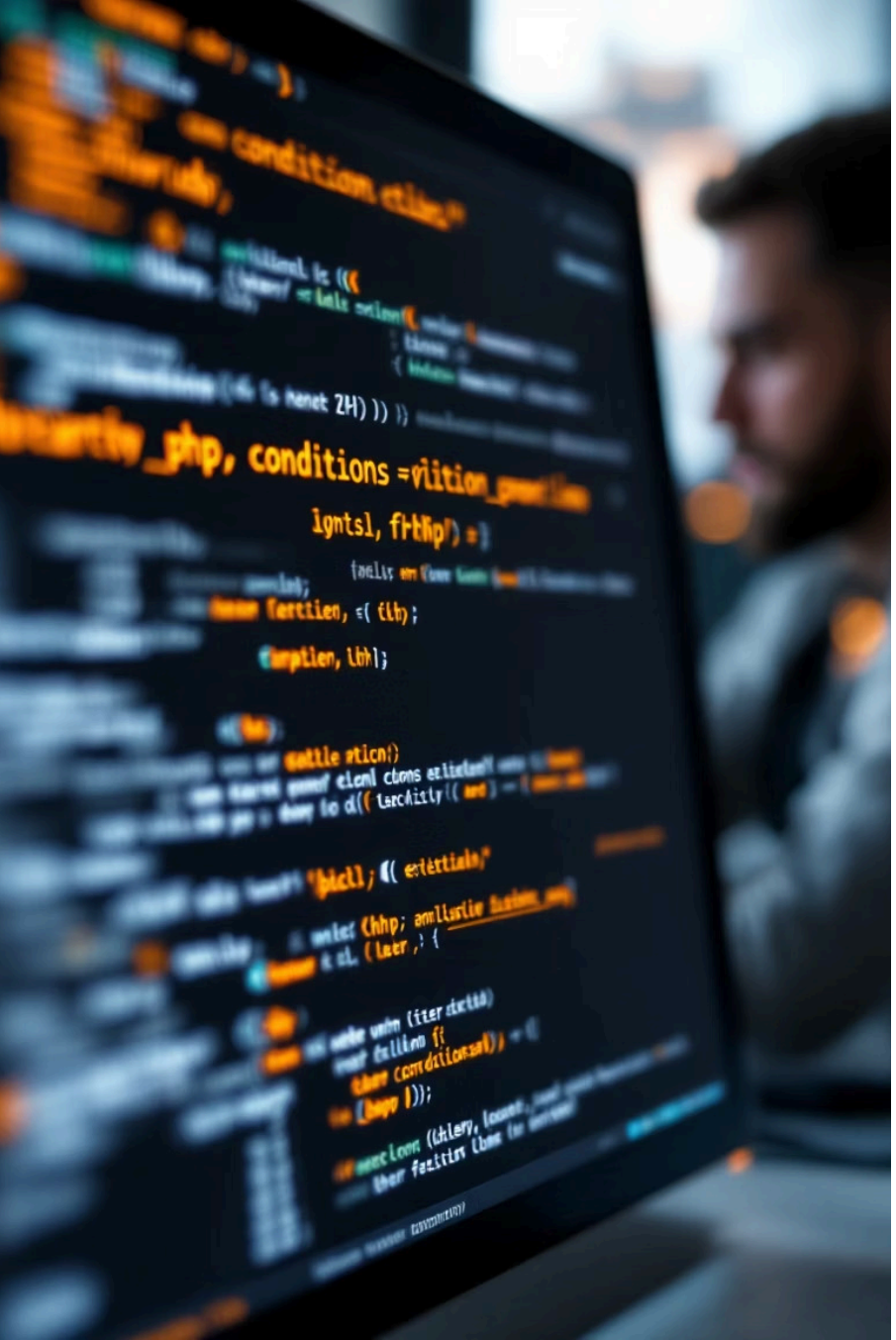


Estruturas Condicionais e de Repetição em PHP

Um guia completo para entender e aplicar as estruturas de controle de fluxo mais importantes na programação PHP.

Agenda

1 Estrutura Condicional if...else

Definição, sintaxe básica e exemplos práticos

2 Estrutura Condicional switch...case

Quando e como utilizar para comparações múltiplas

3 Estrutura de Repetição for

Loops com número definido de iterações

4 Comparativo e Boas Práticas

Quando usar cada estrutura e dicas de otimização

Estrutura Condicional if...else

Definição

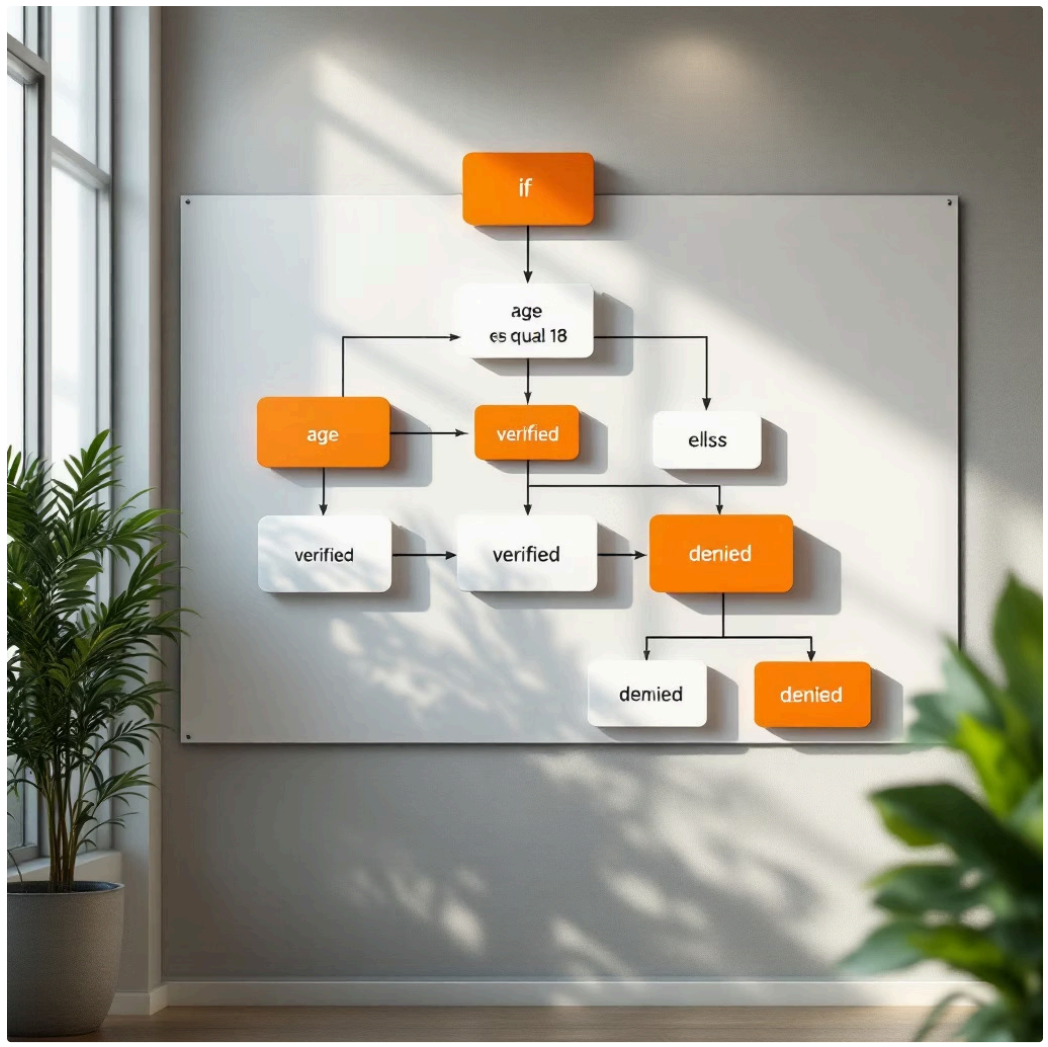
O if...else é usado para executar blocos de código diferentes dependendo de uma condição lógica.

Sintaxe básica

```
if (condição) {
    // Código executado se a condição for verdadeira
} else {
    // Código executado se a condição for falsa
}
```

Exemplo prático

```
$idade = 20;
if ($idade >= 18) {
    echo "Você é maior de idade.";
} else {
    echo "Você é menor de idade.";
}
```



Variantes do if...else

1

if...elseif...else

Para múltiplas condições:

```
$nota = 7;
if ($nota >= 9) {
    echo "Excelente";
} elseif ($nota >= 7) {
    echo "Bom";
} else {
    echo "Precisa melhorar";
}
```

2

If em uma linha

Para condições simples:

```
if ($idade >= 18) echo "Maior de idade";
```

3

Operador ternário

Versão curta do if/else:

```
echo ($idade >= 18) ? "Maior de idade" : "Menor de idade";
```

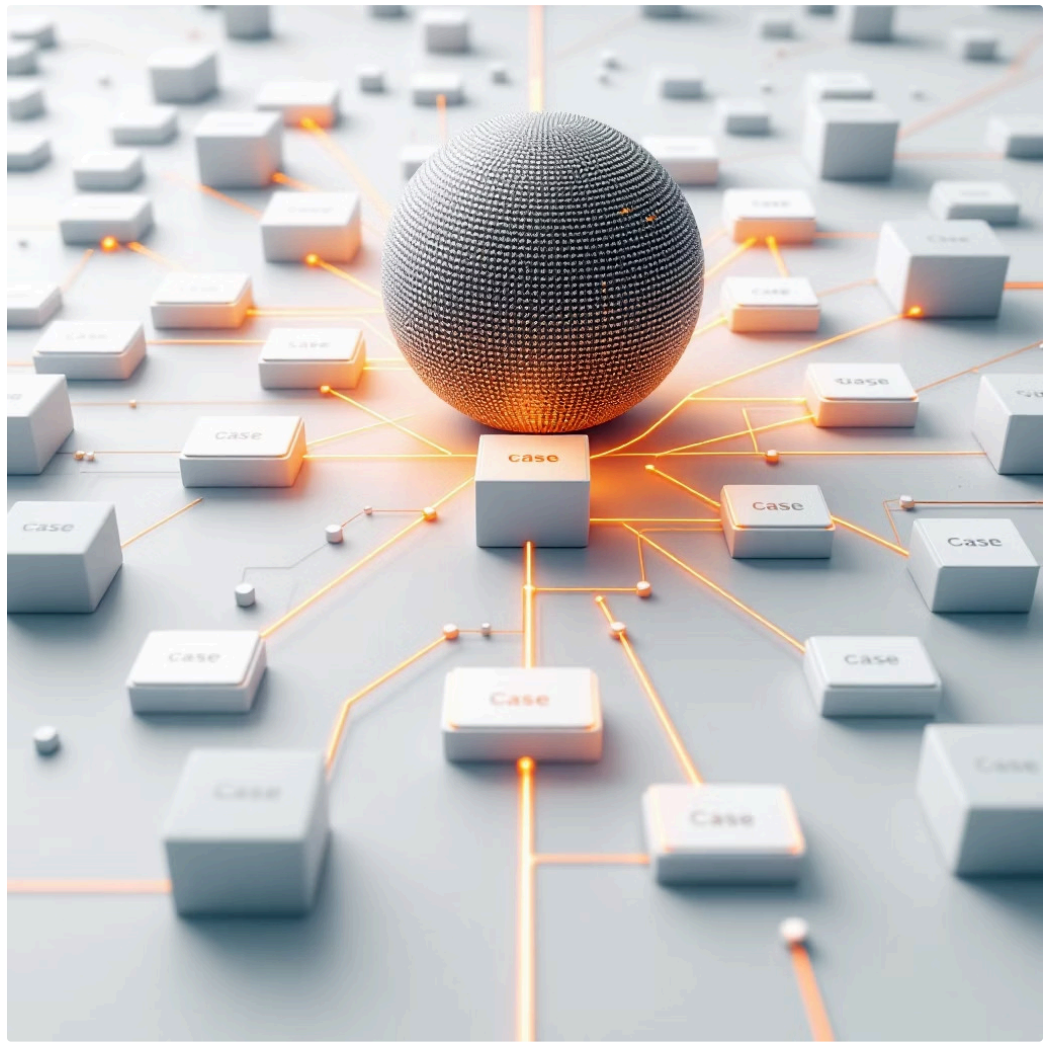
Estrutura Condicional switch...case

Definição

O switch é usado quando há necessidade de **comparar uma mesma variável** com **vários valores possíveis**, tornando o código mais organizado do que múltiplos elseif.

Sintaxe

```
switch (expressão) {
    case valor1:
        // Código executado se expressão == valor1
        break;
    case valor2:
        // Código executado se expressão == valor2
        break;
    default:
        // Código executado se nenhum caso for correspondente
}
```



Exemplo Prático do switch...case

Código de Exemplo

```
$dia = "quarta";
switch ($dia) {
    case "segunda":
        echo "Início da semana.";
        break;
    case "quarta":
        echo "Metade da semana.";
        break;
    case "sexta":
        echo "Último dia útil!";
        break;
    default:
        echo "Dia não reconhecido.";
}
```

Observações importantes

- **Sempre usar break** para evitar que o código continue executando os próximos case
- O default é opcional, mas recomendado
- Comparação é feita com **igualdade simples (==)**, não estrita

Estrutura de Repetição for

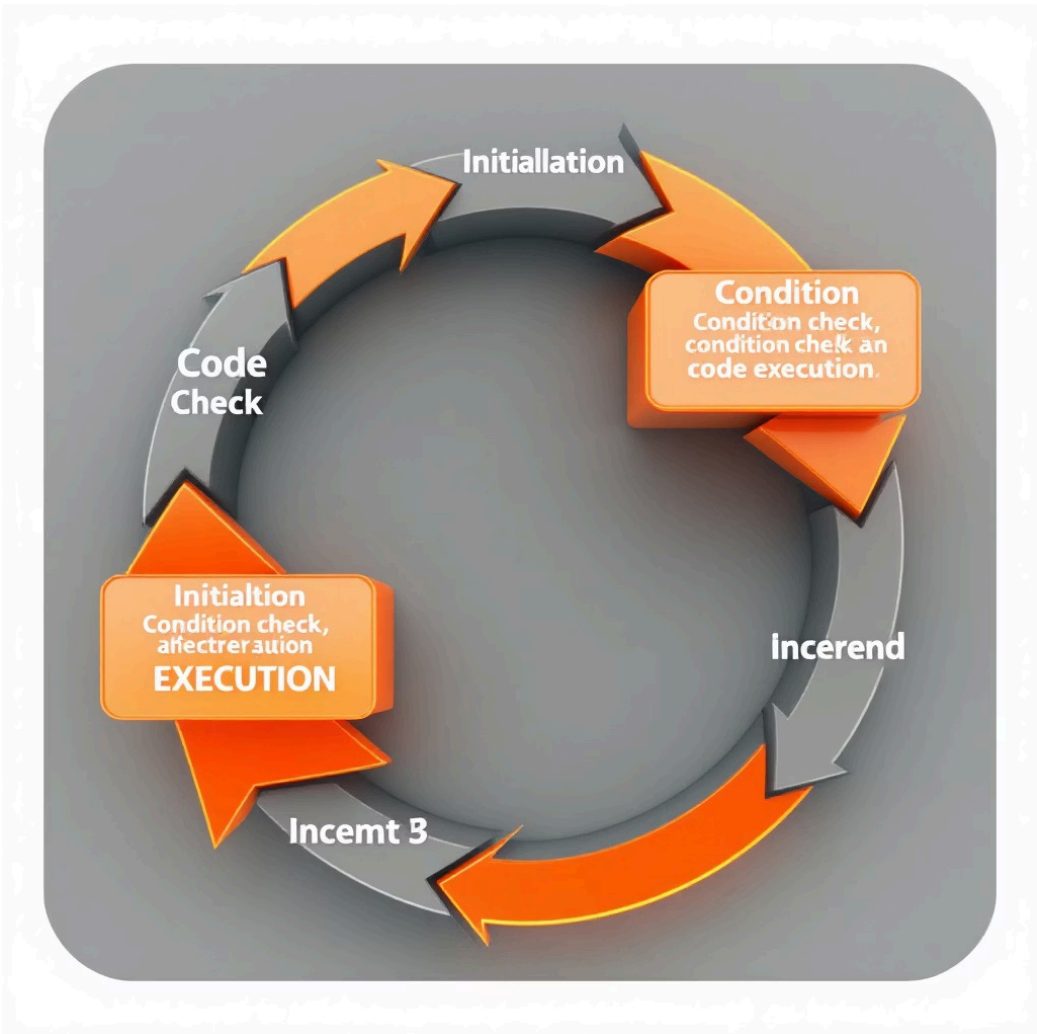
Definição

O for é usado quando se sabe **de antemão** o número de repetições que o código deve executar.

Sintaxe

```
for (inicialização; condição; incremento) {
    // Código executado a cada iteração
}
```

- **Inicialização:** executada apenas uma vez no início
- **Condição:** verificada a cada repetição; se for falsa, o loop para
- **Incremento/Decremento:** altera o valor da variável de controle a cada ciclo



Exemplos Práticos do for

Contagem crescente

```
for ($i = 1; $i <= 5; $i++) {
    echo "Número: $i
";
}
```

Resultado: Número: 1, Número: 2, Número: 3, Número: 4, Número: 5

Contagem regressiva

```
for ($i = 10; $i >= 1; $i--) {
    echo "Contagem regressiva: $i
";
}
```

Resultado: Contagem regressiva: 10, 9, 8... até 1

Percorrendo array

```
$nomes = ["Ana", "Bruno", "Carlos"];
for ($i = 0; $i < count($nomes); $i++) {
    echo $nomes[$i] . "
";
}
```

Resultado: Ana, Bruno, Carlos

Comparando as Estruturas

Estrutura	Uso principal	Melhor quando
if...else	Testar uma ou várias condições lógicas diferentes	Você precisa avaliar expressões booleanas complexas
switch	Comparar uma mesma variável com vários valores fixos	Uma única variável precisa ser comparada com múltiplos valores possíveis
for	Executar código repetidamente um número definido de vezes	Você sabe exatamente quantas iterações serão necessárias

Boas Práticas

Indentação

Indente o código para facilitar a leitura e manutenção.

Comparação estrita

Use === (comparação estrita) quando precisar verificar valor e tipo.

Otimização

No for, evite chamar funções pesadas no count() dentro da condição — armazene o valor antes.

Segurança

No switch, sempre utilize break para evitar execução indesejada de outros casos.

Seguir estas práticas tornará seu código mais legível, eficiente e menos propenso a erros.