

//TAGGED TEMPLATE:

//Tagged Template - Template Marcado - é uma técnica onde podemos separar o que é "string" e o que são "valores de variáveis" em uma template string.

//Veja abaixo como podemos utilizar essa técnica:

//Abaixo temos o exemplo da montagem de um tagged template, perceba que elas são funções preparadas para operar sobre template strings, onde recebem um valor como parâmetro e um spread.

```
function tag(partes, ...valores) {  
    console.log(partes) //a variável "partes" vai receber tudo o que for string e armazenar como valores em um array...  
    console.log(valores) // e o spread "valores" vai receber tudo o que for valor de variável e armazenar em outro array...  
    return console.log(`1) ${valores[1]}${partes[1]} ${valores[0]}`) //Perceba que podemos retornar o que desejarmos, incluindo operações sobre os valores coletados nos arrays de "partes" e "valores".  
}
```

```
const nome = 'Rafael'
```

```
const nota = 'Aprovado'
```

```
tag `${nome}`, está ${nota}` //Perceba que para executar um tagged template usamos a função "tag" sem usar os parênteses, apenas passando o template string em seguida, ela automaticamente fará a divisão de valores e strings dentro dos arrays...
```

//OBS: Perceba também que partes e valor sempre serão intercalados em uma tagged template, partes sempre vai começar primeiro, mesmo que o valor seja vazio, ela sempre vai começar antes, depois valor vai receber outro valor e assim por diante...

//APLICAÇÃO DE TAGGED TEMPLATE NA PRÁTICA:

//Abaixo temos a aplicação de um tagged template, onde pegamos os valores numéricos e aplicamos a notação de real monetário nelas:

```
function real(partes, ...valores) {  
    const resultado = []  
    valores.forEach((valor, indice) => { //Usando um forEach(), capturamos os valores das variáveis e seus respectivos índices...  
        valor = isNaN(valor) ? valor : `R$ ${valor.toFixed(2)}` //Aqui fazemos uma comparação, se o valor não for numérico retornamos o valor, mas se for ele recebe os R$ do real e a os centavos...  
        resultado.push(partes[indice], valor) //Todos os resultados serão passados para o array "resultado" na ordem...  
    })  
    return resultado.join('') //Por fim, usamos um join para juntar tudo...
```

```
}  
  
const preco = 29.9  
const precoParcela = 11  
console.log(real `\\n2) 1x de ${preco} ou 3x de ${precoParcela}`)
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\\Users\\Almoxarifado\\Documents\\javascript\\arquivos_das_aulas\\152-  
Tagged_Template_Interceptando_Valores_e_Strings_de_Uma_Template_String.js"  
[ '', ', está ', '.' ]  
[ 'Rafael', 'Aprovado' ]  
1) Aprovado, está Rafael  
  
2) 1x de R$ 29.90 ou 3x de R$ 11.00  
  
[Done] exited with code=0 in 0.115 seconds
```