

ESCOLA NOVAERENSE



MARCO AURÉLIO MONTEIRO LIMA

LINGUAGENS DE PROGRAMAÇÃO WEB
LINGUAGENS DE PROGRAMAÇÃO II

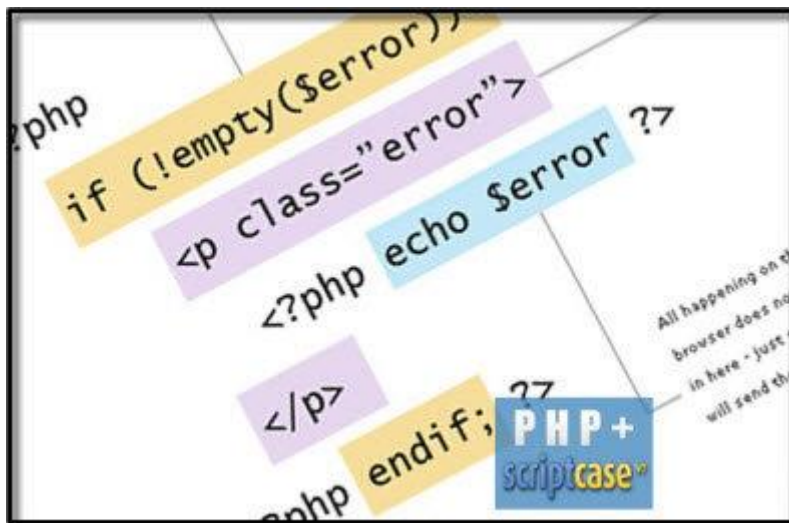
NOVA ERA
2015

. Linguagens de Programação WEB

São linguagens de programação específicas para o desenvolvimento de sites e aplicações que rodem na internet, atualmente o mercado conta com diversas opções de linguagens WEB, como por exemplo:

- PHP
- C# .NET
- VB. NET
- Java

Cada linguagem possui suas próprias características e peculiaridades possuindo formas diferentes de se realizar determinadas ações, a forma como um problema pode ser resolvido em uma linguagem pode ser completamente diferente em outra linguagem. Por isso, estudar e entender bem como cada uma delas funciona faz com que o profissional consiga extrair ao máximo o potencial da linguagem que escolher para trabalhar.



. JavaScript (HISTÓRIA)

JavaScript foi originalmente desenvolvido 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 marca registrada. 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 ela tinha como alvo principal o público 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.⁴

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

JavaScript (CARACTERÍSTICAS)

As seguintes características são comuns a todas as implementações conformantes com o ECMAScript.

. Imperativa e Estruturada

JavaScript suporta os elementos de sintaxe de programação estruturada da linguagem C (por exemplo, if, while, switch). Uma exceção é a questão do escopo: o escopo em blocos ao estilo do C não é suportado, em seu lugar JavaScript utiliza escopo a nível de função. JavaScript 1.7, entretanto, suporta escopo a nível de bloco através do comando let. Como C, JavaScript faz distinção entre expressões e comandos. Uma diferença sintática do C é que a quebra de linha termina automaticamente o comando, sendo o ponto-e-vírgula opcional ao fim do comando.

. Dinâmica

- Tipagem dinâmica

Como na maioria das linguagens de script, tipos são associados com valores, não com variáveis. Por exemplo, a variável x poderia ser associada a um número e mais tarde

associada a uma string. JavaScript suporta várias formas de testar o tipo de um objeto, incluindo duck typing.

- Baseada em objetos

JavaScript é quase inteiramente baseada em objetos. Objetos JavaScript são arrays associativos, aumentados com protótipos. Os nomes da propriedade de um objeto são strings: `obj.x = 10` e `obj["x"] = 10` são equivalentes, o ponto neste exemplo é apenas sintático. Propriedades e seus valores podem ser adicionadas, mudadas, ou deletadas em tempo de execução. A maioria das propriedades de um objeto (e aqueles em sua cadeia de herança via protótipo) pode ser enumerada usando-se uma estrutura de repetição `for...in`. Javascript possui um pequeno número de objetos padrão da linguagem como *window* e *document*.

- Avaliação em tempo de execução

JavaScript inclui a função `eval` que consegue executar em tempo de execução comandos da linguagem que estejam escritos em uma string.

. Funcional

- Funções de primeira classe

No JavaScript as funções são de primeira classe, isto é, são objetos que possuem propriedades e métodos, e podem ser passados como argumentos, serem atribuídos a variáveis ou retornados como qualquer outro objeto.

- Funções aninhadas

Funções 'internas' ou 'aninhadas' são funções definidas dentro de outras funções. São criadas cada vez que a função que as contém (externa) é invocada. Além disso, o escopo da função externa, incluindo constantes, variáveis locais e valores de argumento, se transforma parte do estado interno de cada objeto criado a partir da função interna, mesmo depois que a execução da função interna é concluída.

- Fechamentos

JavaScript permite que funções aninhadas sejam criadas com o escopo léxico no momento de sua definição e possui o operador `()` para invocá-las em outro momento. Essa combinação de código que pode ser executado fora do escopo no qual foi definido, com seu próprio escopo durante a execução, é denominada, dentro da ciência da computação, *fechamento*.

. Baseada em Protótipos

- Protótipos

JavaScript usa protótipos em vez de classes para o mecanismo herança. É possível simular muitas características de orientação a objetos baseada em classes com protótipos.

- Funções e métodos

Diferente de muitas linguagens orientadas a objetos, não há distinção entre a definição de uma função e a definição de um método no JavaScript. A distinção ocorre durante a chamada da função; a função pode ser chamada como um método. Quando uma função é chamada como método de um objeto, a keyword `this` da função é associada àquele objeto via tal invocação.

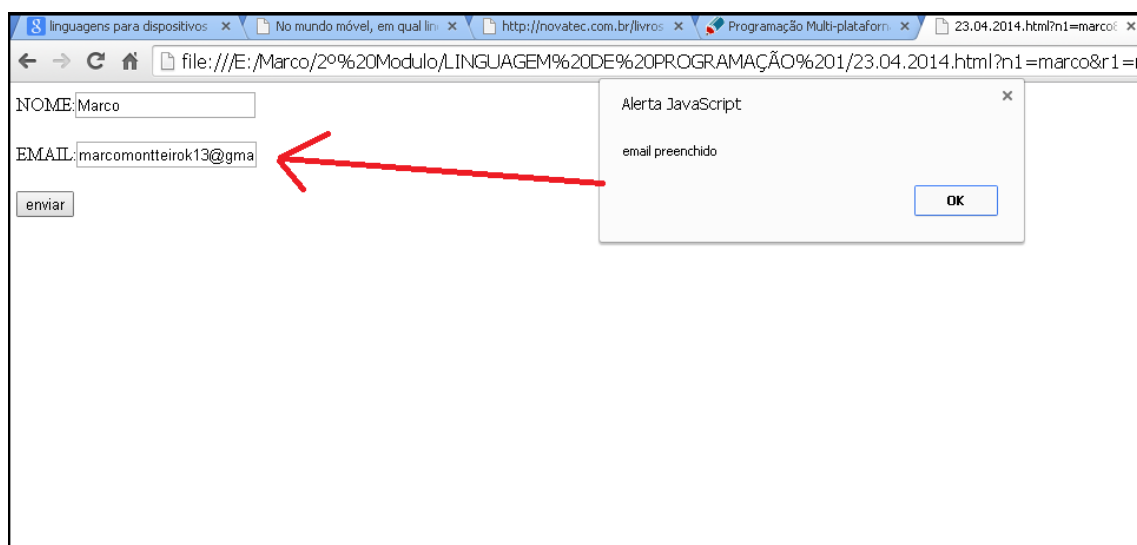
O código abaixo verifica a caixa de texto relacionada ao e-mail ao clicar no botão enviar:

```
1 <html>
2   <script>
3       function c()
4       {
5           if (document.email.r1.value=="")
6           {
7               alert("campo email deverá ser preenchido")
8               return false
9           }
10          else
11          {
12              alert("email preenchido")
13              return true
14          }
15      }
16  </script>
17  <body onload="document.email.n1.focus();"
18  <form name="email" onsubmit="c()"
19      NOME:<input type="text" name="n1" ><br><br>
20      EMAIL:<input type="text" name="r1"><br><br>
21      <input type="submit" name="enviar" value="enviar">
22  </form>
23  </body>
24 </html>
```

- Se a caixa de e-mail não for preenchida retorna um alerta avisando "O campo de e-mail deverá ser preenchido".



- Se a caixa de e-mail for preenchida retorna um alerta avisando “e-mail preenchido”.



. Linguagens para dispositivos móveis

- Quais linguagens posso programar?

Aqui vai uma lista:

- Series 40
 - Java ME
 - Flash Lite
- S60
 - Qt
 - Java ME
 - Flash Lite
 - Symbian C++
 - Open C/C++
 - Python
 - Web Runtime (WRT)
 - Além de outras..
- Maemo
 - C/C++
 - Qt
 - Python
 - Java
 - Ruby
 - Mono

. Características (J2ME – Plataforma Java para dispositivos móveis)

A principal diferença presente na especificação CLDC, é que a KVM não tem suporte a ponto flutuante. Esse suporte foi retirado pelo fato de a maioria dos dispositivos não possuírem suporte em hardware para ponto flutuante e o custo para o suporte em software seria muito elevado [5].

A KVM deve suportar tratamento de exceções. Porém o conjunto de exceções presente é reduzido, logo a capacidade de tratamento de erros é reduzida. Isso se deve por detalhes na recuperação de erros serem muito específicos do dispositivo e o número de classes de erro da especificação completa da linguagem ser muito grande para os dispositivos CLDC [5].

Em relação às características herdadas do ambiente JSE, muitas características foram removidas, as seguintes foram as principais:

Finalização de classe (`Object.finalize()`); Exceções assíncronas (são as exceções que podem ser lançadas em praticamente qualquer lugar [6]); Algumas classes de erro;

Carregadores de classe definidas pelo o usuário; Reflexão (ou introspecção é a capacidade de um programa de investigar fatos sobre si próprios. Por exemplo, um objeto pode "perguntar" a outro quais os métodos que ele possui [7]);

Java Native Interface (padrão de programação que permite que a máquina virtual da linguagem Java acesse bibliotecas construídas com o código nativo de um sistema [8]);

Reflexão, Java Native Interface (JNI) e carregadores de classes definidas pelo usuário são potenciais falhas de segurança. JNI exigem muita memória e pode não ser suportada por dispositivos móveis de pouca memória.