

05

## Funções com parâmetros

### Transcrição

Quando percebemos que estamos sendo repetitivos, podemos sempre considerar a criação de uma nova função. É fácil enxergar que toda vez que jogamos uma `frase` para o navegador utilizando o `document.write` logo em seguida pulamos uma linha. Para unificar isso em um único lugar podemos criar mais uma função, uma que mostre uma frase e coloque também o `<br>`. Por exemplo:

```
function mostra() {
    document.write("alguma frase<br/>");
}
```

Mas, construir essa função não é útil: ela sempre mostra a mesma frase. Ela não serve para atingir nosso objetivo, a frase que desejamos mostrar depende do momento. Mas, não se desespere! É possível resolver esse problema! Quando **declaramos** (criamos) uma função, podemos fazer isso de tal forma que recebamos algo a mais, como uma informação que seja importante. No caso da função que mostra uma frase, qual seria o valor importante? A `frase` que queremos mostrar! Então, faremos isso declarando dentro dos parênteses, como `function(frase)`, e depois utilizamos a `frase` normalmente com as variáveis que já conhecemos:

```
function mostra(frase) {
    document.write(frase + "<br/>");
}
```

Pronto! Mas, como o `mostra` saberá que a `frase` deve ser colocada no navegador? Isso será feito durante a chamada da função! Diferente do `pulaLinha`, que chamamos utilizando os parênteses sem nada dentro, o `mostra` será invocado com a `frase` dentro deles:

```
mostra("Usando funções para melhorar o código");
```

A string que queremos mostrar vai ser **passada** para a função que criamos e lá dentro teremos a variável `frase`. Variáveis que são passadas para funções são frequentemente chamadas de **parâmetros** ou **argumentos**.

Podemos mudar um pouco a função `mostra` para que ela também aproveite a velha `pulaLinha`, em vez de concatenar o `<br>` por si só:

```
function mostra(frase) {
    document.write(frase);
    pulaLinha();
}
```

Uma função chamar outra função é uma prática frequente. Qual a vantagem de fazer isso? O `mostra` também acompanha as mudanças do `pulaLinha`. Se quisermos pular a linha de forma visual, usando o `br`, o `hr` ou outro recurso do HTML, a função `mostra` beneficia-se disso sem nem mesmo precisar ser modificada!

