Trabalho: Consumindo APIs e Json com JavaScript

Em uma mesma página HTML (com CSS e JS) mostre os seguintes exercícios. Separe os exercícios em seções bem distintas.

1- Deixe o arquivo países.json disponível no diretório do seu site. Acesse este arquivo Json via o comando **fetch** e gere um campo do tipo select com os nomes de todos países. Quando o usuário escolher um país mostre todos dados disponíveis deste país.
(Sobre o arquivo: JSON com lista de Países em pt-BR e seus respectivos Gentílicos, Siglas e Nome Internacional - Possui todos os países do Google Maps.
Obtido em: https://gist.github.com/jonasruth/61bde1fcf0893bd35eea)

2- Faça uma página de utilidade pública baseada em consultas ao https://brasilapi.com.br/ . Em todos casos (Nos 6 casos) utilize o comando fetch e proteja as chamadas com **try/catch** ou then/catch.

- Faça 3 consultas via programação e mostre os resultados na tela
- Faça 3 perguntas para o usuário via formulário. Para cada consulta mostre os resultados.
- Coloque para cada caso uma explicação para o usuário saber o que ele está vendo ou escolhendo

3- Faça uma página de utilidade pública baseada em consultas ao http://www.ipeadata.gov.br/api/ . Em todos casos utilize o comando fetch e proteja as chamadas com **try/catch** ou then/catch.

- Peça para o usuário escolher entre pelo menos 3 tipos de metadados. Para cada tipo de dado gere a correspondente tabela. Coloque a explicação completa dos dados e das colunas que serão mostradas (Estas informações estão no correspondente metadado e nos nomes das colunas)
- Dois alunos não podem utilizar os mesmos tipos de dados

Exemplos de URL:

- http://www.ipeadata.gov.br/api/odata4/Metadados/
- http://www.ipeadata.gov.br/api/odata4/Metadados('ABATE_ABPEAV')
- http://www.ipeadata.gov.br/api/odata4/Metadados('ABATE_ABPEAV')/Valores/

4- Desenvolva os códigos citados abaixo. Em todos casos proteja as chamadas com try/catch ou then/catch.

- Faça um botão que ao ser clicado acesse 3 APIs externas utilizando o comando **Promisse.any.** Mostrar o retorno da primeira promessa resolvida (o json) ou rejeitada.
- Faça um botão que ao ser clicado acesse 3 APIs externas utilizando o comando **Promisse.race.** Mostrar o retorno da primeira promessa resolvida (o json) ou rejeitada.
- Faça um botão que ao ser clicado acesse 3 APIs externas utilizando o comando **Promisse.all**. Acessar o retorno de Promisse.all e mostrar os retornos das 3 apis (o json). Mostrar o retorno de todas as promessas ou a mensagem de erro da última promessa rejeitada.

Em algum ponto da página (Pode ser em qualquer exercício anterior ou se for necessário pode ser criado um exercício novo):

- Utilize uma atribuição do tipo **destructuring**.
- Utilize o **map** em um objeto/array vindo de um fetch e mostre o resultado na tela
- Utilize o **filter** em um objeto/array vindo de um fetch e mostre o resultado na tela
- Utilize o **reduce** em um objeto/array vindo de um fetch e mostre o resultado na tela

Requisitos gerais:

- Coloque os códigos em uma página **HTML responsiva**
- A página deve passar pelo **validador da w3c** sem erros (https://validator.w3.org/).
- A página não deve apresentar erros pegos pelo **console do JavaScript**(JS) dos navegadores.
- Evite ou trate os erros das chamadas via fetch. Verifique e trate as situações em que as chamadas podem retornar erro. Também procure evitar demais erros que possam ser cometidos pelos usuários.

Mais APIs:
- https://deividfortuna.github.io/fipe/ Esta API mostra dados de preços de carros vendidos no Brasil.
- https://jsonplaceholder.typicode.com/guide/ é uma API com dados Fake para testes.
- Em https://swapi.dev/documentation tem uma documentação para uma api que conforme o endereço ela retorna a correspondente informação sobre o universo Star Wars (...é só procurar que tem api de tudo por ai….coloque no google “API Marvel” e veja o que acha…). Aqui vai uma lista de APIs interessantes https://devporai.com.br/9-apis-para-seu-novo-projeto/
- https://github.com/public-apis/public-apis lista de APIs.

O trabalho deve **ser entregue até 24/11/2023**. Para tanto deve-se enviar no SIGAA o arquivo com o código do trabalho. Se
Entre os dias **16/11 e 24/11** cada aluno deve apresentar um exercício funcionando para o professor como verificação de andamento. A falta desta apresentação, exercício não funcionando ou falta de conhecimento sobre o código pode acarretar em desconto de até 3pts no trabalho.
Não haverá apresentação final do trabalho.

Bom trabalho!!