

Lógica de Programação com Java Script

Othon Oliveira

SENAC - PROA



Sumário

- 1 Apresentação
- 2 Introdução
- 3 Operadores lógicos
- 4 Estruturas de decisão



Apresentação

Quem sou e objetivo da aula

Othon L. T de Oliveira

Professor:



Apresentação

Quem sou e objetivo da aula

Othon L. T de Oliveira

Professor:

- Ensina: Introdução a Programação, Lógica de Programação, e outras disciplinas



Apresentação

Quem sou e objetivo da aula

Othon L. T de Oliveira

Professor:

- Ensina: Introdução a Programação, Lógica de Programação, e outras disciplinas

Trabalho:

Analista de Infraestrutura: Reitoria da UPE



Apresentação

Quem sou e objetivo da aula

Othon L. T de Oliveira

Professor:

- Ensina: Introdução a Programação, Lógica de Programação, e outras disciplinas

Trabalho:

Analista de Infraestrutura: Reitoria da UPE

Formação acadêmica

Engenharia Informática de Gestão (Politécnico de Portugal)

Apresentação

Quem sou e objetivo da aula

Othon L. T de Oliveira

Professor:

- Ensina: Introdução a Programação, Lógica de Programação, e outras disciplinas

Trabalho:

Analista de Infraestrutura: Reitoria da UPE

Formação acadêmica

Engenharia Informática de Gestão (Politécnico de Portugal) Mestrado em Engenharia de Sistemas (Escola Politécnica de Pernambuco - Poli)

Apresentação

Quem sou e objetivo da aula

Othon L. T de Oliveira

Professor:

- Ensina: Introdução a Programação, Lógica de Programação, e outras disciplinas

Trabalho:

Analista de Infraestrutura: Reitoria da UPE

Formação acadêmica

Engenharia Informática de Gestão (Politécnico de Portugal) Mestrado em Engenharia de Sistemas (Escola Politécnica de Pernambuco - Poli)
Doutorando em Engenharia de Produção (Universidade Tecnológica Federal do Paraná - UTFPR)

Conceitos

Introdução a Lógica de Programação com Java Script

Breve introdução

- a. Breve explicação sobre a importância da lógica de programação



Conceitos

Introdução a Lógica de Programação com Java Script

Breve introdução

- a. Breve explicação sobre a importância da lógica de programação
- b. Como ela é usada em todos os tipos de desenvolvimento de software



Conceitos

Introdução a Lógica de Programação com Java Script

Breve introdução

- a. Breve explicação sobre a importância da lógica de programação
- b. Como ela é usada em todos os tipos de desenvolvimento de software
- c. Java Script é basicamente uma linguagem client-side (navegadores)



Conceitos

Introdução a Lógica de Programação com Java Script

Breve introdução

- a. Breve explicação sobre a importância da lógica de programação
- b. Como ela é usada em todos os tipos de desenvolvimento de software
- c. Java Script é basicamente uma linguagem client-side (navegadores)
- d. Contudo Java Script hoje em dia é também server-side (servidores)



JavaScript

Introdução

JavaScript é uma linguagem de programação amplamente utilizada para desenvolvimento web.

- Interpretada, executada no navegador
- Adiciona interatividade e dinamismo às páginas web
- Compatível com HTML e CSS



JavaScript

Recursos

Recursos Principais:

- Manipulação do DOM
- Eventos e interatividade
- Requisições assíncronas (AJAX)
- Frameworks populares: React, Angular, Vue

Vantagens:

- Amplamente suportado nos navegadores
- Versatilidade e flexibilidade
- Comunidade ativa e vasta documentação



DOM - Modelo de Objeto de Documento

O DOM é uma representação em árvore da estrutura do documento HTML.

- Permite manipular elementos HTML com JavaScript
- Facilita adicionar, remover ou alterar conteúdo
- Atualizações dinâmicas em páginas web



Ajax - Asynchronous JavaScript and XML

Ajax é uma técnica que permite atualizações assíncronas em páginas web.

- Atualiza partes da página sem recarregá-la completamente
- Melhora a experiência do usuário e economiza largura de banda
- Usa requisições HTTP assíncronas para buscar ou enviar dados



React

React é uma biblioteca JavaScript para construir interfaces de usuário.

- Desenvolvido pelo Facebook
- Componentização: construção de UIs em pedaços reutilizáveis
- Atualizações eficientes com a virtual DOM
- Uso em aplicações web e mobile (React Native)



Vue.js

Vue.js é um framework progressivo para construir interfaces de usuário.

- Fácil integração com projetos existentes
- Abordagem gradual: adicione Vue a partes específicas do projeto
- Reativo e eficiente, com renderização de componentes
- Ecossistema sólido e documentação abrangente



JavaScript

Lado do Servidor

Hoje em dia, JavaScript também é utilizado no lado do servidor, com o Node.js.

- Node.js permite executar JavaScript no servidor
- Ideal para desenvolvimento de aplicações escaláveis e em tempo real
- Facilita a criação de servidores web e APIs



Operadores Lógicos

Operador Lógico AND (&&)

Atuação

- a. O operador lógico AND (&&) combina duas condições



Operadores Lógicos

Operador Lógico AND (&&)

Atuação

- a. O operador lógico AND (&&) combina duas condições
- b. Retorna verdadeiro (true) apenas se ambas as condições forem verdadeiras



Exemplo de pseudo-código usando o operador lógico AND

```
SE (condição1 for verdadeira) E (condição2 for verdadeira)
ENTAO
    // Executar código se ambas as condições forem verdadeiras
FIM SE
```

Exemplo em Java Script

```
if (condicao1 && condicao2) {
    // Executar código se ambas as condições forem verdadeiras
}
```



Operadores Lógicos

Operador Lógico OR (||)

Atuação

- a. O operador lógico OR (||) combina duas condições



Operadores Lógicos

Operador Lógico OR (||)

Atuação

- a. O operador lógico OR (||) combina duas condições
- b. Retorna verdadeiro (true) se apenas uma das condições for verdadeira



Exemplo de pseudo-código usando o operador lógico OR

```
SE (condição1 for verdadeira) OU (condição2 for verdadeira)
ENTÃO
    // Executar código se pelo menos uma das condições for verdadeira
FIM SE
```

Exemplo em Java Script

```
if (condicao1 || condicao2) {
    // Executar código se pelo menos uma das condições for verdadeira
}
```



Operadores Lógicos

Operador Lógico NOT (!)

Atuação

- a. O operador lógico NOT (!) inverte o valor de uma condição, transformando verdadeiro em falso e falso em verdadeiro.



Exemplo de pseudo-código usando o operador lógico NOT

```
SE NAO (condição for verdadeira)
ENTAO
    // Executar código se a condição for falsa
FIM SE
```

Exemplo em Java Script

```
if (!condicao) {
    // Executar código se a condição for falsa
}
```



Demonstração de um exemplo prático de uso de operadores lógicos em JavaScript.

Código Java Script:

```
const idade = 15;
    const possuiPermissao = true;

if (idade >= 18 || possuiPermissao) {
    console.log("Acesso permitido!");
} else {
    console.log("Acesso negado!");
}
```

Qual o resultado, após executar o código?



Introdução às Estruturas de Decisão

Breve explicação sobre a importância das estruturas de decisão em programação e como elas permitem que o programa tome decisões com base em condições.



Estrutura Básica de um Programa

- Exemplo de Pseudo-Linguagem:

```
SE (condição for verdadeira)
  ENTAO
    // Executar código se a condição for verdadeira
  FIM SE
```

- Exemplo Java Script:

```
if (condicao) {
  // Executar código se a condição for verdadeira
}
```



Estrutura IF-ELSE IF-ELSE

A estrutura if-else if-else permite avaliar várias condições em sequência e executar o bloco de código correspondente à primeira condição verdadeira.

Exemplo em Pseudo-Linguagem:

```
SE (condição1 for verdadeira)
  ENTAO
    // Executar código se a condição1 for verdadeira
SENAO SE (condição2 for verdadeira)
  // Executar código se a condição2 for verdadeira
SENAO
  // Executar código se nenhuma das condições for verdadeira
FIM SE
```



Estrutura IF-ELSE IF-ELSE

A estrutura if-else if-else permite avaliar várias condições em sequência e executar o bloco de código correspondente à primeira condição verdadeira.

Exemplo em JavaScript:

```
if (condicao1) {  
    // Executar código se a condição1 for verdadeira  
} else if (condicao2) {  
    // Executar código se a condição2 for verdadeira  
} else {  
    // Executar código se nenhuma das condições for verdadeira  
}
```



Estrutura SWITCH-CASE

A estrutura `switch-case` é usada para avaliar um valor e executar um bloco de código correspondente ao valor.

Exemplo em Pseudo-Linguagem:

```
ESCOLHA (valor)
```

```
CASO opcao1:
```

```
    // Executar código se o valor for opcao1
```

```
    INTERROMPER
```

```
CASO opcao2:
```

```
    // Executar código se o valor for opcao2
```

```
    INTERROMPER
```

```
PADRAO:
```

```
    // Executar código se o valor não for correspondente a nenhum caso
```

```
FIM ESCOLHA
```



Estrutura SWITCH-CASE

A estrutura switch-case é usada para avaliar um valor e executar um bloco de código correspondente ao valor.

Exemplo em JavaScript:

```
switch (valor) {  
  case opcao1:  
    // Executar código se o valor for opcao1  
    break;  
  case opcao2:  
    // Executar código se o valor for opcao2  
    break;  
  default:  
    // Executar código se o valor não for correspondente a r  
}
```



Exemplo Prático em JavaScript (IF-ELSE)

Exemplo em JavaScript:

```
const idade = 15;

if (idade >= 18) {
  console.log("Você é maior de idade.");
} else {
  console.log("Você é menor de idade.");
}
```



Exemplo Prático em JavaScript (IF-ELSE IF-ELSE)

Código JavaScript:

```
const nota = 85;

if (nota >= 90) {
  console.log("Nota A");
} else if (nota >= 80) {
  console.log("Nota B");
} else if (nota >= 70) {
  console.log("Nota C");
} else {
  console.log("Nota abaixo de C");
}
```

Com base na nota, é exibida a mensagem correspondente à faixa de notas.



Exemplo Prático em JavaScript (SWITCH-CASE)

Código JavaScript:

```
const diaDaSemana = "quarta";
switch (diaDaSemana) {
  case "segunda":
    console.log("Dia de trabalho"); break;
  case "terca":
    console.log("Dia de trabalho"); break;
  case "quarta":
    console.log("Meio da semana"); break;
  case "quinta":
    console.log("Dia de trabalho"); break;
  case "sexta":
    console.log("Dia de trabalho"); break;
  default:
    console.log("Fim de semana");
}
```



Conversão de Temperatura

Vamos pratica?

Faça um programinha que solicita ao usuário uma temperatura em graus Celsius e realiza a conversão para graus Fahrenheit e Kelvin.

Há duas soluções apresentadas. Uma roda diretamente no console do navegador.

A outra você deve fazer criar um arquivo HTML (com o nome que você escolher).

O código HTML está em dois slides, contudo você deve colocar tudo em um só arquivo.

Lembrando novamente, essa é uma solução possível. Você não deve se limitar ao código do professor.

É muito mais importante você modificar o código, fazendo do seu jeito, ou, fazer um totalmente diferente.



Conversão de Temperatura (Pseudo-Linguagem)

Pseudo-Linguagem: (ou pseudo-código)

```
LER temperaturaCelcius
temperaturaFahrenheit <- (temperaturaCelcius * 9/5) + 32
temperaturaKelvin <- temperaturaCelcius + 273.15
ESCREVER "Temperatura em Fahrenheit:", temperaturaFahrenheit
ESCREVER "Temperatura em Kelvin:", temperaturaKelvin
```



Conversão de Temperatura (JavaScript)

Código JavaScript:

```
const temperaturaCelcius = parseFloat(prompt(
    "Digite a temperatura em Celsius:"));
const temperaturaFahrenheit = (temperaturaCelcius * 9/5) + 32;
const temperaturaKelvin = temperaturaCelcius + 273.15;

console.log("Temperatura em Fahrenheit:", temperaturaFahrenheit);
console.log("Temperatura em Kelvin:", temperaturaKelvin);
```

Esse código pode rodar diretamente na console do navegador (F12), por exemplo; no navegador Firefox ou no navegador Chrome.



Conversão de Temperatura (HTML) - primeira parte

Código HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>Conversão de Temperatura</title>
</head>
<body>
  <h1>Conversão de Temperatura</h1>
  <label>Digite a temperatura em Celsius: <input type="number"
    id="temperaturaInput"></label>
  <button onclick="converterTemperatura()">Converter</button>
  <p id="resultado"></p>
```

Conversão de Temperatura (HTML) - segunda parte

Código HTML:

```
<script>
function conversaoTemperatura() {
  const temperaturaCelsius = parseFloat(
    document.getElementById("temperaturaInput").value);

  const temperaturaFahrenheit=(temperaturaCelsius * 9/5) + 32;
  const temperaturaKelvin = temperaturaCelsius + 273.15;
  const resultado = '
    Temperatura em Celsius: ${temperaturaCelsius}°C<br>
    Temperatura em Fahrenheit: ${temperaturaFahrenheit.toFixed(2)}°F
    Temperatura em Kelvin: ${temperaturaKelvin.toFixed(2)}K
  ';
```



Conversão de Temperatura (HTML) - terceira parte

Código JavaScript:

```
    document.getElementById("resultado").innerHTML = resultado;  
}  
</script>  
</body>  
</html>
```

Terceira e última parte do código html. Juntar as 3 partes num só arquivo

