



07

Faça como eu fiz: Utilizando ternários e templates

Nesta aula vimos alguns conteúdos diversos sobre JavaScript, mas como podemos juntar tudo o que foi estudado?

Retornando ao exemplo de bebidas para menores e maiores de idade, vamos criar algumas variáveis para identificar um cliente, sua idade e as opções de bebidas.

```
const nome = "Leo";  
const idade = 23;  
const bebidaMaior = "cerveja";  
const bebidaMenor = "suco";
```

[COPIAR CÓDIGO](#)

Os operadores - **entre outras coisas** - são usados para comparar condições, algo que é muito usado em lógica de programação. A notação `${}` permite a inserção de valores de variáveis dentro de uma string de texto, mas não somente isso; vamos testar uma coisa nova:

```
const pedido = `${nome} diz: "por favor, quero beber ${idade > 18 ? bebidaMaior : bebidaMenor}"`;  
console.log(pedido)
```

[COPIAR CÓDIGO](#)

O resultado exibido no terminal é a frase completa:

```
Leo diz: "por favor, quero beber cerveja"
```

[COPIAR CÓDIGO](#)

Isso porque no início do código foi atribuída à variável `idade` um valor `23`, ou seja, maior do que `18`.

Com este exemplo, usamos três ferramentas do JavaScript vistas durante esta aula:

- 1) O uso de operadores: `>=` como operador de comparação entre o valor da variável `idade` e `18` e também o operador ternário como condicional para retornar o valor da variável `bebidaMaior` ou `bebidaMenor` de acordo com o resultado da comparação.
- 2) Template strings: Utilizamos a sintaxe do acento grave + `${}` em conjunto com as aspas duplas `""`. Teste também com aspas simples!
- 3) Operador ternário: Vimos que é possível não apenas exibir o valor de variáveis utilizando o `${}`, mas também fazer operações com JavaScript - por exemplo, condicionais - e inserir o correspondente ao `true` ou `false` na string de texto.