

第50回PHP勉強会@関東
懇親会ライトニングトーク
2010年2月22日

テーマ

データ型について気になっていたこと。
私のつぶやき。

yuppyi(ゆっぴい)

twitter:@yuppyi

URL: <http://yuppyi.com/>

フォームの入力値は、何型？

メッセージをどうぞ。

おなまえ

ゆっぴい

メールアドレス

test@yuppyi.com

メールアドレス(再度入力)

test@yuppyi.com

年齢

100

メッセージ

ですと

※2000文字以内でお願いいたします。

確認

リセット

[home](#)

```
IN_NAME=>ゆっぴい(string)
IN_MAIL=>test@yuppyi.com(string)
IN_MAIL2=>test@yuppyi.com(string)
IN_AGE=>100(string)
IN_MESSAGE=>てすと(string)
SYORI=>確認(string)
```

入力値の型を
プリントしてみた。

メッセージをどうぞ。

入力内容をご確認の上、よろしければ送信してください。

おなまえ

メールアドレス

メールアドレス (再度入力)

年齢

メッセージ

	型
お名前	string
メールアドレス	string
年齢	string
メッセージ	string

確認方法

gettype()を用いてチェック

使用例

gettype(\$_POST['age'])

今まで気にしていませんでしたが
postされてくる段階で、型はないのかなあ
と思いました。

そこで、年齢をチェックするには

- 1 一旦数字型に直してからチェックする
- 2 文字列型として、
数字列が使われているかどうかをチェックする。

が考えられます。

1 一旦数字型に直してからチェックする。

```
$intAge = intval($_POST['age']);
```

欠点

入力値が数字以外の場合、
0になってしまう。

2 文字列型として、数字列が使われているかどうかをチェックする。

```
// 年齢  
if (preg_match("/^\d+$/", $_POST['IN_AGE'])) {  
    // 正常処理  
}  
else {  
    // 異常処理  
}
```

1の欠点を補うには、数字列が使われているか
チェックが必要になるので、結局2だけ実施すれば
バリデートという意味では、満足できる。

DBに登録する場合に、直前で下記のような型チェックをしたこともありました。

```
// integer型であるかどうかのチェック
if (-32768 <= $intHoge && $intHoge <= 32767) {
    // 正常範囲
}
else {
    // 異常範囲
}
```

結論

バリデーション、という意味で
数字かどうかをチェックするのであれば
型にこだわらなくてもよい。と思いました。

関数例

`is_numeric`

変数が数字または数値形式の文字列であるかを調べる

型まで調べるのであれば

`is_int`

変数が整数型かどうかを検査する

皆さんは、バリデーションにどんな手法を使われていますか
独自の関数を用意されていますか。
既存のPHP関数を利用されていますか。

フレームワークを使ってみえる場合は
自動的にクリアできているかも知れないですね。

ありがとうございました。