# Nommage

## Généralités

* Un nom de fichiers est toujours au singulier.
* Un nom de dossier est, dans la mesure du possible, toujours au pluriel.
* Chaque block de code défini par des « { } » devra se trouver aligné verticalement avec l’élément de code qui le précède. Chaque « { » étant a la ligne, aligné également sur l’élément de code qui le précède, ainsi sur le « } » de fin de block.

## Classes

* Pour l’utilisation de l’autoload, à chaque fichier correspondra une seul classe. Il découle donc des règles précédentes, qu’une classe devra être toujours nommée au singulier.
* L’autoload de Zend impose de nommer un fichier en fonction de sont arborescence. Pour chaque dossier parent, et dans l’ordre des générations, La classe aura le nom du dossier, première lettre en majuscule, et suivie d’un underscrore (« \_ »).
* Pour éviter toute confusion le nom propre a la classe devra être fait au format UpperCamelCase, composé uniquement de caractère alphanumérique. Les lettres majuscules matérialisant ainsi la séparation des mots syntaxique du nom propre de la classe.
* Une classe abstraite devra, dans le mesure du possible, comporter en fin de nom la syntaxe « Element » afin de les différencier facilement.

### Attributs

* Les attributs d’une classe devront toujours être déclarés avant l’implémentation des fonctions, et dans la mesure du possible, regroupés par visibilité.
* Les attributs privés ou protégés devront être précédé d’un underscrore (« \_ ») et chaque élément syntaxique du nom de l’attribut, séparé par des underscrores (« \_ ») et entièrement typographié en minuscule.
* Les propriétés public, devront, elles être typographié au format UpperCamelCase, chaque élément syntaxique étant symbolisé par une majuscule.

### Fonctions

#### Les accesseurs

* Les getter et les setter seront nommé au format lowerCamelCase, et commenceront systématiquement par le mot « get » ou « set » en minuscule, suivi du nom propre relatif a l’attribut qu’il accède, sans undercore, chaque élément syntaxique étant symbolisé par les majuscule.
* Dans la mesure du possible, chaque attribut protégé devra avoir sa paire d’accesseur qui lui correspond.
* Les accesseurs suivront directement la déclaration des attributs auxquels ils accèdent et seront groupé par direction.

#### Les méthodes

* Elles devront être toujours nommées au format lowerCamelCase, chaque élément syntaxique étant symbolisé par la présence d’une majuscule
* Les procédures ne retournant ni valeur, ni objet devront au minimum retourner un booléen d’état d’exécution.