

**ATIVIDADE 3 - PROGRAMAÇÃO FRONT END - 52\_2025****Período:**02/06/2025 08:00 a 06/07/2025 23:59 (Horário de Brasília)**Status:**ABERTO**Nota máxima:**0,50**Gabarito:**Gabarito será liberado no dia 07/07/2025 00:00 (Horário de Brasília)**Nota obtida:****1ª QUESTÃO**

Desde o seu surgimento, a linguagem Javascript sempre foi considerada, por alguns, uma linguagem limitada e sem muitas possibilidades além de pequenos efeitos e interações em páginas de web. Com o tempo, o Javascript passou por diversas atualizações que introduziram recursos poderosos e mais eficientes. No ES6, ou seja, no ECMAScript 2015, e nas versões seguintes, surgiram diversos recursos modernos que transformaram a forma de programar. Esses elementos contribuíram para modernizar o código, tornando-o mais legível e fácil de manter. Com a utilização desses elementos modernos, desenvolveremos um pedaço de um jogo para incorporar alguns desses recursos, iniciando pela desestruturação de *arrays* e objetos. Uma aplicação relativamente simples, mas que ajuda a entender o funcionamento do desenvolvimento *front-end*, seria um jogo de perguntas e respostas, o famoso jogo de trívia.

Fonte: NOEL, A. A. **Programação Front End**. Florianópolis: Arquê, 2025.

Com base no texto apresentado, quanto aos recursos introduzidos no ECMAScript 2015 (ES6) contribuíram para tornar o código JavaScript mais legível e fácil de manter, assinale a alternativa correta:

**ALTERNATIVAS**

- ☐ PHP, MySQL, Apache e Node.js.
- ☐ *Frameworks* como React, Angular, Vue.js e JQuery.
- ☐ Compilação automática, transpilação binária e otimização algorítmica.
- ☐ Programação orientada a objetos, *Machine Learning* e gerenciamento de banco de dados.
- ☒ *Arrow functions*, desestruturação de objetos e *arrays*, operador de espalhamento e módulos.

**2ª QUESTÃO**

A linguagem JavaScript tem papel central na manipulação dinâmica de páginas web. Sua sintaxe remete à linguagem C, com blocos delimitados por chaves, mas difere ao apresentar tipagem fraca e dinâmica.

Fonte: Elaborado pelo professor, 2025.

Considere o trecho a seguir inserido dentro de uma página HTML:

```
<script>
document.body.style.backgroundColor = "blue";
</script>
```

Com base nas informações do código e nos conceitos apresentados, analise as afirmativas:

- I. O código JavaScript altera dinamicamente o estilo da página, modificando a cor de fundo do elemento `<body>` diretamente via DOM.
- II. Por ser fracamente tipada, a linguagem JavaScript permite que uma variável que inicialmente recebe um número seja posteriormente atribuída a uma string, sem erro de compilação.
- III. A estrutura típica de um projeto web utiliza arquivos separados para HTML, CSS e JavaScript com o objetivo de manter a organização, modularidade e reutilização de código.

É correto o que se afirma em:

#### ALTERNATIVAS

- ☐ I, apenas.
- ☐ II, apenas.
- ☐ I e III, apenas.
- ☐ II e III, apenas.
- ☒ I, II e III.

#### 3ª QUESTÃO

A união entre HTML e CSS tem ficado mais forte com o passar do tempo e à medida que são implementadas novas versões, é certo que essa união tende a ser duradoura, a menos que surja alguma revolução no desenvolvimento da Internet e que essas linguagens não sejam capazes de acompanhar. É importante perceber que, mesmo tendo uma grande quantidade de comandos disponíveis nas duas linguagens, algumas regras são essenciais e devem ser respeitadas. Isso serve para melhorar a qualidade e a compatibilidade entre um código criado e uma das opções de navegadores disponíveis. Regras como fechamento de todas as tags abertas, padronização do código em letras minúsculas e inclusão de todos os elementos essenciais ao código são características que melhoram a compatibilidade, além de serem alguns requisitos da versão HTML 5.

Fonte: Elaborado pelo professor, 2025.

Fonte: TOKUMOTO, Ronie Cesar. **Programação Front End**. Maringá-PR: Unicesumar, 2018.

Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas.

I. Para garantir a melhor compatibilidade com os navegadores em HTML5, os desenvolvedores devem seguir regras específicas de codificação

### PORQUE

II. Práticas como fechar todas as tags, usar letras minúsculas consistentemente e incluir elementos essenciais são requisitos da versão HTML5 que melhoram a compatibilidade entre diferentes navegadores.

A respeito dessas asserções, assinale a opção correta:

### ALTERNATIVAS

- ☐ As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- ☒ As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- ☐ A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- ☐ A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- ☐ As asserções I e II são proposições falsas.

### 4ª QUESTÃO

O Doctype indica para o navegador e para outros meios qual a especificação de código utilizar. Ele não é uma tag do HTML, mas uma instrução para que o browser tenha informações sobre qual versão de código a marcação foi escrita. Esta declaração deve ser a primeira linha em um documento HTML e é fundamental para garantir que o navegador interprete o código corretamente. Ao longo da evolução da web, diferentes versões de Doctype foram utilizadas, refletindo a evolução dos padrões HTML. No HTML5, a declaração foi simplificada para `<!DOCTYPE html>`, em contraste com as declarações mais longas e complexas das versões anteriores que exigiam referências a DTDs (Document Type Definitions) específicos.

Fonte: TOKUMOTO, R. C. **Programação Front End**. Maringá, PR: UniCesumar, 2018.

Com base no texto apresentado, sobre a função do Doctype em um documento HTML, assinale a alternativa correta:

#### ALTERNATIVAS

- ☐ Definir o título da página web.
- ☐ Definir a estrutura do corpo do documento.
- ☐ Estabelecer as regras de CSS a serem aplicadas.
- ☐ Configurar as meta tags para otimização em motores de busca.
- ☒ Indicar ao navegador qual especificação de código HTML está sendo utilizada.

#### 5ª QUESTÃO

Em jogos digitais simples baseados em HTML e JavaScript, é comum vermos a utilização de estruturas organizadas para armazenar dados, bem como a aplicação de funções para modular o comportamento do jogo. Uma prática recorrente é definir perguntas e alternativas em um array de objetos e criar uma função que manipula o DOM para exibir os dados dinamicamente. Com o uso de `getElementById` e `querySelector`, esses dados podem ser distribuídos em elementos da interface. O uso de *arrow functions* pode tornar o código mais enxuto, principalmente quando se lida com comportamentos simples, como gerar números aleatórios ou retornar valores diretos.

Durante o desenvolvimento de um jogo de perguntas e respostas com JavaScript, um estudante decidiu criar uma função para sortear uma pergunta do array perguntas. A ideia era gerar um número aleatório para selecionar uma posição válida dentro do array.

Fonte: NOEL, A. A. - **Programação Front End**. Florianópolis, SC: Arquê, 2025.

Com base no texto apresentado, considerando as boas práticas e os recursos da linguagem JavaScript moderna (ES6), assinale qual das alternativas abaixo representa a maneira mais apropriada e concisa de criar essa função:

#### ALTERNATIVAS

```
☐ const numeroAleatorio = () => Math.floor(Math.random() + perguntas.length);  
☐ const numeroAleatorio = perguntas => { return Math.random() * perguntas.length; }  
☒ const numeroAleatorio = perguntas => Math.floor(Math.random() * perguntas.length);  
  
☐ function numeroAleatorio() {  
    return perguntas.length / Math.random();  
}  
  
☐ function numeroAleatorio(perguntas) {  
    return Math.floor(Math.random()) * perguntas.length;  
}
```

## 6ª QUESTÃO

Tag é o nome dado aos comandos aceitos na linguagem html, funcionam de forma um pouco diferente de outras linguagens de programação compiladas, pois tags mal utilizadas não impedem que o código em si funcione. Assim, o navegador tem o princípio de interpretar e estruturar todas as tags que ele possa compreender e ignorar tags incorretas ou mal estruturadas de acordo com sua sintaxe. A lista de tags é longa e varia em função da versão da linguagem html e do navegador, que pode ou não ser totalmente compatível com cada tag. Nunca é cem por cento garantido que um código html seja totalmente funcional em qualquer equipamento, pois as diferentes opções de navegadores e suas versões e atualizações instaladas interferem no grau de compatibilidade entre navegador e código html.

Fonte: Elaborado pelo professor, 2025.

Fonte: TOKUMOTO, Ronie Cesar. **Programação Front End**. Maringá-PR: Unicesumar, 2018.

Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas.

I. O HTML é mais tolerante a erros de codificação do que linguagens de programação compiladas.

### PORQUE

II. Os navegadores são projetados para interpretar as tags que reconhecem e ignorar aquelas que estão mal estruturadas ou que não fazem parte da versão HTML que suportam.

A respeito dessas asserções, assinale a opção correta:

## ALTERNATIVAS

- ☐ As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- ☒ As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- ☐ A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- ☐ A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- ☐ As asserções I e II são proposições falsas.

## 7ª QUESTÃO

HTML5 introduziu um método para permitir que web designers possam assumir o controle sobre a janela de exibição, por meio do <meta> tag. Você deve incluir o elemento <meta> dentro da área de cabeçalho <HEAD> para indicar as propriedades de exibição em todas as suas páginas web: <meta name="viewport" content="width=device-width, initial-scale=1.0">. Um <meta> elemento viewport dá as instruções do navegador sobre como controlar dimensões e escalonamento da página.

Fonte: TOKUMOTO, R. C. **Programação Front End**. Maringá, PR: UniCesumar, 2018.

Com base no texto apresentado, sobre a função da tag <meta> com o atributo viewport no HTML5, assinale a alternativa correta:

#### ALTERNATIVAS

- ☐ Estabelecer os padrões de cores do website.
- ☐ Definir a estrutura básica do documento HTML.
- ☐ Determinar quais navegadores podem exibir a página.
- ☐ Configurar o acesso ao banco de dados da aplicação web.
- ☒ Permitir que os web designers assumam controle sobre as dimensões e escalonamento da exibição da página.

#### 8ª QUESTÃO

O desenvolvimento de interfaces web responsivas é essencial na atualidade, dada a variedade de dispositivos utilizados para acesso à internet. Uma das estratégias para adaptar o layout ao tamanho da tela é o uso de media queries em CSS.

Fonte: Elaborado pelo professor, 2025.

Considere o seguinte trecho de código:

```
@media screen and (max-width: 800px) {  
  main {  
    flex-direction: column;  
  }  
}
```

Com base no funcionamento das media queries e nos princípios de design responsivo, analise as afirmativas abaixo:

- I. A propriedade flex-direction: column será aplicada apenas se a mídia for de tipo "tela" e a largura máxima for de 800 pixels.
- II. Fora dessa condição (por exemplo, em uma tela de 1024px de largura), o layout manterá o estilo definido anteriormente para o elemento main.
- III. O uso da diretiva @media permite sobrescrever seletivamente propriedades CSS, sem alterar a estrutura HTML, favorecendo a adaptabilidade do layout.

É correto o que se afirma em:

#### ALTERNATIVAS

- ☐ I, apenas.
- ☐ II, apenas.
- ☐ III, apenas.
- ☐ I e II, apenas.
- ☒ I, II e III.

### 9ª QUESTÃO

Em linguagens de programação, variáveis são utilizadas para armazenar informações temporárias durante a execução de um programa. Em linguagens fortemente tipadas, como C ou Java, é necessário declarar previamente o tipo de dado que será armazenado. Já o JavaScript, por ser uma linguagem dinamicamente tipada, permite que o tipo seja determinado automaticamente com base no valor atribuído à variável.

Para declarar variáveis em JavaScript, três palavras-chave podem ser utilizadas:

var, presente nas versões mais antigas, permite escopos amplos e pode causar efeitos colaterais indesejados.  
let, ideal para variáveis que precisam ter seus valores modificados ao longo da execução.  
const, adequada para variáveis cujos valores não devem ser alterados após a atribuição inicial.

A escolha adequada entre essas palavras-chave afeta diretamente a legibilidade, segurança e previsibilidade do código, sendo um aspecto essencial da escrita de programas mais robustos e livres de erros. Durante o desenvolvimento de uma aplicação web interativa em JavaScript, um grupo de estudantes ficou responsável por criar a lógica de controle de rodadas e perguntas de um jogo de perguntas e respostas. Ao iniciar a programação, surgiram dúvidas sobre a melhor forma de declarar variáveis que armazenam a pontuação do jogador e o número total de perguntas do jogo.

Fonte: NOEL, A. A. - **Programação Front End**. Florianópolis, SC: Arquê, 2025.

Considerando boas práticas na declaração de variáveis em JavaScript moderno, analise as afirmativas:

- I. A pontuação do jogador, que pode variar ao longo do jogo, deve ser declarada utilizando let.
- II. O número total de perguntas, que permanece constante durante a execução, deve ser declarado utilizando const.
- III. Para garantir compatibilidade com navegadores antigos, é recomendável usar var em vez de let e const.

É correto o que se afirma em:

### ALTERNATIVAS

- ☐ I, apenas.
- ☒ I e II, apenas.
- ☐ II e III, apenas.
- ☐ III, apenas.
- ☐ I, II e III, apenas.

### 10ª QUESTÃO

O DOM (Document Object Model) é uma interface de programação para documentos HTML e XML. Ele representa a página como uma estrutura em árvore, na qual cada elemento é um nó da árvore, permitindo que linguagens como o JavaScript acessem e modifiquem dinamicamente o conteúdo, estrutura e estilo do documento. No contexto de aplicações interativas, como jogos web, o DOM é fundamental para manipular elementos da interface, como botões, perguntas, pontuação e opções de resposta. Utilizando métodos específicos, como `getElementById`, `querySelector`, `getElementsByTagName` e `querySelectorAll`, é possível capturar elementos HTML e modificar suas propriedades, estilos e comportamentos.

Por meio desses recursos, torna-se viável criar interfaces responsivas e dinâmicas, atualizando o conteúdo da página sem a necessidade de recarregá-la, o que melhora significativamente a experiência do usuário. Durante o desenvolvimento de um quiz interativo em JavaScript, um estudante precisa alterar dinamicamente o texto exibido em um parágrafo com o identificador `perguntaAtual`. Para isso, ele deve capturar esse elemento do DOM e substituir seu conteúdo por uma nova string, representando a próxima pergunta.

Fonte: NOEL, A. A. - **Programação Front End**. Florianópolis, SC: Arquê, 2025.

Com base no texto apresentado, sobre qual das instruções realiza corretamente a tarefa, assinale a alternativa correta.

#### ALTERNATIVAS

- ☐ `document.getElementsByTagName("perguntaAtual").value = "Nova pergunta";`
- ☒ `document.getElementById("perguntaAtual").innerHTML = "Nova pergunta";`
- ☐ `document.querySelector("#perguntaAtual").textContent = "Nova pergunta";`
- ☐ `document.getElementsByTagName("perguntaAtual")`  
0
- ☐ `.innerText = "Nova pergunta";`
- ☐ `let perguntaAtual = getElementById("perguntaAtual").innerHTML = "Nova pergunta";`