

R3.01 - ProgWeb


TP 2 - Trombinoscope

Introduction

Ce TP va vous permettre de mettre en œuvre ce qui a été vu en TD 2 :

- Formulaires
- Sessions et cookies

Vous allez créer un petit site Web de gestion des listes de classes d'une école et de génération de trombinoscopes en PDF et de planches de photos individuelles.

Vous noterez qu'il y a des suggestions de fonctions qui peuvent vous être utiles en plus de celles que vous avez déjà découvertes dans les TD. Ces suggestions sont matérialisées dans la marge par un .

Lisez les paragraphes d'introduction avant de vous lancer. Le travail ne commence réellement qu'à l'**Exercice 1** !

Pages à double fonction

Pour ce TP, vous allez mettre en place des pages à double fonction¹. C'est un mécanisme assez commun qui consiste, pour une page PHP (donc un script), à proposer l'affichage d'un formulaire dont l'action (le script appelé à la soumission du formulaire) est le même script qui a servi à afficher le formulaire. C'est un peu comme si le script s'appelait lui-même.

Le script doit donc identifier s'il est exécuté dans le cadre de la soumission du formulaire ou s'il est exécuté pour simplement afficher le formulaire, comme c'est le cas au 1^{er} affichage de la page.

S'il est appelé dans le cadre de la soumission, il n'affiche pas le formulaire mais un contenu correspondant au traitement des données soumises. Par exemple, avec un formulaire de recherche ou de filtrage, on pourrait afficher une liste de résultats.

S'il n'est pas appelé dans le cadre de la soumission, il affiche le formulaire et évidemment rien concernant la liste de résultats.

C'est une page à double fonction. Ça permet de centraliser le code au sein d'un seul script au lieu de multiplier les scripts en fonction des actions possibles. Et on peut étendre ce mécanisme en codant ainsi des pages multi-fonction.

¹ Au sens "double usage".

Comment distinguer les cas ?

Pour savoir dans quelle situation le script à double fonction s'exécute (affichage du formulaire ou traitement du formulaire soumis), vous utiliserez une astuce qui consiste à vérifier si la superglobale `$_GET` ou `$_POST` (en fonction du choix de méthode que vous ferez) contient ou non une information issue du formulaire. Dans l'affirmative, vous êtes dans le cas de la soumission, sinon c'est le cas de l'affichage du formulaire.

Données


Comme avec le TP 1, vous n'utiliserez pas de BDD dans ce TP. Toutes les informations seront stockées dans des fichiers et les photos dans un dossier dédié. Le format retenu cette fois-ci est le CSV (Comma Separated Values) : des champs texte séparés par des virgules. 1 ligne = 1 enregistrement (1 fiche d'identité).

Voici les informations que vous allez devoir collecter, les contraintes de validité à vérifier pour chacune d'elles² et le type de champ à proposer dans le formulaire de saisie.

Dans un premier temps, vous ne ferez les tests que pour les champs **Nom**, **Sexe** et **Classe**. Une fois le TP terminé, vous reviendrez ici terminer les autres tests.

- Numéro INSEE : champ texte d'exactly 13 chiffres + 2 chiffres de clé
Formule de Luhn : clé = $97 - ((\text{n° INSEE}) \bmod 97)$. Ce sera l'ID unique qui identifie un élève. Les images seront nommées aussi à l'aide de ce numéro.
- **Nom** : champ texte de 40 symboles maximum → lettres, apostrophe, un ou plusieurs tirets (-) et zéro ou un double tiret (--), le tout à convertir en majuscules
- Prénom : champ texte de 40 symboles maximum → lettres, apostrophe, le tout à convertir en capitalisé : 1^{ère} lettre du prénom ou lettre suivant un tiret, en majuscule, le reste en minuscule.
- Email : champ texte de 255 symboles maximum → doit contenir un et un seul @, et le domaine doit contenir au moins un **.com**, **.net**, **.org** ou **.fr**
- **Sexe** : boutons-radio H/F, pas de choix par défaut, choix obligatoire
- **Classe** : liste déroulante 6^{ème} à Terminale, choix obligatoire
- Une photo : format JPEG et ne dépassant pas 200 Ko.

Vous nommerez votre fichier de données de travail **data.csv**.

 **strstr()**, **strtoupper()**, **strtolower()**, **substr()**

Exercice 1 - Remplissage de la base

Vous avez, sur Moodle, un fichier **INSEE.csv** qui **N'EST PAS** votre base de données **data.csv**, mais qui va vous servir à l'alimenter pour démarrer.


² Regex acceptées :

Jetez un œil à ce fichier pour y identifier les différents champs utiles.

Créez une page **import.php**, à double fonction, assurant :

- l’affichage d’un formulaire permettant de soumettre un fichier à importer. Vous testerez avec **INSEE.csv**.
- l’import des données du fichier soumis en créant autant de fiches qu’il y a de lignes dans le fichier. Attention, le fichier de test **INSEE.csv** peut contenir des erreurs sur les codes INSEE. N’importez pas aveuglément, votre script doit faire les vérifications de rigueur avant d’ajouter chaque fiche.
- l’affichage des erreurs détectées lors de l’import.

Votre fichier **data.csv** sera placé aussi avec vos scripts, en sachant que c’est acceptable ainsi uniquement à des fins de test.

 **file()**, **explode()**

Photos

Par simplicité, nous vous avons préparé des photos pour chaque fiche de **INSEE.csv**. Vous pouvez les placer directement et manuellement dans un dossier **photos/** créé par vous au préalable, dans le même répertoire que vos scripts.

Exercice 2 - Création d’une fiche

Vous avez importé une base depuis un fichier. C’est pratique pour démarrer, mais il faut aussi un outil de saisie manuelle pour créer des fiches ponctuellement.

Créez une page **crea.php**, à double fonction, assurant :

- l’affichage d’un formulaire de création d’une nouvelle fiche d’identité avec tous les champs cités dans le paragraphe **Données**, ainsi que la possibilité d’envoyer en même temps une photo.
- le traitement des données soumises par le formulaire, avec vérification de cohérence et nettoyage, si nécessaire, des champs pour lesquels c’est possible.
- la gestion des erreurs : ne créez une fiche que si tout est OK dans les données soumises. Vous afficherez aussi des messages adaptés en fonction des circonstances rencontrées.
- le placement, dans le dossier **photos/**, des photos soumises.

Vous pourrez générer des photos sur **<https://avatarmaker.com>**.

Voici aussi quelques numéros INSEE valides pour faire des tests d’ajout de fiches :

- **1071103123229 / 51**
- **2071095085039 / 92**
- **1070345120534 / 29**
- **2071209033041 / 27**
- **1070956230253 / 10**
- **2071201189941 / 95**

Exercice 3 - Liste des classes


Créez une page **liste-classe.php**, à double fonction, assurant :

- l’affichage d’un formulaire permettant de choisir une classe dans une liste déroulante (6^{ème} à Terminale).
- l’affichage d’une liste des élèves de la classe choisie.

Affichage de la liste d’élèves

Vous ferez l’affichage de la liste d’élèves dans une **<table>** dont les lignes seront formatées ainsi :

Classe de Terminale

H/F	NOM	Prénom	
email			

Exercice 4 - Générateur de PDF

Principe de fonctionnement

Vous allez adapter le script **liste-classe.php** pour pouvoir sélectionner un ou plusieurs élèves dans une ou plusieurs classes, afin de générer un trombinoscope de ces élèves et d’en générer un PDF.

La sélection se fait par l’ajout d’un champ “case à cocher” dans la liste. Ces cases à cocher font partie d’un grand formulaire dans le script liste-classe.php.

Attention, il y a déjà un formulaire dans ce script quand il est exécuté en mode “formulaire de choix d’une classe”. Il s’agit donc de placer un autre formulaire en mode “liste d’une classe choisie”. Ce sont deux formulaires distincts et donc deux traitements distincts. Le script **liste-classe.php** qui était un script à double fonction, devient désormais un script à triple fonction ! Pas d’inquiétude, vous aviez un **if...else**, et maintenant vous allez avoir un **if...elseif...else**, ce n’est pas plus compliqué.

A chaque soumission du formulaire, les ID élèves choisis sont à ajouter dans une liste stockée dans la session.

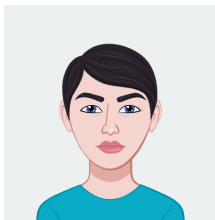
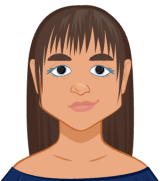



Question 1

Modifiez le script **liste-classe.php** pour tenir compte de ce qui a été décrit dans le **Principe de fonctionnement**.

Question 2

Créez une page **trombino.php** qui affiche la liste (<table> ou <div> avec un style **display: grid** ?) de tous les élèves sélectionnés et présents dans la session.

Voici comment présenter les informations :

				
NOM Prénom Classe	NOM Prénom Classe	NOM Prénom Classe	NOM Prénom Classe	NOM Prénom Classe

Votre page **trombino.php** doit prévoir un lien **Nouveau** qui appelle le script **trombino.php** avec un paramètre spécial permettant la purge de la session pour repartir à vide.

Question 3

Prévoyez un stylage CSS spécifique pour le mode impression dont voici un début :

```
@media print {  
  @page {  
    size: A4;  
    margin: 20mm;  
  }  
  body {  
    margin: 0;  
    padding: 0;  
  }  
  ... à compléter...  
}
```

Le mieux est d'exprimer les dimensions en **mm**. Vous aurez plus de facilité pour positionner précisément les cellules de tableau ou de la **grid**.

Pour générer un PDF du trombinoscope, faites un "Imprimer en PDF" sur votre page.