



# Programmation orientée objet en PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [POO PHP](#) ▶ [Les namespaces](#) ▶ [Introduction](#)

## Introduction

Les [namespaces](#) ou espaces de noms servent à éviter les conflits de nom entre les classes et à organiser de façon logique l'ensemble des classes du programme. Les espaces de noms s'organisent en arborescence à l'image d'un système de fichiers. Par exemple dans un dossier je ne peux avoir deux fichiers de même nom (et de même extension); par contre je peux avoir dans mon système de fichiers deux fichiers de même nom et de même extension s'ils appartiennent à deux répertoires différents. En PHP, cela signifie que deux classes de même nom mais de namespaces différents seront considérées comme différentes. Par exemple la classe `com\greta\Personne` sera différente de `fr\afpa\Personne`.

Dans une classe PHP, un namespace se déclare en utilisant le mot clé "namespace" comme première instruction du script. Le namespace peut contenir des sous-namespaces séparés par des antislashes ("\") : par exemple dans l'espace de nom `com\greta`, l'espace de nom `greta` est un sous espace de nom de `com`. L'espace de nom `com` pourrait très bien contenir d'autres espaces de nom comme `fyligrane` par exemple, ce qui donnerait `com\fyligrane`.

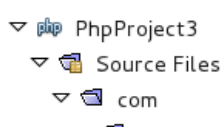
Personne.php dans le répertoire `com/greta`

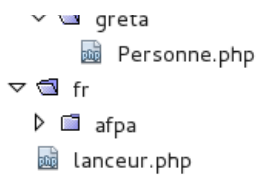
```
1 <?php
2
3 // com/greta/Personne.php
4
5 namespace com\greta;
6
7 class Personne {
8
9     public function __toString() {
10         return 'la classe ' . __CLASS__ . ' fait partie du namespace ' . __NAMESPACE__;
11     }
12
13 }
```

Personne.php dans le répertoire `fr/afpa` :

```
1 <?php
2
3 // fr/afpa /Personne.php
4
5 namespace fr\afpa;
6
7 class Personne {
8
9     public function __toString() {
10         return 'la classe ' . __CLASS__ . ' fait partie du namespace ' . __NAMESPACE__;
11     }
12
13 }
```

Il est souvent très pratique de faire correspondre l'espace de nom et les répertoires des classes : par exemple une classe `com\greta\Personne` sera placée dans un répertoire `com/greta` (sous unix) :





Dans les deux classes nous avons utilisé les constantes magiques `__CLASS__` (qui donne le nom de la classe) et `__NAMESPACE__` (qui donne le nom du namespace de la classe).

Dans notre lanceur nous allons pouvoir utiliser les classes après inclusion en utilisant le nom "qualifié" de la classe c'est-à-dire le namespace + le nom (par exemple : `com\greta\Personne`).

```

1 <?php
2
3 include './com/greta/Personne.php';
4 include './fr/afpa/Personne.php';
5
6 $persGreta = new com\greta\Personne();
7 echo $persGreta;
8 echo PHP_EOL;
9 $persAfpa = new fr\afpa\Personne();
10 echo $persAfpa;
  
```

Il est possible d'importer des espaces de noms. Par exemple en important l'espace de noms `com\greta\Personne`, nous ne serons plus obligés de préciser l'espace de noms à l'instanciation d'une classe `com\greta\Personne`. Par contre, pour instancier `fr\afpa\Personne` nous devrons toujours utiliser l'espace de nom à l'instanciation.

```

1 <?php
2
3 include './com/greta/Personne.php';
4 include './fr/afpa/Personne.php';
5
6 use com\greta\Personne;
7
8 $persGreta = new Personne();
9 echo $persGreta;
10 echo PHP_EOL;
11 $persAfpa = new fr\afpa\Personne();
12 echo $persAfpa;
  
```

Il est aussi possible de créer des alias sur les noms des classes :

```

1 <?php
2
3 include './com/greta/Personne.php';
4 include './fr/afpa/Personne.php';
5
6 use com\greta\Personne as P1;
7 use fr\afpa\Personne as P2;
8
9 $persGreta = new P1();
10 echo $persGreta;
11 echo PHP_EOL;
12 $persAfpa = new P2();
13 echo $persAfpa;
  
```

Ou sur une partie du namespace :

```

1 <?php
2
3 include './com/greta/Personne.php';
4 include './fr/afpa/Personne.php';
5
6 use com\greta as gr;
  
```

```
7 use fr\afpa as af;
8
9 $persGreta = new gr\Personne();
10 echo $persGreta;
11 echo PHP_EOL;
12 $persAfpa = new af\Personne();
13 echo $persAfpa;
```

[Fin](#)

## NAVIGATION



### Accueil

#### ■ [Ma page](#)

[Pages du site](#)[Mon profil](#)[Cours actuel](#)

#### [POO PHP](#)

[Participants](#)[Généralités](#)[La programmation orientée objet : premiers pas](#)[L'héritage](#)[Les interfaces](#)[Le typage](#)[Les namespaces](#) [Introduction](#) [T.P. namespaces](#)[Les exceptions](#)[Les bases de données avec PDO](#)[Les tests avec PHPUnit](#)[Petite application version 2](#)[Petite application version 3](#)[Mes cours](#)

## ADMINISTRATION

[Administration du cours](#)[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))  
[POO PHP](#)



# Programmation orientée objet en PHP

[Accueil](#) ▶ [Mes cours](#) ▶ [Développement logiciel](#) ▶ [POO PHP](#) ▶ [Les namespaces](#) ▶ [T.P. namespaces](#)

## T.P. namespaces

Modifier le devoir typage pour que la classe Voyageur fasse partie du namespace machin\chouette, et que la classe Personne fasse partie du namespace bidule\truc.

Importer les espaces de noms en utilisant le mot clé "use".

Les fichiers seront à remettre dans une archive zip dont le nom sera de la forme "namespaces\_nom\_prenom.zip".

## État du travail remis

Numéro de tentative	Ceci est la tentative 1.
Statut des travaux remis	Aucune tentative
Statut de l'évaluation	Pas évalué
Dernière modification	vendredi 27 mars 2015, 15:36

Ajouter un travail

Modifier votre travail remis

### NAVIGATION



[Accueil](#)

■ [Ma page](#)

Pages du site

Mon profil

Cours actuel

[POO PHP](#)

Participants

Généralités

La programmation orientée objet : premiers pas

L'héritage

Les interfaces

Le typage

Les namespaces

[Introduction](#)

[T.P. namespaces](#)

Les exceptions

Les bases de données avec PDO

Les tests avec PHPUnit

Petite application version 2

Petite application version 3

[Mes cours](#)

**ADMINISTRATION**[Administration du cours](#)[Réglages de mon profil](#)

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))  
[POO PHP](#)