

RAPPEL PROGRAMMATION ORIENTÉE OBJET (POO).

La programmation orientée objet, POO, correspond à une autre manière d'imaginer, de construire et d'organiser son code.

Jusqu'à présent, nous avons codé de manière de procédurale, c'est-à-dire en écrivant une suite de procédures et de fonctions dont le rôle était d'effectuer différentes opérations sur des données (BDD), généralement contenues dans des variables, et tout ceci dans l'ordre d'écriture dans le script.

La POO est une façon différente d'écrire et d'arranger son code, autour de ce que l'on appelle des objets. Un objet est une 'entité' qui va pouvoir contenir un ensemble de variables (propriétés) et des fonctions (méthodes)

L'idée de la programmation orientée objet va donc être de grouper des parties de code qui servent à effectuer une tâche précise, ensemble, au sein d'objets afin d'obtenir une nouvelle organisation du code. (= meilleure relecture, meilleure gestion des bugs, meilleure maintenabilité).

Les intérêts principaux de la POO, vont être :

Une structure générale du code plus claire et aussi plus modulable mais également plus facile à maintenir ET à déboguer.

En informatique et en POO, **TOUT EST OBJET !**

La POO est paradigme d'un langage informatique.

La POO va induire des syntaxes différentes et c'est un peu l'une des raisons principales pour lesquelles PHP POO est souvent vu comme compliqué et parfois obscure.

Au final, vous vous apercevrez que la POO n'est pas si compliquée : ce n'est que façon différente de coder qui va amener de nombreux avantages !

Imaginez qu'on dispose des mêmes outils qu'en procédural, mais dans des objets : ce ne sont plus des fonctions mais des ensembles de fonctions et de variables enfermées (encapsuler) dans des objets (class Php) et qui vont effectuer des tâches, complexe ou non, que l'on pourra réutiliser partout dans notre code au besoin, et faire évoluer ces fonctions plus facilement.

Dans la pratique, la POO se base sur un concept fondamental que sont les classes. Pour comprendre ce qu'est un objet, il faut comprendre ce qu'est une classe.

Déclaration d'une classe (moule)

```
class Exemple  
{  
    private $prop ;  
  
    public function setProp ($prop) {  
        $this->prop = $prop ;  
        return $this ;  
    }  
};
```

Instanciation d'une classe, cela devient un objet (le résultat du moulage)

```
$exemple = new Exemple();
```

```
$exemple->setProp('toto');
```

Architecture (de dossiers) **MVC** :

Model – View – Controller est un **concept pattern** (un patron de conception)

Une application conforme au motif (patron) MVC comporte trois types de modules : les modèles, les vues et les contrôleurs.

Model :

C'est un élément, soit un objet (une classe PHP), qui contient les données ainsi que la logique en rapport avec les données : la validation, la lecture et l'enregistrement des données.

Un Model, dans sa forme la plus simple, peut contenir une seule et simple valeur, ou une structure de données plus complexes.

Le Model représente l'univers dans lequel s'inscrit l'application web (site web).

Par exemple, pour une application de banque, le modèle représente des comptes, des clients, des opérations bancaires telles que dépôt ou retrait d'argent et vérifie que les retraits ne dépassent pas la limite de crédit du compte.

View :

Partie visible d'une interface graphique. La View (vue) se sert du Model et peut prendre n'importe quelle forme graphique (diagramme , un formulaire HTML, des boutons HTML).

Une View contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du Model.

Dans une application de bureau classique (software desktop) la vue obtient les données nécessaires à la présentation du Model et les affiche.

La View peut également transmettre des données ou messages pour mettre à jour le Model.

Dans un application web, la View contient les balises HTML du code.

Controller :

C'est un module qui traite les actions de l'utilisateur, modifie aussi les données du Model et de la View (vue)

Le Controller reçoit toujours une Request (requête http) et retourne toujours une Response (réponse http).

Schéma MVC Symfony :

Model -> Représenter vos Entités et vos Tables sql : le model sert de schéma pour la création des tables de la BDD.

Controller -> le traitement des données, la logique de votre site web : Retourne une réponse HTTP (très souvent c'est une page HTML)

View -> L'affichage de pages HTML, contient tout votre code HTML.

Le MVC n'est un langage, ni une syntaxe, c'est une façon d'architecturer son projet (donc son code) séparé en plusieurs parties (bout de code) et gérer par des class PHP.

Cela sert à la cohérence du code, des données, de la maintenabilité du projet (du site web) et à son évolution future.

INSTALLATION DES LOGICIELS REQUIS

Composer :

Composer est un outil mis en place par la communauté des dev PHP, il sert de gestionnaire de dépendances (bibliothèques PHP) entre applications web et bibliothèques.

De manière plus précise, Composer permet de gérer pour chaque objet, la liste des modules et bibliothèques (bibliothèques) nécessaires à son fonctionnement ainsi que leurs versions.

Composer est utilisable via un terminal en ligne de commandes. De plus, il permet de mettre en place un système d'autoload pour les bibliothèques compatibles.

Pour l'installer -> <https://getcomposer.org/download/>


Pour Windows -> télécharger composer-Setup.exe (flèche rouge)
-> Installer-le en suivant les étapes.

Pour les macs -> copier les lignes de commande (première flèche verte)
-> Puis déplacer le fichier composer.phar dans le bon dossier (deuxième flèche verte)
`sudo mv composer.phar /usr/local/bin/composer`


Download Composer Latest: v2.1.8

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download  on [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.


Command-line installation

To quickly install Composer in the current  directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '756890a4488ce9024fc62c56153228907f1545c228516cbf63f8
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

- Download the installer to the current directory
- Verify the installer SHA-384, which you can also [cross-check here](#)
- Run the installer
- Remove the installer

Most likely, you  want to put the `composer.phar` into a directory on your PATH, so you can simply call `composer` from any directory (Global install), [using](#) for example:

```
sudo mv composer.phar /usr/local/bin/composer
```

Symfony CLI :

CLI veut dire Command Line Interface. C'est comme une interface graphique, avec des boutons et des liens pour effectuer des actions, à la différence où dans une interface en ligne de commande, les actions sont faites dans un terminal grâce à des commandes écrites.

En réalité, derrière une interface graphique, il y a un terminal (shell).

Pour l'installer : <https://symfony.com/download>

Dans un terminal :

```
symfony new --webapp my_project
```

Détails des termes :

symfony : appel de la CLI installée juste avant

new : un nouveau projet

--version=5.4 : OPTIONNEL si la version de PHP=7.4 – Indiquera quelle version de Symfony installer. Par défaut la dernière.

--webapp : lorsqu'il y a des doubles tirets dans une commande '-' c'est comme une option, un paramètre que l'on active. (flag)

my_project : ici on nomme notre projet (donc on remplace ce terme). On peut utiliser plusieurs conventions de nommage, du moment où il n'y a AUCUN espace ! (on préfère le PascalCase, le snake_case, ou kebab-case)

Laisser l'installation se terminer et ensuite regarder dans l'explorateur de fichiers (finder) ce qui a été installé.

Il n'est pas nécessaire que le projet symfony soit dans le dossier HTDOCS ou WWW des serveurs types WAMP XAMPP MAMP ... car symfony a son propre serveur.

Framework :

En programmation informatique, un framework désigne un ensemble cohérent de composants logiciels, structurés et découpés en dossiers et fichiers, qui sert à ne pas recréer les fondations ainsi que les grandes lignes, de tout ou partie, d'un logiciel. Ou un site web, application web.

De plus, un framework fournit des outils supplémentaires pour vous aidez à coder plus simplement, rapidement et proprement (bonnes pratiques).

En résumé, un framework vous fait gagner du temps. (différent d'un CMS, qui lui vous évite de coder).

Extensions VS Code :

Pour une meilleure utilisation de Symfony, on doit installer quelques extensions :

- **PHP intelephense** (améliorer l'utilisation php : autocomplétion)
- **PHP namespace resolver** (importation des class)
- **DotENV** (code couleur dans le fichier .env)
- **Twig Language 2** (amélioration dans le twig, notamment emmet)

Windows : télécharger et *extraire* **Cmder** (fichier .zip).

MacOS : télécharger et installer **Powershell**

Node.js :

Les gestionnaires de dépendances Javascript sont livrés avec Node.js. **Nous en aurons besoin dans notre app Symfony.**

Le premier s'appelle NPM (Node Package Manager)

Le second s'appelle Yarn. **Nous utiliserons Yarn**

Pour télécharger Node.js : <https://nodejs.org/en/download/>