**Règles de codage**

Table des matières

[1Général 2](#__RefHeading__116_1573302270)

[2Nommage 2](#__RefHeading__120_1573302270)

[2.1Règle général de nommage 2](#__RefHeading__122_1573302270)

[2.2Noms des fichiers 2](#__RefHeading__124_1573302270)

[2.3Noms des variables 2](#__RefHeading__126_1573302270)

[2.3.1Nom des variables communes 2](#__RefHeading__128_1573302270)

[2.3.2Noms des variables de classes 2](#__RefHeading__130_1573302270)

[2.4Noms des constantes 2](#__RefHeading__132_1573302270)

[2.5Noms des fonctions et méthodes 3](#__RefHeading__134_1573302270)

[2.5.1Noms des fonctions régulières 3](#__RefHeading__136_1573302270)

[2.5.2Noms des méthodes 3](#__RefHeading___Toc674_1892402734)

[2.5.3Noms des accessors/mutators (getters/setters) 3](#__RefHeading__138_1573302270)

[2.6Noms des classes 3](#__RefHeading__142_1573302270)

[3Commentaires 3](#__RefHeading___Toc457_163950037)

[4Autres 4](#__RefHeading__158_1573302270)

[4.1Accolades 4](#__RefHeading__162_1573302270)

[4.2Identation 4](#__RefHeading___Toc294_1923617368)

[5Sources 4](#__RefHeading__156_1573302270)

# Général

Tout doit être écrit en anglais.

# Nommage

## Règle général de nommage

Les noms de fonctions, de variables et de classes devraient être descriptifs : éviter les abréviations.

## Noms des fichiers

Les noms des fichiers php possède le même nom que la classe qu'il contient.

*LoginController.php => la classe LoginController se trouve dans ce fichier.*

## Noms des variables

### Nom des variables communes

Les variables commencent par une minuscule et chaque nouveau mot commence par une majuscule.

*$reservationDate ;*

### Noms des variables de classes

Public : Même chose que pour les variables communes.

Private, protected : Même chose que pour les variables communes mais il faut ajouter un « \_ » au début de celles-ci.

*class TableInfo*

*{*

*private $\_reservationDate ;*

*protected $\_reservationDate ;*

*public $reservationDate ;*

*}*

## Noms des constantes

Tous est en majuscule et les mots sont séparés par un « \_ ».

*const DAYS\_IN\_WEEK = 7 ;*

## Noms des fonctions et méthodes

### Noms des fonctions régulières

Les fonctions commence par une minuscule et chaque nouveau mot commence par une majuscule.

*function addTableEntry() {… }*

**Il faut éviter les fonctions avec une portée globale au maximum. Il est recommandé de mettre ces fonctions dans des classes statiques.**

### Noms des méthodes

Elles suivent la même convention que les variables de classe. Elles commencent par une minuscule et chaque nouveau mot commence par une majuscule. Ainsi qu'un « \_ » est ajouté aux méthodes privées et protégées.

*Class Example*

*{*

*private function \_doSomething() { … }*

*protected function \_doSomethingToo() { … }*

*public function doNothing() { … }*

*}*

### Noms des accessors/mutators (getters/setters)

Les getters/setters commencent par get/set suivi de la même convention que pour les fonctions normales (le 1er mot commençant par une majuscule!).

*function getNumEntries() ;*

*function setNumEntries($newValue) ;*

## Noms des classes

Les classes ont la même convention que les variables normales. Elles sont juste une majuscule au début.

*class TableInfo { … }*

# Commentaires

*A discuter*

# Autres

## Accolades

Les accolades ouvrantes se placent sur la ligne suivant la déclaration.

*Class A*

*{*

*}*

*if(a == 1)*

*{*

*}*

*else if (a == 2)*

*{*

*}*

*else*

*{*

*}*

*function maFonction()*

*{*

*}*

## Identation

Pour l'indentation, des espaces sont utilisés à la place des tabulations et le nombre d'espaces utilisés est de 3.

## 4.3 Javascript

En javascript les accolades se mettent à la suite de l'instruction (condition, boucle …)

*if(true) {*

*}*

# Sources

<http://framework.zend.com/manual/1.12/fr/coding-standard.naming-conventions.html>

<https://pear.php.net/manual/en/standards.naming.php>