**QUANDO USAR “==” OU “===”?**

Artigo da Alura em https://www.alura.com.br/artigos/quando-usar-ou-em-php

Afinal de contas, quando devemos ou não usar o **operador identico (===)**? Pra que serve esse operador de comparação? Qual a diferença entre eles?

Ao trabalhar com formulários, é comum a necessidade de validar um email e não existe uma forma padrão de fazer isso. Existem diversas regras de , por exemplo, que haja algo escrito antes do @. Uma forma de verificar isso é usando a função **[strpos](https://php.net/manual/pt_BR/function.strpos.php" \t "_blank)** que verifica a posição da primeira ocorrencia de uma string:

public function validaEmail($email) {

$posicao = strpos($email,@) ;

if($posicao == 0) { *//exibe uma menssagem de erro pra view dizendo que faltou a conta*

}

}

Mas, caso o usuario não digite nem o @ precisamos, também, dizer que aquele campo é especifico para email! No caso de não haver ocorrencia de string, a função strpos() retorna null. Portanto, para validar, também, este caso teriamos:

public function validaEmail($email) {

$posicao = strpos($email,@) ;

if($posicao == 0) { *//exibe uma menssagem de erro dizendo que faltou a conta// }*

if($posicao == null) { *//exibe uma menssagem de erro dizendo que aquele campo é especifico para email//*

}

}

}

Beleza, nossa lógica faz todo sentido! se o @ estiver na posição 0 da string $email falamos que faltou a conta. Se não houver @ na variavel $email exibimos outra mensagem dizendo que nosso usuario preencheu o campo errado!

Mas, se rodarmos esse código, as duas menssagens serão exibidas!

Isso acontece porque, na **[tabela de comparação de tipos do php](https://php.net/manual/pt_BR/types.comparisons.php" \t "_blank)** o valor 0 é igual a null! Precisamos, de alguma forma, comparar valores de uma forma mais especifica onde o tipo faça diferença. Neste caso, 0 seria apenas um inteiro qualquer, não outra representação de null.

E é aqui que devemos usar o operador identico ===! Com este operador, diferente do ==, levamos em conta não só o valor mas também o tipo na hora da comparação.

Então, nossa validação de email deve ser escrita um pouco diferente:

public function validaEmail($email) {

$posicao = strpos($email,@) ;

if($posicao === 0) { *//exibe uma menssagem de erro dizendo que faltou a conta// }*

if($posicao === null) { *//exibe uma menssagem de erro dizendo que aquele campo é especifico para email//*

}

}

}

Agora sim somente uma menssagem será exibida. Se a posição for zero, mostramos a menssagem dizendo que faltou a conta. Se não houver @ mostramos a menssagem dizendo que o campo é especifico para email.

Uma pequena diferença de tipos, pode causar um grande impacto nas funcionalidades do nosso sistema.

E você? O que achou do operador identico? Compartilhe com a gente sua opinião!

Para saber mais conceitos em PHP, aqui na [**Alura temos uma formação focada no desenvolvimento PHP**](https://www.alura.com.br/formacao-desenvolvedor-php)