

CSI477 – Sistemas para a WEB I



UFOP

Prof. Fernando Bernardes de Oliveira
<https://sites.google.com/site/fboliveiraufop>

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas – ICEA
Departamento de Computação e Sistemas – DECSI

João Monlevade-MG
1º semestre de 2017

JavaScript

Introdução

- **JavaScript** → originalmente desenvolvido pela Netscape sob o nome de **Mocha**.
- Nome mudado para **LiveScript** → depois para **JavaScript**.
- Autor: **Brendan Eich**.
- **LiveScript** → nome oficial da linguagem quando foi lançada na versão beta do navegador **Netscape 2.0**.
- Nome mudado em um anúncio conjunto com a **Sun Microsystems** → em dezembro de 1995 quando foi implementado no navegador Netscape versão 2.0B3.

JavaScript

Introdução

- **Mudança** → coincidiu com a época em que a Netscape adicionou suporte à tecnologia **Java** em seu navegador.
- **Escolha final do nome** → **confusão** dando a impressão de que a linguagem foi baseada em **Java**;
- **Ação** → caracterizada por muitos como uma **estratégia** de marketing da Netscape para aproveitar a **popularidade** do Java;

JavaScript

Introdução

- ECMAScript / ECMA-262 – <https://tc39.github.io/ecma262/>
- Especificação da linguagem – *scripts*
- Base para Javascript.

JavaScript

Características

- **Comandos e funções JavaScript** → inseridos dentro de um documento da Web, junto com tags HTML e texto;
- Para acessar uma página que possui scripts → o navegador deve ser capaz de interpretar a linguagem.
- Javascript é uma linguagem **baseada em objetos** → orientada a objetos com um **conjunto de objetos** já embutidos.

JavaScript

Características

- **Evento** → sempre que algo acontece em uma página Web.
- **Podem ser qualquer ação** → um botão recebe um clique, o mouse é arrastado, uma página é carregada, um formulário é enviado, e assim por diante.
- **JavaScript** → dirigida por eventos.
- **Projetada** → para reagir quando um evento ocorre.

JavaScript

Características

- Ela não é capaz de recuperar informações de outro arquivo ou salvar dados em um servidor da Web, ou no computador do usuário.
 - Não é possível escrever um script Javascript que, por exemplo, varra os diretórios de um computador, lendo ou apagando arquivos do usuário.
- É uma linguagem **independente de plataforma** → **depende apenas do navegador** que a interpreta.
- JavaScript é *case sensitive*;
- Sintaxe bastante similar a C, C++, Pascal e Delphi;

■ JavaScript não é Java!

- JavaScript é **baseada em objetos** → tem seus próprios objetos embutidos;
- Java é **orientada a objetos** → os objetos são construídos a partir de classes;
- JavaScript usa **tipagem fraca/dinâmica** → as variáveis não precisam ser declaradas, e uma variável ora pode guardar *strings*, ora *números*.
- Java usa **tipagem forte** → as variáveis precisam ser declaradas e usadas para um tipo de dados específico;

JavaScript

Características

- Javascript usa **ligação dinâmica** → referências a objetos são verificadas e resolvidas em tempo de execução.
- Java usa **ligação estática** → referências a objetos devem ser resolvidas quando o programa é compilado.

- *Tag*: `<SCRIPT>...</SCRIPT>`
- Pode-se inserir a *tag* dentro dos elementos `<HEAD>` e `</HEAD>`, `<BODY>` e `</BODY>`, ou ambos;
- É possível incorporar múltiplos elementos `<SCRIPT>` dentro de um documento.
- Scripts dentro da *tag* `<HEAD>` são carregados antes que o resto da página seja carregado;
- Scripts dentro da *tag* `<BODY>` faz com que seja possível criar, dinamicamente, partes de seu documento.

JavaScript

Documento HTML

```
<script>  
codigo-fonte javascript  
</script>
```

```
<script src="funcoes.js"></script>
```

```
<script>...</script>
```

Hierarquia dos Objetos: DOM – Document Object Model

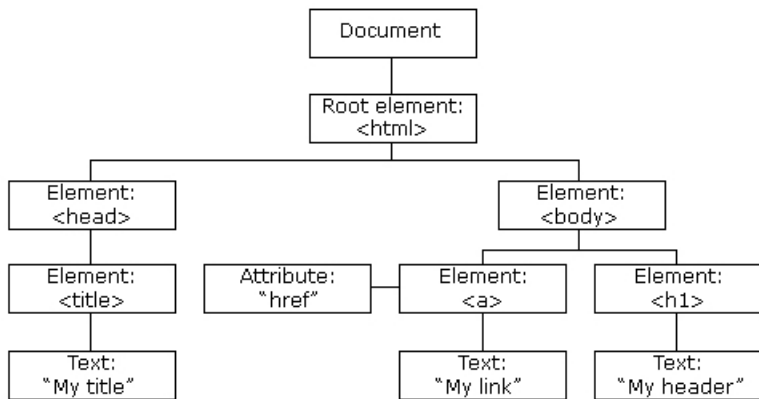


Figura 1: Imagem retirada de http://www.w3schools.com/js/js_htmlDOM.asp

JavaScript

Frameworks e tecnologias

- **JSON** (*JavaScript Object Notation*) → <http://www.json.org/>.
- **jQuery** → <http://jquery.com/>.
- **AngularJS** → <https://www.angularjs.org/>.
- **Node.js** → <https://nodejs.org/>.
- **Express.js** → <http://expressjs.com/>.

Atividade de Pesquisa / Revisão

■ Leituras recomendadas:

- Flanagan (2004)
- Deitel e Deitel (2008): Capítulos 4–11
- Nixon (2009): Capítulos 14–17
- W3Schools (2016): *JavaScript Tutorial*

■ Pesquise sobre os *frameworks* e as tecnologias apresentados.

■ Conteúdos complementares:

- *Netflix Burned by Express.js* – [〈http://www.infoq.com/news/2014/12/expressjs-burned-netflix〉](http://www.infoq.com/news/2014/12/expressjs-burned-netflix)
- *Node.js in Flames* – [〈https://medium.com/netflix-techblog/node-js-in-flames-ddd073803aa4〉](https://medium.com/netflix-techblog/node-js-in-flames-ddd073803aa4)
- *Making Netflix.com Faster* – [〈https://medium.com/netflix-techblog/making-netflix-com-faster-f95d15f2e972〉](https://medium.com/netflix-techblog/making-netflix-com-faster-f95d15f2e972)
- *BrazilJS* – [〈https://braziljs.org/blog/〉](https://braziljs.org/blog/)

Atividade de Pesquisa / Revisão

- *Hipsters Ponto Tech:*

- **#16 – Single Page Applications –**

- ⟨<http://hipsters.tech/single-page-applications-hipsters-16/>⟩

- **#38 – O Reino encantado do JavaScript –**

- ⟨<http://hipsters.tech/o-reino-encantado-do-javascript-hipsters-38/>⟩





Encerramento

Muito obrigado !



Imagens retiradas de: <https://goo.gl/oajVyp> e <https://goo.gl/3H1DM7>

Referências

-  DEITEL, H. M.; DEITEL, P. J. *AJAX, Rich Internet Applications e Desenvolvimento*. [S.l.]: Editora Pearson, 2008.
-  FLANAGAN, D. *Javascript: o Guia Definitivo*. 4. ed. [S.l.]: Bookman, 2004.
-  NIXON, R. *Learning PHP, MySQL, and JavaScript - A Step-by-Step Guide to Creating Dynamic Websites*. [S.l.]: O'Reilly, 2009. I-XVII, 1-505 p. ISBN 978-0-596-15713-5.
-  W3Schools. *W3Schools*. 2016. <http://www.w3schools.com/>.