JavaScript – Introdução

**O que é?**

É a linguagem de programação que permite que uma página web interaja com o usuário, com envio de dados, botões, animações, etc.

Atua no lado do cliente (front-end), mas algumas tecnologias permitem que funcione no lado do servidor.

**Conceitos básicos**

* **Front-end X Back-end**

Front-end é a parte da aplicação que o usuário consegue ver e interagir.

Back-end é a parte da aplicação responsável por guardar e enviar dados, que o usuário não consegue ver.

* **JavaScript X Java**

Não há relação de dependência entre as 2 linguagens. A única semelhança é o nome, que foi apenas um marketing do criador do JavaScript, Brandon Eich.

* **Os 3 pilares de um site**
* **HTML**

Responsável pela estrutura do site. Cabeçalhos, artigos, títulos, parágrafos, rodapés...

* **CSS**

Responsável pelo estilo do site. Cores, tamanhos de fontes e imagens, posições de elementos...

* **JavaScript**

Responsável pela interação do site com o usuário. Animações, envio de dados, manipulação de elementos...

Obs: HTML e CSS não são linguagens de programação. JavaScript sim.

**História**

1970 – Guerra Fria. Surge a ARPANET, que anos depois ficaria conhecida como Internet, que usamos até hoje.

1993 – Tim Berners-Lee cria a linguagem HTML, o protocolo HTTP (HyperText Transfer Protocol) e a WWW (World Wide Web). Marc Andreessen cria o primeiro navegador, chamado Mosaic.

1994 – Jim Clark e Marc Andreessen fundam a empresa Netscape, que criou o navegador Netscape, baseado no Mosaic.

1995 – Brandon Eich cria a linguagem Mocha, ou LiveScript. A empresa Sun Microsystems cria a linguagem Java. Brandon Eich, em uma jogada de marketing, muda o nome de Mocha para JavaScript. A empresa Microsoft cria o navegador Internet Explorer, o SO Windows 95 e o Jscript, como um JavaScript melhorado. A ECMA padroniza o JavaScript, criando assim o ECMAScript.

2002 – Falência da Netscape. Criação da Fundação Mozilla, que criou o navegador Firefox, usado até hoje.

2008 – Surge o Google Chrome, mais popular navegador atualmente.

2009 – Criação do V8, motor interno open-source de JavaScript do Google Chrome.

2010 – Surge o Node.js, tecnologia que consegue rodar JavaScript no lado do servidor (back-end).

**JavaScript X ECMAScript**

* **ECMA**

A Associação Europeia de Fabricantes de Computadores é uma organização sem fins lucrativos que padroniza hardwares, comunicações e linguagens de programação.

* **Diferença**

JavaScript é a linguagem de programação. ECMAScript é a padronização dessa linguagem, para que os navegadores sigam esse padrão e evitem problemas de compatibilidade.

* **ECMAScript – Linha do tempo**

1997 – ECMAScript 1.0

1998 – ECMAScript 2.0

1999 – ECMAScript 3.0

A versão 4.0 nunca foi lançada. A partir da versão 5, a nomenclatura muda.

2009 – ES5

2015 – ES6. Essa é a versão mais compatível com os navegadores atuais, e iremos estudar alguns recursos dela também. Depois dessa versão, a nomenclatura muda novamente.

ES2016

ES2017

ES2018

ES2019

ES2020

ES2021, que foi recentemente liberada.

**Frameworks e Bibliotecas**

* **Framework**

Uma estrutura que tem como objetivo ajudar no desenvolvimento de algum produto.

* **Biblioteca**

Um conjunto de implementações escritas em uma linguagem e implementadas no seu código.

* **Diferença**

Usando uma biblioteca, você pode usar apenas o recurso dela que você precisa e seguir escrevendo seu código normalmente. Usando um framework, você terá que seguir os padrões da estrutura do framework utilizado.

* **Frameworks e bibliotecas JavaScript**
* **Bibliotecas mais populares**

React (para construção de interfaces web)

jQuery (para simplificar o código JS)

* **Frameworks mais populares**

Angular, Vue (para construção de interfaces web)

Ionic, React Native (para desenvolvimento de apps mobile)

Electron (para desenvolvimento de apps desktop)

**Node.js**

* **Definição**

Node.js é um ambiente de execução JavaScript no lado do servidor, ou seja, no back-end da aplicação.

* **Vantagens**

A vantagem do Node.js sobre as outras linguagens back-end (como PHP, Java) é a capacidade de tratar várias requisições simultâneas, enquanto outras linguagens demandam mais capacidade computacional para fazer o mesmo.

* **NPM**

O Node Package Manager (NPM, Gerenciador de Pacotes do Node) é uma ferramenta de linha de comando que ajuda a interagir com plataformas online, instalar e desinstalar pacotes (por exemplo, um framework ou biblioteca que você queira usar no projeto).