

Operadores Relacionais

- ▶ Utilizados para realizar comparações entre valores ou expressões
- ▶ O resultado é sempre um valor booleano: `TRUE` ou `FALSE`

Operadores	Descrição
<code>==</code>	Igual: resulta verdadeiro (<code>TRUE</code>) se expressões forem iguais
<code>===</code>	Igual e do mesmo tipo de dados
<code>!=</code> ou <code><></code>	Diferente: resulta verdadeiro (<code>TRUE</code>) se as expressões forem diferentes
<code><</code>	Menor que
<code>></code>	Maior que
<code><=</code>	Menor ou igual que
<code>>=</code>	Maior ou igual que

Operadores

Relacionais - Exemplos

- Uso errado do operador de atribuição para realizar uma comparação:

```
<?php
    if($a = 5){
        echo "essa operação atribui 5 à variável a e retorna verdadeiro";
    }
?>
```

- Comparação de duas variáveis (inteiro e *string*):

```
<?php
    $a = 1234;
    $b = '1234';

    if($a == $b) {
        echo '$a e $b são iguais';
    }

    if($a != $b) {
        echo '$a e $b são diferentes';
    }
?>
```

Operadores

Relacionais - Exemplos (*continuação*)

- Declaração de duas variáveis (inteiro e *string*) e de seus tipos:

```
<?php
    $c = 1234;
    $d = '1234';

    if($c === $d) {
        echo '$c e $d são iguais e do mesmo tipo';
    }

    if($c !== $d) {
        echo '$c e $d são de tipos diferentes';
    }
?>
```

Operadores

Relacionais - Exemplos *(continuação)*

- O PHP considera o valor zero como falso em comparações lógicas

```
<?php
    $e = 0;

    //zero sempre é igual a FALSE
    if($e == FALSE) {
        echo '$e é FALSO';
    }

    //testa se $e é do tipo FALSE
    if($e === FALSE) {
        echo '$e é do tipo FALSE';
    }

    //testa se $e é igual e do mesmo tipo de zero
    if($e === 0) {
        echo '$e é zero mesmo';
    }
?>
```

Operadores Lógicos

- ▶ Utilizados para combinar expressões lógicas entre si
- ▶ O resultado é sempre um valor booleano: `TRUE` ou `FALSE`

Operadores	Descrição
<code>(\$a and \$b)</code> ou <code>(\$a && \$b)</code>	E: verdadeiro (<code>TRUE</code>) se tanto <code>\$a</code> quanto <code>\$b</code> forem verdadeiros
<code>(\$a or \$b)</code> ou <code>(\$a \$b)</code>	OU: verdadeiro (<code>TRUE</code>) se <code>\$a</code> ou <code>\$b</code> forem verdadeiros
<code>(\$a xor \$b)</code>	XOR: verdadeiro (<code>TRUE</code>) se <code>\$a</code> ou <code>\$b</code> forem verdadeiros, de forma exclusiva
<code>(! \$a)</code>	NOT: verdadeiro (<code>TRUE</code>) se <code>\$a</code> for <code>FALSE</code>

Operadores Lógicos - Exemplos

- Se `$vai_chover` e `$esta_frio` forem ambas verdadeiras, imprime "Não vou sair de casa"

```
<?php
    $vai_chover = TRUE;
    $esta_frio = TRUE;

    if($vai_chover and $esta_frio){
        echo "Não vou sair de casa";
    }
?>
```

- Se uma variável for `TRUE` e a outra `FALSE`, imprime "Vou pensar duas vezes antes de sair de casa"

```
<?php
    $vai_chover = FALSE;
    $esta_frio = TRUE;

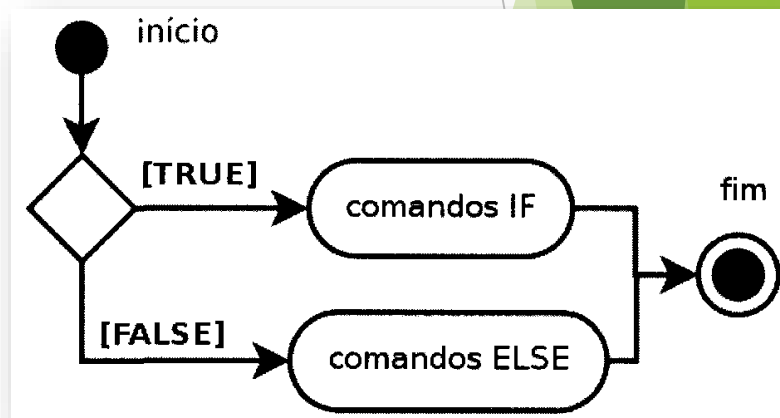
    if($vai_chover xor $esta_frio){
        echo "Vou pensar duas vezes antes de sair de casa";
    }
?>
```

Estruturas de controle

IF

- ▶ Realiza um desvio condicional na execução natural do programa
- ▶ Caso a condição seja satisfeita, serão executadas as instruções do bloco de comandos IF
- ▶ Caso a condição não seja satisfeita, o bloco é ignorado e as instruções do bloco ELSE são executadas (se houver)

```
if(expressão) {  
    comandos se expressão é verdadeira  
}else{  
    comandos se expressão é falsa  
}
```



Estruturas de controle

IF - Exemplos

- ▶ Quando não especificamos o operador lógico, o PHP retorna TRUE sempre que a variável tiver um conteúdo válido:

```
<?php
    $a = 'conteúdo';

    if($a){
        echo '$a tem conteúdo';
    }else{
        echo '$a não tem conteúdo';
    }

    if($b){
        echo '$b tem conteúdo';
    }
?>
```


Estruturas de controle

IF - Exemplos *(continuação)*

- Podemos utilizar um IF aninhados ou usar operadores lógicos:

```
<?php
    $salario = 1020;
    $tempo_servico = 12;
    $tem_reclamacoes = FALSE;

    if($salario>1000){
        if($tempo_servico >= 12){
            if($tem_reclamacoes != TRUE){
                echo 'Parabéns, você foi promovido!';
            }
        }
    }

    if(($salario > 1000) and ($tempo_servico >= 12) and
($tem_reclamacoes != TRUE)){
        echo 'Parabéns, você foi promovido!';
    }
?>
```

Estruturas de controle

IF - Exemplos *(continuação)*

- Atribuição condicional à uma variável

```
if($valor_venda > 100){  
    $resultado = 'muito caro';  
}else{  
    $resultado = 'pode comprar';  
}
```

- O mesmo código acima pode ser escrito utilizando o conceito de **Operador Ternário**

```
$resultado = ($valor_venda > 100) ? 'muito caro' : 'pode comprar';
```

- A sintaxe do **Operador Ternário** é a seguinte:

```
(<condição>) ? <instruções para verdadeiro> : <instruções para falso>
```

Exercícios 9, 10 e 11

- 9) Criar uma variável com valor numérico e informar se é divisível por 10, por 5, por 2 ou se não é divisível por nenhum destes

Salve seu arquivo como `ex9.php`

- 10) Criar duas variáveis numéricas e imprimi-las em ordem decrescente (suponha números diferentes)

Salve seu arquivo como `ex10.php`

- 11) Criar uma variável numérica entre 1 e 12 e escrever o mês correspondente. Caso o número seja fora desse intervalo, informar que não existe mês com este número

Salve seu arquivo como `ex11.php`