



# Desenvolvimento para dispositivos móveis

Prof. Armando Mendes Neto.  
armando.mendes@ifc.edu.br

Comentários e variáveis.

# Conteúdo da aula

- ▶ Comentários em JavaScript.
- ▶ Variáveis - identificadores.
- ▶ Variáveis - tipos de dados.
- ▶ Variáveis - tipos de declarações.

# JavaScript - Comentários

- ▶ `//` Este é um comentário de uma só linha

`// O professor Armando torce para um time TOP, o Vasco!!`

- ▶ `/*` Este é um comentário de várias linhas `*/`

`/*`

`O Flamengo perdeu para o Vascão no último jogo.`

`*/`



# JavaScript - Comentários

- Vamos testar com o código da aula passada (Aula1.html)

```
<script>  
    window.alert("Bem vindo ao SISBEL")  
    window.confirm("Você está gostando o site?")  
    window.prompt("Deixe o seu e-mail para contato")  
</script>
```

- Note que o código em JavaScript está entre as tags `<script>` `</script>`

**ATENÇÃO:** Comentários em HTML: use a tag `<!-- -->`

# JavaScript - Comentários

Abra o arquivo **Aula1.html** e comente os métodos:

- Alert
- Confirm
- Prompt

Descrever o que cada método faz (caderno)

# Variáveis - Definição

- ▶ Elas armazenam valores que são manipulados por nossos programas, permitindo que os algoritmos sejam implementados para dar a resposta esperada para o usuário.
- ▶ Variáveis são **espaços de memória** do computador destinados a dados que são manipulados e alterados durante a execução de um algoritmo.

# Variáveis - Identificadores

- ▶ Podem começar com letra, \$ ou \_.
- ▶ NÃO podem começar com números.
- ▶ É possível utilizar letras ou números.
- ▶ É possível utilizar acentos ou símbolos.
- ▶ NÃO podem conter espaços.
- ▶ NÃO podem ser palavras reservadas.

# Variáveis - Observações

- ▶ Maiúsculas e minúsculas fazem diferença (case-sensitive).
  - ▶ Ex: MÉDIA é diferente de média.
- ▶ Escolher nomes coerentes para as variáveis.
  - ▶ Ex: nome, salário, endereço.
  - ▶ NÃO utilizar a, b, c, d.
  - ▶ Não utilizar x1, x2, x3.



# JavaScript - Tipos de dados

- ▶ JavaScript é uma linguagem **dinamicamente tipada**.
- ▶ Isso significa que você **não precisa especificar o tipo de dado** de uma variável quando declará-la.
- ▶ Tipos de dados são convertidos automaticamente conforme a necessidade **durante a execução do script**.

# JavaScript - Tipos de dados

- ▶ O JavaScript, tem como tipos primitivos embutidos na estrutura básica da linguagem:
  - ▶ **Number:** Inteiros e reais, são do tipo **number**.
  - ▶ **Booleanos:** **True / False**
  - ▶ **Strings**
    - ▶ Utilizamos strings com **aspas duplas, simples ou crase**.

Aula 2: teste.js

```
let a='java'  
let b="java2"  
let c=`java3`  
  
console.log(a)  
console.log(b)  
console.log(c)
```

**Obs:** Temos mais tipos de dados, mas iremos focar nos principais utilizados.

# JavaScript - Tipos de dados

- ▶ Vamos abrir o Visual Stúdio Code e fazer alguns testes.
- ▶ Abra a pasta chamada “Aula 2”.
- ▶ Crie um arquivo chamado `variáveis.js`.

# JavaScript - Tipos de dados

- Declare a seguinte variável:

Podemos declarar uma variável com o `var`, `let` ou `const`.

```
var nome="Armando";
```

O `=` atribui um valor ao operando à esquerda (variável) baseado no valor à direita.

- Nós **não** precisamos dizer para o JavaScript que nome é uma String, ele já sabe que é!

Declare a variável acima e depois, dê o comando:

```
console.log(typeof nome);
```

# JavaScript - Tipos de dados

`console.log(typeof nome);`

**console.log:** Exibe uma mensagem na console do navegador.

**typeof:** retorna uma string informando o tipo de um operando.

# JavaScript - Tipos de dados

- ▶ Agora, abaixo do comando `console.log`, atribua o seguinte valor a variável `nome`:

```
var nome=1.457879;
```

- ▶ E execute o comando novamente:

```
console.log(typeof nome);
```

# JavaScript - Tipos de dados

```
var nome="Armando";  
console.log(typeof nome);  
  
var nome=1.457879;  
console.log(typeof nome);
```



```
string  
number
```

# JavaScript - Tipos de dados

- E se declararmos só o nome da variável, sem atribuir nenhum valor a ela?

```
var nome;
```

```
console.log(typeof nome);
```



# JavaScript - Tipos de dados

```
var nome;  
console.log(typeof nome);
```



undefined



**undefined:** representa um valor indefinido para a variável no momento da execução.

# JavaScript - Tipos de dados

- ▶ Agora vamos atribuir um **valor booleano** a variável nome.

```
var nome;  
console.log(typeof nome);  
  
nome=true;  
console.log(typeof nome);
```



```
undefined  
boolean
```

# JavaScript - Tipos de dados

**document.write:** o método grava a sequência de caracteres em um documento aberto.

- Agora, vamos utilizar o comando **document.write** no arquivo da aula passada (Aula1.html):

```
<script>
  window.alert("Bem vindo ao SISBEL")
  window.confirm("Você está gostando o site?")
  window.prompt("Deixe o seu e-mail para contato")
</script>
```

**document.write**("Esse comando está exibindo uma mensagem na tela em JAVASCRIPT");

# JavaScript - Tratamento de dados

- ▶ Agora vamos criar uma pasta Aula 3
- ▶ Com o arquivo Aula3.html.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aula3</title>
</head>
<body>
  <h1>Tratamento de dados</h1>
  <script>
    window.prompt("Digite o seu nome");
  </script>
</body>
</html>
```

# JavaScript - Tratamento de dados

- ▶ Até o momento, estamos digitando o nome do usuário, porém **sem guardar dado algum**.
- ▶ Para armazenarmos algo temporariamente durante a execução de um programa, utilizamos **variáveis**.
- ▶ Então, vamos criar uma variável chamada **nome**, recebendo o que o usuário digitou no prompt de comando.
- ▶ **var nome = window.prompt("Digite o seu nome");**

Aconteceu algo????

# JavaScript - Tratamento de dados

- ▶ A variável armazenou o dado digitado pelo usuário, porém não mostrou na tela.
- ▶ Vamos testar o método `window.alert`, para dar boas vindas ao usuário:

```
<script>  
    var nome = window.prompt("Digite o seu nome");  
    window.alert("Olá " + nome + ", seja bem vindo");  
</script>
```

# JavaScript - Tratamento de dados

- ▶ Agora, vamos pedir o seguintes dados ao usuário, com uma posterior mensagem com os dados digitados:
- ▶ Nome
  - ▶ Exemplo: Digite o seu nome: -> “Olá João, seja bem vindo”.
- ▶ Idade
  - ▶ Exemplo: Digite a sua idade: -> “João, você tem 20 anos de idade”.
- ▶ Endereço
  - ▶ Exemplo: Digite o seu endereço: -> “João, você tem 20 anos de idade e mora na Avenida Getúlio Vargas 1634”.

# JavaScript - Tratamento de dados

```
<script>  
  var nome = window.prompt("Digite o seu nome");  
  window.alert("Olá " + nome + ", seja bem vindo");  
  var idade = window.prompt("Digite a sua idade");  
  window.alert(nome + " você tem " + idade + " anos de idade");  
  var endereço = window.prompt("Digite o seu endereço");  
  window.alert(nome + " você tem " + idade + " anos de idade" + " e mora na " + endereço);  
</script>
```



# JavaScript - Tratamento de dados

- ▶ Agora, vamos pedir o seguintes dados ao usuário, com uma posterior mensagem com os dados digitados:

Crie um arquivo chamado [soma.html](#)

- ▶ Nome
- ▶ Num1
- ▶ Num2
- ▶ Num3
- ▶ Calcular a soma e informar ao usuário

# JavaScript - Tratamento de dados

```
<script>
  var nome = window.prompt("Digite o seu nome: ");
  var n1 = window.prompt("Digite o primeiro número: ");
  var n2 = window.prompt("Digite o segundo número: ");
  var n3 = window.prompt("Digite o terceiro número: ");
  var soma = (n1+n2+n3)
  window.alert(nome + " a sua soma foi " + soma);
</script>
```

E a Soma? Deu certo?

# JavaScript - Tratamento de dados

- ▶ O método `prompt` lê o que foi digitado como uma `String`. Então, temos que converter para um número inteiro ou real.
- ▶ Quando lemos algo como `String`, o sinal de `+` é visto como concatenação.
- ▶ Ex: `"1" + "2" + "3" = 123`.

# JavaScript - Tratamento de dados

- ▶ Para convertermos os valores Strings lidos anteriormente, utilizamos:
- ▶ `parseInt(n)` → conversão para número inteiro
- ▶ `parseFloat(n)` → conversão para ponto flutuante.

```
var n1 = parseInt(window.prompt("Digite o primeiro número: "));
```

```
var n1 = parseFloat(window.prompt("Digite o primeiro número: "));
```

# JavaScript - Tratamento de dados

```
<script>
  var nome = window.prompt("Digite o seu nome: ");
  var n1 = parseFloat(window.prompt("Digite o primeiro número: "));
  var n2 = parseFloat(window.prompt("Digite o segundo número: "));
  var n3 = parseFloat(window.prompt("Digite o terceiro número: "));
  var soma = (n1+n2+n3)
  window.alert(nome + " a sua soma foi " + soma);
</script>
```

# JavaScript - Tratamento de dados

- ▶ Agora, vamos pedir o seguintes dados ao usuário, com uma posterior mensagem com os dados digitados:

Crie um arquivo chamado [media.html](#)

- ▶ Nome
- ▶ Nota1
- ▶ Nota2
- ▶ Nota3
- ▶ Nota4
- ▶ Calcular a média e informar ao usuário

# JavaScript - Tratamento de dados

```
<script>
  var nome = window.prompt("Digite o seu nome: ");
  var nota1 = Number(window.prompt("Digite a primeira nota: "));
  var nota2 = Number(window.prompt("Digite a segunda nota: "));
  var nota3 = Number(window.prompt("Digite a terceira nota: "));
  var nota4 = Number(window.prompt("Digite a quarta nota: "));
  var media = (nota1+nota2+nota3+nota4)/4
  window.alert(nome + " a sua média foi " + media);
</script>
```

Nas versões atuais do JavaScript, podemos utilizar somente o **Number()**, no lugar do **parseFloat** ou **parseInt**.

# JavaScript - Tratamento de dados

- ▶ Quando trabalhamos com Strings, podemos utilizar alguns recursos para facilitar a visualização, utilizando a **crase** e **`${variável}`**. Esse recurso é chamado de **template string**.

Para concatenação, utilizávamos o “+”

```
window.alert(nome + " você tem " + idade + " anos de idade e mora na " + endereço);  
  
window.alert(`${nome} você tem ${idade} anos de idade e mora na ${endereço}`);
```

Podemos utilizar a **crase** ``` e **`${variável}`**

- ▶ Crie uma pasta Aula4 e um arquivo “aula4.html” (copiando o arquivo aula3.html) e faça o teste.



# JavaScript - Tratamento de dados

► Podemos utilizar alguns métodos interessantes também:

► `variavel.length`

► Determina quantas letras tem a string digitada.

► `variavel.toUpperCase()`

► Deixa a string em MAIÚSCULA

► `variavel.toLowerCase()`

► Deixa a string em MINÚSCULA

Crie um arquivo `TesteString.html` e utilize os 3 recursos.  
Atenção: podemos utilizar tanto `document.write`, como no `window.alert`.  
Faça o teste

# JavaScript - Tratamento de dados

```
<script>
  var nome = window.prompt(`Digite o seu nome:`)
  window.alert(`${nome}, seu nome tem ${nome.length} letras`);
  document.write(`${nome}, seu nome tem ${nome.length} letras<br/>`);
  window.alert(`${nome}, seu nome em maiúsculo é ${nome.toUpperCase()}`);
  document.write(`${nome}, seu nome em maiúsculo é ${nome.toUpperCase()}<br/>`);
  window.alert(`${nome}, seu nome em minúsculo é ${nome.toLowerCase()}`);
  document.write(`${nome}, seu nome em minúsculo é ${nome.toLowerCase()}<br/>`);
</script>
```

# Pesquisa

- ▶ Pesquisar a diferença entre bibliotecas e frameworks JavaScript.
- ▶ Cite 3 bibliotecas utilizadas em JavaScript, com suas vantagens e desvantagens.
- ▶ Cite 2 frameworks utilizados em JavaScript, com suas vantagens e desvantagens.