

## TD N°6

### Compétence : Développer des sites web dynamiques

#### Filière : Développement Digital

##### Exercice 1:

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.

##### Exercice 2:

1. Définir une classe Personne possédant les propriétés nom, prénom, et une méthode **présenter()** qui renvoie la chaîne de caractères "je m'appelle " suivie du nom et prénom.
2. Créer un programme de test qui instancie 2 personnes, puis affiche leurs descriptions.
3. Ajoutez dans la classe Personne une propriété date de naissance et une méthode **age()** renvoyant l'âge
4. Tester la classe Personne

##### Exercice 3:

Classe "Voiture" :

- 1- Ajouter une propriété "marque".
- 2- Ajouter une propriété "modele".
- 3- Ajouter une propriété "vitesse" qui représente la vitesse en km/h de la voiture.
- 4- Ajouter une méthode "accelerer" qui permet d'augmenter la vitesse de la voiture, en vérifiant que la vitesse maximale (par exemple 200 km/h) n'est pas dépassée.
- 5- Ajouter une méthode "freiner" qui permet de diminuer la vitesse de la voiture, en vérifiant que la vitesse minimale (par exemple 0 km/h) n'est pas dépassée.
- 6- Ajouter une méthode "afficherInfos" qui affiche la marque, le modèle et la vitesse de la voiture.

##### Exercice 4:

Classe "**CompteBancaire**" :

- 1- Ajouter une propriété "**solde**" et une méthode "**crediter**" qui permet d'ajouter de l'argent au compte.
- 2- Ajouter une méthode "**debiter**" qui permet de retirer de l'argent du compte, en vérifiant que le solde est suffisant.
- 3- Ajouter une méthode "**afficherSolde**" qui affiche le solde du compte.

### Exercice 5:

1. Définir une classe **Compte** représentant un compte bancaire.
2. Cette classe contiendra deux attributs privés, **montant** et **intérêt** (annuel), un constructeur prenant en paramètre un montant initial et un intérêt, une méthode **get\_montant** retournant le montant du compte, ainsi qu'une méthode **un\_an** modifiant le montant du compte au bout d'un an en fonction de l'intérêt.
3. Créer deux comptes, un d'un montant de 200 dirhams à 20% et l'autre d'un montant de 1000 dirhams à 2%.
4. Afficher le montant de chaque compte au bout de 10 ans.

### Exercice 6:

Vous avez déjà codé en Javascript vous aurez certainement vu l'objet string et les méthodes qui permettent de le manipuler (comme: length, charAt(), indexOf(), substring()...)

L'objectif de cet exercice est de créer une classe PHP au nom de **String** qui contient les méthodes du même nom que celles de Javascript et qui font les mêmes tâches. Autrement dit, à l'aide de la classe que vous aurez créée, quelqu'un qui s'y connaît en Javascript peut bien traiter les chaînes de caractères en PHP même s'il n'a pas de connaissances poussées en ce langage.

Pour récapituler voilà ce que font les méthodes que l'on va créer :

- **length**: Cet attribut retourne le nombre de caractères contenus dans la chaîne.
- **charAt()**: La méthode charAt(x) (avec A majuscule) permet de retourner le caractère qui se trouve à la position x passé en paramètre. Le paramètre est un entier qui commence de 0. La valeur 0 indexe le premier caractère de la chaîne.
- **indexOf()**: La méthode indexOf(car) (avec O en majuscule) permet de retourner la position du caractère car passé en paramètre. Si le caractère existe dans la chaîne, alors sa position (comprise entre 0 et la longueur de la chaîne - 1) est retournée, sinon (le caractère ne figure pas dans la chaîne) alors la valeur -1 est retournée. La méthode indexOf() peut accueillir un deuxième paramètre qui est un entier qui indique à partir de quel position de la chaîne on commencera la recherche du caractère passé en premier paramètre.
- **substring()**: La méthode substring(début,fin) permet d'extraire une partie de la chaîne de caractères commençant de la position début et finissant à la position fin-1.
- **split()**: La méthode split(occurrence) permet de retourner un tableau à partir des fractions de la chaîne de caractères obtenues en divisant celle-ci au niveau de l'occurrence.
- **toLowerCase()**: La méthode toLowerCase() permet de retourner la chaîne de caractère entièrement en minuscules.
- **toUpperCase()**: La méthode toUpperCase() permet de retourner la chaîne de caractère entièrement en majuscules.