



# PHPOpenGL 詳解

2012/06/16 “よや” <yoya@awm.jp>

# OpenGLとは

- 3D を扱えるグラフィック API
- よく DirectX3D と対比して語られる
- 詳しくはこちらで↓
- <http://www.opengl.org/>



# PHPOpenGL とは

- PHP で OpenGL API を使う為の拡張
  - <http://phpopengl.sourceforge.net/>
- 公式プロジェクトは **PHP4 で開発停止**
- 自分が**勝手に PHP5 対応**して github で公開中 ↓
  - <https://github.com/yoya/phpopengl>

# OpenGLライブラリ

- OpenGL 本体(GL)に足りない機能を補う GLU, GLUT といったライブラリがあり、セットで使うと便利。

GL  
(OpenGL本体)

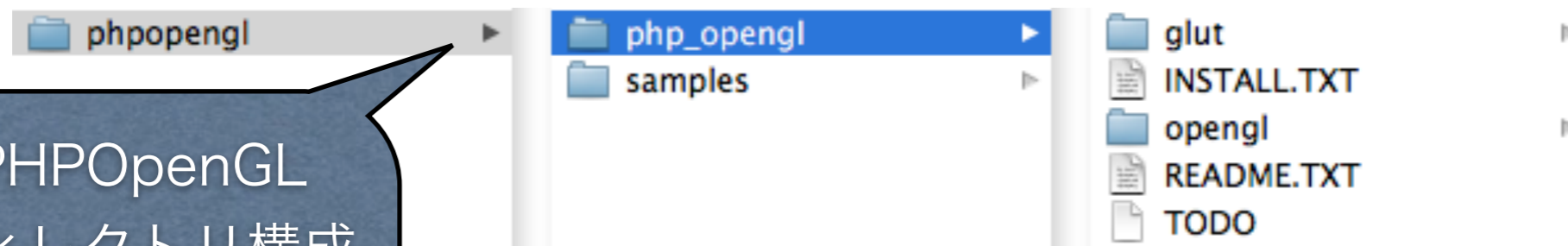
GLU  
gluLookAt

GLUT

キーボード  
やマウス等の  
イベント処理も可能

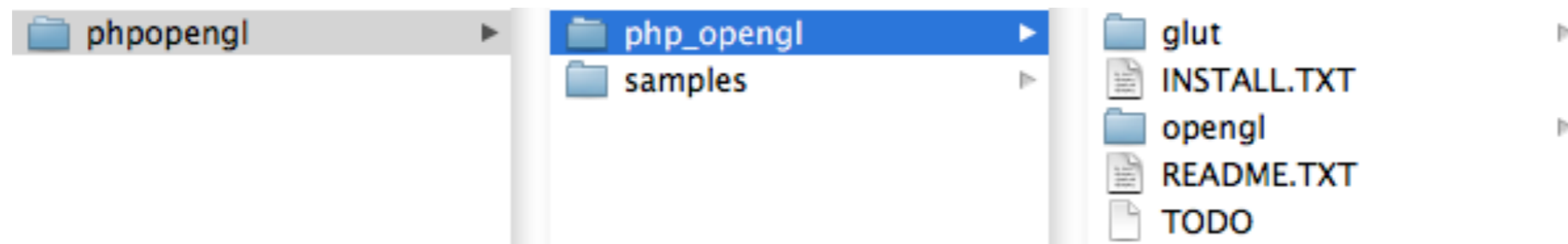
# PHP 拡張モジュール

- OpenGL ライブラリと PHPOpenGL PHP モジュール(.so)の対応関係
- GL+GLU: opengl.so, GLUT: glut.so
- [git://github.com/yoya/phpopengl.git](https://github.com/yoya/phpopengl)



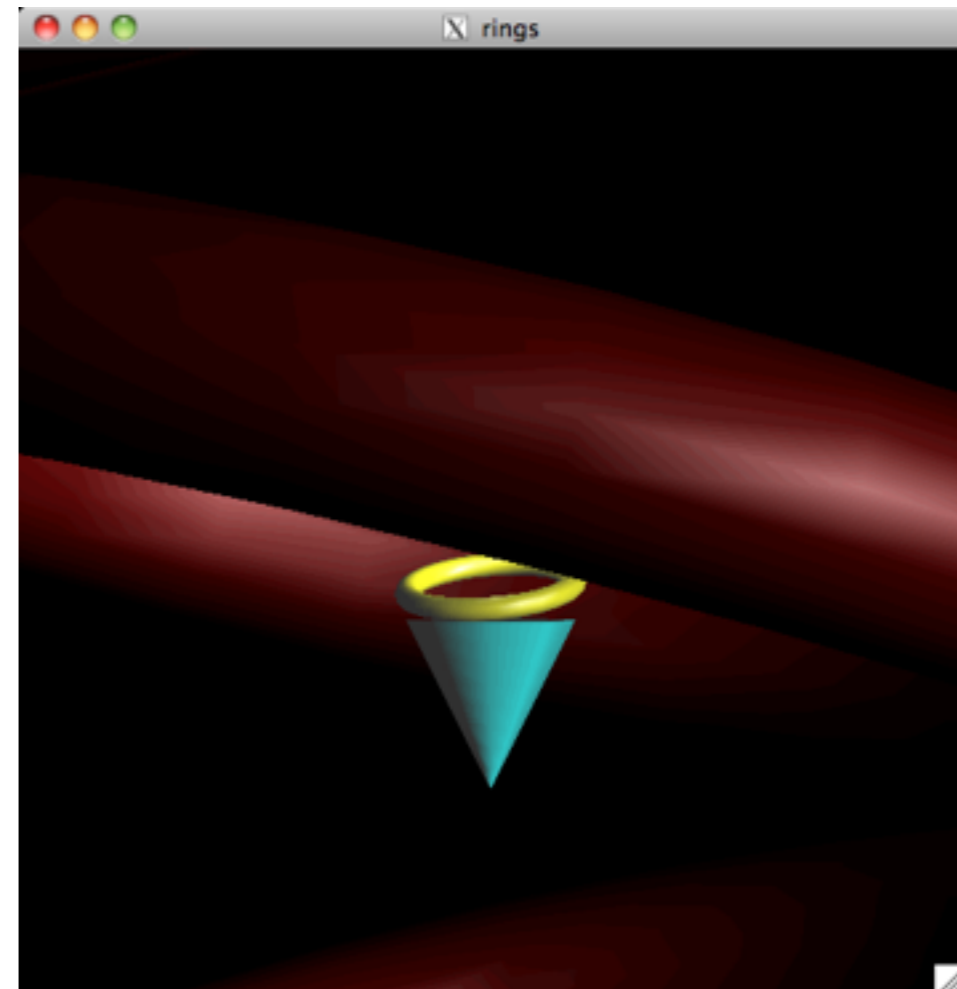
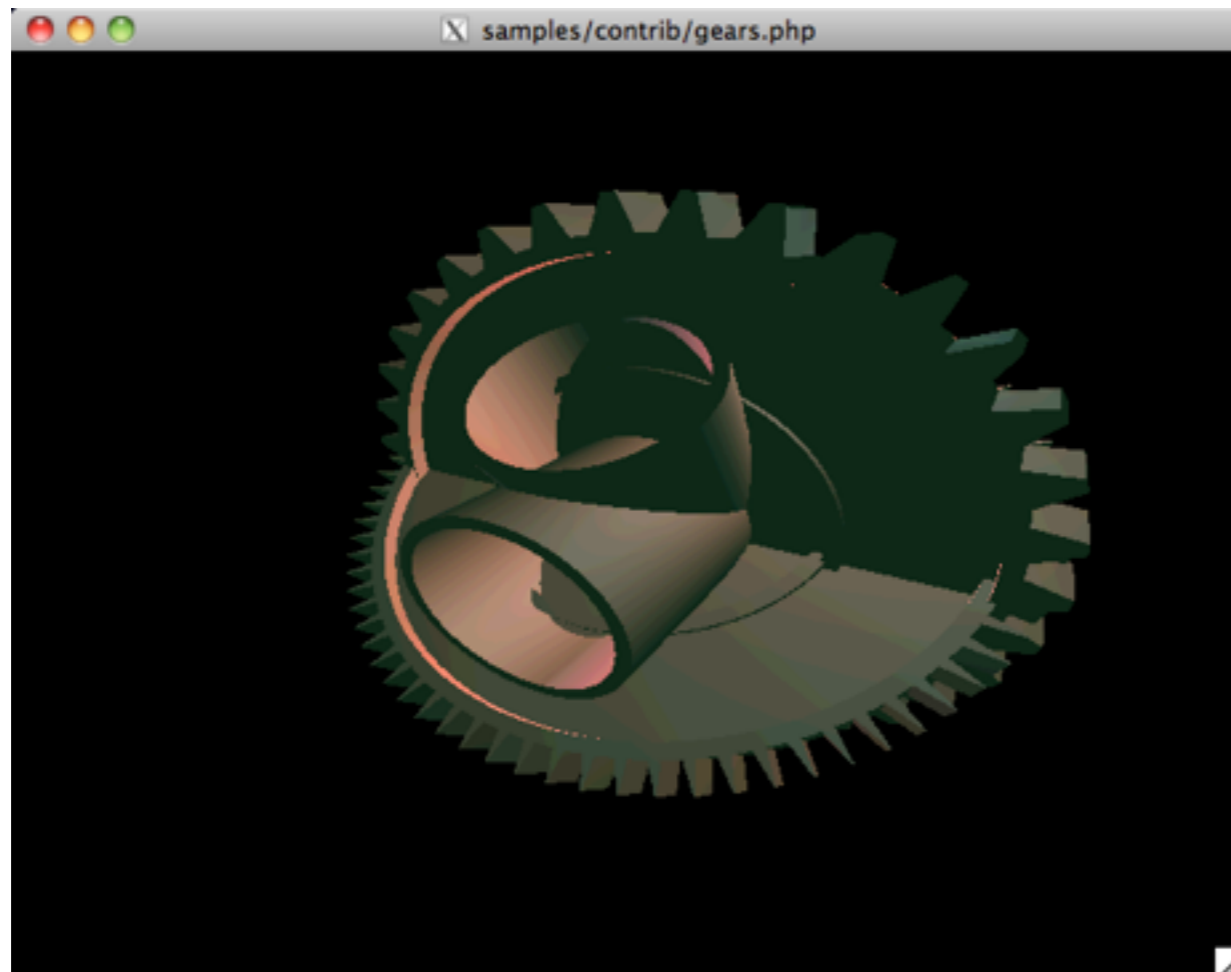
PHPOpenGL  
ディレクトリ構成

# インストール方法



- php\_opengl 以下の opengl, glut ディレクトリで  
phpize ; configure ; make install
- php.ini に追加する (いつもの方法)

# 動作デモ



# ここから詳解 (本題)

- PHPOpenGL binding 詳解
  - PHP extension を少しおさらい
  - モジュール、関数、引数定義
  - 配列変換 (C  $\Leftrightarrow$  PHP)
  - C  $\Rightarrow$  PHP 関数 callback
  - config.m4 と config.w32



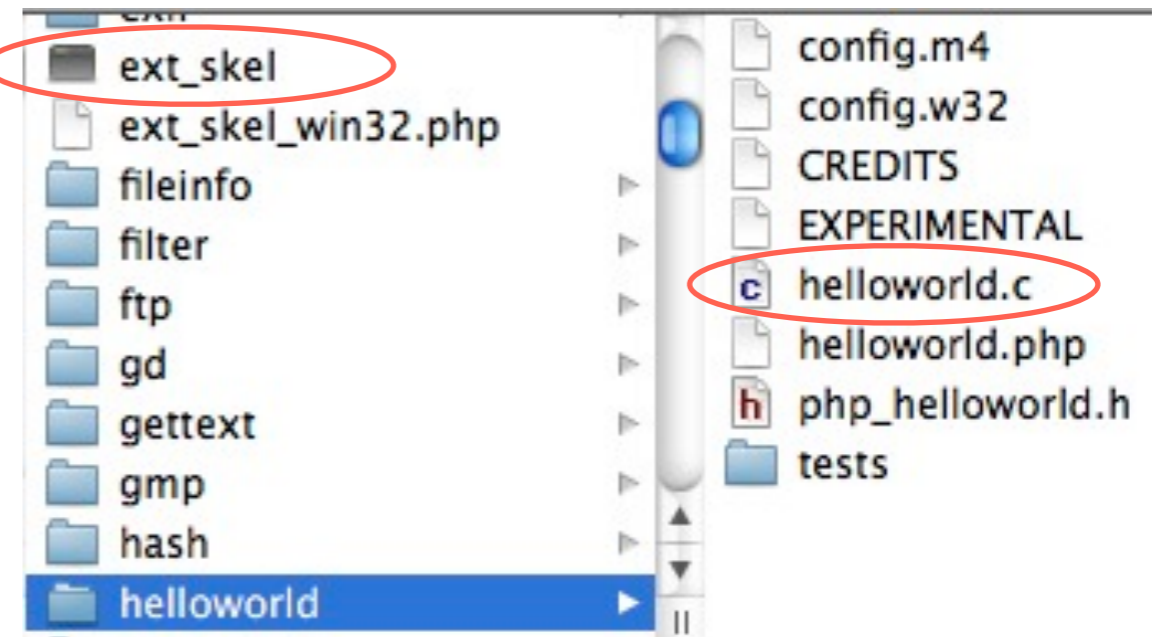
# 説明しない事

- class は今回扱いません。
- OpenGL は C言語の API だから

# 復習

- <http://dsas.blog.klab.org/archives/50777398.html>

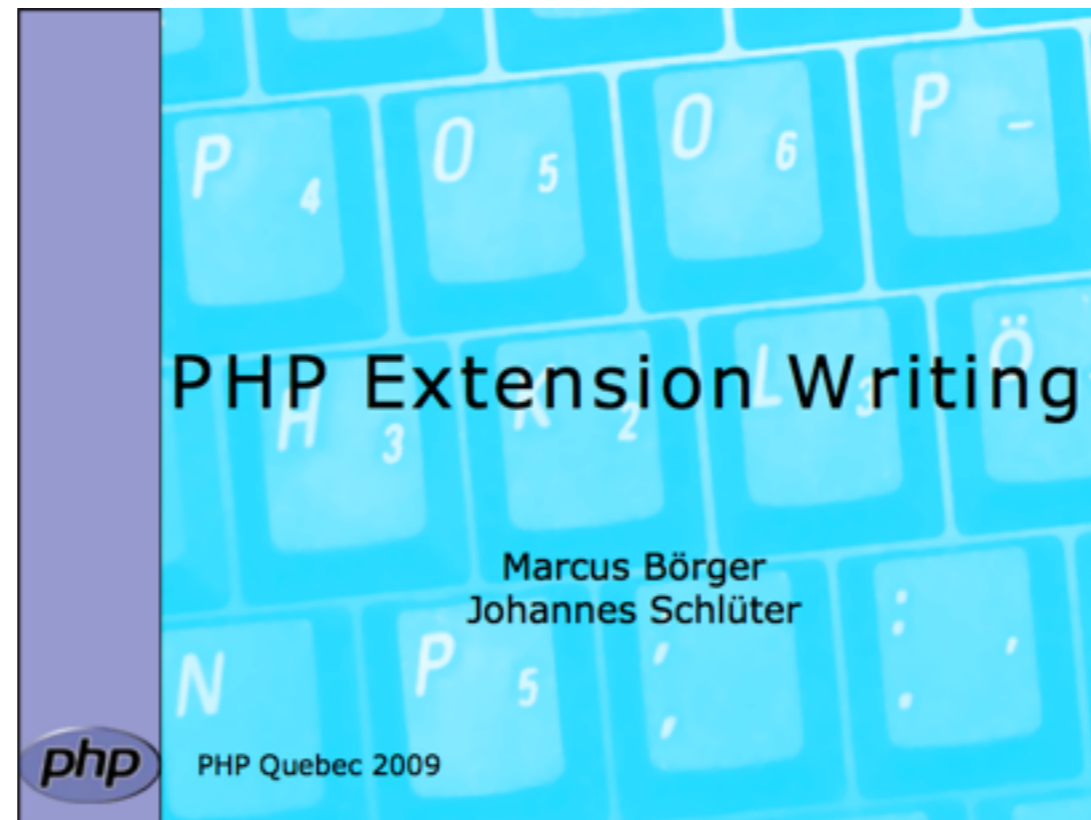
```
$ cd php-x.x.x/ext  
$ ./ext_skel --extname=helloworld
```



- 後は、helloworld.c 内に実行したい処理を作る。(基本はこれだけ)

# 參考資料

- <http://devzone.zend.com/303/extension-writing-part-i-introduction-to-php-and-zend/>
- [http://conf.phpquebec.com/slides/2009/PHP\\_Extension\\_Writing-phpquebec\\_2009.pdf](http://conf.phpquebec.com/slides/2009/PHP_Extension_Writing-phpquebec_2009.pdf)



# PHPモジュール

- zend\_module\_entry

```
zend_module_entry opengl_module_entry = {  
#if ZEND_MODULE_API_NO >= 20010901  
    STANDARD_MODULE_HEADER,  
#endif  
    "OpenGL functions",  
    opengl_functions,  
    PHP_MINIT(opengl),  
    PHP_MSHUTDOWN(opengl),  
    PHP_RINIT(opengl),  
    PHP_RSHUTDOWN(opengl),  
    PHP_MINFO(opengl),  
#if ZEND_MODULE_API_NO >= 20010901  
    NO_VERSION_YET,  
#endif  
    STANDARD_MODULE_PROPERTIES,  
};
```

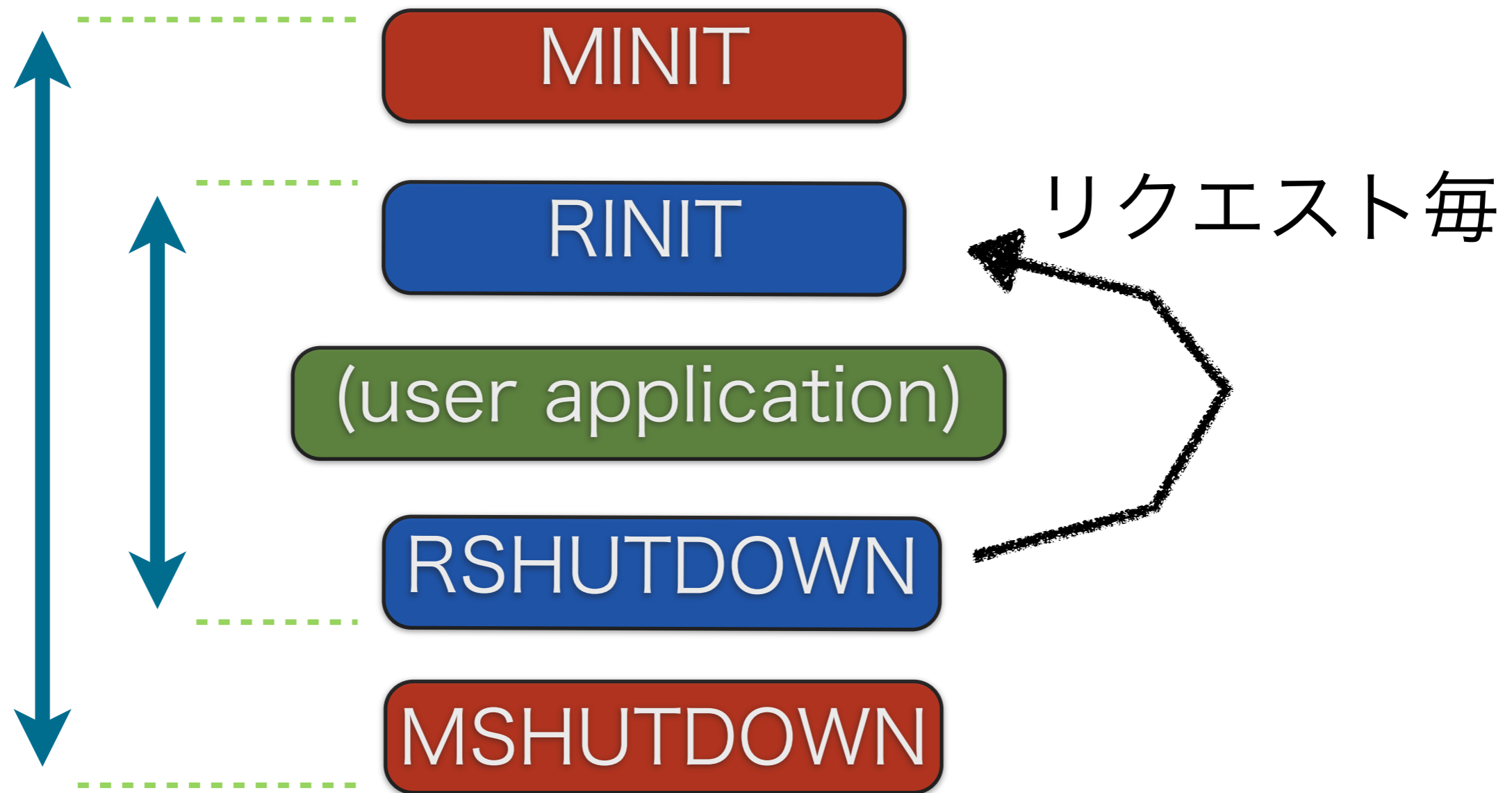
# モジュール初期化

- PHP\_{M|R}\_{INIT|SHUTDOWN}

```
PHP_MINIT(opengl),  
PHP_MSHUTDOWN(opengl),  
PHP_RINIT(opengl),  
PHP_RSHUTDOWN(opengl),
```

- M はモジュール起動で一度だけ
- R はリクエストされる度。

# ライフサイクル



# 関数定義

- zend\_function\_entry (php\_opengl.c)

```
const zend_function_entry opengl_functions[] = {  
    // GL Functions  
    PHP_FE(glaccum, NULL)  
    PHP_FE(glalphafunc, NULL)_
```

- PHP\_FUNCTION

```
// {{{ void glaccum(long op, double value)  
PHP_FUNCTION(glaccum)  
{  
    zval *op, *value;  
    TWO_PARAM(op, value);  
    convert_to_long(op);  
    convert_to_double(value);  
    glAccum((int)Z_LVAL_P(op), (float)Z_DVAL_P(value));  
}
```

# 引数定義 (参照渡し)

- glgetbooleanv

```
PHP_FE(glgetbooleanv, force_ref_second_arg)
```

```
void glGetBooleanv(GLenum pname, GLboolean * params);
```

- ZEND\_BEGIN\_ARG\_INFO\_EX

```
ZEND_BEGIN_ARG_INFO_EX(force_ref_second_arg, 1, 0, 2)  
    ZEND_ARG_INFO(0, arg1)  
    ZEND_ARG_INFO(1, arg2)  
ZEND_END_ARG_INFO()
```

```
function glgetbooleanv(pname, &params)
```



# 配列

- php\_array\_to\_float\_array

```
PHP_FUNCTION(glcolor3fv)
{
    zval *v;
    GLfloat *v_v;
    ONE_PARAM(v);
    convert_to_array(v);
    v_v = php_array_to_float_array(v);
    glColor3fv(v_v);
    efree(v_v);
}
// }}}}
```

- PHP配列をCの配列に変換

# php\_array\_to\_\*\_array

- php\_convert.h

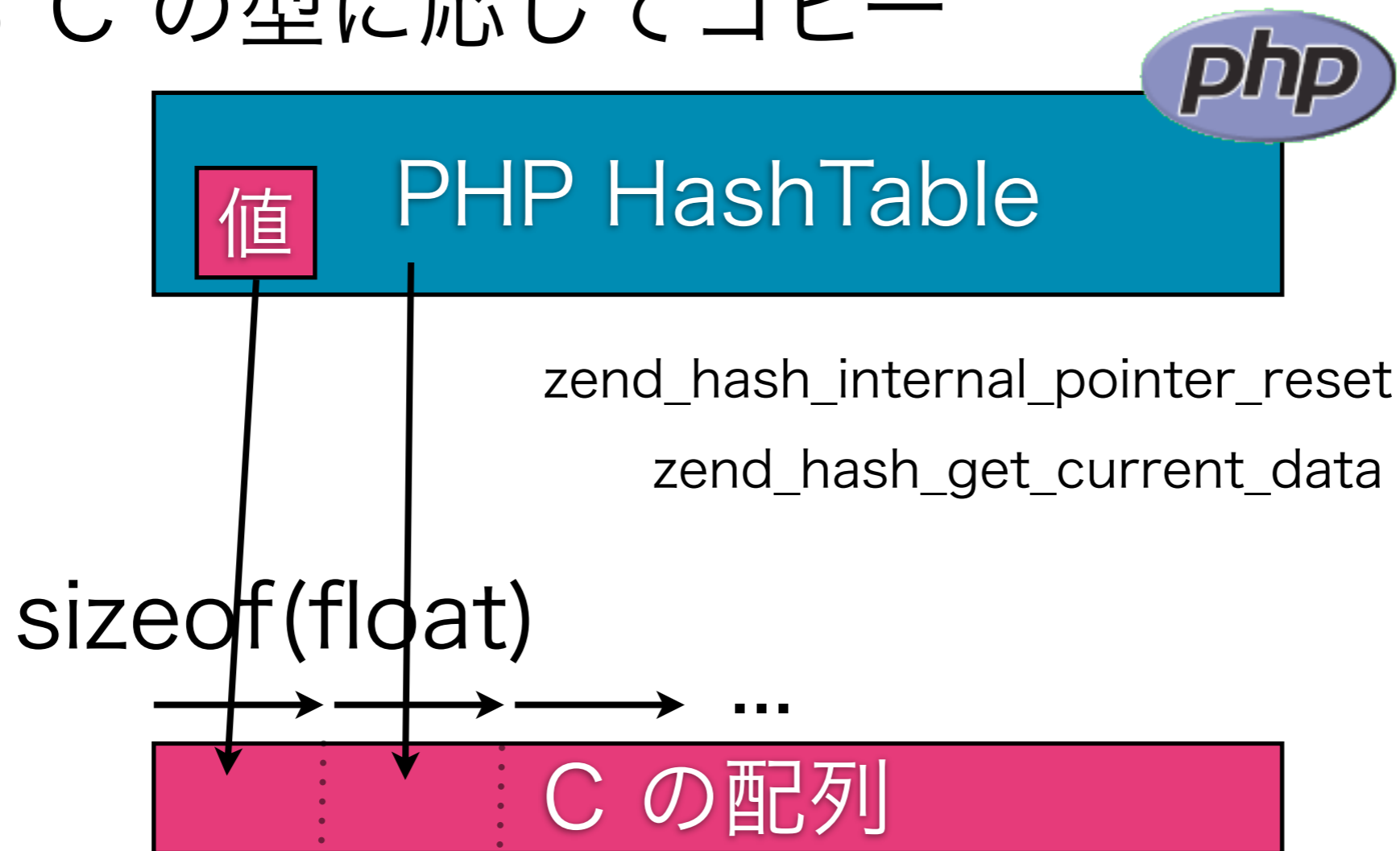
```
#define php_array_to_float_array(z) (float *)php_array_to_c_array(z, TO_C_FLOAT, sizeof(float), NULL)
```

- php\_convert.c

```
zend_hash_internal_pointer_reset(param_ht);  
params = (void *)emalloc(size * tmp_size);  
  
i = 0;  
while(zend_hash_get_current_data(param_ht, (void **)&cur) == SUCCESS)  
{  
    .....  
  
    switch(type)  
    {  
    case TO_C_FLOAT:  
        convert_to_double(*cur);  
        ((float*)params)[i] = (float)Z_DVAL_P(*cur);  
        break;  
    case TO_C_DOUBLE:  
        convert_to_double(*cur);  
        ((double*)params)[i] = Z_DVAL_P(*cur);  
        break;  
    case TO_C_INT:  
        .....  
        break;  
    }
```

# C\_FLOAT の場合

- C の型に応じてコピー



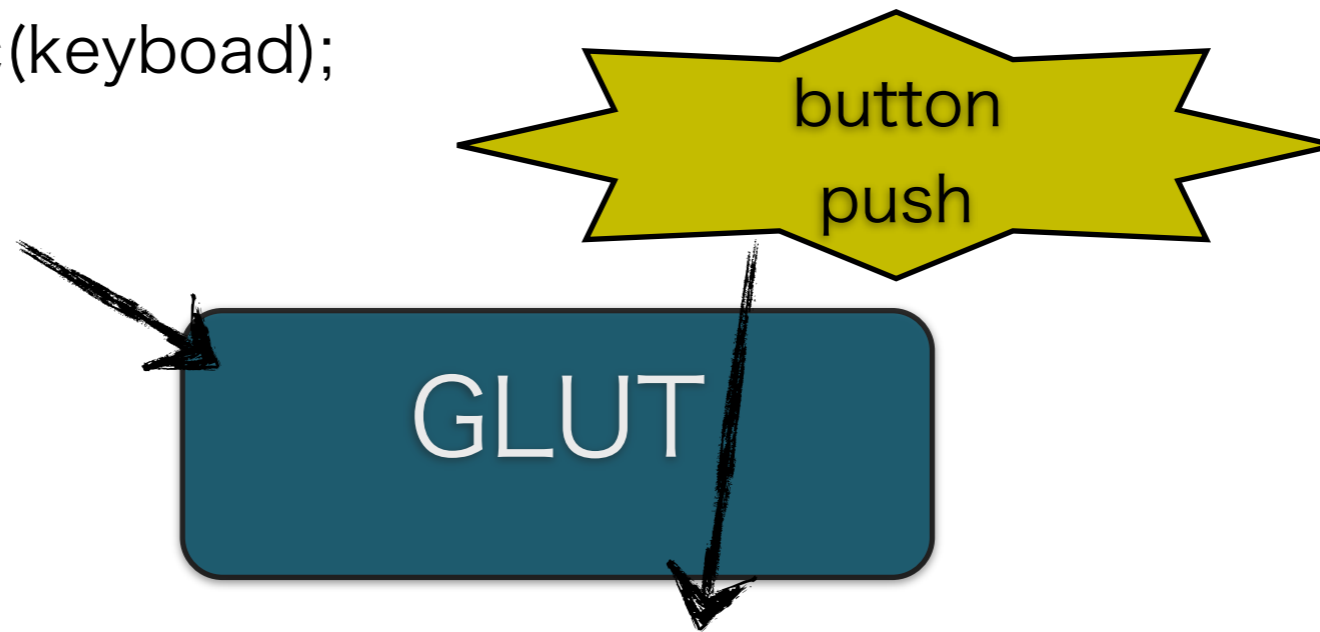
# callback function

- [samples/hajimete/sample05.php](#)

```
function display()  
{  
    glClear(GL_COLOR_BUFFER_BIT); // カラー・バッファを初期化  
    glFlush(); // 命令の実行  
}  
glutDisplayFunc('display'); // コールバック関数登録  
glutMainLoop();  
  
exit (0);
```

# GLUT callback

```
glutKeyboardFunc(keyboard);  
glutMainLoop();
```



```
void keyboard(uchar c, int x, int y)  
{  
    ...  
}
```

# GLUT callback

