

明日のあなたの役に立たない PHP コーディング技法 ～細長い FizzBuzz を書く～

nsfisis (いまむら)

第 150 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

細長い FizzBuzz ?

```
<?php
```

```
for ($i = 1; $i <= 100; $i++) {  
    if ($i % 15 === 0) echo 'FizzBuzz', PHP_EOL;  
    else if ($i % 3 === 0) echo 'Fizz', PHP_EOL;  
    else if ($i % 5 === 0) echo 'Buzz', PHP_EOL;  
    else echo $i, PHP_EOL;  
}
```

細長い FizzBuzz ?

```
<?php
```

```
for ($i = 1; $i <= 100; $i++) {  
    if ($i % 15 === 0) echo 'FizzBuzz', PHP_EOL;  
    else if ($i % 3 === 0) echo 'Fizz', PHP_EOL;  
    else if ($i % 5 === 0) echo 'Buzz', PHP_EOL;  
    else echo $i, PHP_EOL;  
}
```

1行あたりの最大文字数を最小化しよう

1 行あたり最大 6 文字で書いた FizzBuzz

```
<?php
for($i
=1;$i
<=100;
$i++){
echo(
$i%3?
'':
'Fizz'
).($i%
5?'':
'Buzz'
)?$i,
"\n";}
```

細長い FizzBuzz ?

1 行何文字あれば FizzBuzz が書けるのか？

細長い FizzBuzz ？

1 行何文字あれば FizzBuzz が書けるのか？

2 文字あれば書ける！

何が難しいのか？

2 文字までのトークンしか使えない

- ほとんどのキーワードが書けない

- 書けるのは `as`、`do`、`fn`、`if`、`or` のみ

- ほとんどの関数が呼べない

- 呼べるのは `_` (`gettext` の別名)、`pi`、`dl` のみ

- 空でない文字列が書けない

- クォート、中身の文字、クォート、で最低 3 文字必要

ではどうするか？

FizzBuzz を細長くする

```
<?php
for ($i = 1; $i <= 100; $i++) {
    echo ($i%3?'': 'Fizz') . ($i%5?'': 'Buzz') ?: $i, "\n";
}
```

この FizzBuzz をベースに変形する

FizzBuzz を細長くする

```
<?php
for ($i = 1; $i <= 100; $i++) {
    echo ($i%3?'': 'Fizz') . ($i%5?'': 'Buzz') ?: $i, "\n";
}
```

ステップ 1: 絶対に短縮できない、**キーワードを排除する**

for と echo を消す

FizzBuzz を細長くする

```
<?
$a = range(1, 100);
array_walk(
    $a,
    fn($i) => printf(
        (($i%3?'':'Fizz') . ($i%5?'':'Buzz') ?: $i)
        . "\n"),
);
```

- for は array_walk と range に
- echo は printf に
- <?php は <? に (要: short_open_tag オプション)

FizzBuzz を細長くする

```
<?
$a = range(1, 100);
array_walk(
    $a,
    fn($i) => printf(
        (($i%3?'':'Fizz') . ($i%5?'':'Buzz') ?: $i)
        . "\n"),
    );
```

さっきよりも長くなってない？

FizzBuzz を細長くする

```
<?
$a = range(1, 100);
array_walk(
    $a,
    fn($i) => printf(
        (($i%3?'':'Fizz') . ($i%5?'':'Buzz') ?: $i)
        . "\n"),
);
```

ステップ 2: 関数呼び出しを variable function 経由に

FizzBuzz を細長くする

```
<?
$r = 'range'; $w = 'array_walk'; $p = 'printf';
$f = 'Fizz'; $b = 'Buzz';
$a = $r(1, 10*10);
$w($a,
    fn($i) => $p(
        (($i%3?'':$f) . ($i%5?'':$b) ?: $i) . "
    ));
```

- 関数名と Fizz、Buzz を変数に代入
- 100 を 10*10 に
- "\n" を使わず、直接改行を挿入
- 上を無視すれば、2 文字以下のトークンだけで構成されている

FizzBuzz を細長くする

```
<?
$r = 'range'; $w = 'array_walk'; $p = 'printf';
$f = 'Fizz'; $b = 'Buzz';
$a = $r(1, 10*10);
$w($a,
    fn($i) => $p(
        (($i%3?'':$f) . ($i%5?'':$b) ?: $i) . "
    ));
```

ステップ 3: 文字列を幅 2 文字以下で生成する

文字列リテラルの短縮

```
<?php
$a = "12345";
$b = "world";
// $a ^ $b は次のコードと同じ
$result = '';
for ($i = 0; $i < min(strlen($a), strlen($b)); $i++) {
    $result .= $a[$i] ^ $b[$i];
}
echo $result;
// => F]AXQ
```

文字列の XOR 演算を使う

文字列をバイト列と見做し、各要素に XOR 演算を適用して結合する

文字列リテラルの短縮

```
<?php  
echo (  
  "L  
  [p  
  ""^  
  "  
  c!  
  "  
);  
// => Fizz
```

ほとんどの文字列を、1 行 2 文字以下で表せる

細長い FizzBuzz 完成

```
<?  
$a  
=(  
'x  
0m  
'^  
( '  
k!  
o'  
))  
(1  
,  
10
```

細長い FizzBuzz 完成

```
*  
10  
);  
( '  
x!  
s!  
k!  
' ^  
' k  
Sk  
~}  
Ma  
' )
```

細長い FizzBuzz 完成

```
(  
$a  
,  
fn  
(  
$i  
)  
=>  
( '  
x!  
~!  
.^  
'z
```

細長い FizzBuzz 完成

```
Hd  
G'  
)(  
((  
$i  
%3  
?  
..  
:  
c!  
.^  
'L  
[p
```

細長い FizzBuzz 完成

```
' )  
.(  
$i  
%5  
?  
..  
:(  
'H  
.^  
( '  
' )  
) .  
( '
```

細長い FizzBuzz 完成

```
b!  
'^  
'i  
S'  
)(  
3*  
3*  
13  
)  
( '  
p '  
^  
'p
```

細長い FizzBuzz 完成

```
' )  
) ?  
:  
$i  
) .  
"  
")  
);
```

完成！

そしてその先へ

この縛りの下で、 どのくらい複雑なプログラムが書けるのか？

そしてその先へ

この縛りの下で、 どのくらい複雑なプログラムが書けるのか？

任意のプログラムが書ける！

そしてその先へ

この縛りの下で、 どのくらい複雑なプログラムが書けるのか？

任意のプログラムが書ける！

Laravel も動かせる

そしてその先へ

任意のプログラムを動かそうとしたときの壁

`eval`は関数ではないので、variable function が使えない

そしてその先へ

ではどうやって Laravel を動かすか？

おわりに

続きは **PHPerKaigi 2023** で

没スライド

余談：PHP 8 系で動かなくてもいいなら

```
$f  
=F  
.i  
.Z  
.Z  
;
```

未定義の定数が評価されると、その定数の名前の文字列になる (PHP 7.x 以下)

余談：PHP 8 系で動かなくてもいいなら

```
$f  
=@  
F.  
@i  
  
.  
@z  
  
.  
@z  
;
```

警告を抑制するため、@演算子を使う