

Módulo 06

Métodos funcionais de array (Aula 04)

Exercício 1 (upload)

Vamos adicionar uma nova funcionalidade ao último exercício da aula passada (aplicação de cadastro de produtos).

Agora a página deverá ter também um formulário de busca, que é composto por: um input textual e um botão "Buscar".

Ao clicar em "Buscar", a aplicação deverá procurar, entre os produtos já cadastrados, todos cujo "nome" ou "descrição" contenham o termo de busca, e exibir somente esses produtos na tabela.

A tabela deve continuar funcionando da mesma maneira (permite ver as informações do produtos, editar e apagar).

Se o usuário limpar o input de busca e clicar em "Buscar", a tabela volta a exibir todos os produtos (isso é consequência lógica do que já foi descrito, porque nesse caso o termo de busca é "", e a string vazia está contida em toda string).

Sendo mais específico sobre o termo de busca:

Ele deve estar contido no nome ou na descrição do produto, por exemplo se o termo de busca é "ab", então todos os produtos cujo nome tem a substring "ab" ou a descrição tem a substring "ab" devem ser exibidos. Sugestão: pesquise sobre um método de string chamado `includes()`.

E o termo de busca é "case insensitive": quer dizer que digitar "AB" é o mesmo que "ab" ou "Ab" ou "aB". Sugestão: converta o termo de busca e as informações do produto para letra minúscula quando estiver fazendo a busca, pesquise sobre o método de string `toLowerCase()`.

Exercício 2 (upload)

Crie uma página web que peça ao usuário que digite a data de nascimento de uma pessoa e seu gênero. Após clicar em um botão de "verificar" realize os seguintes procedimentos:

Armazene as informações inseridas num objeto chamado `person`, nas propriedades `birthdate` e `gender`, respectivamente.

O objeto criado deve conter em uma propriedade `daysToDeath` uma função (método) que, de acordo com os valores das propriedades `birthdate` e `gender`, retorne a quantidade de dias (aproximadamente) que faltam para a sua morte considerando a expectativa de vida fornecida pelo IBGE de 73,1 anos para homens e 80,1 anos para mulheres.

Mostre o número de dias restantes para o usuário.

Módulo 06

Subprogramação (Aula 04)

Exercício 3 (upload)

Numa certa página web, há um formulário para preenchimento de nome e telefone.

O nome é requerido (não pode ser vazio, após `trim()`), e deve ter tamanho pelo menos 3 caracteres.

O telefone deve estar no formato "(XX) XXXX-XXXX" ou "(XX) XXXXX-XXXX".

A página tem um botão "Cadastrar" que, ao ser clicado, verifica se os inputs estão corretos e, se estiverem, exibe no HTML a mensagem "Cadastro bem sucedido".

Caso haja problemas nos inputs, a página exibe no HTML uma mensagem de erro "Dados inválidos".

O código de ouvinte de evento do botão é como segue:

```
document.querySelector("button").addEventListener("click", function () {  
  // name-input e telephone-input são  
  // <input type="text"> presentes no HTML  
  const name = document.getElementById("name-input").value;  
  const phone = document.getElementById("telephone-input").value;  
  
  if (!nameIsValid(name) || !phoneIsValid(phone)) {  
    showError();  
  } else {  
    showSuccess();  
  }  
});
```

Está faltando nesse código a implementação das funções `nameIsValid`, `phoneIsValid`, `showError` e `showSuccess`.

Implemente essas funções como fica subentendido no código dado, para que o site funcione integralmente (escreva o HTML).

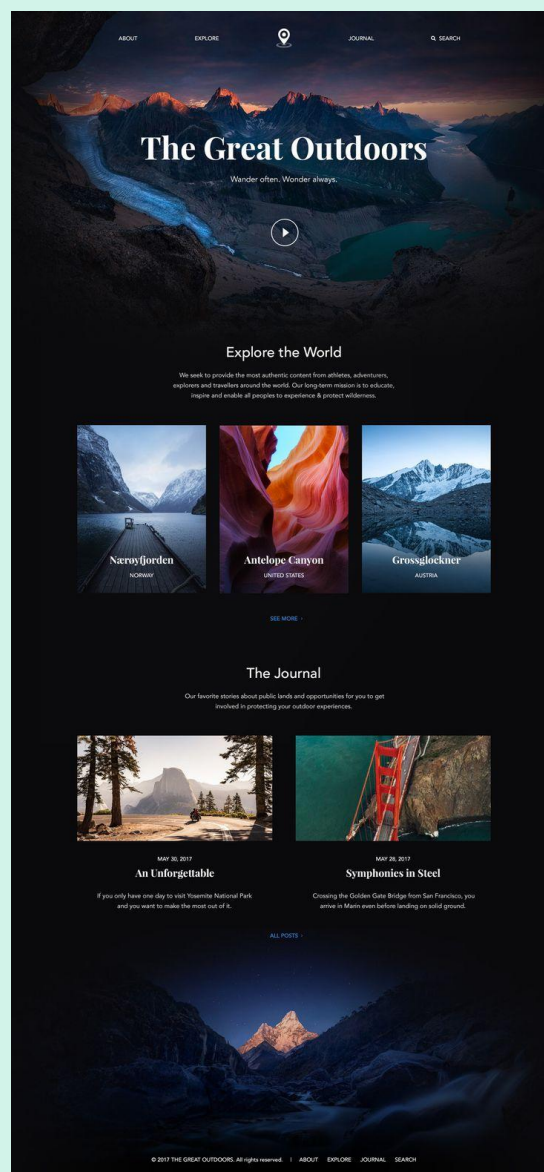
Módulo 06

Módulos (Aula 04)



Questões de aprendizagem

As próximas tarefas visam a criação dos componentes necessários a construção da seguinte página Web:



fonte: br.pinterest.com

Para isso, você deverá utilizar os conceitos aprendidos sobre módulos do Javascript.

Os seguintes passos podem ser tomados para esta implementação:

1. **Header:** deverá retornar o elemento HTML do header da página:

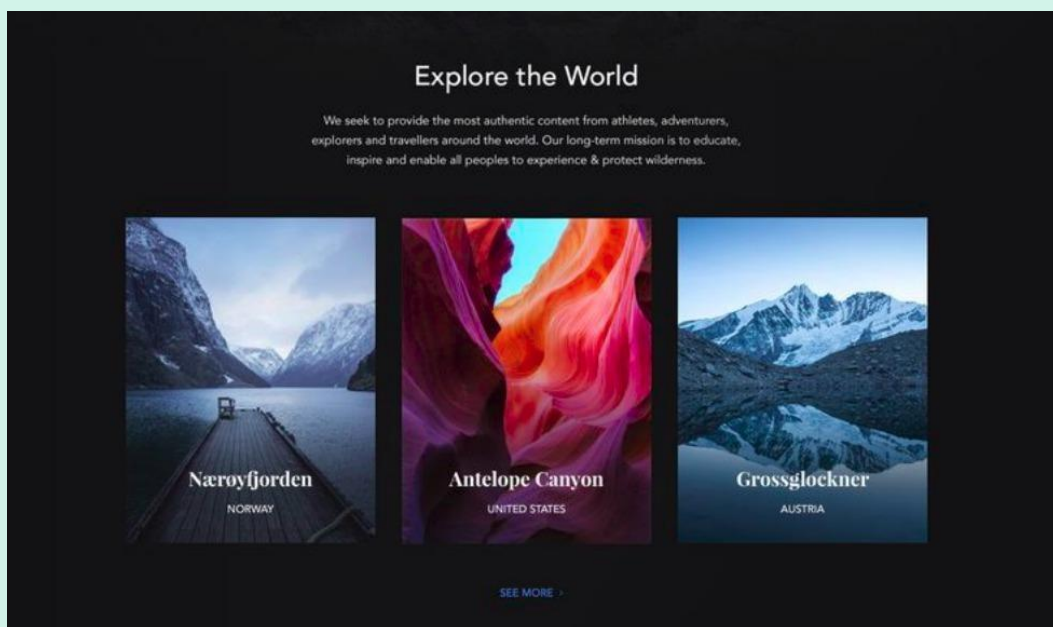


2. **Outdoor:** deverá retornar o element HTML do outdoor:



OBS: Não se preocupe com o vídeo indicado no design.

3. **"Explore the World":** deverá retornar o elemento HTML com todas as informações do componente:



Para construir os cards de pontos turísticos, crie um array não exportado no módulo do componente que contenha a lista de pontos turísticos:


```
[
  {
    city: "", // cidade do ponto turístico
    country: "", // país do ponto turístico
    image: "", // pais do ponto turístico
  },
  ...
]
```

Esse array deverá ser utilizado para a criação dos cards.

Observe que os cards do componente possuem a mesma estrutura. Sendo assim, um componente específico pode ser implementado para a construção do card:

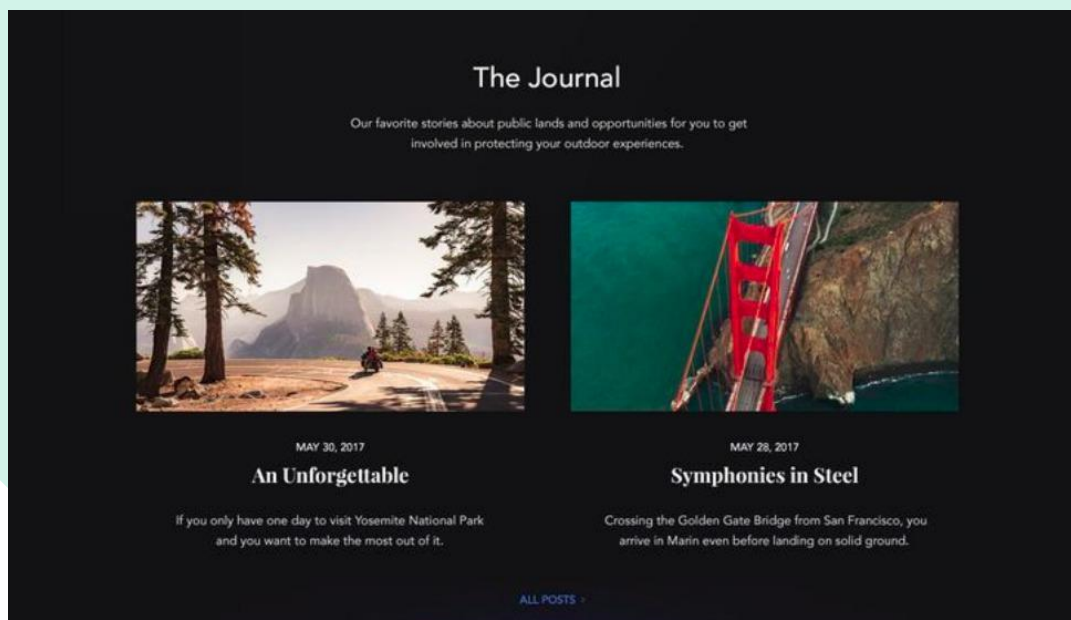


Esse componente deverá receber três parâmetros:

- **city**: cidade do ponto turístico
- **country**: país do ponto turístico
- **image**: url da imagem demonstrativa do ponto turístico

Esses parâmetros deverão ser utilizados na construção do componente do card.

4. **"The Journal"**: deverá retornar o elemento HTML com todas as informações do componente:

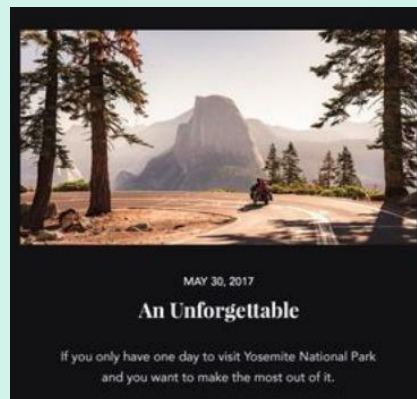


Para construir os cards de notícias, crie um array não exportado no módulo do componente que contenha a lista de notícias:

```
[  
  {  
    date: "", // data da notícia  
    title: "", // título da notícia  
    description: "", // descrição da notícia  
    image: "" // imagem da notícia  
  },  
  ...  
]
```

Esse array deverá ser utilizado para a criação dos cards.

Observe que os cards do componente possuem a mesma estrutura. Sendo assim, um componente específico pode ser implementado para a construção do card:



Esse componente deverá receber quatro parâmetros:

- **date:** data da notícia
- **title:** título da notícia
- **description:** descrição da notícia
- **image:** url da imagem da notícia

Esses parâmetros deverão ser utilizados na construção do componente do card.

5. **Footer:** deverá retornar o elemento HTML para a construção do footer da página:



6. **Page:** este componente deverá importar os componentes anteriores e retornar o elemento html da página completa.
7. Com todos os componentes construídos, importe o componente `page` no arquivo `index.js` da sua página e atribua-o à tag `<body>`, da seguinte forma:

```
document
  .querySelector("body")
  .appendChild(page);
```

Dica: Não configure as imagens do background da página dentro dos componentes.