1 - Créez un dossier ./sio-php-poo-bdd sur ./www (serveur WAMP) et placez y un fichier nommé Voiture.php. Dans ce fichier collez y le code suivant:

Pour accéder à un attribut ou une fonction d'un objet, on utilise le -> au lieu du .
Le constructeur ne porte pas le nom de la classe, mais s'appelle __construct().

2 - Créez des getter et des setter pour $couleur et $immatriculation

L'intérêt des setter est notamment de vérifier ce qui va être écrit dans l'attribut.
Vérifiez que l'immatriculation a ≤ 8 caractères dans le setter correspondant (sinon ne faites rien).

> strlen(string $string): int

3 - Remplissez afficher() qui permet d'afficher les informations de la voiture courante (Regardez le code du constructeur de la classe Voiture : comme en Java, on peut utiliser le mot-clé this mais suivi de ->) ;

```
$prenom = "Helmut"; echo <<< EOT
Texte à afficher
sur plusieurs lignes
avec caractères spéciaux
\t \n
 et remplacement de variables $prenom les caractères
suivants passent : " ' $ / \ ;
EOT;
```

Enregistrez votre travail à l'aide de git add et git commit. Aidez-vous toujours de git status pour savoir où vous en êtes.

4 - Créez un fichier testVoiture.php contenant le squelette HTML classique (html:5...)

Dans la balise <body>, on va vouloir créer des objets Voiture et les afficher :

Incluez Voiture.php à l'aide de require_once 'filename.php' :

- Mettez les balises d'interpretation Php <?php ?>
- Initialisez une variable $vehicule1 de la classe Voiture avec la même syntaxe qu'en Java

  `` `$vehicule1 = new Voiture("Renault", "Rouge", "152548")` ``

  `` `$vehicule1->afficher();` ``

- Affichez cette voiture à l’aide de sa méthode afficher()

5 - Créez un fichier formulaireVoiture.html et dans le body, insérez le formulaire suivant :

```
`<form method="post" action="creerVoiture.php">
<fieldset>
    <legend>Mon formulaire :</legend>
    <p>
        <label for="immat_id">Immatriculation</label> :
        <input type="text" placeholder="256AB34" name="immatriculation" id="immat_id" re
    </p>
    <p>
        <input type="submit" value="Envoyer" />
    </p>
</fieldset>
</form>`
```

Rappels :

L’attribut for de <label> doit contenir l’identifiant d’un champ <input> pour que un clic sur le texte du <label> vous amène directement dans ce champ.

l’attribut placeholder de <input> sert à écrire une valeur par défaut pour aider l’utilisateur.

6 - Créez des champs dans le formulaire pour pouvoir rentrer la marque et la couleur de la voiture.

Prenez l’habitude d’enregistrer régulièrement votre travail sous Git. Vous pouvez utiliser la commande git log pour voir l’ensemble de vos enregistrements passés.

7 - Créez un fichier creerVoiture.php :

Aidez-vous si nécessaire des fichiers dans ./ressources/* pour savoir comment récupérer l’information envoyée par le formulaire avec la méthode POST.
Vérifiez que creerVoiture.php reçoit bien des informations dans le query string. Pour cela, vérifiez que le tableau $_POST n’est pas vide.
En reprenant du code de testVoiture.php, faites que creerVoiture.php affiche les informations de la voiture envoyée par le formulaire.

8 - Essayez de retrouver l’information envoyée par le formulaire avec les outils de développement F12 (Onglet Réseau).

Conseil : Git est ton ami :

```
git commit -m "exo 8"
```

9 - Vous allez programmer les classes d’un site de covoiturage, dont voici la description d’une version minimaliste:

Utilisateur : Un utilisateur possède des champs propres (login, nom, prenom)
Trajet : Une annonce de trajet comprend :

- un identifiant unique ***id***,
- les détails d’un trajet (un point de départ ***depart*** et un point d’arrivée ***arrivee***),
- des détails spécifiques à l’annonce comme une date de départ ***date***,
- un nombre de places disponibles ***nbplaces***,
- un prix ***prix***,
- et le login du conducteur ***conducteur_login***,

Astuce : Pour éviter de taper 7 getters, 7 setters et un constructeur avec 7 arguments pour Trajet, nous allons coder :

- des getters génériques ***get($nom_attribut)*** qui renvoient l’attribut de nom ***$nom_attribut***.
  Utilisez la syntaxe suivante pour accéder à l’attribut de nom ***$nom_attribut*** de l’objet ***$objet*** :
  `$objet->$nom_attribut`
- des setters génériques ***set($nom_attribut, $valeur)*** ;
  un constructeur ***__construct($data)*** qui prend un tableau dont les index correspondent aux attributs de la classe.

10 - Avez-vous pensé à enregistrer vos modifications sous Git ? Faites le notamment en fin de TD pour retrouver plus facilement où vous en êtes la prochaine fois.

Note : Vous pouvez faire git diff à tout moment pour voir les modifications que vous avez faites depuis la dernière validation (commit).