詳説　「参照」
PHP の参照を完全に理解する

nsfisis (いまむら)

PHPerKaigi 2023

自己紹介

nsfisis (いまむら)

@ デジタルサーカス株式会社

アジェンダ

1. 参照の不思議クイズ
2. PHP 処理系のソースを読む
   (a) zval と zend_reference
   (b) 参照代入
3. クイズの解説
4. まとめ

参照の不思議クイズ

参照の不思議クイズ : Q1

```php
$x = 1;
$y =& $x;
$y = 42;
echo "x = $x\n";
// => ???
echo "y = $y\n";
// => ???
```

参照の不思議クイズ : Q1

```php
$x = 1;
$y =& $x;
$y = 42;
echo "x = $x\n";
// => 42
echo "y = $y\n";
// => 42
```

参照の不思議クイズ：Q2

```php
$x = 1;
$y =& $x;
$z = $y;
$z = 42;
echo "x = $x\n";
// => ???
echo "y = $y\n";
// => ???
echo "z = $z\n";
// => ???
```

参照の不思議クイズ：Q2

```php
$x = 1;
$y =& $x;
$z = $y;
$z = 42;
echo "x = $x\n";
// => 1
echo "y = $y\n";
// => 1
echo "z = $z\n";
// => 42
```

参照の不思議クイズ：Q3

```php
$xs = [1, 2];
$x =& $xs[0];
$x = 42;
echo "x = $x\n";
// => ???
echo "xs = [$xs[0], $xs[1]]\n";
// => [???, ???]
```

参照の不思議クイズ：Q3

```php
$xs = [1, 2];
$x =& $xs[0];
$x = 42;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
```

参照の不思議クイズ：Q4

```php
$xs = [1, 2];
$x =& $xs[0];
$ys = $xs;
$x = 42;
$ys[1] = 3;
echo "x = $x\n";
// => ???
echo "xs = [$xs[0], $xs[1]]\n";
// => [???, ???]
echo "ys = [$ys[0], $ys[1]]\n";
// => [???, ???]
```

参照の不思議クイズ：Q4

```php
$xs = [1, 2];
$x =& $xs[0];
$ys = $xs;
$x = 42;
$ys[1] = 3;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
echo "ys = [$ys[0], $ys[1]]\n";
// => [42, 3]
```

PHP 処理系のソースを読む

はじめに

- PHP v8.2.3 (GitHub)
- ソースコードは発表向けに改変しています
- C 言語としては不正確な説明を話したり載せたりすることがあります

zval と zend_reference

zval とは

PHP の「値」全般

- 整数(`0`、`42`、`57`)
- 浮動小数点数(`3.14159265`)
- 文字列(`"Hello, World!"`)
- 配列(`[1, 2, 3]`)
- クラス(`new \Exception`)

など

zval の定義

```
struct zval {
  zend_value value;      /* 値本体 */
  uint32_t   type_info; /* 型情報 */


  union {
    /* その他メタデータ (省略) */
  } u2;
};
```

Zend/zend_types.h#L315-L340

zend_value の定義

```
union zend_value {
    zend_long         lval;    /* 整数 */
    double            dval;    /* 浮動小数点数 */
    zend_string      *str;     /* 文字列 */
    zend_array       *arr;     /* 配列 */
    zend_object      *obj;     /* オブジェクト */
    zend_resource    *res;     /* リソース */
    zend_reference   *ref;     /* 参照 */
    /* (略) */
};
```

Zend/zend_types.h#L295-L313

zend_value の定義

```
union zend_value {
    zend_long         lval;    /* 整数 */
    double            dval;    /* 浮動小数点数 */
    zend_string      *str;     /* 文字列 */
    zend_array       *arr;     /* 配列 */
    zend_object      *obj;     /* オブジェクト */
    zend_resource    *res;     /* リソース */
    zend_reference   *ref;     /* 参照 */
    /* (略) */
};
```

どれが入っているのかをどう区別する？

zval の定義

```
struct zval {
  zend_value value;      /* 値本体 */
  uint32_t   type_info; /* 型情報 */


  union {
    /* その他メタデータ (省略) */
  } u2;
};
```

`type_info`が型情報を保持している

この値を見て、`zend_value`に何が入っているかを区別する

いわゆる"tagged-union"

PHP の型

```
#define IS_UNDEF         0 /* 不明、未初期化 */
#define IS_NULL          1 /* null */
#define IS_FALSE         2 /* false */
#define IS_TRUE          3 /* true */
#define IS_LONG          4 /* 整数 */
#define IS_DOUBLE        5 /* 浮動小数点数 */
#define IS_STRING        6 /* 文字列 */
#define IS_ARRAY         7 /* 配列 */
#define IS_OBJECT        8 /* オブジェクト */
#define IS_RESOURCE      9 /* リソース */
#define IS_REFERENCE    10 /* 参照 */
```

参照は`IS_REFERENCE`

内部的には、独立した型として実装されている

PHP の型

```
union zend_value {
    zend_long      lval;    /* 整数 */
    double         dval;    /* 浮動小数点数 */
    zend_string    *str;    /* 文字列 */
    zend_array     *arr;    /* 配列 */
    zend_object    *obj;    /* オブジェクト */
    zend_resource  *res;    /* リソース */
    zend_reference *ref;    /* 参照 */
    /* (略) */
};
```

`zend_reference`はどんなデータ構造か？

zend_reference の定義

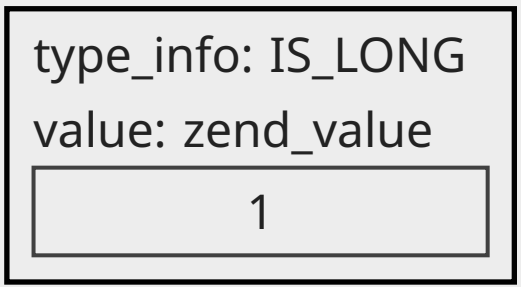
```
struct zend_reference {
  uint32_t   refcount;  /* 参照カウント */
  uint32_t   type_info; /* (説明略) */
  zval       val;       /* 指している値 */
  zend_property_info_source_list sources;   /* (説明略) */
};
```

Zend/zend_types.h#L537-L541

参照カウント : 同じ値への参照がどれだけあるか？

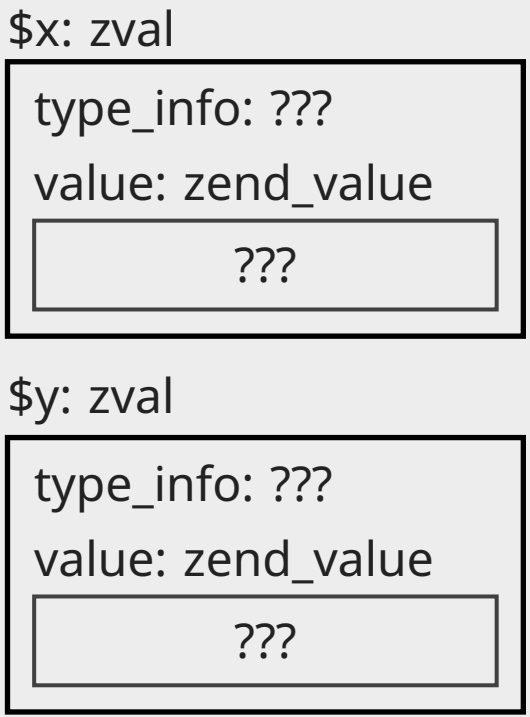
具体例

```
$x = 1;
```



具体例

```
$x = 1;
$y =& $x;
```


$x: zval
type_info: ???
value: zend_value
???
$y: zval
type_info: ???
value: zend_value
???

参照代入

zend_assign_to_variable_reference

```
void zend_assign_to_variable_reference(
  zval *lhs, zval *rhs
) {
  ZVAL_NEW_REF(rhs);


  rhs->value->refcount++;


  lhs->value = rhs->value;
  lhs->type_info = IS_REFERENCE;
}
```

Zend/zend_execute.c#L533-L557

`$lhs =& $rhs;`

`$lhs`= 左辺

`$rhs`= 右辺

ZVAL_NEW_REF

```
ZVAL_NEW_REF(z)
  zend_reference *ref = new zend_reference();
  ref->refcount = 1;
  /* z の中に入っている値を ref にコピー */
  ZVAL_COPY_VALUE(&ref->val, z);
  z->value = ref;
  z->type_info = IS_REFERENCE;
```

Zend/zend_types.h#L1077-L1086

z の中身を参照でラップする

zend_assign_to_variable_reference

```
void zend_assign_to_variable_reference(
  zval *lhs, zval *rhs
) {
  ZVAL_NEW_REF(rhs);


  rhs->value->refcount++;


  lhs->value = rhs->value;
  lhs->type_info = IS_REFERENCE;
}
```

具体例

```
$rhs = 1;
$lhs =& $rhs;
```

```
ZVAL_NEW_REF(rhs);


rhs->value->refcount++;


lhs->value = rhs->value;
lhs->type_info = IS_REFERENCE;
```

$rhs: zval

type_info: IS_LONG

value: zend_value

1

具体例

```
$rhs = 1;
$lhs =& $rhs;
```

```
/* ZVAL_NEW_REF */
zend_reference *ref =
  new zend_reference();
ref->refcount = 1;
ZVAL_COPY_VALUE(&ref->val, rhs);
rhs->value = ref;
rhs->type_info = IS_REFERENCE;
```

$rhs: zval

type_info: IS_LONG

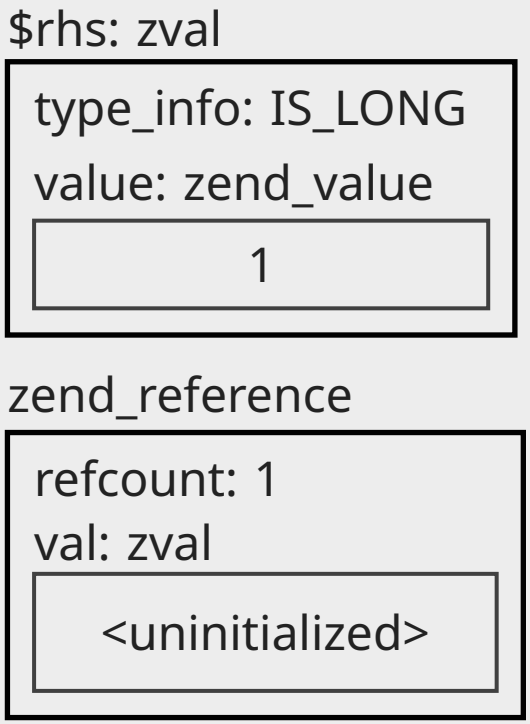
value: zend_value

1

具体例

```
$rhs = 1;
$lhs =& $rhs;
```

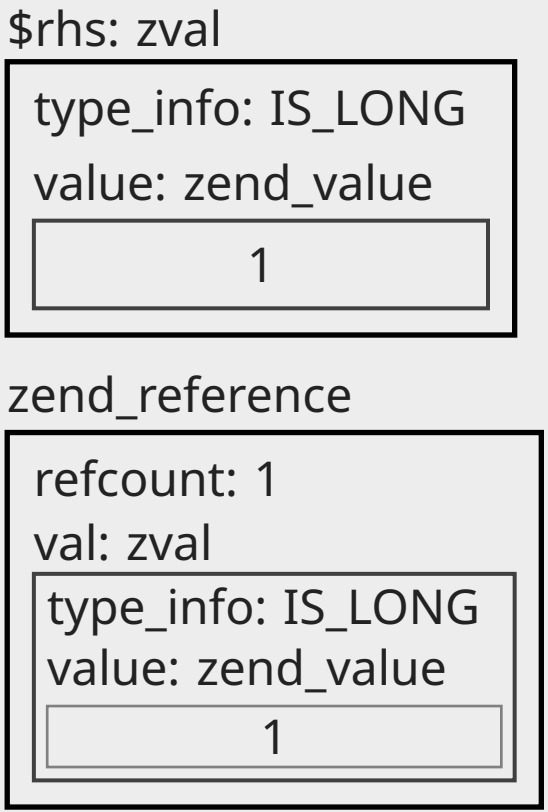
```
/* ZVAL_NEW_REF */
zend_reference *ref =
  new zend_reference();
ref->refcount = 1;
ZVAL_COPY_VALUE(&ref->val, rhs);
rhs->value = ref;
rhs->type_info = IS_REFERENCE;
```



具体例

```
$rhs = 1;
$lhs =& $rhs;
```

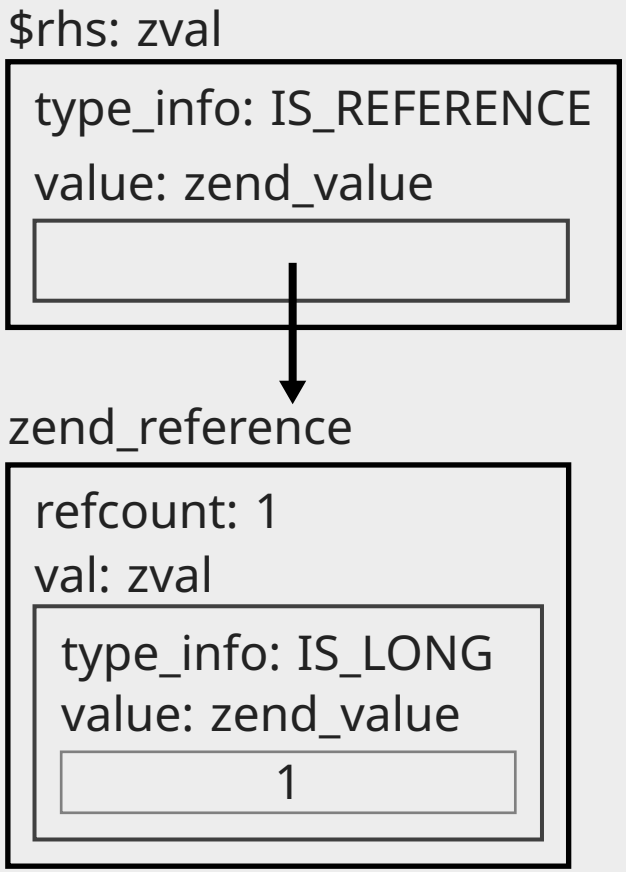
```
/* ZVAL_NEW_REF */
zend_reference *ref =
  new zend_reference();
ref->refcount = 1;
ZVAL_COPY_VALUE(&ref->val, rhs);
rhs->value = ref;
rhs->type_info = IS_REFERENCE;
```



具体例

```
$rhs = 1;
$lhs =& $rhs;
```

```
/* ZVAL_NEW_REF */
zend_reference *ref =
  new zend_reference();
ref->refcount = 1;
ZVAL_COPY_VALUE(&ref->val, rhs);
rhs->value = ref;
rhs->type_info = IS_REFERENCE;
```



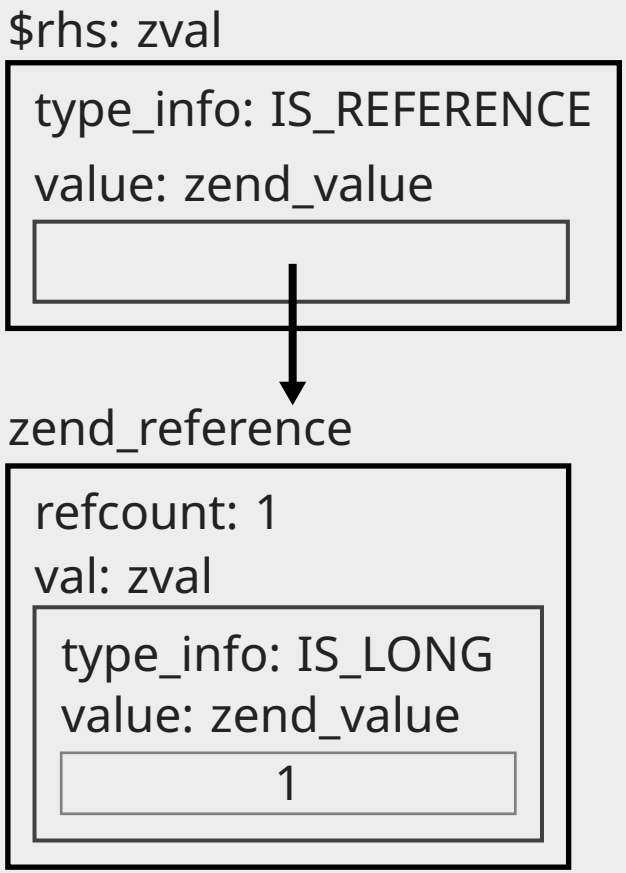
具体例

```
$rhs = 1;
$lhs =& $rhs;
```

```
ZVAL_NEW_REF(rhs);


rhs->value->refcount++;


lhs->value = rhs->value;
lhs->type_info = IS_REFERENCE;
```



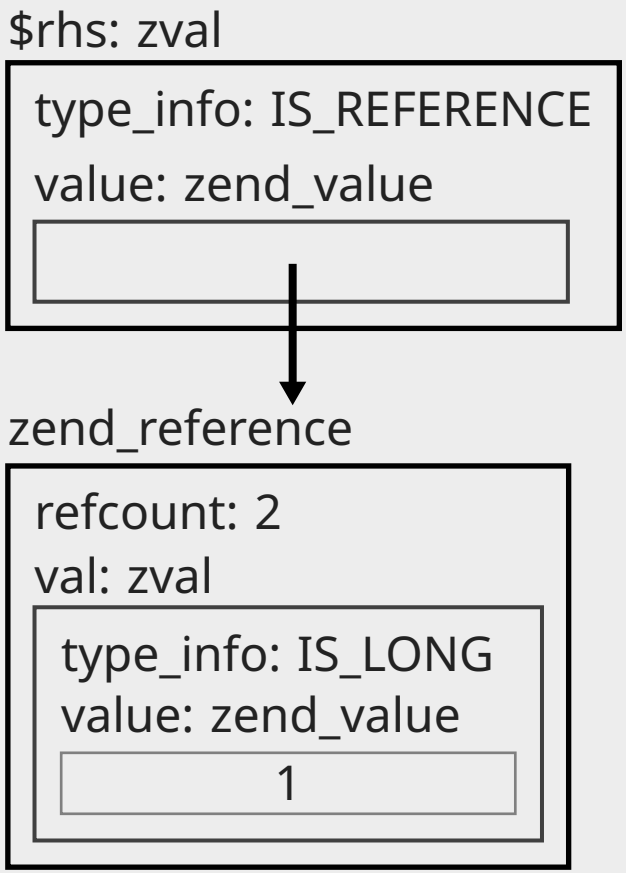
具体例

```
$rhs = 1;
$lhs =& $rhs;
```

```
ZVAL_NEW_REF(rhs);


rhs->value->refcount++;


lhs->value = rhs->value;
lhs->type_info = IS_REFERENCE;
```



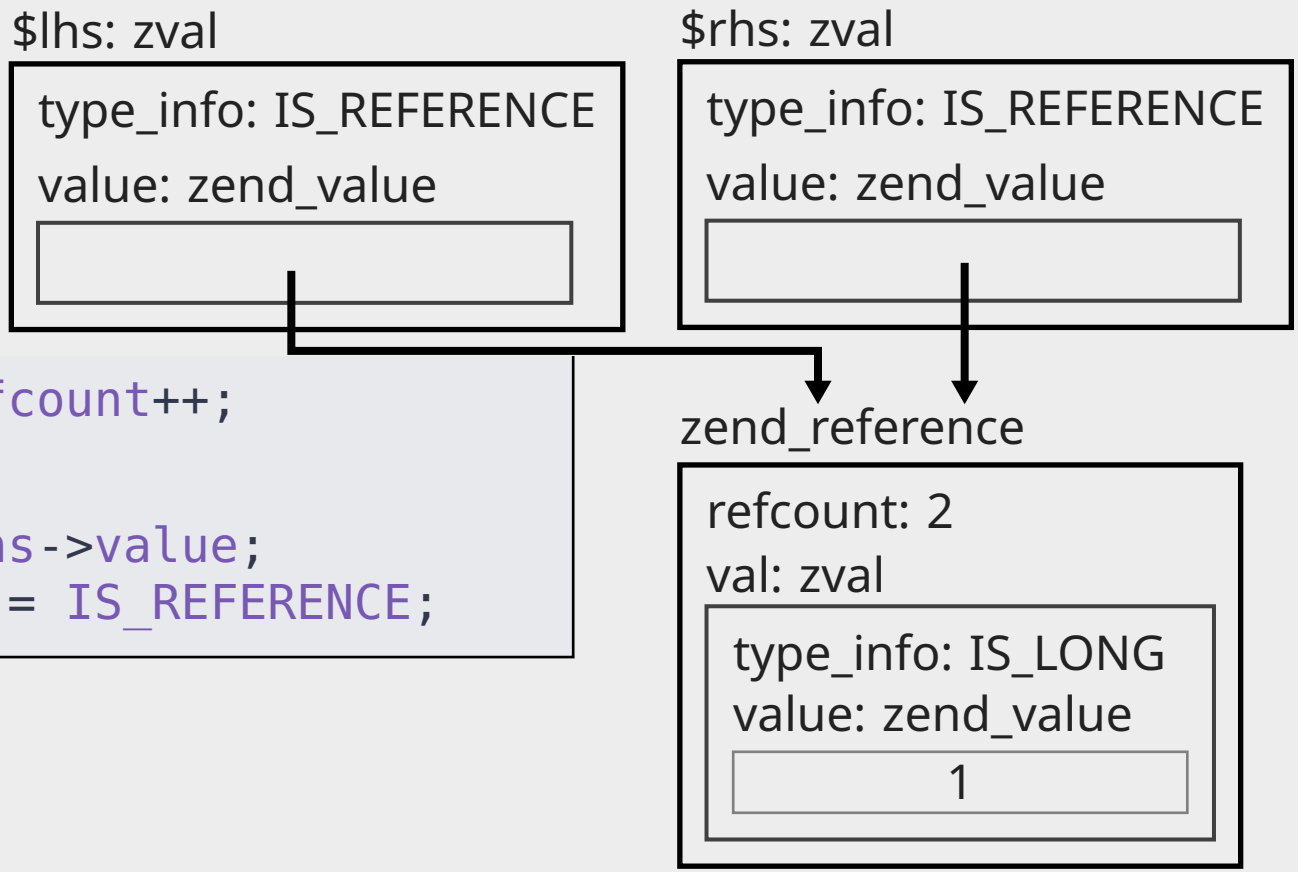
具体例

```
$rhs = 1;
$lhs =& $rhs;
```

```
ZVAL_NEW_REF(r

rhs->value->refcount++;

lhs->value = rhs->value;
lhs->type_info = IS_REFERENCE;
```



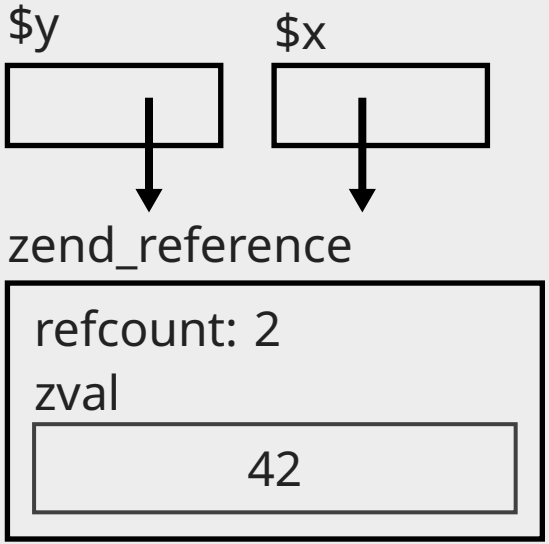
クイズの解説

クイズの解説 : Q1

```php
$x = 1;
$y =& $x;
$y = 42;
echo "x = $x\n";
// => 42
echo "y = $y\n";
// => 42
```

クイズの解説 : Q1

```php
$x = 1;
$y =& $x;
$y = 42;
echo "x = $x\n";
// => 42
echo "y = $y\n";
// => 42
```

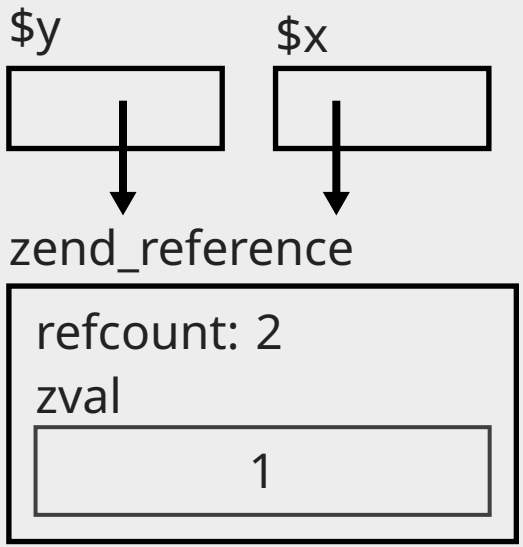


クイズの解説 : Q2

```php
$x = 1;
$y =& $x;
$z = $y;
$z = 42;
echo "x = $x\n";
// => 1
echo "y = $y\n";
// => 1
echo "z = $z\n";
// => 42
```

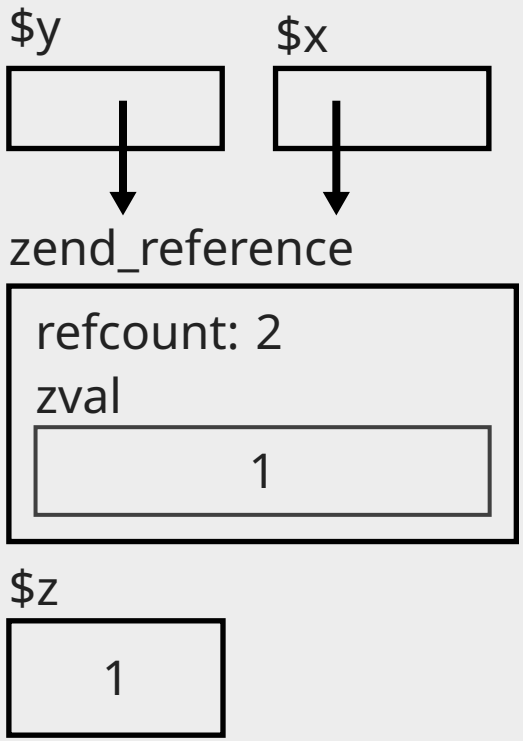
クイズの解説 : Q2

```php
$x = 1;
$y =& $x;
$z = $y;
$z = 42;
echo "x = $x\n";
// => 1
echo "y = $y\n";
// => 1
echo "z = $z\n";
// => 42
```



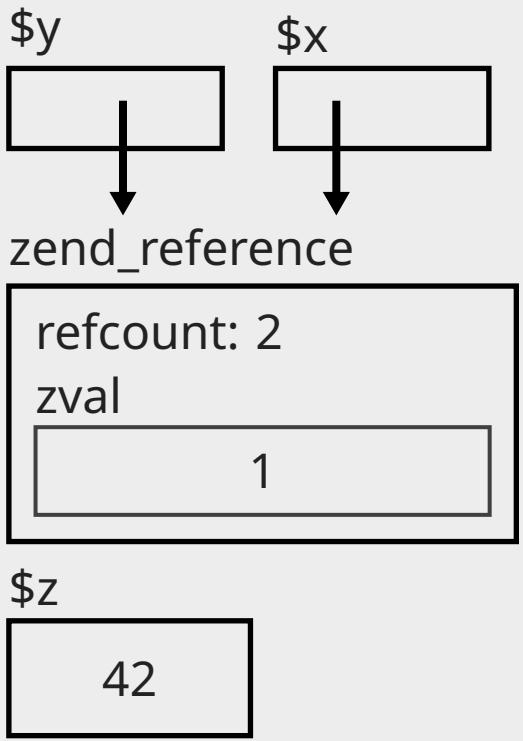
クイズの解説 : Q2

```php
$x = 1;
$y =& $x;
$z = $y;
$z = 42;
echo "x = $x\n";
// => 1
echo "y = $y\n";
// => 1
echo "z = $z\n";
// => 42
```



クイズの解説 : Q2

```php
$x = 1;
$y =& $x;
$z = $y;
$z = 42;
echo "x = $x\n";
// => 1
echo "y = $y\n";
// => 1
echo "z = $z\n";
// => 42
```

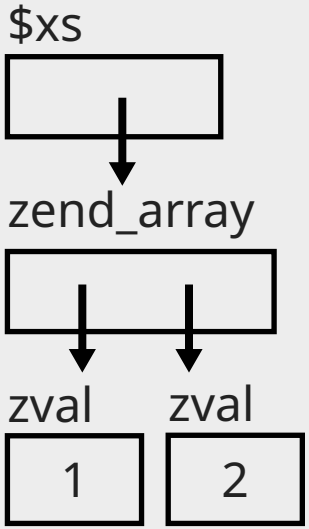


クイズの解説 : Q3

```php
$xs = [1, 2];
$x =& $xs[0];
$x = 42;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
```

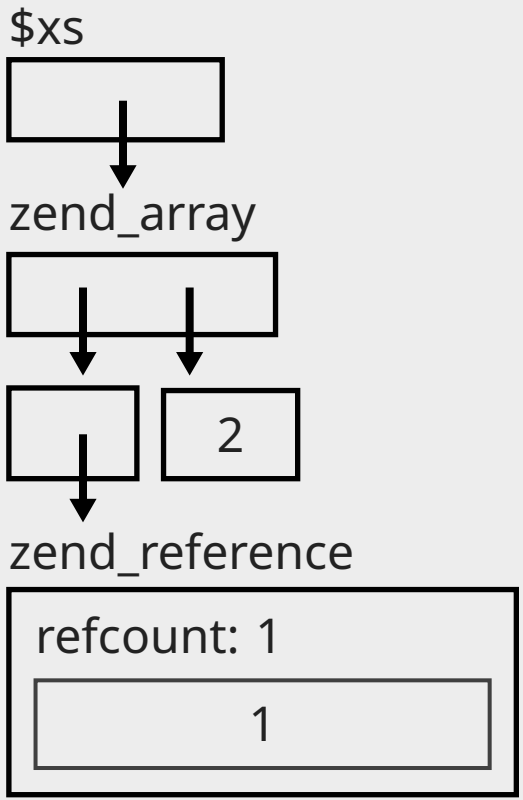
クイズの解説 : Q3

```php
$xs = [1, 2];
$x =& $xs[0];
$x = 42;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
```



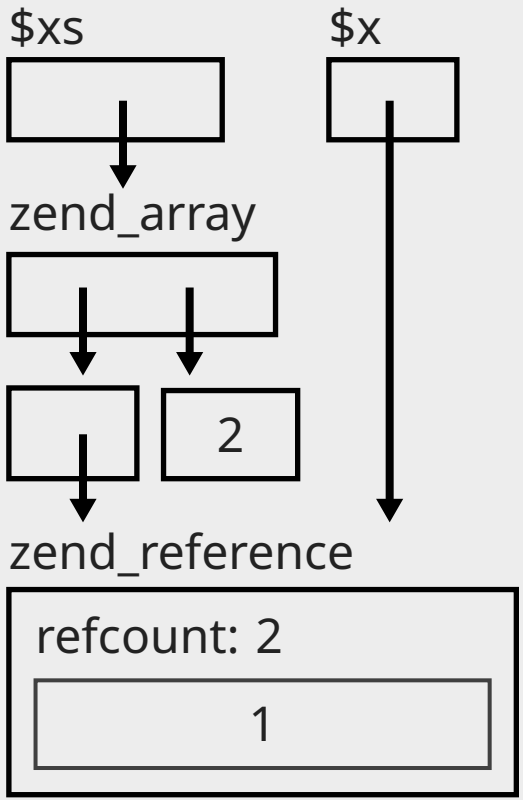
クイズの解説 : Q3

```php
$xs = [1, 2];
$x =& $xs[0];
$x = 42;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
```



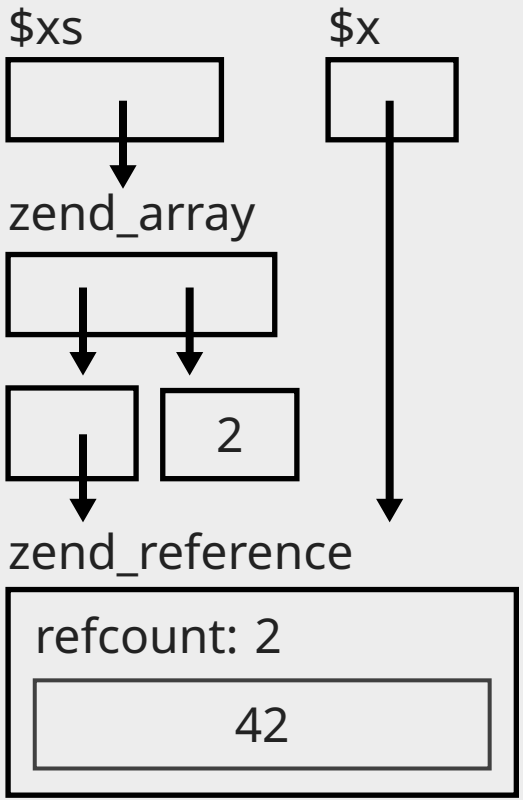
クイズの解説 : Q3

```php
$xs = [1, 2];
$x =& $xs[0];
$x = 42;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
```



クイズの解説 : Q3

```php
$xs = [1, 2];
$x =& $xs[0];
$x = 42;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
```

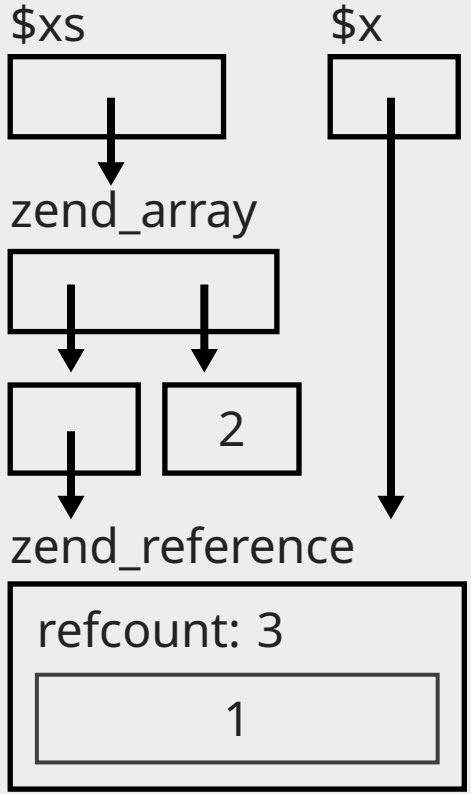


クイズの解説 : Q4

```php
$xs = [1, 2];
$x =& $xs[0];
$ys = $xs;
$x = 42;
$ys[1] = 3;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
echo "ys = [$ys[0], $ys[1]]\n";
// => [42, 3]
```

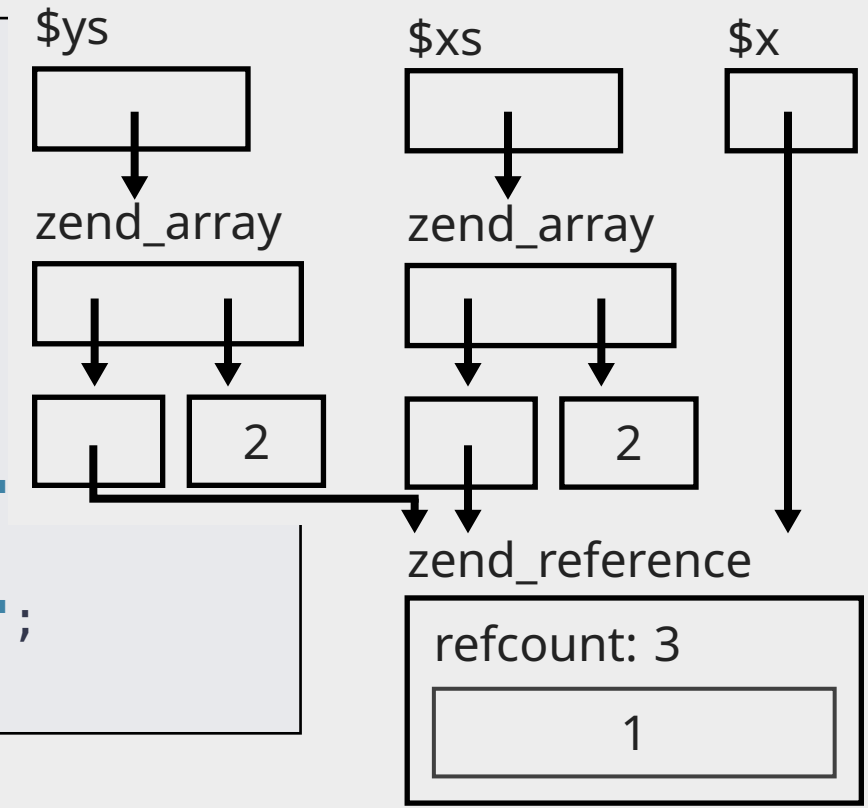
クイズの解説 : Q4

```php
$xs = [1, 2];
$x =& $xs[0];
$ys = $xs;
$x = 42;
$ys[1] = 3;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
echo "ys = [$ys[0], $ys[1]]\n";
// => [42, 3]
```



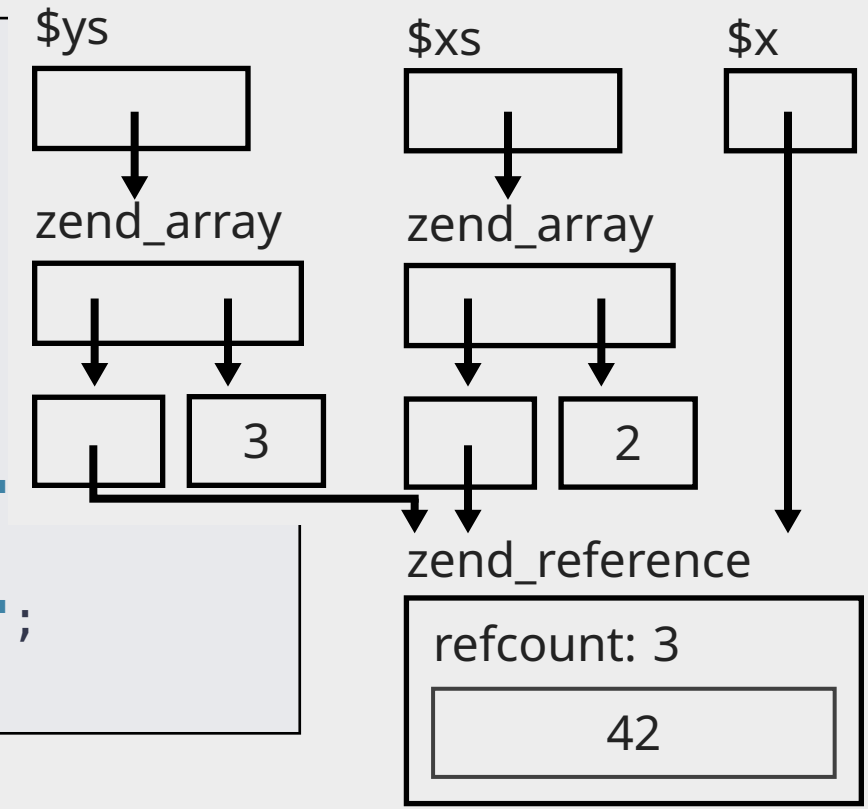
クイズの解説 : Q4

```php
$xs = [1, 2];
$x =& $xs[0];
$ys = $xs;
$x = 42;
$ys[1] = 3;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
echo "ys = [$ys[0], $ys[1]]\n";
// => [42, 3]
```



クイズの解説 : Q4

```php
$xs = [1, 2];
$x =& $xs[0];
$ys = $xs;
$x = 42;
$ys[1] = 3;
echo "x = $x\n";
// => 42
echo "xs = [$xs[0], $xs[1]]\n";
// => [42, 2]
echo "ys = [$ys[0], $ys[1]]\n";
// => [42, 3]
```



まとめ

- C が読めると世界が広がる
  - PHP、Apache httpd、MySQL 等
- PHP の処理系を気軽に読もう

おまけ

入り切らなかったクイズ

参照の不思議クイズ：Q5

```php
$g = 1;
function f(&$x) {
  $x =& $GLOBALS['g'];
}
$y = 0;
f($y);
$y = 42;
echo "y = $y", PHP_EOL;
// => ???
echo "g = $g", PHP_EOL;
// => ???
```

参照の不思議クイズ：Q5

```php
$g = 1;
function f(&$x) {
  $x =& $GLOBALS['g'];
}
$y = 0;
f($y);
$y = 42;
echo "y = $y", PHP_EOL;
// => 42
echo "g = $g", PHP_EOL;
// => 1
```

参照の不思議クイズ：Q6

```php
class C {
  public int $x = 1;
}
$c = new C();
$y =& $c->x;
$y = 'PHPerKaigi';
// => ???
```

参照の不思議クイズ：Q6

```php
class C {
  public int $x = 1;
}
$c = new C();
$y =& $c->x;
$y = 'PHPerKaigi';
// => Fatal error: TypeError
```