OBS: Para contextualizar melhor o meu progresso, achei que seria muito legal retratar a parte do CSS adiante visualmente em conjunto com o texto, então pra ilustrar melhor as fazes, eu trouxe algumas imagens para o formulário. Deixei também um PDF pra ficar mais legível lá no projeto do github, mas se quiser ver por aqui tudo bem, deixei o link das imagens também ; )

Na sessão do <head> eu procurei utilizar muito como referência o meu código anterior por que lembro que fiz uma pergunta para o professor de quais eram os códigos fundamentais para páginas que deveriam ser utilizadas no <meta>.

Também fazendo consultas no código já feito antes, eu lembrei que tanto o título, tanto a referência ao CSS e JS também deveriam ser colocadas na sessão do <head>.

Já adentrando na sessão do body, percebi algo bem diferente do código que foi passado na apostila. Diferente do conteúdo ensinado, o desafio não tem uma imagem padrão que você passa por ela e depois pega as imagens da API ou um botão. As imagens da API devem ser apresentadas assim que abrir a página, então vou precisar ver outra forma de fazer isso.

A forma com que eu imaginei fazendo os quatro blocos de personagens com informação foi criando 4 <section> e depois personalizando todas no CSS.

Achei melhor deixar todas as <section> sem classe específica para ficar mais fácil na hora de padronizar todas elas e ficar mais organizado e limpo o código do CSS. Adicionei também comentários em cima de cada uma pra poder me organizar melhor.

Eu parei e revisei melhor o que eu teria que fazer no futuro e pensei que se eu não desse uma "class" específica para cada uma das <section>, na hora de programar o JS eu teria muitos problemas na hora de codar e mudar um único elemento sem alterar todos com o mesmo nome ou que pertencesse à mesma <section>, então resolvi descartar a ideia de deixar tudo padronizado genericamente e só reutilizar o código do CSS de uma, para todas as outras. O intuito de deixar tudo sem classe específica, era deixar mais limpo e encurtado o código CSS.

Adentrando na parte do CSS eu já comecei fazendo cagada. Eu fiquei por mais de 30 minutos tentando editar as minhas imagens que coloquei como "placeholder" pra poder editar e configurar o meu site sem sucesso. Fiquei comparando o meu CSS que tava fazendo atualmente com o da apostila sem sucesso pra encontrar o erro. Tudo isso, só pra depois perceber que eu utilizei <script src="syle.css" defer></script> ao invés de <link rel="stylesheet" href="style.css"> na área do HTML.

Com a parte do CSS solucionada, comecei a editar os elementos da página. Mais especificamente, o alinhamento delas na tela, espaçamento entre uma e outra, o tamanho delas, e a posição específica na tela.

Eu olhei no grupo de WhatsApp e algumas pessoas estavam usando técnicas avançadas de posicionamento que eu não entendi então resolvi ficar com posicionamento fixo que é como sei trabalhar.

Depois de posicionar elas no local permanente, eu comecei a trabalhar na parte do texto. O nome dos personagens e a posição deles em relação à foto. Durante esse processo, percebi que meu arquivo CSS tava muito bagunçado, então tomei a liberdade de comentar certas partes pra saber quais partes alteravam o que.

Depois disso, passei duas horas tentando aprender como utilizar o background e aprendendo o que a tag <div> fazia. Não consegui e parei pra descansar.

Após retornar para o projeto novamente e tentar por mais 30 minutos como resolver a parte do CSS de maneira que me agradasse, continuei não conseguindo mesmo procurando na internet. Resolvi então seguir em frente pra parte do JS e voltar outra hora e tentar achar uma solução.

JavaScript é muita questão de lógica então precisei ficar revisando o meu código anterior pra entender o que eu tinha feito e utilizar como base pra moldar o meu a partir daquilo (OBS:Essa é minha primeira vez trabalhando com JavaScript).

JavaScript se revelou ser o meu pior pesadelo na hora de lidar com todo esse projeto, atualmente é Domingo e estou desde Sábado de tarde tentando implementar uma forma de fazer o API alterar as informações no Site sem nenhum sucesso. Resolvi então pedir ajuda de um colega meu programador na esperança que ele consiga me encaminhar na direção certa.

Minha maior fonte de frustração até esse ponto é o fato de eu não estar conseguindo fazer com que as minhas "const" encontrem o caminho correto até as "class" e "id" e estou pensando seriamente em recomeçar o código todo do 0 a essa altura por não conseguir encontrar o erro.

Eu resolvi não começar do 0, porém eu pensei que fazer todos os 4 nomes e imagens ao mesmo tempo estava me prejudicando na hora de solucionar o problema, então resolvi apagar três e trabalhar somente com uma única imagem template e nome template de "Personagem1" e tentar resolver a partir dai.

Depois de pelo menos 5 horas procurando ajuda com amigos programadores e ajuda no grupo de WhatsApp criado pra essa turma do Estrelas Fora da Caixa, um rapaz me ajudou a identificar o erro."Content-Type": 'application/json' tava como "Content Type": 'application/json' e isso não deixava funcionar. A sensação de alivio ao encontrar esse erro é muito boa por que já estava começando a entrar em desespero sem saber o que tava errado. A essa altura eu já tinha mudado 200% meu código pra encontrar o erro, que eu pensava ser de lógica, quando na verdade era um erro de sintaxe.

Com a primeira e imagem funcionando, parti pra tentar fazer a segunda imagem. Eu sabia que se conseguisse fazer duas imagens funcionar era só questão de copiar o que eu fiz com as duas e repetir mais quatro vezes.

Eu comecei tentando colocar a segunda imagem junto com a primeira dentro do "data" recebido da "promise" e descobri rapidamente que fazendo isso, repetia tanto o nome quanto a imagem.

Depois disso, descobri que colocar mais de uma imagem seria um desafio por elas não terem nenhum tipo de "id" ou "class" no meu html, então procurei como dar nomenclatura para elas no meu "html" e conseguir indicar o caminho correto para elas utilizando minhas "const". Após uma breve lida na documentação da tag <img>, descobri que da sim pra utilizar "id" nela e receber ela no JS utilizando "getElementById".

Depois disso criei todas as 4 <section> para os meus personagens e "const" necessárias para fazer tudo isso funcionar.

Com o meu JS funcionando corretamente em conjunto com o "HTML", só me restava descobrir como eu iria fazer pra conseguir exibir 4 imagens e nomes diferentes utilizando a API.

Pelos próximos 30 minutos eu passei procurando por uma solução de como fazer e lembrei que uma pessoa do grupo havia dito brevemente que conseguiu realizar a transição utilizando "foreach", como eu não sabia como utilizar nenhum tipo de array em JS, procurei tutoriais na internet para solucionar o problema. Logo após eu tentar implementar o foreach, descobri que eu estava enfrentando um problema onde eu não sabia como fazer um laço "foreach" no funcionar JavaScript e estava tendo dificuldades na hora de entender a lógica também. Resolvi então tentar usar o laço "for" que é como eu sabia utilizar em outras linguagens e pelo pesquisei ele também funcionava da mesma forma em JS.

Eu então segui criando uma "var = 0;" pra armazenar o valor de onde o laço começaria e criei 2 arrays, um de "nomes" e o outro de "imagems", ambos contem "nome1, image1" até o "nome4, image4". Junto com o array, eu também criei o laço "for(i = 0; i < 4; i++){}" que serve como instrução para alimentar os dados para o array "nomes" e "imagems" com base na função "PegarPersonagem".

Com os arrays e laço feitos, eu me deparei com outro problema. Toda vez que ele chamava a função "PegarPersonagem" ele alimentava informações a mais e acabava ficando errado. Quando isso acontecia, ele mostrava o nome de outro personagem e não o da foto, porém o que acontecia é que ele tava preenchendo o array múltiplas vezes e isso causava com que o nome não ficasse o mesmo. Descobri isso ao inspecionar elemento da página e abrir o <body> do navegador e encontrar que havia um nome sendo preenchido entre o "<img>[aqui]</img>".

Para solucionar esse problema, eu utilizei um index "[i]" na frente de cada array do meu "for" pra ele verificar a posição em que ele está e não preencher mais de uma vez o array e terminar quando chegar no 4, deixando o "for" da seguinte forma: "for(i = 0; i < 4; i++){PegarPersonagem(nomes[i], imagems[i]);}".

Com o meu laço funcionando, eu precisava atribuir um parâmetro pra minha função "PegarPersonagem" pra que ela conseguisse alimentar o nome e a imagem quando ela fosse invocada pelo laço, então utilizei "NomeAPI e ImagemAPI" como parâmetro e na hora que chegasse pra receber as informações do API através do "data" que é recebido, é só eu utilizar "NomeAPI.innerHTML = data.name;" e "ImagemAPI.src = data.image;" e assim, toda vez que a função fosse chamada ele alimentaria os parâmetros através dessa linha de código =).

Após finalizar todo esse processo (que demorou pelo menos umas 6 horas e meia de troubleshooting e mudança de código) resolvi dar inicio à sessão do CSS na manhã seguinte.

Desse ponto adiante, eu achei que seria uma boa ideia dar um pouco de contexto visual do meu progresso pra ficar um pouco mais interativo o relatório e não só ter o produto final com um bando de texto acompanhado.

O meu site estava 100% placeholder a essa altura, onde eu tinha 4 fotos aleatórias de 4 personagens literalmente escrito "Personagem1" ao 4 do lado por que decidi que a parte de lógica do JS seria mais importante e mais difícil do que o CSS então foi a minha prioridade até então, ficando assim: <https://i.imgur.com/hPOCHlg.jpg>.

Após concluir a parte do meu JS com toda a dificuldade do mundo, ele ficou com essa aparência: <https://i.imgur.com/UGa2NFX.png>.

Só que todos os elementos do meu site estão com uma posição fixa, então se eu, ou qualquer outra pessoa, entrar nele e tirar o zoom da página, ele ficaria assim: <https://i.imgur.com/o1pjov7.png>, o que é uma péssima noticia, e o meu principal foco daqui em diante vai ser em como resolver isso.

Meu principal foco adentrando a sessão do CSS é tentar resolver o posicionamento das imagens e nomes dos personagens. Precisarei estudar melhor a documentação do CSS para saber como vou fazer isso.

Olhando o meu código do CSS, reparei que removendo a posição fixa que eu havia colocado e alterando o "position" de "absolute" pra "relative" já solucionava 80% dos meus problemas de posicionamento e a página já não tinha mais dificuldades comigo dando zoom ou tirando ele, deixando a página assim: <https://i.imgur.com/RFqs20q.png>.

Após solucionar o problema de posicionamento da página, eu comecei a olhar exemplos de bordas e cores que combinavam, como eu sempre gostei de como branco e preto/cinza combinam, acabei deixando a borda branca com o fundo preto. Pra ficar um pouco estiloso, resolvi deixar um rosa fluorescente como a cor da fonte, deixando o resultado assim: <https://i.imgur.com/ffOyZmu.png>.

Com um esquema de cores pronto, eu decidi ajustar a fonte. Atualmente a fonte não tem estilo algum, então resolvi colocar a fonte "impact" e centralizar ela junto com as imagens. (<https://i.imgur.com/VVJ7Fjb.png>)

Agora que o meu site já tem uma identidade própria, só me resta trabalhar a estética em cima disso e finalizar com pequenos detalhes.

O primeiro problema que comecei enfrentando com o CSS é um onde se o texto for muito grande a imagem acaba se desalinhando (<https://i.imgur.com/mfyQaR7.png>)), ficando bem feio o resultado e eu já tentei diversas formas de consertar isso porém sem sucesso. Por enquanto vou deixar assim e revisar esse detalhe depois.

Como um dos objetivos do desafio era atualizar a página, achei que seria uma boa ideia adicionar um botão fixo na tela para poder reiniciar os elementos da página quando for necessário. Isso tornaria mais intuitivo o processo de dar "refresh" na página.

Depois de 3 horas trabalhando na parte do CSS, consegui achar uma solução para o problema de desalinhamento com a imagem. Era só eu delimitar a borda do texto pra que ela não conseguisse ficar maior que a imagem.

Eu também resolvi dar uma personalizada melhor no site e adicionar mais alguns conteúdos na hora de receber informações da API e ficar mais apresentável, eu adicionei a localização atual do personagem dentro da série, assim como o seu status de "vivo" ou "morto", utilizando mais outros dois "array" pra realizar essa função. Adicionei também um padding que daria uma impressão melhor sobre o site e ficar parecendo menos a tela de seleção de perfil da Netflix e ficar mais bonito. (<https://i.imgur.com/IyHzFFK.png> somente com a localização do personagem por que esqueci de tirar foto com o resto)

Uma outra coisa que eu também fiz durante todo esse processo, foi alterar a forma com que o meu JS recebe e alimenta informações da API. Ao invés dele simplesmente jogar os dados da API no HTML, eu resolvi filtrar ele passando por uma tag de <span> dentro do JS alterando somente o nome pra poder diferenciar as informações na hora de mostrar no site. Como toque final, eu reorganizei todo o meu código CSS pra ficar melhor estruturado, percebi que poderia simplificar tudo como eu queria desde o início, então tomei a oportunidade pra fazer isso. Também aproveitei e coloquei um pouco de borda lateral nas letras e finalizei o dia. (<https://i.imgur.com/2JRnXsJ.png>)

Na manhã de terça feira, eu consegui resolver o ultimo problema que eu realmente estava tendo. As minhas imagens elas não estavam ficando alinhadas umas com as outras, depois de muito procurar, descobri que era o meu "height" que tava quebrando meu código, então eu alterei ele pra um tamanho que me agradasse com o alinhamento das imagens e adicionei um pouco de "padding" pra dar o resultado final do site que é possível ver hoje assim que abrir ele (<https://i.imgur.com/fdBspLI.png>).

Para finalizar esse relatório, gostaria de dizer que sou muito grata pela oportunidade que a ZUP me deu de me desafiar dessa forma. À alguns meses atrás eu jamais imaginaria que eu seria capaz de evoluir tanto então pouco período de tempo, foi muito divertida e educativa toda a jornada e espero poder participar de muitas outras. Obrigada! :D