

Charte de Codage

Sommaire

Table de suivi

.....

2

1. Formatage des fichiers

.....	3
1.1. Indentation	3
1.2. Longueur maximum d'une ligne.....	3
1.3. Terminaison de lignes	3
2. Convention de nommage	3
2.1. Classes	3
2.2. Noms de fichiers	4
2.3. Fonctions et méthodes	4
2.4. Variables	4
2.5. Constantes	4
3. Style de codage	5
3.1. Classes	5
3.2. Fonctions et méthodes	6
3.3. Structures de contrôle	6
3.4. Documentation intégrée	7

4. Modélisation de bases de données	7
4.1. Normes de nommage	7
4.2. Entités	7
4.3. Associations	7
4.4. Attributs	7
4.5. Tables	8
4.6. Colonnes d'une table	8
4.7. Procédures stockées	8
4.8. Bases de données	8
4.9. Serveurs	8
5. Références	8

1. Formatage des fichiers

1.1. Indentation

Utilisez une indentation avec tabulations.

1.2. Longueur maximum d'une ligne

La longueur souhaitée d'une ligne est de 80 caractères, c'est-à-dire que les développeurs devraient avoir pour but de ne pas dépasser les 80 caractères pour des raisons pratiques. Cependant, des lignes plus longues sont acceptables. La longueur maximum de toute ligne de code PHP est de 120 caractères.

1.3. Terminaison de lignes

La terminaison de ligne est la terminaison standard pour les fichiers textes UNIX.

Les lignes doit finir seulement avec un "linefeed" (LF). Les linefeeds sont représentés comme 10 en ordinal, ou 0x0A en hexadécimal.

2. Convention de nommage

2.1. Classes

- Les noms des classes doivent être significatifs.
- Les noms de classes ne peuvent contenir que des caractères alphanumériques. Les nombres sont autorisés, mais déconseillés.

- Si un nom de classe comprend plus d'un mot, les mots seront séparées par des tirets bas « _ ».

2.2. Noms de fichiers

- Pour tous les autres fichiers, seuls des caractères alphanumériques, tirets bas et tirets demi-cadratin ("-") sont autorisés. Les espaces et les caractères spéciaux sont interdits.

2.3. Fonctions et méthodes

- Les noms de fonctions ne peuvent contenir que des caractères alphanumériques. Les tirets bas ("_") ne sont autorisés. Les nombres sont autorisés mais déconseillés.
- Les noms de fonctions doivent toujours commencer avec une lettre en minuscule. Quand un nom de fonction est composé de plus d'un seul mot, les mots seront séparés par des tirets bas.
- La clarté est conseillée. Le nom des fonctions devrait être aussi explicite que possible, c'est un gage de compréhension du code.
- Pour la programmation orientée objet, les accesseurs aux objets doivent toujours être préfixés par soit "get" soit "set".

2.4. Variables

- Les noms de variables ne peuvent contenir que des caractères alphanumériques. Les tirets bas sont permis. Les nombres sont autorisés mais

déconseillés.

- Tout comme les noms de fonction, les noms de variables doivent toujours commencer par un caractère en minuscule et séparés par des tirets bas.
- La clarté est conseillée. Les variables devraient toujours être aussi claires que pratiques. Des noms de variables comme "\$i" et "\$n" sont déconseillé pour tout autre usage que les petites boucles. Si une boucle contient plus de 20 lignes de code, les variables pour les indices doivent avoir des noms descriptifs.

2.5. Constantes

- Les constantes peuvent contenir des caractères alphanumériques et des tirets bas. Les nombres sont autorisés.
- Les constantes doivent toujours être en majuscule, cependant les mots pouvant les composer doivent être séparés par des tiret-bats ("_").

- Les constantes doivent toujours être définies comme des membres d'une classe, en utilisant la construction "const". Définir des constantes globales avec "define" est permis mais déconseillé.

3. Style de codage

3.1. Classes

- Les classes doivent être nommées conformément aux conventions de nommage de Spotlight.
- Toutes les classes doivent avoir un bloc de documentation.

- L'accolade est toujours écrite à la fin de la ligne qui contient le nom de la classe.
- Une seule classe est permise par fichier PHP.

Voici un exemple d'une déclaration de classe autorisée :

```
/**
Bloc documentation
**/

Class Exemple_classe {
    //contenu de la classe
    //qui doit être indenté avec des tabulations
}
```

- Si une classe étend et/ou implémente autres classes, la déclaration sera sur la même ligne quand si possible.
- Si la déclaration dépasse la taille maximale d'une ligne ; découper la ligne avant le mot « extends » et/ou « implements ».
- Si une classe implémente plusieurs interfaces et la déclaration dépasse la taille maximale d'une ligne, découper la ligne avant le mot « implements » et après chaque virgule.
- Les variables membres doivent être nommées en respectant les conventions de nommage.
- Toute variable déclarée dans une classe doit être listée en haut de cette classe, avant toute déclaration de méthode.

- La construction `var` n'est pas permise. Les variables membres déclarent

toujours leur visibilité en utilisant la construction `private`, `protected` ou `public`. L'accès direct à ces variables membres en les rendant publiques est permis mais déconseillé. Il est préférable d'utiliser des accesseurs (`set/get`).

3.2. Fonctions et méthodes

- Les fonctions doivent être nommées en respectant les conventions de nommage.
- Les fonctions internes aux classes doivent toujours déclarer leur visibilité en utilisant la construction `private`, `protected` ou `public`.
- Tout comme les classes, l'accolade ouvrante est toujours écrite après le nom de la fonction. Il n'y a pas d'espace entre le nom de la fonction et les parenthèses des arguments. Il n'y a pas d'espace entre la parenthèse fermante et l'accolade.
- Les fonctions globales sont fortement déconseillées.
- Les arguments d'une fonction sont séparés par un espace après la virgule de délimitation.

3.3. Structures de contrôle

- Les structures de contrôles basées sur les constructions `if` et `elseif` doivent avoir un seul espace avant la parenthèse ouvrante de la condition, et un seul espace après la parenthèse fermante.
- Pour la condition entre les parenthèses, les opérateurs doivent être séparés par des espaces pour une meilleure lisibilité. Les parenthèses internes sont conseillées pour améliorer le regroupement logique de longues conditions.
- L'accolade ouvrante est écrite sur la même ligne que la condition. L'accolade fermante est toujours écrite sur sa propre ligne. Tout contenu présent à

l'intérieur des accolades doit être indenté par 4 espaces.

- Les instructions de contrôle avec "switch" ne doivent avoir qu'un seul espace avant la parenthèse ouvrante de l'instruction conditionnelle, et aussi un seul espace après la parenthèse fermante.

- Tout le contenu à l'intérieur de l'instruction "switch" doit être indenté avec 4 espaces. Le contenu sous chaque "case" doit être indenté avec encore 4 espaces supplémentaires.

3.4. Documentation intégrée

- Tous les fichiers de code source écrit doivent contenir un bloc de documentation « docblock », en haut de chaque fichier, et un docblock de classe immédiatement au-dessus de chaque classe.
- Chaque fonction, méthode, doit avoir un docblock contenant au minimum :
 - o Une description de la fonction.
 - o Tous les arguments.
 - o Toutes les valeurs de retour possibles.

4. Modélisation de bases de données

4.1. Normes de nommage

Les noms des objets d'un modèle ou d'un schéma devront être significatifs et pertinents, respectant les critères suivants :

- Contenir deux zones de caractères alphabétiques séparées par le caractère souligné « _ »
- La première zone est spécifique au type de l'objet.
- La deuxième zone doit respecter le format de nommage décrit précédemment.

4.2. Entités

La première zone d'un nom d'entité doit contenir la lettre « E ».

4.3. Associations

La première zone d'un nom d'association doit contenir « A ».

4.4. Attributs

Le nom d'un attribut doit signifier la nature du type de données qu'il représente.

4.5. Tables

- La première zone d'un nom de table doit contenir la lettre « T »
- Le nom d'une table doit reprendre le corps du nom de l'entité, ou le nom de la relation si cette dernière est nommée

4.6. Colonnes d'une table

L'ordre de création et de description des colonnes devra répondre aux règles suivantes :

- Les colonnes composant la clef primaire de la table devront être les premières colonnes décrites de la table.
- Les colonnes composant les clefs étrangères devront être en dernier.
- Les colonnes doivent être regroupées lorsqu'ils font partie d'un sous-ensemble significatif de la table.

4.7. Procédures stockées

- La première zone d'un nom de procédure doit contenir « P » suivi d'une indication pertinente.
- Les paramètres et les variables doivent respecter le format lowerCamelCase.

4.8. Bases de données

La première zone d'un nom de base de données doit contenir « PGS » suivi d'une description pertinente.

4.9. Serveurs

La première zone d'un nom de serveur doit contenir la lettre « S » suivi d'une description pertinente.

5. Références

- <http://www.codeigniter.com/>
- <http://www.developpez.com/>

Charte de codage