

TUTORIAIS JAVASCRIPT

O que é a linguagem JavaScript

Copyright 2013 – Todos os Direitos Reservados
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

O que é a linguagem JavaScript

Introdução

JavaScript é a linguagem de script para desenvolvimento de páginas Web. Essa linguagem é usada em bilhões de páginas Web para adicionar funcionalidade, validar formulários, comunicação com servidores, entre outras coisas. Além de ser muito fácil de aprender, ela é a linguagem mais popular do mundo para programação na Web. Podemos afirmar que JavaScript é uma sequência de declarações a serem executadas pelo browser na sequência em que eles foram escritos. Essas declarações são comandos para o browser, ou seja, elas dizem ao browser o que fazer.

Um pouco de história

JavaScript foi originalmente desenvolvida por Brendan Eich da Netscape sob o nome de Mocha, posteriormente teve seu nome mudado para LiveScript e por fim JavaScript. LiveScript foi o nome oficial da linguagem quando foi lançada pela primeira vez na versão beta do navegador Netscape 2.0 em setembro de 1995, mas teve seu nome mudado em um anúncio conjunto com a Sun Microsystems em dezembro de 1995 quando foi implementado no navegador Netscape versão 2.0B3.

A mudança de nome de LiveScript para JavaScript coincidiu com a época em que a Netscape adicionou suporte à tecnologia Java em seu navegador (Applets). A escolha final do nome causou confusão dando a impressão de que a linguagem foi baseada em java, sendo que tal escolha foi caracterizada por muitos como uma estratégia de marketing da Netscape para aproveitar a popularidade do recém-lançado Java.

JavaScript rapidamente adquiriu ampla aceitação como linguagem de script client-side de páginas web. Como consequência, a Microsoft desenvolveu um dialeto compatível com a linguagem de nome JScript para evitar problemas de trademark. JScript adicionou novos métodos para consertar métodos do Javascript relacionados a data que apresentavam problemas. JScript foi incluído no Internet Explorer 3.0, liberado em Agosto de 1996. Javascript e Jscript são tão similares que os dois termos são comumente usados de forma intercambiável. A Microsoft entretanto declara muitas características nas quais JScript não conforma com a especificação ECMA. Em novembro de 1996 a Netscape anunciou que tinha submetido JavaScript para Ecma internacional como candidato a padrão industrial e o trabalho subsequente resultou na versão padronizada chamada ECMAScript.

JavaScript tem se transformado na linguagem de programação mais popular da web. Inicialmente, no entanto, muitos profissionais denegriam a linguagem pois a mesma tinha como alvo principal o público alvo leigo. Com o advento do Ajax, JavaScript teve sua popularidade de volta e recebeu mais atenção profissional. O resultado foi a proliferação de frameworks e bibliotecas, práticas de programação melhoradas e o aumento no uso do JavaScript fora do ambiente de navegadores bem como o uso de plataformas de JavaScript server-side.

Em janeiro de 2009 o projeto CommonJS foi fundado com o objetivo de especificar uma biblioteca padrão para desenvolvimento JavaScript fora do navegador.

O que você pode fazer com JavaScript

No final de 1990, com a introdução do HTML dinâmico (DHTML), JavaScript também foi usado para fornecer uma experiência de usuário mais dinâmica através de menus drop-down e similares.

A popularidade do JavaScript tem crescido bastante recentemente porque ele é um componente chave no Ajax (Asynchronous JavaScript and XML), que promete reestruturar a forma como as aplicações web devem interagir com os usuários. Com o tempo, muitos problemas de plataforma cruzada foram resolvidos, e a linguagem se tornou mais sofisticada, tanto que JavaScript não é mais apenas uma linguagem de script, é uma linguagem de programação completa.

Então, o que você pode fazer com JavaScript? Vejamos algumas das tarefas que você poderá fazer com essa linguagem:

1) Validar os campos do formulário

Validar entrada de formulário antes de enviar o conteúdo para o servidor. Isso economiza tempo e recursos de servidor e fornece um feedback imediato.

2) Definir e recuperar cookies web

Manter as informações como nomes de usuários, números de contas ou preferências em um ambiente controlado e seguro, facilitando aos usuários da próxima vez que acessar um site.

3) Alterar dinamicamente a aparência de um elemento da página

Fornecer feedback, destacando entradas de forma incorreta, aumentar o tamanho da fonte de uma seção com base no pedido do leitor, entre outros.

4) Ocultar e mostrar elementos

Com base na preferência pessoal ou ações do usuário, mostrar ou ocultar o conteúdo da página, como elementos de formulário, escrita em expansão, e alterar o tamanho de uma imagem exibida.

5) Mover elementos sobre a página

Criar um menu drop-down, ou fornecer um cursor animado para destacar os elementos da página.

6) Capturar eventos do usuário e ajustar a página de acordo

Com base em ações de teclado ou mouse, fazer uma seção da página editável.

7) Rolar conteúdo

Para imagens maiores ou áreas de conteúdo, fornecer uma maneira de pegar o elemento com um mouse ou teclado, e deslocá-lo para a direita ou esquerda, para cima ou para baixo.

8) Interface com uma aplicação de servidor sem sair da página

Esta é a base do Ajax e é usado para preencher as listas de seleção, os dados de atualização e atualizar toda uma lista sem ter que recarregar a página. Isso ajuda a eliminar a ida e volta para o servidor, que pode ser caro em termos de tempo e recursos.

Onde inserir seus códigos JavaScript

Os códigos JavaScript devem ser inseridos dentro das tags **<script>** e **</script>** de uma página HTML. E esse código pode ser colocado tanto dentro da tag **<body>** quanto da tag **<head>**, e pode ser executado pela maioria dos browsers. As tags **<script>** e **</script>** informam onde o JavaScript começa e termina.

Veja a seguir um exemplo de código JavaScript inserido em uma página HTML:

Neste exemplo o código foi inserido dentro das tags **<body>** e **</body>**, e será executado quando a página for carregada:

```
<!DOCTYPE html>
<html>
<body>
.
.
<script>
document.write("<h1>Isto é um cabeçalho</h1>");
document.write("<p>Isto é um parágrafo</p>");
</script>
.
.
</body>
</html>
```

Você pode colocar um número ilimitado de scripts dentro de um documento HTML.

Ponto e Vírgula

Em JavaScript, assim como em algumas outras linguagens, o ponto-e-vírgula separa comandos ou declarações. Geralmente são adicionados no final de cada declaração. Usando esse recurso é possível escrever muitas declarações em uma única linha. Dessa forma economiza-se mais espaço. Você deve ter observado em alguns códigos o não uso do ponto-e-vírgula, é porque o uso do ponto-e-vírgula no JavaScript é opcional, mas é uma boa prática utilizá-lo, seu código fica mais organizado, e com a garantia que vai funcionar.

Caso sensitivo

JavaScript é caso sensitivo, ou seja, ela diferencia caracteres maiúsculos de minúsculos. Por exemplo, a variável:

- **Ana** é diferente de **ana** que é diferente de **ANA**.

Portanto, cuidado quando utilizar suas variáveis e comandos.

Espaços em branco

JavaScript ignora espaços em branco. Você pode adicionar quanto espaço você quiser, não vai fazer a menor diferença. Por exemplo:

```
var nome="Maria";
```

É o mesmo que:

```
var nome = "Maria";
```

Use espaços em branco o suficiente para organizar o seu código e ficar mais fácil de ser lido por outras pessoas, se for o caso.

Blocos em JavaScript

As declarações em JavaScript devem ser agrupadas em blocos. Os blocos começam com uma chave esquerda ({) e terminam com uma chave direita (}). Um bom exemplo de declarações agrupadas em blocos são as funções. Veja o exemplo a seguir:

```
function minhaFuncao()
{
document.getElementById("demo").innerHTML="Olá Jorge";
document.getElementById("myDIV").innerHTML="Como passou o Natal?";
}
```

Quebra de linha de código em JavaScript

Você pode quebrar uma linha de código dentro de uma string com uma barra invertida (\), conforme mostra o exemplo a seguir:

```
document.write("Olá \
Mundo!");
```

Entretanto, você não pode quebrar uma linha de código do JavaScript, conforme mostra o exemplo a seguir:

```
document.write \
("Olá Mundo!");
```

Onde posso escrever os meus códigos em JavaScript

Você pode escrever os seus códigos em qualquer editor de textos como o Bloco de Notas do Windows, o NotePad++ e mais uma infinidade de editores existentes no mercado. Entretanto, para códigos muito extensos dentro de páginas HTML que fazem parte de um site, você deve optar por uma ferramenta mais robusta do tipo Dreamweaver ou semelhante. A grande vantagem dessas ferramentas é que elas ajudam com o código, como por exemplo, codificar com cores diferentes os diversos tipos de comandos como variáveis, funções, declarações, etc. Além disso, corrigem o seu código quando o mesmo for executado e facilitam a digitação.

Funções em JavaScript

As funções criadas em JavaScript são geralmente inseridas entre as tags **<head>** e **</head>**, ou no rodapé da página. Dessa forma elas ficarão em um único lugar e não deverão interferir no conteúdo da página, embora elas possam ser inseridas nas tags **<body>** e **</body>**. Mas essa prática não é considerada ideal.

Vejamos alguns exemplos utilizando essas duas situações:

Exemplo 1

Nesse exemplo vamos utilizar uma função e inseri-la entre as tags **<head>** e **</head>** de uma página HTML.

Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js1-exemplo1.html**.

```

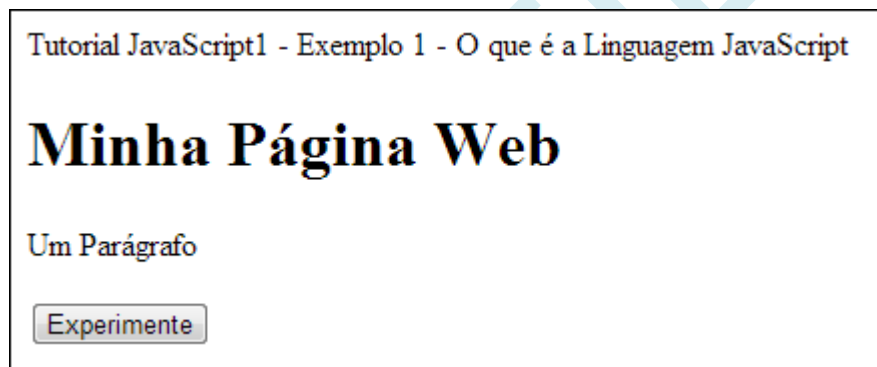
<!DOCTYPE html>
<html>
Tutorial JavaScript1 - Exemplo 1 - O que é a Linguagem JavaScript
<head>
  <title>Tutorial JavaScript1 - Exemplo 1 - O que é a Linguagem JavaS-
cript</title><p>
<script>
function minhaFuncao()
{
document.getElementById("demo").innerHTML="Minha Primeira Função JavaScript ";
}
</script>
</head>

<body>
<h1>Minha Página Web</h1>
<p id="demo">Um Parágrafo</p>
<button type="button" onclick="minhaFuncao()">Experimente</button>

</body>
</html>

```

2. Execute no seu browser preferido e confira o resultado com o mostrado abaixo.



Exemplo 2

Nesse exemplo vamos utilizar um função e inserí-la entre as tags **<body>** e **</body>** de uma página HTML.

Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js1-exemplo2.html**.

```

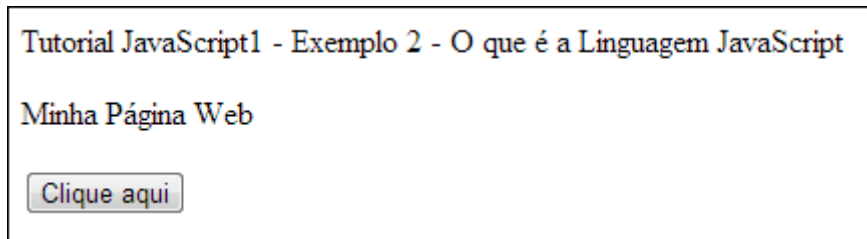
<!DOCTYPE html>
<html>
Tutorial JavaScript1 - Exemplo 2 - O que é a Linguagem JavaScript
<head>
  <title>Tutorial JavaScript1 - Exemplo 2 - O que é a Linguagem JavaS-
cript</title><p>
</head>

<body>
<p id="demo">Minha Página Web</p>
<button type="button" onclick="minhaFuncao()">Clique aqui</button>
<script>
function minhaFuncao()

```

```
{
document.getElementById("demo").innerHTML="Minha primeira Função JavaScript ";
}
</script>
</body>
</html>
```

2. Execute no seu browser preferido e confira o resultado com o mostrado abaixo.



Scripts JavaScript externos

Os scripts JavaScript também podem estar localizados em arquivos externos. Esses arquivos geralmente contêm códigos que podem ser utilizados por páginas Web. Esse tipo de arquivo possui a extensão **.js**.

Para usar um arquivo externo JavaScript na sua página você deve inserir o arquivo com a extensão **.js** no atributo "**src**" da tag **<script>** conforme mostra o exemplo a seguir:

```
<!DOCTYPE html>
<html>
<body>
<script src="minhasFuncoes.js"></script>
</body>
</html>
```

OBS.: Arquivos de script externos não poderão conter tags **<script>**.

Exercícios de fixação

- 1) Os códigos em JavaScript só podem ser escritos em ferramentas do tipo Dreamweaver.
 - a) Verdadeiro
 - b) Falso
- 2) JavaScript é uma linguagem de:
 - a) Programação compilável independente de browser.
 - b) Script para criação de páginas web.
 - c) Script que funciona dentro de códigos CSS.
 - d) Marcação semelhante a XML.
- 3) Os códigos JavaScript são inseridos dentro de páginas HTML entre as tags:
 - a) <body> e </body>
 - b) <script> e </script>
 - c) <javascript> e </java>
 - d) <scriptjava> e </scriptjava>
- 4) Os arquivos externos em JavaScript possuem a extensão:
 - a) .java
 - b) .jscript
 - c) .js
 - d) .jsc
 - e) .javascript
- 5) Deixar espaços em branco entre as declarações causa erro de execução.
 - a) Verdadeiro
 - b) Falso
- 6) JavaScript e Java são a mesma linguagem.
 - a) Verdadeiro
 - b) Falso
- 7) Como JavaScript é caso sensetivo, então:
 - a) A variável **nome** é igual à variável **NOME**.
 - b) A variável **idade** é diferente de **Idade** mas é igual à **IDADE**.
 - c) Se uma varável **x=2**, e uma variável **X=-x**, então a soma dessas variáveis seria igual a 2.
 - d) Se uma varável **nome="Jorge "** e **Nome="Silva"**, então **nome + Nome** seria igual a "Jorge Silva".
- 8) Toda função em JavaScript começa com _____ e termina com _____:
 - a) [e]
 - b) { e }
 - c) (e)
 - d) // e \\
- 9) Toda declaração em JavaScript deve terminar com:
 - a) Um ponto

- b) Uma vírgula
- c) Um ponto e vírgula
- d) Dois pontos

10) Se quisermos carregar um arquivo externo JavaScript para uma página web, que atributo devemos utilizar?

- a) load
- b) get
- c) type
- d) src

JORGE EIDER