
📘 Guia Completo de Estudos – PHP Revisão

🔹 Tema 1 – Conhecendo a linguagem PHP

Explicação: PHP é usado para criar páginas web dinâmicas. O código é processado no servidor.
Exemplo:

```
<?php
// Exibe uma mensagem na tela
echo "Olá, mundo!";
?>
```

Saída esperada:

```
Olá, mundo!
```

🔹 Tema 2 – Variáveis e Tipos de Dados

Explicação: Variáveis armazenam informações e começam com $. PHP detecta o tipo automaticamente.
Exemplo:

```
$nome = "Maria";   // String
$idade = 20;      // Inteiro
$altura = 1.65;   // Float
$ativo = true;    // Boolean

echo $nome;       // Exibe o nome
```

Saída esperada:

```
Maria
```

🔹 Tema 3 – Operadores

Explicação:

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

Exemplo:

```
echo 10 + 5;      // soma
echo 10 % 3;      // resto da divisão

var_dump(5 == "5");   // compara valor
var_dump(5 === "5");  // compara valor e tipo
```

Saída esperada:

15
1
bool(true)
bool(false)

➤ **Operadores Lógicos**

- **&& (E lógico)** → Verdadeiro só se **as duas condições forem verdadeiras**
- **|| (OU lógico)** → Verdadeiro se **pelo menos uma condição for verdadeira**
- **! (NÃO lógico)** → Inverte o valor lógico

Exemplo:

```
$idade = 20;
$temCarteira = true;

// E lógico (&&)
var_dump($idade >= 18 && $temCarteira);

// OU lógico (||)
var_dump($idade >= 18 || $temCarteira);

// NÃO lógico (!)
var_dump(!$temCarteira);
```

📌 **Saída esperada:**

bool(true) // maior de idade E tem carteira

bool(true) // maior de idade OU tem carteira

bool(false) // negação de true

🔹 **Tema 4 – Estruturas de Controle**

Explicação: Permitem executar blocos de código dependendo de condições ou repetir ações.

- **if/else**

```
$nota = 7;
if($nota >= 6){
  echo "Aprovado";
} else {
  echo "Reprovado";
}
```

Saída esperada:

Aprovado

- **switch**

```
$cor = "azul";
switch($cor){
  case "vermelho": echo "Pare"; break;
  case "verde": echo "Siga"; break;
  default: echo "Cor inválida";
}
```

Saída esperada:

Cor inválida

- **while**

```
$contador = 1;
while($contador <= 3){
  echo $contador . " ";
  $contador++;
}
```

Saída esperada:

1 2 3

- **for**

```
for($i=1; $i<=3; $i++){
  echo $i . " ";
}
```

Saída esperada:

1 2 3

🔹 **Tema 5 – Arrays**

Explicação: Arrays armazenam múltiplos valores (numéricos ou associativos).

- **Array numérico**

```
$frutas = array("Maçã", "Banana", "Laranja");
echo $frutas[1];
```

Saída esperada:

Banana

- **Array associativo**

```
$pessoa = array("nome"=>"Ana", "idade"=>25);
echo $pessoa["nome"];
```

Saída esperada:

Ana

🔹 **Tema 6 – Métodos GET e POST**

Explicação:

- **GET:** envia dados pela URL (não seguro para senhas).
- **POST:** envia dados no corpo da requisição (mais seguro).

Exemplo GET:

```
URL: pagina.php?nome=Joao
echo $_GET["nome"];
```

Saída esperada:

Joao

Exemplo POST:

```
// Recebendo dados de um formulário

echo $_POST["senha"];
```

Saída esperada: depende do que o usuário digitou no formulário (ex.: 12345).

🔹 **Tema 7 – Funções**

Explicação: Funções permitem **organizar código**, receber parâmetros e opcionalmente retornar valores.

- **Função sem retorno**

```
function saudacao(){
  echo "Bem-vindo!";
}
saudacao();
```

Saída esperada:

Bem-vindo!

- **Função com parâmetros e retorno**

```
function soma($a, $b){
  return $a + $b;
}
echo soma(5, 3);
```

Saída esperada:

8

🔹 Tema 8 – Cookies e Sessões

Explicação:

- **Cookies:** armazenam dados **no navegador** do usuário.
- **Sessões:** armazenam dados **no servidor**, mais seguras para informações sensíveis.
- **Cookie**

```
setcookie("usuario", "Maria", time()+3600); // 1 hora
echo $_COOKIE["usuario"];
```

Saída esperada:

Maria

- **Sessão**

```
session_start();
$_SESSION["nome"] = "João";
echo $_SESSION["nome"];
```

Saída esperada:

João

🔹 Tema 9 – Conexão com Banco de Dados (PDO)

Explicação: PDO conecta PHP a diversos bancos de dados com segurança e flexibilidade.

```
try {
  $pdo = new PDO("mysql:host=localhost;dbname=todo_app", "root", "");
  echo "Conectado!";
} catch (PDOException $e) {
  echo "Erro: " . $e->getMessage();
}
```

Saída esperada (se conexão OK):

Conectado!

- **Inserir dados**

```
$stmt = $pdo->prepare("INSERT INTO usuarios (nome, email) VALUES (?, ?)");
$stmt->execute(["Maria", "maria@email.com"]);
```

- **Buscar dados**

```
$stmt = $pdo->query("SELECT * FROM usuarios");
while($row = $stmt->fetch()){
  echo $row["nome"];
}
```

📌 **Dicas finais**

- Use echo para exibir valores.
- Variáveis começam com $.
- == compara valor, === compara valor e tipo.
- Laços: for (contador), while (condição).
- Arrays podem ser numéricos ou associativos.
- GET = URL, POST = corpo.
- Cookies → navegador, Sessões → servidor.
- session_start() obrigatório para sessões.
- PDO conecta PHP a MySQL com segurança.
