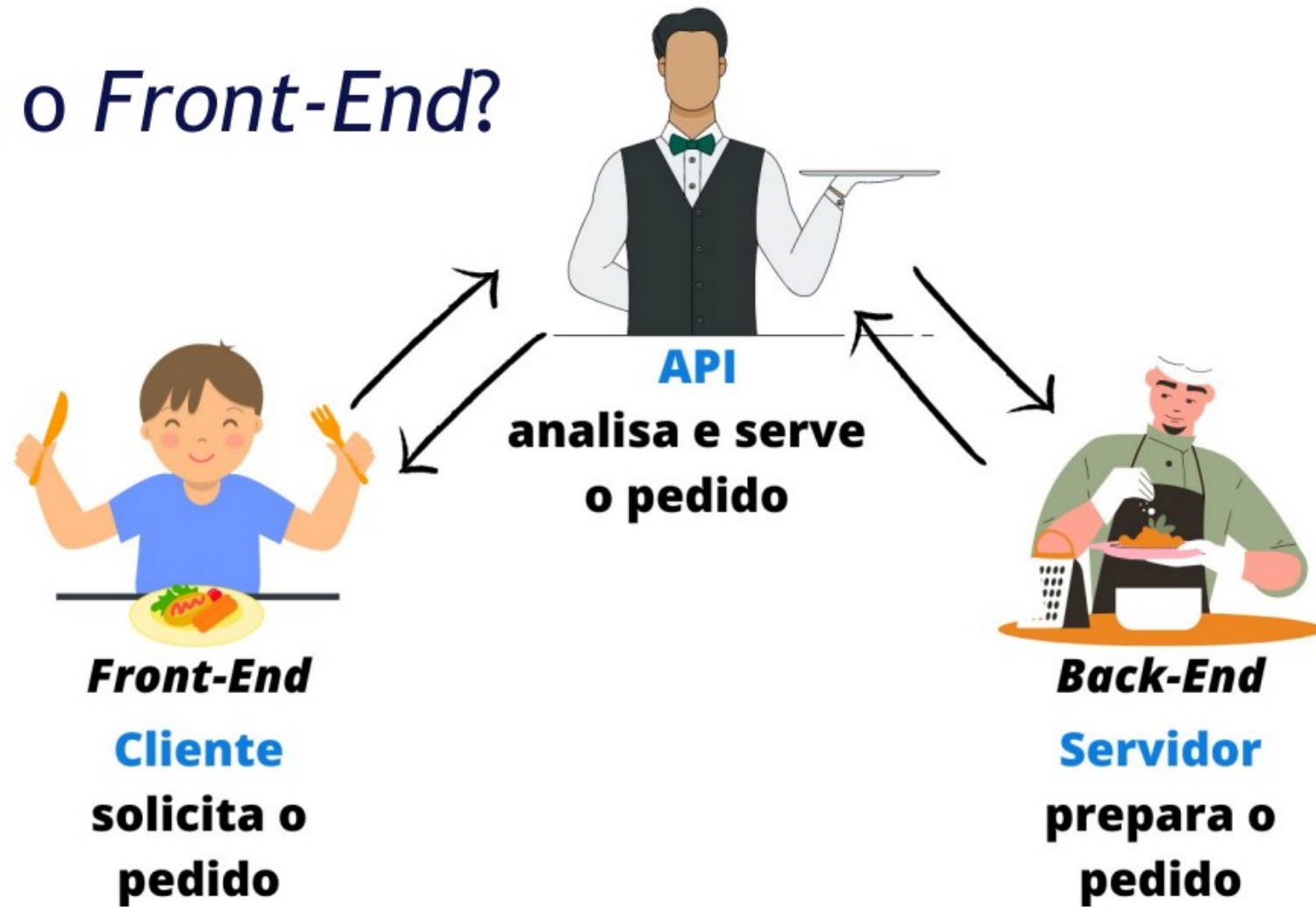
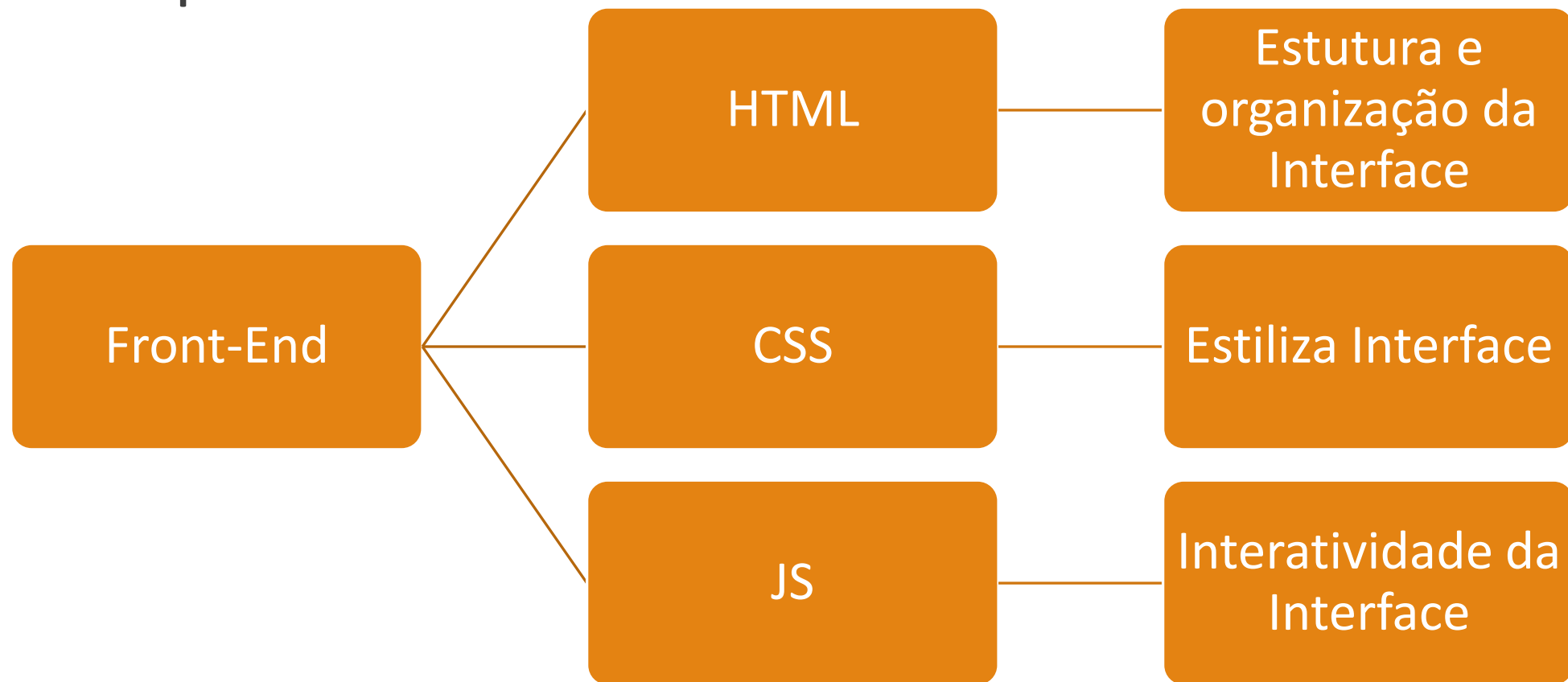


WEB

O que é o *Front-End*?



Principais Elementos



```

<!-- HTML Básico para deonstrar elementos de HTML, CSS e JavaScript -->
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8" />
  <title>HTML Básico</title>
  <style>
    h1 {
      color: ■ red;
    }
    p {
      color: ■ blue;
    }
  </style>
</head>
<body>
  <h1>Aula 1</h1>
  <p>Este é um parágrafo.</p>
  <p id="clique">Este parágrafo mudará ao clicar o botão abaixo.</p>
  <button onclick="muda_paragrafo()">Clique aqui</button>
  <script>
    function muda_paragrafo() {
      let p = document.getElementById("clique");
      p.innerHTML = "Botão clicado!";
    }
  </script>
</body>
</html>

```

Diagram illustrating the structure of the HTML document:

- HTML** (indicated by a large red bracket on the left):
 - CSS** (indicated by a red bracket):
 - `<style>`
 - `h1 {`
 - `color: ■ red;`
 - `}`
 - `p {`
 - `color: ■ blue;`
 - `}`
 - `</style>`
 - JavaScript** (indicated by a red bracket):
 - `<script>`
 - `function muda_paragrafo() {`
 - `let p = document.getElementById("clique");`
 - `p.innerHTML = "Botão clicado!";`
 - `}`
 - `</script>`

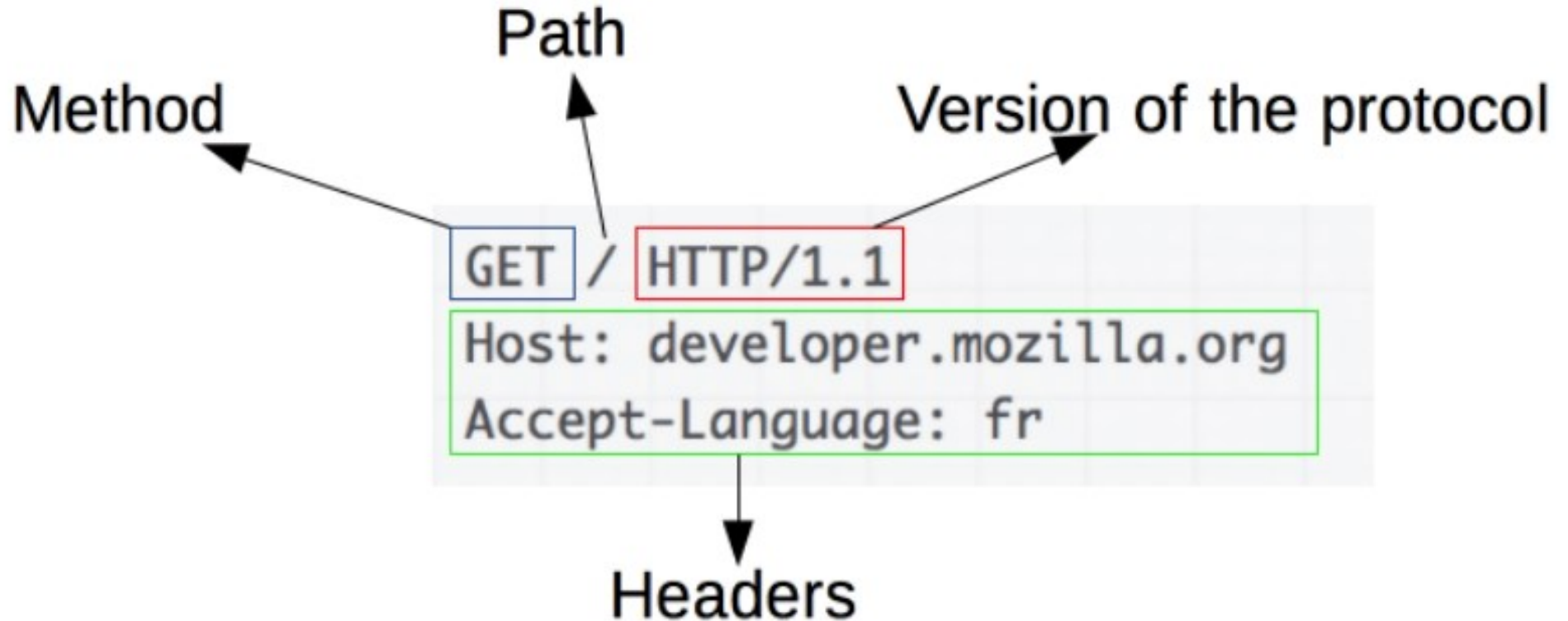


Requisição HTTP

Resposta HTTP



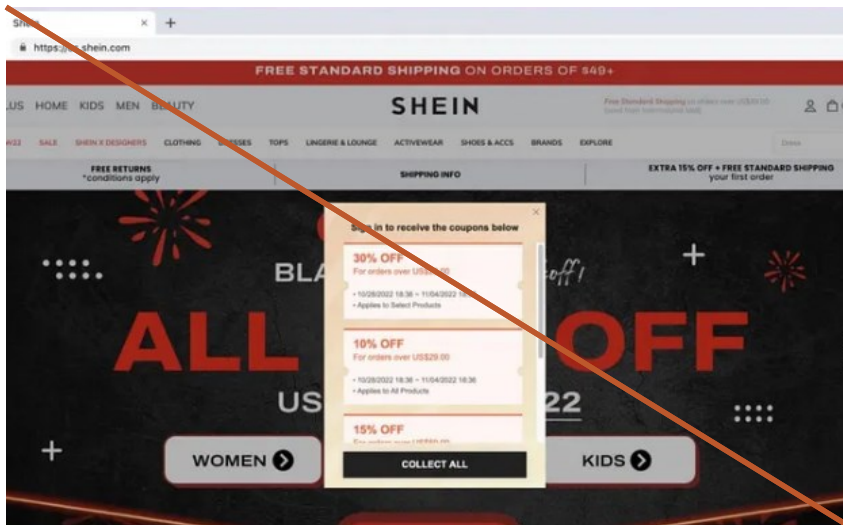
Estrutura das requisições HTTP



[Uma visão geral do HTTP - HTTP | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview)

Código de Resposta

Códigos de Status	Descrição
100-199	Respostas informativas indicando que o servidor recebeu a solicitação e o processo continua.
200-299	Respostas de sucesso indicando que a solicitação foi recebida, compreendida e processada com êxito pelo servidor.
300-399	Respostas de redirecionamento indicando que a solicitação requer ações adicionais para ser concluída.
400-499	Respostas de erro do cliente indicando que ocorreu um erro na solicitação feita pelo cliente.
500-599	Respostas de erro do servidor indicando que ocorreu um erro no servidor ao processar a solicitação.

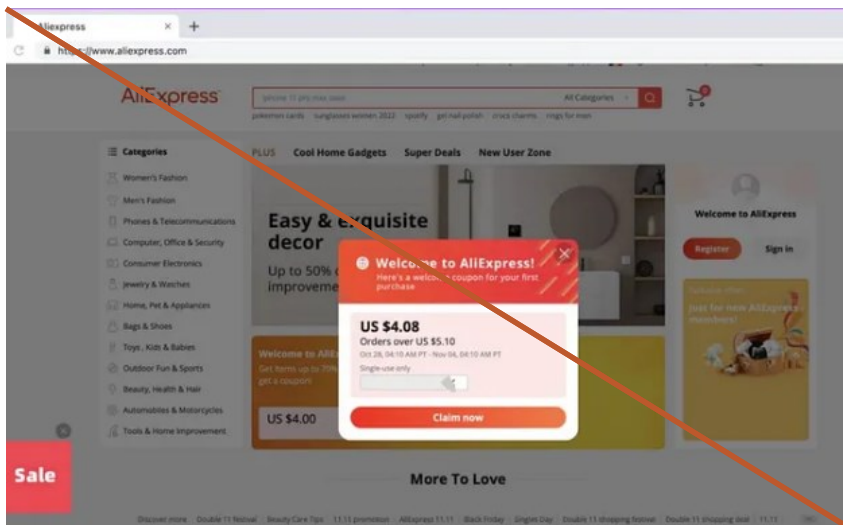


Front-end

Lida com a interface e a experiência do usuário

Camada de interação que o usuário final vê e com a qual interage em um site ou aplicativo

- uma interface agradável
- Intuitiva
- responsiva para os usuários





Responsiva?

Usuários podem acessar e navegar pelo conteúdo de forma fácil e intuitiva, independentemente do dispositivo que estejam usando.

Responsividade: Adaptando a Interface para Diferentes Dispositivos e Tamanhos de Tela

Algumas linguagens básicas - HTML

Linguagem de marcação de hipertexto

Simples usada para criar páginas de internet

- não permitir variáveis ou funções
- não é considerado uma “linguagem de programação”, mas sim uma “linguagem de markup”, uma linguagem que usa tags para definir elementos dentro de um documento.

Sir Tim Berners-Lee criou o HTML no final de 1991, mas a linguagem de codificação não foi disponibilizada para o público em 1993.

Exemplo de linguagens de script - JavaScript, PHP, Python, Ruby, Perl, Lua, Bash, PowerShell, R, VBA



Algumas linguagens básicas - HTML

https://www.w3schools.com/html/html_intro.asp



Questões 1

GV - SES - Analista de Sistemas - Área Administrador de Banco de Dados – 2024

Reconhecendo a importância estratégica deste profissional na gestão e otimização de sistemas de informação, infraestrutura tecnológica e bancos de dados, buscamos elucidar questões fundamentais que permeiam suas responsabilidades e desafios. Ao mergulhar neste conjunto de questionamentos, pretendemos não apenas compreender as competências técnicas exigidas, mas também as habilidades interpessoais necessárias para enfrentar os complexos cenários do mundo digital contemporâneo.

HTML5 e CSS3 desempenham um papel fundamental no desenvolvimento web contemporâneo, fornecendo uma extensa variedade de recursos que permitem a criação de páginas web atrativas visualmente, adaptáveis a diferentes dispositivos e enriquecidas semanticamente.

Com relação as linguagens de marcação, assinale (V) para a afirmativa verdadeira e (F) para falsa.

() O CSS3 suporta elementos multimídia integrados, como áudio e vídeo, sem a necessidade de plugins externos.

() HTML5 e CSS3 são fundamentais para criar websites responsivos, que se adaptam automaticamente a diferentes dispositivos e tamanhos de tela, proporcionando uma melhor experiência de usuário em smartphones, tablets e desktops.

() O HTML5 e o CSS3 são as linguagens de marcação e de estilo mais recentes para desenvolvimento web

() O HTML 5 oferece flexibilidade e controle mais precisos sobre o posicionamento e o design dos elementos em uma página webb

As afirmativas são, respectivamente,

A) F – F – F – V.

B) V – V – V – V.

C) F – V – V – F.

D) V – V – V – F.

Questões 1

Letra B - O CSS3, apesar de ser uma linguagem de estilo, contribui para a responsividade através de media queries, que permitem adaptar o layout de uma página a diferentes tamanhos de tela. O HTML5 e o CSS3 são, de fato, as versões mais recentes e avançadas para desenvolvimento web, trazendo uma série de novas funcionalidades e melhorias em relação às suas versões anteriores. Além disso, o HTML5 oferece maior controle sobre o design e posicionamento dos elementos, facilitando a criação de interfaces mais dinâmicas e interativas.

Questões 2

CESPE/CEBRASPE - Prefeitura de Cachoeiro de Itapemirim - Analista de Sistemas – 2024

Com base em tecnologias e práticas frontend web, julgue os seguintes itens.

O HTML é uma linguagem de programação que recupera as informações de um servidor e as apresenta na tela do usuário.

- C) Certo
- E) Errado

Questões 2

Letra E. HTML é uma linguagem de marcação usada para criar a estrutura de páginas web. Ele define a estrutura e o layout do conteúdo, mas não possui lógica de programação ou capacidade de realizar operações como recuperação de dados de um servidor.

HTML (HyperText Markup Language) é uma linguagem de marcação utilizada para criar e estruturar seções, parágrafos, links e outros elementos de uma página web. Diferente de linguagens de programação como JavaScript, Python ou Java, o HTML não possui lógica de programação e não pode executar operações como loops, condicionais ou recuperação de dados de um servidor. A recuperação de dados de um servidor é geralmente realizada por meio de requisições HTTP, que podem ser feitas utilizando JavaScript, por exemplo, através da API Fetch ou da biblioteca Axios.

Questões 3

VUNESP - SPTRANS - Analista de Informática – 2024Um analista de informática elaborou o seguinte programa na Linguagem HTML:

Um analista de informática elaborou o seguinte programa na Linguagem HTML:

```
<!DOCTYPE html>
<html><body>
<h1>Site da Empresa XYZ</h1>
<p>Veja as nossas ofertas!</p>
</body></html>
```

Caso seja inserida a tag <hr> após a quarta linha, quando o programa for executado por um navegador que suporte HTML, como o Chrome ou o Edge,

- A) o texto da quarta linha será exibido de forma piscante.
- B) o texto da quarta linha será exibido em letras maiores.
- C) será inserida uma linha horizontal antes do texto da quarta linha.
- D) será inserida uma linha horizontal após o texto da quarta linha.
- E) uma mensagem de erro será apresentada, pois o fechamento da tag </hr> não foi colocado.

Questões 3

Letra D. É usada para inserir uma linha horizontal no ponto em que é colocada no documento HTML. Se inserida após a quarta linha, a linha horizontal aparecerá após o texto da quarta linha.

Questões 4

FGV - INPE - Tecnologista Júnior - Área: Especificações de Rede – 2024

Com relação à linguagem de marcação de hipertexto (HTML - HyperText Markup Language), analise os itens a seguir.

I. O elemento `<script>` serve para incluir trechos de códigos para serem executados.

II. O elemento `` serve para incluir comentários no código.

III. O elemento `
` não necessita de fechamento.

Está correto o que se afirma em

A) I, apenas.

B) II, apenas.

C) III, apenas.

D) I e III, apenas.

E) II e III, apenas.

Questões 4

Letra A. A questão aborda o conhecimento sobre a linguagem de marcação de hipertexto (HTML), especificamente sobre a função de alguns elementos HTML. O objetivo é verificar se o candidato conhece a utilidade e a sintaxe correta desses elementos.

Questões 5

Instituto Access - Instituto de Acesso à Educação, Capacitação Profissional e Desenvolvimento Humano - BANESTES - Analista em Tecnologia da Informação - Área Desenvolvimento de Sistemas – 2024

Considerando os princípios fundamentais do desenvolvimento web, temos a importância dos formulários na interação entre usuários e servidores, analise as afirmações a seguir e marque a alternativa correta no uso e a validação de formulários em HTML.

- A) Por ser o único elemento que viabiliza a interação com o servidor, o formulário deve permanecer isolado em uma página web, sem qualquer outra tag de conteúdo.
- B) A validação é um processo que pode ocorrer tanto no lado cliente quanto no lado servidor.
- C) Ao enviar dados para um servidor, é suficiente utilizar os campos presentes no próprio formulário, sem a necessidade de especificar qualquer ação adicional.

Questões 5

Letra B. A validação no lado cliente melhora a experiência do usuário ao fornecer feedback imediato, enquanto a validação no lado servidor é crucial para garantir a segurança e integridade dos dados.

A validação de formulários pode ser realizada no lado cliente usando JavaScript ou atributos HTML5, e no lado servidor utilizando linguagens de programação como PHP, Python, etc., para garantir a segurança e integridade dos dados.

Questões 6

Quadrix - IBICT - Tecnologista - Área: Tecnologia da Informação - 2024

O elemento `` é utilizado para criar itens de lista, tanto em listas ordenadas quanto não ordenadas.

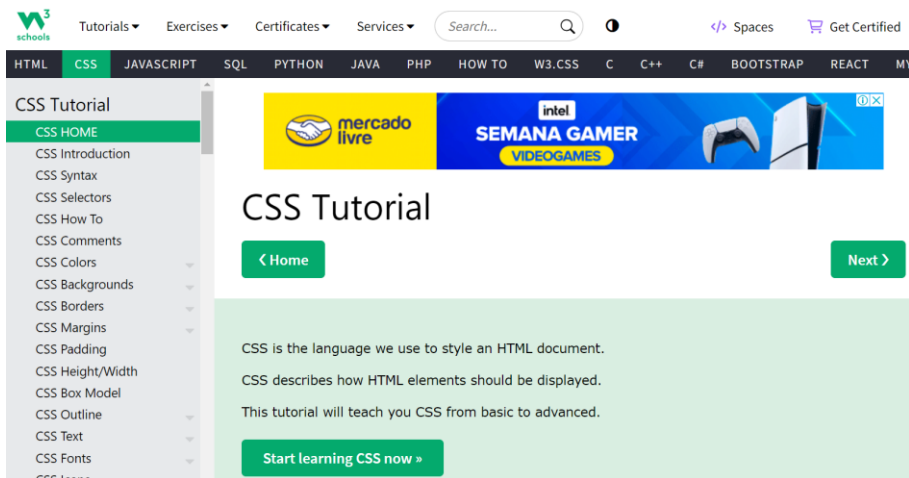
C) Certo

E) Errado

Fonte - [https://developer.mozilla.org/pt-BR/docs/Web/CSS -](https://developer.mozilla.org/pt-BR/docs/Web/CSS-)

Questões 6

O elemento HTML `` (ou a Lista dos Itens de um elemento HTML) é usado para representar um item que faz parte de uma lista.



CSS

Linguagem de estilização

Controle do Layout, cores, fontes – aspectos visuais

Personaliza o Front-End



CSS


CSS (*Cascading Style Sheets* ou Folhas de Estilo em Cascata) é uma linguagem de [estilo](#) usada para descrever a apresentação de um documento escrito em [HTML](#) ou em [XML](#) (incluindo várias linguagens em XML como [SVG](#), [MathML](#) ou [XHTML](#)).

O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias.

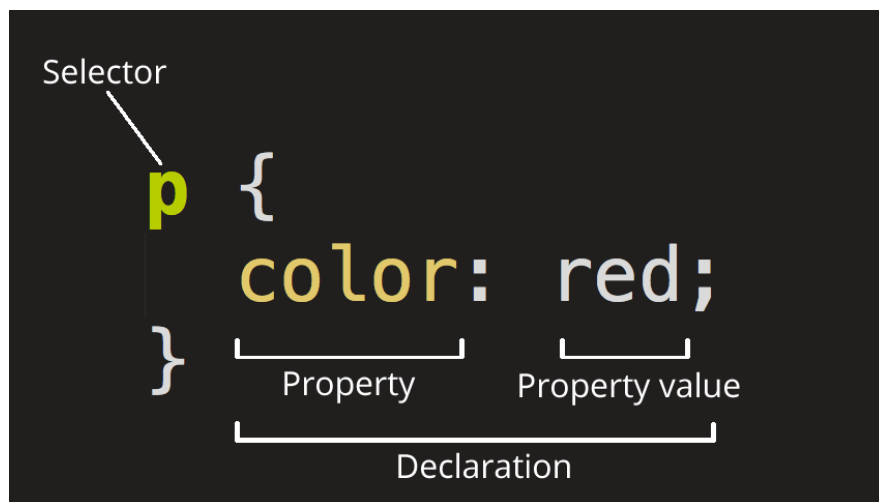
CSS

Assim como o HTML, o CSS não é realmente uma linguagem de programação. Também não é uma linguagem de marcação — é uma *linguagem de folhas de estilos*. Isso significa que o CSS permite aplicar estilos seletivamente a elementos em documentos HTML.

Por exemplo, para selecionar **todos** os elementos parágrafo de uma página HTML e tornar o texto dentro deles vermelho, você escreveria este CSS:

```
CSS   
  
p {  
  color: red;  
}
```

CSS



CSS

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

Fonte - <https://developer.mozilla.org/pt-BR/docs/Web/CSS> -

CSS - Caixas, caixas, é tudo sobre caixas

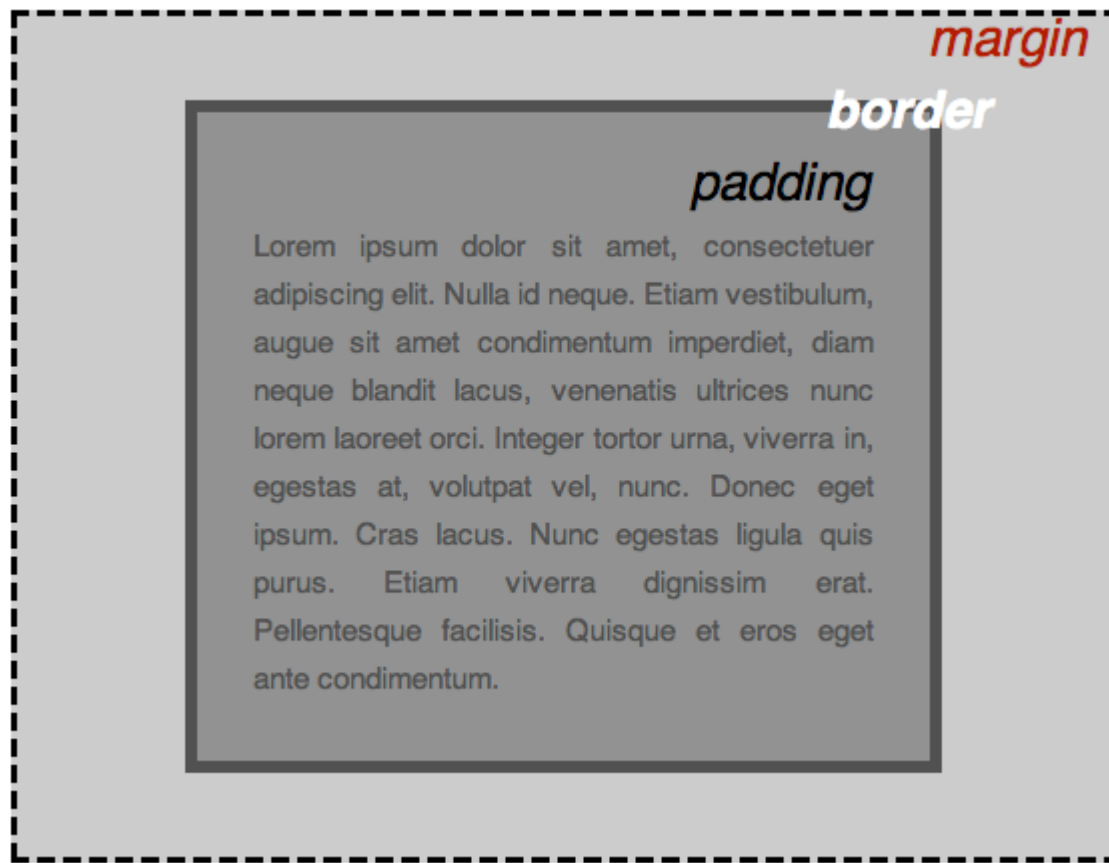
Uma coisa que você notará sobre escrever CSS é que muito disso é sobre caixas — indicar seu tamanho, cor, posição, etc. Muitos dos elementos HTML da sua página podem ser pensados como caixas umas em cima das outras.



Como esperado, o layout CSS é baseado principalmente no *modelo de caixas*. Cada um dos blocks que ocupam espaço na sua página tem propriedades como essas:

- padding, o espaço ao redor do conteúdo (ex.: ao redor do texto de um parágrafo).
- border, a linha sólida do lado de fora do padding.
- margin, o espaço externo a um elemento.

CSS - Caixas, caixas, é tudo sobre caixas



Temos:

- width (largura de um elemento).
- background-color, a cor atrás do conteúdo de um elemento e do padding.
- color, a cor do conteúdo de um elemento (geralmente texto).
- text-shadow: cria uma sombra no texto dentro de um elemento.
- display: define a maneira de dispor um elemento (não se preocupe com isso ainda).

Fonte - [https://developer.mozilla.org/pt-BR/docs/Web/CSS -](https://developer.mozilla.org/pt-BR/docs/Web/CSS-)

CSS

Explore - <https://www.w3schools.com/css/>

Fonte - <https://developer.mozilla.org/pt-BR/docs/Web/CSS> -

JavaScript

Linguagem de programação mais popular do mundo.

Linguagem de programação da Web.

Revisão JavaScript - <https://www.w3schools.com/jsref/default.asp>

Explore - <https://www.w3schools.com/css/>

JavaScript

Desenvolver - pedir dois números para o usuário e apresentar o resultado da soma destes números na tela.

JavaScript

Desenvolver - pedir dois números para o usuário e apresentar o resultado da soma destes números na tela.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Calcula</title>
</head>
<script>
function Soma()
{
  var num1 = parseInt(document.getElementById("n1").value);
  var num2 = parseInt(document.getElementById("n2").value);
  var soma = num1 + num2;
  document.getElementById("res").innerHTML = "Resultado: " + soma;
}
</script>
<body>
  <input type="text" id="n1"> <br>
  <input type="text" id="n2"> <br><p id="res">
  <p> Resultado: </P>
  <input type="button" value="Soma" onclick="Soma()">
</body>
</html>
```

JavaScript

Um Ambiente de Desenvolvimento Integrado (IDE) é uma ferramenta que pode ser usada para melhorar a produtividade e eficiência no desenvolvimento de software. Ela oferece uma série de recursos que podem ajudar os programadores a realizar suas tarefas de forma mais rápida e precisa:

- Automação
- Realce de sintaxe
- Complementação de código
- Refatoração
- Gerenciamento de projetos
- Controle de versões
- Integração Git, facilitando a colaboração entre equipes.
- Plugins e extensões

Fonte - <https://developer.mozilla.org/pt-BR/docs/Web/CSS> -

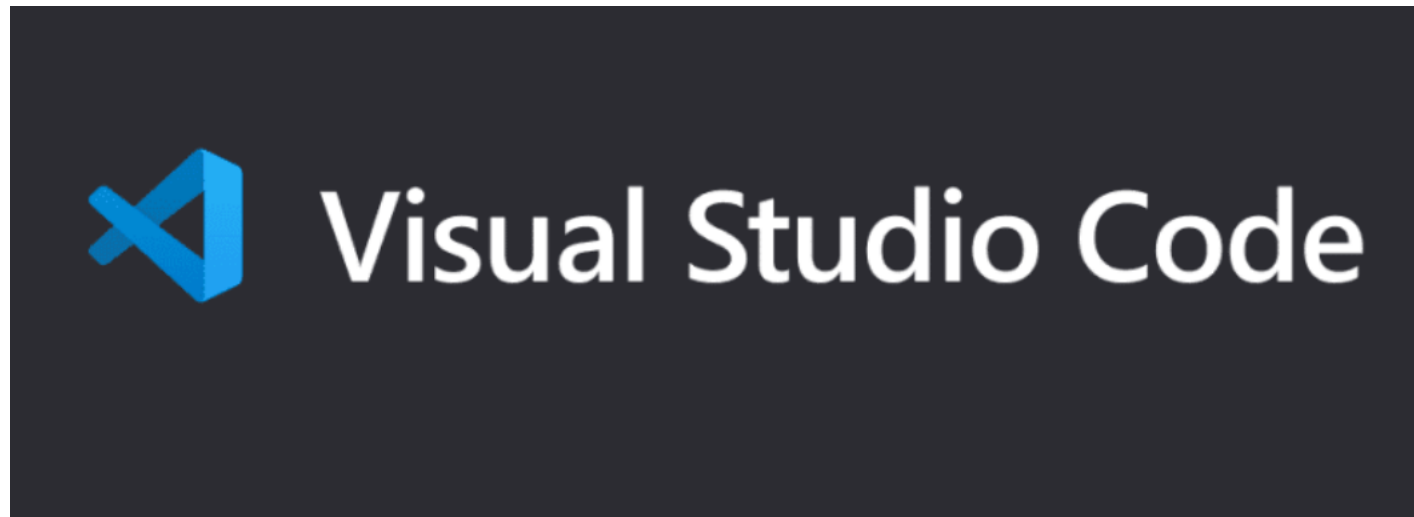
JavaScript



<https://eclipseide.org/> -

Fonte - <https://developer.mozilla.org/pt-BR/docs/Web/CSS->

JavaScript



<https://code.visualstudio.com/>

Fonte - <https://developer.mozilla.org/pt-BR/docs/Web/CSS->

JavaScript – Laços de Repetição

- if/else e operador ternário
- switch case
- while e do...while
- for, for...in e for...of

JavaScript – Laços de Repetição

● if/else e operador ternário

Ler nota do aluno

Se nota é número

então

Se nota maior ou igual a 60

então Aprovado

senão Reprovado

JavaScript – Laços de Repetição

- if/else e operador ternário

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Verificação de Nota</title>
</script>
  function verificarNota() {
    // Obtém o valor do campo de entrada
    let notaAluno = document.getElementById("nota").value;
    notaAluno = parseFloat(notaAluno); // Converte a entrada para número

    let resultado; // Variável para armazenar o resultado

    if (!isNaN(notaAluno)) { // Verifica se a entrada é um número
      if (notaAluno >= 60) {
        resultado = "Aprovado!";
      } else {
        resultado = "Reprovado!";
      }
    } else {
      resultado = "Por favor, insira um número válido para a nota.";
    }

    // Exibe o resultado na página
    document.getElementById("resultado").innerText = resultado;
  }
</script>
</head>
<body>
  <h1>Verificação de Nota</h1>
  <label for="nota">Digite sua nota:</label>
  <input type="text" id="nota" placeholder="Ex: 85">
  <button onclick="verificarNota()">Verificar</button>

  <h2 id="resultado"></h2> <!-- Onde o resultado será exibido -->
</body>
</html>
```

Ler nota do aluno

Se nota é número

então

Se nota maior ou igual a 60

então Aprovado

senão Reprovado

JavaScript – Laços de Repetição

```
Faça sempre
Ler valor (1 a 7 ou 0 para sair)
Se valor o solicitado
então
    Se valor igual a 0
    então saia
    Caso
    valor seja 1
        exibir Domingo
    valor seja 2
        exibir segunda-feira
    ...
    Fim Caso
senão
    exibir Por favor, insira um número válido de 1 a 7
Fim Faça
```

Utilize o comando CASE e While

JavaScript – Laços de Repetição

- while e do...while / Case

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de While Loop</title>
</script>
function verificarDia() {
  let diaNumero;
  let resultado = "";

  // Loop enquanto o diaNúmero não for 0
  while (true) {
    diaNumero = prompt("Digite um número de 1 a 7 para o dia da semana (ou 0 para sair):");

    // Converte a entrada para número inteiro
    diaNumero = parseInt(diaNumero);

    // Condição para sair do loop
    if (diaNumero === 0) {
      resultado = "Saindo...";
      break;
    }

    // Verifica se a entrada é um número válido
    if (!isNaN(diaNumero) && diaNumero >= 1 && diaNumero <= 7) {
      // Usando switch case para determinar o dia da semana
      switch (diaNumero) {
        case 1:
          resultado = "Domingo";
          break;
        case 2:
          resultado = "Segunda-feira";
```

```
          break;
        case 3:
          resultado = "Terça-feira";
          break;
        case 4:
          resultado = "Quarta-feira";
          break;
        case 5:
          resultado = "Quinta-feira";
          break;
        case 6:
          resultado = "Sexta-feira";
          break;
        case 7:
          resultado = "Sábado";
          break;
      }
    } else {
      resultado = "Por favor, insira um número válido de 1 a 7.";
    }

    // Exibe o resultado
    alert(resultado);
  }

  // Exibe a mensagem de saída
  alert(resultado);
}
</script>
</head>
<body>
  <h1>Verificação do Dia da Semana</h1>
  <button onclick="verificarDia()">Iniciar</button>
</body>
</html>
```

JavaScript – Laços de Repetição

```
Faça sempre
Ler valor (1 a 7 ou 0 para sair)
Se valor o solicitado
então
    Se valor igual a 0
    então saia
    Caso
    valor seja 1
        exibir Domingo
    valor seja 2
        exibir segunda-feira
    ...
    Fim Caso
senão
    exibir Por favor, insira um número válido de 1 a 7
Fim Faça
```

Utilize o comando CASE, While e FOR

JavaScript – Laços de Repetição

- while e do...while / Case / For

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de For Loop</title>
<script>
  function verificarDia() {
    let resultado;
    const tentativas = 3; // Número de tentativas
    let diaNumero;

    // Loop for para o número de tentativas
    for (let i = 0; i < tentativas; i++) {
      diaNumero = parseInt(prompt("Digite um número de 1 a 7 para o dia da semana:"));

      // Verifica se a entrada é um número válido
      if (!isNaN(diaNumero) && diaNumero >= 1 && diaNumero <= 7) {
        // Usando switch case para determinar o dia da semana
        switch (diaNumero) {
          case 1:
            resultado = "Domingo";
            break;
          case 2:
            resultado = "Segunda-feira";
            break;
          case 3:
            resultado = "Terça-feira";
            break;
          case 4:
```

```
            resultado = "Quarta-feira";
            break;
          case 5:
            resultado = "Quinta-feira";
            break;
          case 6:
            resultado = "Sexta-feira";
            break;
          case 7:
            resultado = "Sábado";
            break;
        }
        // Exibe o resultado e sai do loop
        alert("Você digitou: " + resultado);
        break; // Sai do for após uma entrada válida
      } else {
        alert("Por favor, insira um número válido de 1 a 7.");
      }

      // Se for a última tentativa
      if (i === tentativas - 1) {
        alert("Você excedeu o número de tentativas.");
      }
    }
  }
</script>
</head>
<body>
  <h1>Verificação do Dia da Semana</h1>
  <button onclick="verificarDia()">Iniciar</button>
</body>
</html>
```

JavaScript – Laços de Repetição

- **for, for...in e for...of**

Estruturas de repetição For, For ... in e For ... of Javascript

<https://youtu.be/VGOhchtuQAc>

JavaScript – Laços de Repetição

- Laços de Repetição – CALCULADORA SOMA

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Laços de Repetição</title>
</script>
function somaComFor() {
  let soma = 0;
  const tentativas = 5; // Número de tentativas

  for (let i = 0; i < tentativas; i++) {
    let numero = parseFloat(prompt("Digite um número para somar (tentativa " + (i + 1) + "):"));
    if (!isNaN(numero)) {
      soma += numero;
    } else {
      alert("Por favor, insira um número válido.");
      i--; // Mantém a contagem de tentativas
    }
  }
  alert("A soma total usando for é: " + soma);
}

function somaComWhile(){
  let soma = 0;
  let continuar = true;

  while (continuar) {
    let numero = parseFloat(prompt("Digite um número para somar (ou digite '0' para sair:)"));
    if (numero === 0) {
      continuar = false;
    } else if (!isNaN(numero)) {
      soma += numero;
    } else {
      alert("Por favor, insira um número válido.");
    }
  }
  alert("A soma total usando while é: " + soma);
}
```

```
function somaComDoWhile() {
  let soma = 0;
  let continuar;

  do {
    let numero = parseFloat(prompt("Digite um número para somar (ou digite '0' para sair:)"));
    if (numero === 0) {
      continuar = false;
    } else if (!isNaN(numero)) {
      soma += numero;
      continuar = true; // Continua a soma
    } else {
      alert("Por favor, insira um número válido.");
      continuar = true; // Permite o acidente de um número inválido
    }
  } while (continuar);

  alert("A soma total usando do...while é: " + soma);
}
</script>
</head>
<body>
  <h1>Laços de Repetição em JavaScript</h1>
  <button onclick="somaComFor()">Soma com for</button>
  <button onclick="somaComWhile()">Soma com while</button>
  <button onclick="somaComDoWhile()">Soma com do...while</button>
</body>
</html>
```

JavaScript – Laços de Repetição

- Laços de Repetição Calculadora com CSS

```
<!DOCTYPE html>
<html lang="pt-BR">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Laços de Repetição</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    h1 {
      color: #333;
    }

    button {
      background-color: #4CAF50; /* Verde */
      color: white;
      border: none;
      padding: 15px 20px;
      margin: 10px;
      cursor: pointer;
      font-size: 16px;
      border-radius: 5px;
      transition: background-color 0.3s;
    }

    button:hover {
      background-color: #45a049; /* Verde escuro */
    }

    button:active {
```

```
      background-color: #3e8e41; /* Tom mais escuro */
    }

    .alert {
      border: 2px solid #4CAF50;
      padding: 10px;
      margin-top: 15px;
      border-radius: 5px;
      background-color: #e7f4e4;
      color: #333;
      width: 300px;
      text-align: center;
    }
  </style>
  <script>
    function somaComFor() {
      let soma = 0;
      const tentativas = 5; // Número de tentativas

      for (let i = 0; i < tentativas; i++) {
        let numero = parseFloat(prompt("Digite um número para somar (tentativa " + (i + 1) +
          "):"));
        if (!isNaN(numero)) {
          soma += numero;
        } else {
          alert("Por favor, insira um número válido.");
          i--; // Mantém a contagem de tentativas
        }
      }
      alert("A soma total usando for é: " + soma);
    }

    function somaComWhile() {
      let soma = 0;
      let continuar = true;

      while (continuar) {
        let numero = parseFloat(prompt("Digite um número para somar (ou digite '0' para
sair):"));
        if (numero === 0) {
          continuar = false;
        } else if (!isNaN(numero)) {
          soma += numero;
        }
      }
    }

    function somaComDoWhile() {
      let soma = 0;
      let continuar = true;

      do {
        let numero = parseFloat(prompt("Digite um número para somar (ou digite '0' para
sair):"));
        if (numero === 0) {
          continuar = false;
        } else if (!isNaN(numero)) {
          soma += numero;
          continuar = true; // Continua a soma
        } else {
          alert("Por favor, insira um número válido.");
          continuar = true; // Permite o acidente de um número inválido
        }
      } while (continuar);

      alert("A soma total usando do...while é: " + soma);
    }
  </script>
</head>

<body>
  <h1>Laços de Repetição em JavaScript</h1>
  <button onclick="somaComFor()">Soma com for</button>
  <button onclick="somaComWhile()">Soma com while</button>
  <button onclick="somaComDoWhile()">Soma com do...while</button>
</body>

</html>
```

```
    } else {
      alert("Por favor, insira um número válido.");
    }
  }
  alert("A soma total usando while é: " + soma);
}

function somaComDoWhile() {
  let soma = 0;
  let continuar;

  do {
    let numero = parseFloat(prompt("Digite um número para somar (ou digite '0' para
sair):"));
    if (numero === 0) {
      continuar = false;
    } else if (!isNaN(numero)) {
      soma += numero;
      continuar = true; // Continua a soma
    } else {
      alert("Por favor, insira um número válido.");
      continuar = true; // Permite o acidente de um número inválido
    }
  } while (continuar);

  alert("A soma total usando do...while é: " + soma);
}
</script>
</head>

<body>
  <h1>Laços de Repetição em JavaScript</h1>
  <button onclick="somaComFor()">Soma com for</button>
  <button onclick="somaComWhile()">Soma com while</button>
  <button onclick="somaComDoWhile()">Soma com do...while</button>
</body>

</html>
```

JavaScript

- Exercício 1

Criar um *script* para solicitar o ano atual e o ano de nascimento – calcule a diferença.

JavaScript

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Cálculo de Idade</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    h1 {
      color: #333;
    }

    button, input {
      padding: 10px;
      margin: 5px;
      font-size: 16px;
    }
```

```
    button {
      background-color: #4CAF50; /* Verde */
      color: white;
      border: none;
      cursor: pointer;
      border-radius: 5px;
      transition: background-color 0.3s;
    }

    button:hover {
      background-color: #45a049; /* Verde escuro */
    }

    button:active {
      background-color: #3e8e41; /* Tom mais escuro */
    }

    .resultado {
      margin-top: 20px;
      font-size: 20px;
    }
  </style>
<script>
  function calcularIdade() {
    // Solicita o ano atual
    let anoAtual = parseInt(prompt("Digite o ano atual:"));
    if (isNaN(anoAtual)) {
      alert("Por favor, insira um ano válido.");
      return;
    }
  }
```

```

  // Solicita o ano de nascimento
  let anoNascimento = parseInt(prompt("Digite seu ano de
nascimento:"));
  if (isNaN(anoNascimento)) {
    alert("Por favor, insira um ano de nascimento válido.");
    return;
  }

  // Calcula a idade
  let idade = anoAtual - anoNascimento;

  // Valida a idade
  if (idade < 0) {
    alert("Você ainda não nasceu!");
  } else {
    alert("Sua idade é: " + idade + " anos.");
  }
</script>
</head>
<body>
  <h1>Cálculo de Idade</h1>
  <button onclick="calcularIdade()">Calcular Idade</button>
</body>
</html>
```


JavaScript

- Exercício 2

Criar um *script* que peça dois valores ao usuário e a operação que ele deseja executar (+, -, *, /). Execute a operação desejada e mostre o resultado na tela.

JavaScript

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Calculadora Simples</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    h1 {
      color: #333;
    }

    input {
      padding: 10px;
      margin: 5px;
      font-size: 16px;
      width: 100px;
    }

    button {
      background-color: #4CAF50; /* Verde */
      color: white;
      border: none;
```

```
padding: 10px 15px;
cursor: pointer;
font-size: 16px;
border-radius: 5px;
transition: background-color 0.3s;
}

button:hover {
  background-color: #45a049; /* Verde escuro */
}

button:active {
  background-color: #3e8e41; /* Tom mais escuro */
}

.resultado {
  margin-top: 20px;
  font-size: 20px;
}
</style>
<script>
  function calcular() {
    // Solicita os dois valores
    let valor1 = parseFloat(prompt("Digite o primeiro valor:"));
    let valor2 = parseFloat(prompt("Digite o segundo valor:"));
    let operacao = prompt("Digite a operação desejada (+, -, *, /).");

    let resultado;

    // Verifica a operação e executa a correspondente
    switch (operacao) {
      case '+':
        resultado = valor1 + valor2;
        break;
```

```
      case '-':
        resultado = valor1 - valor2;
        break;
      case '*':
        resultado = valor1 * valor2;
        break;
      case '/':
        if (valor2 !== 0) {
          resultado = valor1 / valor2;
        } else {
          alert("Erro: Divisão por zero não é permitida.");
          return;
        }
        break;
      default:
        alert("Operação inválida. Por favor, insira uma operação
válida (+, -, *, /).");
        return;
    }
  }

  // Exibe o resultado
  alert("O resultado é: " + resultado);
}
</script>
</head>
<body>
  <h1>Calculadora Simples</h1>
  <button onclick="calcular()">Calcular</button>
</body>
</html>
```

JavaScript

● Exercício 3

Criar um *script* que peça dois valores ao usuário e a operação que ele deseja executar (+, -, *, /). Execute a operação desejada e mostre o resultado na tela.

→ utilize as estrutura IF

JavaScript

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Calculadora Simples</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    h1 {
      color: #333;
    }

    button {
      background-color: #4CAF50; /* Verde */
      color: white;
      border: none;
      padding: 10px 15px;
      cursor: pointer;
      font-size: 16px;
      border-radius: 5px;
      transition: background-color 0.3s;
```

```

    }

    button:hover {
      background-color: #45a049; /* Verde escuro */
    }

    button:active {
      background-color: #3e8e41; /* Tom mais escuro */
    }

    .resultado {
      margin-top: 20px;
      font-size: 20px;
    }
  </style>
  <script>
    function calcular() {
      // Solicita os dois valores
      let valor1 = parseFloat(prompt("Digite o primeiro valor:"));
      let valor2 = parseFloat(prompt("Digite o segundo valor:"));
      let operacao = prompt("Digite a operação desejada (+, -, *, /):");

      let resultado;

      // Verifica a operação e executa a correspondente
      if (operacao === '+') {
        resultado = valor1 + valor2;
      } else if (operacao === '-') {
        resultado = valor1 - valor2;
      } else if (operacao === '*') {
        resultado = valor1 * valor2;
      } else if (operacao === '/') {
```

```

      if (valor2 !== 0) {
        resultado = valor1 / valor2;
      } else {
        alert("Erro: Divisão por zero não é permitida.");
        return;
      }
    } else {
      alert("Operação inválida. Por favor, insira uma operação
válida (+, -, *, /).");
      return;
    }

    // Exibe o resultado
    alert("O resultado é: " + resultado);
  }
</script>
</head>
<body>
  <h1>Calculadora Simples</h1>
  <button onclick="calcular()">Calcular</button>
</body>
</html>
```

JavaScript

- Exercício 4

(`while`, `do...while` e `for`): crie um script que seja capaz de gerar a tabuada de qualquer número inteiro digitado pelo usuário.

JavaScript

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Tabuada de um Número</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    h1 {
      color: #333;
    }

    button {
      background-color: #4CAF50; /* Verde */
      color: white;
      border: none;
      padding: 10px 15px;
      cursor: pointer;
      font-size: 16px;
      border-radius: 5px;
      transition: background-color 0.3s;
```

```
      margin: 5px;
    }

    button:hover {
      background-color: #45a049; /* Verde escuro */
    }

    button:active {
      background-color: #3e8e41; /* Tom mais escuro */
    }

    .resultado {
      margin-top: 20px;
      font-size: 20px;
      text-align: left;
      width: 200px; /* Largura fixa para o resultado */
    }
  </style>
  <script>
    function gerarTabuada() {
      let numero = parseInt(prompt("Digite um número inteiro
para gerar a tabuada:"));

      if (isNaN(numero)) {
        alert("Por favor, insira um número válido.");
        return;
      }

      let resultado = "";

      // Usando while
      let i = 1;
```

```
      while (i <= 10) {
        resultado += `${numero} x ${i} = ${numero * i}\n`;
        i++;
      }

      resultado += "\n"; // Adiciona um espaço para separação

      // Usando do...while
      i = 1;
      do {
        resultado += `${numero} x ${i} = ${numero * i}\n`;
        i++;
      } while (i <= 10);

      resultado += "\n"; // Adiciona um espaço para separação

      // Usando for
      for (i = 1; i <= 10; i++) {
        resultado += `${numero} x ${i} = ${numero * i}\n`;
      }

      // Exibe o resultado
      alert(`Tabuada do ${numero}:\n\n${resultado}`);
    }
  </script>
</head>
<body>
  <h1>Gerador de Tabuada</h1>
  <button onclick="gerarTabuada()">Gerar Tabuada</button>
</body>
</html>
```

JavaScript

- Exercício 5

Crie um *script* que exiba na tela os 10 primeiros múltiplos de 3. Para isso, faremos uso do operador **MOD (%)**, este operador devolve como resposta o resto de uma divisão inteira. Por exemplo, ao dividir 5/2 o resultado é 2.5.

JavaScript

```
<!DOCTYPE html>
<html lang="pt-BR">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Múltiplos de 3</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #e0f7fa; /* Cor de fundo suave */
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    h1 {
      color: #00796b; /* Cor do título */
      margin-bottom: 20px;
    }

    .resultado {
      font-size: 20px;
      color: #004d40; /* Cor do resultado */
      text-align: center;
      border: 1px solid #00796b;
      border-radius: 5px;
      padding: 10px;
    }
```

```
    */
    background-color: white; /* Fundo branco para o resultado */
  }

  width: 300px;
}

button {
  background-color: #00796b; /* Verde */
  color: white;
  border: none;
  padding: 10px 15px;
  cursor: pointer;
  font-size: 16px;
  border-radius: 5px;
  transition: background-color 0.3s;
  margin-top: 20px;
}

button:hover {
  background-color: #004d40; /* Verde escuro */
}

button:active {
  background-color: #00332d; /* Tom mais escuro */
}
</style>
</head>

<body>
  <h1>10 Primeiros Múltiplos de 3</h1>
  <button onclick="exibirMultiplosDeTres()">Mostrar
Múltiplos</button>
```

```
<div class="resultado" id="resultado"></div>

<script>
  function exibirMultiplosDeTres() {
    let resultado = "";
    let contador = 0; // Contador para os múltiplos encontrados

    for (let i = 1; contador < 10; i++) {
      if (i % 3 === 0) { // Verifica se é múltiplo de 3
        resultado += `${i} `;
        contador++;
      }
    }

    // Exibe os resultados
    document.getElementById("resultado").innerText = "Os 10
primeiros múltiplos de 3 são: " + resultado;
  }
</script>
</body>

</html>
```


Prova Prática

Montagem de um BI para acompanhar o campeonato Brasileiro e Brasileirão.

6 - pontos