

# Programmation Web Dynamique

## Travaux dirigés et pratiques PHP – Série N°4

L'objectif de ce TP est de manipuler les formulaires en utilisant les fonctionnalités des chaînes de caractères et des fichiers en PHP.

**Pré-requis** : le langage de description html et le langage C

### Exercice N°1

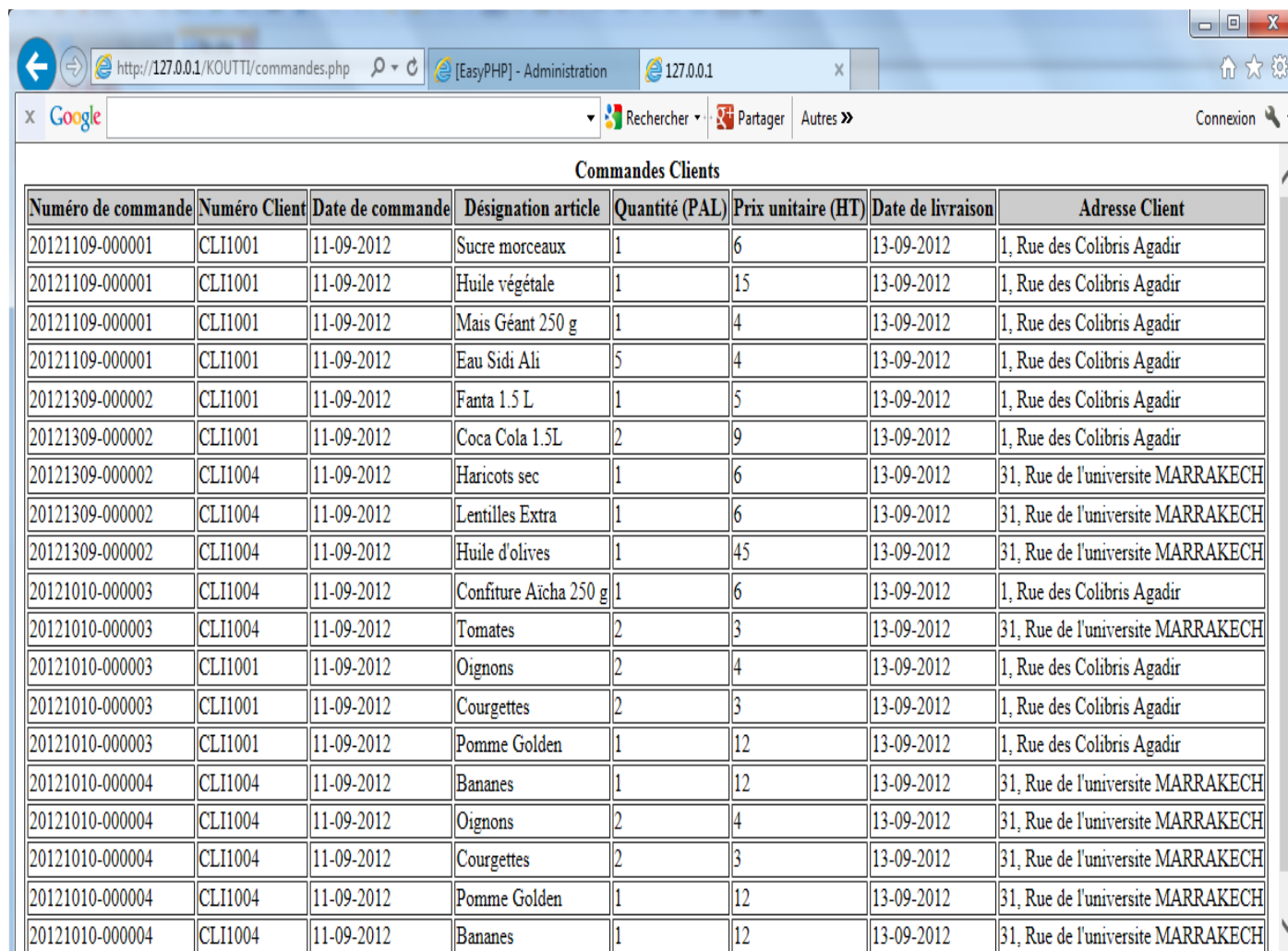
Créer une fonction qui prend en paramètre une chaîne \$string et un délimiteur \$car. La fonction aura comme tâche de couper la chaîne en utilisant le délimiteur et renvoie le résultat sous forme d'un tableau.

### Exercice N°2 : Manipuler les chaînes de caractères

Un fichier de données contient les commandes envoyées par tous les clients (dans notre cas deux clients "CLI1001" et "CLI1004").

Écrire un script PHP qui permet d'archiver les commandes du client CLI1001 dans un fichier de données pscde01\_CLI1001.txt et celle du client CLI1004 dans le fichier psccl01\_CLI1004.txt.

Utiliser au maximum les fonctions de manipulations des chaînes de caractères vus en cours.



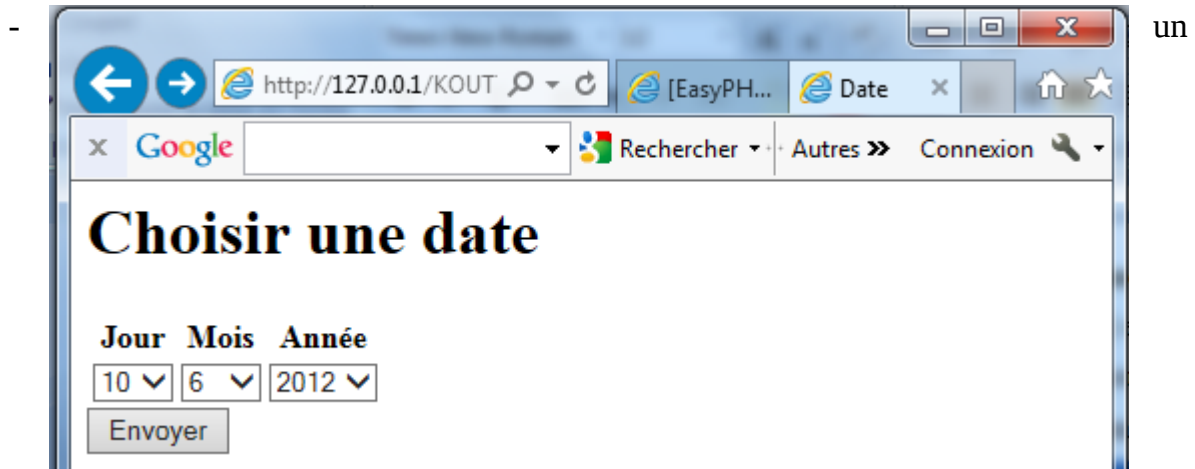
The screenshot shows a web browser window with the address bar displaying 'http://127.0.0.1/KOUTTI/commandes.php'. The page title is '[EasyPHP] - Administration' and the version is '127.0.0.1'. The main content area is titled 'Commandes Clients' and contains a table with 8 columns: 'Numéro de commande', 'Numéro Client', 'Date de commande', 'Désignation article', 'Quantité (PAL)', 'Prix unitaire (HT)', 'Date de livraison', and 'Adresse Client'. The table lists 20 orders, alternating between clients CLI1001 and CLI1004. The orders for CLI1001 are numbered 20121109-000001 to 20121010-000004, and the orders for CLI1004 are numbered 20121309-000002 to 20121010-000004. The table is displayed in a grid format with alternating row colors.

Numéro de commande	Numéro Client	Date de commande	Désignation article	Quantité (PAL)	Prix unitaire (HT)	Date de livraison	Adresse Client
20121109-000001	CLI1001	11-09-2012	Sucre morceaux	1	6	13-09-2012	1, Rue des Colibris Agadir
20121109-000001	CLI1001	11-09-2012	Huile végétale	1	15	13-09-2012	1, Rue des Colibris Agadir
20121109-000001	CLI1001	11-09-2012	Mais Géant 250 g	1	4	13-09-2012	1, Rue des Colibris Agadir
20121109-000001	CLI1001	11-09-2012	Eau Sidi Ali	5	4	13-09-2012	1, Rue des Colibris Agadir
20121309-000002	CLI1001	11-09-2012	Fanta 1.5 L	1	5	13-09-2012	1, Rue des Colibris Agadir
20121309-000002	CLI1001	11-09-2012	Coca Cola 1.5L	2	9	13-09-2012	1, Rue des Colibris Agadir
20121309-000002	CLI1004	11-09-2012	Haricots sec	1	6	13-09-2012	31, Rue de l'universite MARRAKECH
20121309-000002	CLI1004	11-09-2012	Lentilles Extra	1	6	13-09-2012	31, Rue de l'universite MARRAKECH
20121309-000002	CLI1004	11-09-2012	Huile d'olives	1	45	13-09-2012	31, Rue de l'universite MARRAKECH
20121010-000003	CLI1004	11-09-2012	Confiture Aicha 250 g	1	6	13-09-2012	1, Rue des Colibris Agadir
20121010-000003	CLI1004	11-09-2012	Tomates	2	3	13-09-2012	31, Rue de l'universite MARRAKECH
20121010-000003	CLI1001	11-09-2012	Oignons	2	4	13-09-2012	1, Rue des Colibris Agadir
20121010-000003	CLI1001	11-09-2012	Courgettes	2	3	13-09-2012	1, Rue des Colibris Agadir
20121010-000003	CLI1001	11-09-2012	Pomme Golden	1	12	13-09-2012	1, Rue des Colibris Agadir
20121010-000004	CLI1004	11-09-2012	Bananes	1	12	13-09-2012	31, Rue de l'universite MARRAKECH
20121010-000004	CLI1004	11-09-2012	Oignons	2	4	13-09-2012	31, Rue de l'universite MARRAKECH
20121010-000004	CLI1004	11-09-2012	Courgettes	2	3	13-09-2012	31, Rue de l'universite MARRAKECH
20121010-000004	CLI1004	11-09-2012	Pomme Golden	1	12	13-09-2012	31, Rue de l'universite MARRAKECH
20121010-000004	CLI1004	11-09-2012	Bananes	1	12	13-09-2012	31, Rue de l'universite MARRAKECH

### Exercice 3 : Interface de saisie d'une date

Le formulaire de saisie (voir figure 1) est composé de :

- trois listes pour la saisie du jour, du mois et de l'année



bouton d'envoi des informations saisies au serveur pour validation

Figure 2 Interface de saisie d'une date

### Validation du formulaire

L'utilisateur commence par choisir une date, il valide ensuite en cliquant sur le bouton « Validez ». Le script *valideDate.php* est alors appelé en affichant la date saisie (voir figure ci-dessous)

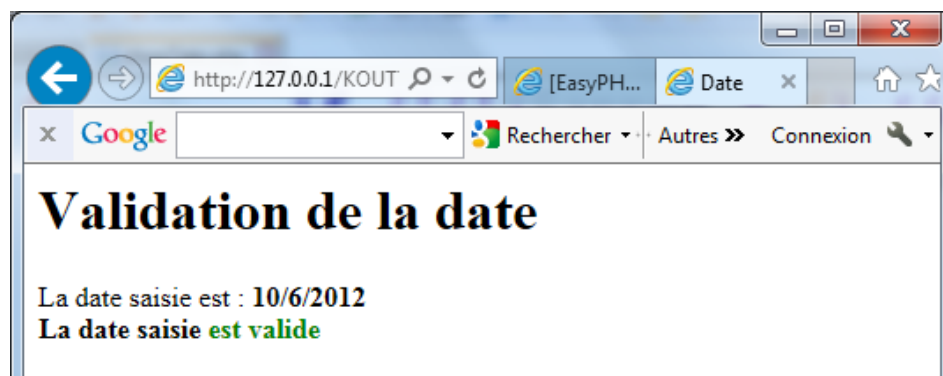


Figure 3 Validation d'une date

Lorsque l'utilisateur saisit une date incorrecte, l'application doit afficher un message d'erreur signalant l'anomalie (voir figure ci-dessous)

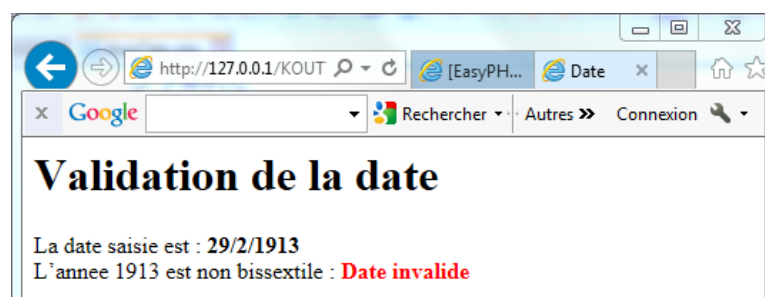


Figure 4 Gestion des erreurs

## Exercice 4 Authentification avec mail et mot de passe

Utiliser au maximum les fonctions de manipulations des chaînes de caractères vus en cours.

Le travail à faire consiste à réaliser un système d'authentification composé de deux pages **accueil.php** et **authentification.php**. Le page **accueil.php** contient un formulaire de deux champs : login et mot de passe. Le login est une adresse mail et le mot de passe doit vérifier les conditions suivante :

1. Comporter au moins 8 caractères
2. Avoir au moins un chiffre de 0 à 9
3. Comporter un caractères spécial : \* & % \$ # @ ! ? /
4. Avoir au au moins une lettre majuscule

Le mots passe est déclaré invalide si une de ces conditions n'est pas satisfaite.

Dans la page cible **authentification.php** :

A) En utilisant les motifs vus dans le cours, créer une fonction qui retourne un TRUE si le login est une adresse mail valide ( sous la forme [tata@cocomail.com](mailto:tata@cocomail.com) ) et retourne FALSE sinon.

B) Créer une fonction qui retourne un TRUE si le mot de passe est valide et FALSE sinon.

Utiliser au maximum les fonctions de manipulations des chaines de caractères vus en cours.

Dans un fichier **login.txt** est stocké une liste login ( adresses mail) et mots de passe.

C) Écrire un script php qui permet de vérifier d'abord si les deux fonctions de A) et B) ont retournées TRUE, ensuite il vérifie si une adresse et un mot de passe dans le fichier **login.txt**

correspondent au login et mot de passe reçu par le formulaire. Si c'est le cas, le message « Authentification réussie » est affiché. Sinon suivant les cas un des messages est affiché : « Login inexistant » « Mot de passe invalide ».