

Operações com textos e números

Transcrição

Vamos fazer mais um teste! O objetivo é subtrair 2016 da idade de Flávio, portanto, vamos inserir o seguinte código:

```
document.write(2016 - 39);
```

No navegador a frase exibida é "A idade do Flávio é 1977". Mas essa não é a idade do Flávio e sim o ano em que ele nasceu. O programa precisa ser coerente, então, vamos alterar a mensagem para:

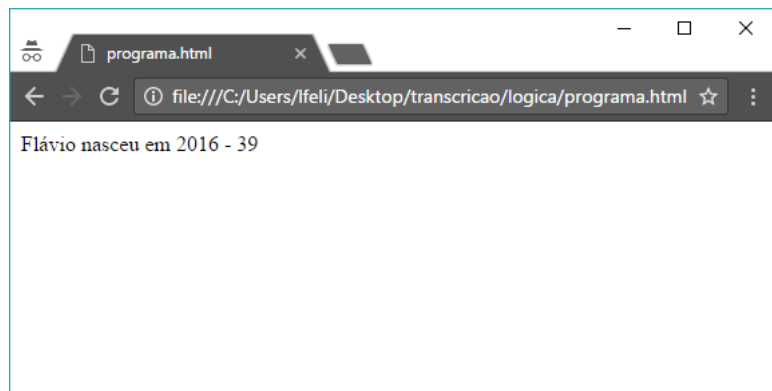
```
document.write("Flávio nasceu em ");  
document.write(2016 - 39);
```

Lembre-se de sempre salvar o arquivo e recarregar a página do navegador para visualizar a versão mais recente.

E o que acontece se fizermos o seguinte:

```
document.write("Flávio nasceu em 2016 - 39");
```

Veremos apenas um texto, observe:



Se fizermos a mesma frase, mas juntando os dois textos teremos o mesmo resultado:

```
document.write("Flávio nasceu em " + "2016 - 39");
```

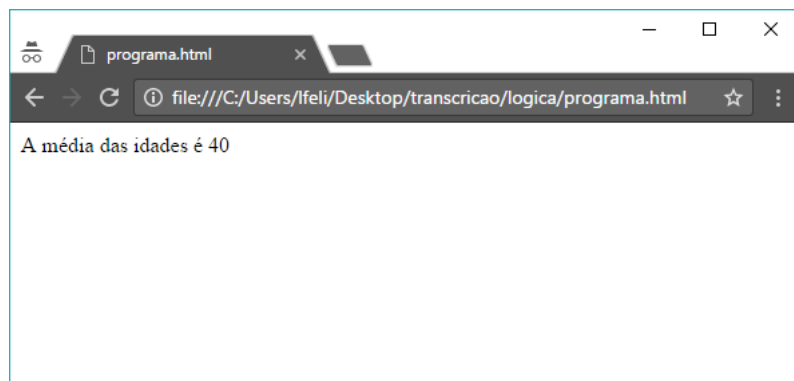
Vamos voltar e testar utilizar o `document.write` duas vezes:

```
document.write("Flávio nasceu em ");  
document.write(2016 - 39);
```

Agora, o objetivo é calcular a média das idades de três pessoas. Para calcular a média vamos somar as idades e dividir pela quantidade de pessoas:

```
document.write("A média das idades é ");  
document.write(20 + 10 + 30 / 3);
```

Nesse exemplo, a soma das três idades é 60 e ao dividir o valor por 3 o resultado é 20 que é a média das idades. Então, 20 é o resultado que desejamos ver no navegador. Entretanto, o número que aparece é 40:



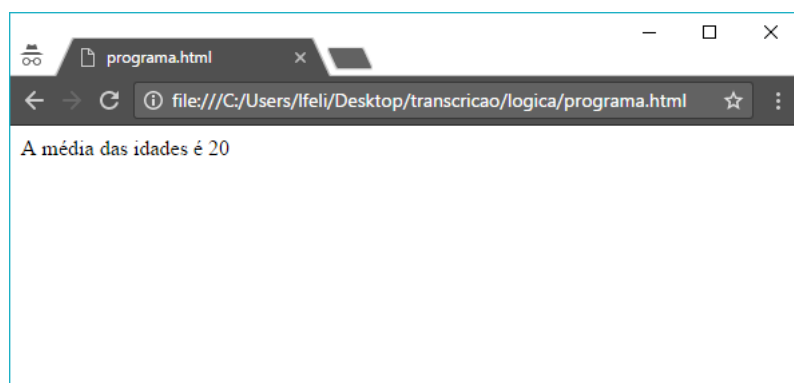
Por que isso acontece? Em programação, assim como na matemática, multiplicação e divisão são avaliadas primeiro. Primeiro é feita a divisão de 30/3, que resulta em 10. Nesse momento teríamos algo como:

```
document.write(20 + 10 + 10);
```

Em seguida as somas são realizadas e o valor 40 é passado como argumento para o `document.write()`. Assim como na matemática, quando queremos que algo seja avaliado primeiro, colocamos entre parênteses. Cuidado para não confundir com os parênteses do `document.write`:

```
document.write((20 + 10 + 30) / 3);
```

Agora sim! O que está entre parênteses será avaliado primeiro e teremos o resultado esperado:



Vamos fazer um novo teste! O que acontecerá se juntarmos uma `String` com números:

```
document.write("A soma das idades é " + 20 + 10 + 30);
```

O JavaScript verifica a existência de parênteses na expressão e como eles não existem a expressão será avaliada da esquerda para a direita. Então ele começará fazendo:

```
"A soma das idades é " + 20
```

O que ocorre quando juntamos texto e número? Uma concatenação! O resultado será a seguinte string:

```
"A soma das idades é 20"
```

Em seguida o JavaScript pega o resultado para fazer:

```
"A soma das idades é 20" + 10
```

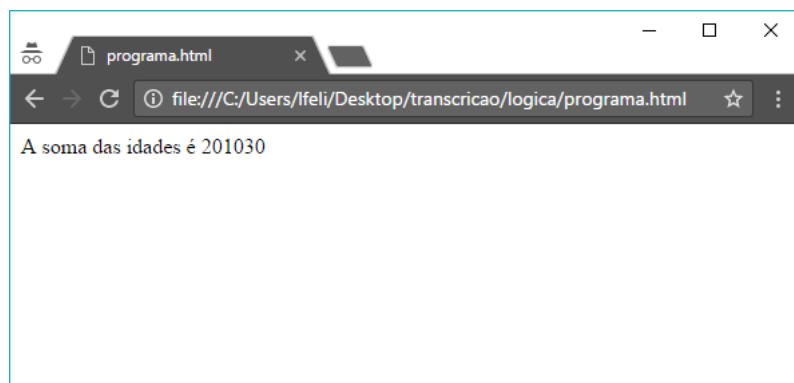
Como vimos, String + número resultará em texto. Então o resultado será:

```
"A soma das idades é 2010"
```

Por fim, teremos o seguinte:

```
"A soma das idades é 2010" + 30
```

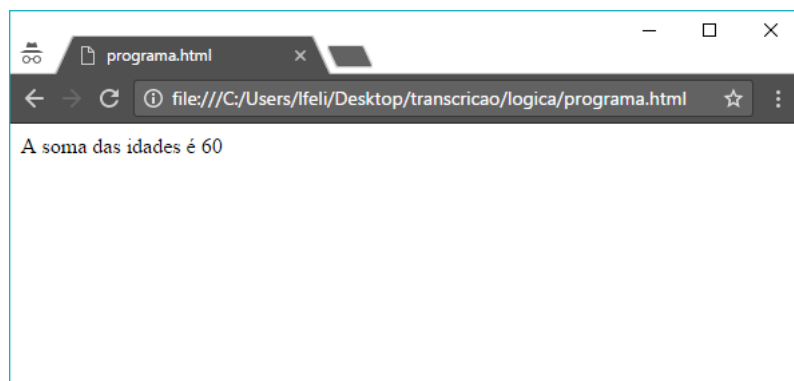
Vamos ver o resultado:



Mas, o objetivo era obter outro resultado. A operação de soma deve ser a primeira, por isso vamos recorrer aos parênteses:

```
document.write("A soma das idades é " + (20 + 10 + 30));
```

Agora chegamos ao resultado esperado:



Poderíamos também colocar o cálculo da média em uma única instrução:

```
document.write("A média das idades é " + (20 + 10 + 30) / 3);
```

Nesse caso, o JavaScript irá avaliar primeiro o que está entre parênteses para em seguida realizar a divisão e só então concatenar o resultado com o texto.

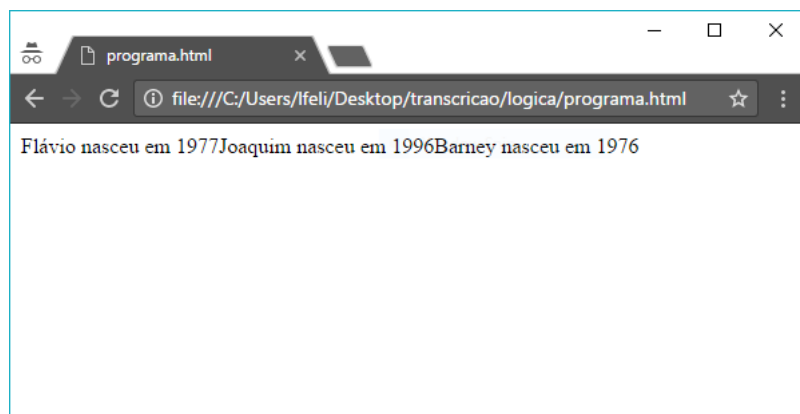
Vamos tentar algo diferente! Vamos calcular o ano em que uma pessoa nasceu subtraindo o ano atual pelo ano de seu nascimento. Por exemplo:

```
document.write("Flávio nasceu em " + (2016 - 39));
```

Lembrado que a operação de subtração deve vir entre parênteses, para que ela seja efetuada primeiro. Vamos adicionar mais mensagens:

```
document.write("Flávio nasceu em " + (2016 - 39));  
document.write("Joaquim nasceu em " + (2016 - 20));  
document.write("Barney nasceu em " + (2016 - 40));
```

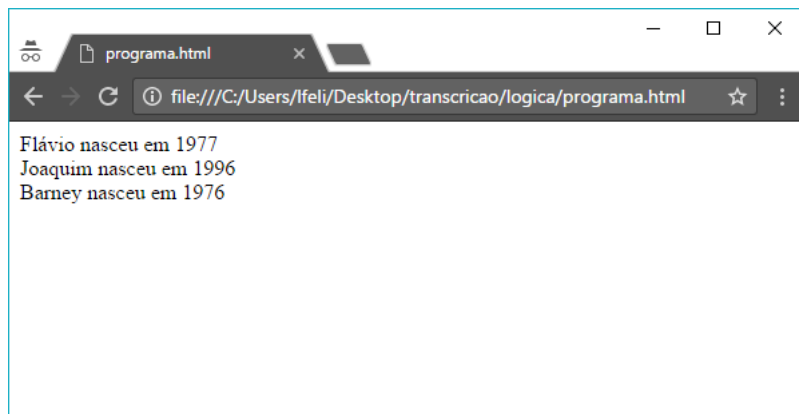
O resultado é o seguinte:



Como queremos que cada mensagem ocupe uma linha, podemos concatenar um `
` no final de cada instrução:

```
document.write("Flávio nasceu em " + (2016 - 39) + "<br>");  
document.write("Joaquim nasceu em " + (2016 - 20) + "<br>");  
document.write("Barney nasceu em " + (2016 - 40) + "<br>");
```

O resultado é o seguinte:



O JavaScript resolve primeiro o que está entre parênteses, a subtração. Como depois disso não existem mais parênteses, a expressão será avaliada da esquerda para a direita. E a mensagem "Flávio nasceu em " será concatenada com o valor 1977 e em seguida concatenado com a *tag* HTML `
`, que serve para pular uma linha.

Outra solução seria utilizar os `
` em instruções separadas:

```
document.write("Flávio nasceu em " + (2016 - 39));  
document.write("<br>");  
document.write("Joaquim nasceu em " + (2016 - 20));  
document.write("<br>");  
document.write("Barney nasceu em " + (2016 - 40));  
document.write("<br>");
```

O resultado é o mesmo!

Assim, existem maneiras diferentes de se chegar a um mesmo resultado, o programador usa aquela que prefere ou que é mais fácil de entender. Particularmente, a instrução separada é a que utilizaremos, pois o `
` solto é mais fácil para perceber que existe um pulo de linha. Mas, nada impede que a concatenação seja utilizada no código!

