

Questões de Revisão para Prova 2

Desenvolvimento Web I – DSM - Prof. Arley

Revisão das aulas 1 a 4 do material

1 – Ao criar uma página web, é essencial entender a estrutura básica de um documento HTML. A organização correta dos elementos principais garante que o navegador interprete o conteúdo corretamente. A raiz de um documento HTML define sua estrutura fundamental, contendo marcações obrigatórias que organizam tanto os metadados quanto o conteúdo exibido ao usuário.

Todo documento HTML deve seguir uma estrutura padrão para que seja corretamente interpretado pelos navegadores. Quais são as marcações essenciais que devem estar na raiz de um documento HTML?

- a) <html>, <head>, <title>, <body>, <footer>
- b) <html>, <head>, <meta>, <body>, <footer>
- c) <html>, <head>, <body>
- d) <html>, <head>, <header>, <body>, <footer>
- e) <doctype>, <html>, <head>, <body>

2 – A declaração DOCTYPE é um elemento fundamental em um documento HTML. Embora não seja uma tag propriamente dita, sua presença no início do código é essencial para garantir que os navegadores interpretem a página corretamente. Sem essa declaração, os navegadores podem renderizar a página em um modo de compatibilidade, o que pode gerar

inconsistências na exibição dos elementos da página. Qual é a função da declaração <!DOCTYPE html> em um documento HTML?

- a) Define o idioma principal do documento para os navegadores.
- b) Declara a codificação de caracteres usada na página.
- c) Especifica ao navegador que o documento deve ser interpretado como HTML5.
- d) Ativa automaticamente todas as funcionalidades de CSS3 e JavaScript na página.
- e) Garante que todos os elementos da página sejam carregados antes da execução de scripts.

3 – Em documentos HTML, os comentários são utilizados para adicionar anotações no código que não são exibidas no navegador. Eles são úteis para documentar trechos de código, facilitar a manutenção e indicar informações importantes para desenvolvedores. No entanto, é essencial utilizar corretamente a sintaxe para evitar problemas na estrutura do documento. Qual é a sintaxe correta para inserir um comentário em um documento HTML?

- a) <!-- Este é um comentário -->
- b) /* Este é um comentário */
- c) // Este é um comentário

- d) <! Este é um comentário !>
- e) # Este é um comentário

4 - Os elementos HTML são a base da construção de páginas web. Cada elemento é composto por uma tag de abertura, que define seu nome (**tagname**), atributos, que modificam seu comportamento, e um corpo, que contém o conteúdo visível ou aninhado dentro da tag. Alguns elementos também possuem uma tag de fechamento, enquanto outros são elementos vazios, que não possuem conteúdo interno.

Considere a seguinte estrutura de um elemento HTML:

```
<a href="http://www.test.com"  
target="_blank">
```

Clique aqui

Com base nesse código, qual das afirmações a seguir é correta?

- a) O nome da tag (**tagname**) é `<a>`, o único atributo é `href`, e o corpo contém o texto "Clique aqui".
- b) O nome da tag (**tagname**) é `<a>`, e como possui atributos, não precisa de um corpo.
- c) O nome da tag (**tagname**) é `href`, pois ele define o comportamento do elemento.
- d) O nome da tag (**tagname**) é `<a>`, os atributos são `href` e `target`, e o corpo contém o texto "Clique aqui".**
- e) O nome da tag (**tagname**) é `<a>`, mas como o elemento não possui atributos

obrigatórios, o navegador ignora sua funcionalidade.

5 - A seção `<head>` de um documento HTML contém metadados e informações que auxiliam na configuração e comportamento da página, mas que não são exibidas diretamente para o usuário. Alguns elementos são exclusivos dessa seção, enquanto outros devem ser inseridos apenas no `<body>`, onde o conteúdo principal da página é estruturado. Quais dos seguintes elementos devem estar exclusivamente dentro da tag `<head>` de um documento HTML?

- a) `<header>`, `<meta>`, `<nav>`, `<title>`, `<script>`
- b) `<title>`, `<section>`, `<meta>`, `<link>`, `<footer>`
- c) `<title>`, `<meta>`, `<link>`, `<style>`, `<script>`**
- d) `<meta>`, `<aside>`, `<link>`, `<style>`, `<title>`
- e) `<title>`, `<div>`, `<meta>`, `<script>`, `<link>`

6 - A tag `<meta>` é utilizada dentro do `<head>` de um documento HTML para fornecer metadados sobre a página. Esses metadados podem incluir informações sobre codificação de caracteres, descrição para motores de busca, configuração de viewport para responsividade e até mesmo redirecionamento automático. Embora não seja visível para os usuários, a tag `<meta>` desempenha um papel essencial na acessibilidade, SEO e compatibilidade da página. Qual das alternativas a seguir não

representa um uso válido da tag <meta> em um documento HTML?

- a) <meta charset="UTF-8"> – Define a codificação de caracteres do documento.
- b) <meta name="viewport" content="width=device-width, initial-scale=1.0"> – Configura a exibição do site em dispositivos móveis.
- c) <meta name="description" content="Página sobre tecnologia e inovação."> – Fornece uma descrição do site para mecanismos de busca.
- d) <meta http-equiv="refresh" content="5; url=https://www.exemplo.com"> – Redireciona automaticamente o usuário após 5 segundos.
- e) <meta tag="header" content="Cabeçalho do site"> – Define um cabeçalho personalizado para exibição no navegador.

7 - No HTML, as tags <div> e são utilizadas para agrupar elementos e aplicar estilos, mas possuem diferenças fundamentais em sua forma de uso e comportamento na página. A escolha entre essas tags pode impactar diretamente a estrutura e a apresentação do conteúdo. Qual é a principal diferença entre as tags <div> e no HTML?

- a) <div> é usada para criar botões interativos, enquanto serve para criar parágrafos de texto.

b) <div> é um elemento de bloco que ocupa toda a largura disponível, enquanto é um elemento inline que ocupa apenas o espaço necessário para o conteúdo.

- c) <div> e possuem exatamente a mesma função e podem ser usadas indistintamente.
- d) <div> é usada apenas dentro da <head>, enquanto pode ser usada dentro do <body>.
- e) <div> é utilizada apenas para criar tabelas, enquanto é usada para estilizar textos.

8 - No HTML, existem diferentes tipos de elementos para estruturar o conteúdo de uma página. A tag <p> é usada para definir parágrafos de texto, enquanto a tag <div> é mais genérica e utilizada para agrupar elementos e estruturar o layout da página. Saber a diferença entre essas duas tags ajuda a construir um HTML mais semântico e organizado. Qual das alternativas a seguir melhor descreve a diferença entre as tags <div> e <p> no HTML?

- a) <div> é um elemento de bloco usado para agrupar outros elementos, enquanto <p> é um elemento de bloco específico para parágrafos de texto.
- b) <div> e <p> são exatamente iguais e podem ser usadas de forma intercambiável sem impacto na estrutura da página.

- c) <div> deve ser usada apenas para criar menus de navegação, enquanto <p> é utilizada exclusivamente dentro da tag <body>.
- d) <div> é um elemento inline, enquanto <p> é um elemento de bloco.
- e) <p> pode conter qualquer tipo de elemento HTML dentro dele, incluindo <div>, sem restrições

9 - A tag <a> é utilizada para criar hiperlinks em um documento HTML, permitindo que os usuários naveguem entre páginas da web, documentos ou até seções dentro da mesma página. Essa tag pode receber diversos atributos que influenciam o comportamento do link, como definir o destino, a forma de abertura e a acessibilidade do link. Qual das alternativas a seguir apresenta atributos válidos e comumente utilizados na tag <a> no HTML?

- a) href, src, target, rel
- b) href, alt, src, download
- c) href, target, name, id
- d) src, target, class, id
- e) href, value, placeholder, method

10 - A tag é utilizada para exibir imagens em páginas HTML. Diferente de outras tags, ela é um elemento vazio, ou seja, não possui uma tag de fechamento. Para que uma imagem seja exibida corretamente, a tag deve conter atributos essenciais que definem a origem da imagem e informações adicionais para acessibilidade e layout. Qual das alternativas a seguir contém apenas

atributos válidos e comumente utilizados na tag no HTML?

- a) src, href, alt, width, height
- b) src, placeholder, value, alt, target
- c) source, href, alt, width, length
- d) src, alt, width, height, title
- e) src, id, action, method, rel

11 - As listas não ordenadas () são utilizadas no HTML para agrupar itens que não seguem uma sequência específica, como menus de navegação ou listas de categorias. Qual das alternativas a seguir melhor descreve o funcionamento da tag no HTML?

- a) A tag cria uma lista numerada, onde os itens são organizados em ordem crescente.
- b) A tag só pode ser utilizada dentro de uma tag <nav>, pois é exclusiva para menus de navegação.
- c) A tag precisa obrigatoriamente ser usada com o atributo type="circle" para funcionar corretamente.
- d) A tag não pode conter elementos dentro dela, apenas <p> e <div>.
- e) A tag é usada para criar listas sem ordem específica, e seus itens são representados por padrão com marcadores visuais (bullet points).

12 - A tag é usada para criar listas ordenadas, onde os itens seguem uma sequência numérica, alfabética ou romana. Essa estrutura é útil quando a ordem dos

itens importa, como instruções passo a passo ou classificações. Qual das alternativas a seguir corretamente descreve o comportamento da tag `` no HTML?

- a) A tag `` cria uma lista onde os itens aparecem sem ordem definida, apenas com marcadores visuais (bullet points).
- b) A tag `` permite que os itens da lista sejam organizados em ordem específica, como numeração crescente ou alfabética.
- c) A tag `` só pode ser utilizada se for combinada com a tag ``.
- d) A tag `` não pode conter atributos como `start` ou `type`, pois seu comportamento é fixo.
- e) A tag `` funciona da mesma forma que ``, sem diferença no estilo padrão dos itens.

13 - No HTML, é possível combinar listas ordenadas (``) e não ordenadas (``) para estruturar informações de forma hierárquica. Esse recurso é útil, por exemplo, para representar tópicos com subtópicos ou categorias com subitens.

Considere o seguinte código HTML:

```
<ol>
  <li>Passo 1</li>
  <li>Passo 2</li>
<ul>
  <li>Detalhe do Passo 2.1</li>
```

`Detalhe do Passo 2.2`

``

`Passo 3`

``

Qual é o **erro** nesse código e como ele pode ser corrigido?

- a) O código está correto, pois `` e `` podem ser usados livremente juntos.
- b) O erro é que a `` não pode ser usada dentro de um ``, e a correção é remover a lista não ordenada.
- c) O erro é que os itens `` da `` estão fora de um `` da ``, e a correção correta é colocar a `` dentro de um ``.
- d) O erro é que o `` deveria ter um atributo `type="A"` para suportar a combinação de listas.
- e) O erro é que `` e `` não podem ser combinados de forma alguma no HTML.

14 - No HTML, as tabelas podem ser organizadas em seções para facilitar a estruturação e a estilização dos dados. As tags `<thead>`, `<tbody>` e `<tfoot>` permitem separar os cabeçalhos, o corpo e o rodapé da tabela, respectivamente. Essa organização melhora a acessibilidade e a legibilidade da tabela, além de permitir estilizações diferenciadas para cada parte.

Considere a seguinte estrutura de tabela HTML.

`<table>`

```

<thead>
<tr>
<th>Produto</th>
<th>Preço</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notebook</td>
<td>R$ 4.500</td>
</tr>
<tr>
<td>Mouse</td>
<td>R$ 90</td>
</tr>
</tbody>
<tfoot>
<tr>
<td>Total</td>
<td>R$ 4.590</td>
</tr>
</tfoot>
</table>

```

Com base nesse código, qual das afirmações a seguir está correta sobre o uso das tags <thead>, <tbody> e <tfoot>?

- a) O <thead> é opcional e pode ser omitido sem prejudicar a estrutura da tabela, mas <tfoot> é obrigatório.
- b) O <tbody> é responsável apenas pelos dados numéricos da tabela, enquanto <thead> e <tfoot> lidam com texto.
- c) O <tfoot> deve sempre estar posicionado após o <tbody>, independentemente da ordem em que os dados aparecem no código.
- d) A tag <tfoot> precisa obrigatoriamente conter apenas uma única linha (<tr>), independentemente da quantidade de colunas.
- e) O <thead>, <tbody> e <tfoot> são usados apenas para estilização e não afetam a estrutura da tabela.

15 - No HTML, as tabelas utilizam os atributos colspan e rowspan para mesclar células e melhorar a organização visual das informações. O atributo colspan permite que uma célula ocupe múltiplas colunas, enquanto o rowspan permite que uma célula ocupe múltiplas linhas.

Considere o seguinte código HTML de uma tabela:

```

<table border="1">
<tr>
<th>Nome</th>
<th colspan="2">Contato</th>
</tr>
<tr>

```

```

<td>Ana</td>

<td>Email</td>

<td>Telefone</td>

</tr>

<tr>

<td rowspan="2">Bruno</td>

<td>bruno@email.com</td>

<td>9999-8888</td>

</tr>

<tr>

<td>bruno.site.com</td>

<td>9999-7777</td>

</tr>

</table>

```

- a) A célula "Contato" ocupa duas colunas porque o atributo colspan="2" foi utilizado.
- b) A célula "Bruno" ocupa duas colunas porque o atributo rowspan="2" foi utilizado.
- c) O atributo colspan pode ser aplicado apenas em <th>, enquanto rowspan pode ser usado apenas em <td>.
- d) O atributo colspan faz com que uma célula ocupe várias linhas, enquanto rowspan faz com que uma célula ocupe várias colunas.
- e) O código HTML contém um erro, pois rowspan e colspan não podem ser usados na mesma tabela.

16 - Os formulários HTML são fundamentais para a interação entre usuários e aplicações web. A tag <form> define um formulário que pode conter diversos elementos, como campos de entrada (<input>), seletores (<select>), áreas de texto (<textarea>) e botões (<button>). Além disso, a tag <form> possui atributos importantes que influenciam o envio dos dados.

Considere o seguinte código de um formulário HTML:

```

<form action="server.php"
method="post">

<label for="nome">Nome:</label>

<input type="text"
       id="nome" name="nome" required>

<label for="email">E-mail:</label>

<input type="email"
       id="email" name="email" required>

<button type="submit">Enviar</button>

</form>

```

Com base no código acima, qual das alternativas está correta sobre a marcação <form>?

- a) O atributo method="post" indica que os dados do formulário serão enviados de forma visível na URL.
- b) O atributo action="server.php" define a página que receberá os dados enviados pelo formulário.

- c) O botão <button type="submit"> é opcional e o formulário será enviado automaticamente ao preencher todos os campos.
- d) O atributo required nos campos <input> não tem efeito prático e depende de validação JavaScript para funcionar.
- e) O formulário pode ser enviado sem o atributo method, mas, nesse caso, os dados não serão transmitidos ao servidor.

17 - A tag <input> é utilizada em formulários HTML para capturar diferentes tipos de dados inseridos pelo usuário. Ela possui vários atributos que definem o comportamento do campo de entrada, como type, name, placeholder, required, maxlength, entre outros.

Considere o seguinte código HTML:

```
<form>

<label for="usuario">Usuário:</label>

<input type="text"
       id="usuario" name="usuario"
       placeholder="Digite seu nome"
       required maxlength="20">

<label for="senha">Senha:</label>

<input type="password" id="senha"
       name="senha" required>

<label for="email">E-mail:</label>

<input type="email" id="email"
```

```
       name="email">

<button type="submit">Enviar</button>

</form>
```

Com base no código acima, qual das afirmações é correta sobre a tag <input>?

- a) O atributo type="password" permite que o campo aceite apenas números.
- b) O campo de e-mail (type="email") pode aceitar qualquer entrada de texto, sem restrições.
- c) O atributo required na senha faz com que o campo seja preenchido automaticamente com um valor padrão.
- d) O atributo placeholder define um valor fixo que será enviado ao servidor caso o usuário não insira nenhum dado.
- e) O atributo maxlength="20" limita a quantidade de caracteres inseridos no campo "Usuário".

18 - A tag <select> é utilizada para criar listas suspensas (dropdowns) em formulários HTML, permitindo que os usuários escolham uma opção entre várias disponíveis. Os elementos <option> dentro do <select> representam cada uma das opções que podem ser selecionadas. Além disso, atributos como name, value, selected e disabled influenciam o comportamento do campo.

Considere o seguinte código HTML:

```
<form>

<label for="cidade">
```

Escolha sua cidade:

```
</label>

<select id="cidade" name="cidade">

    <option value="sp">São Paulo</option>
    <option value="pr">Paraná</option>
    <option value="mg" selected>
        Minas Gerais
    </option>
    <option value="ba" disabled>
        Bahia
    </option>
</select>

<button type="submit">Enviar</button>

</form>
```

- a) O atributo name="cidade" define o rótulo que será exibido na tela para o usuário.
- b) O atributo disabled na opção "Bahia" permite que ela seja escolhida, mas com uma aparência diferente.
- c) O campo <select> só pode conter opções de texto e não aceita valores numéricos no atributo value.
- d) A opção "Minas Gerais" será a selecionada por padrão ao carregar a página devido ao atributo selected.
- e) Se nenhuma opção for selecionada, o formulário enviará um valor vazio ("") para o servidor.

19 - A tag <button> é utilizada para criar botões interativos em páginas HTML. Ela pode ser usada para enviar formulários, acionar scripts em JavaScript ou simplesmente servir como um elemento clicável. O comportamento do botão pode ser modificado pelo atributo type, que define sua função dentro de um formulário.

Considere o seguinte código HTML

```
<form action="server.php"
method="post">

    <label for="nome">Nome:</label>

    <input type="text" id="nome"
name="nome" required>

    <button type="submit">Enviar</button>
    <button type="reset">Limpar</button>
    <button type="button"
onclick="alert('Botão clicado!')">
        Clique aqui
    </button>
</form>
```

Com base no código acima, qual das afirmações é correta sobre a tag <button>?

- a) O botão com type="submit" envia os dados do formulário para o arquivo server.php.
- b) O botão com type="reset" tem a mesma função do botão submit, mas envia os dados em branco.
- c) O botão com type="button" sempre recarrega a página ao ser clicado,

independentemente do código JavaScript.

- d) Se nenhum atributo type for definido, o botão terá o comportamento padrão de type="button".
- e) O botão reset só funciona corretamente se estiver associado a um evento JavaScript específico.

20 - Os botões de opção (radio buttons) permitem que os usuários escolham uma única opção entre várias dentro de um grupo. Eles são utilizados em formulários para coletar respostas exclusivas, como selecionar um gênero. Para que os botões de opção funcionem corretamente como um grupo, é necessário utilizar o mesmo atributo name para todos os botões relacionados.

Considere o seguinte código HTML.

```
<form>

<label>Escolha seu plano:</label>

<input type="radio" id="planoA"
       name="plano" value="A">

<label for="planoA">Plano A</label>

<input type="radio" id="planoB"
       name="plano" value="B" checked>

<label for="planoB">Plano B</label>

<input type="radio" id="planoC"
       name="plano" value="C">

<label for="planoC">Plano C</label><br>

<button type="submit">Enviar</button>
```

</form>

Com base no código acima, qual das afirmações é correta sobre os botões de opção (radio buttons)?

- a) Se um usuário selecionar o "Plano C", os três botões de opção permanecerão marcados ao mesmo tempo.
- b) Cada botão de opção deve ter um name diferente para que o formulário envie corretamente a escolha do usuário.
- c) O atributo value é opcional nos botões de opção, pois o formulário enviará automaticamente o texto dentro do <label>.
- d) O botão de "Plano A" será automaticamente selecionado caso o atributo checked não esteja presente em nenhuma opção.
- e) O atributo checked no botão do "Plano B" garante que ele será selecionado por padrão ao carregar a página.

21 - O CSS (Cascading Style Sheets) é usado para estilizar elementos HTML, permitindo personalizar cores, tamanhos, espaçamentos e muito mais. Cada regra CSS segue a estrutura:

```
seletor {
    propriedade: valor;
}
```

Diferentes propriedades CSS afetam diferentes aspectos da página, como layout, tipografia e interatividade.

Considere o seguinte trecho de código CSS aplicado a um elemento <div>:

```
div {  
    width: 200px;  
    height: 100px;  
    background-color: blue;  
    border: 2px solid black;  
    margin: 10px;  
    padding: 20px;  
}
```

Com base no código acima, qual das afirmações é correta sobre as propriedades CSS utilizadas?

- a) A propriedade width define a largura total do elemento, incluindo borda, preenchimento e margem.
- b) A propriedade border define apenas a cor da borda do elemento, sem especificar o tamanho ou estilo.
- c) A propriedade margin define um espaço interno entre o conteúdo do elemento e sua borda.
- d) A propriedade background-color define a cor de fundo do elemento.
- e) A propriedade padding define o espaço externo entre o elemento e os outros elementos ao seu redor.

22 - Os seletores CSS são usados para aplicar estilos a elementos HTML específicos. Existem diversos tipos de seletores, como os básicos (elemento, .classe, #id), os combinadores (>, +, ~), e os pseudo-seletores (:hover, :nth-child()). A escolha do seletor adequado influencia diretamente na aplicação dos estilos.:

Considere o seguinte código HTML e CSS:

```
<!DOCTYPE html>  
<html lang="pt">  
<head>  
    <style>  
        p { color: blue; }  
        .destaque { color: red; }  
        #importante { color: green; }  
        div p { font-weight: bold; }  
    </style>  
</head>  
<body>  
    <p>Um</p>  
    <p class="destaque">Dois</p>  
    <p id="importante">Três</p>  
    <div>  
        <p>Quatro</p>  
    </div>  
</body>  
</html>
```

Com base no código acima, qual das afirmações é correta sobre os seletores utilizados?

- a) Todos os parágrafos serão estilizados com a cor azul, pois a regra `p { color: blue; }` tem prioridade sobre as outras.
- b) O "Três" será exibido na cor verde porque o seletor `#importante` tem maior especificidade do que `p` e `.destaque`.
- c) O "Quatro" será exibido apenas em negrito, sem alteração de cor, pois a regra `div p { font-weight: bold; }` tem a maior especificidade.
- d) O "Dois" será azul porque o seletor `p { color: blue; }` se aplica antes do seletor `.destaque { color: red; }`.
- e) O "Quatro" não será afetado pelo CSS porque está dentro de uma `<div>`, e não diretamente no `<body>`.

23 - O CSS permite definir layouts para páginas da web por meio de diferentes técnicas, como o uso de `display`, `position`, `flexbox` e `grid`. Essas propriedades possibilitam organizar elementos na tela de forma responsiva e estruturada.

Considere o seguinte código CSS da classe `container` aplicada a uma `<div>` e o CSS da classe `item` aplicada aos elementos filhos dessa `<div>`:

```
.container {  
    display: flex;  
    justify-content: center;
```

```
    align-items: center;  
    width: 400px;  
    height: 200px;  
    border: 1px solid black;  
}  
  
.item {  
    width: 50px;  
    height: 50px;  
    background-color: blue;  
}
```

Com base nesse código, qual das afirmações é correta sobre a exibição dos elementos dentro do `.container`?

- a) Os elementos `.item` ficarão alinhados no canto superior esquerdo da `.container`, pois esse é o comportamento padrão do `display: flex`.
- b) O `align-items: center` não afeta a posição vertical dos elementos `.item`, pois o `display: flex` só controla o alinhamento horizontal.
- c) O `display: flex` faz com que os elementos `.item` se sobreponham uns aos outros, ocupando a mesma posição na tela.
- d) O `border: 1px solid black;` aplicado à `.container` impede que os elementos `.item` fiquem centralizados dentro dela.

- e) O uso de justify-content: center centraliza os elementos .item horizontalmente dentro da .container.

24 - No modelo de layout flexível (display: flex), as propriedades justify-content e align-items são usadas para controlar o posicionamento dos elementos filhos dentro do container flexível. Enquanto justify-content controla o alinhamento ao longo do eixo principal (horizontal, por padrão), align-items ajusta o alinhamento ao longo do eixo secundário (vertical, por padrão).

Dado o seguinte código CSS:

```
.container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    width: 500px;
    height: 300px;
    border: 2px solid black;
}

.item {
    width: 80px;
    height: 80px;
    background-color: red;
}
```

Com base nesse código, qual das afirmações é correta sobre a disposição dos elementos .item dentro do .container?

- a) Os elementos .item ficarão distribuídos uniformemente ao longo do eixo vertical, pois justify-content: space-between; garante esse espaçamento.
- b) Os elementos .item ficarão distribuídos horizontalmente com espaços iguais entre eles e alinhados verticalmente ao centro do .container.
- c) A propriedade align-items: center; força os elementos .item a ficarem alinhados no topo do .container, independentemente da altura definida.
- d) justify-content: space-between; faz com que os elementos .item se sobreponham, pois força o espaçamento entre eles, ignorando as dimensões do container.
- e) align-items: center; não tem efeito, pois só funciona quando display: grid; é utilizado no .container.