) O `typeof` é um operador unário que é colocado antes de seu único operando, que pode ser de qualquer tipo.
 - b) O `typeof` retorna um valor booleano.
 - c) Seu valor é uma string que indica o tipo de dado do operando.
 - d) As alternativas A e C estão corretas.
 - e) Nenhuma das opções anteriores.

6. Qual dos seguintes tipos de variável está visível em todo o seu código JavaScript?

- a)** Variável global.
- b)** Variável local.
- c)** Em objetos.
- d)** Em uma classe.
- e)** Nenhuma das opções anteriores.

Gabarito

1. D

A letra D é a alternativa correta, pois os operadores lógicos tem como retorno os valores verdadeiro ou falso.

2. E

A letra E é a alternativa correta, pois o valor da variável A é 2 e 2 dividido por 2 resulta em 1.

3. B

A letra B é a alternativa correta, pois a agregação é um tipo especial de associação onde tenta-se demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte); conhecemos como todo/parte.

4. D

A letra D é a alternativa correta, pois JavaScript chama o método `valueOf` para converter um objeto em um valor primitivo. Se um objeto não tem um valor primitivo, `valueOf` retorna o próprio objeto. Você pode usar `valueOf` dentro do seu próprio código para converter um objeto embutido, em um valor primitivo.

5. D

A letra D é a alternativa correta, pois o `typeof` é um operador unário colocado antes de seu operando, o qual pode ser de qualquer tipo. Seu valor é uma string que especifica o tipo do operando.

6. A

A letra A é a alternativa correta, pois variável global é uma variável acessível em todos os escopos de um programa de computador.