

1. Entenda o que o jogo faz

Antes de começar, entenda o propósito do jogo:

- O jogo gera um número secreto entre 1 e 10.
- O jogador tenta adivinhar esse número digitando um "chute".
- O jogo dá dicas (maior ou menor) até o jogador acertar.
- Quando acerta, mostra quantas tentativas foram necessárias e permite reiniciar.

Vamos criar o código em etapas, começando pelas partes mais simples.

2. Crie as variáveis iniciais

Precisamos guardar algumas informações importantes para o jogo funcionar:

- Uma lista para evitar repetir números sorteados.
- O número máximo (limite) do jogo.
- O número secreto que o jogador vai tentar adivinhar.
- A contagem de tentativas.

Escreva isso no início do arquivo:

```
let listaDeNumerosSorteados = []; // Lista para guardar os números já sorteados

let numeroLimite = 10; // O maior número possível no jogo

let numeroSecreto = gerarNumeroAleatorio(); // Gera o número secreto (função que faremos depois)

let tentativas = 1; // Começa com 1 tentativa
```

- **let** é usado para criar variáveis que podem mudar.
 - **listaDeNumerosSorteados** começa vazia ([]) porque ainda não sorteamos nada.
 - **numeroSecreto** será definido por uma função que criaremos mais tarde.
-

3. Crie uma função para mostrar texto na tela

Precisamos de uma função para atualizar o texto no HTML (como o título <h1> e o parágrafo <p>).

- Essa função vai pegar uma tag HTML e substituir o texto dela.
- Também vai usar uma voz para ler o texto em português.

Escreva:

```
function exibirTextoNaTela(tag, texto) {  
  
    let campo = document.querySelector(tag); // Pega a tag HTML (ex.: 'h1' ou 'p')  
  
    campo.innerHTML = texto; // Coloca o novo texto na tag  
  
    responsiveVoice.speak(texto, 'Brazilian Portuguese Female', {rate: 1.2}); // Lê o texto em voz alta  
  
}
```

- **document.querySelector(tag)** encontra o elemento HTML pela tag (ex.: 'h1').
 - **innerHTML** muda o conteúdo desse elemento.
 - **responsiveVoice.speak** é uma ferramenta externa que lê o texto (já incluída no HTML).
-

4. Mostre a mensagem inicial do jogo

Agora, crie uma função para exibir a mensagem inicial quando o jogo começa:

```
function exibirMensagemInicial() {  
  
    exibirTextoNaTela('h1', 'Jogo do número secreto'); // Título  
  
    exibirTextoNaTela('p', 'Escolha um número entre 1 e 10'); // Instrução  
  
}  
  
exibirMensagemInicial(); // Chama a função para mostrar a mensagem assim que o jogo começa
```

- Essa função usa a **exibirTextoNaTela** para definir o título e a instrução inicial.
 - Chamamos ela logo em seguida para que o jogo já comece com esses textos.
-

5. Crie a função para gerar o número secreto

O número secreto deve ser aleatório e não pode se repetir enquanto houver números disponíveis:

- Vamos usar **Math.random()** para gerar um número aleatório.
- Vamos garantir que ele não esteja na lista de números já sorteados.

Escreva:

```

function gerarNumeroAleatorio() {

    let numeroEscolhido = parseInt(Math.random() * numeroLimite + 1); // Gera número
    entre 1 e 10

    let quantidadeDeElementosNaLista = listaDeNumerosSorteados.length; // Conta
    quantos números já foram sorteados

    if (quantidadeDeElementosNaLista == numeroLimite) { // Se já sortearmos todos os
    números possíveis

        listaDeNumerosSorteados = []; // Limpa a lista para começar de novo

    }

    if (listaDeNumerosSorteados.includes(numeroEscolhido)) { // Se o número já foi
    sorteado

        return gerarNumeroAleatorio(); // Gera outro número

    } else { // Se o número é novo

        listaDeNumerosSorteados.push(numeroEscolhido); // Adiciona à lista

        console.log(listaDeNumerosSorteados); // Mostra a lista no console (para teste)

        return numeroEscolhido; // Retorna o número

    }

}

```

- **Math.random()** gera um número entre 0 e 1, multiplicamos por **numeroLimite** (10) e somamos 1 para ficar entre 1 e 10.
- **parseInt** remove as casas decimais.
- Usamos **includes** para verificar se o número já está na lista e **push** para adicionar um novo número.

6. Crie a função para verificar o chute do jogador

Agora, vamos fazer a lógica para quando o jogador clicar no botão "Chutar":

- Pegamos o valor digitado no `<input>`.

- Comparamos com o numeroSecreto e damos uma dica ou comemoramos se acertar.

Escreva:

```
function verificarChute() {  
  
    let chute = document.querySelector('input').value; // Pega o número digitado no input  
  
    if (chute == numeroSecreto) { // Se o chute for igual ao número secreto  
  
        exibirTextoNaTela('h1', 'Acertou'); // Mostra "Acertou"  
  
        let palavraTentativa = tentativas > 1 ? 'tentativas' : 'tentativa'; // Escolhe singular ou plural  
  
        let mensagemTentativas = `Você descobriu o número secreto com ${tentativas} ${palavraTentativa}!`;   
  
        exibirTextoNaTela('p', mensagemTentativas); // Mostra a mensagem de vitória  
  
        document.getElementById('reiniciar').removeAttribute('disabled'); // Ativa o botão "Novo jogo"  
  
        document.getElementById('chutar').setAttribute('disabled', true); // Desativa o botão "Chutar"  
  
    } else { // Se o chute estiver errado  
  
        if (chute > numeroSecreto) { // Se o chute for maior  
  
            exibirTextoNaTela('p', 'O número secreto é menor');  
  
        } else { // Se o chute for menor  
  
            exibirTextoNaTela('p', 'O número secreto é maior');  
  
        }  
  
        tentativas++; // Aumenta o número de tentativas  
  
        limparCampo(); // Limpa o input para o próximo chute  
  
    }  
}
```

- **document.querySelector('input').value** pega o que o jogador digitou.
 - Usamos **if** para comparar o **chute** com o **número secreto**.
 - O operador ternário (**?:**) escolhe entre "tentativa" e "tentativas" com base no valor de tentativas.
 - **getElementById** e **setAttribute** / **removeAttribute** controlam os botões.
-

7. Crie a função para limpar o campo de input

Precisamos limpar o <input> após cada chute errado:

```
function limparCampo() {  
  
    let chute = document.querySelector('input'); // Pega o input  
  
    chute.value = ""; // Define o valor como vazio  
  
}
```

- Isso deixa o campo pronto para o próximo chute do jogador.
-

8. Crie a função para reiniciar o jogo

Por fim, crie a lógica para o botão "Novo jogo":

- Gera um novo número secreto.
- Reseta as tentativas e os textos.

Escreva:

```
function reiniciarJogo() {  
  
    numeroSecreto = gerarNumeroAleatorio(); // Gera um novo número secreto  
  
    limparCampo(); // Limpa o input  
  
    tentativas = 1; // Reseta as tentativas  
  
    exibirMensagemInicial(); // Volta para a mensagem inicial  
  
    document.getElementById('reiniciar').setAttribute('disabled', true); // Desativa o  
    botão "Novo jogo"  
  
    document.getElementById('chutar').removeAttribute('disabled'); // Ativa o botão  
    "Chutar"  
  
}
```

- Essa função "zera" o jogo para uma nova rodada.
-

9. Teste o código

Abra o arquivo HTML no navegador e teste:

- Digite um número no <input> e clique em "Chutar".
 - Veja se as dicas aparecem e se o jogo reinicia corretamente ao acertar.
-

Resumo

- Começamos definindo variáveis para controlar o jogo.
- Criamos funções para:
 1. Mostrar texto na tela.
 2. Gerar um número secreto único.
 3. Verificar o chute do jogador.
 4. Limpar o campo de entrada.
 5. Reiniciar o jogo.
- Cada função tem uma tarefa específica, tornando o código organizado e fácil de entender.

Tente modificar algo simples, como mudar o **numeroLimite** para 20, e testar novamente. **Isso ajuda a fixar o aprendizado!**