

FULL STACK WEB DEVELOPER

FERNANDO LIRA

FLAG

JAVASCRIPT

APRESENTAÇÃO – FERNANDO LIRA

3



it.fernandolira@gmail.com



<https://www.linkedin.com/in/fernandolira74/>



+351 93 317 99 21



@fernandolira74



Javascript

JavaScript (ou "JS") é uma linguagem de programação utilizada principalmente para scripts dinâmicos do lado do cliente em páginas web, podendo também ser utilizada no lado do servidor, usando um interpretador como o Node.js

Javascript

O JavaScript não deve ser confundido com a linguagem de programação Java. Apesar de "Java" e "JavaScript" serem marcas comerciais (ou marcas registradas) da Oracle nos EUA e em outros países, essas duas linguagens de programação são significativamente diferentes em sintaxe, semântica e casos de uso.

Javascript

O JavaScript é utilizado principalmente no navegador, permitindo que os programadores manipulem o conteúdo de uma página web por meio do DOM, manipulem dados com o AJAX e o IndexedDB, desenhem gráficos com o canvas, interajam com o dispositivo que está executando o browser através de várias APIs e muito mais.

Javascript – Origem e História

Concebido como uma linguagem do lado do servidor por Brendan Eich (então empregado pela Netscape Corporation), o JavaScript logo chegou ao Netscape Navigator 2.0 em setembro de 1995. O JavaScript obteve sucesso imediato e o Internet Explorer 3.0 (en-US) introduziu o suporte ao JavaScript sob o nome JScript em agosto de 1996.

Javascript – Origem e História

A Microsoft, vendo tudo isso que estava acontecendo com o JavaScript, percebeu que era uma boa ideia e, em vez de ajudar no desenvolvimento do JavaScript e se tornar parceira desta linguagem, resolveu seguir um caminho paralelo ao JavaScript e criou o JScript para rodar somente no Internet Explorer, o que causou uma grande dor de cabeça aos desenvolvedores WEB por volta de 1997.

Javascript – Origem e História

Em novembro de 1996, a Netscape começou a trabalhar com a ECMA International para tornar o JavaScript um padrão do setor. Desde então, o JavaScript padronizado é chamado de ECMAScript e especificado sob a ECMA-262, cuja última edição (décima primeira, ES2020) está disponível desde junho de 2020.

Javascript – Origem e História

Não obstante, ao perceber este problema, uma empresa chamada ECMA resolveu padronizar a linguagem de modo que pudesse funcionar em todos os navegadores. Desta ideia, nasceu um padrão chamado ECMA, e a linguagem foi batizada de ECMAScript.

Javascript – Origem e História

Na verdade, atualmente, todos os programas desenvolvidos em JavaScript são, na realidade, programas em ECMAScript. No entanto, por motivos de marketing e como o nome JavaScript já estava muito bem consolidado, o nome permaneceu JavaScript, em todos os lugares, como livros, revistas, sites e outros.

ECMAScript só é usado para se referir às versões da linguagem.

<https://www.ecma-international.org/>

Javascript – Histórico das versões

ECMAScript 1 (Junho de 1997)

A primeira versão (“de dez dias”).

ECMAScript 2 (Agosto de 1998)

Mudanças editoriais para alinhar a ECMA-262 com o padrão ISO/IEC 16262.

ECMAScript 3 (Dezembro de 1999)

Foi nessa versão onde o JavaScript ganhou implementações importantes como do-while, expressões regulares, novos métodos para o objeto string, tratamento de exceções, entre outras coisas.

Javascript – Histórico das versões

ECMAScript 4 (abandonada em Julho de 2008)

A ECMAScript 4 foi desenvolvida para ser a nova versão do JavaScript com um protótipo escrito em ML. Porém, o TC39 rejeitou o protótipo por causa de suas novas implementações. A quantidade de novas funcionalidades tornaria a migração da ECMAScript 3 para a 4 muito disruptiva.

Javascript – Histórico das versões

ECMAScript 5 (Dezembro de 2009)

Modo estrito ('use strict'), getters e setters, novos métodos de array, suporte a JSON entre outras coisas. Essa é a atualização incremental acordada no fechamento da ECMAScript 4.

ECMAScript 5.1 (Junho de 2011)

Mudanças editoriais para alinhar a ECMA-262 com a terceira versão do padrão ISO/IEC 16262:2011

Javascript – Histórico das versões

ECMAScript 6 (Junho de 2015)

Também conhecida como ECMAScript 2015, é a primeira fase da versão Harmony. Inclui sintaxe muito mais enxuta e funcionalidades como arrow functions, binary data, arrays tipados, coleções (maps, sets e weak maps), promises, melhorias em numerais e matematica, reflection, e proxies.

ECMAScript 7 (Junho de 2016)

Também conhecida como ECMAScript 2016, é a ultima fase da versão Harmony. Inclui features como operadores exponenciais e o método `Array.prototype.includes`.

Alguns browsers ainda não suportam totalmente a versão 6 e 7 da ECMAScript. Porém é possível “transpilar” para ECMAScript 5 através de bibliotecas como o Babel ou Polyfills.

Paradigmas da programação

Um paradigma é um estilo de programação, um modelo, uma metodologia. Não se trata de uma linguagem, mas a forma como se soluciona problemas usando uma determinada linguagem de programação.

Paradigmas da programação

Existem muitas linguagens de programação conhecidas, mas todas elas precisam seguir algumas regras quando implementadas. E essas regras são os paradigmas.

Desta forma, quando uma nova linguagem de programação é desenvolvida, conforme suas especificidades, ela tende a enquadrar-se num paradigma ou até mesmo multiparadigma, como o JavaScript, por exemplo.

Paradigmas da programação

Programação Procedural

A programação procedural é excelente para programação de uso geral e consiste numa lista de instruções para informar ao computador o que fazer passo a passo.

Exemplos:

- C
- Pascal

Paradigmas da programação

Programação Procedural

Quando é recomendado usar programação procedural:

- Quando existir uma operação complexa que inclui dependências entre operações e quando há necessidade de visibilidade clara dos diferentes estados do aplicativo.
- O programa é muito único e poucos elementos foram compartilhados.
- O programa é estático e não se espera que mude muito ao longo do tempo.
- Espera-se que nenhum ou apenas alguns recursos sejam adicionados ao projeto ao longo do tempo.

Paradigmas da programação

Programação Orientada a Eventos

O paradigma de orientação a eventos é usado por toda linguagem de programação que tem uso de recursos gráficos, como jogos e formulários. Dessa forma, a execução do programa dá-se à medida que determinados eventos são despoletados pelo utilizador. Portanto, quem usa é responsável pelo momento em que o programa é executado.

Paradigmas da programação

Programação Orientada ao Objeto

A programação orientada ao objeto (OOP) é o paradigma de programação mais popular devido aos seus benefícios, como a modularidade do código e a capacidade de associar diretamente problemas reais em termos de código.

Neste caso, o programa é escrito como uma coleção de classes e objetos para uma boa comunicação. A entidade menor e básica é objeto e todo tipo de cálculo é realizado apenas nos objetos.

Paradigmas da programação

Programação funcional

O paradigma de programação funcional tem suas raízes na matemática e é independente da linguagem.

A base desse paradigma é a execução de uma série de funções matemáticas. Todo o programa é constituído por funções curtas. Todo o código está dentro de uma função. Todas as variáveis têm escopo definido para a função.

No paradigma de programação funcional, as funções não modificam nenhum valor fora do escopo dessa função e as próprias funções não são afetadas por nenhum valor fora do escopo.