ICT 3.2.1.1 PHP Coding Standards

ICT Volume 3 : Application Standards

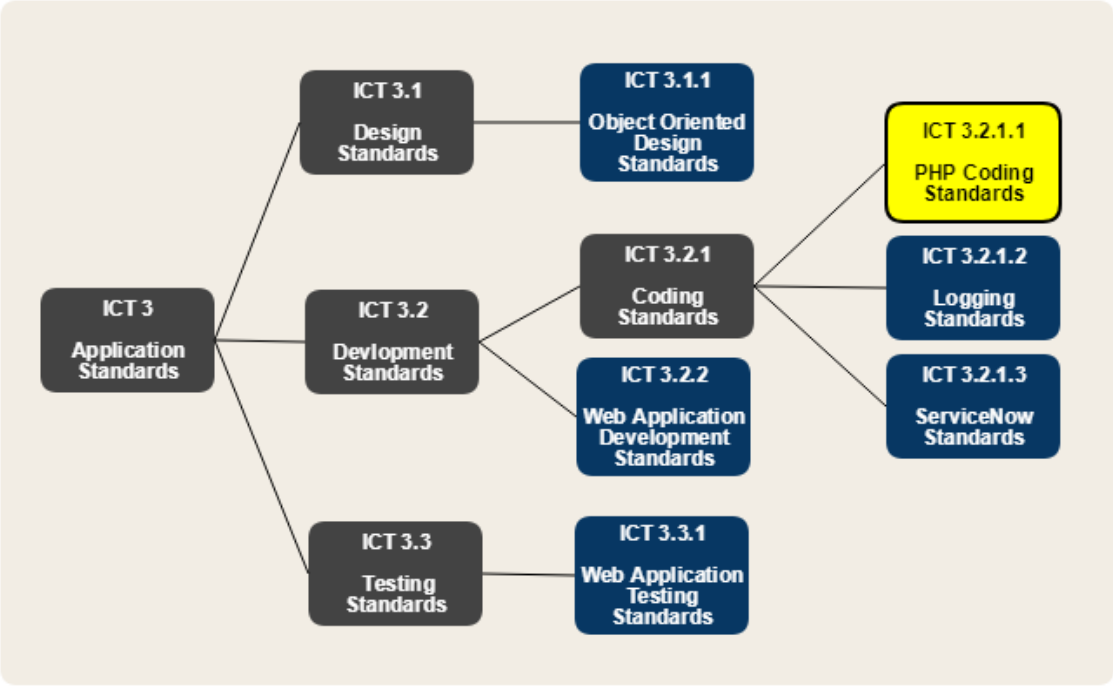
ICT 3.2.1.1 PHP Coding Standards

Abstract

This document defines the standards applicable to PHP coding.

Table of Contents

Document Hierarchy Diagram



Standards Brief
This document serves to outline standards that shall apply within Deakin University.

Standard Document Access
All Deakin University staff and authorised/approved contracted personnel are provided access to this document.

Policy
These standards must be used in conjunction with all other referenced standards, and when considered in isolation from the referenced standards may not constitute adequate conformance.

Conflict of Information or Clarification
Whenever a conflict of information occurs or clarification of instruction is required all queries shall be made to the Deakin University eSolutions.

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119.

1. Coding Standards

1.1 The *PSR-1* and *PSR-2* coding standards shall be adopted

All conventions outlined in the PSR-1, PSR-2 and Coding Standards shall be adopted with exception to items identified under section 1.2 herein.

https://www.php-fig.org/psr/psr-1/

https://www.php-fig.org/psr/psr-2/

1.2 Exceptions

Exceptions to standard 1.1 for Deakin University coding standards.

Unless otherwise stated, a “section” refers to the relevant section within the PSR-1 and PSR-2 document referenced in 1.1 above.

1.2.1 Alternative control structures

PSR-2 Section 5

The alternate syntax for control structures (if, ifelse, else, switch, while, do while, for foreach) within view scripts SHOULD be used when intermixing PHP and HTML code (http://php.net/manual/en/control-structures.alternative-syntax.php) .

e.g.,

```
    <?php if ($a == 5): ?>
    A is equal to 5
    <?php endif; ?>
```

1.3 Additions

The following additions shall apply to the PSR-1 and PSR-2 standards referenced in 1.1 above.

1.3.1 Output of HTML via PHP shall be avoided where practical

Any HTML output that is required to be returned to the client SHOULD be emitted outside of PHP delimiter tags, rather than via PHP functions such as `echo` or `print` where practical.

1.3.2 "Magic numbers" shall not be used

Constants shall be employed to represent any constant value. Configuration files shall be employed to represent any configurable value. Magic numbers SHALL NOT be used within any program code.

Class constants SHOULD be used in preference over global constants (per https://framework.zend.com/manual/2.4/en/ref/coding.standard.html#constants)

1.3.3 PHPDoc

For functions, the `@param`, `@throws` and `@return` tags MUST be included, along with the type of parameter that is expected.

If two types are accepted, they SHOULD both be specified, eg. `string|boolean`

If more than two types are accepted, then the parameter type SHOULD be `mixed`.

1.3.4 Short array syntax

Arrays MUST be declared using the short array syntax, e.g., `[ ]` instead of `array()`

1.3.5 Multi-line array syntax

The last line of a multi-line array MUST have a trailing comma, ref https://framework.zend.com/manual/2.4/en/ref/coding.standard.html#numerically-indexed-arrays

e.g.

```
    [
        'a',
        'b',
        'c',
    ]
```

1.3.6 Whitespace in blank lines

Blank lines MUST not contain any whitespace characters

2. Appendix A

Definitions

Term/Abbreviation	Definition
Class	The basic building block of software in the object-oriented programming paradigm.
PSR	A series of PHP Proposed Standards Recommendations agreed to by a consensus of members of the PHP Framework Interoperability Group