

Trabalho Bimestral

Desenvolvimento Web (JS e PHP)



Aluno: Lucas Eduardo Ariosi.

Registro Acadêmico: 13635

Docente: Alex Morgado.

Temas: JSON e XML; Sites Dinâmicos; Requisições na Web e APIs; Importância da Criação de Funções; Parâmetros de Função.

Introdução:

A evolução da tecnologia da informação revolucionou a maneira como interagimos com a web e com sistemas de software. Nesse contexto, conceitos como sites dinâmicos, requisições na web, APIs, formatos de dados (JSON e XML) e a criação de funções desempenham um papel crucial. Este artigo tem como objetivo explorar esses conceitos de forma acessível, aprofundando-se em suas aplicações práticas e importância.

O Que É um Site Dinâmico?

Um site dinâmico é um tipo de site que oferece conteúdo interativo e personalizado, em contraste com sites estáticos que apresentam informações fixas e imutáveis. A principal característica dos sites dinâmicos é a capacidade de responder às ações do usuário, atualizar o conteúdo em tempo real e se adaptar de acordo com as preferências individuais. Essa dinamicidade é possibilitada por meio de linguagens de programação, bancos de dados e tecnologias de servidor.

Exemplo 1:

Pense em um site de notícias. Em vez de atualizar a página o tempo todo para ver as últimas notícias, um site dinâmico carrega novos artigos automaticamente à medida que são publicados. Isso economiza tempo e oferece conteúdo atualizado instantaneamente.

Exemplo 2:

Imagine um site de streaming de música, como o Spotify. À medida que você ouve músicas, o site cria automaticamente listas de reprodução personalizadas com base nas suas preferências musicais. Isso é possível graças à dinamicidade do site, que se adapta às suas escolhas.

Exemplo 3:

Em um site de comércio eletrônico, como o eBay, os leilões em andamento são atualizados em tempo real. Quando alguém faz uma oferta, a página é automaticamente atualizada para mostrar a oferta mais alta. Isso cria uma experiência dinâmica para os usuários.

A dinamicidade dos sites resolve problemas de personalização, usabilidade e atualização constante de conteúdo, proporcionando uma experiência mais envolvente aos usuários e facilitando a manutenção do site, já que as informações podem ser gerenciadas de forma centralizada em um banco de dados.

Requisições na Web e APIs:

Requisições na web são solicitações que um cliente, geralmente um navegador da web, envia a um servidor para obter informações ou realizar ações em um site. Isso pode incluir solicitar uma página da web, enviar dados de um formulário, obter informações de um banco de dados ou buscar dados de um serviço externo.

Exemplo 1 - Busca no Google:

Quando você digita uma pergunta no Google, como "Qual é a capital do Brasil?", o Google envia sua pergunta para seus servidores. Eles procuram a resposta em seus registros gigantes e, em seguida, mostram a resposta para você na página de resultados. É como fazer uma pergunta a um professor e receber uma resposta na lousa.

Exemplo 2 - Compra em uma Loja Online:

Ao comprar algo em uma loja online, você adiciona itens ao seu carrinho e clica em "Finalizar Compra". Isso envia uma requisição para o servidor da loja. Eles pegam os itens que você quer, calculam o preço total e pedem para você confirmar o pagamento. É como fazer compras em uma loja física, mas no mundo virtual.

Esses exemplos mostram como a web funciona de maneira simples e formal. Você faz perguntas, faz pedidos e envia informações, e os servidores da web respondem de acordo. É como uma conversa educada entre você e a internet.

Esses são exemplos simples de como as coisas acontecem na web quando você visita sites e interage com eles. É como se você estivesse fazendo pedidos educados para obter informações ou realizar ações nos sites que visita.

APIs (Interfaces de Programação de Aplicativos):

Elas são conjuntos de regras e protocolos que permitem que diferentes aplicativos ou sistemas se comuniquem e interajam entre si. As APIs possibilitam que desenvolvedores acessem recursos e serviços de um aplicativo ou plataforma, como obter dados de uma rede social, acessar informações de localização em um aplicativo de mapeamento ou enviar mensagens de um aplicativo de mensagens.

Exemplo 1:

Quando você pesquisa algo no Google, seu navegador envia uma requisição para os servidores do Google. Os servidores processam sua pesquisa e retornam os resultados, que são exibidos na página. Isso é feito por meio de requisições na web.

Exemplo 2:

Aplicativos de previsão do tempo usam APIs para obter dados atualizados. Eles fazem requisições para serviços de meteorologia, como o Weather.com, e recebem informações sobre as condições climáticas, temperaturas e previsões.

Exemplo 3:

Redes sociais como o Facebook usam APIs para permitir que outros aplicativos compartilhem conteúdo em seus feeds. Por exemplo, quando você vê um vídeo do YouTube incorporado em uma postagem do Facebook, isso acontece por meio de uma API.s.

JSON e XML:

JSON (Notação de Objetos JavaScript) e XML (Linguagem de Marcação Extensível) são formatos de dados que permitem a representação estruturada e a troca de informações entre sistemas. Ambos são amplamente utilizados em comunicações web e em serviços de API.

JSON é um formato de dados leve e legível por humanos que utiliza uma estrutura de pares chave-valor para representar dados. É amplamente suportado por diversas linguagens de programação e é frequentemente utilizado em serviços da web.

Exemplos de JSON:

Exemplo 1 - Detalhes de um livro:

```
{
  "titulo": "O Senhor dos Anéis",
  "autor": "J.R.R. Tolkien",
  "publicacao": 1954,
  "genero": ["Fantasia", "Aventura"],
  "editora": "HarperCollins",
  "paginas": 1178,
  "preco": 45.99,
  "disponivel": true,
  "avaliacoes": [
    {"usuario": "Ana", "classificacao": 5},
    {"usuario": "Carlos", "classificacao": 4},
    {"usuario": "Marta", "classificacao": 5}
  ]
}
```

Exemplo 2 - Dados de um Produto:

```
{
  "nome": "Smartphone Modelo X",
  "marca": "MarcaTech",
  "preco": 799.99,
  "estoque": 125,
  "especificacoes": {
    "tela": "6.2 polegadas",
    "camera": "12 MP",
    "armazenamento": "64 GB"
  },
  "disponivel": true,
  "avaliacoes": [
    {"usuario": "João", "classificacao": 4.5},
    {"usuario": "Ana", "classificacao": 5},
    {"usuario": "Carlos", "classificacao": 4}
  ]
}
```

XML é mais complicado, cheio de marcas e lembra um pouco o HTML

Exemplo de XML 1 - Configuração de Aplicativo:

```
<configuracao>
  <idioma>Português</idioma>
  <tema>Claro</tema>
  <notificacoes>
    <email>true</email>
    <sms>false</sms>
  </notificacoes>
</configuracao>
```

Exemplo de XML 2 - Documento de Livro:

```
<livro>
  <titulo>O Senhor dos Anéis</titulo>
  <autor>J.R.R. Tolkien</autor>
  <genero>Fantasia</genero>
  <ano_publicacao>1954</ano_publicacao>
  <editora>HarperCollins</editora>
</livro>
```

Exemplo de XML 3 - Dados de Pedido de Compras:

```
<pedido>
  <numero_pedido>12345</numero_pedido>
  <data_pedido>2023-10-15</data_pedido>
  <itens>
    <item>
      <produto>Computador</produto>
      <quantidade>2</quantidade>
      <preco_unitario>800.00</preco_unitario>
    </item>
    <item>
      <produto>Monitor</produto>
      <quantidade>3</quantidade>
      <preco_unitario>250.00</preco_unitario>
    </item>
  </itens>
  <total>3650.00</total>
</pedido>
```

A Importância da Criação de Funções e Parâmetros:

Funções são como pequenos programas dentro do programa maior. Elas tornam o código mais organizado e fácil de gerenciar, as funções aceitam parâmetros, que são como regras que você dá à função. Você diz o que precisa fazer e fornece informações para fazer o trabalho.

Exemplo 1 - Função de Verificação de Número Primo em PHP:

```
<?php
function verificar_numero_primo($numero) {
    if ($numero <= 1) {
        return false;
    }
    for ($i = 2; $i < $numero; $i++) {
        if ($numero % $i == 0) {
            return false;
        }
    }
    return true;
}

// Exemplo de uso da função
$numero = 17;
if (verificar_numero_primo($numero)) {
    echo $numero . " é um número primo.";
} else {
    echo $numero . " não é um número primo.";
}
?>
```

Exemplo 2 - Função de Conversão de Moeda em PHP:

```
<?php
function converter_moeda($valor, $taxa_cambio, $moeda_destino) {
    $valor_convertido = $valor * $taxa_cambio;
    return "$valor $moeda_destino equivalem a $valor_convertido reais.";
}

// Exemplo de uso da função
$valor_em_dolar = 100;
$taxa_de_cambio = 5.50;
$moeda_destino = "BRL";
$resultado = converter_moeda($valor_em_dolar, $taxa_de_cambio, $moeda_destino);
echo $resultado;
?>
```

Exemplo 3 - Função de Ordenação de Lista em PHP:

```
<?php
function ordenar_lista($lista) {
    sort($lista);
    return $lista;
}

// Exemplo de uso da função
$numeros = array(32, 18, 5, 47, 12, 70);
$lista_ordenada = ordenar_lista($numeros);
echo "Lista ordenada: " . implode(', ', $lista_ordenada);
?>
```

Conclusão:

A internet é uma parte enorme de nossas vidas hoje em dia, e entender como ela funciona é como saber como dirigir um carro. Sites dinâmicos, requisições na web, APIs, JSON, XML e funções são como as engrenagens desse carro. Saber como tudo isso funciona pode nos ajudar a criar coisas incríveis e a entender melhor o mundo digital que nos cerca. Mesmo se você estiver apenas começando na faculdade, esses conceitos vão te acompanhar por toda a jornada de aprendizado.