

Programação para Web I

Aula - 15

O que é PHP?

- **PHP (Hypertext Preprocessor).**
 - Linguagem de programação executada no lado do servidor.
 - Projetada para desenvolvimento web.
 - É uma linguagem de código aberto e amplamente utilizada.

Características do PHP

- Interpretação em tempo real:
 - Executado no servidor antes de enviar ao cliente.
- Suporte a várias plataformas:
 - Funciona no Windows, Linux e macOS.
- Integração com bancos de dados:
 - Compatível com MySQL, PostgreSQL, SQLite, entre outros.
- Fácil de aprender:
 - Sintaxe semelhante ao C e Java.

Instalação do PHP

- **XAMPP**

- É uma distribuição gratuita e de código aberto que facilita a instalação de um ambiente de desenvolvimento web.
- Suportado pelos sistemas operacionais como Windows, macOS e Linux.
- O nome "XAMPP" é um acrônimo que representa os seguintes componentes:
 - **X**: Qualquer sistema operacional (cross-platform).
 - **A**: Apache (servidor web).
 - **M**: MySQL (ou MariaDB, que é uma alternativa ao MySQL, usado como sistema de gerenciamento de banco de dados).
 - **P**: PHP (linguagem de programação).
 - **P**: Perl (outra linguagem de programação, embora não seja tão usada quanto PHP).

Principais Características do XAMPP

- **Fácil Instalação:** oferece um instalador que simplifica o processo de configuração.
 - Permitindo que usuários de todos os níveis, até mesmo iniciantes, possam configurar um ambiente de desenvolvimento sem complicações.
- **Portabilidade:** pode ser instalado em diversas plataformas, tornando-o uma escolha popular entre desenvolvedores que trabalham em diferentes sistemas operacionais.
- **Interface Gráfica:** inclui um **painel de controle** que permite iniciar e parar os serviços (Apache, MySQL, etc.) de forma fácil.
- **Ambiente de Desenvolvimento:** ideal para testar e desenvolver aplicações PHP **localmente** antes de publicá-las em um servidor web.
- **Serviço para gerenciamento de bancos de dados MySQL:** phpMyAdmin.

Como Usar o XAMPP

- **Download e Instalação:** baixe o instalador do XAMPP do site oficial (<https://www.apachefriends.org/index.html>) e siga as instruções de instalação.
- **Iniciar o Painel de Controle:** após a instalação, abra o painel de controle do XAMPP para iniciar os serviços que você precisa (como Apache e MySQL).
- **Desenvolvimento:**
 - Coloque seus arquivos PHP na pasta "**htdocs**" (localizada na pasta de instalação do XAMPP).
 - Acesse-os através do navegador, utilizando a URL `http://localhost/nome_do_arquivo.php`.

Estrutura Básica de um Script PHP

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Título da página</title>
</head>

<body>

    <?php
        echo "Olá, Mundo!";
    ?>

</body>
</html>
```

Comentários em PHP

- `//`
 - Para comentários de uma linha.
- `/* */`
 - Para comentários de múltiplas linhas.

Variáveis em PHP

- Uma variável em PHP é representada por um sinal de cifrão (\$) seguido pelo nome da variável.
- As variáveis são case-sensitive (sensíveis a maiúsculas e minúsculas).
- Exemplo de declaração de variável:
 - `$nome = "Alex";`
 - `$idade = 25;`

Tipos de Dados em PHP

- **Inteiros:** Números inteiros, positivos ou negativos.
 - `$numero = 42;`
- **Float:** Números de ponto flutuante.
 - `$preco = 19.99;`
- **String:** Cadeias de texto.
 - `$texto = "Olá, Mundo!";`
- **Booleano:** Verdadeiro (true) ou falso (false).
 - `$ativo = true;`
- **NULL:** Representa uma variável sem valor.
 - `$variavelNula = null;`

Conversão de Tipos em PHP

- Conversão de tipos em PHP refere-se à mudança do tipo de uma variável para outro tipo.
- PHP realiza a conversão automática em muitas situações, mas também permite conversões manuais.

Conversão Implícita

- Ocorre automaticamente quando o PHP precisa mudar o tipo de uma variável para realizar uma operação.

```
$numero = "10"; // String
```

```
$soma = $numero + 5; // Conversão implícita para inteiro
```

```
echo $soma; // Resultado: 15
```

Conversão Explícita

- Feita manualmente pelo programador usando casting.
 - Sintaxe:
 - (tipo) variável;
- Exemplos de conversão explícita:
 - Para inteiro:

```
$numero = "42";  
$inteiro = (int)$numero; // Agora $inteiro é do tipo inteiro
```
 - Para float:

```
$stringFloat = "3.14";  
$float = (float)$stringFloat; // Agora $float é do tipo float
```

Conversão Explícita

- Outros exemplos:

- De Inteiro para String

`$numero = 100;`

`$string = (string)$numero; // $string = "100"`

- De Float para Inteiro

`$float = 10.99;`

`$inteiro = (int)$float; // $inteiro = 10`

Funções de Conversão

- PHP também oferece funções para conversão de tipos:

- intval() : Converte para inteiro.

```
$valor = "123.45";
```

```
$inteiro = intval($valor); // $inteiro = 123
```

- floatval() : Converte para float.

```
$valor = "123.45";
```

```
$float = floatval($valor); // $float = 123.45
```

- strval() : Converte para string.

```
$numero = 100;
```

```
$string = strval($numero); // $string = "100"
```

Estrutura if

- A estrutura if avalia uma condição e executa um bloco de código se a condição for verdadeira.

```
$idade = 18;
```

```
if ($idade >= 18) {  
    echo "Você é maior de idade."  
}
```


Estrutura if else

- O else é usado para executar um bloco de código se a condição do if não for verdadeira.

```
$idade = 16;
```

```
if ($idade >= 18) {  
    echo "Você é maior de idade.";  
} else {  
    echo "Você é menor de idade.";  
}
```

Estrutura elseif

- O elseif permite testar múltiplas condições.

```
$nota = 75;
```

```
if ($nota >= 90) {  
    echo "Aprovado com Distinção.";  
} elseif ($nota >= 70) {  
    echo "Aprovado.";  
} else {  
    echo "Reprovado.";  
}
```

Operadores Lógicos

- AND (&& ou and)
 - Retorna true se ambas as expressões forem verdadeiras.
- OR (|| ou or)
 - Retorna true se pelo menos uma das expressões for verdadeira.
- NOT (!)
 - Retorna o valor oposto da expressão.
 - Se a expressão for true, retorna false, e vice-versa.
- XOR (xor)
 - Retorna true se apenas uma das expressões for verdadeira, mas não ambas.

Estrutura switch

- O switch é utilizado para testar uma variável contra múltiplos valores.

```
$fruta = "maçã";
```

```
switch ($fruta) {  
    case "laranja":  
        echo "Você escolheu uma laranja.";  
        break;  
    case "maçã":  
        echo "Você escolheu uma maçã.";  
        break;  
    default:  
        echo "Fruta não reconhecida.";  
}
```

Estrutura for

- O **for** é usado para executar um bloco de código um número específico de vezes.

```
for ($i = 0; $i < 5; $i++) {  
    echo "Número: $i<br>";  
}
```

Estrutura while

- O **while** executa um bloco de código enquanto a condição for verdadeira.

```
$i = 0;
```

```
while ($i < 5) {  
    echo "Número: $i<br>";  
    $i++;  
}
```

Estrutura do-while

- O **do-while** garante que o bloco de código seja executado pelo menos uma vez.

```
$i = 0;
```

```
do {  
    echo "Número: $i<br>";  
    $i++;  
} while ($i < 5);
```

Array

- Estruturas de dados que armazenam múltiplos valores.

```
$frutas = array("maçã", "banana", "laranja");  
echo $frutas[0]; // Exibe "maçã"
```


Estrutura foreach

- O foreach é usado para iterar sobre arrays.

```
$frutas = array("maçã", "banana", "laranja");
```

```
foreach ($frutas as $fruta) {  
    echo "Fruta: $fruta<br>";  
}
```

Arrays Associativos

- São arrays onde as chaves são nomes atribuídos aos elementos, em vez de índices numéricos.
- Essas chaves são geralmente strings
 - Facilitam a associação entre os valores e suas respectivas chaves.

```
$aluno = array("nome" => "João", "idade" => 20, "curso"  
=> "Engenharia Civil");  
echo $aluno["curso"]; // Exibe "Engenharia Civil"
```

Arrays Associativos

```
$aluno = array(  
    "nome" => "Alex",  
    "idade" => 22,  
    "curso" => "Sistemas de Informação",  
    "cidade" => "Monte Carmelo"  
);
```

```
// Exibindo as informações do aluno  
foreach ($aluno as $chave => $valor) {  
    echo "$chave: $valor <br>";  
}
```

Arrays Associativos

```
$alunos = array(
    array("nome" => "Alex", "idade" => 22, "curso" =>
        "Sistemas de Informação", "cidade" => "Monte
        Carmelo"),

    array("nome" => "Maria", "idade" => 20, "curso" =>
        "Ciência da Computação", "cidade" =>
        "Uberlândia")
);
// Exibindo as informações do aluno
foreach ($alunos as $aluno) {
    echo "Nome: " . $aluno['nome'] . ", Idade: " . $aluno['idade'] . ",
        Curso: " . $aluno['curso'] . ", Cidade: " . $aluno['cidade']
        . "\n";
}
```

Funções

- Definição de Funções:

```
function soma($a, $b) {  
    return $a + $b;  
}
```

Classe e objetos

- Uma **classe** é um modelo para criar objetos.
 - Ela pode conter propriedades (atributos) e métodos (funções).

```
class Carro {  
    public $cor;  
    public $modelo;  
  
    public function mostrarDetalhes() {  
        return "Modelo: $this->modelo, Cor: $this->cor";  
    }  
}
```

Classe e objetos

- Para criar um **objeto**, utiliza-se a palavra-chave **new**.

```
$meuCarro = new Carro();
```

```
$meuCarro->cor = "vermelho";
```

```
$meuCarro->modelo = "Fusca";
```

```
echo $meuCarro->mostrarDetalhes();
```

```
// Saída: Modelo: Fusca, Cor: vermelho
```

Classe e objetos

- Construtores: é um método especial que é chamado automaticamente ao criar um objeto. Ele é definido com o nome `__construct()`.

```
class Carro {  
    public $cor;  
    public $modelo;  
  
    public function __construct($cor, $modelo) {  
        $this->cor = $cor;  
        $this->modelo = $modelo;  
    }  
  
    public function mostrarDetalhes() {  
        return "Modelo: $this->modelo, Cor: $this->cor";  
    }  
}
```

```
$meuCarro = new Carro("azul", "Civic");  
echo $meuCarro->mostrarDetalhes(); // Saída: Modelo: Civic, Cor: azul
```


Classe e objetos

- Métodos definem comportamentos.

```
class Carro {  
    public $cor;  
    public $modelo;  
  
    public function ligar() {  
        return "O carro $this->modelo está ligado.";  
    }  
}
```

```
$meuCarro = new Carro();  
$meuCarro->modelo = "Fusca";  
echo $meuCarro->ligar(); // Saída: O carro Fusca está  
ligado.
```

Exemplo 01 - PHP com HTML

```
<html>
<body>
  <?php
    $idade = 20;

    if ($idade >= 18) {
      echo '<p>Você é maior de idade.</p>';
    } else {
      echo '<p>Você é menor de idade.</p>';
    }
  ?>
</body>
</html>
```

Exemplo 02 - PHP com HTML

```
<html>
<body>
  <h3>Lista de Frutas:</h3>
  <ul>
    <?php
      $frutas = ["Maçã", "Banana", "Laranja", "Uva"];

      foreach ($frutas as $fruta) {
        echo '<li>' . $fruta . '</li>';
      }

    ?>
  </ul>
</body>
</html>
```

Exercícios

1) Desenvolva um script em PHP que exiba uma lista de notas e informe se cada aluno foi **Aprovado** ou **Reprovado**. As notas dos alunos são fornecidas em um array. O critério para aprovação é ter uma nota maior ou igual a 60. A saída deve ser uma página HTML que apresente os resultados de cada aluno, informando a nota e o respectivo status de aprovação.

Notas: 35, 77, 65, 49, 28 e 95.

Exercícios

2) Usando PHP, gere uma lista HTML () que exiba os números de 1 até 30 inserido pelo usuário, utilizando o loop for.

3) Crie um array associativo em PHP que contenha informações de três produtos (nome, preço e quantidade). Em uma página HTML, exiba esses produtos em formato de tabela (<table>), usando PHP para iterar sobre o array.