

Aula 9

Desenvolvimento JavaScript I

Variáveis e Operações

Declaração de variável, seus tipos e como usá-las em operações

O que são variáveis

Variáveis são contêineres que guardam informações. Essas informações podem ser números inteiros ou decimais ou texto. As variáveis possuem este nome pois podem ser modificadas e variam seu valor com declarações ou operações matemáticas.

No exemplo a seguir, usamos a keyword 'var' para declarar 3 variáveis:

```
var x = 5;  
var y = 6;  
var z = x+y;
```

Neste caso, as variáveis 'x' e 'y' recebem valores numéricos inteiros, mas 'z' recebe o valor da soma das outras duas variáveis. Sendo assim, o valor de 'z' não é 11, pois as outras variáveis podem mudar seu valor, então o valor de 'z' é sempre o resultado de 'x'+y'.

Keywords para variáveis

Como vimos anteriormente, na seção de sintaxe, podemos declarar variáveis com 3 keywords diferentes, mas quando usar cada uma?

Keyword Var

A keyword 'var' para declaração de variáveis foi usada no JavaScript de 1995 até 2015, mas foi substituída em 2015 por 'let' e 'const' devido a bugs que causava em certas circunstâncias. Então se for criar um código que será executado em um navegador antigo, com versão anterior a 2015, utilize 'var'.

```
var x = 5;  
var y = 6;  
var z = x+y;
```

Keyword Let

Já o 'let', que foi implementado em 2015, é semelhante ao 'var', porém teve os bugs corrigidos em relação ao seu antecessor. Ele funciona com declaração em blocos e não pode ser re-declarado.

No exemplo a seguir, podemos ver o que ocorreria ao tentar declarar a variável duas vezes, com o 'let' teríamos um erro, com o 'var' seria possível redeclarar. Apesar de parecer uma funcionalidade que deixaria o código flexível, na verdade essa situação poderia causar diversos bugs no código.

```
let nome = "John Doe";  
let nome = 0; // SyntaxError: 'x' has already been declared
```

```
var x = "John Doe";  
var x = 0; //declara a variável novamente
```

O 'let' também possui outra funcionalidade importante, que é o escopo de bloco. Isso significa que uma variável criada dentro de um bloco { } não pode ser usada fora deste bloco.

```
{  
  let x = 2;  
}  
// x não pode ser usado aqui
```

```
{  
  var x = 2;  
}  
// x PODE ser usado aqui
```

O escopo de bloco impacta também na forma como se dão as declarações das variáveis. Uma variável 'let' chamada 'n1' declarada dentro de um bloco não será a mesma que outra variável de mesmo nome declarada fora do bloco. Já no 'var', a variável seria a mesma, independente de estar dentro do bloco ou não. Vamos ver o exemplo para entender melhor:

```
var x = 10;  
// Aqui x é 10
```

```
{  
  var x = 2;  
  // Aqui x é 2  
}
```

```
// Aqui x é 2
```

```
let x = 10;  
// Aqui x é 10
```

```
{  
  let x = 2;  
  // Aqui x é 2  
}
```

```
// Aqui x é 10
```

Keyword Cons

Variáveis 'cons' não podem ser re-declaradas ou modificadas após a declaração, seu valor deve ser definido no momento da declaração. Também possuem escopo de bloco.

```
const PI = 3.141592653589793;  
PI = 3.14;           // Vai mostrar erro  
PI = PI + 10;        // Também vai mostrar erro
```

Como regra geral, sempre declare uma variável com 'const' a menos que você saiba que o valor será alterado.

Operador de Atribuição

Em JavaScript, o sinal de igual = é um operador de "atribuição", não um operador "igual a". Isso é diferente da álgebra. O operador "igual a" é escrito como == em JavaScript.

```
let num = 1; //atribui o valor 1 à variavel num  
  
num == 2; //verifica se a variavel é igual a 2,  
          // não é então será falso
```

Tipos de dados

As variáveis JavaScript podem conter números como 100 e valores de texto como "João". Na programação, os valores de texto são chamados de strings de texto. JavaScript pode lidar com muitos tipos de dados, mas por enquanto, pense apenas em números e strings.

Strings são escritas entre aspas duplas ou simples. Os números são escritos sem aspas. Se você colocar um número entre aspas, ele será tratado como uma string de texto.

```
const pi = 3.14;  
let nome = "João";  
let numero = '123';
```

Uma declaração, muitas variáveis

Você pode declarar muitas variáveis em uma instrução. Inicie a instrução com 'let' e separe as variáveis por vírgula:

```
let nome = "John Doe", carNome = "Volvo", preco = 200;
```

Ou então, declare em múltiplas linhas, usando vírgula ao invés de ponto e vírgula.

```
let nome = "John Doe",  
    carNome = "Volvo",  
    preco = 200;
```

Valor indefinido

É recorrente as variáveis serem declaradas sem um valor, pois o valor pode ser algo que precisa ser calculado a partir de outras variáveis ou que será providenciado após o início da execução, como entrada de dados do usuário.

Quando uma variável é declarada sem valor, seu valor será definido como indefinido.

```
let carModelo; //variável de valor indefinido
```

Aritmética

Para fazer os testes de aritmética, precisamos de um arquivo HTML para visualizar os resultados obtidos. Portanto, crie um arquivo HTML dentro da sua pasta e insira a estrutura inicial:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title></title>
</head>
<body>

  <p id="teste"></p>

  <script>
    let x = 1;
    document.getElementById("teste").innerHTML = x;
  </script>

</body>
</html>
```

Vamos testar a adição nas 3 variáveis a seguir:

```
<script>
  let x = 5 + 2 + 3;
  document.getElementById("teste").innerHTML = x;
</script>
```

```
<script>
  let x = "John" + " " + "Doe";
  document.getElementById("teste").innerHTML = x;
</script>
```

```
<script>
    let x = "5" + 2 + 3;
    document.getElementById("teste").innerHTML = x;
</script>
```

Podemos observar que na primeira variável o resultado retornado é a soma dos números. Já na segunda, como não são números e sim texto, isto é, uma variável String, os dois textos se concatenam e aparecem juntos ao imprimir na tela. Na terceira, ocorre outra situação interessante, ao transformar o primeiro número em String, os números seguintes da equação também serão tratados como String e vão apenas concatenar.

Tarefa de Casa

Resolva as seguintes expressões usando variáveis e imprima o resultado na tela usando *document.getElementById*

1. $6 + 20 - 12 \times 2$
2. $30 \div [10 + (2 + 3)]$
3. $[(18 + 3 \cdot 2) \div 8 + 5 \cdot 3] \div 6$
4. $[(64 - 16 \cdot 4) + (48 \cdot 10 - 180)] \cdot 5$
5. $\{[(8 \cdot 4 + 3) \div 7 + (3 + 15 \div 5) \cdot 3] \cdot 2 - (19 - 7) \div 6\} \cdot 2 + 12$