



# Primeiros passos em PHP

PROGRAMAÇÃO WEB II

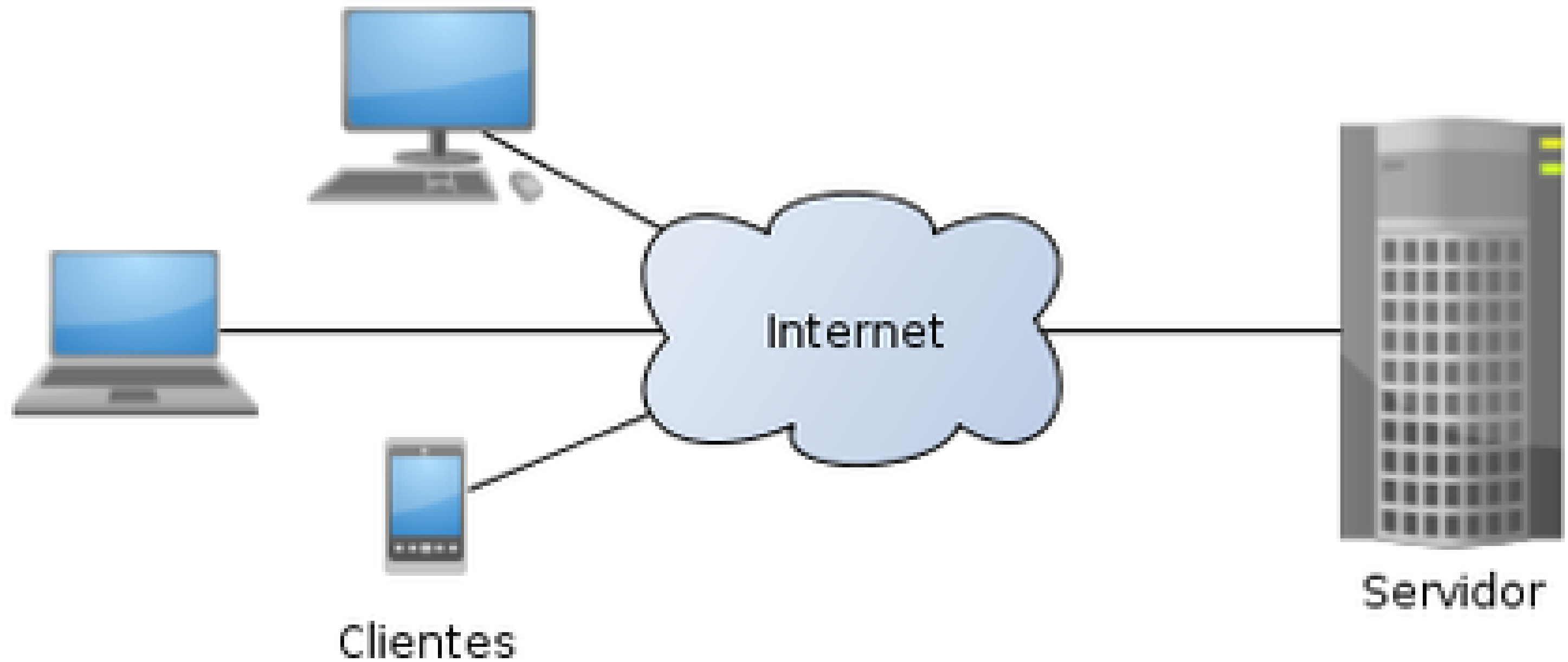
# O que é PHP?

- PHP significa "Hypertext Preprocessor". Originalmente, a sigla era para "Personal Home Page Tools", mas hoje é um acrônimo recursivo que se refere a um processador de hipertexto. É uma linguagem de script de uso geral, especialmente adequada para desenvolvimento web e que pode ser incorporada ao HTML.
- É uma linguagem de script server-side (executada no servidor).
- Muito utilizada para criação de sites dinâmicos e sistemas web.
- Código PHP é embutido em HTML.

# Arquitetura Cliente-Servidor

- É um modelo que divide as aplicações em dois componentes principais: o **cliente** e o **servidor**.
- Essa separação de responsabilidades melhora a modularidade e a manutenção do sistema.
- **Cliente:** É a aplicação que o usuário interage diretamente. Ele é responsável por apresentar a interface do usuário e coletar entradas para enviar ao servidor. Em uma aplicação web, o navegador web atua como cliente, enquanto em uma aplicação desktop, o próprio software instalado no computador do usuário é o cliente.
- **Servidor:** é a aplicação que recebe e processa as solicitações do cliente. Ele gerencia os dados, executa a lógica de negócios e retorna as respostas apropriadas ao cliente. Servidores podem ser configurados para lidar com múltiplas solicitações simultaneamente, garantindo eficiência e escalabilidade.

# Arquitetura Cliente-Servidor



# Como funciona o PHP

- O PHP é uma linguagem de script do lado do servidor, amplamente utilizada para desenvolvimento web, permitindo a criação de páginas dinâmicas e interativas. Funciona processando código PHP no servidor e retornando HTML para o navegador do usuário.

## Como funciona o PHP em detalhes:

- 1.O cliente acessa uma URL (por exemplo: `http://localhost/index.php`) em um navegador.
- 2.O navegador envia essa requisição HTTP para o servidor web (como o Apache, que vem no XAMPP).
- 3.O servidor Apache identifica que é um arquivo .php e chama o interpretador PHP.
- 4.O PHP executa o script no servidor (por exemplo, calcula valores, busca dados, etc.).
- 5.O resultado da execução é transformado em HTML puro e enviado de volta para o navegador.
- 6.O navegador exibe o conteúdo para o usuário.

## Importante!

O código PHP nunca é visto pelo cliente, apenas o resultado final (geralmente HTML).

Isso garante segurança, pois a lógica do sistema fica no servidor.

# Como funciona o PHP



# Aplicações em PHP

- Sistemas de gestão (ERP, CRM)
- Blogs e CMS (WordPress, Joomla)
- Lojas virtuais (Magento, WooCommerce)
- Sistemas educacionais, fóruns, intranets



# Diferenças entre HTML x JS x PHP

## Comparação das Linguagens na Programação Web

Linguagem	Tipo	Onde é executada	Função principal	Visível ao usuário?
HTML	Linguagem de marcação	Navegador (cliente)	Estrutura estática da página (conteúdo)	Sim
PHP	Linguagem de programação	Servidor	Geração dinâmica de conteúdo	Não
JavaScript	Linguagem de programação	Navegador (cliente)	Interações e comportamentos dinâmicos	Sim



# Ambiente de Desenvolvimento

- XAMPP é um pacote gratuito que reúne os principais softwares necessários para desenvolver aplicações web localmente (sem precisar de internet ou servidor real).
- É muito utilizado para testar páginas PHP em um ambiente local, antes de publicá-las em um servidor online.

**X:** Cross-platform (multiplataforma: Windows, Linux, macOS)

**A:** Apache (servidor web)

**M:** MySQL ou MariaDB (banco de dados)

**P:** PHP (linguagem de programação)

**P:** Perl (outra linguagem de script, opcional)



- Permite criar um servidor local no seu computador.
- Você pode executar e testar arquivos PHP sem precisar de hospedagem paga.
- Ideal para estudantes e desenvolvedores iniciantes.


# Ambiente de Desenvolvimento


## Principais componentes que usaremos:


- Apache: interpreta os arquivos .php e serve as páginas no navegador.
- PHP: executa os scripts PHP.
- MySQL/MariaDB: banco de dados para armazenar informações (iremos usar em aulas futuras).





# Ambiente de Desenvolvimento


 XAMPP Control Panel v3.2.4 [ Compiled: Jun 5th 2019 ]


 **XAMPP Control Panel v3.2.4**


 **Config**


 **Netstat**

 **Shell**

 **Explorer**

 **Services**

 **Help**

 **Quit**

Service	Module	PID(s)	Port(s)	Actions			
<input type="checkbox"/>	Apache	16652 20364	80, 443	Stop	Admin	Config	Logs
<input type="checkbox"/>	MySQL	19916	3307	Stop	Admin	Config	Logs
<input type="checkbox"/>	FileZilla			Start	Admin	Config	Logs
<input type="checkbox"/>	Mercury			Start	Admin	Config	Logs
<input type="checkbox"/>	Tomcat			Start	Admin	Config	Logs

22:24:26 [main]

Starting Check-Timer

22:24:26 [main]

Control Panel Ready

22:24:26 [Apache]

Autostart active: starting...

22:24:26 [Apache]

Attempting to start Apache app...

22:24:26 [mysql]

Autostart active: starting...

22:24:26 [mysql]

Attempting to start MySQL app...

22:24:32 [Apache]

Status change detected: running

22:24:32 [mysql]

Status change detected: running

# Delimitadores de Código

O código deve estar contido entre os delimitadores:

**<?php**

echo "Olá, mundo!";

**?>**

# Comentários

De linha:

```
<?php  
// echo "a";  
# echo "a";  
?>
```

De bloco:

```
<?php  
/* echo "a";  
echo "b"; */  
?>
```

# Comandos de Saída

**echo – Exibe uma ou mais Strings**

**<?php**

**echo 'a', 'b', 'c';**

**echo ('a');**

**?>**

Sem parênteses: pode ter vários parâmetros

Com parênteses: somente um parâmetro

# Variáveis

- Sempre preenchidas de \$
- Tipagem dinâmica: não precisa declarar o tipo.

```
<?php
    $nome = "João";
    $sobrenome = "da Silva";
    echo 'Sobrenome: $sobrenome - Nome: $nome<br/>';
    echo 'Sobrenome: ' . $sobrenome . ' - Nome: ' . $nome . '<br/>';
    echo "Sobrenome: $sobrenome - Nome: $nome<br/>";
    echo "Nome Completo: $nome $sobrenome<br/>";
?>
```

# Tipos de Dados e Operadores

## Tipos de Dados:

- String: texto entre aspas "texto"
- Integer: números inteiros 10
- Float: números decimais 10.5
- Boolean: true ou false
- Array, Object, etc.

## Operadores:

- Aritméticos: +, -, \*, /, %
- Comparação: ==, !=, <, >, <=, >=
- Lógicos: &&, ||, !



# Variáveis

Sempre preenchidas de \$

```
<?php
    $a = 10;
    $A = 20;
    echo "O valor de \$a é $a e o de \$A é $A";
?>
```

# Primeiros passos em PHP

```
<html>
<head>
  <title>ETEC ZONA LESTE</title>
</head>
<body>
  <?php
    $base = 10;
    $altura = 20;
    $area = $base * $altura;
    echo "A área do retângulo é: " . $area;
  ?>
</body>
</html>
```

# Tipos Numéricos – Operações

```
<?php
$x = 10;
echo ($x + 5);
echo "<br/>";
echo ($x - 5);
echo "<br/>";
echo ($x * 5);
echo "<br/>";
echo ($x / 5);
echo "<br/>";
echo ($x % 8);
?>
```

# Operadores relacionais

<, >, <=, >=, ==, !=, ===

=== (idêntico a): Verdadeiro se os operandos forem iguais e do mesmo tipo.

## Exemplo:

```
$a = 5;    // inteiro
```

```
$b = "5";  // string
```

```
var_dump($a == $b); // true (valor igual, tipo diferente é ignorado)
```

```
var_dump($a === $b); // false (tipo diferente)
```

## var\_dump

- var\_dump() é uma função que exibe informações detalhadas sobre uma variável.
- Mostra o tipo da variável e seu valor.
- Muito útil para debugging (ajudar a encontrar erros).

# Operadores relacionais

var\_dump

- var\_dump() é uma função que exibe informações detalhadas sobre uma variável.
- Mostra o tipo da variável e seu valor.
- Muito útil para debugging (ajudar a encontrar erros).

Exemplo de uso:

```
$a = 10;
```

```
$b = "texto";
```

```
$c = [1, 2, 3];
```

```
var_dump($a);
```

```
/* Saída:
```

```
int(10)
```

```
*/
```

```
var_dump($b);
```

```
/* Saída:
```

```
string(5) "texto"
```

```
*/
```

```
var_dump($c);
```

```
/* Saída:
```

```
array(3) {
```

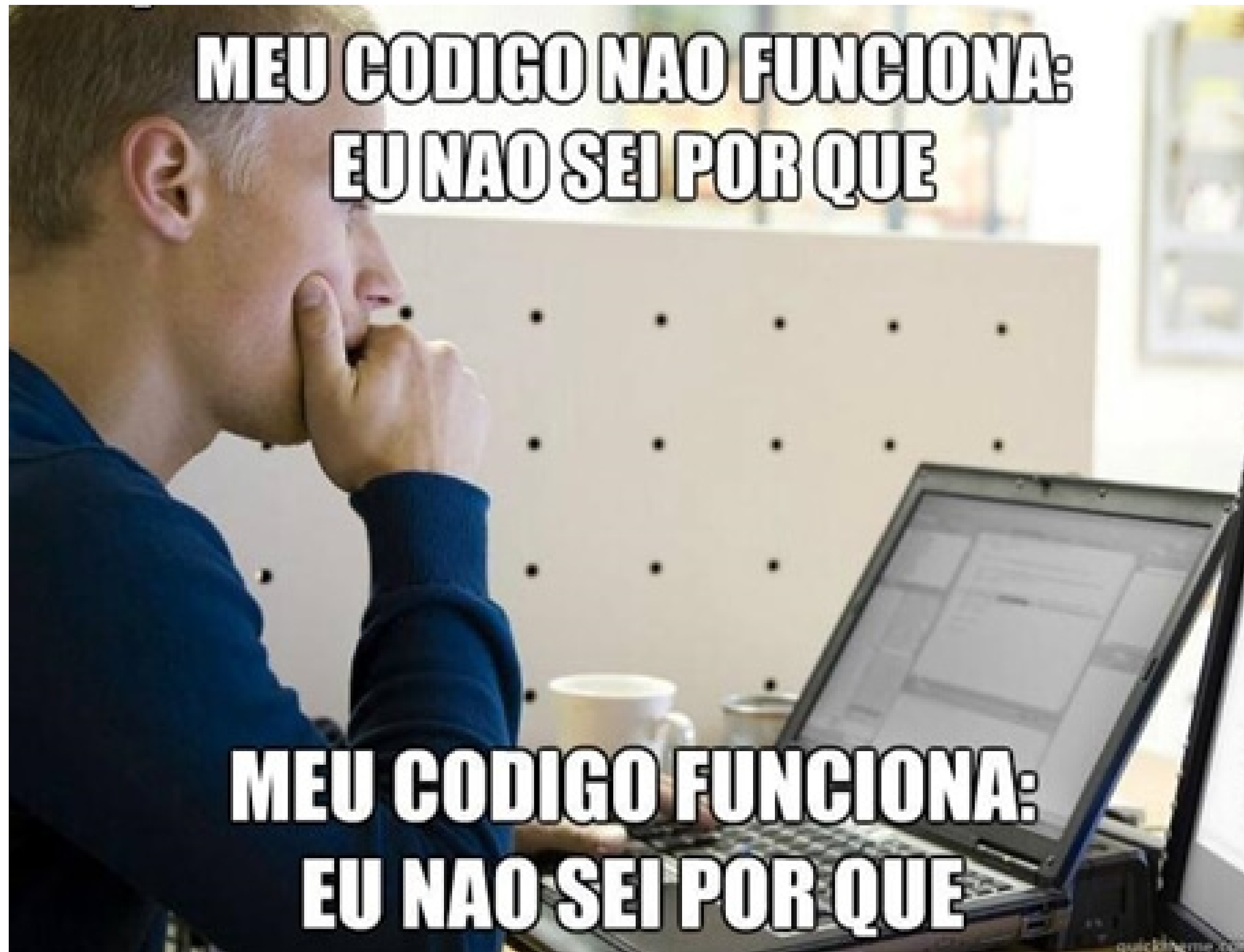
```
    [0]=> int(1)
```

```
    [1]=> int(2)
```

```
    [2]=> int(3)
```

```
}
```

# EXERCÍCIOS!



1. Crie um script que mostre seu nome e idade.
2. Declare duas variáveis com valores numéricos e mostre a soma.
3. Crie um script que concatene duas strings.
4. Mostre o uso de comentários de linha e múltiplas linhas.
5. Crie duas variáveis com valores numéricos e exiba: A soma/ subtração/ multiplicação/ divisão/ resto da divisão (módulo %)
6. Crie uma variável com um número inteiro e use o operador % para verificar se ele é par ou ímpar. Exiba o resultado.
7. Crie duas variáveis: uma com valor inteiro 10 e outra com valor string "10". Use var\_dump com == e === para mostrar a diferença entre os dois operadores.
8. Crie variáveis com diferentes tipos (string, int, float, boolean) e use var\_dump() para exibir o tipo e o valor de cada uma.
9. Crie uma variável que armazene uma temperatura em graus Celsius. Converta esse valor para Fahrenheit usando a fórmula:  **$F = (C \times 9/5) + 32$**
10. Crie um script PHP que armazene três notas em variáveis e calcule a média aritmética entre elas. Em seguida, exiba a média e uma mensagem de acordo com a seguinte regra: Se a média for maior ou igual a 7, exiba: "Aluno aprovado com média X". Caso contrário, exiba: "Aluno reprovado com média X"

**Obrigada!**

