Google olabi

<APIdoJavaScript>

{ API do JavaScript

API (*Application Programming Interface,* ou Interface de Programação de Aplicações em português) é um conjunto de regras compartilhadas por um serviço que determina formato e comandos acessíveis pelo aplicativo, bem como os dados que este serviço pode retornar em uma resposta. Ela funciona como uma camada entre o aplicativo e o serviço externo, uma forma de buscar ou enviar dados entre interfaces. A API permite que o aplicativo forneça ao usuário dados em tempo real obtidos no servidor e que você modifique ou adicione dados a outro ponto.

Considere o preparo do almoço em sua casa: para cozinhar arroz, você precisa primeiro colocá-lo em uma panela. Colocar o arroz diretamente no fogo seria, além de estranho, perigoso. Nesse caso, a API seria o arroz, o intermediário entre a comida e o cozimento.

{ Listagem de APIs nativas do JavaScript

Existem algumas APIs nativas do JavaScript que auxiliam no desenvolvimento:

- ***action:*** adiciona um botão à barra de ferramentas do navegador;
- ***alarms:*** agenda o código a ser executado no futuro, como *setTimeout() e setInterval(),* apesar dessas funções não funcionarem com páginas de fundo carregadas sob demanda;
- ***bookmarks:*** a API do *WebExtensions* (diferente da API padrão da área de transferência) permite que uma extensão manipule o sistema de marcação do navegador, marque páginas, recupere marcadores existentes, edite, remova e organize marcadores *.bookmarks. ;*
- ***browserAction:*** adiciona um botão à barra de ferramentas do navegador;
- ***browserSettings:*** permite que uma extensão modifique configurações globais do navegador. Cada propriedade representa um objeto com capacidade de modificar uma configuração *.types.BrowserSetting.* específica;
- ***browsingData:*** permite que extensões limpem dados acumulados enquanto o usuário estiver navegando;

• ***captivePortal:*** página web exibida no momento que o usuário se conecta a uma rede Wi-Fi, determina o estado do portal cativo da conexão. O usuário fornece informações ou age na página do portal cativo para obter acesso amplo a recursos da rede, como aceitar termos e condições ou fazer um pagamento;

• ***clipboard:*** API WebExtension que permite que uma extensão copie itens para a área de transferência do sistema. Atualmente, só suporta cópia de imagens, mas pretende apoiar a cópia de texto e HTML no futuro .*clipboardclipboard ;*

• ***commands:*** ouve os comandos de execução do usuário registrados pela tecla *manifest.json;*

• ***contentScripts:*** registra scripts de conteúdo e os insere em páginas que correspondam aos padrões de URL recebidos;

• ***contextualIdentidades:*** lista, cria, remove e atualiza identidades contextuais;

• ***cookies:*** permite que extensões obtenham e definam cookies, além de serem notificadas ao mudarem;

• ***devtools:*** permite que as extensões interajam com ferramentas de desenvolvedor do navegador para criar páginas, interagir com a janela inspecionada, ou inspecionar o uso de rede da página;

• ***dns:*** permite que uma extensão resolva nomes de domínio;

• ***downloads:*** permite que extensões interajam com o gerenciador de downloads do navegador. Você pode usar este módulo para baixar arquivos, cancelar, pausar, retomar downloads ou exibir arquivos baixados no gerenciador de arquivos;

• ***events:*** tipos comuns usados por APIs que despacham eventos;

• ***extension:*** utilitários relacionados à sua extensão, obtém URLs para pacotes de recursos, objeto “janela” para página e valores para várias configurações;

• ***extensionTypes:*** tipos comuns usados em outras APIs do WebExtension;

• ***find:*** encontra texto em uma página web e destaca partidas;

• ***history:*** interage com o histórico do navegador;

• ***i18n:*** funções que internacionalizam a extensão, essas APIs obtêm *strings* localizadas de arquivos embalados por ela, descobrem o idioma atual do navegador e o valor de seu cabeçalho *Accept-Language;*

• ***identity:*** obtém código de autorização ou *token* OAuth2, usado por uma extensão para acessar dados de usuários de serviço que suporte acesso ao OAuth2 (como Google ou Facebook);

• ***Idle:*** identifica quando o sistema do usuário está ocioso, bloqueado ou ativo;
• ***management:*** obtém informações sobre complementos instalados;
• ***menus:*** adiciona itens ao sistema de menu do navegador;
• ***notifications:*** exibe notificações ao usuário usando o mecanismo de notificação do sistema operacional subjacente. Como esta API usa notificações do sistema operacional, detalhes sobre como aparecem e se comportam podem diferir de acordo com o sistema operacional e configurações do usuário;
• ***omnibox:*** permite que extensões implementem comportamentos personalizados quando o usuário digita na barra de endereços do navegador;
• ***pageAction:*** ícone clicável dentro da barra de endereço do navegador;
• ***permissions:*** permite que extensões solicitem permissões extras no tempo de execução, depois de instaladas;
• ***pkcs11:*** extensão para enumerar módulos de segurança PKCS #11 e torná-los acessíveis ao navegador, como fontes de chaves e certificados *.pkcs11 ;*
• ***privacy:*** acessa e modifica configurações de navegador relacionadas à privacidade;
• ***proxy:*** *proxy* de solicitações da web. Você pode usar o ouvinte de eventos para interceptar solicitações e retornar um objeto que descreve se e como *proxy-los.proxy.onRequest ;*
• ***runtime:*** fornece informações sobre sua extensão e o ambiente no qual está funcionando;
• ***scripting:*** insere JavaScript e CSS em sites oferecendo duas abordagens para inserção de conteúdo;
• ***search:*** recupera mecanismos de busca e executa pesquisa por meio de mecanismo de busca específico;
• ***webNavigation:*** adiciona ouvintes de eventos para as várias etapas de uma navegação (quadro em transição de uma URL para outra no navegador, geralmente em resposta a uma ação do usuário, como clicar em um link ou inserir uma URL na barra de localização);
• ***webRequest:*** adiciona ouvintes do evento para as várias etapas de solicitação HTTP, como solicitações de *websocket on*. O ouvinte do evento recebe informações detalhadas sobre a solicitação e pode modificá-la ou cancelá-la. *ws://wss:// ;*
• ***windows:*** *i*nterage com as janelas do navegador, obtém informações sobre janelas abertas, abre, modifica ou fecha janelas. Também permite ouvir a janela abrir, fechar ou ativar eventos.

<Exercícios>

Procurando Pokémon

Vamos criar uma POKE-API para encontrar um Pokémon pelo número de id entre 1 e 893. Você vai precisar usar um HTML e um arquivo CSS já prontos.

HTML:

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PokéAPI</title>
    <link rel="stylesheet" href="pokeStyle.css">
    <link href="https://fonts.googleapis.com/css2?family=Hind+Siliguri:wght@300;400;500
        ;600;700&display=swap" rel="stylesheet">
</head>

<body>

    <img src="img/logoPoke.png" class="logo" alt="">

    <div class="content">
        <input class="enter" type="text" id="infoPoke" placeholder="Digite um número de
            1 a 893">
        <button class="btnBusca" id="button">Buscar</button>
    </div>

    <div class="muestraPoke">
        <img src="" id="img" class="pokemon" width="180" alt="">
        <h2 id="info" class="text-white"></h2>
        <p id="descp"></p>
    </div>
    <script src="PokeAPI.js"></script>
</body>
</html>
```

Classes no CSS:

```css
body {
    margin: 0;
    background-color: rgba(27, 23, 23);
    font-family: 'Hind Siliguri', sans-serif;
}
.logo{
    display: block;
    margin: auto;
    margin-top: 20px;
    width: 350px;
    height: 150px;
  }
```

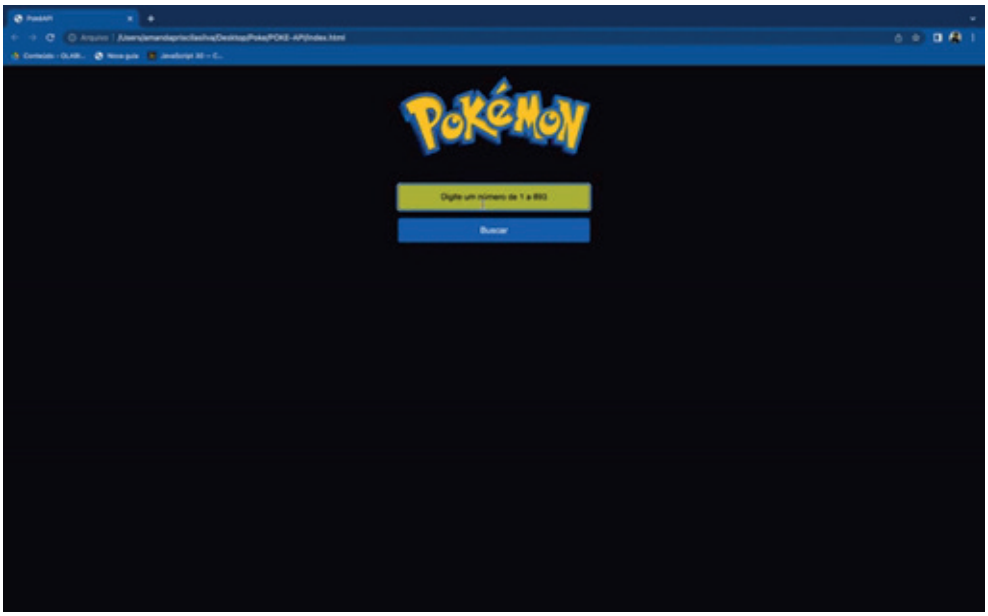
```
input {
    text-align: center;
    height: 45px;
    border-radius: 5px;
    background-color: rgba(255, 255, 0, 0.712);
    border: none;
    margin-top: 50px;
}
input::placeholder {
    color: black;
    font-size: 15px;
}
.btnBusca {
 margin-top: 15px;
    height: 45px;
    background-color: rgb(23, 94, 175);
    border-radius: 5px;
    border: none;
    font-size: 16px;
    color: white;
}
.btnBusca:hover {
    background-color:rgba(255, 255, 0, 0.712);
    transition: all 0.3s;
    color: black;
}
.content{
    text-align: center;
    margin: auto;
    width: 350px;
    display: flex;
    justify-content: center;
    align-content: center;
    flex-direction: column;
}
```

```
h2 {
    color: yellow;
    text-transform: capitalize;
    margin-bottom: 50px;
    font-size: 35px;
    margin-top: -10px;
}
.muestraPoke {
    text-align: center;
    margin: auto;
    margin-top: 20px;
    display: block;
    justify-content: center;
    align-content: center;
    flex-direction: column;
}
.redes {
    display: flex;
    justify-content: center;
    margin-bottom: 5px;
}
.redes img {
    width: 40px;
}
.redes img:hover {
    border-radius: 100%;
    background-size: cover;
    background-color: grey;
}
```

Dentro da pasta img, insira:

Seu desafio é criar o arquivo PokeAPI.js e o código mínimo para consumir a API https://pokeapi.co/api/v2/pokemon/${infoPoke}.

O resultado do projeto será uma tela inicial com campo no qual seja possível digitar um número que retorne um Pokémon.



Resposta: https://github.com/Yairix/POKE-API/blob/master/PokeAPI.js

<Referências>

JavaScript APIs - Mozilla | MDN
Trabalhando com APIs em JavaScript - GeeksforGeeks