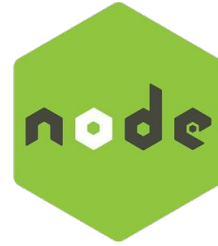


NodeJS



Prof. Celso Henrique Masotti

Necessidades Prévias

- Ter o node.JS instalado;
- Ter o gerenciador de pacotes (NPM) instalado;
- Ter um editor de códigos IDE^(*), dê preferência ao “[Visual Studio Code](#)”
- Conhecer JavaScript básico.

(*) IDE, do inglês Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo, como um editor de texto para a digitação de códigos.

Você verá nesta aula:

Objetivo: Uso de variáveis, constantes, comentários, concatenação de elementos no output e código de “escape”.

[Variáveis e Constantes](#)

[Case Sensitive](#)

[Tipagem](#)

[Tipagem Dinâmica](#)

[Variável: Usabilidade e Abrangência](#)

[Comentário](#)

[Concatenação](#)

[Códigos de “escape”](#)

[Praticando](#)

Variáveis e Constantes

Podemos armazenar informações na memória do computador. Para isso é preciso dar um nome a um determinado espaço de memória.

Quando o conteúdo existente em um espaço de memória se modifica durante a execução do programa, denominamos este espaço de “variável”. Quando o conteúdo não se modifica chamamos de “constante”.

Na linguagem javascript a nomenclatura para uma variável vem depois do comando “var”. exemplo: var [nome do espaço de memória].

Da mesma forma a nomenclatura para uma constante vem depois do comando “const”. exemplo: const [nome do espaço de memória].

Embora a diferença de comandos para a criação dos tipos de espaços de memória, no dia a dia os desenvolvedores só fazem uso do comando “var”.

```
var    nomePF  
const  pi
```

Case Sensitive

Case Sensitive significa algo como “sensível à caixa das letras” ou “sensível a maiúsculas e minúsculas”. Isto significa que a denominação do espaço de memória - seja variável ou constante - deve ser observada com muita atenção. Observe:

“var Nome” não é igual a “var nome”.

```
proj01 > JS var.js > ...  
1  var nome = "Maria"  
2  console.log(Nome)
```

ERRADO:

o código acima está errado pois é solicitado a impressão do conteúdo da variável “Nome” e esta não existe.

```
proj01 > JS var.js > ...  
1  var nome = "Maria"  
2  console.log(nome)
```

CERTO:

o código acima está correto pois é solicitado a impressão do conteúdo da variável “nome” e esta de fato existe.

Tipagem

Um espaço de memória (variáveis ou constantes) recebem alguns tipos de informação:

- **String** — Texto;
- **Numbers** — São os números, seja eles integer (123), float (123,55), double etc.
- **Booleans** — São os operadores booleanos (*true* ou *false*)
- **Arrays** — É uma estrutura de dado para armazenar uma coleção de valores, sendo eles de qualquer tipo.
- **Objects** — Conjunto de atributos aninhados a uma variável denomina-se um objeto.
- **Functions** — Em JavaScript é possível declarar uma variável como uma função, podendo fazer operações e retornando o valor para a variável de declaração. Obs: muito utilizado no paradigma de programação funcional.

Tipagem Dinâmica

Na linguagem javascript não precisamos informar que tipo de variável ou constante estamos produzindo - como é comum em algumas linguagens. Basta apenas criá-las e a tipagem é realizada na máquina automaticamente (Tipagem Dinâmica).

Exemplos:

```
const pi = 3.14159265359  
var nome = "Maria"  
var v1 = 325  
var v2 = 45.87  
var gt = True
```

“pi” é uma constante tipo number [float] (pois tem casas decimais)

“nome” é variável tipo string (obrigatório as aspas);

“v1” é variável number [inteiro]

“v2” é variável tipo number [float]

“gt” é variável tipo booleana

*Tipagem **number float** é feita usando ponto como divisor de centavos e nunca vírgula.*

Variável: Usabilidade e Abrangência

As variáveis recebem denominações diferentes quanto à sua usabilidade e capacidade de abrangência. Estas são: locais, de sessão e globais.

- **Locais:** são variáveis e constantes que são usadas rotineiramente em um sistema e sua usabilidade está restrita apenas a soluções imediatas não possuindo memória compartilhada;
- **Sessão:** trata-se de um tipo de variável que armazena informações que alimentam oportunamente uma série de ações no sistema praticamente inteiro, como permissões, dados para envio de e-mail, etc. Normalmente é aplicada em sistema onde o login e senha são exigidos para acesso.
- **Globais:** trata-se de uma variável acessível em todos os escopos de um programa de computador. O mecanismo de interação com variáveis globais é chamada ambiente global.

Comentário

Um comentário é uma parte de código que não é interpretada pelo navegador. Pode ser utilizada, também, para colocar textos que ajudam a identificar o que está sendo feito naquele trecho de código. Pensando num trabalho em equipe, esta é uma excelente prática.

Existem dois tipos de comentários:

– **barra dupla:** serve para comentar uma linha de código.

```
1  var nome = "Maria"
2  // este espaço de memória é uma variável tipo string
3  console.log(nome)
```

```
//console.warn("Este aviso não será mostrado");
```

– **barra asterisco:** para comentar várias linhas. Deve-se iniciar com /* e terminar com */.

/* Exemplo de comentário para
várias linhas.

Basta escolher o início e fim. */

```
1  var nome = "Maria"
2  // este espaço de memória é uma variável tipo string
3  console.log(nome)
4  /* console é um
5  objeto com muitas
6  propriedades */
```

Concatenação

Denominamos “concatenação” a ação de imprimir unindo dois ou mais elementos, sendo estes textos, variáveis ou constantes. Para concatenar fazemos o uso do sinal de adição.

exemplo:

```
console.log(nome + " " + sobrenome)
```

Resultado:

```
C:\NodeEstudos\proj01>node var.js  
Maria Carmo  
  
C:\NodeEstudos\proj01>
```

```
1  var nome = "Maria"  
2  var sobrenome = "Carmo"  
3  console.log(nome + " " + sobrenome)  
4  /* o símbolo de adição concatena (une)  
5  os elementos variáveis e string */
```

Códigos de “escape”

Existem determinadas marcações que possuem a capacidade de realizar determinadas ações. Estas marcações são chamadas de códigos de “escape”.

\0 O caractere NUL (\u0000)

\b Backspace (\u0008)

\t Tabulação horizontal (\u0009)

\n Nova linha (\u000A)

\v Vertical tab (\u000B)

\f Alimentação de formulário (\u000C)

\r Retorno do carro [início de linha] (\u000D)

\" Imprimir aspas duplas (\u0022)

' Imprimir aspas simples (\u0027)

**** Imprimir barra invertida (\u005C)

Os códigos de “escape” são textos, portanto é preciso usá-las entre aspas.

```
1 var t1 = "Estrada"
2 var t2 = "de"
3 var t3 = "Itapecerica"
4 console.log(t1 + "\n" + t2 + "\n" + t3)
```

```
C:\NodeEstudos\proj01>node var.js
Estrada
de
Itapecerica
```

Praticando

- 1 - Crie um arquivo denominado “nome.js” contendo duas variáveis tipo string. Exiba-as concatenadas de tal forma que no console apareça seu nome completo.
- 2 - Crie um arquivo denominação “brasil.js” contendo uma variável para cada parte do texto “República Federativa do Brasil”. Exiba uma palavra em cada linha no console.
- 3 - Pegue as primeiras quatro linhas da letra de uma música que gosta. Usando variáveis, imprima a letra no console.
- 4 - Crie a variável “v1” e “v2” que contenham valor tipo number [integer]. Exiba seus valores e o resultado da soma de tal forma que fique algo como:

```
C:\NodeEstudos\proj01>node var.js
12
64
76
```

Obrigado

Profº Celso Henrique Masotti