Luís Dalmolin

Tecnologia, Web, PHP, SEO, e jQuery

Pesquisar

- Sobre
- Inverter URL
- Localizar IP

11 dicas de PHP que talvez você não conheça

Postado por luisdalmolin 2 Comentários

Navegando por aí, achei no blog do Davi Ferreira um post bem interessante sobre dias em **PHP** que você talvez não conheça.

Você sabia que a função **isset()**, em alguns casos, pode substituir a strlen()? Que o "**?>**" no final dos arquivos PHP é opcional? E que, a partir do **PHP 5**, é possível encadear métodos de uma classe?

Confira 11 dicas de PHP que talvez você não conheça.

1. Utilize os operadores === e !==

Essa muita gente desconhece e pode evitar muito tempo perdido com debug. Algumas funções do PHP retornam tanto um valor booleano como um valor inteiro, principalmente funções que tratam strings. E no caso de você querer verificar se a função retornou false e a mesma retornar 0. Como vocês sabem, 0 e false no PHP, em uma condicinal, significam a mesma coisa. Para esses casos, utilizamos os operadores === e !==. Esses operadores comparativos retornam true apenas quando os valores comparados são iguais e do mesmo tipo.

```
function validaUrl( $url ) {
    $pos = strpos( $url, 'http://' );
    if( $pos === false ) $url = 'http://' . $url;
    return $url;
}
echo validaUrl( 'www.daviferreira.com' );
echo validaUrl( 'http://www.apple.com' );
```

2. isset() brincando de strlen()

Aprendi faz pouco tempo, mas já a utilizo bastante. Primeiro porque, para verificar o tamanho de uma string, função isset é aproximadamente cinco vezes mais rápida do que a strlen. E segundo porque, caso a variável não exista, isset ainda funciona enquanto que strlen retornaria um erro.

```
if( !isset( $senha[5] ) ) {
   echo 'Sua senha deve possuir no mínimo 6 caracteres!';
}
```

3. Evitando aquele maldito erro dos Ifs

Sabe quando você esquece um sinal de igual usando if. Por exemplo: if(x = 1). Este exemplo retornaria sempre true e o PHP não acusaria nenhum erro. Existe uma maneira fácil de resolver isso, simplesmente mudar a variável de posição, como no exemplo abaixo. Neste caso, um if(1 = x) resultaria em erro.

```
if ( 5 == $area ) {
   echo 'Você está no menu 5.';
}
```

4. Zeros à esquerda (e à direita também!)

Utilizo muito essa dica em sistemas de cadastro de produtos, com códigos internos da loja. Geralmente o cliente pede para preencher o número com zeros ou algum outro caractere somente para impressão. A função str_pad faz isso pra gente. Ela recebe quatro parâmetros: a string a ser preenchida, o número de espaços a serem preenchidos, o caractere utilizado no preenchimento e a posição

(LEFT, RIGHT ou BOTH – esquerda, direita ou ambos). No caso do número de espaços ser menor do que o tamanho da string, a função não faz nada.

```
echo str_pad( '9', 10, '0', STR_PAD_LEFT ); // 0000000009
```

5. Com vocês, a função list()

Ah, não sei porque vivo esquecendo esta função! Talvez porque a maneira como é chamada seja pouco intuitiva. A falta de padrão no PHP é irritante às vezes (strpos e str_pad, por exemplo). Enfim, o que ela faz é pegar os valores de um array e criar variáveis com eles.

```
$localizacao = array( 'Brasil', 'RJ', 'Rio de Janeiro', 'Centro' );
$list( $pais, $estado, $cidade, $bairro ) = $localizacao;

// índices não-numéricos
$bd_config = array();
$bd_config['usuario'] = 'root';
$bd_config['senha'] = 'root';
$bd_config['banco'] = 'teste';
list( $usuario, $senha, $banco ) = array_values( $bd_config );
```

6. Esqueça (de vez) o '?>' no final dos arquivos

É isso mesmo! O "?>" no final dos arquivos PHP é desnecessário. E você deveria parar de utilizá-lo. Utilizando o fechamento, qualquer espaço em branco ou caractere estranho poderá gerar um erro no browser, no conteúdo exibido. Sem o fechamento seria retornado um erro pelo parser do PHP.

7. Métodos em cadeia

Desde o lançamento da versão 5 do PHP com melhorias na orientação a objetos é possível encadear métodos de uma classe. Para funcionar, essa dica necessita que seu método retorne um objeto. No exemplo abaixo, através do encadeamento de métodos o sistema valida o login de um usuário.

```
class usuarios {
        var $email;
        var $senha;
        var $erro;

        function __construct( $email, $senha ) {
                $this->email = $email;
                $this->senha = $senha;
                $this->erro  = NULL;
        }

        function criticaDados() {
                if( !$this->email ) $this->erro ='E-mail inválido';
                elseif( !isset( $this->senha[6] ) ) $this->erro = 'Senha inválida';
                return $this;
        }

        function login() {
                if( is_null( $this->erro ) ) {
                        return true;
                } else {
                        return $this->erro;
                }
        }
}
```

8. Par ou ímpar?

Tá bom, pode ser boba, mas em alguns casos é muito útil. O operador matemático % retorna o resto de uma divisão, portanto, podemos utilizá-lo para descobrir se um número é par ou ímpar. No exemplo abaixo, é gerada uma lista HTML alternando a classe dos ítens.

```
$count = 1;
for( $i = 0; $i < = 10; $i++ ) {
    if( $count%2 ) {
        $classe = 'normal';
    } else {
        $classe = 'alt';
```