

Estruturas de controle e funções em JavaScript

1. Introdução

Depois de compreender variáveis, objetos e operadores, o próximo passo no aprendizado de JavaScript envolve as estruturas de controle e as funções. As estruturas de controle permitem que um programa tome decisões e repita ações, enquanto as funções possibilitam organizar o código em blocos reutilizáveis, evitando repetição e facilitando a manutenção. Juntos, esses recursos formam a base da lógica de programação em qualquer aplicação JavaScript.

2. Estruturas condicionais

As estruturas condicionais permitem que o programa execute diferentes blocos de código dependendo de uma condição ser verdadeira ou falsa. A estrutura mais comum é o "if...else":

```
const idade = 20;

if (idade >= 18) {
  console.log("Maior de idade");
} else {
  console.log("Menor de idade");
}
```

Quando há muitas condições possíveis para uma mesma variável, o uso da estrutura "switch" pode tornar o código mais legível do que uma longa sequência de "else if":

```
switch (diaDaSemana) {
  case 1:
    console.log("Segunda-feira");
    break;
  case 2:
    console.log("Terça-feira");
    break;
  default:
    console.log("Dia inválido");
}
```

3. Estruturas de repetição (loops)

As estruturas de repetição, ou loops, permitem executar um mesmo bloco de código diversas vezes, sendo especialmente úteis para percorrer listas de dados ou repetir uma ação até que uma condição seja satisfeita.

Estrutura	Uso comum
for	Repetição com número definido de iterações
while	Repetição enquanto uma condição permanecer verdadeira
do...while	Repetição que executa ao menos uma vez antes de testar a condição
for...of	Percorre os valores de um array ou outra estrutura iterável
for...in	Percorre as chaves (propriedades) de um objeto

```
for (let i = 0; i < 5; i++) {
  console.log("Contagem: " + i);
}
```

Nesse exemplo, o laço "for" inicializa a variável "i" com 0, repete o bloco de código enquanto "i" for menor que 5, incrementando "i" em 1 a cada repetição, exibindo os números de 0 a 4 no console.

4. Funções

As funções são blocos de código nomeados que podem ser reutilizados diversas vezes ao longo de um programa, recebendo valores de entrada (parâmetros) e, opcionalmente, retornando um valor de saída. Em JavaScript, existem diferentes formas de declarar funções:

```
// Função tradicional (declaração de função)
function somar(a, b) {
  return a + b;
}

// Expressão de função (arrow function)
const somar2 = (a, b) => a + b;

console.log(somar(3, 4)); // exibe 7
```

As funções do tipo "arrow function" (função de seta), introduzidas no ECMAScript 6 (ES6), oferecem uma sintaxe mais concisa e são amplamente utilizadas na programação JavaScript moderna, especialmente em bibliotecas e frameworks como React. Além dos parâmetros comuns, funções também podem receber valores padrão e um número variável de argumentos, ampliando sua flexibilidade.

5. Exemplos de aplicação

As estruturas de controle e as funções são utilizadas em praticamente todas as aplicações JavaScript do mundo real:

- **Validação de senha em formulários de cadastro** — estruturas condicionais verificam se a senha atende a requisitos mínimos, como quantidade de caracteres e presença de números, exibindo mensagens de erro específicas para cada caso.
- **Listagem de produtos em uma loja virtual** — estruturas de repetição, como "for" ou "for...of", percorrem um array de produtos, gerando dinamicamente os elementos HTML exibidos na página para cada item.
- **Cálculo de frete e descontos** — funções reutilizáveis recebem o valor total da compra e o CEP de entrega como parâmetros, retornando o valor final calculado, evitando repetir essa lógica em diferentes partes do sistema.
- **Menus de navegação dinâmicos** — estruturas condicionais verificam se o usuário está autenticado para exibir ou ocultar opções específicas do menu, como "Minha conta" ou "Sair".
- **Jogos e quizzes interativos** — laços de repetição controlam o avanço entre perguntas de um quiz, enquanto funções calculam a pontuação final do usuário com base nas respostas selecionadas.
- **Processamento de dados em massa** — funções combinadas com métodos de array, como "forEach" e "map" (que internamente utilizam o conceito de repetição), são utilizadas para transformar ou filtrar grandes conjuntos de dados recebidos de uma API.

6. Considerações finais

As estruturas de controle e as funções são elementos essenciais para a construção de qualquer lógica de programação em JavaScript. Enquanto as estruturas condicionais e de repetição permitem que o programa tome decisões e execute tarefas repetitivas de forma eficiente, as funções organizam o código em blocos reutilizáveis, tornando os programas mais legíveis, organizados e fáceis de manter à medida que crescem em complexidade.