PHP Objet

Pour cette série d’exercices, créez un répertoire « serie-07 ». Créez un fichier PHP pour chaque exercice.

# Création d’une classe simple

Écrivez une classe représentant une ville. Elle doit avoir les propriétés nom et département et une méthode affichant « la ville X est dans le département Y ». Créez des objets ville, affectez leurs propriétés, et utilisez la méthode d’affichage.

Résultat attendu :

La ville de Nantes est dans le département : Loire Atlantique

La ville de Lyon est dans le département : Rhône

# Le constructeur

Modifiez la classe précédente en la dotant d’un constructeur. Réalisez les mêmes opérations de création d’objets et d’affichage.

# Getter et setters

Créez une classe représentant une personne. Elle doit avoir les propriétés nom, prénom et adresse, ainsi qu’un constructeur.

Une méthode getPersonne() doit retourner les coordonnées complètes de la personne.

Une méthode setAdresse() doit permettre de modifier l’adresse de la personne.

Créez des objets personne, et utilisez l’ensemble des méthodes.

# Les classes abstraites et l’héritage

Créez une classe abstraite représentant une personne. Elle déclare les propriétés nom et prénom et un constructeur.

Créez une classe client dérivée de la classe personne en y ajoutant la propriété adresse et une méthode getcoord() qui affiche les coordonnées complètes (nom, prénom et adresse) de la personne.

Créez une classe électeur dérivée de la même classe abstraite, et ajoutez-y deux propriétés bureau\_de\_vote et vote (un booléen), ainsi qu’une méthode avoter(), qui enregistre si une personne a voté dans la propriété vote.

Créez différentes personnes, affichez le résultat et faites voter ceux qui le peuvent.

# Les méthodes magiques

Implémentez un code pour chacune des méthodes magiques sur les classes de l’exercice précédent :

* \_\_construct();
* \_\_destruct() ;
* \_\_call() ;
* \_\_callStatic() ;
* \_\_get() ;
* \_\_set() ;
* \_\_isset() ;
* \_\_unset() ;
* \_\_toString() ;

Mettez un œuvre du code les faisant intervenir. Pour chacune d’entre elle, sauf le constructeur et toString, contentez-vous d’un echo avec un affichage.

# Json

A partir du code JSON suivant, instanciez un objet de la classe décrite par le-dit JSON. Créez une classe Employee bien sûr. Le code JSON est copié sous forme de texte dans une variable texte.

Rappel : json\_decode()

1. {
2. "employee": {
3. "name":       "sonoo",
4. "salary":      56000,
5. "married":    true
6. }
7. }

# Json et tableau

A partir du code JSON suivant créez un tableau de salariés. Crée une classe Employee bien sûr. Le code JSON est copié sous forme de texte dans une variable texte.

1. {"employees":[
2. {"name":"Shyam", "email":"shyamjaiswal@gmail.com"},
3. {"name":"Bob", "email":"bob32@gmail.com"},
4. {"name":"Jai", "email":"jai87@gmail.com"}
5. ]}

Créez un fichier employees.JSON et chargez les données depuis le fichier crée avec file\_get\_contents().