

Esse teste cobre praticamente **todas as bases essenciais de JavaScript moderno para manipular o DOM e controlar lógica de jogo.**

Vou te mostrar **exatamente o que você precisa dominar** — sem ir além do necessário — para conseguir resolver algo nesse nível com confiança.

## 1. Fundamentos de JavaScript

Antes de tudo, você precisa entender a **lógica base da linguagem**:

### ◆ Variáveis e Tipos

- `let`, `const` e `var` (diferenças e escopo)
- Tipos primitivos: `string`, `number`, `boolean`, `array`, `object`, `null`, `undefined`

### ◆ Operadores

- Aritméticos (`+`, `-`, `*`, `/`, `%`)
- Comparação (`==`, `===`, `!=`, `!==`, `>`, `<`, `>=`, `<=`)
- Lógicos (`&&`, `||`, `!`)
- Atribuição (`=`, `+=`, `-=`, etc.)

### ◆ Estruturas de Controle

- `if`, `else if`, `else`
- `for`, `while`, `forEach`
- `return` e `break`

### ◆ Funções

- Funções declaradas: `function nome() {}`
  - Arrow functions: `() => {}`
  - Passagem de parâmetros e retorno de valores
-

## 2. Arrays e Métodos Úteis

O teste usa **vários métodos de array**, então você deve saber:

- Criar arrays: `let ships = [20, 21, 22];`
  - Acessar elementos: `ships[0]`
  - Verificar se contém: `ships.includes(valor)`
  - Adicionar elementos: `hits.push(valor)`
  - Percorrer: `cells.forEach(cell => { ... })`
  - Saber a propriedade `.length`
- 

## 3. Manipulação do DOM

Essa é a parte **mais importante** para o teste.

Você precisa entender como **ligar o JavaScript ao HTML**.

### Selecionar elementos

- `document.getElementById("id")`
- `document.querySelector("seletor")`
- `document.querySelectorAll("seletor")`

### Alterar conteúdo e classes

- `element.textContent = "texto"`
- `element.classList.add("classe")`
- `element.classList.remove("classe")`
- `element.classList.contains("classe")`

## ◆ Atributos

- `element.getAttribute("nome-do-atributo")`
  - `element.setAttribute("nome", "valor")`
- 

## 4. Eventos e Manipulação de Clique

Essencial no jogo:

- `element.addEventListener("click", função)`
  - `element.removeEventListener("click", função)`
  - `event.target` (quem foi clicado)
  - Como usar funções para lidar com eventos (`handleCellClick`)
- 

## 5. Controle de Estado e Lógica do Jogo

Aprenda a controlar variáveis globais e condições:

- Atualizar valores no DOM (`textContent`)
  - Criar condições para vitória ou derrota (`if, else`)
  - Bloquear jogadas (`remove event listeners`)
  - Reiniciar variáveis e interface (`restart-button`)
-



## 6. Funções de Reinício e Encerramento

Você deve entender:

- Como "resetar" valores (`bombsRemaining = 3; hits = []`)
- Como limpar classes do DOM (`classList.remove`)
- Como reatribuir listeners depois do reset



## 7. Extras úteis

Não são obrigatórios, mas ajudam:

- Template strings: ``Olá, ${playerName}``
- Conversão de tipos: `parseInt()`, `Number()`
- `trim()` para limpar espaços



## Resumo rápido — o que estudar primeiro

Nível	Tópico	Exemplo
Básico	Variáveis e tipos	<code>let x = 10; const nome = "João";</code>
Básico	Condições	<code>if (x &gt; 5) { ... }</code>
Intermediário	Arrays	<code>arr.includes(2); arr.push(3);</code>
Intermediário	Funções	<code>function soma(a,b){return a+b}</code>
Essencial	DOM	<code>document.querySelectorAll(".cell")</code>
Essencial	Eventos	<code>addEventListener("click", fn)</code>
Avançado	Lógica de estado	reiniciar jogo, verificar vitória