UNINOVE

Sintaxe: Operadores, variaveis e expressões em JavaScript. Estrutura básica da linguagem JavaScript

CRIAR UMA ESTRUTURA BÁSICA DE UM SCRIPT JAVASCRIPT. CRIAR E UTILIZAR VARIÁVEIS EM CÓDIGO JAVASCRIPT.

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR

Variáveis

Variáveis são espaços ou endereços de memória, reservados para armazenamento de dados. Em outras palavras, variáveis são locais onde o seu programa ou código pode armazenar dados das mais diversas formas, para serem utilizados e alterados posteriormente quando necessário, ou seja, este espaço reservado de memória pode ter seu conteúdo ou valor modificado. Vale lembrar, que estes dados das variáveis estarão armazenados apenas enquanto o programa estiver em execução. Quando o programa for finalizado os valores destas variáveis não estarão mais disponíveis.

As variáveis possuem tipos primitivos, ou seja, tipos básicos e diferentes destinados a armazenar dados de diferentes tipos.

TIPO	EXEMPLO	FINALIDADE
Inteiro	10	Valores inteiros
Real	31.45	Valores fracionados
Caractere	“prova”	Caracteres alfanuméricos ou símbolos
Lógico	Verdadeiro (true)	Booleano (true ou false)

Observe que os valores fracionados estão no padrão americano, ou seja, a parte fracionada está separada por ponto e não por virgula.

Diversas linguagens são fortemente tipadas, pois no momento da criação de uma variável é necessário determinar o seu tipo, por exemplo nas linguagens C, C++, Java, etc.

Outras linguagens permitem a determinação de tipos dinamicamente, ou seja, não é necessário determinar o tipo da variável no momento de sua criação, por exemplo em linguagens como PHP, VB, VBScript, Javascript, dentre outras.

Em Javascript é possível declarar as variáveis de forma implícita ou explicita, veja os exemplos.

Método explicito:

```
var nome_Cliente;
```

Método implícito

```
nome_Cliente = “João da Silva”;
```

Em nossos exemplos utilizaremos o método explicito, por ser uma boa prática de programação. No método explicito, adicionamos o valor var antes do nome da variável.

Nomenclatura de variáveis

Existem algumas regras clássicas para se determinar o nome das variáveis, pois não podemos criar nomes de variáveis de qualquer maneira, são elas:

- Podem conter letras, dígitos, underscores, e o símbolo do dólar “$”.
- Devem começar com uma letra, mas também podem ser iniciados com os símbolos $ e _ .
- São case sensitive, ou seja, x e X são diferentes (minúscula são diferentes de MAIUSCULAS)
- Não dever ser iniciados por números.
- Palavras reservadas da linguagem não podem ser utilizadas.
- Não devem conter espaços em branco ou caracteres especiais, a não ser o $ e o _ .

Exemplos corretos:

```
var nome;
var nome_Completo;
var cod_Usuario;
var _valor;
var nota_1;
```

Exemplos incorretos:

```
var 1nome;
var if;
var nome completo;
var@idade;
```

> ## JAVACRIPT UTILIZA O CAMELCASE!
>
> Dica: A linguagem Javacript é *sensivel*, pois utiliza o camelCase, que de maneira resumida é diferenciar uma variável com nome composto (mais de uma palavra) iniciando a segunda palavra com letra maiúscula, por exemplo: nomeFornecer, notaFiscal, valorInicial, valorFinal, primeiroSegundo. Considere utilizar este padrão em seus códigos.

Constantes

Em algumas situações necessitamos criar variáveis que não devem ser alteradas durante a execução de seu programa. Nestes casos, criamos as constantes. As constantes são um tipo especial de "variável" pois o seu valor não é alterado no momento da execução do programa. Pode-se sim, alterar o valor inicial da constante, mais isso é possível apenas reiniciando o seu valor inicial. Para determina-la, adicionando const antes no nome da constante.

Um exemplo prático de constante é uma variável como o número PI:

```
const  pi;
```

Operações com variáveis

Uma grande vantagem de se trabalhar com variáveis, além de armazenar dados temporariamente, é poder realizar operações matemáticas com estes valores. Vamos verificar alguns exemplos:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 3</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 3)</div>
9.         <script type="text/javascript">
10.            var nome = "Javascript";
11.            var valor = 10.50;
12.            var idade = 20;
13.            var verificado = true;
14.        </script>
15.    </body>
16. </html>
```

Observe o código acima. As linhas 10 a 13 são destinas a criação de variáveis explicitas, cada uma com sua finalidade, com um tipo de dado significativo.

- var nome = "Javascript"; (esta variável recebe um valor caracter ou string [*conjunto de caracteres*]).
- var valor = 10.50; (esta variável recebe um valor real, valor numérico de ponto flutuante [*casas decimais*]) .
- var idade = 20; (esta variável recebe um valor inteiro, valor numérico do tipo inteiro, sem precisão matemática).
- var verificado = true; (esta variável recebe um valor lógico, podendo ser verdadeiro ou falso).

Vale lembrar, que todo o fim de linha é finalizado por ponto e virgula.

Vamos verificar algunas operações:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 3</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 3)</div>
9.         <script type="text/javascript">
10.            var a = 1;
11.            var b = 2;
12.            document.write(a + b);
13.        </script>
14.    </body>
15. </html>
```

Observe que no código acima foram criadas duas variáveis, a e b. A variável a recebeu o valor inicial 1 e a variável b recebeu o valor inicial 2. A seguir, na linha 12 do código acima, é utilizado o comando document.write( ). Este comando permite exibir um valor no corpo do documento HTML, o valor a ser apresentado deve estar dentro dos parenteses.

Neste exemplo, o resultado deve ser o numero 3, pois é o resultado da soma da variável a e b (1 + 2).

Vamos verificar um novo exemplo:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 3</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 3)</div>
9.         <script type="text/javascript">
10.            var nome = "Ayrton";
11.            var sobrenome = "Senna";
12.            const profissao = "Piloto";
13.            document.write("<p>1) Piloto = " + nome + sobrenome + "</p>");
14.            document.write("<p>2) Piloto = " + nome + " " + sobrenome + "<
15.            // Linha de comentário
16.            document.write("<p><b>3) " + profissao + " = " + nome + " " +
17.        </script>
18.    </body>
19. </html>
```

Neste exemplo, estamos utilzando as variáveis nome e sobrenome, alem da constante profissao, todas trabalhando com valores string *[conjunto de caracteres]*. Vamos analisar o resultado das linhas 13, 14 e 16.

O resultado será o seguinte HTML:

A linha 13 resulta em:

1) Piloto = AyrtonSenna

Os valores nome e sobrenome estão juntos, pois os valores das variáveis não possuem espaços.

A linha 14 resulta em:

2) Piloto = Ayrton Senna

Agora há uma separação entre o nome e sobrenome, pois foi adicionado um espaco em branco entre as variáveis.

A linha 16 resulta em:

3) Piloto = Ayrton Senna

O texto é igual ao exemplo 2, a não ser pelo valor "3)" e por estar em negrito. No entanto, estamos utilizando uma constante e duas variáveis para compor esta frase.

COMENTáRIO EM JAVASCRIPT

Dica: Podemos adicionar comentários em nossos códigos Javascript. Os comentários não são executados e servem para documentar ou explicar alguma parte ou funcionalidade do seu programa. Existem duas maneiras. Os comentários em linha, onde apenas a linha é comentada.

// Linha comentada. (iniciado com duas barras no início da frase)

E existe os comentários em bloco, delimitados no inicio com /* e finalizado com */, todo o conteúdo entre eles serão comentados.

```
/*
 * Comentários em bloco.
* Várias linhas
* podem ser adicionadas.
*/
```

Vamos analisar um outro exemplo:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 3</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 3)</div>
        <script type="text/javascript">
            var produto_1 = "30.50 por cada Mouse";
            var produto_2 = "Cada Mouse por R$ 30.50";
            var mouse_1 = parseInt(produto_1);
            var mouse_2 = parseInt(produto_2);
            var mouse_3 = parseFloat(produto_1);

            document.write("<br>Mouse (Inteiro) = R$ " + mouse_1 );
            document.write("<br>Mouse (Inteiro) = R$ " + mouse_2 );
            document.write("<br>Mouse (Real) = R$ " + mouse_3 );
        </script>
   </body>
</html>
```

Verifique que a variável produto_1 e produto_2 são do tipo String e as variáveis mouse_1 e mouse_2 são do tipo inteiro, por fim a variável mouse_3 é do tipo real. Como conseguimos converter um tipo String para Inteiro e Real? Isso foi possível pela utilização das funções parseInt() e parseFloat(), onde os valores String são convertidos para inteiro e real.

Quando não for possível a conversão, a mensagem de erro NaN será exibida.

O resultado HTML deste código ficou assim:

Programação de Interfaces (aula 3)

Mouse (Inteiro) = R$ 30

Mouse (Inteiro) = R$ NaN

Mouse (Real) = R$ 30.5

Analise o código atentamente e compare com o resultado obtido.

Vamos analisar outro caso:

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Tópico 3</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 3)</div>
9.         <script type="text/javascript">
10.
11.             var a = "5";
12.             var x = a + 1 + 2;
13.             document.write("<br>x = " + x );
14.
15.             var y = a + (1 + 2);
16.             document.write("<br>y = " + y );
17.
18.             var y = parseInt(a) + (1 + 2);
19.             document.write("<br>y (parse) = " + y );
20.         </script>
21.     </body>
22. </html>
```

Observe o resultado da linha 13:

x = 512

Este valor foi obtido pois houve a concatenação da string "5" com os valores 1 + 2, assumidos também como string.

Na linha 16 obtemos o seguinte resultado:

y = 53

Este valor é a concatenação entre o valor "5" e a soma de (1+2), portanto, temos o valor 53. Neste caso, temos uma string e um inteiro.

Na linha 19 temo o resultado:

y (parse) = 8

O valor é a soma de 5 + 1 + 2, pois a string "5" está na função parseInt( ) e somado com a expressão (1+2). Assim, todos os valores são do tipo inteiro, tornando possível a soma entre os elementos.

Cuidado com as conversões de tipos!

Vamos analisar mais um exemplo:

```
1. <!DOCTYPE html>
2. <html>

3.     <head>
4.         <title>Tópico 3</title>
5.         <meta charset="UTF-8">
6.     </head>
7.     <body>
8.         <div>Programação de Interfaces (aula 3)</div>
9.         <script type="text/javascript">
10.            var vendas = 10000.50;
11.            var salario_Fixo = 3000.00;
12.            var salario_Final = salario_Fixo + (vendas * 0.15);
13.            document.write("Valor a receber = R$ " + salario_Final );
14.        </script>
15.    </body>
16. </html>
```

Neste último exemplo, temos alguns valores calculando um percentual de vendas (linha 12), somado ao salário fixo, concatenando com uma String no início da frase (linha 13).

IMPORTANTE:

Neste tópico você deve ter aprendido:

- O que são variáveis.
- Os tipos de dados primitivos.
- Como determinar a nomenclatura de variáveis.
- O que são constantes.
- Como criar e inicializar variáveis.
- Utilizar a conversão de tipos.

Caso alguns destes assuntos não tenha ficado claro, releia novamente e faça novos exercícios.

ATIVIDADE

Analise o código a seguir e escolha a opção correta da expressão:

```
var a = "5";
var y = a + 0 + (5 + 2);
document.write("n = " + y );
```

A. n = 5052
B. n = 507
C. n = 12
D. n = 57
E. n = 5072

ATIVIDADE

A nomenclatura de variáveis é uma atribuição importante. Escolha a alternativa correta sobre a nomenclatura de variavel.

```
var 1nome;
var if;
var nome completo;
var@idade
```

A. var if;
B. var nome_completo;
C. var 2nome;
D. var #idade;

ATIVIDADE

A nomenclatura de variáveis é uma atribuição importante. Escolha a alternativa correta sobre a nomenclatura de variavel.

```
var 1nome;
var if;
var nome completo;
```

var@idade

A. var if;

B. var nome_completo;

C. var 2nome;

D. var #idade;

ATIVIDADE

A utilização de comentários em código javascript é uma prática útil, pois possibilita realizar o mínimo de documentação a um código fonte. Escolha a alternativa correta que permite criar um bloco (várias linhas) de comentário.

A. **
   * Criação de variáveis - versão 1.
   **

B. _*
   * Criação de variáveis - versão 1.
   *_

C. /*
   * Criação de variáveis - versão 1.
   */

D. */
   * Criação de variáveis - versão 1.
   /*

REFERÊNCIA

MORRISON, M. Use a cabeça JavaScript. 5º Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO. C. A. J. Faça um site JavaScript orientado por projeto. 6º ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8º Ed. São Paulo: Novatec, 2010. 245 p.