

明日のあなたの役に立たない PHP コーディング技法 ～ polyglot ～

nsfisis (いまむら)

第 148 回 PHP 勉強会@東京

自己紹介

nsfisis (いまむら)



@ デジタルサーカス株式会社

Polyglot とは

単一のソースコードが複数の言語として解釈可能なプログラム

今回は PHP ・ Ruby ・ Perl の組み合わせ

ソースコード

```
#<?php
$a = 'a'; $/* 0; # */$a
=begin
();
echo "php\n";
function begin() {} $a = <<<nil
=end
puts "#ruby"; '
=cut
print "#perl\n";
'# ';
nil;
```

実行結果: #php、#ruby、#perl

PHP として解釈する

```
#<?php
```

と PHP タグ

PHP として解釈する

```
$a = 'a'; $/* 0; # */$a  
=begin  
();
```

\$a に 'a' を代入

\$\$a = begin(); (variable variable)

PHP として解釈する

```
echo "php\n";
```

出力部分。実際の出力には先頭に # が付く

PHP として解釈する

```
function begin() {} $a = <<<nil
=end
puts "#ruby"; '
=cut
print "#perl\n";
'# ';
nil;
```

関数 `begin()` を定義

Here document (delimiter: `nil`) で残りを飲み込む

おわりに

これにもう 1 言語足したものを

PHPerKaigi 2023 の PHPer チャレンジに出題します！

Ruby として解釈する

```
#<?php
```

#は行コメント

Ruby として解釈する

```
$a = 'a'; $/* 0; # */$a
```

\$a に 'a' を代入

\$/ は Ruby の特殊なグローバル変数。* は乗算

#以降は行コメント

Ruby として解釈する

```
=begin  
();  
echo "php\n";  
function begin() {} $a = <<<nil  
=end
```

=begin から =end はコメント

Ruby として解釈する

```
puts "#ruby"; '
```

出力部。'からは文字列リテラル

Ruby として解釈する

```
puts "#ruby"; '  
=cut  
print "#perl\n";  
'# '  
nil;
```

文字列リテラルとして読み飛ばし、行コメント

`nil` は PHP でいう `null`

Perl として解釈する

```
#<?php
$a = 'a'; $/* 0; # */$a
=begin
();
echo "php\n";
function begin() {} $a = <<<nil
=end
puts "#ruby"; '
=cut
print "#perl\n";
'# ';
nil;
```

ほぼ Ruby と同じ。Perl だと `=begin` から `=cut` までがコメント