

PROGRAMMATION ORIENTÉE OBJET : POO

<https://www.php.net/manual/fr/oop5.intro.php>

Introduction à la Programmation Orientée Objet : POO

Rappels de la programmation :

- Tout programme contient 2 parties fondamentales :
 - **Les données** : dans un programme, les données sont véhiculées par des variables. Le stockage des informations se fait via une base de données.
 - **Les traitements** : ce sont les lignes d'instructions qui permettent d'utiliser les données (ajouter / modification / lecture)

Deux logiques de programmation

➤ La programmation séquentielle / procédurale :

- Cette méthode de programmation dissocie fortement les données et les traitements et chaque instruction se réalise indépendamment des données. Il n'y a pas de corrélation directe entre les deux.
- Les fonctions créées permettent de structurer le code et de le rendre réutilisable (concept de factorisation)
- La logique de programmation est basée sur l'enchaînement d'instructions

➤ La Programmation Orientée Objet :

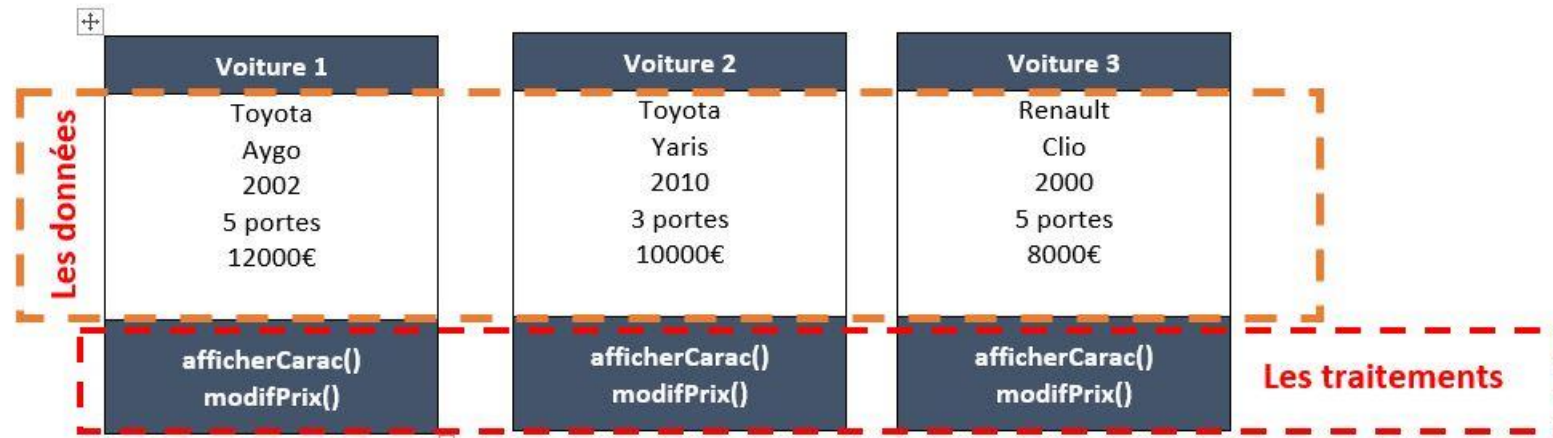
- Cette méthode permet de regrouper des informations et des traitements en un bloc ayant une « existence propre ».
- La logique est basée sur la création d'entités et non sur l'enchaînement d'instructions

La logique de la POO

- En POO, on va se poser des questions sur les éléments ayant une « existence propre ».
- Exemple : Vous souhaitez créer une application qui gère une concession automobile et va permettre aux clients d'acheter des voitures.
 - En POO, vous allez commencer par vous poser la question : quelles sont les entités qui auront une « existence propre » dans l'application ?
 - Les voitures
 - Les clients
 - Le parc automobile
 - Quelles sont les informations dont j'aurai besoin sur chacune de mes entités?
 - Voitures : modèle, marque, plaque d'immatriculation...
 - Client : nom, prénom, téléphone....
 - Parc automobile : adresse, nom, téléphone....

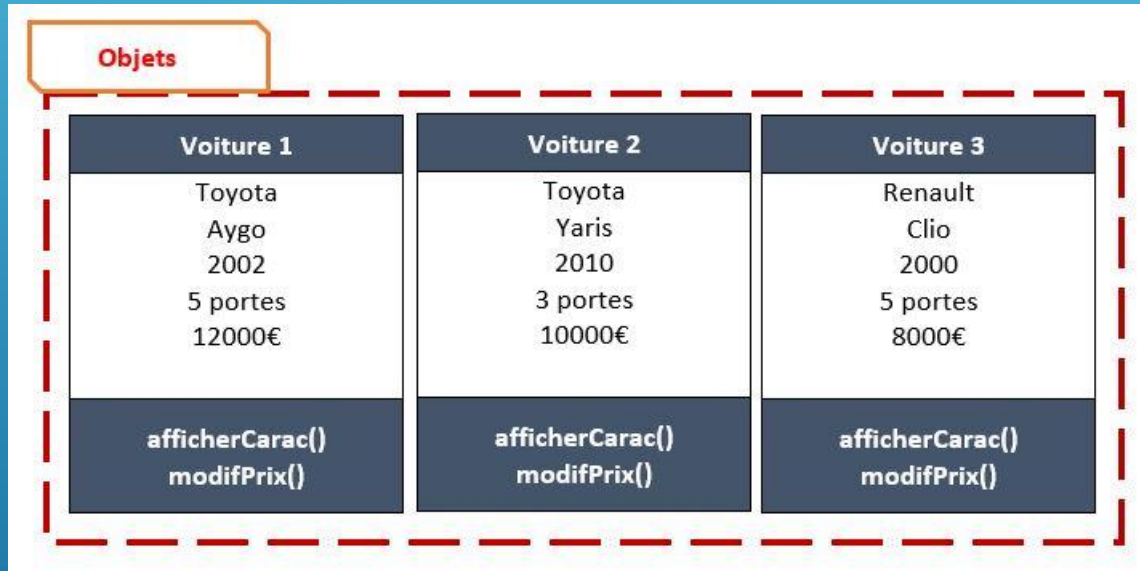
La logique de la POO

- Quelles sont les traitements associés à ces entités ?
 - Voiture : afficher les caractéristiques, modifier l'état du véhicule, modifier le prix....
 - Client : réserver, modifier les informations personnelles, commander....
 - Parc automobile : lister les voitures, ajouter une voiture, supprimer une voiture...
- On va créer des blocs logiques qui donneront des « objets ».
Exemple pour les voitures :



La logique de la POO

- De cette logique objet, on va pouvoir créer des « moules » permettant de grouper la structure des informations (données + traitements).
- Cette structure sera appelée une **classe**



Donc on va pouvoir créer un moule qui va permettre de générer des objets disposant d'une structure semblable.

Des variables aux objets

- En programmation procédurale nous utilisons des variables pour véhiculer des informations.
- Par exemple pour un personnage, on va décrire ses informations comme :

- Son nom
- Sa taille
- Son âge
- Sa vitesse
- Son sexe
-



Algorithme : algoVariable1

Variable

Nom : Chaîne de caractères

Taille : entier

Age : entier

Vitesse : entier

Sexe : booléen

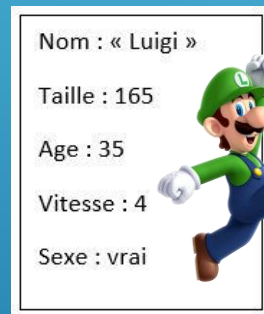
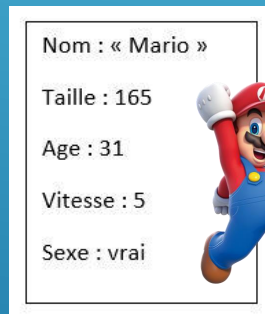
Début

//Instruction

Fin

Des variables aux objets

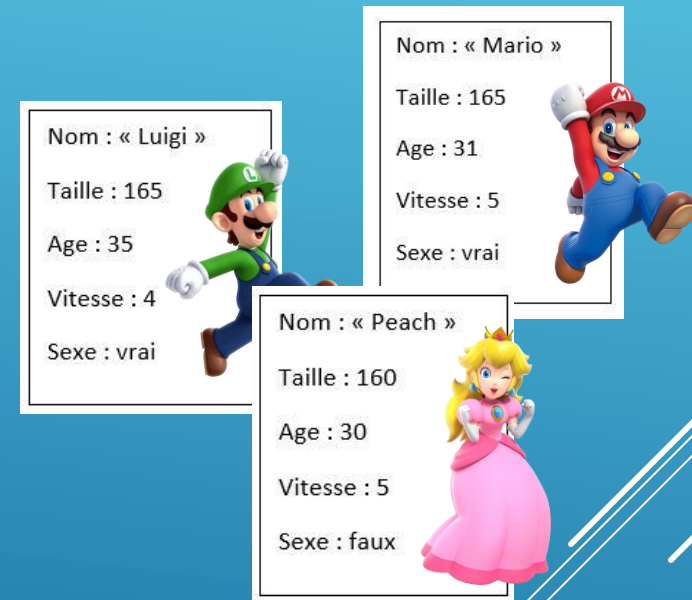
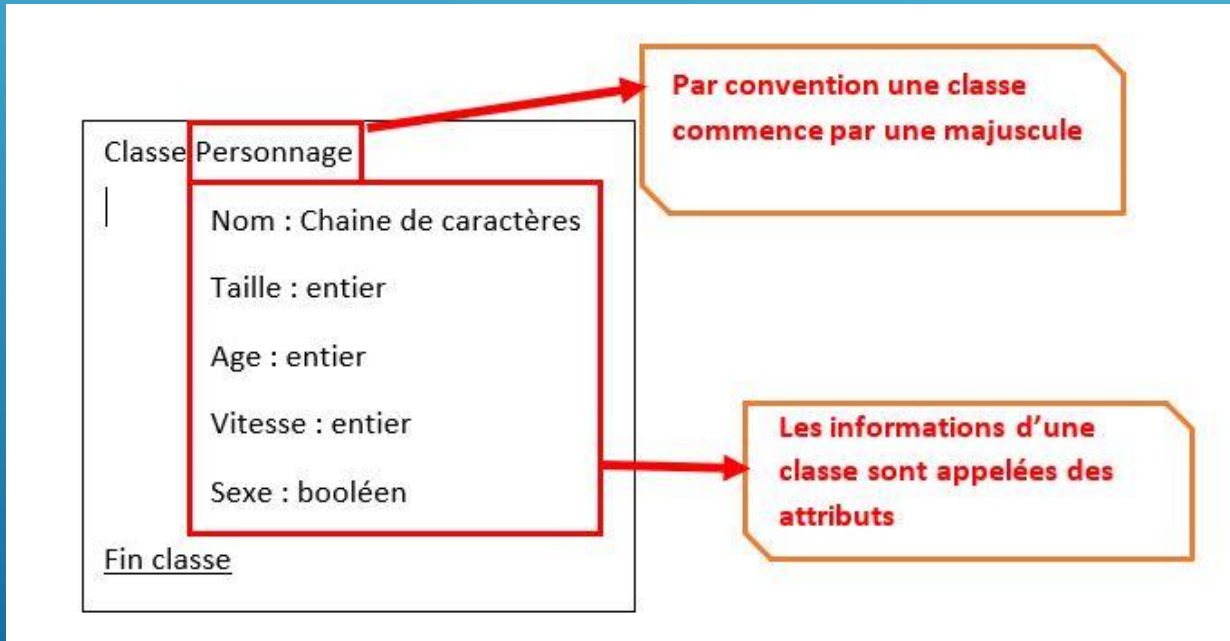
- Si nous ne disposons que d'un seul personnage il est « simple » à gérer, puisque nous n'avons à stocker qu'une variable par information.
- Mais si nous avons 3 personnages disposant des même types d'informations, il va devenir très compliqué à gérer et on pourrait créer des incohérences ou des erreurs dans nos programmes.



- Pour gérer l'ensemble de ces informations, il nous faut donc disposer de 5 variables multipliés par le nombre de personnages que nous avons.

Des variables aux objets

- Pour simplifier l'utilisation de toutes ces informations, on va utiliser une « super variable » qui s'appelle objet
- Et pour créer un objet, on va devoir créer une structure, cette structure s'appelle **une classe**



Des variables aux objets - Exercice

Proposer une classe permettant de créer les objets suivants :

- Riri est un guerrier avec une attaque de 10 et 100 points de vie des forces du bien
- Fifi est une archère avec une attaque de 5 et 50 points de vie des forces du bien
- Loulou est une archère avec attaque de 5 et 55 points de vie des du mal

Correction :

- Riri est un guerrier avec une attaque de 10 et 100 points de vie des forces du bien
- Fifi est une archère avec une attaque de 5 et 50 points de vie des forces du bien
- Loulou est une archère avec attaque de 5 et 55 points de vie des forces du mal

nom

rôle

attaque

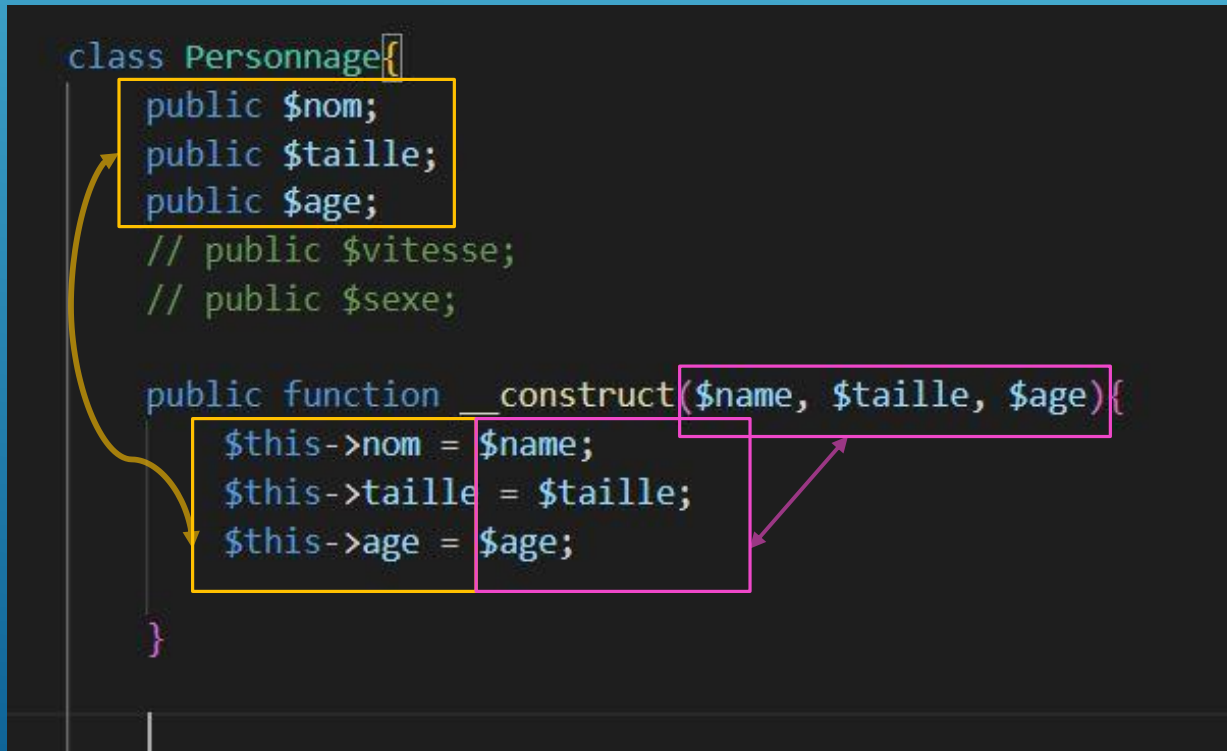
pv

forceDuBien

Création d'objet : Constructeur

- Pour créer des objets à partir d'une classe, il nous faut définir un constructeur
- Un constructeur est une fonction intégrée comme l'élément d'une classe
- L'instanciation est la création et l'initialisation d'un objet. On parle donc « d'instanciation » de la classe Personnage.

```
class Personnage{  
    public $nom;  
    public $taille;  
    public $age;  
    // public $vitesse;  
    // public $sexe;  
  
    public function __construct($name, $taille, $age){  
        $this->nom = $name;  
        $this->taille = $taille;  
        $this->age = $age;  
    }  
}
```



Visibilité des informations : Public / Private & Getter /Setter

```
class Personnage{  
    public $nom;  
    public $taille;  
    public $age;  
    // public $vitesse;  
    // public $sexe;  
  
    public function __construct($name, $taille, $age){  
        $this->nom = $name;  
        $this->taille = $taille;  
        $this->age = $age;  
    }  
  
    public function presentation(){  
        echo "Bonjour, je suis $this->nom ma taille est : $this->taille cm et mon age est : $this->age ans";  
    }  
}
```

```
$personnage1 = new Personnage("Mario",165,31);  
$personnage2 = new Personnage("Luigi",165,35);  
$personnage3 = new Personnage("Peach",160,30);
```

```
$personnage1->presentation();  
echo "<br/>";  
$personnage2->presentation();  
echo "<br/>";  
$personnage3->presentation();  
echo "<br/>";
```

EXERCICES POO 1 | Moyen – Les objets

5°) Reprendre l'exercice 4 Tableau associatif : Chien et Chat

- Modifier la structure pour utiliser des objets. Pour cela vous devez créer une classe qui permettra de définir la structure des animaux
- Créer une classe avec des attributs en public
- Réaliser les affichages demandés

Visibilité des informations : Public / Private & Getter /Setter

- Une information « public » est une information accessible à tout le monde et n'importe quand
- Une information « private » est une information qui n'est accessible que par l'objet lui-même.
- Tout attribut ou fonction peut-être déclaré en « public » ou « private ».
- En règle générale, on définira tous les attributs de la classe en PRIVATE, afin d'éviter que n'importe qui ne puisse faire n'importe quoi sur les objets.
- Afin de toujours accéder aux attributs, on mettra en place des fonctions permettant d'obtenir la valeur « Getter » et de modifier la valeur « Setter »