

introdução a lógica JS.

Professor: Jader Jackson

Atenção: Após realizar as atividades anteriores, execute as práticas abaixo e por fim desenvolva um projeto de sua autoria utilizando como inspiração, todas ou boa parte das questões que foram feitas.

Ao final, fazer um video e me mandar pelo whatsapp

- 1. Crie um programa em JavaScript que recebe um valor de índice de qualidade do ar (AQI) como entrada e determina o nível de poluição do ar com base nesse valor. O programa deve utilizar múltiplas condições lógicas para classificar o nível de poluição em uma das seguintes categorias: "Bom", "Moderado", "Prejudicial a grupos sensíveis", "Prejudicial à saúde", "Muito prejudicial à saúde" ou "Perigoso".**

```
// Função que determina o nível de poluição com base no AQI
```

```
function verificarNivelDePoluicao(aqi) {  
  if (aqi >= 0 && aqi <= 50) {  
    return "Bom - Qualidade do ar é boa.";  
  } else  
  if (aqi > 50 && aqi <= 100) {  
    return "Moderado - Qualidade do ar razoável.";  
  }  
  else if (aqi > 100 && aqi <= 150) {  
    return "Prejudicial a grupos sensíveis.";  
  }  
  else if (aqi > 150 && aqi <= 200){  
    return "Prejudicial à saúde.";  
  }  
  else if (aqi > 200 && aqi <= 300){  
    return "Muito prejudicial à saúde.";  
  }  
  else  
  {  
    return "Perigoso - Evite atividades ao ar livre."; }  
}  
// Exemplo de uso const aqi = 120;  
// Altere o valor do AQI conforme necessário  
const nivelDePoluicao = verificarNivelDePoluicao(aqi);  
console.log(`Nível de Poluição: ${nivelDePoluicao}`);
```

- 2. Crie um programa em JavaScript que classifica o estado de uma lâmpada com base nas seguintes condições:**
 - Se a lâmpada estiver desligada e a quantidade de luz no ambiente for abaixo de 100 lux, a lâmpada deve ser ligada.
 - Se a lâmpada estiver desligada e a quantidade de luz no ambiente for igual ou acima de 100 lux, a lâmpada deve permanecer desligada.

- Se a lâmpada estiver ligada, ela deve permanecer ligada, independentemente da quantidade de luz no ambiente.
- Se a lâmpada estiver ligada e a quantidade de luz no ambiente for igual ou acima de 200 lux, a lâmpada deve piscar para alertar sobre a quantidade excessiva de luz.

// Função que classifica o estado da lâmpada com base na luz ambiente e no status atual da lâmpada

```
function classificarEstadoDaLampada(statusLampada, luxAmbiente) {
  if (statusLampada === "desligada" && luxAmbiente < 100) {
    return "Ligue a lâmpada.";
  } else if (statusLampada === "desligada" && luxAmbiente >= 100) {
    return "Lâmpada desligada. A quantidade de luz é adequada.";
  } else if (statusLampada === "ligada" && luxAmbiente < 200) {
    return "Lâmpada ligada.";
  } else { return "Lâmpada piscando para alertar sobre a quantidade excessiva de luz.";
  }
}

// Exemplo de uso
const statusLampada = "desligada"; // Altere o status da lâmpada conforme necessário ("ligada" ou "desligada")
const luxAmbiente = 150; // Altere a quantidade de luz ambiente em lux conforme necessário
const resultado = classificarEstadoDaLampada(statusLampada, luxAmbiente);
console.log(`Estado da Lâmpada: ${resultado}`);
```

3. Crie um programa em JavaScript que avalie a elegibilidade de um candidato para um emprego. As seguintes condições devem ser aplicadas:

- O candidato deve ter pelo menos 18 anos.
- O candidato deve ter pelo menos 2 anos de experiência no campo.
- Se o candidato tiver menos de 18 anos, ele é automaticamente desqualificado.
- Se o candidato tiver menos de 2 anos de experiência, ele é desqualificado.
- Se o candidato atender a ambas as condições acima, ele é considerado elegível.

// Função que avalia a elegibilidade do candidato

```
function avaliarElegibilidade(idade, anosExperiencia) {
  if (idade < 18) {
    return "Desculpe, você é muito jovem para se candidatar.";
  }
  else if (anosExperiencia < 2)
  { return "Desculpe, você não possui experiência suficiente para se candidatar.";
  } else {
    return "Parabéns, você é elegível para se candidatar ao emprego!";
  }
}

// Exemplo de uso
const idadeDoCandidato = 20;
// Altere a idade conforme necessário
const anosDeExperiencia = 3;
```

```
// Altere a experiência em anos conforme necessário
const resultado = avaliarElegibilidade(idadeDoCandidato, anosDeExperiencia);
console.log(resultado);
```

Neste exemplo, criaremos um programa de gerenciamento de tarefas simples que permite adicionar, listar e marcar tarefas como concluídas. Vamos criar um objeto que representa uma lista de tarefas.

```
// Definição do objeto Lista de Tarefas
const listaDeTarefas = {
  tarefas: [],

  // Função para adicionar uma nova tarefa
  adicionarTarefa: function(descricao) {
    this.tarefas.push({
      descricao: descricao,
      concluida: false
    });
  },

  // Função para listar todas as tarefas
  listarTarefas: function() {
    console.log("Lista de Tarefas:");
    this.tarefas.forEach((tarefa, indice) => {
      const estado = tarefa.concluida ? "Concluída" : "Pendente";
      console.log(`${indice + 1}. [${estado}] ${tarefa.descricao}`);
    });
  },

  // Função para marcar uma tarefa como concluída
  marcarTarefaConcluida: function(indice) {
    if (indice >= 0 && indice < this.tarefas.length) {
      this.tarefas[indice].concluida = true;
    }
  }
};

// Exemplo de uso do programa de gerenciamento de tarefas
listaDeTarefas.adicionarTarefa("Estudar JavaScript");
listaDeTarefas.adicionarTarefa("Fazer compras");
listaDeTarefas.adicionarTarefa("Fazer exercícios");

listaDeTarefas.listarTarefas(); // Listar tarefas antes de marcar alguma como concluída

listaDeTarefas.marcarTarefaConcluida(1); // Marcar a segunda tarefa como concluída
```

```
listaDeTarefas.listarTarefas(); // Listar tarefas novamente após a marcação
```