

PHP : espace de noms (namespace)

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



- 1 Introduction
- 2 Problématique
- 3 Solution avec les namespaces

Namespace ?

- Un concept repris du langage **C++**
- Un espace de stockage abstrait (n'existe pas physiquement)
- Un répertoire virtuel
- Conçu pour développer des structures fonctionnelles (variable, constante, fonction...) sans avoir de conflit de nom

PHP

Considérons le fichier salutationFr.php **défini dans un répertoire** fonctions

```
function direBonjour(): void
{
    echo "Bonjour";
}
```

Et un deuxième fichier salutationEn.php

```
function direBonjour(): void
{
    echo "Good morning";
}
```

PHP

Dans index.php, commençons par inclure les deux fichiers

```
<?php
    include 'fonctions/salutationEn.php';
    include 'fonctions/salutationFr.php';
?>
```

PHP

Dans index.php, commençons par inclure les deux fichiers

```
<?php
    include 'fonctions/salutationEn.php';
    include 'fonctions/salutationFr.php';
?>
```

Dans le body d'index.php, appelons la fonction direBonjour()

```
<?php
    direBonjour();
?>
```

PHP

Dans index.php, commençons par inclure les deux fichiers

```
<?php
    include 'fonctions/salutationEn.php';
    include 'fonctions/salutationFr.php';
?>
```

Dans le body d'index.php, appelons la fonction direBonjour()

```
<?php
    direBonjour();
?>
```

En exécutant, le résultat est

Fatal error: Cannot redeclare direBonjour()

Première solution

Faire `include` d'un seul fichier

Première solution

Faire `include` d'un seul fichier

Mais

dans certains cas on a besoin d'utiliser les deux

PHP

Première solution

Faire `include` d'un seul fichier

Mais

dans certains cas on a besoin d'utiliser les deux

Deuxième solution

Définir chacune de ses fonctions dans un namespace différent.

PHP

Définissons un namespace French dans salutationFr.php

```
namespace French;

function direBonjour(): void
{
    echo "Bonjour";
}
```

Et un deuxième English dans salutationEn.php

```
namespace English;

function direBonjour(): void
{
    echo "Good morning";
}
```

Il ne reste qu'à préciser le namespace au moment de l'appel

```
<?php
    French\direBonjour();
    English\direBonjour();
?>
```

Il ne reste qu'à préciser le namespace au moment de l'appel

```
<?php
    French\direBonjour();
    English\direBonjour();
?>
```

En exécutant, le résultat est

Bonjour
Good morning

Dans index.php, on peut donner des alias à nos namespaces avec use ... as

```
<?php
    include 'fonctions/salutationEn.php';
    use English as en;
    include 'fonctions/salutationFr.php';
    use French as fr;
?>
```

Dans index.php, on peut donner des alias à nos namespaces avec use ... as

```
<?php
    include 'fonctions/salutationEn.php';
    use English as en;
    include 'fonctions/salutationFr.php';
    use French as fr;
?>
```

Il reste qu'à utiliser les alias au moment de l'appel

```
<?php
    fr\direBonjour();
    en\direBonjour();
?>
```

Dans index.php, on peut donner des alias à nos namespaces avec use ... as

```
<?php
    include 'fonctions/salutationEn.php';
    use English as en;
    include 'fonctions/salutationFr.php';
    use French as fr;
?>
```

Il reste qu'à utiliser les alias au moment de l'appel

```
<?php
    fr\direBonjour();
    en\direBonjour();
?>
```

En exécutant, le résultat est

Bonjour

Good morning

On peut aussi imbriquer les namespaces (contenu de salutationEn.php)

```
namespace English;

function direBonjour(): void
{
    echo "Good morning";
}

namespace English\American;

function direBonjour(): void
{
    echo "Good mornin";
}
```

Dans index.php, on commence par les include

```
<?php
    include 'fonctions/salutationEn.php';
    use English as en;
    use English\American as am;
    include 'fonctions/salutationFr.php';
    use French as fr;
?>
```

Dans index.php, on commence par les include

```
<?php
    include 'fonctions/salutationEn.php';
    use English as en;
    use English\American as am;
    include 'fonctions/salutationFr.php';
    use French as fr;
?>
```

Il reste qu'à utiliser les alias au moment de l'appel

```
<?php
    fr\direBonjour();
    en\direBonjour();
    am\direBonjour();
?>
```

Dans index.php, on commence par les include

```
<?php
    include 'fonctions/salutationEn.php';
    use English as en;
    use English\American as am;
    include 'fonctions/salutationFr.php';
    use French as fr;
?>
```

Il reste qu'à utiliser les alias au moment de l'appel

```
<?php
    fr\direBonjour();
    en\direBonjour();
    am\direBonjour();
?>
```

En exécutant, le résultat est

Bonjour
Good morning
Good mornin

On peut définir des éléments prédéfinis : par exemple la fonction substr (dans un fichier chaine.php)

```
namespace Chaine;

function substr($ch, $num, $pos)
{
    // \ : la fonction d'origine
    return \substr($ch, $pos, $num);
}
```

Dans index.php, on commence par l'inclure

```
<?php
    include 'fonctions/chaine.php';
    use chaine as str;
?>
```

Dans index.php, on commence par l'inclure

```
<?php
    include 'fonctions/chaine.php';
    use chaine as str;
?>
```

On compare les deux fonctions PHP

```
<?php
echo "la nouvelle <br>";
echo str\substr("bonjour", 2, 5);
echo "<br>l'originale <br>";
echo substr("bonjour", 5, 2);
?>
```

Dans index.php, on commence par l'inclure

```
<?php
    include 'fonctions/chaine.php';
    use chaine as str;
?>
```

On compare les deux fonctions PHP

```
<?php
echo "la nouvelle <br>";
echo str\substr("bonjour", 2, 5);
echo "<br>l'originale <br>";
echo substr("bonjour", 5, 2);
?>
```

En exécutant, le résultat est

la nouvelle
ur
l'originale
ur

PHP

Propriétés

- On peut déclarer plusieurs namespaces par fichier (mais déconseillé)
- Par défaut (si on ne déclare aucun namespace), on est dans le namespace global
- Pour appeler une fonction du namespace global, on utilise \
- Pour retourner au namespace global, il suffit d'écrire `namespace`
- On peut déclarer plusieurs sous-namespaces, comme pour les dossiers sous windows, il faut les séparer par

`Espace1\Espace2\Espace3`

Si les namespaces `Espace1` et `Espace2` n'existent pas, ils seront créés.

- La constante `__NAMESPACE__` contient toujours le nom du namespace courant
- Pour accéder à un élément d'un namespace différent : `namespace\nomElement`

PHP

Propriétés

- On peut déclarer plusieurs namespaces par fichier (mais déconseillé)
- Par défaut (si on ne déclare aucun namespace), on est dans le namespace global
- Pour appeler une fonction du namespace global, on utilise \
- Pour retourner au namespace global, il suffit d'écrire `namespace`
- On peut déclarer plusieurs sous-namespaces, comme pour les dossiers sous windows, il faut les séparer par

`Espace1\Espace2\Espace3`

Si les namespaces `Espace1` et `Espace2` n'existent pas, ils seront créés.

- La constante `__NAMESPACE__` contient toujours le nom du namespace courant
- Pour accéder à un élément d'un namespace différent : `namespace\nomElement`

Le namespace doit être le même que le dossier dans lequel se trouve la classe.