Roteiro

JavaScript

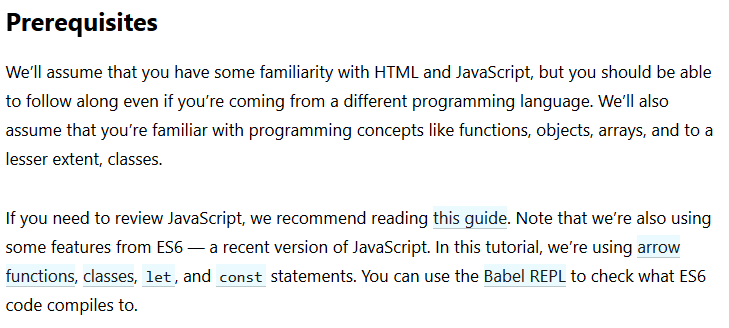
Tópicos

Opção 1

* DOM
* Tipos de dados
* Variáveis
* Operadores
* Estrutura de controles
* Objetos
* Arrays
* Funções
* Objetos customizados
* Closures
* Arrow functions
* Classes (propriedades, métodos e herança e entender o "this")
* let e const

Pré-Requisitos – React

<https://reactjs.org/tutorial/tutorial.html>



**Links**

<https://www.w3schools.com/js/default.asp>

<https://flaviocopes.com/javascript-introduction/>

<https://medium.com/trainingcenter/afinal-javascript-e-ecmascript-s%C3%A3o-a-mesma-coisa-498374abbc47>

<https://medium.freecodecamp.org/javascript-naming-conventions-dos-and-don-ts-99c0e2fdd78a>

**O que é JavaScript?**

Linguagem de programação



**Onde?**

Back-end, Mobile, Front-End, Banco de Dados

**Como?**

**Criar repositório GitHub – 3 sprint**

**2s2019-t2-sprint-3-frontend**

**Subir apresentação (somente do javascript) e pasta: ‘gufos.base’ (que contém somente o css e o html).**

**Apresentação – 40 minutos**

HTML -> Estrutura

CSS -> Estilização

JavaScript -> Comportamento

[https://cdn-images-1.medium.com/max/1600/0\*ILZZpnli\_R1FU3p4.gif](https://cdn-images-1.medium.com/max/1600/0*ILZZpnli_R1FU3p4.gif)

http-server

<https://imasters.com.br/desenvolvimento/escopos-em-javascript>

DOM

Representa o documento HTML na página.

Retomar o backend e quais endpoints que vocês irão construir.

Api/login

Api/categorias

C:\\_github\senai-dev-2s2019\sprint-3-frontend\backend.projeto\Senai.Gufos.WebApi\Senai.Gufos.WebApi

Abra o cmd e digite dotnet run.

Localhost:5000/Swagger

Fazê-los pensar quais passos deverão ser feitos para eles conseguirem enviar os dados do frontend para o backend.

**Início**

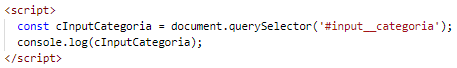
Manipulação do DOM e buscando a referência do botão.

categorias.dom.html



categorias.botao.html

Iniciar este exercício somente buscando a referência do botão anterior.



Fazer a alteração para criar um cadastrarCategoria() dentro do script.



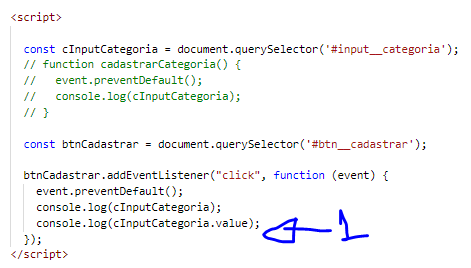
Evitar o comportamento padrão da página.



Colocar um evento para que seja escutado.



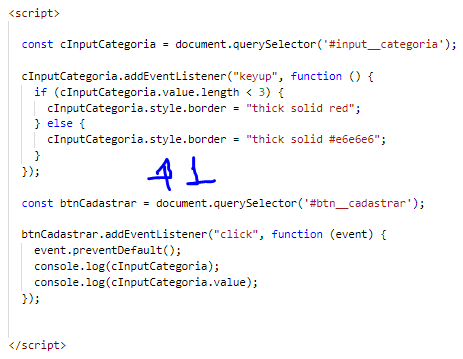
Para buscar o valor que o usuário digitou.



categorias.input.evento.html

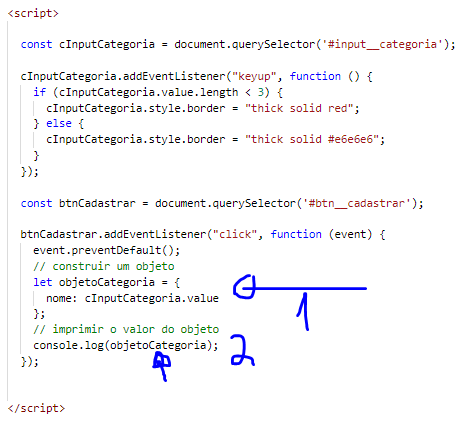
Validando o valor de input do usuário.

Observação: keyup não é obrigatório. Após, terá atividade com desafios para eles verem essa parte.

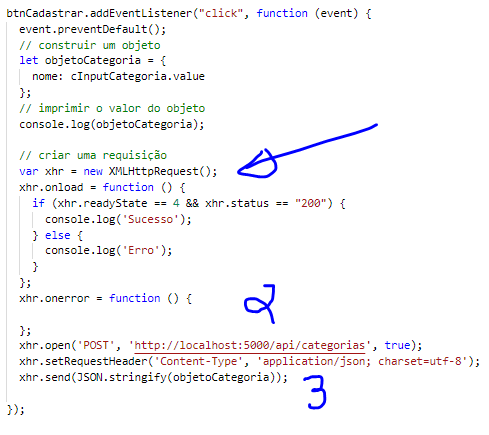


categorias.post.html

Construir o objeto que será enviado para à API.



Criar o evento que será feito para à API.



**Exercícios**

3.frontend.js.ex1.validacoes.docx – formulários de input, validações de campo

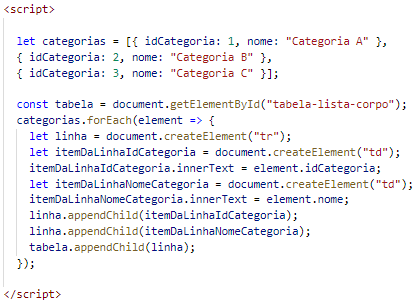
3.frontend.js.ex1.post.docx – sstop – api, post

categorias.html

Manipulação do DOM e declaração de variáveis.

**API**

Mostrar uma lista fixa na tabela – criando elementos e mostrando na tela.







Comunicando com a API.

<https://medium.com/beginners-guide-to-mobile-web-development/the-fetch-api-2c962591f5c>



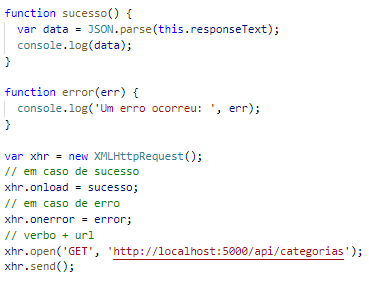
[**https://developers.google.com/web/fundamentals/primers/promises**](https://developers.google.com/web/fundamentals/primers/promises)

**XML é somente modelo caso eles queiram.**

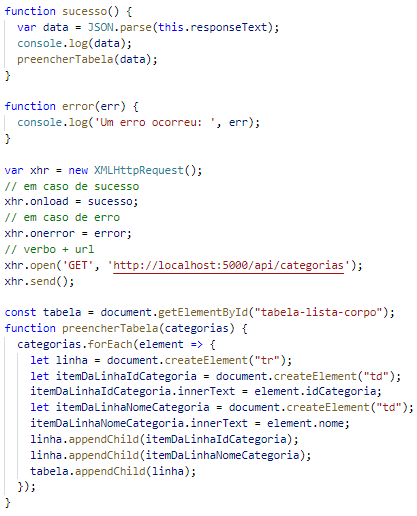
**XMLHttpRequest**



Adicionar as duas funções, de erro e sucesso.

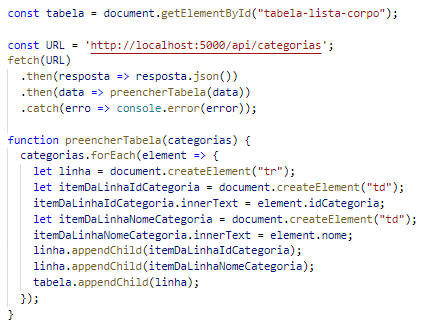


Criar uma função e atribuir o valor como parâmetro de entrada da função.



**FETCH**

Listando tipos de eventos utilizando a Fetch API.



<https://braziljs.org/blog/fetch-api-e-o-javascript/>

**Exercícios**

3.frontend.js.ex1.get.docx - sstop

3.frontend.js.ex4.get.docx - personagens

arrays

categorias.arrays.html







PUSH/POP



Funções pré-definidas



MAP



FILTER



REDUCE



**Exercícios**

3.frontend.js.ex2.docx

Extra

Cadastrando uma categoria com FETCH.

Realize o cadastro de um tipo de evento.



Construir o JSON.



Fazer o envio dos dados a API.

Com Fetch.

