

PROJETO - JOGO DESAFIO

Importante:

Fazer projeto em JavaScript, utilizando o visual studio code, entregar no github. O projeto será feito em grupo, uma pessoa deverá enviar o e-mail com a entrega do projeto para os e-mails claudiany.martins@redecidada.org.br e projetosrecife@redecidada.org.br.

No e-mail, deverá ter os nomes das pessoas da equipe que fez o projeto e o link do github com o projeto feito por cada pessoa. **Todas as pessoas do grupo deverão fazer o projeto.**

A entrega deverá ser feita até dia 08/06/2022, as 23:59h.

Haverá uma apresentação do código no dia 09/06/2022 de no máximo 5 minutos, treinem a apresentação para não passar o tempo.

Projeto

Neste desafio, praticaremos o que vimos nas nossas aulas para que você possa aprimorar suas habilidades e se sentir confiante em levá-las ao mundo real.

Pedra, Papel ou Tesoura

Rock-Paper-Scissor é um jogo clássico para dois jogadores. Cada jogador escolhe pedra, papel ou tesoura. Os itens são comparados e qualquer jogador que escolher o item mais poderoso ganha.

Os possíveis resultados são:

- Pedra destrói tesouras. (*Pedra ganha da Tesoura*)
- Tesoura corta papel. (*Tesoura ganha do papel*)
- Papel embrulha pedra. (*Papel ganha da pedra*)
- Se houver empate, o jogo termina empatado.

Nosso algoritmo, é dividido em quatro partes:

1. O jogador usuário, vai escolher uma opção.
2. O computador, vai escolher aleatoriamente uma opção.
3. Compare as duas opções e determine um vencedor.

4. Inicie o programa e exiba os resultados.

Vamos fazer esse jogo passo a passo, a ideia é você ir seguindo as instruções abaixo e ir construindo o jogo e o melhor entendendo cada parte do código.

Sugiro você ficar com uma tela do editor de código aberto, junto com o do console para ir testando e ver funcionando.

INSTRUÇÕES

1. O usuário deve escolher 'pedra', 'papel' ou 'tesoura' quando o jogo começar.

Usando **const** na variável e a função tipo arrow, crie uma função com o nome **getUserChoice** que use um único parâmetro **userInput**.

```
const getUserChoice = (userInput) => {  
  }  
}
```

2. Como um usuário pode passar um valor, como 'Rock' ou 'rock' ou 'ROCK' com diferentes letras maiúsculas, comece utilizando a função do JavaScript **toLowerCase()** para tornar todos os valores passados em **userInput** em letras minúsculas.

=> Lembrando que o código deve estar dentro da função **getUserChoice()**

3. Quando o usuário escolher uma das opções (rock, paper ou scissor), verifique também se o usuário digitou uma opção válida: rock, paper ou scissor.

Dentro da função **getUserChoice()**, escreva uma declaração **if/else** ou **switch** que garanta que o **userInput** seja **'rock'**, **'paper'** ou **'scissors'**.

Se isso acontecer, então utilize o **return** para devolver o valor de **userInput**.

Caso contrário, use `console.log` para imprimir uma mensagem informando que a opção informada não está correta, escolha uma mensagem que faça sentido e mostre ela no console.

4. Teste a função chamando-a com entrada válida e inválida e imprimindo os resultados no console. Você pode comentar os testes quando souber que sua função funciona.
5. Agora precisamos que o computador faça uma escolha.
 - Crie uma nova função chamada `getComputerChoice` sem parâmetros de entrada.
 - Dentro de sua função, utilizar `Math.random()` e `Math.floor()` para um número inteiro entre 0 e 2, você deverá escolher qual número representa cada item. *Mais informações sobre os métodos [Math.random](#) e [floor](#).*
 - Em seguida, dependendo do número, deverá devolver se o valor é `'rock'`, `'paper'` ou `'scissors'`.
 - Em seguida, você pode utilizar o `if/else` ou `switch` para retornar escolha do computador.
6. Teste a função chamando-a várias vezes e imprimindo os resultados no console. Você pode comentar os testes quando souber que sua função funciona.
7. Agora é hora de determinar um vencedor.
 - Crie uma função nomeada `determineWinner` que use dois parâmetros com os nomes `userChoice` e `computerChoice`.
 - Esta função irá comparar as duas opções jogadas e, em seguida, verificar o que está sendo retornado pelo `return` para saber se o jogador humano vai vencer, perder ou empatar.
8. Vamos lidar primeiro com a condição de empate.

- Dentro da função `determineWinner()`, verifique se o parâmetro `userChoice` é igual ao parâmetro `computerChoice`. Se assim for, deverá retornar uma mensagem que o jogo foi empatado.

9. Se o jogo não estiver empatado, você precisará determinar um vencedor.

- Utilize as regras do jogo:
Pedra destrói tesouras. (*Pedra ganha da Tesoura*)
Tesoura corta papel. (*Tesoura ganha do papel*)
Papel embrulha pedra. (*Papel ganha da pedra*)
Se houver empate, o jogo termina empatado.
- Verifique as escolhas do computador e do ser humano para determinar quem ganhou.
- Não se esqueça de testar sua função!

10. Após escrever o código e testar, você precisa iniciar o jogo e registrar os resultados.

- Crie uma função chamada `playGame`.
- Dentro da função `playGame()`, crie uma variável chamada `userChoice` que recebe o resultado da chamada `getUserChoice()`.
- Crie outra variável chamada `computerChoice` e que recebe o resultado da chamada `getComputerChoice()`.
- Use o `console.log` para imprimir os valores das variáveis no console.

11. Finalmente, vamos determinar quem ganhou.

- Dentro da função `playGame()` chame a função `determineWinner()`.
- Passe as variáveis `userChoice` e `computerChoice` como seus parâmetros.
- Coloque essa chamada de função dentro de uma instrução `console.log()` para poder ver o resultado.
- Então, para iniciar o jogo, chame a função `playGame()` na última linha do seu programa.