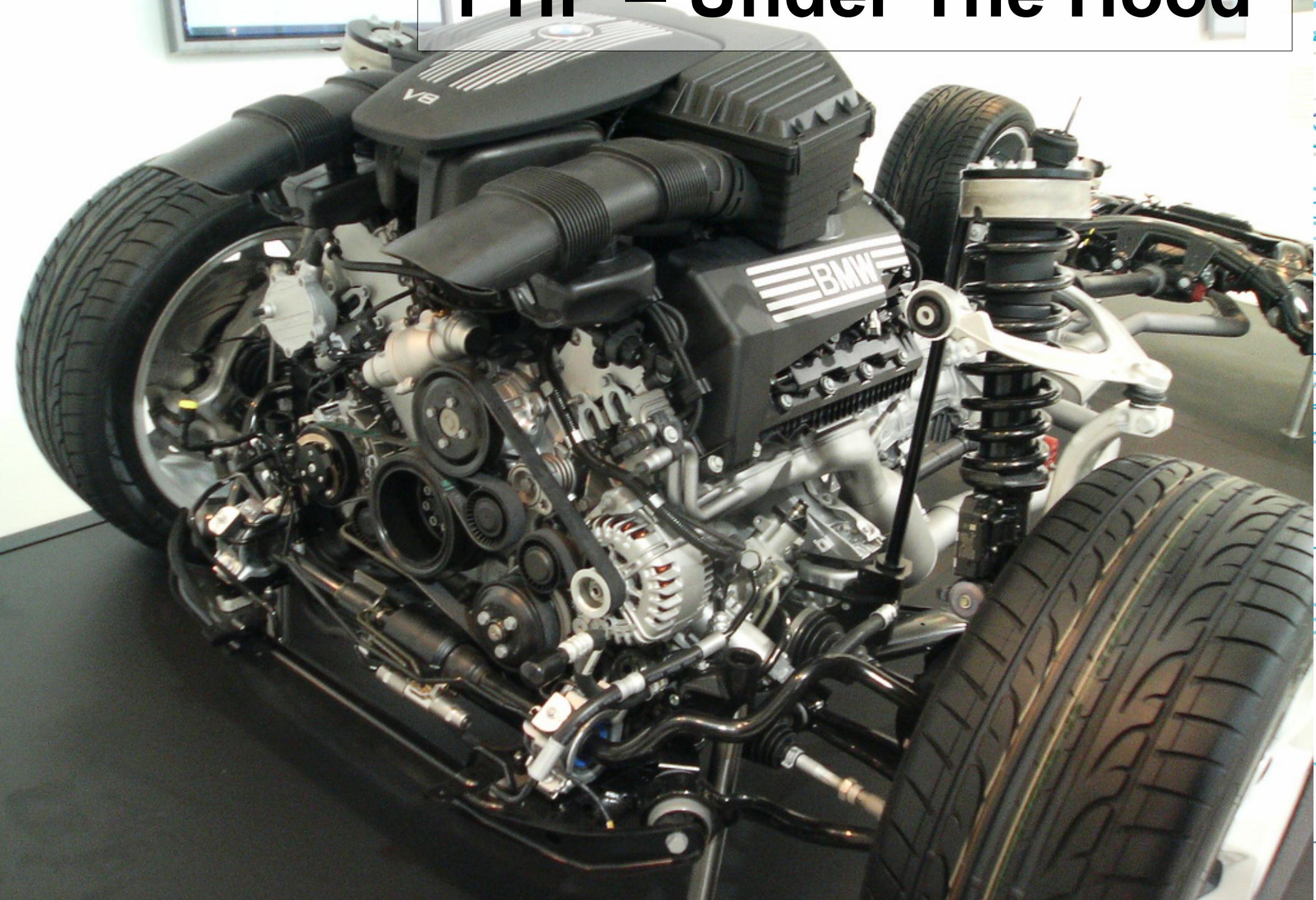


PHP – Under The Hood



Some time ago on IRC

“What is the fastest way to check for a PHP feature?”

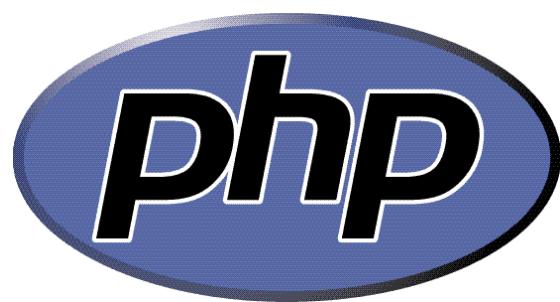
- `extension_loaded("gd");`
- `function_exists("imagecreatefrompng");`



Johannes Schlüter

johannes@schlueters.de
<http://schlueters.de>

Twitter: @phperror



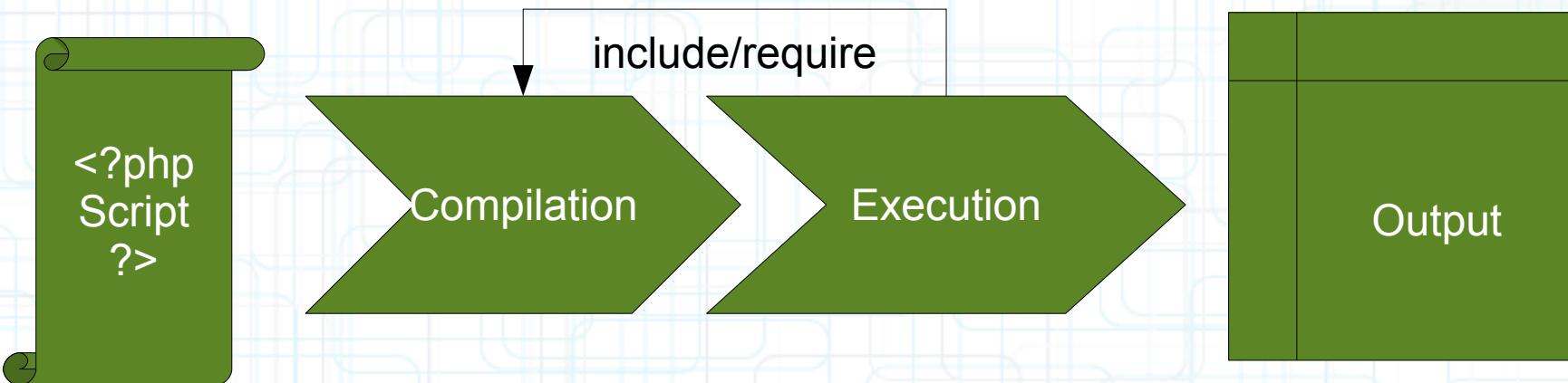
ORACLE®



PHP Lifecycle

- PHP Startup
 - Request startup
 - Script runs
 - Request shutdown
- PHP Shutdown

Script Execution



Compilation

```
<?php  
for ($i = 0; $i < 10; $i++) {  
    echo "Hello world";  
}  
?>
```

line	#	*	op	fetch	ext	return	operands
2	0	>	ASSIGN				!0, 0
	1	>	IS_SMALLER			~1	!0, 10
2		>	JMPZNZ		6		~1, ->8
3	>		POST_INC			~2	!0
4			FREE				~2
5		>	JMP				->1
3	6	>	ECHO				'Hello+world'
4	7		> JMP				->3
6	8	>	> RETURN				1

Function Calls

```
echo Foo(1, FOO, $bar);
```

#	op	ext	return	operands
0	INIT_FCALL_BY_NAME			'foo', 'Foo'
1	SEND_VAL			1
2	FETCH_CONSTANT		~0	'FOO'
3	SEND_VAL			~0
4	SEND_VAR			!0
5	DO_FCALL_BY_NAME	3		
6	ECHO			\$1
7	RETURN			null

Execution

```
int ZEND_ASSIGN_HANDLER(ARGS) { ... }
int ZEND_IS_SMALLER_HANDLER(A) { ... }

...
opcode_handlers handler[] = {
    ZEND_ASSIGN_HANDLER,
    ZEND_IS_SMALLER_HANDLER,
    ...
}

for (opc = start; opc <= end; opc++) {
    handlers[opc]();
}
```

--with-zend-vm=TYPE

- SWITCH

```
while (++o) {
    switch (o) {
        case ASSIGN:
            ...
            break;
        case IS_SMALLER:
            ...
            break;
            ...
    }
}
```

- GOTO

```
ZEND_ASSIGN:
...
    goto oploop;
ZEND_IS_SMALLER:
...
    goto oploop;

while (++o) {
    oploop:
        goto &o;
}
```

**include
require
include_once
require_once**

include vs. include_once

- compile_file();
- execute();
- resolve_path();
- if (seen_before()) {
 return;
}
- open_file();
- if (success) {
 store_seen();
}
- compile();
- execute();

\$

Variables

- Short quiz: How many data types are there in PHP?
 - Null
 - Boolean
 - Integer/Long
 - Float/Double
 - String
 - Array
 - Object
 - Resource



Copies

- `$big_array = array(/* */);`

```
$copy = $big_array;
```

```
function do_something($data) {  
    /* .... */  
}  
do_something($copy);
```

- `$copy[] = 'Awesome new data!';`

References

- `$big_array = array(/* */);`

```
$reference = &$big_array;
```

```
function do_something(&$data) {  
    /* .... */  
}  
do_something($reference);
```

- `$reference[] = 'Awesome new data!';`

Copy on Write

```
$big_array = array( /* ... */ ); ←  
$copy1 = $big_array;  
$copy2 = $big_array;  
$copy2[ ] = 42;
```

zval 1	
refcount	2
is_ref	0

zval 2	
refcount	1
is_ref	0

zval 3	
refcount	0
is_ref	0

References Hurt Performance

- `function do_something(&$data) {
 $size = sizeof($data);
}`

```
$big_array = array(/* .... */);  
do_something($big_array);
```



zval 1	
refcount	2
is_ref	1

zval 2	
refcount	1
is_ref	0

zval 3	
refcount	0
is_ref	0

Yes – References can hurt the performance!



References have side-effects

- `$data = array(0, 1, 2, 3);`

```
foreach ($data as $key => &$item) {}  
foreach ($data as $key => $item) {}  
print_r($data);
```

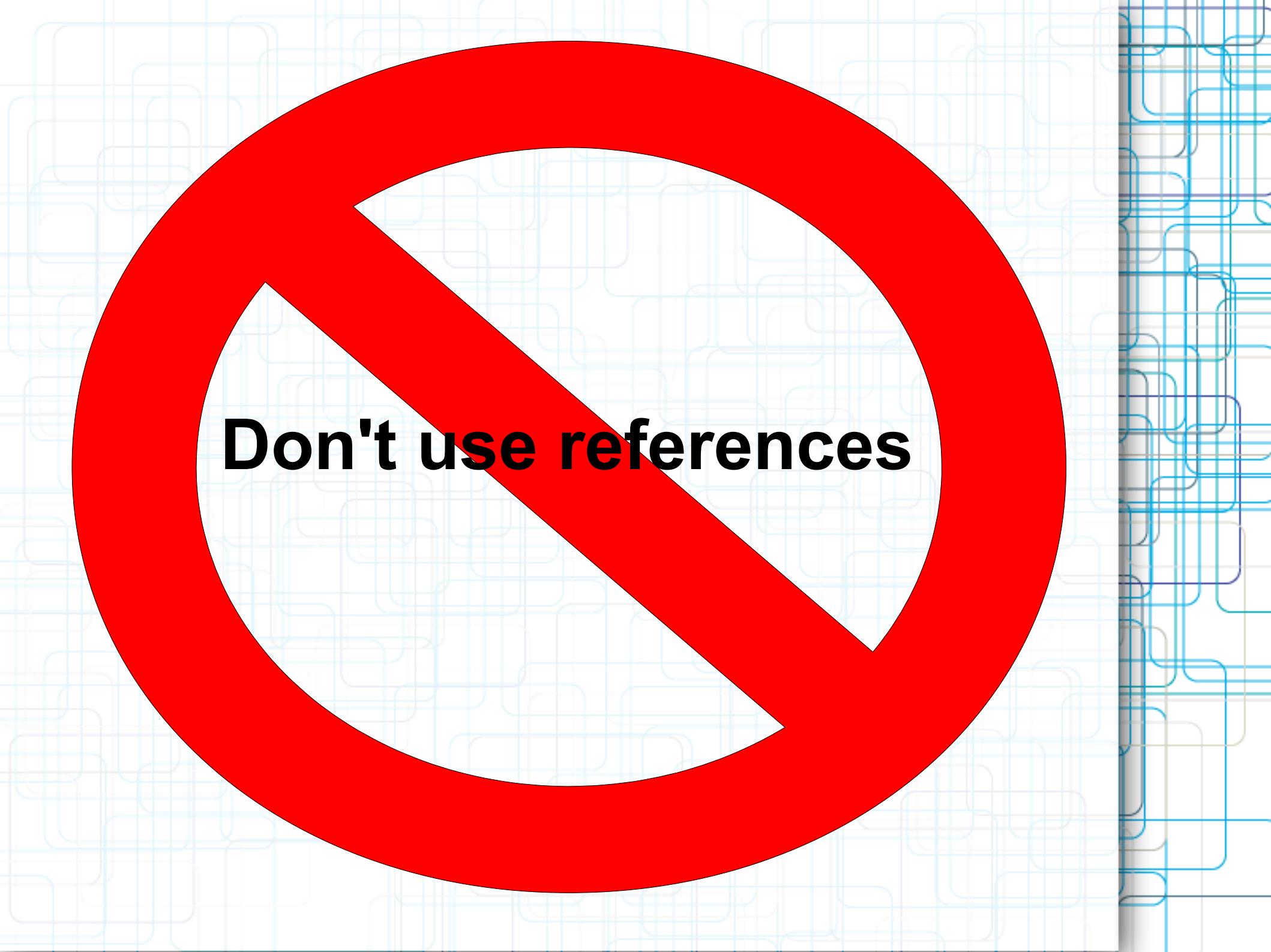
- Array
 - (
 - [0] => 0
 - [1] => 1
 - [2] => 2
 - [3] => 2)

References as cause unintuitive APIs

sort(\$data);

vs.

\$result = sort(\$data);



Don't use references

b..b..b..but ... Objects in PHP 5
are passed by reference!

No – They Are Not



Reference vs. Object Handle

```
function foo(&$var) {  
    $var = null;  
}
```

```
$data = array(/* ... */);  
foo($data);  
print_r($data);
```

NULL

```
function foo($var) {  
    $var = null;  
}
```

```
$data = new stdclass;  
foo($data);  
print_r($data);
```

object(stdClass)#1 (0) {
}

Value vs. Handle

```
function foo($var) {  
    $var[0] = 0;  
}
```

```
$data = array();  
foo($data);  
print_r($data);
```

Array
(
)

```
function foo($var) {  
    $var->foo = 'bar';  
}
```

```
$data = new stdclass;  
foo($data);  
print_r($data);
```

StdClass Object [
 [foo] => bar
]

Memory Management





Cyclic References

```
$a = new stdclass();
```

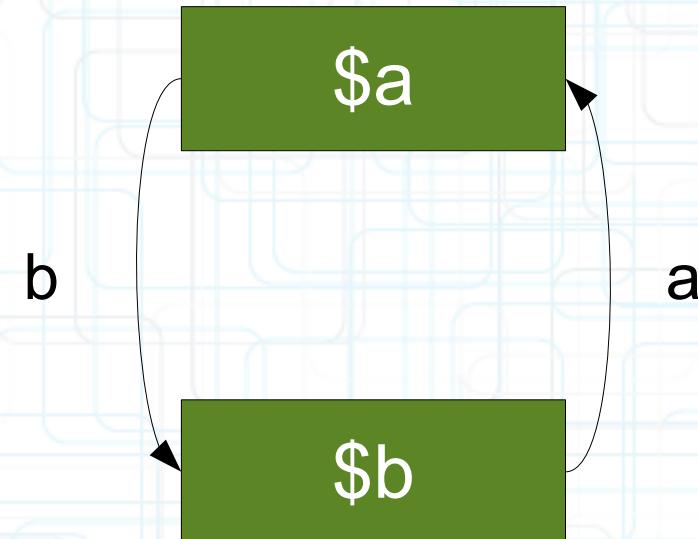
```
$b = new stdclass();
```

```
$b->b = $a;
```

```
$a->a = $b;
```

```
unset($a);
```

```
unset($b);
```







Shutdown sequence

- Call all possible shutdown functions
- Call all possible `__destruct()` functions
- Flush all output buffers
- Send the set HTTP headers
- Call all extensions RSHUTDOWN functions
- Destroy super-globals
- free last error information
- Destroy stream hashes

Now ...

- ... which is faster
 - extension_loaded or
 - function_exists?

```
ZEND_FUNCTION(extension_loaded)
{
    char *extension_name;
    int extension_name_len;
    char *lcname;

    if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC,
        "s", &extension_name, &extension_name_len) == FAILURE) {

        return;
    }

    lcname = zend_str_tolower_dup(extension_name,
                                  extension_name_len);
    if (zend_hash_exists(&module_registry, lcname,
                        extension_name_len+1)) {
        RETVAL_TRUE;
    } else {
        RETVAL_FALSE;
    }
    efree(lcname);
}
```

```
ZEND_FUNCTION(function_exists)
```

```
{
```

```
    char *name;      int name_len;
    zend_function *func;    char *lcname;
    zend_bool retval;
```

```
    if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "s",
        &name, &name_len) == FAILURE) {
        return;
```

```
}
```

```
    lcname = zend_str_tolower_dup(name, name_len);
```

```
    name = lcname;
```

```
    if (lcname[0] == '\\') {
        name = &lcname[1];
        name_len--;
    }
```

```
    retval = (zend_hash_find(EG(function_table), name, name_len+1, (void **)&func) == SUCCESS);
    efree(lcname);
```

```
    if (retval && func->type == ZEND_INTERNAL_FUNCTION &&
        func->internal_function.handler == zif_display_disabled_function) {
        retval = 0;
    }
```

```
    RETURN_BOOL(retval);
```

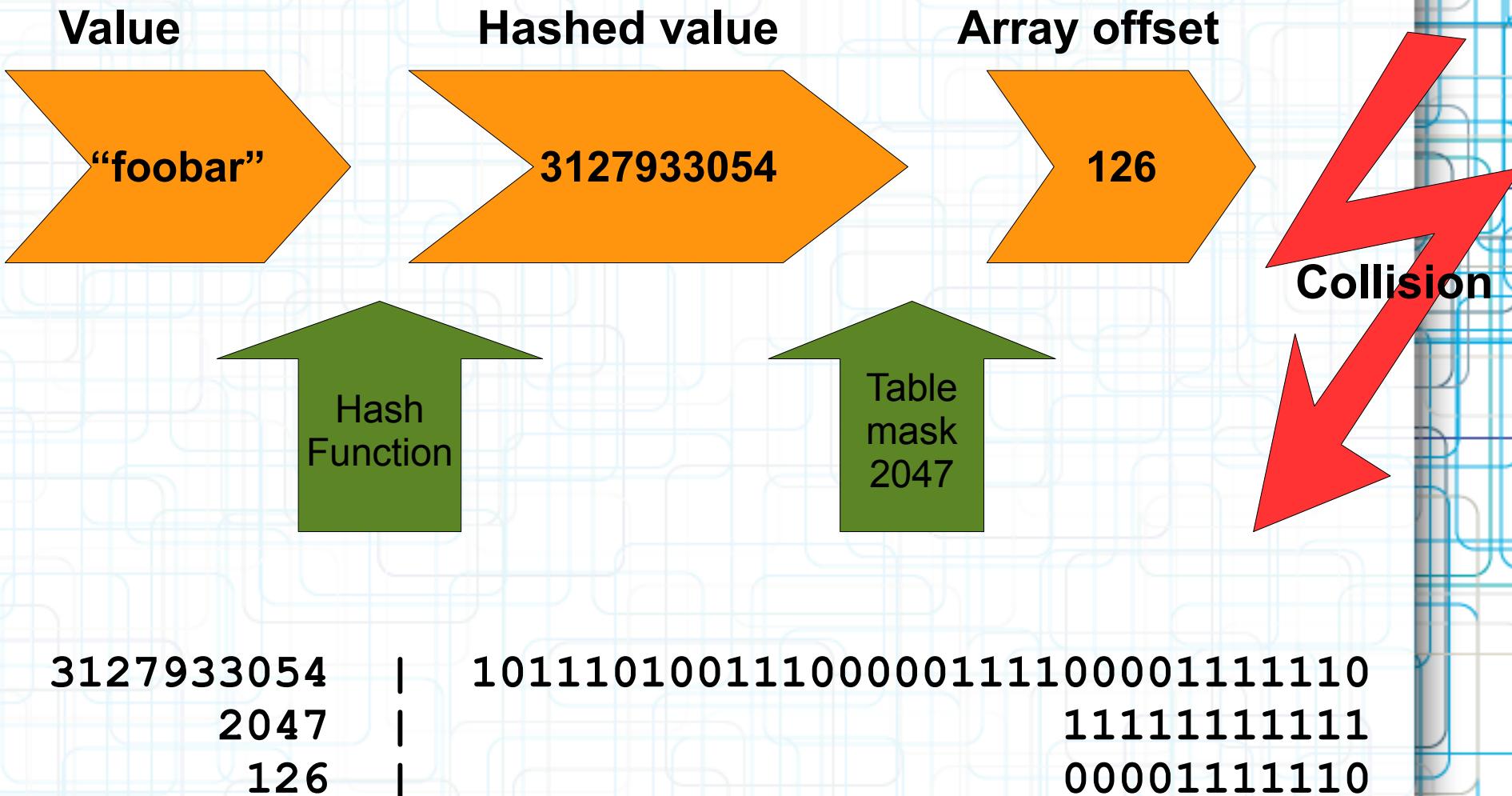
```
}
```

HashTable

- THE Data structure
 - Symbol tables
 - Class table
 - Function Table
 - Extension list
 - PHP Arrays
 - ...



HashTables



Thank You!

Johannes Schlüter
johannes@schlueters.de
<http://schlueters.de>
[@phperror](https://twitter.com/phperror)



<https://joind.in/4967>