

Laboratório Prático: Escrevendo Sua Primeira Função JavaScript

Tempo estimado necessário: 30 minutos

O que você vai aprender

Neste laboratório, você irá explorar conceitos fundamentais de JavaScript que são vitais para o desenvolvimento web. Você aprenderá a usar funções para criar um aplicativo web interativo que calcula a área de um retângulo com base em valores fornecidos pelo usuário.

Objetivos de aprendizagem

Após concluir este laboratório, você será capaz de:

- **Compreender a estrutura HTML:** Reconhecer a estrutura de um arquivo HTML e entender como criar uma interface de usuário interativa implementando elementos HTML como campos de entrada e botões.
- **Declaração de variáveis em JavaScript:** Compreender o conceito de declarar variáveis em JavaScript e entender seu papel no armazenamento de dados recuperados da entrada do usuário.
- **Recuperando entrada do usuário com JavaScript:** Aprender a usar JavaScript para buscar e processar dados fornecidos pelo usuário inseridos em campos de entrada dentro de um documento HTML.
- **Implementação e execução de funções:** Compreender a criação e execução de funções JavaScript e seu papel na execução de tarefas específicas, como realizar cálculos e atualizar dinamicamente o conteúdo HTML com base nas ações do usuário.

Pré-requisitos

- Conhecimento básico de HTML e comandos Git.
- Compreensão básica de funções e sua sintaxe.
- Navegador web com console (Chrome DevTools, Firefox Console, etc.).

Passo 1: Configurando o ambiente

1. Primeiramente, você precisa clonar seu repositório principal no **Ambiente Skills Network** que você criou no primeiro laboratório e onde você tem enviado todo o trabalho relacionado aos laboratórios anteriores. Siga os passos dados para clonar este repositório:

- Clique no terminal no canto superior direito e depois selecione **Novo Terminal**.
- Execute o comando `git clone` escrevendo o comando dado no terminal.

```
git clone <github-repository-url>
```

Nota: Coloque o link do seu próprio repositório GitHub em vez de `<github-repository-url>`.

- O passo acima clonará a pasta do seu repositório GitHub na pasta do projeto no explorador. Você também terá várias pastas dentro da pasta clonada.
- Agora você precisa navegar para dentro da pasta clonada. Para isso, escreva o comando dado no terminal:

```
cd <repository-folder-name>
```

Nota: Escreva o nome da sua pasta clonada em vez de `<repository-folder-name>`. Execute `git clone` se você saiu do **Ambiente Skills Network** e não consegue ver nenhum arquivo ou pasta após fazer login novamente.

2. Agora selecione a pasta **cloned Folder Name**, clique com o botão direito sobre ela e clique em **Nova Pasta**. Insira o nome da pasta como **calculateArea**. Isso criará a pasta para você. Em seguida, selecione a pasta **calculateArea**, clique com o botão direito e selecione **Novo Arquivo**. Insira o nome do arquivo como **calculate_Area.html** e clique em OK. Isso criará seu arquivo HTML.
3. Agora, selecione a pasta **calculateArea** novamente, clique com o botão direito e selecione **Novo Arquivo**. Insira o nome do arquivo como **calculate_Area.js** e clique em OK para criar seu arquivo JavaScript.
4. Crie a estrutura básica do template para o arquivo **calculate_Area.html** adicionando o conteúdo fornecido.
 - Dentro do arquivo HTML, crie um formulário de entrada para coletar o comprimento e a largura do retângulo junto com um botão para acionar o cálculo.
 - Para isso, inclua o código fornecido no arquivo **calculate_Area.html**.

```

<html>
<head>
    <title>Calculadora de Área do Retângulo</title>
</head>
<body>
    <h1>Calculadora de Área do Retângulo</h1>
    <label for="length">Digite o comprimento: </label>
    <input type="number" id="length"><br><br>
    <label for="width">Digite a largura: </label>
    <input type="number" id="width"><br><br>
    <button onclick="calculateArea()">Calcular Área</button><br><br>
    <p id="result"></p>
<script src=".//calculate_Area.js"></script>
</body>
</html>

```

5. O código HTML fornecido inclui:

- **Título e cabeçalho da página:** Configura uma página da web intitulada “Calculadora de Área do Retângulo” e apresenta um cabeçalho principal `<h1>` exibindo o mesmo título, garantindo clareza sobre o propósito da página.
- **Campos de entrada para dados do usuário:** Fornece campos de entrada rotulados para comprimento e largura `<input type="number" id="length">` e `<input type="number" id="width">` para permitir que os usuários insiram valores numéricos para as dimensões do retângulo.
- **Acionador de cálculo e exibição:** Inclui um botão `<button onclick="calculateArea()">Calcular Área</button>` para executar uma função JavaScript chamada `calculateArea()` ao ser clicado. O resultado calculado da área do retângulo será exibido dentro do elemento `<p>` com o ID ‘result’.
- **Exibição dinâmica do resultado:** Prepara um espaço reservado `<p id="result"></p>` para exibir dinamicamente a área calculada, criando uma interface amigável para feedback em tempo real após a execução do cálculo.
- Uma tag `<script>` é adicionada para incluir o arquivo js no arquivo `calculate_Area.html` usando o atributo `src`.

Passo 2: Definindo variáveis e função para calcular área

1. Declare duas variáveis chamadas **length** e **width** no arquivo `calculate_Area.js`, mas não atribua nenhum valor a elas ainda. Essas variáveis serão usadas para armazenar o comprimento e a largura do retângulo fornecidos pelo usuário através de um formulário no arquivo HTML.

```
let length;
let width;
```

2. Agora crie uma função chamada `calculateArea` no arquivo `calculate_Area.js` da seguinte forma:

```
function calculateArea() {
```

3. Dentro da função acima, obtenha os valores do usuário como entrada. Para isso, você precisa pegar os valores usando `document.getElementById` da entrada do usuário dentro da função `calculateArea` da seguinte forma:

```
function calculateArea() {
    length = parseFloat(document.getElementById('length').value);
    width = parseFloat(document.getElementById('width').value);
}
```

4. O código acima inclui:

- **`document.getElementById ('length')`:** Esta parte do código recupera um elemento HTML pelo seu ID, especificamente procurando um elemento com o ID 'length'.
- **.value:** Após acessar o elemento HTML, `.value` é usado para recuperar o valor inserido no campo de entrada associado a esse elemento. Por exemplo, se um usuário inserir '5' no campo de entrada para o comprimento, `.value` recupera a string '5'.

- **parseFloat(...):** A função parseFloat() converte o valor da string recuperado do campo de entrada em um número de ponto flutuante. Essa conversão garante que a entrada, tipicamente texto inserido pelo usuário, seja tratada como um número e possa ser usada em operações matemáticas.
- **length e width:** Finalmente, os números de ponto flutuante obtidos (representando os valores de comprimento e largura inseridos pelo usuário) são armazenados nas variáveis length e width, respectivamente. Essas variáveis serão utilizadas para cálculos adicionais, como determinar a área de um retângulo neste contexto.

5. Em seguida, declare uma variável chamada **area** e inicialize-a com **length * width** no arquivo **calculate_Area.js** da seguinte forma:

```
function calculateArea() {
  length = parseFloat(document.getElementById('length').value);
  width = parseFloat(document.getElementById('width').value);
  let area = length * width;
}
```

6. Após o cálculo da área do retângulo ser concluído e armazenado na variável chamada **area**, o código fornecido envolve apresentar ou exibir esse resultado na interface do usuário. Inclua o código fornecido dentro da função após o cálculo da área.

```
document.getElementById('result').innerText = `The area of the rectangle is: ${area}`;
```

7. O código acima inclui:

- **document.getElementById ('result'):** Esta parte do código recupera um elemento HTML pelo seu ID. Especificamente, ela mira em um elemento com o ID 'result'.
- **.innerText = A área do retângulo é: \${area};:** Uma vez que o elemento é acessado, .innerText é usado para modificar o conteúdo de texto dentro desse elemento HTML.
- Os backticks e a notação \${} permitem a inclusão de variáveis JavaScript dentro de uma string (usando literais de template). Neste caso, ele define o conteúdo de texto para exibir uma mensagem junto com a área calculada armazenada na variável **area**. Por exemplo, se **area** contém um valor de 25, o texto exibido será “A área do retângulo é: 25”.

Passo 3: Executar comandos Git

1. Execute `git add` para adicionar os arquivos e pastas mais recentes no ambiente git.

```
git add --a
```

- Certifique-se de que o terminal esteja com o mesmo caminho conforme abaixo:

2. Em seguida, execute `git commit` no terminal. Ao executar `git commit`, o terminal pode mostrar uma mensagem para configurar seu `git config --global` para `user.name` e `user.email`. Então você precisará executar o comando `git config` também para `user.name` e `user.email` conforme indicado.

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Seu Nome"
```

Agora, após isso, você pode executar o comando `git` conforme abaixo:

```
git commit -m "mensagem"
```

3. Em seguida, execute `git push` apenas escrevendo o comando dado no terminal.

```
git push origin
```

Após o comando `push`, o sistema solicitará que você insira seu nome de usuário e senha. Insira o nome de usuário da sua conta GitHub e a senha que você criou no primeiro laboratório. Após inserir as credenciais, todas as suas pastas e arquivos mais recentes serão enviados para o seu repositório GitHub.

Passo 4: Verifique a saída

1. Para visualizar sua página HTML, clique com o botão direito no arquivo **calculate_Area.html** após selecioná-lo, e em seguida selecione "Abrir com Live Server."
 - O servidor deve iniciar na porta 5500, indicado por uma notificação no canto inferior direito.
2. Clique no botão Skills Network à esquerda (consulte o número 1). Essa ação abrirá a "Caixa de Ferramentas do Skills Network." Em seguida, selecione "Iniciar Aplicação" (consulte o número 2). Uma vez lá, insira o número da porta **5500** em "Porta da Aplicação" (consulte o número 3) e clique neste botão .
3. Isso abrirá seu navegador padrão onde você verá o nome da pasta **cloned-folder-name**. Clique nessa pasta **cloned-folder-name**. Após clicar, você verá vários nomes de pastas; entre essas pastas, clique na pasta **calculateArea**. Você verá arquivos relacionados a essa pasta onde novamente você clicará no arquivo **calculate_Area.html**, conforme mostrado abaixo.
3. Isso abrirá a página HTML onde você poderá inserir os valores de comprimento e largura, conforme mostrado abaixo.
4. Em seguida, clique no botão **Calcular Área** e você verá a resposta.

Nota: Após colar o código, lembre-se de salvar seu arquivo. Se você editar seu código, basta atualizar seu navegador, que está rodando pela porta 5500. Dessa forma, não há necessidade de iniciar a aplicação repetidamente.

Tarefa de prática

Nesta tarefa, você precisa criar uma função chamada **groceryTracker** para calcular o valor total dos itens de supermercado comprados. Para isso:

1. Inclua o seguinte no arquivo HTML:
 - Você precisa criar pelo menos três caixas de entrada com IDs nomeados como “`grocery1`” e assim por diante.
 - Também rotule-as usando `<label>` como “Insira o valor do primeiro item de supermercado” e assim por diante.
 - Crie um botão que calcula o total gasto nas compras de supermercado.
2. Inclua o seguinte no arquivo JavaScript:
 - Crie uma função que aceite esses valores inseridos pelos usuários como parâmetro.
 - Em seguida, escreva a lógica para calcular o total gasto na compra de supermercado.
 - Chame essa função de tal forma que, após clicar no botão, ela mostre o valor total da compra de supermercado.

Resumo

1. **Configurando o ambiente:** Criando arquivos HTML e JavaScript, inicializando a estrutura básica do HTML, campos de entrada e ativando a funcionalidade de cálculo.
2. **Definindo variáveis e funções:** Declarando variáveis para armazenamento de entrada do usuário e elaborando uma função `calculateArea()` para processar essa entrada. Recuperando os valores inseridos pelo usuário e calculando a área de um retângulo com base nesses valores.
3. **Exibindo o resultado calculado:** Atualizando dinamicamente o conteúdo HTML com a área calculada. Utilizando JavaScript para modificar elementos específicos dentro do documento HTML, fornecendo feedback em tempo real ao usuário.