Centro Universitário UNA
Aimorés
Usabilidade, Desenvolvimento Web, Mobile e Jogos
Wesley Dias Maciel
2025.02

Exercícios de Revisão

Nome:_____ Data:_____

HTML Semântico

1. A HTML semântica melhora a acessibilidade e a organização do conteúdo, utilizando elementos que descrevem claramente sua função (como <header>, <main>, <footer>, <nav> e <article>). Crie um trecho de código HTML que represente a estrutura básica de uma página de blog com cabeçalho, conteúdo principal e rodapé, utilizando elementos semânticos.

Formulários em HTML

2. Formulários (<form>) são elementos HTML usados para coletar dados do usuário. Elementos como <input>, <textarea> e <select> permitem diferentes tipos de entrada em um formulário. Desenvolva um formulário HTML que permita ao usuário cadastrar-se em um evento cultural, incluindo campos para nome, e-mail, tipo de ingresso e uma área para observações.

Estilização com CSS

3. CSS permite controlar a aparência dos elementos HTML, como cores, tamanhos, margens e posicionamento. Crie um trecho de código CSS que estilize um botão com cor de fundo azul, texto branco, bordas arredondadas, *padding* e efeito de hover que muda a cor de fundo para verde.

Layout com Flexbox

4. Flexbox é uma técnica de *layout* em CSS que facilita o alinhamento e distribuição de elementos em uma página. Escreva um código HTML e CSS que organize três caixas lado a lado e centralizadas horizontalmente usando Flexbox.

Responsividade com Media Queries

5. Media queries permitem adaptar o layout da página a diferentes tamanhos de tela, como celulares e *desktops*. Crie um trecho de código CSS que altere a cor de fundo de um elemento para azul em telas maiores que 768px e para amarelo em telas menores.

Manipulação de DOM com JavaScript

6. O DOM (Document Object Model) representa a estrutura da página. Com JavaScript, é possível modificar elementos do DOM dinamicamente. Escreva um código JavaScript que altere o texto de um parágrafo com id "mensagem" quando o usuário clicar em um botão.

Eventos em JavaScript

7. Eventos permitem interações com o usuário, como cliques, teclas pressionadas e envio de formulários. Crie um código JavaScript que exiba um alerta com o texto "Formulário enviado!" quando um formulário for submetido.

Validação de Formulário

8. Validar dados antes do envio evita erros e melhora a experiência do usuário. Pode ser feito com HTML e JavaScript. Desenvolva um código JavaScript que verifique se o campo de *e-mail* de um formulário está preenchido corretamente antes de permitir o envio.

CSS Grid

9. CSS Grid é uma técnica para criar *layouts* bidimensionais, permitindo organizar elementos em linhas e colunas com facilidade. É ideal para estruturas mais complexas que exigem controle preciso do posicionamento. Crie um trecho de código HTML e CSS que utilize CSS Grid para organizar uma galeria de imagens em três colunas e duas linhas, com espaçamento entre os itens.

Bootstrap

10. O Bootstrap é um *framework* CSS que facilita a criação de interfaces responsivas e contemporâneas. Ele oferece componentes prontos e um sistema de *grid* baseado em Flexbox. Desenvolva um trecho de código HTML que utilize o sistema de *grid* do Bootstrap para criar uma página com três colunas que se reorganizam em uma única coluna em telas menores que 768px.

React com Next.js – Estrutura de Página

11. O Next.js é um framework para React que permite renderização no lado do servidor, rotas automáticas e otimização de desempenho. Crie um componente de página em Next.js que exiba um título, uma descrição e um botão.

React com Next.js – Hook useState

12. O *hook useState* permite adicionar estado a componentes funcionais no React. Em Next.js, ele é usado da mesma forma que no React puro. Desenvolva um componente funcional em Next.js que exiba um contador. O contador deve começar em 0 e ser incrementado a cada vez que o usuário clicar em um botão.

React com Next.js – Listagem com .map()

13. O método .map() é usado para iterar sobre *arrays* (vetores) e renderizar elementos dinamicamente em React. É útil para exibir listas de dados como produtos, usuários ou postagens. Crie um componente funcional em Next.js que receba uma lista de nomes e exiba cada nome dentro de um item de lista (<li>), utilizando o método .map().

React Native – Layout com Flexbox

14. O React Native utiliza Flexbox como principal sistema de layout, permitindo organizar elementos de forma responsiva em dispositivos móveis. Crie um componente em React

Native que exiba três caixas coloridas lado a lado, centralizadas horizontalmente e com espaçamento entre elas.

React Native – Hook useState

15. O *hook useState* é utilizado em React Native para controlar o estado dos componentes, como textos, cores ou visibilidade. Desenvolva um componente em React Native que exiba um texto e um botão. Ao clicar no botão, o texto deve mudar de "Olá!" para "Você clicou!".

React Native – Listagem com .map()

16. Em React Native, o método .map() é utilizado para renderizar listas de componentes, como textos ou botões, a partir de *arrays* (vetores). Desenvolva um componente em React Native que exiba uma lista de tarefas (*strings*) usando o método .map(), em que cada tarefa aparece dentro de um componente.