

Algoritmo: Pedra, Papel e Tesoura

Versão: Instruções passo a passo (sem código) — pronto para implementação

Preparação

- Crie um arquivo HTML simples e vincule (ou insira) um `<script>` para o JavaScript.
- Teste abrindo a página e escrevendo uma mensagem de teste no console (ex.: "Hello World") para garantir que o JS está rodando.

Função: obter escolha do computador

- Use as palavras: "pedra", "papel", "tesoura" (minúsculas).
- Gere um número aleatório entre 0 e 2 (inclusive).
- Associe 0 → pedra, 1 → papel, 2 → tesoura e retorne a palavra correspondente.
- Teste chamando a rotina várias vezes e verifique no console que as três opções aparecem aleatoriamente.

Função: obter escolha do humano

- Mostre uma caixa de entrada para o usuário digitar a jogada (ex.: prompt) e capture o texto.
- Converta o texto para minúsculas e remova espaços extras (trim).
- Retorne a string normalizada.
- Teste digitando variações como "PAPEL" ou " pedra " e confirme que retorna 'papel' e 'pedra' corretamente.

Função: jogar uma rodada (decidir vencedor)

- Recebe duas entradas: humanChoice e computerChoice.
- Se escolhas forem iguais → Empate.
- Caso contrário, verifique as combinações em que o computador vence:
 - computador = pedra e humano = tesoura → computador vence
 - computador = papel e humano = pedra → computador vence
 - computador = tesoura e humano = papel → computador vence
- Se não for nenhum dos casos acima → humano vence.
- Retorne um rótulo do resultado (ex.: 'Empate', 'Você venceu', 'Computador venceu').
- Teste com entradas fixas para conferir os resultados.

Lógica do jogo completo (5 rodadas)

- Inicialize humanScore = 0 e computerScore = 0.
- Repita 5 vezes:
 - Obter escolha do humano.

- Obter escolha do computador.
- Chamar a função que decide vencedor da rodada.
- Atualizar humanScore ou computerScore conforme o resultado.
- Mostrar no console o resultado da rodada e o placar atual.
- Ao final, comparar placares e declarar o vencedor ou empate.

Validações e comportamento esperado

- Se o usuário digitar algo inválido:
- Opção A (simples): trate como jogada inválida (não pontua) e informe no console.
- Opção B (melhor): re-prompt até digitar algo válido (requer loop extra).
- Garanta case-insensitive convertendo a entrada para minúsculas antes de comparar.

Boas práticas e organização

- Separe responsabilidades: cada função deve fazer apenas uma tarefa.
- Teste cada função isoladamente antes de integrar tudo.
- Use mensagens no console claras (ex.: 'Rodada 3 — Resultado: Empate — Placar: Você 1 x 2 Computador').

Extensões opcionais (após terminar)

- Re-prompt em caso de entrada inválida.
- Permitir escolher número de rodadas pelo usuário.
- Jogar até alguém chegar a N vitórias.
- Criar interface gráfica com HTML/CSS e botões.
- Mostrar histórico das rodadas.

Checklist rápido (antes de rodar)

- JS está vinculado e o console mostra mensagem de teste.
- getComputerChoice() retorna 'pedra'/'papel'/'tesoura' aleatoriamente.
- getHumanChoice() pede e retorna a escolha em minúsculas.
- playRound() recebe duas strings e retorna o resultado correto.
- playGame() roda 5 vezes, chama funções, atualiza placar e imprime o resultado final.
- Testes manuais feitos e resultados conferidos no console.

Gerado por ChatGPT — instruções didáticas para implementação do projeto Pedra, Papel e Tesoura.