Homeworks

Informações Gerais

Os slides das aulas podem ser encontrados na pasta **Slides** dentro do Google Drive.

Todos os exercícios devem ser entregues via GitHub. Cada exercício deve ser entregue no repositório do módulo correspondente. Em cada homework haverá o repositório correto que aquele exercício deve ser comitado.

É extremamente recomendável que ao final de cada aula/exercício você faça anotações dos principais pontos aprendidos.

**Exemplo:**

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **COMENTÁRIO** |
| <div class=”minha-classe”></div> | .minha-classe { } | Acesso classes no CSS via “.” |
| <div id=”meu-id”></div> | #meu-id { } | Acesso IDs no CSS via “#” |

**Dicas de Ouro:**

Otimize seu tempo priorizando as **partes mais fáceis** dos homeworks. Dessa forma você terá sensações de **avanço** e **bem-estar**. Caso esteja com dificuldade em algum problema em especifico, **não hesite em buscar ajuda!**

**01 - HTML & CSS**

Faça um fork e clone o seguinte repositório do GitHub: <https://github.com/herowaybrasil/heroway-html-css>

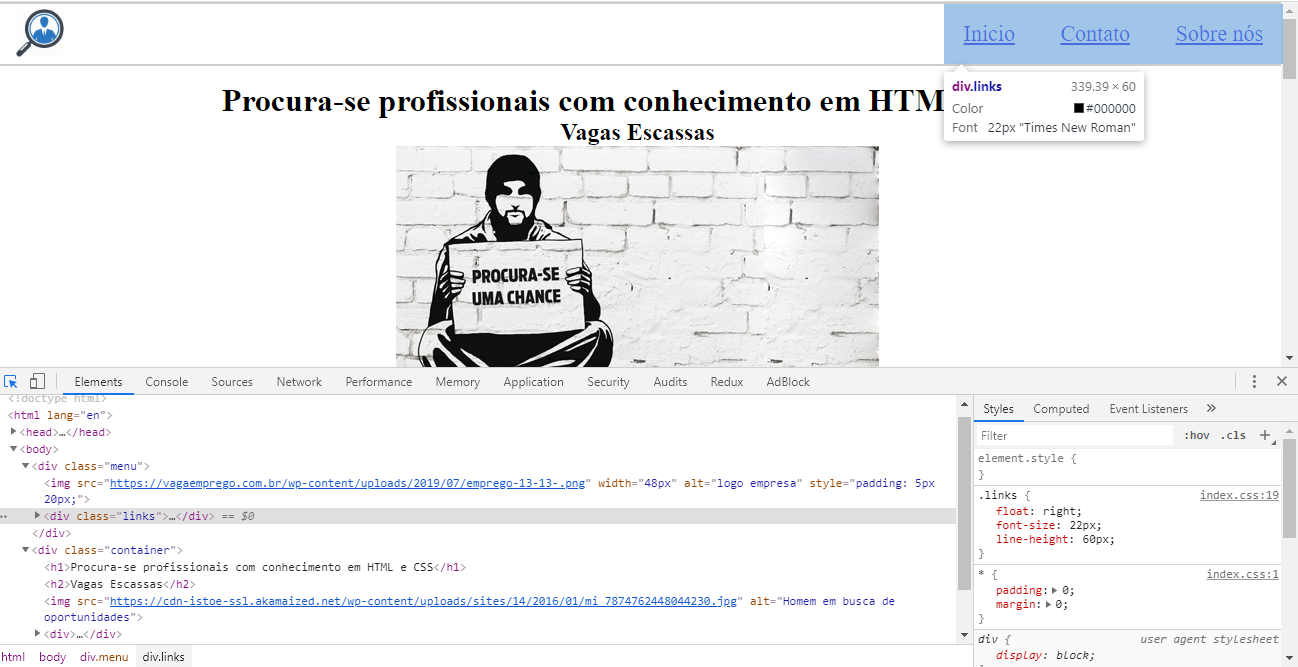
Abrir uma instância do VSCode e navegar até pasta ***heroway-html-css/homeworks/job-seeker*** e explorar os arquivos **index.html** e **index.css.**Praticar o autodidatismo procurando tags HTML e atributos CSS desconhecidos tentando entendê-los e trazer uma lista de dúvidas para a próxima aula (gravar essas dúvidas no seu arquivo pessoal de anotações).

Abrir outra instância do VSCode, navegar até o exercício **job-seeker** criado em aula e utilizar a criatividade para evoluir o mesmo.

**Inspiração:**



**Dica:** Você pode abrir o arquivo **index.html** utilizando o Google Chrome e brincar com CSS utilizando as ferramentas do desenvolvedor (F12).



**Entrega do exercício:** [Criar um novo repositório no GitHub](https://drive.google.com/open?id=1ZIe4imfg4dAEOMP4iKzv7ZqmHseAH5QI) chamado **heroway-html-css**, com os arquivos dentro de uma pasta chamada **job-seeker**. Não esqueça de enviar suas alterações para o GitHub (git push).

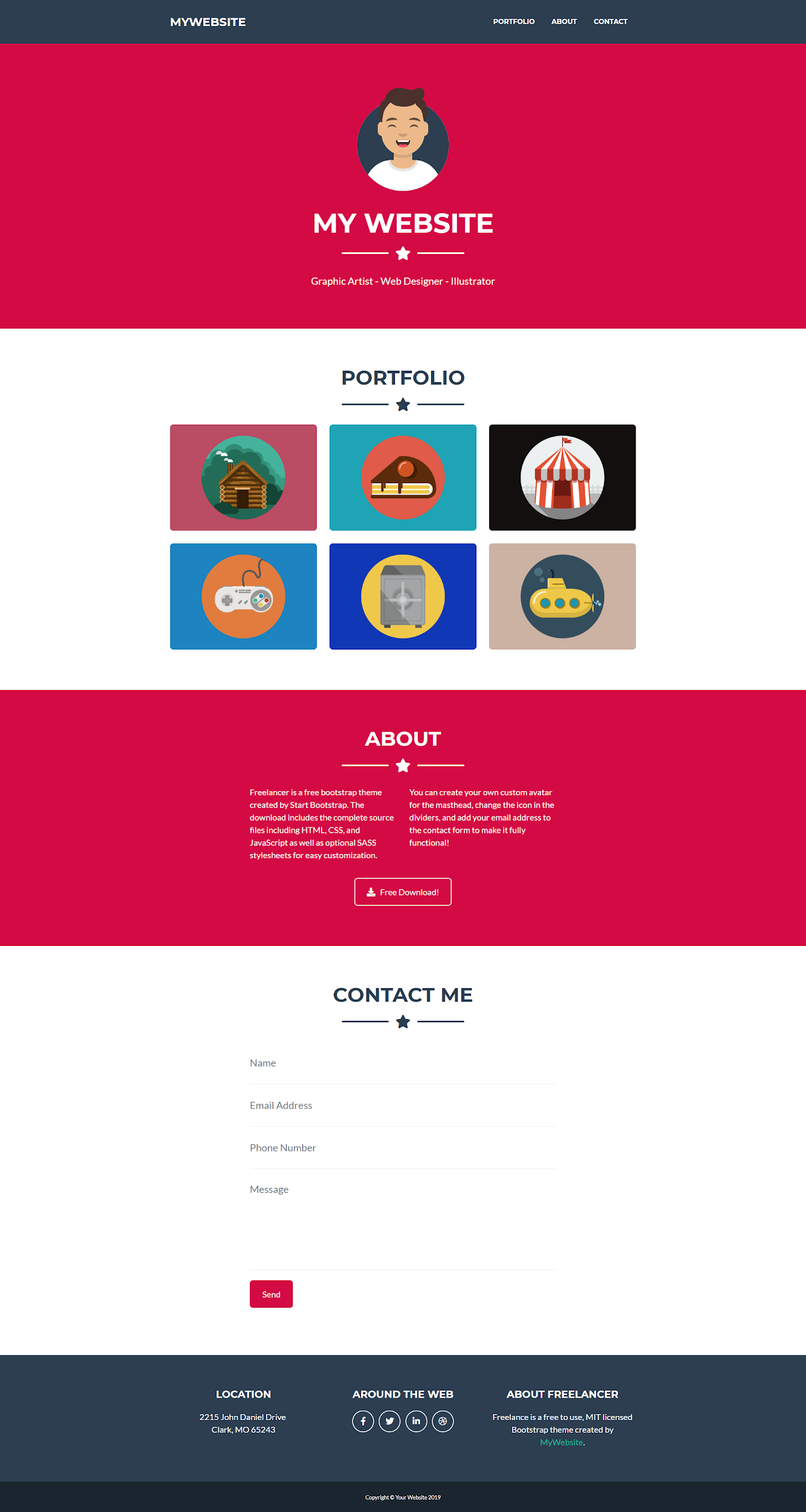
**Extra:** Resolver os exercícios 1 e 2 do site CSSBattle - <https://cssbattle.dev/play/1>

**02 - HTML & CSS**

Estudar flexbox completando todos os níveis do jogo [Flexbox Froggy](https://flexboxfroggy.com/).

Estudar CSS Positions através do site [Learn Layout](https://pt-br.learnlayout.com/position.html) ou [CSS Tricks](https://css-tricks.com/almanac/properties/p/position/).

Concluir (ou chegar o mais próximo possível) do desenvolvimento do layout **MyWebsite**.



**Importante:** A não ser que você tenha tempo,não gaste sua energia mudando o tamanho das fontes, criando um layout responsivo ou tentando desenvolver detalhes, como por exemplo:



**Atenção:** A seção **CONTACT ME** é **opcional**.  
**Atenção:** Com o intuito de trazer um ambiente de trabalho real, não disponibilizamos os assets necessários para compor o seguinte bloco:



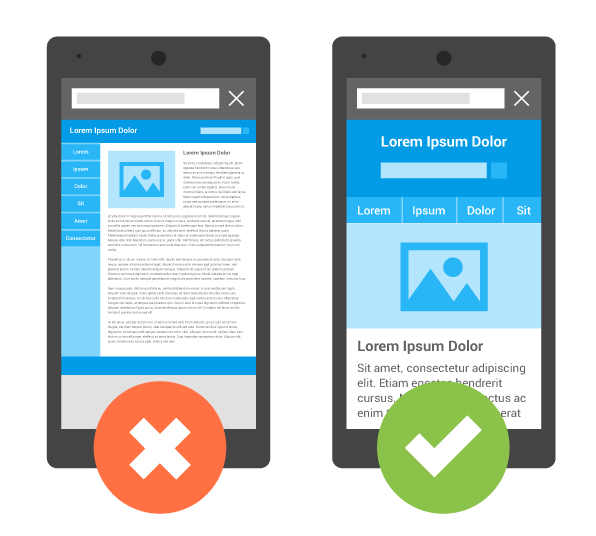
Sua missão é [encontrar esses assets na internet](https://fontawesome.com/) e inseri-los ao projeto. (**opcional**)

**Dica:** Você pode usar a extensão do Chrome [ColorZilla](https://chrome.google.com/webstore/detail/colorzilla/bhlhnicpbhignbdhedgjhgdocnmhomnp?hl=pt-BR) para capturar as cores do layout.  
**P.S:** Não esqueça de commitar (subir as alterações para o GitHub), ok? =)

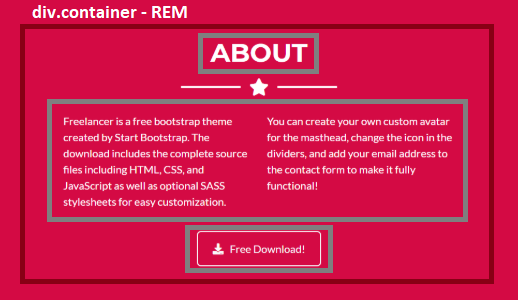
**03 - HTML & CSS**

Abrir uma instância do VSCode, navegar até pasta ***heroway-html-css/homeworks/spider-man*** e aplicar as seguintes alterações ao layout:

* Remova todos os elementos de largura (width) com tamanho fixo (px), substituindo-os por porcentagem (%).
* Transforme todos os textos em REM e EM.
* Transforme todas as imagens em porcentagem (%).
* Adicione Media Queries de forma que o layout consiga ser adaptado para Mobiles e Desktop.
* Transforme todas as margins em REM ou EM **(opcional)**



**Dica:** Pense nos trechos HTML como pequenos componentes, onde o container (box - vinho) será definido a utilizando REM e os demais font-sizes/margins/paddings (box - cinza) serão definidos utilizando EM. É importante lembrar que essa é apenas uma técnica e não deve ser considerada solução ou regra absoluta. Apenas com a prática e vivência você saberá qual é o melhor caminho a seguir para cada tipo de cenário.



**Leituras adicionais:**[Pixel vs REM vs EM](https://tableless.com.br/unidade-pixels-em-rem/) | [Dicas para criação de um Layout Responsivo](https://www.devmedia.com.br/responsive-design-dicas-para-tornar-seu-site-acessivel-em-qualquer-resolucao/28316)

**01 - Javascript**

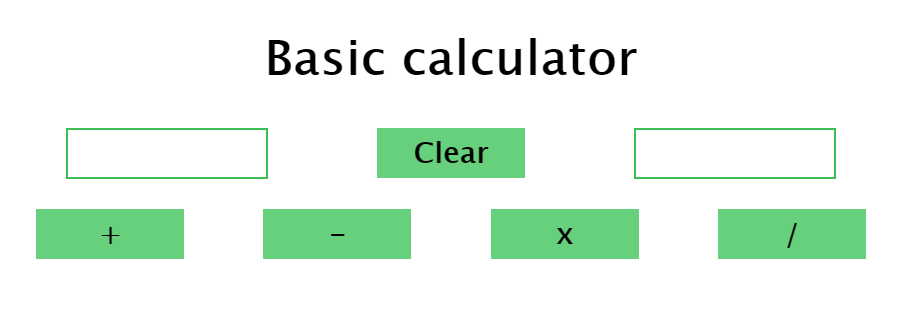
Faça um fork e clone o seguinte repositório do GitHub: <https://github.com/herowaybrasil/heroway-javascript>

Abrir uma instância do VSCode e navegar até pasta ***heroway-javascript/modules*** e explorar os conceitos abordados em aula.

Explore e habitue-se a linguagem Javascript. Crie variáveis (strings, numbers, objetos, arrays), funções, if/else, loops, etc.

Estude **muito** a função filter feita em aula (você pode acompanhar pela gravação da aula).

Navegar até pasta ***heroway-javascript/homeworks/calculator*** *e* criar uma calculadora funcional utilizando javascript.



**Regras:**

* Não permita divisões por zero.
* Não permita strings como valores.
* Exibir resultado na tag **p#resultado** contido no layout.
* Limpar campos quando o usuário clicar na tag **button#btn-clear**
* Melhorar o CSS e deixá-lo responsivo (**opcional - inspiração abaixo**)



**02 - Javascript**

Fixar o conteúdo passado em aulaEstude a teoria dos seguintes conceitos em Javascript:

* [Callbacks](https://medium.com/totvsdevelopers/entendendo-fun%C3%A7%C3%B5es-callback-em-javascript-7b500dc7fa22)
* Closures
  + [Link 1](https://medium.com/@stephanowallace/javascript-mas-afinal-o-que-s%C3%A3o-closures-4d67863ca9fc) | [Link 2](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Closures)
* Funções Puras & Programação funcional
  + [Link 1](https://medium.com/@iagoangelimc/programa%C3%A7%C3%A3o-funcional-com-javascript-fun%C3%A7%C3%B5es-puras-c0403410cf6d) | [Link 2](https://bognarjunior.wordpress.com/2018/02/10/funcoes-puras-em-javascript-programacao-funcional/)
* Funções nativas dos Arrays (Arrays Prototype)
  + [Push](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/push) | [Pop](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/pop) | [Find](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/find) | [Filter](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/filter) | [Map](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/map) | [Reducer](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/reducer)

**03 - Javascript**

Coloque todo o conhecimento teórico adquirido em prática:

* Faça uma chamada HTTP para o seguinte endpoint:<https://akabab.github.io/superhero-api/api/all.json>
* Filtre apenas os heróis com a primeira letra do seu nome.
* Com o resultado da operação anterior, mapeie apenas os heróis que pertencem a DC Comics.

Estude e desenvolva o resultado do arquivo ***heroway-javascript/modules/api/free-nudes.js***

Se prepare para a aula de React lendo o [e-book Create React App](https://drive.google.com/open?id=1hCPq_0jUess_rA4D5UI9PCg5a2KKEx96) disponibilizado no Google Drive.

Navegue até a pasta onde você armazena todos os módulos das aulas. Abra o terminal (CMD/PowerShell/Bash) e execute o seguinte comando:

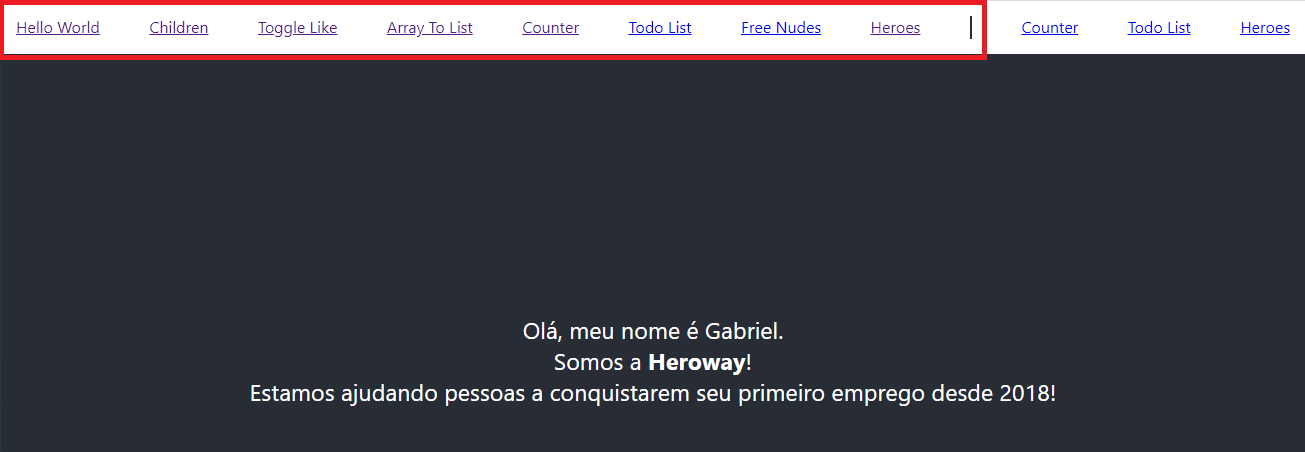
* npm install -g create-react-app
* create-react-app heroway-react --template typescript

**01 - React**

Faça um fork e clone o seguinte repositório do GitHub: <https://github.com/herowaybrasil/heroway-react>

Abrir uma instância do VSCode e navegar até pasta ***heroway-react*** rode o comando **npm start.** Explore - principalmente os componentes desenvolvidos em aula - dentro da pasta ***heroway-react/src/components/withoutRedux***.

Uma vez que a aplicação tenha iniciado, você pode navegar pelos componentes pelo menu destacado em vermelho:



Após explorar e os componentes, é a hora de colocar a mão na massa!

Abra outra instância do VSCode e abra o projeto criado em aula. A sua missão é desenvolver os seguintes componentes:

**<StyledButton />**

interface IProps {  
 backgroundColor: string;  
 text: string;  
}

<StyledButton backgroundColor={"red"} text={"Clique aqui..."} />

Esse componente deve renderizar um botão respeitando as props **backgroundColor** e **text.**

**<BankAccount />**

interface IProps {  
 initialValue: number;  
}

function BankAccount(props: IProps) {  
 const [moneyInBankAccount, updateMoneyInBankAccount] = React.useState(props.initialValue);

return (  
 <div>  
 <input type="text" placeholder="Depositar valor..." />  
 <div>{moneyInBankAccount}</div>  
 </div>  
 );  
}

<BankAccount initialValue={100} />

Esse componente deve renderizar o dinheiro guardado no banco. Além disso, após o evento de **onChange** no input, o valor do input deve ser somado ao da conta.

**Dica:** Lembre-se que os eventos (onClick, onChange, etc) emitem uma parâmetro chamada **event** na função de callback. É através dele que você será capaz de capturar o valor digitado no input.

**Dica 2:** Você pode acessar a documentação do [React](https://pt-br.reactjs.org/). Além disso, O Google é seu melhor amigo! =)

**Leitura obrigatória:**  
[Pensando em React](https://pt-br.reactjs.org/docs/thinking-in-react.html)

**02 - React**

Crie um novo projeto React com Typescript chamado **heroway-facebook-react**.

* create-react-app heroway-facebook-react --template typescript

Clone o seguinte repositório do GitHub:  
<https://github.com/herowaybrasil/heroway-facebook-raw>

Abrir uma instância do VSCode e navegar até pasta ***heroway-facebook-raw*.**Abrir outra instância do VSCode e navegar até pasta ***heroway-facebook-react*.**

Sua missão é componentizar a aplicação **heroway-facebook-react** com baseado no conteúdo HTML e CSS disponibilizado no projeto **heroway-facebook-raw.**

**Importante:** O código da página de login está comentado na aplicação **heroway-facebook-raw**. Você deve criar um componente para ele da mesma forma.

Após a separação dos componentes, você deve criar um sistema de rotas utilizando a biblioteca [React Router](https://reacttraining.com/react-router/web/guides/quick-start) onde a aplicação terá duas rotas.

* **/login -** Deve renderizar o componente <Login />
* **/feed -** Deve renderizar o componente <Feed />

**Dica**: Já separamos os componentes do projeto **heroway-facebook-raw** para você através de comentários.