

Programmation orientée Objet

La POO pour programmation orienté objet ou OOP en anglais. Jusqu'ici nous programmions en procédurale. Nous allons voir que la POO comporte des avantages non négligeable.

Les avantages de la formation orientés objets :

- Plus rapide et facile d'exécution
- Structure cohérente et claire
- Permet de ne pas se répéter
- Plus facile à maintenir, débbugger, modifier
- ...

Classes et objets :

Les classes et les objets sont les principaux aspects de la programmation orientée objet.

Une classe va être une sorte de « plan » qui va nous permettre de créer des objets à partir de ce modèle. Une classe peut avoir des propriétés(attributs) et des méthodes (fonctions).

Chaque objet individuel peut ainsi vivre, être modifier, avoir ses propres valeurs tout en partant de la même classe.

Créer une classe :

```
class Stagiaire{  
    public $nom;  
    public $prenom;
```

Créer un objet :

```
$benjamin = new Stagiaire();
```

Propriétés et méthodes :

Créer une méthode :

```
public function hello(){  
    echo "hello $this->nom";
```

Modifier une propriété :

Il y a deux possibilités :

Vous pouvez, à condition que la propriété soit en public, la modifier directement comme ceci :

```
$benjamin->nom="toto";
```

Sinon il faut utiliser des Setters comme ceci :

```
public function setNom($nom)
{
    $this->nom = $nom;

    return $this;
}
$benjamin = new Stagiaire();
$benjamin->setNom("Bailly");
```

Constructeur :

La fonction `__construct` va vous permettre de créer un objet avec des propriétés dès sa création.

ATTENTION la fonction « construct » s'écrit avec deux underscore `__construct()`.

Exemple :

```
function __construct($name,$prenom)
{
    $this->nom = $name;
    $this->prenom = $prenom;
}
```

Encapsulation

Il y a trois type de visibilité :

- Protected
- Private
- Public

Faites une recherche pour comprendre la différence entre ses trois types. Et complétez cette partie du cours

Qu'est-ce que l'encapsulation ?

TP POO 1 :

Etape 1 classe :

Créez une classe utilisateur, dans un dossier à part nommé class par exemple.

Cette classe aura comme attribut privé login et password .

Créez un constructeur dans cette classe pour pouvoir à la création d'une instance définir ce login et ce mot de passe.

Etape 2 html :

Faites un fichier html dans un dossier à part nommé view par exemple.

Cette page html contiendra un formulaire de connexion (login password) quand personne n'est connecté.

À l'envoi il renverra vers une page php.

Si quelqu'un est connecté il y aura écrit le login de l'utilisateur.

Etape 3 php :

Cette page php s'occupera de traiter le formulaire c'est dire :

- Si le login et le password sont renseignés vous pourrez créer une instance d'User que l'on stockera dans une session.
- S'il manque un champ on renverra un message d'erreur.
- La page gèrera la déconnexion également.
- À la fin on retourne quoi qu'il arrive sur la page html.