

MISE EN PRATIQUE DE LA POO EN PHP 1/6

Auteur : Sébastien Inion

Code source support : <https://replit.com/@Sebastien11/Point-et-geometrie#Point.class.php>

RAPPELS DES CONCEPTS DE LA P00

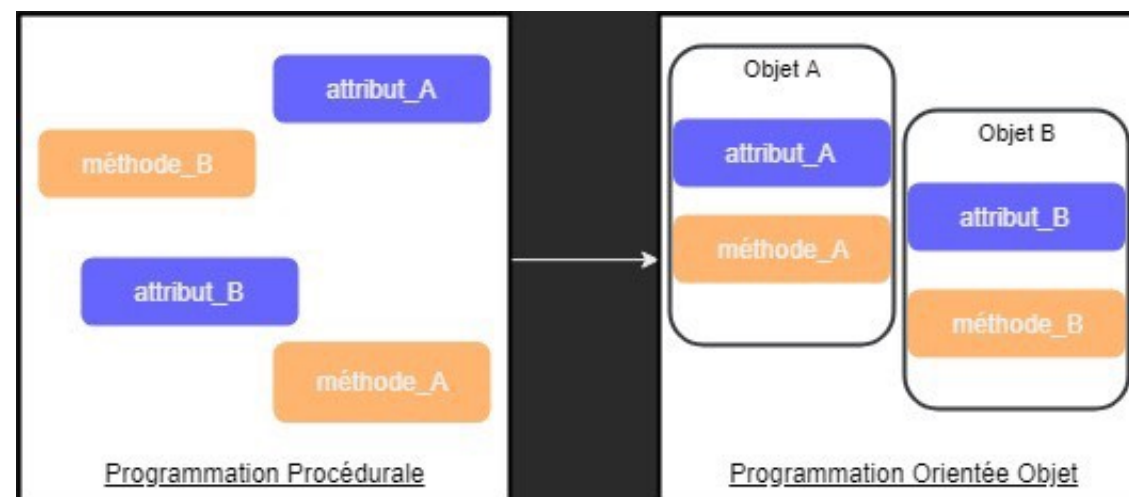
- ▶ En fonction du paradigme d'approche on peut classer les langages en plusieurs catégories :
 - ▶ **Les langages impératifs** (très majoritaire jusqu'aux année 1990) -> suites d'instructions qui modifie l'espace mémoire et les entrées/sorties.
Exemples : Basic, Cobol, C, etc.
 - ▶ **Les langages fonctionnels** forment une famille ou un programme est une fonction au sens mathématiques du terme. Les données peuvent être considérées comme des fonctions.
Exemples : Ocaml, Scala, Lisp, etc.

RAPPELS DES CONCEPTS DE LA P00

- ▶ En fonction du paradigme d'approche on peut classer les langages en plusieurs catégories :
 - ▶ **Les langages logiques** qui définissent un programme comme un ensemble de formules logiques qui se réfère à une logique formelle bien définie.
Exemple : Prolog (1972)
 - ▶ **Les langages de programmation orientée objet** qui organise le code autour d'objets. Ces objets sont des structures qui regroupent des données, appelées attributs, et des comportements, appelés méthodes.
Exemples : C++, Java

RAPPELS DES CONCEPTS DE LA POO

- ▶ Beaucoup de langages ne sont pas des langages purs dans le sens où ils peuvent appartenir à plusieurs de ces familles.
- ▶ C'est le cas du PHP, Python
- ▶ En effet en PHP on peut programmer en impératif ou en POO.



LA P00

- ▶ La programmation orientée objet est **un paradigme de programmation** qui organise le code autour d'**objets**. Ces objets sont des structures qui regroupent **des données**, appelées **attributs**, et **des comportements ou traitements**, appelés **méthodes**.
- ▶ Caractéristiques principales des langages de POO :
 - ▶ **Encapsulation** : Regroupe les données et les méthodes dans des objets, masquant les détails internes pour protéger les données de modifications externes non autorisées.
 - ▶ **Héritage** : Permet aux objets ou aux classes de dériver d'autres classes, réutilisant ainsi du code. Les sous-classes héritent des propriétés et des méthodes de la classe parent.

LA P00

► Caractéristiques :

- **Polymorphisme** : Permet aux objets d'être traités de manière interchangeable, même s'ils proviennent de classes différentes. Par exemple, une méthode peut être redéfinie dans des sous-classes pour fournir des implémentations spécifiques.
- **Abstraction** : Permet de se concentrer sur les aspects essentiels d'un objet en masquant les détails complexes. Cela aide les développeurs à travailler avec des concepts simplifiés et à éviter les erreurs.

CRÉATION D'OBJETS AVEC UNE CLASSE

Un objet

- Attention à ne pas confondre avec l'objet de la vie courante.
- Un objet est un concept à qui on attribue des valeurs grâce aux variables (on parle d'attributs) mais aussi des traitements ou actions grâce à des fonctions (on parle de méthodes)
- *Exemple d'objet* : une carte, un jeu de cartes, un voyage, un stage

Une classe

- La fabrication de l'objet est confiée à un moule qu'on appelle **une classe** en POO.
- C'est la classe qui contient les éléments (attributs et méthodes) qu'il faudra transmettre pour construire l'objet.
- Pour créer un objet on utilise l'opérateur **new()**
- Le fait de créer un objet à partir d'une classe s'appelle : **instancier**

Classe **Personnage**

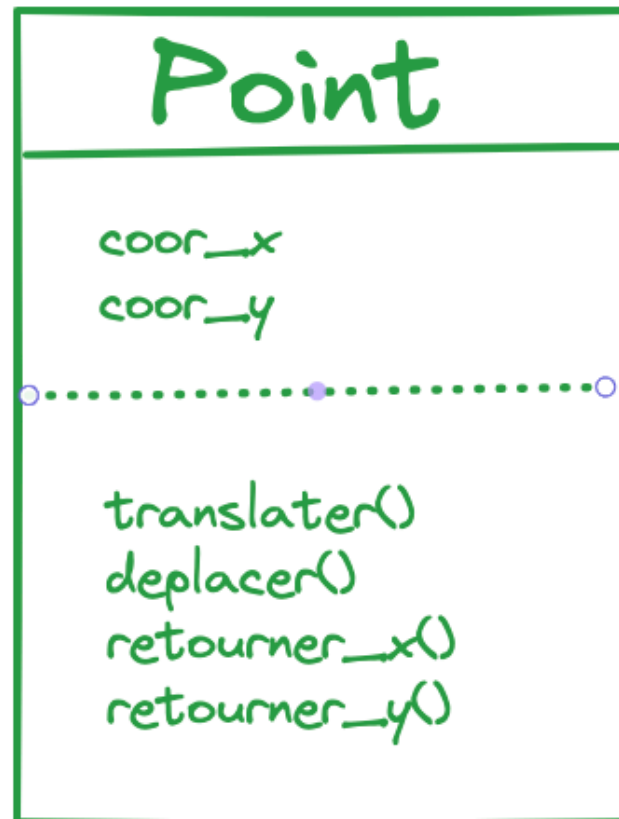
\$force
\$localisation
\$experience
\$degats

fonction frapper()
fonction gagnerExperience()
fonction deplacer()

Création de notre première classe

- Nous allons créer notre première classe. La classe Point.
- Attention une classe commence par une majuscule : Point
- Normalement une classe par fichier avec comme extension class.php
- La première question à se poser est : **qu'est-ce qui caractérise un point ?** Si on est dans un repère à deux dimensions on pourra dire ses coordonnées en x et y par exemple. La réponse à cette question nous donnera les attributs de notre classe.
- La deuxième question à se poser est : **qu'est-ce qu'on peut faire avec ce point ?** On pourra par exemple vouloir le déplacer. La réponse à cette question nous donnera la ou les méthodes de notre objet.

Création de notre première classe



Implémentation en PHP

Point.class.php

```
1  <?php
2  class Point {
3
4      // -- Propriétés --
5      private $_x;
6      private $_y;
7
8
9      // -- Constructeur --
10     public function __construct($x,$y) {
11         $this->_x = $x;
12         $this->_y = $y;
13     }
14
15     // Méthodes
16
17     // Méthode magique appelée quand on veut afficher un objet de type Point
18     public function __toString() {
19         return "Point x: ".$this->_x." y: ".$this->_y;
20     }
21
22     // Pour connaitre la valeur de x
23     public function getX() {
24         return $this->_x;
25     }
26
27     // Pour connaitre la valeur de y
28     public function getY() {
29         return $this->_y;
30     }
31
32     // Pour modifier la valeur de x
33     public function setX($x) {
34         $this->_x = $x;
35     }
36
37     // Pour modifier la valeur de y
38     public function setY($y) {
39         $this->_y = $y;
40     }
41
42 }
43
44 ?>
```




**MERCI POUR
VOTRE ATTENTION**