**1 JAVASCRIPT**

**1.1 DEFINIÇÃO**

“JavaScript é a linguagem de programação da Web. A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos - em computadores de mesa, consoles de jogos, tables e smartphones - incluem interpretadores JavaScript, tornando a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas. JavaScript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientada a objetos e funcionais. A sintaxe de JavaScript é derivada da Linguagem Java, das funções de primeira classe de Scheme e da herança baseada em protótipos de Self. A linguagem JavaScript básica define uma API mínima para trabalhar com texto, arrays, datas e expressões regulares, mas não inclui funcionalidade alguma de entrada ou saída. Entrada e saída (assim como recursos mais sofisticados, como conexão em rede, armazenamento gráficos) são de responsabilidade do "ambiente hospedeiro" dentro do qual JavaScript está incorporada.” (FLANAGAN, 2013).

“JavaScript é a linguagem de script do lado do cliente mais amplamente utilizada. Ela é realmente poderosa e dinâmica. Outras populares linguagens de script do lado do cliente incluem VBScript e Python. Linguagens do lado do cliente são usadas para tornar seus sites mais interativos, animados e responsivos. Você pode usar JavaScript para ajudar os usuários a interagirem melhor com páginas da Web. O JavaScript também pode ser usado para controlar o navegador, comunicar-se de forma assíncrona com o servidor, alterar de forma dinâmica o conteúdo de uma página Web e para desenvolver jogos e aplicações móveis e de desktop.” (PRESCOTT, 2016).

**1.2 APLICAÇÕES**

https://becode.com.br/aplicacoes-que-podem-ser-criadas-aprendendo-html-css-e-javascript/

**1.3 EVENTOS (O QUE SÃO, COMO USAR E OS PRINCIPAIS)**

“Genericamente falando, podemos dizer que eventos são ações capazes de disparar uma reação. Veja este exemplo: clicar em um link ou colocar o ponteiro do mouse sobre um elemento são eventos. Ao clicar um link, desencadeamos uma reação que poderá ser a de abrir uma nova página. Ao colocar o ponteiro do mouse sobre um elemento, podemos, por exemplo, alterar a opacidade dele.

Eventos são muito usados em JavaScript e viabilizam a interatividade em uma página Web. Na verdade, eventos viabilizam a própria existência da JavaScript. Sem eles, não teríamos como fazer funcionar os scripts.

Mostraremos apenas os eventos denominados onlick e onload que ocorrem respectivamente quando o usuário clica um elemento qualquer da página (não necessariamente um link) e quando um documento é carregado. Eventos podem ser disparados pelo usuário, como é o caso de onclick, ou ocorrer independentemente da interferência do usuário, como é o caso de onload.

<button type="button" onclick="alert('Caixa de alerta');"Alert</button>

<button type="button" onclick="confirm('Caixa de confirmação');"Confirm</button>

<button type="button" onclick="prompt('Caixa de dados');"Prompt</button>

Observe que atrelamos o evento onclick a três botões que, ao serem clicados, criarão, cada um deles, uma caixa de diálogo. Note que os eventos previstos na linguagem JavaScript são atributos de HTML cujo valor determina a ação a ser disparada”. (SILVA, 2010).

“Os eventos são basicamente um conjunto de ações que são realizadas em um determinado elemento da página web, seja ele um texto, uma imagem, ou uma div, por exemplo. Muitas das interações do usuário que está visitando sua página com o conteúdo do seu site podem ser consideradas eventos.

Existe uma infinidade de eventos definidos para uso em JavaScript, abaixo podemos ver alguns dos principais e mais utilizados:

onBlur remove o foco do elemento

onChange muda o valor do elemento

onClick o elemento é clicado pelo usuário

onFocus o elemento é focado

onKeyPress o usuário pressiona uma tecla sobre o elemento

onLoad carrega o elemento por completo

onMouseOver define ação quando o usuário passa o mouse sobre o elemento

onMouseOut define ação quando o usuário retira o mouse sobre o elemento

onSubmit define ação ao enviar um formulário.” (DEVMEDIA, 2013).

**1.4 FUNÇÕES**

“Em linhas gerais, uma função destina-se a definir um bloco de código de código a ser executado em determinados pontos do script. Pode ocorrer que determinados cálculos se repitam em vários lugares diferentes de um script. Para não reescrever os mesmos códigos repetidamente, podemos criar uma função que efetue os cálculos e "chamar" a função sempre que necessário.” (SILVA, 2010).

**1.5 COMO USAR (INTERNO E EM UM ARQUIVO EXTERNO)**

“Inline: Inserir o script diretamente na seção body do documento. Trata-se de prática não recomendada de acordo com o princípio da separação das camadas de desenvolvimento.

Incorporado: Inserir o script na seção head do documento.

Externo: Escrever o script em um arquivo externo e inserir com um link na seção head do documento.” (SILVA, 2010).

**2 DOM**

**2.1 O QUE É**

“O Document Object Model (DOM) é uma interface de programação para os documentos HTML e XML. Representa a página de forma que os programas possam alterar a estrutura do documento, alterar o estilo e conteúdo. O DOM representa o documento com nós e objetos, dessa forma, as linguagens de programação podem se conectar à página. Uma página da Web é um documento. Este documento pode ser exibido na janela do navegador ou como a fonte HTML. Mas é o mesmo documento nos dois casos. O DOM (Document Object Model) representa o mesmo documento para que possa ser manipulado. O DOM é uma representação orientada a objetos da página da web, que pode ser modificada com uma linguagem de script como JavaScript.” (MOZILLA, 2022).

“DOM, que em português significa Modelo de Objeto de Documentos, é a interface de programação utilizada para acesso e manipulação de dados em documentos HTML e XML. É um modelo que disponibiliza os dados como um conjunto de nós, os quais podem ser acessados por meio de um código JavaScript. Os nós seguem uma hierarquia, tornando, portanto, mais organizado o acesso, a inclusão, a alteração e a exclusão de informações.” (NIEDERAUER, 2007).

**2.2 INTERAÇÃO JS COM O DOM**

O DOM não é uma linguagem de programação, mas sem ela, a linguagem JavaScript não teria nenhum modelo ou noção de páginas da web, documentos HTML, documentos XML e suas partes componentes (por exemplo, elementos). Cada elemento de um documento - o documento como um todo, o cabeçalho, as tabelas do documento, os cabeçalhos da tabela, o texto nas células da tabela - faz parte do modelo de objeto do documento desse documento, para que todos possam ser acessados e manipulados usando o método DOM e uma linguagem de script como JavaScript. (MOZILLA, 2022).

“Ainda que o DOM seja uma API independente de linguagem, sua interface, no navegador, é implementada em JavaScript. Visto que a maior parte do trabalho na criação de scripts no lado do cliente está relacionada ao documento subjacente, o DOM é uma parte importante da codificação JavaScript diária. É comum nos navegadores que as implementações DOM e JavaScript sejam mantidas independentes. No Internet Explorer, por exemplo, a implementação JavaScript é chamado JScript e está localizada em um arquivo de biblioteca chamado jscript, dll, enquanto a implementação DOM encontra-se em outra biblioteca, a mshtml.dll (internamente chamada de Trident). Essa separação permite que outras tecnologias e linguagens, como VBScript, se beneficiem do DOM e da funcionalidade de renderização que a Trident oferece. O safari utiliza o WebCore do WebKit para DOM e renderização e apresenta uma engine JavaScriptCore separada (chamada SquirrelFish em suas versões mais recentes). O Google Chrome também usa bibliotecas WebCore do WebKit para renderização de páginas, mas implementa sua própria engine JavaScript chamada V8. No Firefox, o SpiderMonkey (TraceMonkey na última versão) é a implementação JavaScript, uma parte separada da engine Gecko de renderização.” (ZAKAS, 2010).

**Referências:**

FLANAGAN, David. **JavaScript. O guia definitivo**. Bookman Editoria, Rio Grande do Sul, 2013.

Prescott, Preston. **Programando em JavaScript**. Editora Babelcube, 2016.

SILVA, Mauricio. **JavaScript. Guia do Programador**. Novatec Editora, São Paulo, 2010.

DEVMEDIA. **Trabalhando com eventos em JavaScript**.<https://www.devmedia.com.br/trabalhando-com-eventos-em-javascript/28521>

MOZILLA. **Introdução ao DOM.** <<https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model/Introduction>>

NIEDERAUER, Juliano. **Web Interativa com Ajax e PHP**. Novatec Editora, São Paulo, 2007.

ZAKAS, Nicholas. JavaScript de Alto Desempenho. Editora Novatec, São Paulo, 2010.