

Atividade 1: Descrevendo Efeitos Visuais com JavaScript

Resposta:

O JavaScript é amplamente utilizado para criar efeitos visuais interativos e dinâmicos em páginas web. Ele permite alterar estilos CSS de elementos HTML em tempo real, o que possibilita animações, transições, efeitos de fade-in/out, movimentação de elementos, entre outros.

Como funciona:

- O JavaScript manipula o DOM (Document Object Model), acessando e alterando propriedades de elementos da página.
- Pode mudar classes, estilos, posições, tamanhos, opacidade etc.

Exemplo:

js

```
document.getElementById("meuElemento").style.backgroundColor = "red";
```

Efeitos simples:

- Mostrar/esconder elementos
- Alterar cores ao passar o mouse
- Aplicar transições suaves

Bibliotecas úteis:

- jQuery: Facilita a manipulação do DOM com efeitos como `.fadeIn()`, `.slideUp()` etc.
- GSAP (GreenSock): Criação de animações avançadas com controle de tempo, curvas e sequências.
- Anime.js: Foco em animações SVG, CSS, DOM, canvas.

Fontes:

- [MDN JavaScript](#)
 - [Documentação do jQuery](#)
 - [GSAP](#)
-



Atividade 2: Analisando a Validação de Formulários com JavaScript

Resposta:

A validação de formulários com JavaScript ocorre no lado do cliente, antes dos dados serem enviados ao servidor. Isso melhora a experiência do usuário e evita tráfego desnecessário.

Como funciona:

- JavaScript captura os valores dos campos usando `document.querySelector()` ou `document.getElementById()`.
- Verifica se os dados atendem a critérios específicos (ex: campos obrigatórios, formatos corretos).

Exemplo:

Js

```
let email = document.getElementById("email").value;

if (!email.includes("@")) {

    alert("Digite um e-mail válido.");

}
```

Expressões Regulares:

Usadas para validar formatos como CPF, e-mail, senha, telefone.

Exemplo de expressão para e-mail:

Js

```
const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
```

Essa expressão verifica se há texto antes e depois de um @ e um domínio válido.

Fontes:

- [W3Schools - JavaScript Forms](#)
 - [Regex101](#)
-



Atividade 3: Detalhando a Arquitetura Cliente-Servidor na Web

Resposta:

Na web, a comunicação ocorre entre dois principais elementos: o cliente (navegador) e o servidor (onde está hospedado o site).

Como funciona:

1. O usuário digita uma URL.
2. O navegador envia uma requisição HTTP para o servidor.
3. O servidor processa e responde com arquivos (HTML, CSS, JS).
4. O navegador exibe a página.

Papéis:

- Cliente: Solicita recursos e interpreta os arquivos recebidos.
- Servidor: Armazena os arquivos e responde às solicitações com conteúdo.

Fluxo:

1. Navegador solicita www.site.com.

2. O DNS converte para um IP.
3. O navegador faz a requisição GET / para o IP.
4. O servidor responde com a página.
5. O navegador renderiza.

Conceitos:

- HTTP Request: Solicitação do cliente (ex: GET, POST).
- HTTP Response: Resposta do servidor (com códigos como 200 OK, 404 Not Found).
- DNS: Traduz nomes de domínio em endereços IP.

Fontes:

- [MDN HTTP](#)
- [How DNS works](#)