



Software Engineering





Plan

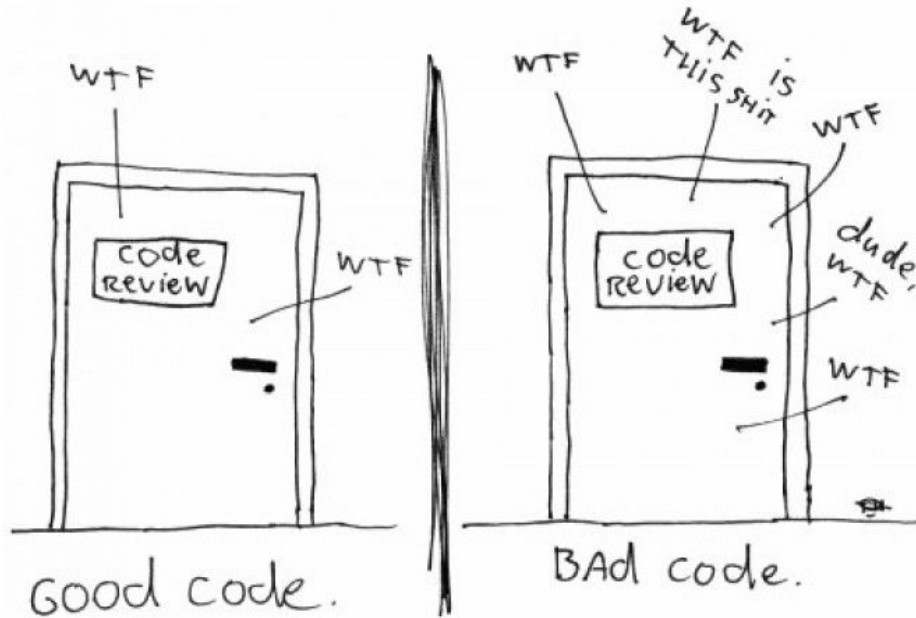
- ◇ Clean Code
- ◇ Motivation
- ◇ Writing Clean Code
- ◇ Clean Code best practice
 - Commenting & Documentation
 - Consistent Indentation
 - DRY Principle
 - etc.





Clean Code

The ONLY valid measurement
of code quality: WTFs/minute



(c) 2008 Focus Shift



Clean Code Importance

- ◇ Your target audience is other programmers (and yourself) not the computer.
- ◇ The ratio of time spent by a programmer reading code to writing code is normally 10 to 1.
- ◇ You are constantly reading old code in order to write new code.





Writing Clean Code



Naming Conventions



Naming Conventions

Identifier	Naming Rules	Example
Variables	A short, but meaningful, name that communicates to the casual observer what the variable represents, rather than how it is used. Begin with a lowercase letter and use camel case (mixed case, starting with lower case).	mass hourlyWage isPrime
Constant	Use all capital letters, separating internal words with the underscore character.	BOLTZMANN MAX_HEIGHT
Class	A noun that communicates what the class represents. Begin with an uppercase letter and use camel case for internal words.	Complex Charge PhoneNumber
Method	A verb that communicates what the method does. Begin with a lowercase letter and use camelCase for internal words.	move() draw() enqueue()





Functions



Functions

- ◇ The smaller the better
- ◇ A function should only do one thing
- ◇ Less arguments are better
- ◇ Error Handling is one thing
- ◇ Throwing exceptions is better than returning different codes dependent on errors.
- ◇ Use try/catch instead of conditions if possible
- ◇ Don't repeat yourself; functions must be atomic





General Guidelines



Clean Code

- ◇ Naming convention
- ◇ Documentation
- ◇ Code grouping
- ◇ Code indentation
- ◇ DRY principle
- ◇ Line length
- ◇ File & Folder organization





Clean Code

- ◇ Dead code
- ◇ Large classes
- ◇ Multiple languages in one file
- ◇ Long if conditions - replace with function
- ◇ Overuse of static
- ◇ Circular references
- ◇ Hard-coding





Clean Code best practice



Commenting & Documentation



```
test.php X
1  <?php
2
3
4  $manager = Doctrine_Manager::getInstance();
5
6  $conn = $manager->getConnection('default');
```

getConnection(\$name)
Get the connection instance for the passed name

@param name String name of the connection, if empty numeric key is used

@return Doctrine_Connection

Resolved return types: Doctrine_Connection

Press 'F1' for help - Press 'F2' for focus


```
17
18

Manager.php X
445  /**
446   * Get the connection instance for the passed name
447   *
448   * @param string $name          name of the connect
449   * @return Doctrine_Connection
450   * @throws Doctrine_Manager_Exception if trying to get a
451   */
452  public function getConnection($name)
453  {
454      if ( ! isset($this->_connections[$name])) {
455          throw new Doctrine_Manager_Exception('Unknown conne
456      }
457
458      return $this->_connections[$name];
459  }
460
```





Consistent Indentation



```
1 function foo()
2 {   if ($maybe)
3     {   do_it_now();
4         again();
5     }
6     else
7     {   abort_mission();
8     }
9     finalize();
10 }
```

```
1 function foo() {
2     if ($maybe) {
3         do_it_now();
4         again();
5     } else {
6         abort_mission();
7     }
8     finalize();
9 }
```

```
1 function foo()
2 {
3     if ($maybe)
4     {
5         do_it_now();
6         again();
7     }
8     else
9     {
10        abort_mission();
11    }
12    finalize();
13 }
```



Avoid Obvious Comments



```
1 // get the country code
2 $country_code = get_country_code($_SERVER['REMOTE_ADDR']);
3
4 // if country code is US
5 if ($country_code == 'US') {
6
7     // display the form input for state
8     echo form_input_state();
9 }
```





Code Grouping



```
1 // get list of forums
2 $forums = array();
3 $r = mysql_query("SELECT id, name, description FROM forums");
4 while ($d = mysql_fetch_assoc($r)) {
5     $forums []= $d;
6 }
7
8 // load the templates
9 load_template('header');
10 load_template('forum_list',$forums);
11 load_template('footer');
```





Consistent Naming Scheme



```
1 class Foo_Bar {  
2     public function someDummyMethod() {  
3  
4     }  
5  
6     public function procedural_function_name() {  
7  
8     }  
9 }
```





```
1 // Dirty
2 const loadConfigFromServer = () => {
3   ...
4 };
```

```
1 // Clean
2 const loadConfig = () => {
3   ...
4 };
```





DRY Principle




“Don’t Repeat Yourself”

The DRY principle states that these small pieces of knowledge may only occur exactly once in your entire system.




Avoid Deep Nesting



```
1 function do_stuff() {
2
3 // ...
4
5     if (is_writable($folder)) {
6
7         if ($fp = fopen($file_path,'w')) {
8
9             if ($stuff = get_some_stuff()) {
10
11                 if (fwrite($fp,$stuff)) {
12
13                     // ...
14
15                 } else {
16                     return false;
17                 }
18             } else {
19                 return false;
20             }
21         } else {
22             return false;
23         }
24     } else {
25         return false;
26     }
27 }
```

```
1 function do_stuff() {
2
3 // ...
4
5     if (!is_writable($folder)) {
6         return false;
7     }
8
9     if (!$fp = fopen($file_path,'w')) {
10         return false;
11     }
12
13     if (!$stuff = get_some_stuff()) {
14         return false;
15     }
16
17     if (fwrite($fp,$stuff)) {
18         // ...
19     } else {
20         return false;
21     }
22 }
```





Limit Line Length



```
1 // bad
2 $my_email->set_from('test@email.com')->add_to('programming@gmail.com')->set_subject('Methods Chained')->set_body('Some long message')->send();
3
4 // good
5 $my_email
6     ->set_from('test@email.com')
7     ->add_to('programming@gmail.com')
8     ->set_subject('Methods Chained')
9     ->set_body('Some long message')
10    ->send();
```





File and Folder Organization



- ▷ folder css
- ▲ folder system
 - ▲ folder application
 - ▷ folder config
 - ▷ folder controllers
 - ▷ folder errors
 - ▷ folder fixtures
 - ▷ folder helpers
 - ▷ folder hooks
 - ▷ folder language
 - ▷ folder libraries
 - ▷ folder models
 - ▷ folder plugins
 - ▷ folder views
 - index.html
 - ▷ folder cache
 - ▷ folder codeigniter
 - ▷ folder database
 - ▷ folder fonts
 - ▷ folder helpers
 - ▷ folder language
 - ▷ folder libraries
 - ▷ folder logs
 - ▷ folder plugins
 - ▷ folder scaffolding
 - index.html
- .htaccess
- index.php
- license.txt

- ▷ folder database
- ▷ folder fonts
- ▷ folder helpers
- ▷ folder language
- ▲ folder libraries
 - Benchmark.php
 - Calendar.php
 - Cart.php
 - Config.php
 - Controller.php
 - Email.php
 - Encrypt.php
 - Exceptions.php
 - Form_validation.php
 - Ftp.php
 - Hooks.php
 - Image_lib.php
 - index.html
 - Input.php
 - Language.php
 - Loader.php
 - Log.php
 - Model.php
 - Output.php
 - Pagination.php
 - Parser.php
 - Profiler.php
 - Router.php
 - Session.php
 - Shout.php





Consistent Temporary Names



```
1 // $i for loop counters
2 for ($i = 0; $i < 100; $i++) {
3
4     // $j for the nested loop counters
5     for ($j = 0; $j < 100; $j++) {
6
7     }
8 }
```





Read Open Source Code



descriptive
file header

separator ->

class
documentation

array elements
tab indented

separator ->

class method
documentation

```
Exceptions.php
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2 /**
3  * CodeIgniter
4  *
5  * An open source application development framework for PHP 4.3.2 or newer
6  *
7  * @package      CodeIgniter
8  * @author        ExpressionEngine Dev Team
9  * @copyright     Copyright (c) 2008 - 2009, EllisLab, Inc.
10  * @license      http://codeigniter.com/user_guide/license.html
11  * @link         http://codeigniter.com
12  * @since        Version 1.0
13  * @filesource
14  */
15
16 // -----
17
18 /**
19  * Exceptions Class
20  *
21  * @package      CodeIgniter
22  * @subpackage    Libraries
23  * @category     Exceptions
24  * @author        ExpressionEngine Dev Team
25  * @link         http://codeigniter.com/user_guide/libraries/exceptions.html
26  */
27 class CI_Exceptions {
28     var $action;
29     var $severity;
30     var $message;
31     var $filename;
32     var $line;
33     var $ob_level;
34
35     var $levels = array(
36         E_ERROR           => 'Error',
37         E_WARNING         => 'Warning',
38         E_PARSE           => 'Parsing Error',
39         E_NOTICE          => 'Notice',
40         E_CORE_ERROR      => 'Core Error',
41         E_CORE_WARNING    => 'Core Warning',
42         E_COMPILE_ERROR   => 'Compile Error',
43         E_COMPILE_WARNING => 'Compile Warning',
44         E_USER_ERROR       => 'User Error',
45         E_USER_WARNING    => 'User Warning',
46         E_USER_NOTICE      => 'User Notice',
47         E_STRICT          => 'Runtime Notice'
48     );
49
50
51     /**
52      * Constructor
53      *
54      */
55     function CI_Exceptions()
56     {
57         $this->ob_level = ob_get_level();
58         // Note: Do not log messages from this constructor.
59     }
60
61     // -----
62
63     /**
64      * Exception Logger
65      *
66      * This function logs PHP generated error messages
67      *
68      * @access private
69      * @param string the error severity
70      * @param string the error string
71      * @param string the error filepath
72      * @param string the error line number
73      * @return string
74      */
75     function log_exception($severity, $message, $filepath, $line)
76     {

```





Thanks!

Any questions?

You can find me at:

- ◇ Github:
<https://github.com/MoAgamia/SE-Boot-Camp>





References

- ◇ <https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118>
- ◇ <https://www.amazon.co.uk/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882>
- ◇ <http://americanexpress.io/clean-code-dirty-code/>

