

□ 12

## Acumulando valores

### Transcrição

A pergunta agora é: Qual das duas maneiras devo usar?

A resposta a essa pergunta é simples, a que você preferir e achar mais simples de escrever, pois ambas, seja com o `for`, seja com o `while`, fazem exatamente o mesmo trabalho. Escolha a forma que você achar mais interessante e fácil de entender e continue com ela. Na sequência das aulas, vamos realizar as repetições de acordo com o caso.

Agora que entendemos o funcionamento das repetições por meio do `while` e do `for`, podemos partir para problemas mais elaborados. Portanto vamos revisitar o exercício no qual fazíamos o cálculo das médias de idades.

Até agora, precisávamos criar uma variável para cada idade para realizar a soma. Então tínhamos:

```
var idadeAdriano = 26;
var idadePaulo = 32;
var idadeSuzana = 25;
```

Sempre que queríamos incluir mais uma pessoa no cálculo da média de idade, precisávamos criar uma nova variável para ela e incluí-la no cálculo da média, por exemplo:

```
var idadeAdriano = 26;
var idadePaulo = 32;
var idadeSuzana = 25;
var idadeMarcela = 28;

var media = (idadeAdriano + idadePaulo + idadeSuzana + idadeMarcela) / 4;
mostra(media);
```

Ou seja, sempre que quisermos desconsiderar uma pessoa ou considerar uma nova pessoa no cálculo da média de idades, temos um baita trabalho de modificar esse código. Será que conseguimos fazer esse cálculo de média de idades de uma maneira genérica, ou seja, que funcione para qualquer quantidade de pessoas?

Esse problema é um pouco mais sutil que os anteriores e vamos resolvê-lo em um arquivo chamado `media_idade_familiares.html`. Primeiro, crie o novo arquivo.

Ao observar bem, temos uma característica de repetição, pois reproduzimos o processo da criação de uma variável. Como queremos que o processo funcione de maneira genérica, quem estiver usando o programa é que deve dizer quantos familiares deseja colocar no cálculo das médias. Vamos pedir essa informação através do `prompt`.

```
<script>
var totalDeFamiliares = prompt("Quantos familiares são?");
</script>
```

Agora que sabemos quantos familiares existem, vamos perguntar um por um suas idades. Dessa forma, se fossem informados 4 familiares, seriam perguntadas 4 idades. Repare que temos uma repetição, que nesse caso, vai de 1 até o número total de familiares.

```
var totalDeFamiliares = prompt("Quantos familiares são?");

var numero = 1;
while(numero <= totalDeFamiliares) {
    numero++;
}
```

Analizando o código, temos uma nova variável, chamada `numero`. Ela nos auxilia a saber até quando temos que repetir o trecho do código dentro do `while`. A cada repetição aumentamos seu valor em 1, através do `numero++`. E a cada nova repetição, temos que perguntar qual a idade da pessoa. Observe o código:

```
var totalDeFamiliares = prompt("Quantos familiares são?");

var numero = 1;
while(numero <= totalDeFamiliares) {
    var idadeTexto = prompt("Qual é a idade?");
    var idade = parseInt(idadeTexto);
    numero++;
}
```

Repare que usamos o `parseInt` para transformar o valor digitado no `prompt` em número. Isso porque vamos somar os valores `idade` e se deixarmos eles como estão, em `string`, eles seriam concatenados.

É comum fazer tudo isso numa única linha, descartando a necessidade de criar a variável `idadeTexto`, uma vez que essa é o tipo de variável que chamamos de temporária, já que ela só existe para nos auxiliar a calcular a `idade`. Para fazer isso direto, usamos o `parseInt(prompt("Qual é a idade?"))`. Parece complicado, mas não é! O JavaScript começa a execução pelos parênteses mais internos, no caso, executa o `prompt`. Com o resultado do `prompt` em mãos ele executa o `parseInt`, devolvendo o valor do que foi digitado como número pronto para ser usado nas operações matemáticas. Nossa código fica:

```
var totalDeFamiliares = prompt("Quantos familiares são?");

var numero = 1;
while(numero <= totalDeFamiliares) {
    var idade = parseInt(prompt("Qual é a idade?"));
    numero++;
}
```

Para calcular a média das idades, somamos todos os valores e dividimos pelo `totalDeFamiliares`. A cada repetição pegamos uma nova idade e somamos ao valor total, o problema é que este não é guardado. Para resolver o problema, vamos criar uma nova variável chamada `somaDeIdades`, que vai começar com o valor 0 e a cada repetição vamos somar com a nova idade digitada:

```
var totalDeFamiliares = prompt("Quantos familiares são?");
var somaDeIdades = 0;
```

```
var numero = 1;
while(numero <= totalDeFamiliares) {
    var idade = parseInt(prompt("Qual é a idade?"));
    somaDeIdades = somaDeIdades + idade;

    numero++;
}
```

Com isso, considere que temos 3 familiares. O primeiro com 20, o segundo com 30 e o terceiro com 40 anos. Na primeira repetição, o valor de `somaDeIdades` será 0 e somado a 20, passa a valer 20. Na segunda repetição o `somaDeIdades` vale 20 e deve ser somado com 30 e passará a valer 50. Por fim, `somaDeIdades` é 50 e somado a 40 valerá 90.

No fim das repetições, o valor da soma das idades é igual a 90! Agora que acabamos as repetições, se quisermos obter a média de idades, basta dividir o `somaDasIdades` por `totalDeFamiliares`. O valor 90 deve ser dividido por 3 familiares, então a média será 30 anos.

Então, após as repetições serem feitas - depois do `while` terminar seu trabalho - vamos calcular a média e mostrá-la. Para que isso ocorra temos as duas últimas linhas do código a seguir:

```
var totalDeFamiliares = prompt("Quantos familiares são?");
var somaDeIdades = 0;

var numero = 1;
while(numero <= totalDeFamiliares) {
    var idade = parseInt(prompt("Qual é a idade?"));
    somaDeIdades = somaDeIdades + idade;

    numero++;
}

var media = somaDeIdades / totalDeFamiliares;
alert("A média é: " + media);
```

Pronto, agora ao abrir o programa em seu navegador, ele perguntará primeiro pela quantidade de familiares e em seguida questionará a idade de cada um. Ao final dessa aula conseguimos construir um código que além de fazer as questões mencionadas anteriormente também calcula a média de idade dos familiares.

