

## JavaScript: Programando na linguagem da web

- Esse curso capacita a:
  - Entender mais como funciona esta popular linguagem do mercado;
  - Aprender a manipular os elementos da página;
  - Trabalhar bem com eventos do navegador;
  - Entender como utilizar e validar formulários;
  - Conheça boas práticas de organização de código Javascript;
  - Buscar dados em outro servidor com AJAX;

Aulas:

### 1. Introdução:

- **alert(“Mensagem”)** [Manda uma mensagem pop-up no navegador];
- Variável **document**, é a representação do DOM ou Document Object Model, que é a representação da nossa página HTML para o navegador (representação em memória do HTML, que o navegador usa para desenhar a página), que é utilizada pelo Javascript para manipular a página;
- **document.querySelector()** [Quando deseja-se pegar apenas parte do HTML, faz a busca através de seletores CSS];
  - Separar a organização do código em arquivos .js;
  - Importar arquivos .js sempre ao final do body;
  - A representação do HTML pela variável **document**, a função que faz busca **querySelector()** no qual utilizar preferindo selecionar por #id ou .classe;

### 2. Variáveis e Operadores:

- O operador lógico de OU ( || );
- O operador lógico de E ( && );
- Busca de elementos;

### 3. Arrays, Loop e Estilos:

- **querySelectorAll()** [Retorna um array com os elementos indicados];
- **.style.<comandoCSS> = <css>** [Altera o visual da página injetando uma tag style com o Javascript];
- **.classList.add()** [Adiciona uma classe ao elementos trazidos pelo **querySelector**];
- A o método **toFixed()** controla o número de casas decimais exibidas, usada só em número;
  - Como selecionar diversos elementos com a função **querySelectorAll()**;
  - Como manipular estilos com a propriedade **.style**;
  - Utilizar camelCase quando quiseremos modificar uma propriedade que tenha duas palavras, como a **background-color**;

- A boa prática de não alterar o estilo diretamente no Javascript e sim modificar as classes dos elementos com a propriedade `.classList`;

#### 4. Eventos, Formulários e Criando Elementos:

- `.addEventListener()` ["Escuta" a página];
- `.createElement("tr")` [Cria um elemento no HTML, o parâmetro passado é o nome da tag a ser criada];
- `.appendChild()` [Insere as tag filhos dentro da tag mãe];
  - A diferença entre as funções nomeadas e as funções anônimas;
  - A escutar eventos do browser com a função `addEventListener()`;
  - Que a função criadora de elementos é `.createElement()`;
  - A pegar o valor de um input por meio da propriedade `.value`;
  - A acessar os input de um form por meio da propriedade `_name_`;
  - A adicionar elementos na página e dentro de outros elementos com a função `appendChild()`;

#### 5. Boas Práticas com Javascript:

- Dividir o código, separando por funcionalidades do sistema;
- Os objetos em Javascript;
- A quebrar funções grandes em funções menores, com cada uma com sua responsabilidade;
- A função `form.reset()` para limpar o formulário;

#### 6. Validação de Formulários:

- `.forEach()` [Todo array possui, passa-se para ela uma função anônima por parâmetro, e nessa fazemos o que quisermos para cada item, ele é recebido por parâmetro na função interna];
- `objetoDeUmElementoHTML.innerHTML = "Novo conteudo"` [Para editar o HTML interno de um elemento];
  - Mais organização de código, exportando as funções de validação;
  - O operador de negação NOT (!);
  - Como validar um formulário;
  - A função `push` para colocar elementos dentro de um array;
  - A propriedade `innerHTML` dos elementos, que foi usada para apagar os itens da `<ul>`;
  - O método `forEach` para percorrer arrays;

#### 7. Remoção, Delegação e Animação:

- O novo evento de `dblclick`;
- Como os eventos propagam através da página;
- A delegação de eventos;
- Como animar uma remoção com o `setTimeout`;

## 8. Filtrando uma tabela:

- **variavel = new RegExp()** [Expressão Regular: O primeiro parâmetro que deve-se passar é o texto da expressão regular, o que deve ser buscado e o segundo parâmetro são uma ou mais flags (representando como queremos que a expressão regular busque). Por exemplo, pode-se definir que não queremos que haja distinção entre letras maiúsculas e minúsculas, através da flag **i**;
  - O novo evento de input;
  - O truque de deixar invisível adicionando uma classe com **display:none**;
  - Como realizar uma busca, escondendo todos os elementos e mostrando apenas aqueles que nos interessam;
  - Como buscar por apenas um pedaço de uma palavra com as Expressões Regulares em Javascript;

## 9. Buscando pacientes com AJAX:

- **var xhr = new XMLHttpRequest();** [O objeto XMLHttpRequest é quem é responsável por fazer requisições HTTP assíncronas com Javascript. Apesar de ter o XML no nome hoje em dia pode trafegar diversos outros tipos de dados além do XML, como JSON;
- **xhr.open("POST","www.xyz.com.br/compras");** [Para configurar a requisição para dizer para qual servidor queremos enviá-la e também qual método HTTP será usado;
  - Como fazer uma requisição assíncrona com o Javascript (Conceito de AJAX);
  - O objeto **XMLHttpRequest**;
  - Como configurar e enviar uma requisição;
  - O evento de **load**;
  - O que é a técnica AJAX;
  - O formato JSON;
  - Como converter JSON para um objeto Javascript com a função **JSON.parse()**;
  - Como lidar com erros durante o AJAX;