

1. Escopo Global (O "Escritório Principal")

Variáveis e funções declaradas **fora de qualquer função** têm um **Escopo Global**. Isso significa que elas podem ser acessadas e modificadas por *qualquer* parte do script.

No seu código, é o seguinte:

JavaScript

```
let num = document.getElementById('fnum') // Variável Global  
let lista = document.getElementById('flista') // Variável Global  
let msg = document.getElementById('msg') // Variável Global  
let valores = [] // Variável Global (guarda todos os números)
```

```
// As funções também são globais, pois foram declaradas no nível mais alto  
function isNumero (n){ ... }  
function inLista (n, l){ ... }  
function adicionar(){ ... }  
function Finalizar(){ ... }
```

Por que estas variáveis são Globais?

Elas precisam ser acessíveis a várias funções diferentes:

Variável	Usada na Função	Usada na Função	Usada na Função
valores	adicionar (para colocar um número)	inLista (para checar se existe)	Finalizar (para fazer os cálculos)
lista	adicionar (para mostrar o número na caixa)	-	-
msg	adicionar (para limpar a mensagem)	Finalizar (para exibir os resultados)	-

Regra do Escopo Global: Se uma variável precisa ser usada por **mais de uma função**, ou se ela precisa **manter seu valor** entre chamadas de funções (como o valores[]), ela deve ser declarada no escopo **Global** (no topo do arquivo).

2. Escopo de Função (As "Salas de Reunião")

Variáveis declaradas **dentro de uma função** têm um **Escopo de Função** (também chamado de Escopo Local). Elas só existem e só podem ser acessadas dentro daquela função.

Exemplo na Função Finalizar()

JavaScript

```
function Finalizar() {  
    // ...  
  
    // Estas variáveis SÓ existem dentro da função Finalizar  
    let tot = valores.length  
    let maior = valores[0]  
    let menor = valores[0]  
    let soma = 0  
    let media = 0  
  
    for(let pos in valores ) {  
        // ... (usa tot, maior, menor, soma)  
    }  
  
    media = soma / tot  
    // ... (usa tot, maior, menor, soma, media)  
}  
  
// Depois que a função Finalizar() termina, tot, maior, menor, etc., desaparecem.
```

Por que estas variáveis são Locais?

1. **Relevância Local:** Elas só são necessárias para o cálculo final. Nenhuma outra função precisa saber o que é maior ou soma.
2. **Limpeza:** Se você declarasse soma globalmente, ela ficaria sempre guardada na memória. Ao ser local, ela é criada, usada e destruída sempre que a função Finalizar é chamada, economizando recursos.

Regra do Escopo de Função: Se uma variável é usada **apenas para um propósito temporário** (como contadores, somas, ou cálculos intermediários) **dentro de uma única função**, declare-a no escopo **Local** (dentro da função).

Resumo e Dicas para Ligar os Pontos

Escopo	Declaração	Quando Usar	Exemplos no Seu Código
Global	Fora de qualquer função (no topo).	Para variáveis que precisam ser acessadas por várias funções ou que precisam manter seu estado/valor (como a lista de números) entre diferentes ações.	valores, num, lista, msg
Local	Dentro de uma função (usando let ou const).	Para variáveis que são criadas, usadas e descartadas dentro de uma única função .	tot, maior, soma, pos (dentro do for...in)

Como entender isso na prática:

- Trace o caminho dos dados:** A lista valores é o coração do seu programa. Ela começa vazia no **escopo global**.
- Ação do Adicionar:** A função adicionar *adiciona* dados à lista global valores.
- Ação do Finalizar:** A função Finalizar *lê* os dados da lista global valores e *cria* suas próprias variáveis locais (soma, media) para fazer os cálculos.

Se você tentar usar soma fora da função Finalizar(), o JavaScript irá dar um erro, porque soma só existe dentro daquela "sala de reunião" específica!