

---

# JavaScript

---

referências à linguagem

Livro escrito no [Wikibooks](#) em língua portuguesa, livre pensar e aprender.

Esta obra é o resultado da experiência de várias pessoas, que acreditam que a melhor forma do conhecimento é o conhecimento compartilhado. Aqui temos uma pequena referência a estas pessoas:

- [Sérgio Eduardo Rodrigues](#)
- [Lightningpirit](#)

Este livro está para download livre e gratuito neste endereço:

- <http://pt.wikibooks.org/wiki/Javascript>

A versão on-line e em contínuo desenvolvimento está nesse endereço:

- <http://pt.wikibooks.org/wiki/Javascript>
- A versão para este arquivo PDF foi modificada pela última vez em 30 de Julho de 2006, modificações posteriores a esta data no site do Wikibooks não estão inclusas.
- Este arquivo PDF foi feito através do software [OpenOffice.org versão 2.0.3](#).

## ÍNDICE:

• <a href="#">Introdução</a> .....	4
• <a href="#">Tipos de dados</a> .....	5
• <a href="#">Conversão de tipos</a> .....	7
• <a href="#">Variáveis</a> .....	8
• <a href="#">Definição</a> .....	8
• <a href="#">Escopo da variável</a> .....	9
• <a href="#">Constantes</a> .....	10
• <a href="#">Matrizes</a> .....	11
• <a href="#">Numéricas</a> .....	12
• <a href="#">Strings</a> .....	13
• <a href="#">Operadores</a> .....	14
• <a href="#">Fluxo de controle</a> .....	17
• <a href="#">Bloco de comandos</a> .....	18
• <a href="#">If/else</a> .....	18
• <a href="#">While/do..while</a> .....	19
• <a href="#">Switch/case</a> .....	19
• <a href="#">For</a> .....	20
• <a href="#">Label</a> .....	20
• <a href="#">Continue</a> .....	20
• <a href="#">Break</a> .....	21
• <a href="#">Throw</a> .....	21
• <a href="#">Try/Catch/Finally</a> .....	21
• <a href="#">Funções</a> .....	23
• <a href="#">Objetos</a> .....	24
• <a href="#">Definição</a> .....	25
• <a href="#">Propriedades</a> .....	26
• <a href="#">Objetos predefinidos</a> .....	26
• <a href="#">Conclusão</a> .....	42
• <a href="#">Referências na Internet</a> .....	43

## INTRODUÇÃO

Muito se fala sobre Javascript, porém existe muito pouca documentação em português desta linguagem de programação.

O maior problema é a confusão que se faz entre a linguagem Java e o Javascript com relação a semelhança no nome, na mesma linha de raciocínio fazem a relação de C++ com Java pela semelhança de comandos e sintaxes, embora a implementação de algumas sintaxes e construções sejam parecidas são linguagens distintas, criadas para resolver problemas diferentes, e que por isto possuem capacidades diferentes, o mesmo ocorre com Java e Javascript.

Enquanto a Linguagem Java é fortemente tipada e possui tipos estáticos o Javascript oferece uma reduzida quantidade de tipos, isto é, o necessário para criação de telas dinâmicas e certa lógica as páginas html.

Javascript é pequena, leve, portátil (está presente em quase todos os navegadores e em todas as plataformas que estes navegadores rodam). Ela não é uma linguagem para rodar sozinha, precisando de um navegador para tal.

Javascript segue uma linha de quanto menor melhor, ou seja, ela é pequena na sua escrita, e criação de objetos. Os tipos de variáveis são dinâmicos, e possui objetos globais já predefinidos conforme o ambiente que se encontra.

Inicialmente ela foi criada pela Netscape para ser a linguagem padrão no navegador, para produzir certa verificação aos dados. Porém com o passar do tempo, ela foi padronizada pela ECMA(European Computer Manufactures Association) (<http://www.ecma-international.org>) vide ECMA-262, também reconhecida pela ISO ISO-16262.

Porém esta padronização não avançou sobre os objetos públicos e seus respectivos métodos, o que torna difícil a padronização de programas entre navegadores.

Aqui será abordada a especificação JavaScript 1.5 que segue a ecma262 edição 3.

## TIPOS DE DADOS

O Javascript possui poucos tipos de dados, sendo eles:

- **Numéricos:**

Este tipo de dado armazena valores, tanto valores inteiros como ponto flutuante, por exemplo:

- 1
- 84
- 2e10
- 3.141516
- 0.000001

Os valores numericos podem fazer parte de operações aritmética, como a soma, subtração, multiplicação e divisão.

Valores numéricos também podem fazer parte de operações aritméticas de bits. Como por exemplo ( $\gg$ ) rotação de bits para direita, ( $\ll$ ) rotação de bits para esquerda, ( $\ggg$ ) rotação de bits a direita sem levar em conta o sinal, ( $\wedge$ ) OU exclusivo (= XOR), ( $\&$ ) operação E binária (= AND), ( $\mid$ ) operação OU binária (= OR), ( $\sim$ ) Negação binária (= NOT).

Existem várias funções para manipulação de valores numéricos, como funções trigonométricas, funções de arredondamento e exponenciação, funções de transformação de tipos, etc.

Existem alguns valores numéricos especiais, são eles :

- NaN - Na verdade NaN é a abreviação de (*Not a Number*) = (Não um Número), ele é o resultado geralmente de operações inválidas com números. Como por exemplo, o resultado da operação (0/0), irá resultar no NaN. Ele também é uma constante, que pode ser atribuída a uma variável, como veremos mais adiante.
- Infinity - Representa um valor infinito, podendo ser tanto positivo quanto negativo. Todas as operações com valores infinitos resultarão num valor infinito, exceto divisão e subtração que resultará no NaN.

- **Lógicos:**

Os valores lógicos podem assumir dois valores, *true* (verdadeiro) e *false* (falso).

- **Strings:**

São cadeias de caracteres, o maior número que uma string pode conter depende do navegador em uso.

Valores strings são valores delimitados por apóstrofo(') ou por aspas(""), ex: "texto" ou 'texto'  
uma barra inversa permite a inserção de caracteres especiais, ex:

"\b" - Representa o backspace (caracter ascii 8)  
"\t" - Tabulação (caracter ascii 9)  
"\r" - Retorno de carro (caracter ascii 13)  
"\n" - Nova linha (caracter ascii 10)  
"\v" - Tabulação vertical (caracter ascii 11)  
"\uNNNN" - Caracter unicode (NNNN representa um valor hexadecimal de 0000 a FFFF)  
"\xNN" - Caracter ascii (NN representa um valor hexadecimal de 00 a FF)  
"\'" - Apóstrofo  
"\"" - Aspas  
"\" - Barra inversa

- **Null:**

O null é um valor especial, representa um objeto nulo, não deve ser confundido com uma variável não inicializada, pois o valor null existe.

Portanto uma variável com o conteúdo null existe em memória, referenciando este objeto especial.

- **Undefined:**

O valor undefined significa que a variável não foi instanciada, inicialmente todas as variáveis se encontram neste estado.

## CONVERSÃO DE TIPOS

- **Atribuindo valores**

Diferente da maioria das linguagens o javascript define as variáveis dinamicamente, portanto ao atribuir uma variável ele escolhe o tipo conforme o valor passado para a variável, não sendo necessário especificar o mesmo.

```
var numero = 1;
var texto = "Sérgio Eduardo Rodrigues";
var valor = 123.45;
var ativo= true;
var nascimento = new Date(1969,1,4)
```

Nome	Tipo
<b>numero</b>	numerica inteira
<b>texto</b>	string
<b>valor</b>	numerica com ponto flutuante
<b>ativo</b>	booleana
<b>nascimento</b>	objeto date

- **Convertendo**

Uma variável pode ser atribuída para outro tipo, ou utilizando uma função de conversão ou então fazendo operações aritméticas.

Como por exemplo, quando atribuímos ao **numero** o valor 1, ele se encontra no tipo numérico inteiro, se o dividirmos por 2 ele irá para o tipo numérico ponto flutuante:

```
numero = 1; // inteiro 1
numero = numero / 2; // Vai para flutuante 0.5
numero = " " + numero; // ele é convertido para string, pois está sendo somado
à outra string
numero = parseFloat(numero); // Ele irá resultar no número 0.5
numero = parseInt(numero); // Vai para o inteiro 0
```

## VARIÁVEIS: DEFINIÇÃO:

As variáveis são representadas por nomes chamados de identificadores, estes identificadores tem certa regra para ser montado:

- 1º Deve iniciar obrigatoriamente por letra ou pelo símbolo “\_” ou “\$”
- 2º A partir daí além de letras, “\_” e \$ pode conter dígitos(0 até 9).

você pode declarar uma variável de duas formas:

- 1ª Atribuindo diretamente a variável,

exemplo

```
nome="Sérgio";
```

- 2ª Utilizando a palavra reservada “var”,

exemplo

```
var nome = “Sérgio Eduardo rodrigues”;
```

Uma variável ou matriz que não tenha sido inicializada, possui o valor de “undefined”, observe que o Javascript é sensível para *case-sensitive*, ou seja, **letras minúsculas e maiúsculas são diferentes**, portanto, undefined e null devem ser escritos sempre em letra minúsculas.

Se uma variável é declarada apenas, com o comando var, o seu conteúdo é “undefined”, ou NaN(*Not a Number*), caso esteja num contexto numérico.

Exemplo:

```
var x;  
x = x * 2;
```

o resultado será NaN.

ou se for utilizado

```
x = x + “teste”
```

causará um erro de execução, pois x não tem valor definido.



## VARIÁVEIS: ESCOPO:

Caso a variável seja declarada fora do corpo de uma função ela será considerada como pública, ou seja poderá ser alcançada por todas as funções, caso ela seja declarada dentro de uma função ela é considerada privada, pois somente pode ser vista pelo código da função.

Exemplo 1(variável pública):

```
var x=10;
function fx(){
  ... será possível utilizar o valor de x ...
}function fy() {
  ... será possível utilizar o valor de x ...
}
```

Exemplo 2 (variável privada):

```
function fx() {
  var x = 5;
  ... será possível utilizar o valor de x ...
}
function fy() {
  ... x terá valor undefined, ou seja não será visto por fy ...
}
```

## VARIÁVEIS: CONSTANTES:

São variáveis declaradas com a palavra chave “const”, que não podem sofrer alteração de seu conteúdo e nem de sua declaração no escopo da rotina. Exemplo:

```
const fator = 1.34;  
const nome = "Sérgio"
```

se tentar efetuar uma redeclaração ocorrerá um erro de execução, exemplo:

```
const fator = 1.34;  
var fator = 22;
```

ou então se este já tiver sido declarado

```
function funcao() {}  
const funcao="teste";
```

## Gracias por visitar este Libro Electrónico

Puedes leer la versión completa de este libro electrónico en diferentes formatos:

- HTML(Gratis / Disponible a todos los usuarios)
- PDF / TXT(Disponible a miembros V.I.P. Los miembros con una membresía básica pueden acceder hasta 5 libros electrónicos en formato PDF/TXT durante el mes.)
- Epub y Mobipocket (Exclusivos para miembros V.I.P.)

Para descargar este libro completo, tan solo seleccione el formato deseado, abajo:

