

Atividade - Git e GitHub

Respostas

1. É um provedor HTTP que hospeda sites, disponibilizando para o uso do cliente, sendo ele estático (Envia os arquivos tal como estão no servidor para o navegador) ou dinâmico (tudo que o servidor estático já faz porém sempre se atualizando com outras aplicações, como banco de dados)

2. É usado para interpretar HTML, CSS e JavaScript vindo do servidor HTTP (Servidor Web), sendo alguns dos principais navegadores:

- Chrome

- o Vantagens: Fácil e Intuitivo, Integração nativa com serviços da Google, alta variedade de extensões.

- o Desvantagens: Alto consumo de Memória RAM, Coleta de dados do Google

- Mozilla Firefox

- o Vantagens: Código aberto, alta performance, personalizável com extensões

- o Desvantagens: Baixa compatibilidade com alguns serviços, carregamento de páginas mais lento que o do chrome.

- Microsoft Edge

- o Vantagens: Rápido e fácil de usar por usar o mesmo motor de busca do chrome, baixo consumo de bateria.

- o Desvantagens: Integração intrusiva no Windows, visual poluído, menor número de extensões que o chrome

- Safári

- o Vantagens: Otimizado para produtos da Apple, baixo consumo de energia, alta velocidade de busca.

- o Desvantagens: Disponível apenas em dispositivos Apple, quase nenhuma compatibilidade com extensões.

- Opera

- o Vantagens: VPN gratuita integrada, bloqueador de anúncio integrado, extremamente personalizável, consumo de Memória RAM e Processamento customizável, recursos que facilitam a intuição e facilidade de acesso.

- o Desvantagens: Menor base de usuários, extensões populares podem não estar disponíveis, anúncios intrusivos nos blocos de favoritos.

3. Ele permite que o navegador envie requisições (requests) ao servidor e receba respostas (responses) para exibir páginas da web.
4. Servidores armazenam arquivos .html e os disponibilizam via HTTP. Quando o navegador faz uma requisição, o servidor envia o HTML, que o navegador interpreta e renderiza como visual para o usuário.
5. É usado para marcar partes do conteúdo (ex: <p>, <a>,). Os atributos (ex: class="exemplo", href="link.html") fornecem informações extras que definem o comportamento ou estilo da tag
6. URL (Uniform Resource Locator) é o endereço de um recurso na web.
7. URLs absolutas indicam o endereço completo (incluindo o domínio), enquanto URLs relativas apontam para arquivos no mesmo site ou pasta. URLs relativas são úteis ao organizar arquivos internamente no site, enquanto as absolutas são usadas para links externos.
8. Links usados no desenvolvimento:
 - Inserir links com Clique aqui.
 - Carregar imagens: .
 - Importar scripts: <script src="js/script.js"></script>.
9. A tag <p> define um parágrafo e o navegador formata automaticamente o texto. A tag <pre> mantém a formatação exata do texto, incluindo espaços e quebras de linha, e usa fonte monoespçada.
10. Tudo que estiver dentro da tag <head> tem metadados como link para css e script enquanto o body é onde tem todo conteúdo visual da página.
11. O navegador pode interpretar a página de forma incorreta, causar falhas na estrutura visual, ignorar partes do código ou exibir erros de formatação.

12. Um site é um conjunto de páginas web interligadas, acessadas por um navegador. Ele pode ficar armazenado em servidores locais ou servidores na internet, chamados de servidores de hospedagem.

13. A aparência é alterada com CSS.

14. É usado para estilizar elementos HTML.

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

15. CSS in-line, interno, externo:

- Externo: arquivo separado, ex: <link rel="stylesheet" href="estilo.css">
- Interno: dentro da tag <style> no <head>
- Embutido: direto no elemento: <p style="color:red;">Texto</p>

16. Tipos de listas:

- Ordenadas: (com números)
- Não ordenadas: (com marcadores)
- De definição: <dl> (termo e descrição)

17. Tags de tabela

- <tr> define uma linha
- <th> define uma célula de cabeçalho
- <td> define uma célula de dados

18. GIT é um sistema de controle de versão que permite registrar o histórico de mudanças no código, trabalhar em equipe, criar ramificações e restaurar versões anteriores do projeto.

19. Um repositório é onde ficam armazenados os arquivos de um projeto, incluindo todo o seu histórico de versões. Pode ser local (no seu computador) ou remoto (como no GitHub).

20. Comandos do Git:

- `git init`: Transforma um diretório comum em um repositório Git. Adiciona dentro dessa pasta um subpasta oculta chamada `.git` que dentro tem metadados para o git poder rastrear os arquivos, porém precisa usar o `git add` para adicionar todos os metadados depois de usar esse comando.
- `git add`: Prepara os arquivos para ser “commitados”, não salva nada permanente.
- `git commit`: Pega tudo que está na staging area e adiciona autor, data e uma mensagem, atualizando o HEAD dentro do `.git`.

21. Após ter desenvolvido o site usando HTML, CSS, JavaScript, etc. precisa registrar um domínio, contratar um serviço de hospedagem de sites para assim os clientes poderem acessar os arquivos.

22. São os diferentes tipos de formatos de dados inseridos pelo o usuário, eles são mandados pra algum servidor ou processado por algum script.

23. Principais atributos do elemento `input`.

- `type`: Define o formato de dados que o usuário deve inserir (texto, email, senha...).
- `name`: Define o formato de dados antes de enviar para o servidor
- `value`: Define um valor padrão para o campo. Já vai estar lá quando a página carregar. O usuário pode alterar o valor se quiser.

- **placeholder:** Exibe um texto de dica dentro do campo com opacidade baixa.
- **required:** Deixa o campo obrigatório para preencher.
- **disable:** Desativa o campo para qualquer tipo de alteração, esse valor não é enviado para o servidor quando o usuário envia, aparece acinzentado.
- **readonly:** Desativa o campo para qualquer tipo de alteração, o valor é enviado ao servidor e aparece normalmente para o usuário.
- **maxlength:** Limita a quantidade de caracteres para o usuário.

24. Atributos do elemento form.

- **action:** Define a URL do arquivo que vai processar os dados que o usuário vai inserir.
- **target:** Define onde a resposta vai ser exibida, o mais comum é target: "_self".

```
<form action="processa.php" target="_blank">
```

Aqui o arquivo processa.php vai processar todos os dados e a resposta vai ser uma nova guia

25. Atributos do elemento method.

- **get:** Envia todos os dados ao servidor pela URL, ou seja, tudo fica visível caso o usuário clique na barra de navegação.

Suponha esse formulário:

```
<form method="GET" action="buscar.php">
<input type="text" name="nome" value="joao">
<button type="submit">Buscar</button>
</form>
```

Quando você clica no botão, o navegador envia o formulário assim para o servidor:

```
buscar.php?nome=joao
```

É muito usado para campo de navegação, filtros,

- post: os dados do formulário são enviados no corpo (body) da requisição HTTP, em vez de serem anexados à URL, como acontece com get.