

Linguagem de Programação



Prof. Renato Carioca Duarte

Unidade 05

Programação Modular

Programação Modular

- A modularização é o processo de quebrar um grande código JavaScript em partes menores, chamadas de módulos.
- Os módulos podem ser usados para organizar o código, torná-lo mais reutilizável e facilitar a manutenção.
- Existem várias maneiras de modularizar código JavaScript.
- Uma maneira é usar funções.
- As funções são blocos de código que podem ser chamados repetidamente.
- Você pode usar funções para criar módulos que executam tarefas específicas.

Programação Modular

- Outra maneira de modularizar código JavaScript é usar módulos.
- Os módulos são arquivos que contêm código JavaScript.
- Você pode usar módulos para agrupar código relacionado e torná-lo mais reutilizável.
- A modularização é uma técnica importante para escrever código JavaScript bem organizado e reutilizável.
- Ela pode ajudar a tornar o código mais fácil de entender, de manter e de testar.

Programação Modular

- Vantagens de usar módulos JavaScript:

1. Organização: Os módulos podem ajudar a organizar o código, tornando-o mais fácil de entender e de manter.

2. Reutilização: Os módulos podem ser usados para reusar o código, o que pode economizar tempo e esforço.

3. Testabilidade: Os módulos podem ser usados para tornar o código mais testável, o que pode ajudar a encontrar e corrigir erros.

4. Eficiência: Os módulos podem ajudar a tornar o código mais eficiente, reduzindo a quantidade de código que precisa ser carregado.

Programação Modular

- Para separar programas JavaScript em módulos, você pode utilizar o sistema de módulos disponível na especificação ECMAScript (ES6) e suportado pelos navegadores modernos e ambientes Node.js.
- O sistema de módulos permite dividir o código em arquivos separados, tornando o desenvolvimento mais organizado, reutilizável e fácil de gerenciar.

Programação Modular

- No arquivo que deseja exportar como módulo, você pode usar as palavras-chave ``export`` ou ``export default`` para tornar variáveis, funções ou classes disponíveis para outros módulos.
- O ``export`` é usado para exportar várias coisas, enquanto o ``export default`` é usado para exportar uma coisa única por módulo.

Programação Modular

- O sistema de módulos do JS baseia-se em quatro pilares:
 1. Cada script é um módulo por padrão, que esconde suas variáveis e funções do mundo externo.
 2. A instrução **export** permite que um módulo exporte os artefatos que deseja compartilhar.
 3. A instrução **import** permite que um módulo importe artefatos de outros módulos. Apenas artefatos exportados podem ser importados.

Exportando módulos

- A maneira mais fácil de usá-lo é colocá-lo na frente de qualquer item que você queira exportar para fora do módulo, por exemplo:

```
// modulo.js
export function soma(a, b) {
  |   return a + b;
}
```

Exportando módulos

- Uma maneira mais conveniente de exportar todos os itens que você deseja exportar é usar uma **única instrução de exportação no final do arquivo do módulo**, seguida por uma lista separada por vírgula dos recursos que você deseja exportar envoltos em chaves

```
// modulo.js
function soma(a, b) { return a + b; }
function subtrai(a, b) { return a - b; }

export { soma, subtrai }
```

Importando módulos

- Depois de exportar alguns recursos do seu módulo, é necessário importá-los para o script para poder usá-los.

```
// main.js  
import { soma } from './modulo.js';  
  
console.log(soma(2, 3)); // Saída: 5
```



JS

Dúvidas

??????