

01

Reduzindo alterações

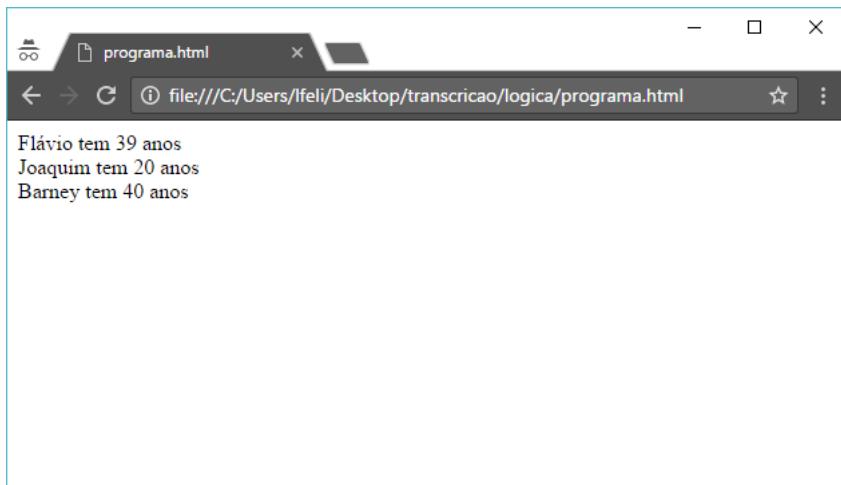
Transcrição

Começando deste ponto? Faça o [DOWNLOAD \(https://github.com/alura-cursos/logica-de-programacao-I/archive/aula-3.zip\)](https://github.com/alura-cursos/logica-de-programacao-I/archive/aula-3.zip) do projeto do capítulo anterior e continue seus estudos a partir deste capítulo.

A proposta é, ao em vez de mostrar o ano de nascimento, exibir a idade. Para fazer isso alteramos o texto e subtraímos o ano atual pelo ano de nascimento:

```
document.write("Flávio tem " + (2016 - 1977) + " anos");
document.write("<br>");
document.write("Joaquim tem " + (2016 - 1996) + " anos");
document.write("<br>");
document.write("Barney tem " + (2016 - 1976) + " anos");
document.write("<br>");
```

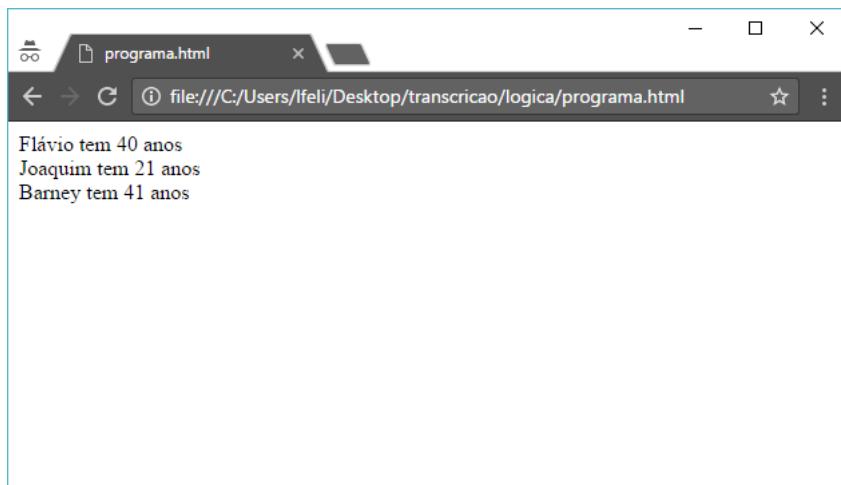
O resultado é o seguinte:



O que acontece se o ano mudar? Isto é, a medida que o tempo for passando e os anos avançando: 2017, 2018, 2019... Em 2017, o Flávio terá 39 ou 40? O Joaquim, terá 20 ou 21? E Barney, 40 ou 41? Nesse caso, sempre que o ano mudar teremos que trocar o valor do código:

```
document.write("Flávio tem " + (2017 - 1977) + " anos");
document.write("<br>");
document.write("Joaquim tem " + (2017 - 1996) + " anos");
document.write("<br>");
document.write("Barney tem " + (2017 - 1976) + " anos");
document.write("<br>");
```

Ao salvar o arquivo e atualizar a página no navegador, podemos ver as idades atualizadas:



Quando chegar o ano de 2018, teremos que alterar o código novamente. E se tivermos uma lista com informações de 500 pessoas? Teremos que trocar o ano em diversos pontos do programa. Nesse caso, será que conseguimos lembrar de fazer a modificação em todos os lugares do código? Não!

Precisamos descobrir uma maneira de escrever o ano atual em um único lugar do código e fazer com que ele seja acessado por outros diversos lugares. E, caso alteremos sua informação, queremos que todos os lugares que utilizam esse valor sejam automaticamente modificados, disparados.

É como se tivéssemos um envelope e na frente dele estivesse escrito o ano e dentro existisse um papel com o valor 2016. Vamos iniciar trocando o numeral corresponde ao ano pelo valor "ano". Teremos:

```
document.write("Flávio tem " + (ano - 1977) + " anos");
document.write("<br>");
document.write("Joaquim tem " + (ano - 1996) + " anos");
document.write("<br>");
document.write("Barney tem " + (ano - 1976) + " anos");
document.write("<br>");
```

Em programação, para resolver problemas como esse, fazemos uso de variáveis. A nomenclatura variável é porque justamente ela pode variar. O `ano` pode ser 2016 em um momento e 2017 em outro momento e por aí vai. Para declarar uma variável, utilizamos a palavra `var` que é a abreviação de *variable* (variável):

```
var ano;
```

Mas, qual é o valor do `ano`? Ele precisa receber um valor, assim, precisamos **inicializar** a variável. Para isso, utilizamos o símbolo `=` (igual):

```
var ano = 2016;
```

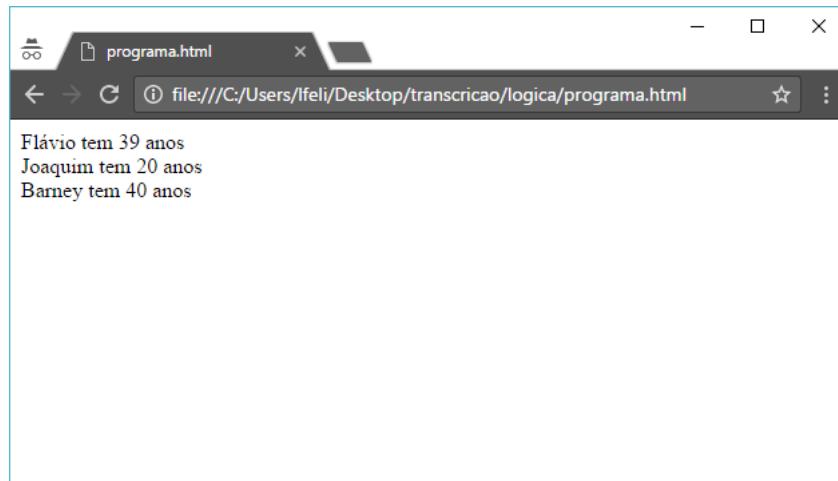
Para falar do valor de uma variável não nos referimos dizendo "a variável `ano` é igual a 2016", mas sim "a variável `ano` **recebe** 2016". O símbolo de `=` faz o que chamamos de **atribuição**. O código fica da seguinte maneira:

```
var ano = 2016;

document.write("Flávio tem " + (ano - 1977) + " anos");
document.write("<br>");
```

```
document.write("Joaquim tem " + (ano - 1996) + " anos");
document.write("<br>");
document.write("Barney tem " + (ano - 1976) + " anos");
document.write("<br>");
```

Ao salvar o arquivo e recarregar o navegador, temos o seguinte resultado:

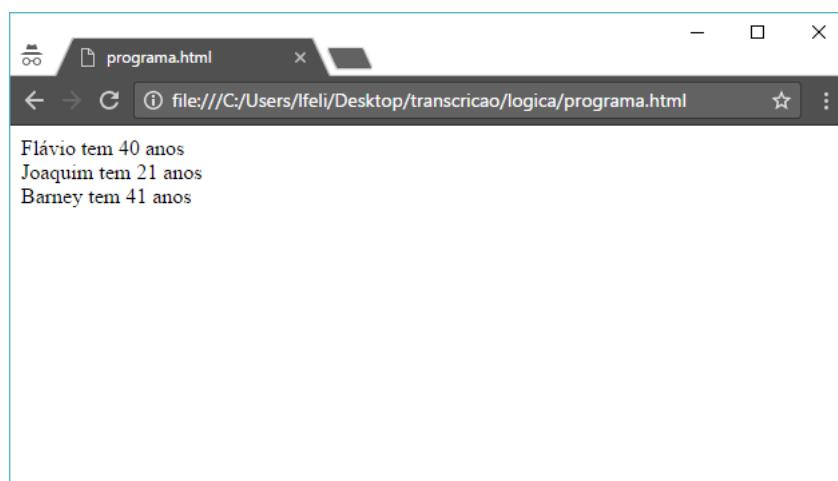


Agora, quando o ano passar e for preciso modificar para 2017 a data, basta fazer a alteração da informação em um único lugar:

```
var ano = 2017;

document.write("Flávio tem " + (ano - 1977) + " anos");
document.write("<br>");
document.write("Joaquim tem " + (ano - 1996) + " anos");
document.write("<br>");
document.write("Barney tem " + (ano - 1976) + " anos");
document.write("<br>");
```

Dessa forma todos os lugares que utilizam a variável `ano`, passam a ter o valor 2017. Ao salvar o arquivo e recarregar, temos:



O que o JavaScript faz quando encontra a variável `ano`, como na linha a seguir?

```
document.write("Flávio tem " + (ano - 1977) + " anos");
```

O Javascript verifica se `ano` é uma `String`, mas sabemos que não, pois não está entre aspas. Ele verifica se é um número, mas a resposta também é não. Então, também verifica se é uma variável e em seguida se existe a variável com nome `ano`. Por fim, ele troca `ano` por valor, que agora é 2017:

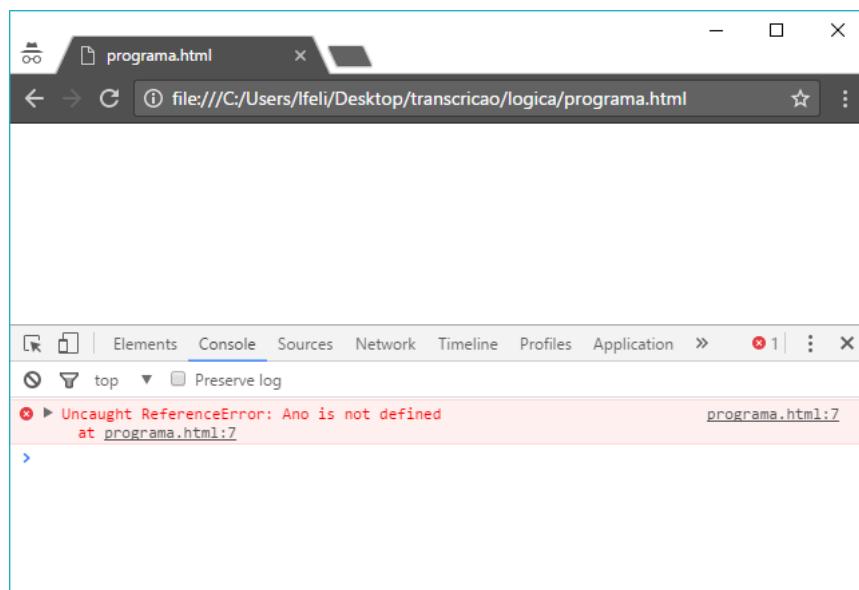
```
document.write("Flávio tem " + (2017 - 1977) + " anos");
```

Trabalhar com variáveis é interessante pois temos a possibilidade de declarar uma variável em um único lugar. E como várias instruções dependem da variável declarada, quando alteramos seu valor ocorre a mudança em todos os lugares do código onde ela é utilizada.

Cuidado! A variável deve ser sempre escrita com todas as letras em minúsculo. Se você fizer algo do tipo, letra "A" de `ano` em maiúscula, o código não funcionará! Faça o teste:

```
document.write("Flávio tem " + (Ano - 1977) + " anos");
```

Ao abrir o console, podemos ver o erro:



O erro indica que a variável `Ano` não foi definida. É um erro comum quando estamos começando a programar. A variável `ano` deve ser escrita toda em minúsculas, portanto ela deve ser repetida sempre dessa maneira ao longo do código.

Podemos alterar, a qualquer momento, o valor da variável:

```
var ano = 2016;

document.write("Flávio tem " + (Ano - 1977) + " anos");
document.write("<br>");
document.write("Joaquim tem " + (ano - 1996) + " anos");
document.write("<br>");

ano = 2017;
document.write("Barney tem " + (ano - 1976) + " anos");
document.write("<br>");
```

Quando o navegador processa a `tag script`, ele lê linha a linha das instruções na ordem em que elas foram declaradas. Na primeira linha, a `ano` recebe `2016`. Então nos primeiros dois usos da variável `ano`, o valor será `2016`. Mas, antes de calcular a idade do `Barney`, o valor passa a ser `2017`, devido a seguinte declaração:

```
ano = 2017;
```

Você pode estar se perguntando: Por que não utilizar mais o `var` no momento em que atribuímos o valor `2017`? Utilizamos `var` apenas na primeira declaração da variável. Se você deseja atribuir um novo valor a variável, é simples, basta colocar o nome da variável e atribuir o novo valor.

