

Lógica de Programação

A blue padlock is the central focus, set against a dark blue background filled with glowing binary code (0s and 1s) and hexadecimal characters (A-F, 0-9). The padlock is slightly open, and its keyhole is visible. The overall aesthetic is digital and technical.

MSc Sandir Rodrigues
Campos

Interagindo com o Usuário

- `prompt("texto a ser inserido na pergunta")`
- `prompt("Informe sua altura");`
- `var altura = prompt("Informe sua altura");`
- `var alturaInformada = prompt("Informe sua altura");`
- `var pesoInformado = prompt("Informe seu peso");`

```
var nome = prompt("Informe o seu nome"); var alturaInformada = prompt(nome + ",  
informe sua altura"); var pesoInformado = prompt(nome + ", informe seu peso"); var imc  
= calculaImc(alturaInformada, pesoInformado); document.write(nome + ", o seu IMC  
calculado é " + imc);
```

Exercício 1

Crie um código JS que exiba o resultado de duas funções que leiam dois números do teclado e na primeira função, some os números e na segunda função divida o primeiro pelo segundo. O resultado final deve ser imprimir a soma dos resultados das funções.

Exercício 1

```
<meta charset="UTF-8"> <script> function pulaLinha() {  
document.write("<br>"); } function mostra(frase) {  
document.write(frase); pulaLinha(); } function a(numero1, numero2) {  
return numero1 + numero2; } function b(numero1, numero2) { return  
numero1 / numero2; } var resultado = a(10,20) + b(30,2);  
mostra(resultado); </script>
```

Analisar linha a linha

```
<meta charset="UTF-8"> <script> function pulaLinha() {  
document.write("<br>"); } function mostra(frase) {  
document.write(frase); pulaLinha(); } function a(texto) {  
return "(" + texto + ")"; } function b(texto) { return "@" +  
texto + "@"; } function c(nome, sobrenome) { return  
b(nome + " " + a(sobrenome)); } var resultado =  
c("Sandir", "Campos"); mostra(resultado); </script>
```



Analisar linha a linha

```
<meta charset="UTF-8"> <script> function pulaLinha() { document.write("<br>"); } function mostra(frase) { document.write(frase); pulaLinha(); } function a(texto) { return "(" + texto + " "; } function b(texto) { return "@" + texto + "@"; } function c(nome, sobrenome) { return b(nome + " " + a(sobrenome)); } var resultado = c("Flávio", "Almeida"); mostra(resultado); </script>
```

@Sandir(Campos)@

A função b esta recebendo a concatenação do nome com o sobrenome, mas veja que o sobrenome é o retorno da função a. A função a devolverá o sobrenome entre parênteses, daí, esse valor será concatenado com o nome e o resultado finalmente será passado para a função B.

FJ – Rodando

- Nosso objetivo será perguntar ao usuário o número de vitórias e empates do seu time. Se você não gostar de futebol, pode inserir números aleatórios. Criaremos uma variável vitorias que receberá um prompt():

```
<meta charset="UTF-8"> <script> function pulaLinha() { document.write("<br>");  
document.write("<br>"); } function mostra(frase) { document.write(frase);  
pulaLinha(); } var vitorias = prompt("Entre com o número de vitórias.");  
var empates = prompt("Entre com o número de empates."); var pontos = vitorias  
* 3 + empates; mostra("Os pontos do seu time são " + pontos);  
</script>
```

FJ – Rodando

- Os pontos do seu time são 91
- Observamos que mesmo incluindo os parênteses e isolando a operação de multiplicação, o resultado seria o mesmo. O return da função `prompt()` - que não foi criada por nós, e é do JavaScript -, sempre retorna como texto, não como número.
- Por que isso funcionou para nosso cálculo de IMC, e não está funcionando agora? Vamos criar um `alert()` para somar o número de vitórias com 10:

Convertendo texto em números

- A função `parseInt`
- **SINTAXE** : `parseInt("texto a ser convertido")`

```
var vitorias = parseInt(prompt("Entre com o  
número de vitórias.)); var empates =  
parseInt(prompt("Entre com o número de  
empates.));
```