JavaScript - Trabalhando com variáveis

Como já falamos anteriormente, variáveis são contêineres identificados que podem armazenar dados. Por exemplo, um número, uma string de texto, um objeto. Já sabemos também que cada variável precisa ter um nome. Há regras específicas que devemos seguir quando escolhemos o nome de uma variável:

- Os nomes de variável podem incluir letras do alfabeto, tanto letras minúsculas como maiúsculas, também podem incluir os dígitos de 0 a 9 e o caractere sublinhado (_);

- Não podem incluir espaços nem quaisquer outros caracteres de pontuação;

- O primeiro caractere do nome deve, obrigatoriamente ser uma letra ou um sublinhado;

- Letras maiúsculas e minúsculas são diferenciadas, portanto, TotalGeral, totalgeral e TOTALGERAL são nomes de variáveis distintos;

- Não existe limite no JavaScript para o comprimento dos nomes de variáveis, eles não devem ser muito curtos, mas também não devem ser muito longos de modo que consigamos digitar o mesmo nome duas vezes.

Baseando-se nestas regras, os nomes abaixo são exemplos de nomes de variáveis válidos:

TotalDeAlunos
DataDoSistema
Contador
Auxiliar

Note que eu costumo colocar sempre a primeira letra de cada palavra em maiúscula e as demais minúsculas, isso é apenas uma convenção minha, se você a achar interessante, nada impede que a use.

Variáveis Locais e Globais

O JavaScript inclui a palavra-chave var, que pode ser utilizada para declarar uma variável. Podemos omitir var em muitos casos; a variável ainda é declarada na primeira vez que atribuímos um valor a ela.

Para entender onde declarar uma variável precisamos entender o conceito de escopo. O escopo de uma variável é a área do script em que a variável pode ser utilizada. Existem dois tipos de variáveis:

- Variáveis globais: têm o script inteiro (e outros scripts no mesmo documento de HTML) como seu escopo. Podem ser utilizadas em qualquer lugar, mesmo dentro das funções.

- Variáveis locais: têm uma única função com seu escopo. Elas podem ser utilizadas somente dentro da função em que foram criadas.

Para criarmos uma variável global, a declaramos no script principal, fora de quaisquer funções. Podemos utilizar a palavra-chave var para declarar a variável, como nesse

exemplo:

var NomeDoUsuario = "Joãozinho";

Essa instrução declara uma variável chamada NomeDoUsuario e atribui um valor de Joãozinho. Se essa instrução for utilizada fora de funções, ela cria uma variável global. A palavra-chave **var** é opcional nesse caso, então essa instrução é equivalente à anterior:

NomeDoUsuario = "Joãozinho";

Podemos estão concluir o seguinte:

Uma variável local pertence a uma função particular. Qualquer variável que declaramos em uma função é uma variável local. Por exemplo, as variáveis na lista de parâmetro de uma função são variáveis locais.

Para nos certificarmos de estamos criando uma variável local dentro de uma função, podemos utilizar a palavra-chave **var**. Isso força o JavaScript a criar uma variável local, mesmo se existir uma variável global como o mesmo nome.

Pode parecer ainda meio confuso essa idéia de variáveis locais e globais, mas não se preocupe, iremos explorar durante todo o nosso estudo esse tipo de recurso. Para começar segue abaixo a listagem de um exemplo utilizando variáveis locais e globais:

```
<HTML>
<HEAD>
<TITLE>Usando Variáveis Globais e Locais</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var Nome1 = "Maria";
var Nome2 = "Francisco";

function BemVindo(NomeDoUsuario)
{
alert ("Seja Bem-vindo, " + NomeDoUsuario);
var Nome2 = "Joaquim";
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<H1>Exemplo de uso de Variáveis Globais e Locais</H1>
<P>Prepare-se para ser saudado!</P>
<SCRIPT LANGUAGE="JavaScript">
<!--
BemVindo(Nome1);
BemVindo(Nome2);
//-->
</script>
</BODY>
</HTML>
```

A listagem acima utiliza as seguintes variáveis:

- Nome1 e Nome2 são variáveis globais definidas no cabeçalho;

- NomeDoUsuario é uma variável local criada na lista de parâmetros da função BemVindo();

- Atenção: a função BemVindo() cria uma variável local denominada Nome2. Já que a palavra-chave var é utilizada, isso não afeta a variável global Nome2. (se afetasse o nome da segunda saudação mudaria).

Note que as variáveis globais são declaradas dentro do cabeçalho do documento de HTML. Na realidade, podemos declarar variáveis em qualquer script no documento, no entanto, o cabeçalho é o lugar mais indicado pelo fato de ele ser o primeiro a ser executado em um documento de HTML, não existindo assim o risco de usarmos uma variável antes de a declararmos.

Atribuindo valores a variáveis

Mesmo que ainda não soubéssemos disso, já atribuímos valores a variáveis várias vezes, vamos ver a seguir alguns exemplos:

A instrução a seguir atribuir o valor 40 à variável Linhas:

```
Linhas = 40;
```

Podemos utilizar qualquer expressão à direita do sinal de igual, incluindo outras variáveis. A instrução a seguir adiciona 1 a variável Linhas:

```
Linhas = Linhas + 1;
```

Como incrementar ou decrementar variáveis é bem comum, o JavaScript inclui dois tipos de abreviação para essa sintaxe. A primeira utiliza o operador +=:

```
Linhas += 1;
```

Da mesma forma, podemos subtrair um número de uma variável utilizando o operador -=:

```
Linhas -= 1;
```

Ainda temos outra opção para este tipo de atribuição, podemos utilizar os operadores de incremento do JavaScript, ++ e --. A instrução a seguir adiciona um valor a Linhas:

```
Linhas ++;
```

Da mesma forma, a instrução a seguir subtrai um valor de Linhas:

```
Linha --;
```

Podemos também utilizar o operador ++ ou - antes do nome de uma variável, mas neste caso o resultado será diferente. Se o operador de incremento está depois do

nome da variável, o incremento ou decremento acontece depois de a expressão atual ser avaliada. Se o operador está antes do nome da variável, o incremento ou decremento acontece antes de a expressão atual ser avaliada. As seguintes instruções têm efeitos diferentes:

alert(Linhas++); alert(++Linhas);

A primeira instrução exibe um alerta com o valor 40 e então incrementa Linhas para 41. A segunda instrução primeiro incrementa Linhas para 41, depois exibe um alerta com o valor 41.

Tipos de dados no JavaScript

O JavaScript não exige que declaremos o tipo de dado de cada variável, mas para o nosso próprio bem, é muito importante conhecermos os tipos de dados que ele pode tratar. Estes são os tipos de dados básicos do JavaScript:

- **Número:** como 3, 25, 4,3542. O JavaScript suporta tanto números inteiros como pontos flutuantes;

- **Valores Booleanos ou Lógicos:** Estes podem ter um de dois valores: **verdadeiro ou falso**.

- **Strings:** Como "Maria tinha um carneirinho". Estas consistem em um ou mais caracteres de texto.

- **Valor Nulo:** representado pela palavra-chave **null**. Esse é o valor indefinido de uma variável.

Embora o JavaScript monitore o tipo de dados atualmente armazenado em cada variável, ele não impede você de alterar os tipos intermediários. Por exemplo, suponha que você declarou uma variável atribuindo-lhe um valor:

Total = 31;

Essa instrução declara uma variável chamada Total e lhe atribui o valor 31. Essa é uma variável numérica. Agora suponha que você alterou o valor Total:

Total = "Salgadinho";

Isso atribui um valor de string à soma. JavaScript não exibirá um erro quando essa instrução for executada; ela é perfeitamente válida, embora provavelmente não seja útil.

Conversão de valores

O JavaScript trata conversões entre tipos de dados sempre que pode. Por exemplo, quando utilizamos instruções desse tipo:

document.write("Total de Alunos: " + Total);

Essa instrução imprime uma mensagem como "Total de Alunos: 38". Neste caso o interpretador de JavaScript automaticamente converteu o conteúdo da variável Total

para string para que ele pudesse ser mostrado na página.

Isso funciona igualmente bem com valores de ponto flutuante e booleanos. Entretanto, há algumas situações onde não funcionará. Por exemplo, a seguinte instrução funcionará bem se o valor de Total for numérico, por exemplo, 40:

Media = Total / 3;

Entretanto, a variável Total também pode conter uma string; nesse caso, a instrução acima resultaria um erro.

Em algumas situações, podemos terminar com uma string contendo um número e precisaríamos convertê-la em uma variável numérica regular. O JavaScript inclui duas funções para esse propósito:

- parseInt() - Converte uma string em um número inteiro;

- parseFloat() - Converte uma string em um número de ponto flutuante.

Por exemplo, a instrução seguinte converte uma string em um valor inteiro:

```
MinhaString = "30 carneirinhos tinha Maria";
MeuNumero = parseInt(MinhaString);
```

Depois que essas instruções forem executadas, a variável MeuNumero armazenará o valor 30. A parte não numérica da string é ignorada.

Armazenando Dados do Usuário em variáveis

Uma utilização comum de variáveis é armazenar as informações que vêm do usuário. Como um exemplo, agora iremos criar um script que solicita informações ao usuário e cria um documento de HTML contendo essas informações.

Primeiramente solicitaremos um primeiro nome, um sobrenome e um título para a página:

```
PrimeiroNome = prompt("Digite o seu primeiro Nome:");
Sobrenome = prompt("Digite o seu sobrenome:");
TituloPagina = prompt("Digite um Título para a Página:");
```

Agora podemos utilizar o conteúdo das variáveis para personalizar o documento de HTML:

```
document.write("<h1>" + TituloPagina + "</h1>");
document.write("<h2>Usuário: " + PrimeiroNome + " " + Sobrenome + "</h2>"
```

Para completar esse script basta agora adicionarmos as tags <SCRIPT> normais e uma estrutura básica de HTML, como mostra a listagem a seguir:

```
<HTML>
<HEAD>
<TITLE> Personalizando uma Página </TITLE>
</HEAD>
```

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--
PrimeiroNome = prompt("Digite o seu primeiro Nome:");
Sobrenome = prompt("Digite o seu sobrenome:");
TituloPagina = prompt("Digite um Título para a Página:");


document.write("<h1>" + TituloPagina + "</h1>");
document.write("<h2>Usuário: " + PrimeiroNome + " " + Sobrenome + "</h2>");


//-->
</script>
<P>Obrigado por visitar minha Página.</P>
</BODY>
</HTML>
```