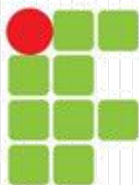


# Algoritmos (11006009)

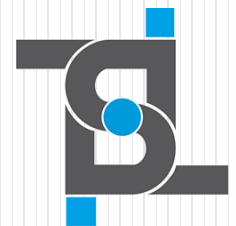
## Aula 3 – Variáveis, Constantes e parâmetros

Bruno B. Boniati

[bruno.boniati@iffarroupilha.edu.br](mailto:bruno.boniati@iffarroupilha.edu.br)



**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
**FARROUPILHA**  
Câmpus Frederico Westphalen



**Sistemas para  
Internet**

# Variáveis o que são?

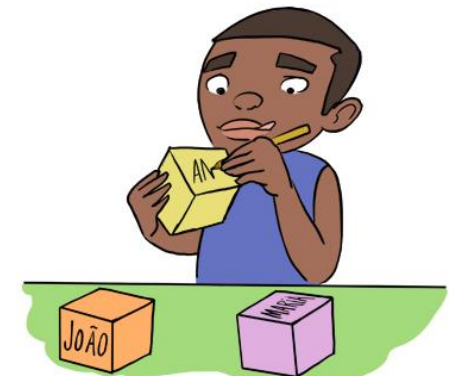
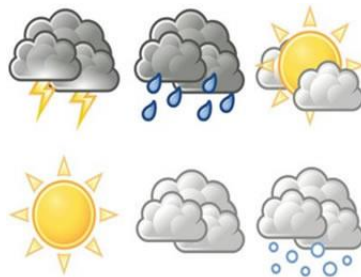
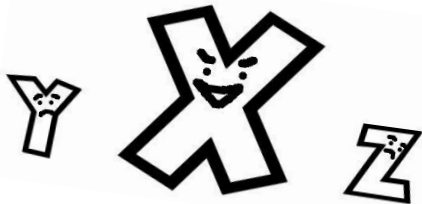


## Para a estatística ...

- Atributo, mensurável ou não, sujeito à variação quantitativa ou qualitativa, no interior de um conjunto

## Para a matemática...

- incógnita, em geral representado por uma letra, que pode assumir diferentes valores numéricos em uma expressão algébrica, numa fórmula ou num algoritmo.



$$\begin{array}{r} 4x + \cancel{2y} = 10 \\ + \quad 3x - \cancel{2y} = 11 \\ \hline 7x = 21 \end{array}$$

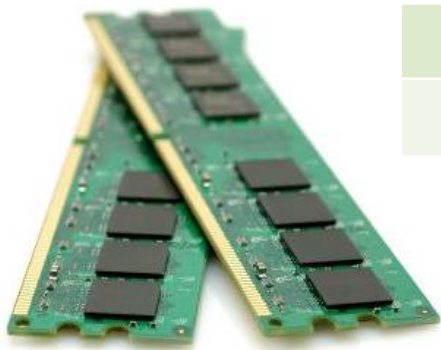
# Variável

“objeto situado na memória que representa um valor ou expressão”

“símbolo do programa que refere-se a um endereço de memória”

```
var idade = 25
```

```
var sexo = "M"
```



Endereço	Variável
0x45F1F8	idade
0x45A12E	sexo

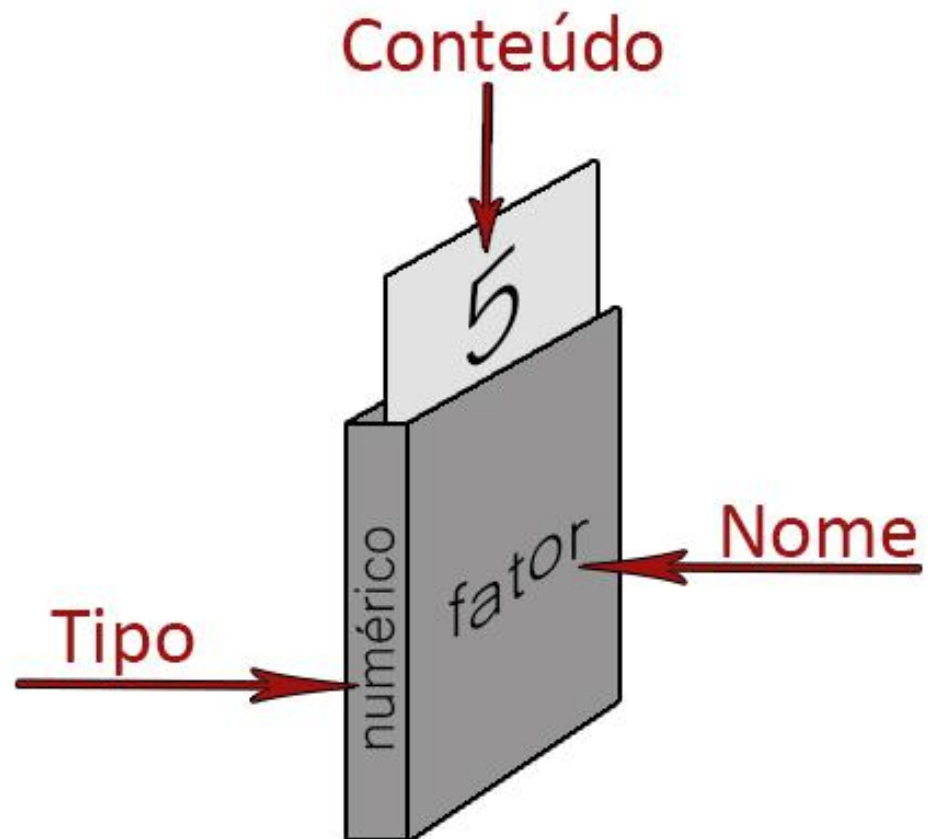


Softwares precisam transformar  
dados em informações

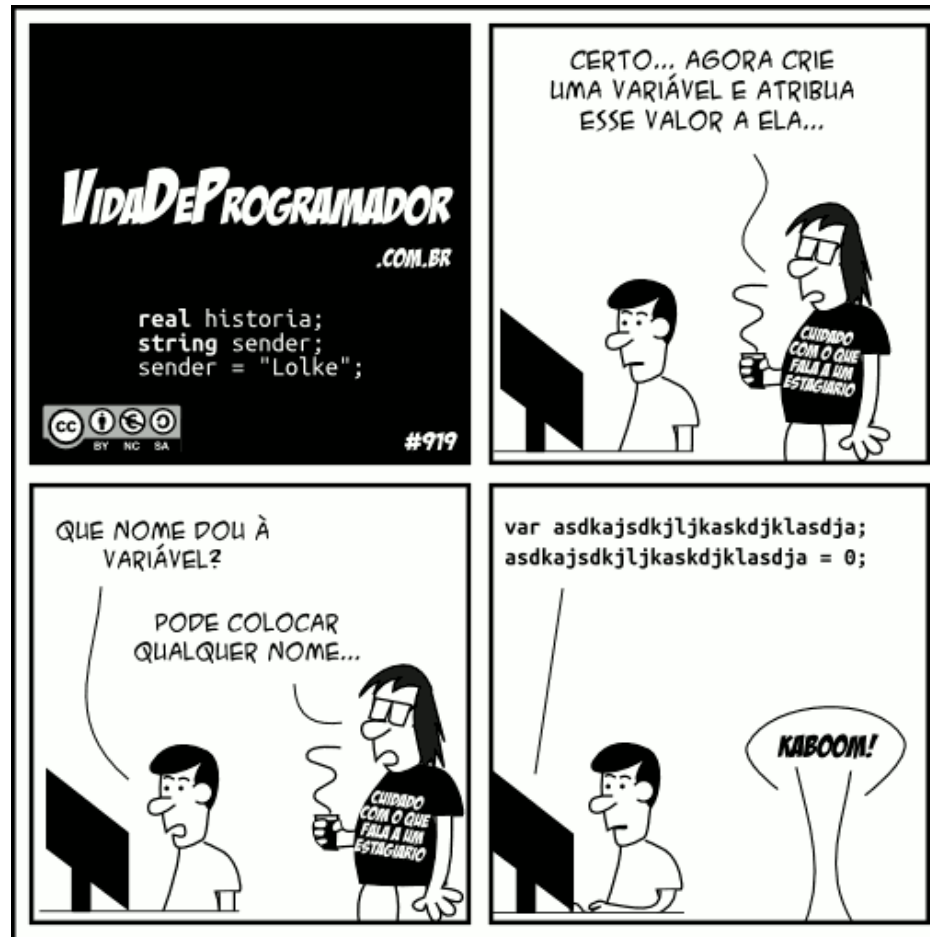
A memória é um dispositivo “endereçável”  
(os endereços são valores em hexadecimal)  
Ex. 0x45F1F8, 0x45A12E

# Variáveis devem ter ...

- Nome
- Tipo
- Conteúdo



# Nomes das variáveis...



# Qual nome minha variável terá?

- Procure utilizar nomes significativos, onde através de seu nome seja possível imaginar seu conteúdo.
- Nomes de variáveis **não podem:**
  - Começar com números (ex. 1ano)
  - Conter caracteres especiais que tem outro significados para a linguagem de programação (ex. \$, %, &, ;, @, etc.)
  - Conter espaços em branco entre palavras (ex. nome completo)
  - Conter palavras acentuadas ou sinais de pontuação.
  - Utilizar palavras reservadas.

# Palavras reservadas do JavaScript

- |             |                |            |
|-------------|----------------|------------|
| 1. break    | 11. finally    | 21. this   |
| 2. case     | 12. for        | 22. throw  |
| 3. catch    | 13. function   | 23. true   |
| 4. continue | 14. if         | 24. try    |
| 5. debugger | 15. in         | 25. typeof |
| 6. default  | 16. instanceof | 26. var    |
| 7. delete   | 17. new        | 27. void   |
| 8. do       | 18. null       | 28. while  |
| 9. else     | 19. return     | 29. with   |
| 10. false   | 20. switch     |            |

# Palavras reservadas pelo Browser

- |                |                 |
|----------------|-----------------|
| 1. alert       | 12. navigator   |
| 2. blur        | 13. open        |
| 3. closed      | 14. outerHeight |
| 4. document    | 15. outerWidth  |
| 5. focus       | 16. parent      |
| 6. frames      | 17. screen      |
| 7. history     | 18. screenX     |
| 8. innerHeight | 19. screenY     |
| 9. innerWidth  | 20. statusBar   |
| 10. length     | 21. window      |
| 11. location   |                 |



# Palavras reservadas para “propósitos futuros”

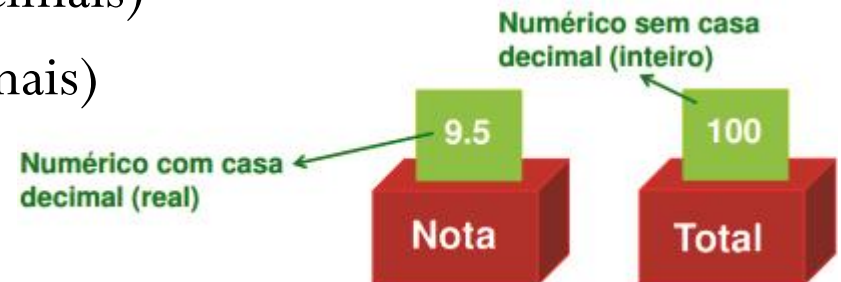
- |             |                |                  |
|-------------|----------------|------------------|
| 1. abstract | 12. float      | 23. protected    |
| 2. boolean  | 13. goto       | 24. public       |
| 3. byte     | 14. implements | 25. short        |
| 4. char     | 15. import     | 26. static       |
| 5. class    | 16. int        | 27. super        |
| 6. const    | 17. interface  | 28. synchronized |
| 7. double   | 18. let        | 29. throws       |
| 8. enum     | 19. long       | 30. transient    |
| 9. export   | 20. native     | 31. volatile     |
| 10. extends | 21. package    | 32. yield        |
| 11. final   | 22. private    |                  |

# Tipos das variáveis...



# Tipos podem ser...

- **String** (cadeias de caracteres)
  - Qualquer valor entre aspas
- **Lógico** (verdadeiro ou falso)
  - true ou false
- **Numéricos**
  - Inteiros (valores sem casas decimais)
  - Reais (valores com casas decimais)



# Para praticar ...

- Deseja-se informatizar um formulário para inscrição de interessados em fazer um intercâmbio. Quais informações são necessárias e de que tipo elas seriam?



Contato(s)  
Telefônico(s)



Meses que ficará fora



Endereço



Data de nascimento



E-mail



Documentos  
(Ex. Cpf, Identidade)



Renda Mensal



Outro contatos



Fala inglês



Sexo (gênero)

# Escopos das variáveis

- **Variável global**

- Variável cujo valor pode ser lido por qualquer subprograma (function) do código.
- **Atenção:** Em JavaScript, quando a variável é declarada sem o uso do “var” ela é automaticamente considerada uma variável GLOBAL.
- Seu uso deve ser evitado ou moderado, pois o excesso de variáveis globais dificulta a reutilização do código.

- **Variável local**

- Variável que é declarada dentro de um subprograma (function) e cujo conteúdo somente pode ser acessado pelo subprograma.

# Escopos das variáveis

```
<script>
```

```
var numero = 1
```

```
function mostraNumero() {
```

```
var nome = "Bruno"
```

```
instituicao = "IFF-FW"
```

```
  alert (numero)
```

```
  alert (nome)
```

```
}
```

```
</script>
```

Variável Global

Variável Local

Neste caso a variável **instituicao** é global pois não foi utilizado o **var** para declará-la.

# Constantes

- Assim como variáveis, constantes também são símbolos utilizados pelo programador para “apelidar” endereços de memória.
- A diferença da constante em relação à variável é o valor de uma constante, uma vez atribuído, não é mais alterado.
- Normalmente são declaradas em um escopo global e tratam de configurações do sistema que devem ser consideradas por todas as partes de um programa.

# Constantes (cont.)

- Em JavaScript uma constante é declarada com o uso da palavra reservada **const**;
- Não é incomum que programadores convençionem o uso de constantes em letras maiúsculas;
- Trata-se de uma especificação recente, que não faz parte da linguagem JavaScript original e que pode não ser suportada por todos os navegadores.



# Tipos de dados em JavaScript

- JavaScript, assim como outras linguagens interpretadas (ex. PHP e Python) utiliza o conceito de “tipagem dinâmica”, que significa que as variáveis não precisam ter um tipo fixo.
- O tipo de uma variável pode ser alterado ao longo da execução do programa e está relacionado diretamente ao seu conteúdo.
- O comando **typeof** testa o tipo da variável a partir de seu conteúdo, podendo assumir os seguintes valores:
  - string
  - number
  - boolean
  - object
  - undefined

# Teste você mesmo ...

```
<body>
  <script>
    var nome = "Bruno"
    var letras = 5
    var altura = 1.70
    var ehGremista = true
    var foto = document.getElementById("figura")
    var pensamento

    document.writeln(typeof(nome))      //string
    document.writeln(typeof(letras))    //number
    document.writeln(typeof(altura))    //number
    document.writeln(typeof(ehGremista))//boolean
    document.writeln(typeof(foto))      //object
    document.writeln(typeof(pensamento))//undefined

  </script>
  
</body>
```

# Conversões entre tipos

- Em geral dados coletados de formulários HTML ou mesmo dados provenientes de comandos JavaScript (como prompt) são tratados como strings.
- Para fazer operações matemáticas é necessário converter texto (string) em números (inteiros ou reais).
  - `parseInt("10")` // converte uma string em um valor inteiro
  - `parseFloat("56.36")` // converte uma string em um valor real
- A conversão inversa (numérico para string) é direta, bastando concatenar ao número qualquer valor entre “aspas”

# Parâmetros

- Parâmetros são dados passados para os subprograms (functions) permitindo a troca de informações entre eles sem a necessidade de variáveis globais.
- Vamos a um exemplo:
  - Observe no exemplo a seguir que temos dois procedimentos que fazem coisas muito parecidas ... Basicamente o comportamento de ambos é idêntico, só muda o conteúdo da mensagem.

```
function mensagemBoasVindas() {  
    alert("Seja bem vindo")  
}
```

```
function mensagemFinalizacao() {  
    alert("Sua seção foi encerrada!")  
}
```

# Parâmetros (cont.)

- Para resolver a duplicação do código podemos parametrizar a mensagem, veja...

```
function mensagem(msg) {  
    alert(msg)  
}
```

```
mensagem("Seja bem vindo")
```

```
mensagem("Sua seção foi encerrada!")
```

- Parâmetros tem escopo local, ou seja, se comportam como variáveis locais (só tem validade dentro do subprograma onde foram definidos)

## Para saber

# Formas de se usar JavaScript no HTML

1. Colocando-se o código entre a *tag* `<script>` e a *tag* `</script>`
2. Incluindo um arquivo externo no atributo `src` da *tag* `<script>`

```
<!DOCTYPE HTML>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8">
    <title></title>
    <script language="JavaScript" src="OlaMundo.js"></script>
  </head>
  <body>
    <script language="JavaScript">
      document.writeln("Hello World")
    </script>
  </body>
</html>
```

2

1

# Desafio – Coletando Informações

- Para colocar em prática o exercício dessa aula, a atividade a ser desenvolvida consiste na construção de uma página HTML para coletar informações a respeito de algum formulário (ex. doador de sangue, inscrições para um campeonato, ficha de cadastro, etc.)
- Você deve utilizar no mínimo uma variável de cada tipo (string, inteiro, real e lógico) e bem como controles visuais compatíveis com as variáveis. Ao clicar em um botão mostre uma mensagem de confirmação com os dados coletados.

