

JavaScript (JS)

Escopo

O escopo determina quais elementos podem ser acessados e em quais partes de um programa.

Imagine um escopo como uma caixa, que o que for criado nela poderá ser acessado por elementos dentro dessa caixa. Por exemplo, quando uma função é criada, um escopo é criado.

Toda vez que um programa é executado, é criado um escopo global, uma grande caixa em que outras caixas poderão ser colocadas.

Escopo global

Os elementos declarados fora de funções têm escopo global. Os elementos do escopo global podem ser acessados em qualquer parte de um programa.

Exemplo:

```
var limite = 9;    // Variável global
function exibe()
{
    console.log("limite vale " + limite);
}
exibe();           // Chama a função exibe()
```

Resultado:

limite vale 9

Escopo de função

Os elementos definidos em uma função têm escopo de função, isto é, só podem ser usadas na função em que foram definidos. Tais elementos são elementos locais à função em que foram definidos.

Exemplo:

```
function soma(n1, n2)
{
    var total = n1 + n2;
    return total;
}
var result = soma(2, 4);           // Chama a função soma()
console.log("Resultado da adição: " + result);
console.log("Total: " + total);
```

Resultado:

Resultado da adição: 6

A última linha (em negrito) causa erro, pois a variável *total* foi declarada na função *soma()*, não podendo ser usada fora desta.

Escopo de bloco

Os elementos escritos entre chaves (`{ ... }`) têm escopo de bloco, isto é são locais ao bloco em que foram definidos. Foi implementado com a palavra-chave **let**.

Exemplo:

```
var idade = 22;  
if (idade > 18)  
{  
  let estado = false;  
}  
console.log(estado);
```

A última linha (em negrito) causa erro, pois a variável *estado* foi declarada no bloco de código do comando **if**, não podendo ser usada fora deste.