# TD - Exercices

1. **Introduction** 
   1. Tout premier programme : afficher les caractéristiques PHP et Apache

Voyons si notre configuration PHP & Apache est correcte.  
Afficher dans une page toutes informations de configuration du serveur Web.

*Utile :*phpinfo();

1. Punition

Écrire un script qui écrit 500 fois « *Je dois faire des sauvegardes régulières de mes fichiers.* »

1. Factoriel

Écrire un script qui donne le résultat d’un factoriel. Donner un nombre dans un formulaire, en validant le formulaire, la page vous retourne le résultat du factoriel.

1. **Mise en bouche : tableaux** 
   1. **Basique**

Créer un tableau ayant pour valeur les noms des douze mois de l’année. Afficher un tableau associant à chaque mois de l’année le nombre de jours du mois. (On supposera que l’année n’est pas bissextile.)

* 1. **(A peine) moins basique**

Il s’agit ici de travailler sur la manipulation de tableaux d'entiers en PHP. Prenons un tableau stockant des notes d’étudiants, par exemple de L3 à un examen.

Écrire les fonctions PHP qui :

1. Affiche toutes les notes
2. calcule la moyenne obtenue,
3. indique combien de personne ont eu la moyenne
4. Indique combien de personne sont recalés
5. teste si la note 20 est présente
6. détermine la meilleure note obtenue.
7. **Table de multiplication**

Créer un script PHP permettant d’afficher une table de multiplication dynamique.   
La taille de la table doit être modifiable.  
Bonus :

Affichons quelque chose d’autre que du noir&blanc à l’écran. Utiliser du CSS pour un rendu.

*Utile :*

* *Opérateur modulo*
* Styles CSS :

|  |
| --- |
| <style type=”text/css”>  table  {  border- collapse : collapse ;  }  th,td  {  border : 1px solid black;  width : 25px;  height : 25px;  }  th  {  background-color : #FFFFFF;  }  .impair  {  background-color : #A9F5A9;  }  .pair  {  background-color : #D8F6CE;  }  </style> |

1. **Sessions**
   1. **Formulaire**

Ecrire un formulaire dans une page web (enregistrement.php), demandant le prénom de l’utilisateur.

Ecrire une page, page1.php. Cette page, ouverte dans un second onglet du navigateur doit afficher la valeur saisie dans l’onglet « enregistrement.php » par simple actualisation.

Si vous afficher la page1.php sans avoir saisie le le formulaire, un message doit vous l’indiquer, sinon la page doit afficher le prénom saisi dans la page enregistrement.php.

* 1. **Jeu : deviner un chiffre**
* L’ordinateur choisit un nombre entier aléatoire compris entre deux nombres.
* Le joueur a 7 coups pour trouver ce nombre.
* A chaque tentative, si le nombre entré par l’utilisateur est différent du nombre aléatoire, le programme indique si le nombre saisi est plus petit ou plus grand que le nombre aléatoire.

*Peut être utile :*

$nb = mt\_rand(1,100);  
$\_SESSION

1. **Objet : formulaire et objets**
   1. Classe de base

Créez une classe nommée « form » représentant un formulaire HTML.

* Le constructeur doit créer le code d’en-tête du formulaire en utilisant les éléments <form> et <fieldset>.
* Une méthode settext() doit permettre d’ajouter une zone de texte.
* Une méthode setsubmit() doit permettre d’ajouter un bouton d’envoi. Les paramètres de ces méthodes doivent correspondre aux attributs des éléments HTML correspondants.
* La méthode getform() doit retourner tout le code HTML de création du

formulaire.

Pour tester, écrire une page instanciant des objets form ; Ajoutez par exemple deux zones de texte et un bouton d’envoi.

* 1. Classe dérivée

Créez une sous-classe nommée form2 en dérivant la classe form de l’exercice précédent. Cette nouvelle classe doit permettre de créer des formulaires ayant en plus des boutons radio et des cases à cocher. Elle doit donc avoir les méthodes supplémentaires qui correspondent à ces créations. Créez des objets, et testez le résultat.

1. **Classe et PDO : réalisation d’un singleton**

Ecrire une classe dérivant de PDO afin de pouvoir accéder à une base de données et de façon à pouvoir la réutiliser.

Ecrire une base et une table de test afin de valider le fonctionnement de la classe.

*Définition d’un singleton :*

*En*[*génie logiciel*](http://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel)*, le singleton est un*[*patron de conception*](http://fr.wikipedia.org/wiki/Patron_de_conception)*(design pattern) dont l'objet est de restreindre l'instanciation d'une classe à un seul objet (ou bien à quelques objets seulement). Il est utilisé lorsque l'on a besoin d'exactement un objet pour coordonner des opérations dans un système*

1. **Parser CSV**

Ecrire un fichier CSV simpliste :

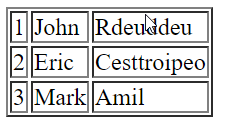
id,first\_name,last\_name

1,John,Rdeuddeu

2,Eric,Cesttroipeo

3,Mark,Amil

Ecrire un script PHP permettant de parser ce fichier CSV et ainsi d’afficher à l’écran :



**Tips :**

Utilisation de composer : https://getcomposer.org/download/

L’utilisation d’une classe existante : par exemple JamesGordo\CSV\Parser

1. **Protéger l'accès à une page**

Il s’agit de verrouiller l’accès à une page : accès limité aux personnes autorisées.

Lors de l’accès à cette page, il sera proposé un formulaire d’authentification. L’utilisateur à la possibilité de demandé l’accès (inscription).

Il sera possible via une zone super administrateur d’activer l’accès aux utilisateurs l’ayant demandé.

Utiliser la classe dérivant de PDO de l’exercice 9.



1. **Chat**

Ecrire un chat simple.

Il sera composé d’un formulaire permettant de poster des messages et de la liste des derniers messages postés.

Il sera judicieux de réutiliser la classe « form » créé dans un exercice plus tôt, ainsi que la classe permettant d’accéder à la base de données.

*Les fichiers à écrire :*

*chat.php : contient le formulaire permettant d'ajouter un message et liste les n derniers messages ;*

*chat\_post.php : insère le message reçu dans la base de données puis redirige vers chat.php.*

***ID pseudo* message**

La table en base devra être au minimum de la forme :

|  |  |  |
| --- | --- | --- |
| ***id*** | ***login*** | ***message*** |
| 1 | Vincent | Salut ça va ? |
| 2 | Pauline | Non |