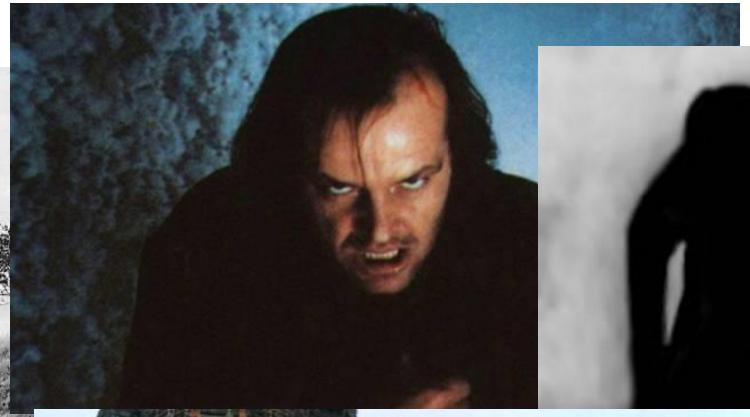
Coding Horrors

A Horror Film Fan's Guide to PHP Coding Nightmares

Coding Horrors





Coding Horrors

Coding Horrors – Friday the 13th

Coding Horrors – Friday the 13th

```
// 3 month appointments
$date_to = date('d/m/Y');
$date_from = date("d/m/Y", strtotime(" +3 months"));

$get = "select * from appointments where date between '$date_from' and '$date_to'";
$get_connect = mysqli_query($con2, $get);
$get_rows = mysqli_num_rows($get_connect);
```

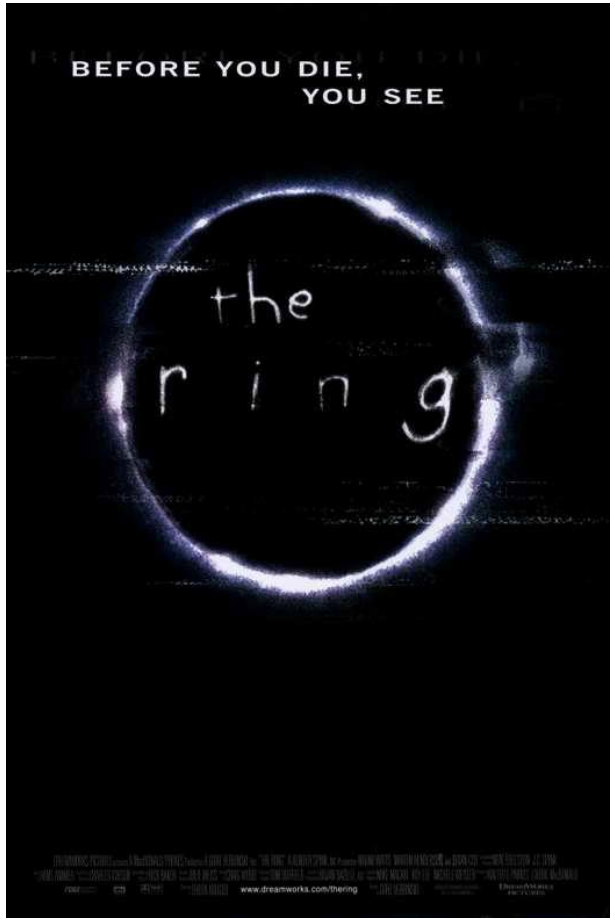
Coding Horrors – Friday the 13th

```
mysql> select * from appointments;
+----+-----------+-----------------------+
| id | date      | appointment           |
+----+-----------+-----------------------+
|  1 | 2/1/2018  | PHPNW January meetup  |
|  2 | 6/2/2018  | PHPNW February Meetup |
|  3 | 6/3/2018  | PHPNW March Meetup    |
|  4 | 3/4/2018  | PHPNW April Meetup    |
|  5 | 1/5/2018  | PHPNW May Meetup      |
|  6 | 5/6/2018  | PHPNW June Meetup     |
|  7 | 3/7/2018  | PHPNW July Meetup     |
|  8 | 7/8/2018  | PHPNW August Meetup   |
|  9 | 4/9/2018  | PHPNW September Meetup |
| 10 | 2/10/2018 | PHPNW October Meetup  |
| 11 | 6/11/2018 | PHPNW November Meetup |
| 12 | 4/12/2018 | PHPNW December Meetup |
+----+-----------+-----------------------+
12 rows in set (0.01 sec)
```

```
mysql> select * from appointments
    -> where date between '2/1/2018' and '5/1/2018';
+----+-----------+-----------------------+
| id | date      | appointment           |
+----+-----------+-----------------------+
|  1 | 2/1/2018  | PHPNW January meetup  |
|  4 | 3/4/2018  | PHPNW April Meetup    |
|  7 | 3/7/2018  | PHPNW July Meetup     |
|  9 | 4/9/2018  | PHPNW September Meetup |
| 10 | 2/10/2018 | PHPNW October Meetup  |
| 12 | 4/12/2018 | PHPNW December Meetup |
+----+-----------+-----------------------+
6 rows in set (0.00 sec)
```

Coding Horrors

Coding Horrors – The Ring



Coding Horrors – The Ring

```
$studentArray = array();
$query1 = "SELECT * FROM students ORDER BY name";
$studentList = mysqli_query($conn,$query1)
    or die ("cannot query the table1: " . mysqli_error($conn));
while ($student = mysqli_fetch_array($studentList)) {
    $studentId = $student['id'];
    $courses = array();
    $query2 = "SELECT * FROM classes WHERE student_id='$studentId'";
    $classList = mysqli_query($conn,$query2)
        or die ("cannot query the table2: " . mysqli_error($conn));
    while ($class = mysqli_fetch_array($classList)) {
        array_push($classes, $class['name']);
    }
    array_push($studentArray, array($studentId, $student['name'], implode(', ', $classes)));
}
```

Coding Horrors – The Ring

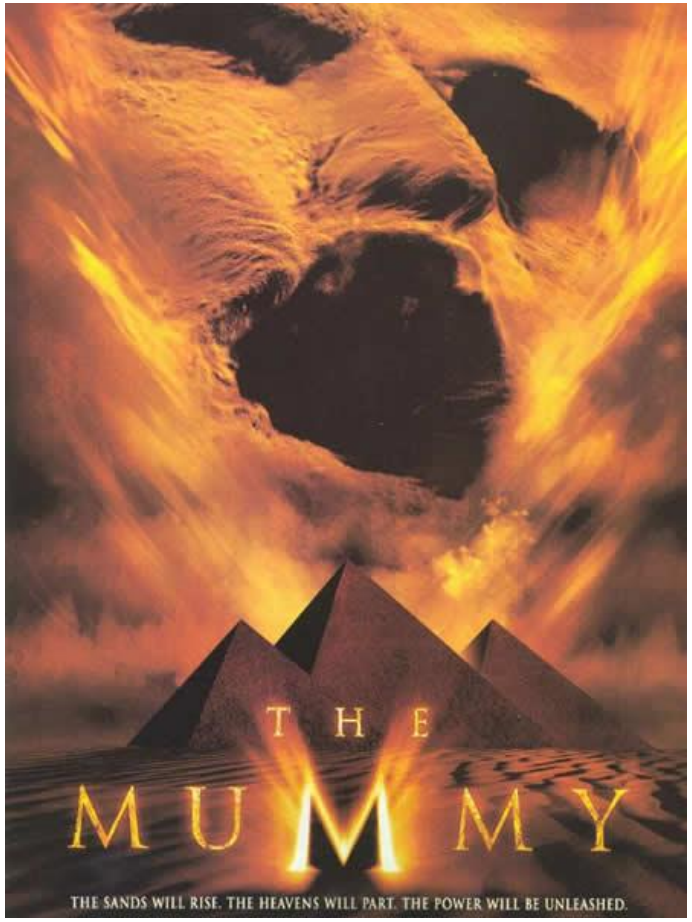
```
SELECT student.id,
       student.name,
       group_concat(class.name separator ', ') as class_names
  FROM students as student
  LEFT JOIN classes as class
         ON class.student_id = student.id
 GROUP BY student.id,
          student.name
 ORDER BY student.name;
```

Coding Horrors – The Ring

```
SELECT student.id,
       student.name,
       class.name as class_name
  FROM students as student
  LEFT JOIN classes as class
         ON class.student_id = student.id
 ORDER BY student.name;
```

Coding Horrors

Coding Horrors – The Mummy

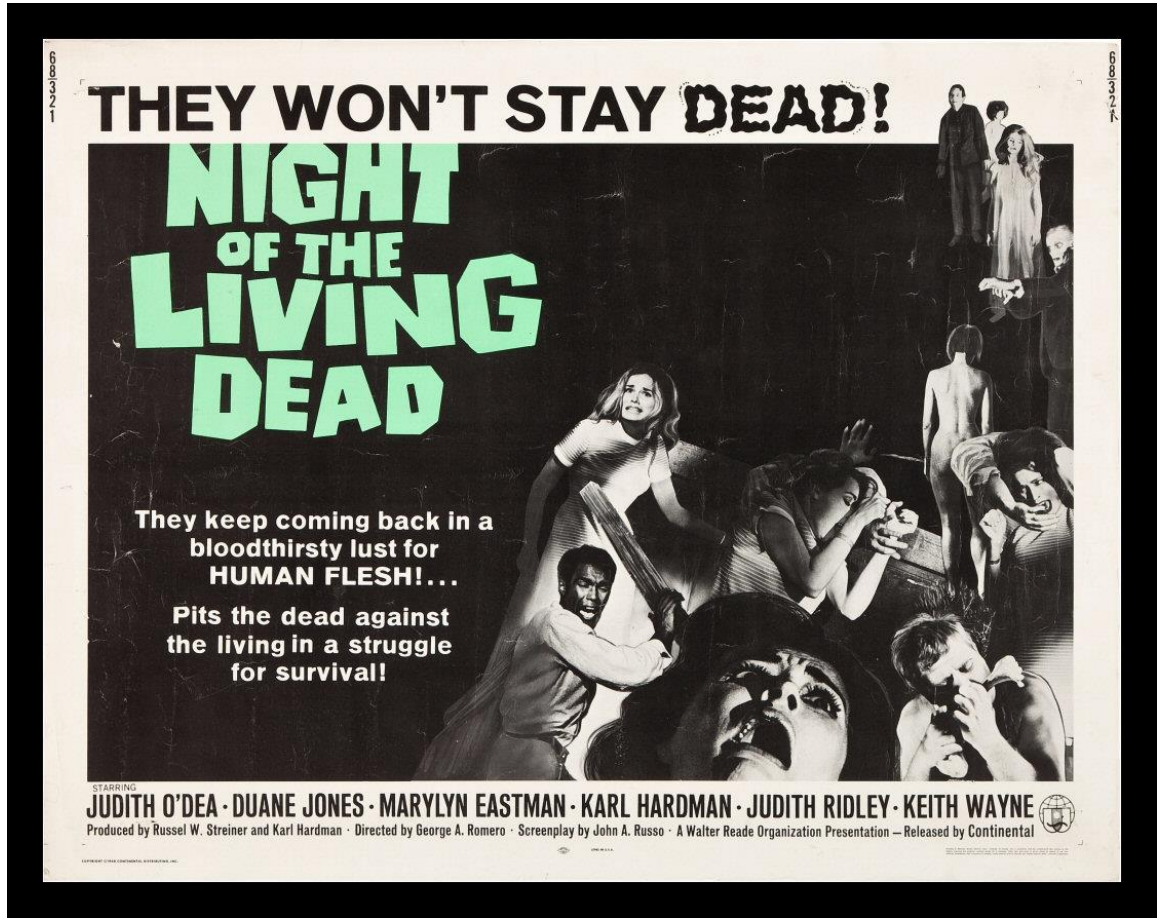


Coding Horrors – The Mummy

```
$Data = $connection->prepare(
    "SELECT * FROM :TableName ORDER BY :OrderByColumn :OrderDirection LIMIT 10 OFFSET :Offset"
);
$Data->bindValue(':TableName', $TableName, PDO::PARAM_STR);
$Data->bindValue(':Offset', $Offset, PDO::PARAM_INT);
$Data->bindValue(':OrderByColumn', $OrderColumn, PDO::PARAM_STR);
$Data->bindValue(':OrderDirection', $OrderDirection, PDO::PARAM_STR);
$Data->execute();
```

Coding Horrors

Coding Horrors – Night of the Living Dead



Coding Horrors – Night of the Living Dead

```
function getAddressDetails() {
    $data->name = 'Mark';
    // $data->address = 'mi casa';
    // $data->city = 'Manchester';
    // $data->country = 'UK';
    // return json_encode($data, JSON_UNESCAPED_UNICODE);
    return json_encode($data);
}
```

Coding Horrors – Night of the Living Dead

```
function getAddressDetails() {
    $data->name = 'Mark';
    return json_encode($data);
}
```

```
git commit -m "Remove address, which is now accessed through a separate API call"
```

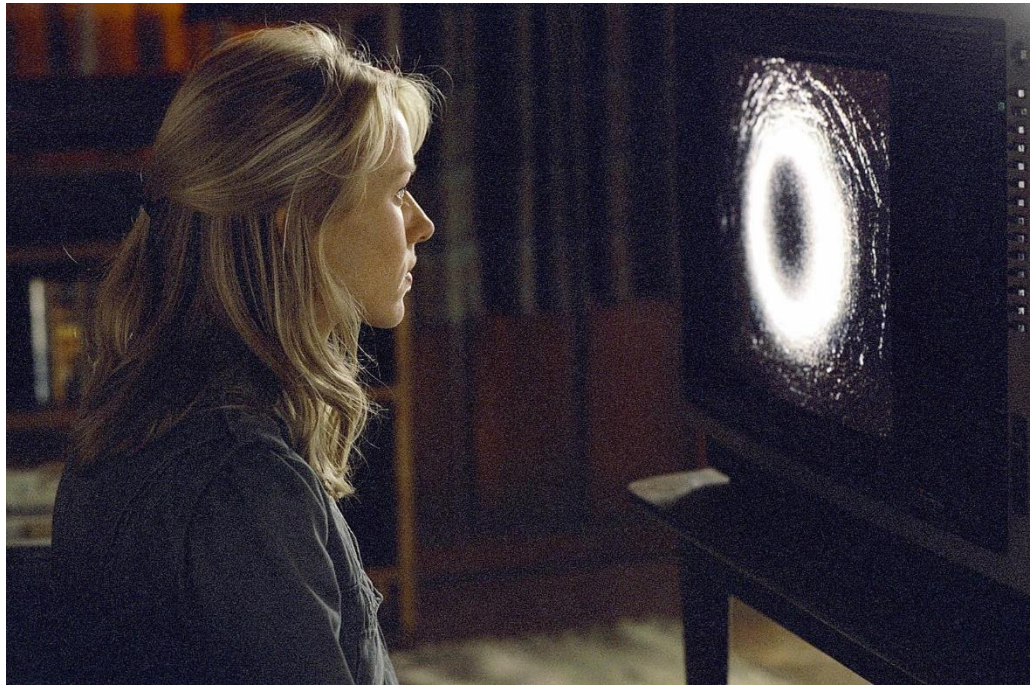
Coding Horrors – Night of the Living Dead

```
function addValues($a, $b, $options) {
    $sum = 0;
    return $a + $b;
}
```

Coding Horrors – Night of the Living Dead

```
function squareIt($bar) {
    return $bar ** 2;
    $bar *= $bar;
    return $bar;
}
```

Coding Horrors



Coding Horrors – The Ring (US Remake)

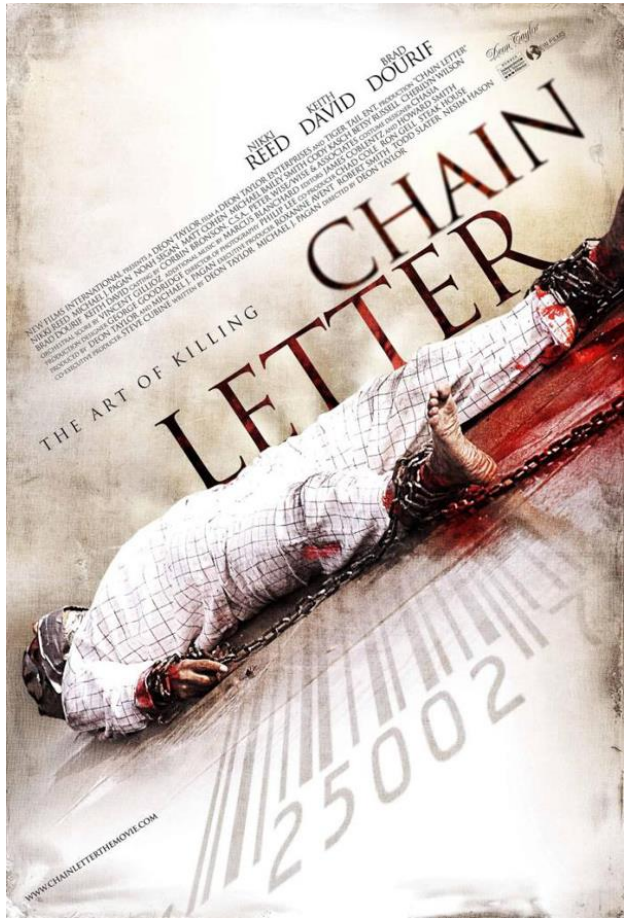
Coding Horrors – The Ring (US Remake)

DRY

Don’t Repeat Yourself

Coding Horrors

Coding Horrors – Chain Letter



Coding Horrors – Chain Letter

```
mysql> select * from users;
+----+----------+----------------------------------+-----------+
| id | username | password                         | followers |
+----+----------+----------------------------------+-----------+
|  1 | Shirley  | 5f4dcc3b5aa765d61d8327deb882cf99 | 8,6,10    |
|  2 | Deborah  | d1133275ee2118be63a577af759fc052 | 4         |
|  3 | Julie    | 25d55ad283aa400af464c76d713c07ad | 4,5       |
|  4 | Jane     | 4297f44b13955235245b2497399d7a93 | 3,5       |
|  5 | Jennifer | d8578edf8458ce06fbc5bb76a58c5ca4 | 3,4       |
|  6 | Alison   | 827ccb0eea8a706c4c34a16891f84e7b | 9,1       |
|  7 | Philipa  | 8621ffdbc5698829397d97767ac13db3 | NULL      |
|  8 | Sue      | acc6f2779b808637d04c71e3d8360eeb | 1         |
|  9 | Annabel  | 0d107d09f5bbe40cade3de5c71e9e9b7 | 6         |
| 10 | Cathy    | 5badcaf789d3d1d09794d8f021f40f0e | 3,1       |
+----+----------+----------------------------------+-----------+
10 rows in set (0.00 sec)
```

Coding Horrors – Chain Letter

```
mysql> select * from users where followers like '%1%';
+----+----------+----------------------------------+-----------+
| id | username | password                         | followers |
+----+----------+----------------------------------+-----------+
|  1 | Shirley  | 5f4dcc3b5aa765d61d8327deb882cf99 | 8,6,10    |
|  6 | Alison   | 827ccb0eea8a706c4c34a16891f84e7b | 9,1       |
|  8 | Sue      | acc6f2779b808637d04c71e3d8360eeb | 1         |
| 10 | Cathy    | 5badcaf789d3d1d09794d8f021f40f0e | 3,1       |
+----+----------+----------------------------------+-----------+
4 rows in set (0.00 sec)
```

Coding Horrors – Chain Letter

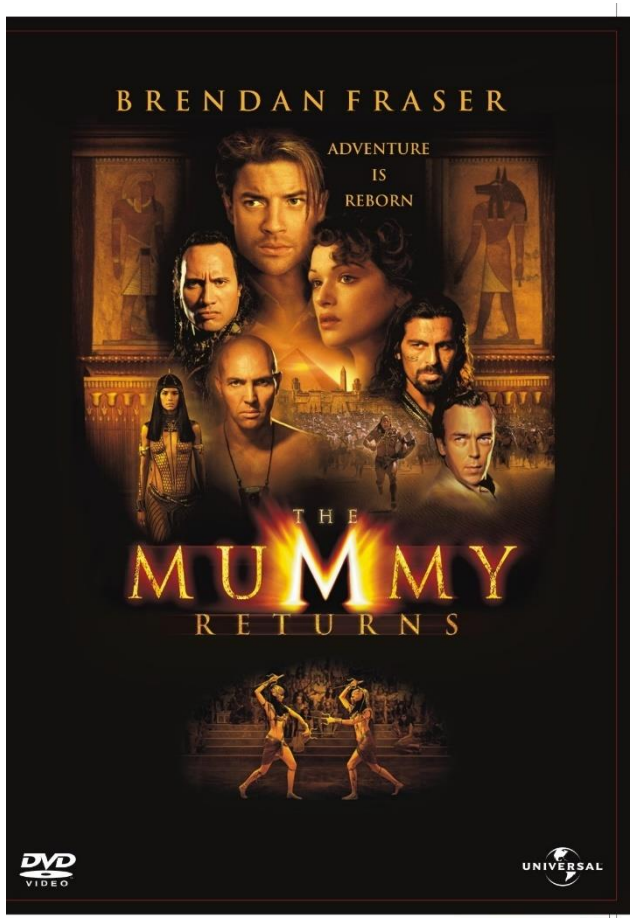
```
mysql> select * from users where followers like '%,1,%';
Empty set (0.00 sec)

mysql> select * from users
    -> where followers = '1' or followers like '1,%' or followers like '%,1,%' or followers like '%,1';
+----+----------+----------------------------------+---------+
| id | username | password                         | friends |
+----+----------+----------------------------------+---------+
|  6 | Alison   | 827ccb0eea8a706c4c34a16891f84e7b | 9,1     |
|  8 | Sue      | acc6f2779b808637d04c71e3d8360eeb | 1       |
| 10 | Cathy    | 5badcaf789d3d1d09794d8f021f40f0e | 3,1     |
+----+----------+----------------------------------+---------+
3 rows in set (0.00 sec)
```

Coding Horrors – Chain Letter

```
mysql> select * from users where find_in_set(1,friends);
+----+----------+----------------------------------+---------+
| id | username | password                         | friends |
+----+----------+----------------------------------+---------+
|  6 | Alison   | 827ccb0eea8a706c4c34a16891f84e7b | 9,1     |
|  8 | Sue      | acc6f2779b808637d04c71e3d8360eeb | 1       |
| 10 | Cathy    | 5badcaf789d3d1d09794d8f021f40f0e | 3,1     |
+----+----------+----------------------------------+---------+
3 rows in set (0.00 sec)
```

Coding Horrors

Coding Horrors – The Mummy Returns



Coding Horrors – The Mummy Returns

```
$q = "SELECT id, name FROM test WHERE name like '%:foo%'";
$s = "carrot";
$sth = $dbh->prepare($q);
$sth->bindParam(':foo', $s);
$sth->execute();
```

Coding Horrors – The Mummy Returns

```
$q = "SELECT id, name FROM test WHERE name like :foo";
$s = "carrot";
$s = "%{$s}%";
$sth = $dbh->prepare($q);
$sth->bindParam(':foo', $s);
$sth->execute();
```

Coding Horrors – The Mummy Returns

```
$q = "SELECT * FROM posts WHERE post_title LIKE :q OR post_text LIKE :q";
$s = "carrot";
$s = "%{$s}%";
$sth = $dbh->prepare($q);
$sth->bindParam(':q', $s);
$sth->execute();
```

Coding Horrors – The Mummy Returns

```
$q = "SELECT * FROM posts WHERE post_title LIKE :q1 OR post_text LIKE :q2";
$s = "carrot";
$s = "%{$s}%";
$sth = $dbh->prepare($q);
$sth->bindParam(':q1', $s);
$sth->bindParam(':q2', $s);
$sth->execute();
```

Coding Horrors

Coding Horrors – Halloween

Coding Horrors – Halloween

```php
function monthList() {
    $startDate = new DateTime();
    $endDate = (clone $startDate)
        ->add(new DateInterval('P1Y'));
    $dateRange = new DatePeriod($startDate, new DateInterval('P1M'), $endDate);
    foreach($dateRange as $date) {
        yield $date;
    }
}

foreach(monthList() as $date) {
    echo $date->format("M Y") . PHP_EOL;
}
```

Coding Horrors – Halloween

```
Jan 2018
Feb 2018
Mar 2018
Apr 2018
May 2018
Jun 2018
Jul 2018
Aug 2018
Sep 2018
Oct 2018
Nov 2018
Dec 2018
```

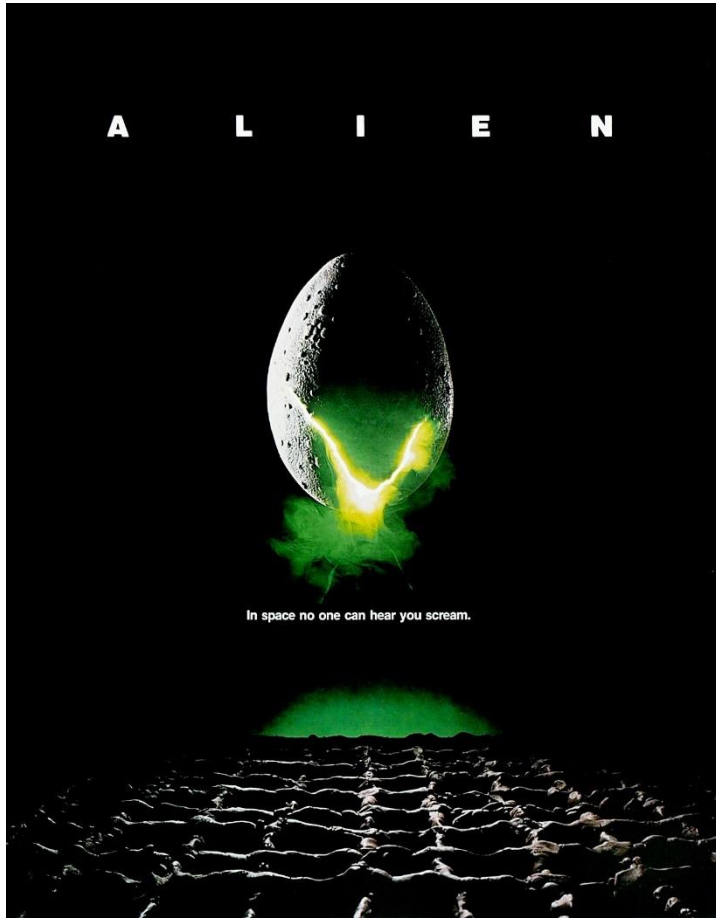
```
Jan 2018
Mar 2018
Apr 2018
May 2018
Jun 2018
Jul 2018
Aug 2018
Sep 2018
Oct 2018
Nov 2018
Dec 2018
Jan 2019
```

Coding Horrors – Halloween

```php
function monthList() {
    $startDate = new DateTime('first day of this month');
    $endDate = (clone $startDate)
        ->add(new DateInterval('P1Y'));
    $dateRange = new DatePeriod($startDate, new DateInterval('P1M'), $endDate);
    foreach($dateRange as $date) {
        yield $date;
    }
}

foreach(monthList() as $date) {
    echo $date->format("M Y") . PHP_EOL;
}
```

Coding Horrors

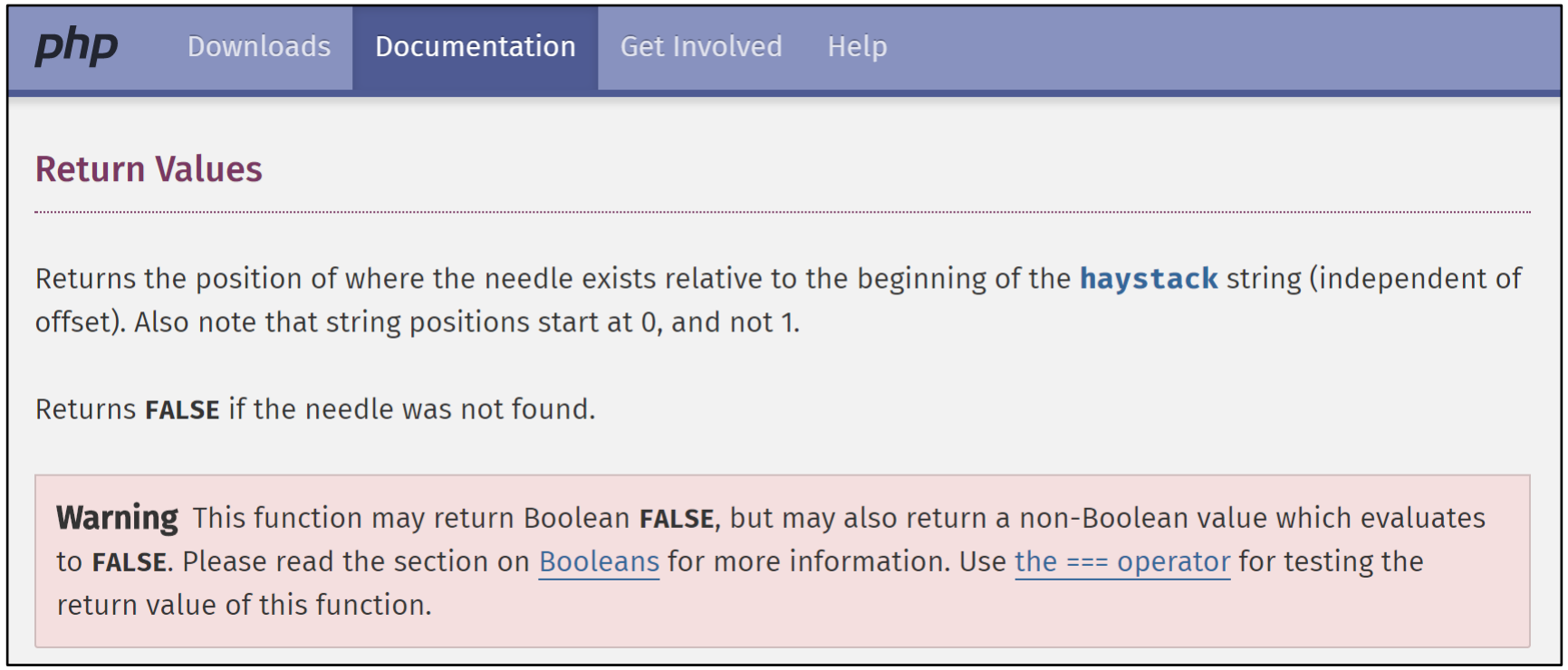
Coding Horrors – Alien



Coding Horrors – Alien

```
if (strpos(" " . $fileURL, "s3://") >= 1) {
    // Do something
}
```

Coding Horrors – Alien



php Downloads Documentation Get Involved Help

Return Values

Returns the position of where the needle exists relative to the beginning of the **haystack** string (independent of offset). Also note that string positions start at 0, and not 1.

Returns **FALSE** if the needle was not found.

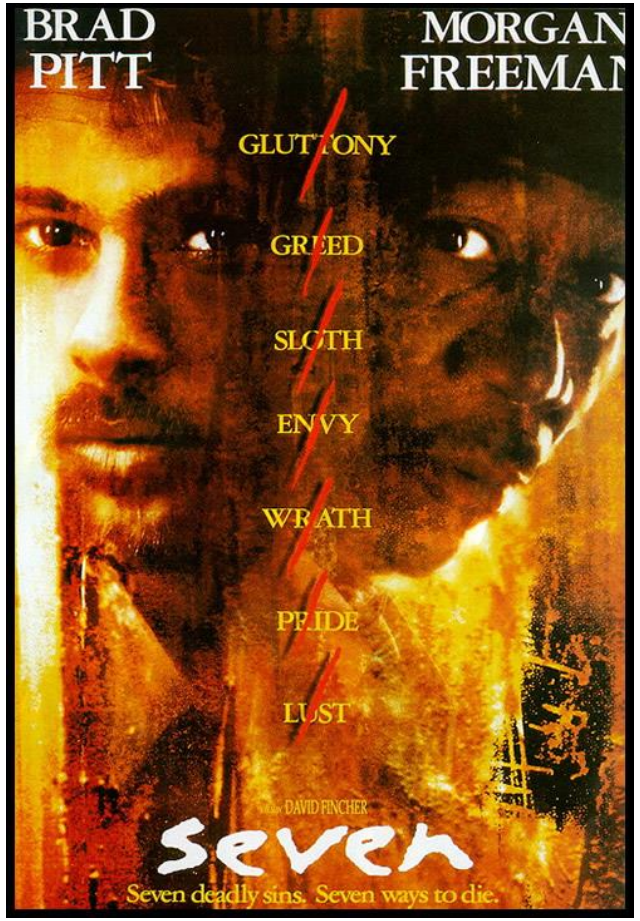
Warning This function may return Boolean **FALSE**, but may also return a non-Boolean value which evaluates to **FALSE**. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.

Coding Horrors – Alien

```
if (strpos($fileURL, "s3://") !== false) {
    // Do something
}
```

Coding Horrors

Coding Horrors – Se7en



Coding Horrors – Se7en

```php
$calendar = [
    01 => 'January',
    02 => 'February',
    03 => 'March',
    04 => 'April',
    05 => 'May',
    06 => 'June',
    07 => 'July',
    08 => 'August',
    09 => 'September',
    10 => 'October',
    11 => 'November',
    12 => 'December',
];
```

Coding Horrors - Se7en

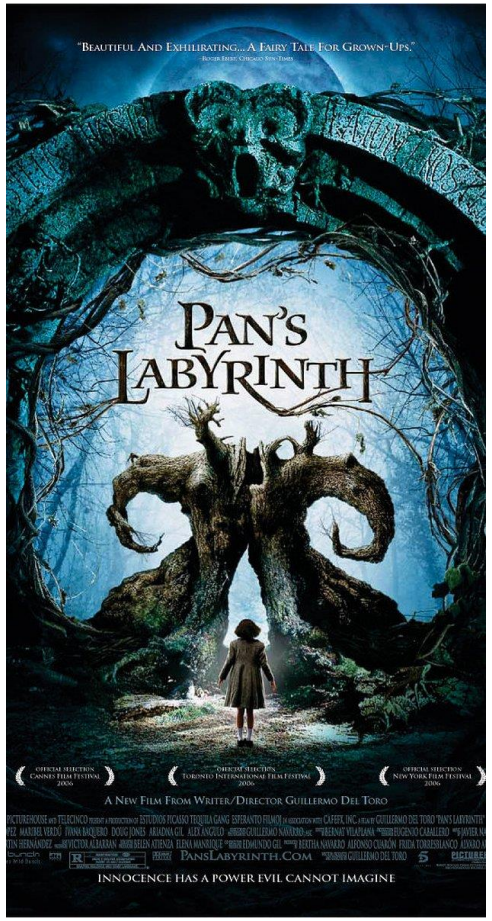
```
1 => January
2 => February
3 => March
4 => April
5 => May
6 => June
7 => July
0 => September
10 => October
11 => November
12 => December
```

Coding Horrors – Se7en

```php
$calendar = [
    1 => 'January',
    2 => 'February',
    3 => 'March',
    4 => 'April',
    5 => 'May',
    6 => 'June',
    7 => 'July',
    8 => 'August',
    9 => 'September',
    10 => 'October',
    11 => 'November',
    12 => 'December',
];
```

Coding Horrors

Coding Horrors – Pan’s Labyrinth



Coding Horrors – Pan's Labyrinth

```
public function __construct(
    \Magento\Framework\Model\Context $context,
    \Magento\Framework\View\DesignInterface $design,
    \Magento\Framework\Registry $registry,
    \Magento\Store\Model\App\Emulation $appEmulation,
    \Magento\Store\Model\StoreManagerInterface $storeManager,
    \Magento\Framework\App\RequestInterface $request,
    \Magento\Newsletter\Model\Template\Filter $filter,
    \Magento\Framework\App\Config\ScopeConfigInterface $scopeConfig,
    \Magento\Newsletter\Model\TemplateFactory $templateFactory,
    \Magento\Framework\Filter\FilterManager $filterManager,
    array $data = []
) {
    parent::__construct($context, $design, $registry, $appEmulation, $storeManager, $data);
    $this->_storeManager = $storeManager;
    $this->_request = $request;
    $this->_filter = $filter;
    $this->_scopeConfig = $scopeConfig;
    $this->_templateFactory = $templateFactory;
    $this->_filterManager = $filterManager;
}
```

Coding Horrors – Pan's Labyrinth

The ideal number of arguments for a function is zero (niladic). Next comes one (monadic) followed closely by two (dyadic). Three arguments (triadic) should be avoided where possible. More than three (polyadic) requires very special justification—and then shouldn't be used anyway.

Bob Martin – "Clean Code"

Coding Horrors

Coding Horrors – Scream

Coding Horrors – Scream

```
$sql = "SELECT MAX(id) AS max_page FROM videos";
$result = @$conn->query($sql);
$row = @mysql_fetch_array($result);
echo $row["max_page"];
```

Coding Horrors – Scream

```
@dns_get_record();

dns_get_record();
```

Coding Horrors – Scream

```
Warning: dns_get_record() expects at least 1 parameter, 0 given in
%filename% on %line%
```

Coding Horrors – Scream

```
set_error_handler(function($errno, $errstr, $errfile, $errline, array $errcontext) {
    if (error_reporting() === 0) {
        // error was suppressed with the @-operator
        throw new \ErrorException(
            'WHY ARE YOU SUPPRESSING ERRORS RATHER THAN HANDLING THEM!'
        );
        // return false;
    }
    throw new \ErrorException($errstr, 0, $errno, $errfile, $errline);
});
```

Coding Horrors – Scream

```php
try {
    @dns_get_record();
} catch (\ErrorException $e) {
    echo 'EXCEPTION: ', $e->getMessage(), PHP_EOL;
}

try {
    dns_get_record();
} catch (\ErrorException $e) {
    echo 'EXCEPTION: ', $e->getMessage(), PHP_EOL;
}
```

Coding Horrors – Scream

```
EXCEPTION: WHY ARE YOU SUPPRESSING ERRORS RATHER THAN HANDLING THEM?

EXCEPTION: dns_get_record() expects at least 1 parameter, 0 given
```

Coding Horrors – Scream

Error Control Operators

PHP supports one error control operator: the at sign (@). When prepended to an expression in PHP, any error messages that might be generated by that expression will be ignored.

If you have set a custom error handler function with set_error_handler() then it will still get called, but this custom error handler can (and should) call error_reporting() which will return 0 when the call that triggered the error was preceded by an @.

If the **track_errors** feature is enabled, any error message generated by the expression will be saved in the variable `$php_errormsg`. This variable will be overwritten on each error, so check early if you want to use it.

Coding Horrors – Scream

- Scream
  - PECL Extensions

- Disables the @ error control operator so that all errors are reported.

```
ini_set('scream.enabled', true);
```

Coding Horrors

Coding Horrors – Dawn of the Dead

Coding Horrors – Dawn of the Dead

```
/**
 * Convert a date from PHP to Excel
 *
 * @param   mixed   $dateValue  unix timestamp or datetime object
 * @param   string  $timezone   Optional timezone name for adjustment from UTC
 * @return  mixed   Excel date/time value
 **/
public function PHPToExcel(DateTimeImmutable $dateValue, DateTimeZone $timezone = null) {
     // Do stuff
}
```

Coding Horrors

Coding Horrors

Who am I?

Mark Baker

Senior Software Engineer
MessageBird BV, Amsterdam

Coordinator and Developer of:

Open Source PHPOffice library
PHPExcel, PHPWord, PHPPowerPoint, PHPProject, PHPVisio
Minor contributor to PHP core (SPL Datastructures)
Other small open source libraries available on github

@Mark_Baker

https://github.com/MarkBaker

http://uk.linkedin.com/pub/mark-baker/b/572/171

http://markbakeruk.net