

# R3.01 - ProgWeb


## TP 1 - Mini market

### Introduction

Ce TP va vous permettre de mettre en œuvre ce qui a été vu en TD 1 :

- Scripts autonomes
- Pages Web
- Tableaux
- Génération de listes et de tables dynamiques
- Passage de paramètres sur URL
- Pagination
- Sérialisation

Vous allez créer un petit site Web de gestion des stocks d'une boutique en ligne.

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 

### Exercice 1 - Création de la “BDD”

Ce n'est pas un SGBD que vous allez utiliser mais un simple fichier dans lequel vous lirez et écrirez des données sérialisées.

Les pages PHP de votre petit site Web (ce sera sans doute toutes les pages) qui auront besoin de manipuler (lire et/ou écrire) les données de la boutique devront :

- Commencer par désérialiser les données depuis le fichier servant de “bdd”
- Exploiter les données (les afficher et/ou les modifier, en fonction du script)
- Si les données ont été modifiées, sérialiser et écrire le nouvel état dans le fichier servant de “bdd”

**Avez-vous bien compris ce qui vient d'être décrit ?**

**Dans le doute, faites-vous expliquer le mécanisme attendu !**

Pour l'instant vous n'avez pas encore ce fichier qui va servir de “bdd”, il va falloir le créer à partir d'une liste d'articles.

Vous disposez sur Moodle d'un fichier **articles** contenant cette liste d'articles formatée ainsi, 1 ligne par produit :

- Code : numérique
- Libellé : texte
- Prix HT : réel
- Taux TVA : réel

Écrivez un script autonome **init.php** qui va :

- Lire le contenu du fichier **articles**. Conseil : utilisez **file()** pour charger votre fichier dans un tableau et parcourez ensuite votre tableau. Chaque cellule du tableau = un produit. A vous de découper les informations (**explode()** est peut-être utile pour ça). Ceci a déjà été vu en R1.04 l'an dernier.
- Construire un tableau d'articles structuré de façon judicieuse pour une utilisation aisée ensuite. Vous avez libre choix.
- Prévoyez aussi les champs **stock** et **vendu**, qui ne font pas partie des données importées depuis le fichier **articles** et qui doivent donc être initialisés à **0**.
- Sérialiser le tableau d'articles dans un fichier **data**.

Lancez votre script et vérifiez qu'il vous a bien créé un fichier **data** avec un contenu qui semble cohérent (vous identifiez bien des données sérialisées).



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

## Exercice 2 - Liste des articles

Créez une page **liste.php** qui affiche les articles dans une table avec les colonnes :

- Code (centré)
- Libellé (cadré à gauche)
- Prix HT (cadré à droite)
- Taux TVA (cadré à droite)
- Montant TVA (cadré à droite)
- Prix TTC (cadré à droite)
- Stock (centré)
- Quantité vendue (centré)

La table doit avoir des en-têtes et être un minimum stylée (CSS : padding, borders etc)

La liste doit être paginée par **8** articles / page avec 2 liens de navigation (préc/suiv) et doit savoir détecter le début et la fin de pagination (pas de possibilité d'aller plus loin en amont ou en aval)

Sous la table doivent s'afficher :

- La valeur totale HT du stock
- La quantité totale d'articles en stock (tous articles confondus)
- La valeur totale TTC des produits déjà vendus
- La quantité totale d'articles vendus (tous articles confondus)

## Exercice 3 - Mouvements de stock

Modifiez votre page **liste.php** pour ajouter une colonne “actions” contenant :

- Un lien libellé “+10” qui mène vers un script **stock.php** ayant en paramètre d’URL un champ **code=<code article>** (à renseigner dynamiquement pour chaque article) et un champ **qte=10**. Ce lien va ajouter **10** articles dans le stock. Voir l’explication du script **stock.php** plus loin.
- Un lien libellé “vente” qui mène vers le même script **stock.php** ayant en paramètre d’URL un champ **code=<code article>** (à renseigner dynamiquement pour chaque article) et un champ **qte=-1**. Ce lien va vendre 1 article. Voir l’explication du script **stock.php** ci-après.

Créez un script **stock.php** qui exploite les champs paramètres d’URL suivants :

- **code** : le code de l’article
- **qte** : si **qte > 0** on ajoute la quantité au stock de l’article (c’est ce qu’on appelle un réassort d’article) et si **qte < 0** on retire du stock (c’est une vente) et on ajoute cette quantité au champ **vendu**. Attention, il faudra vérifier que le stock est suffisant avant de retirer !

Le script **stock.php** est aussi une page Web qui doit permettre, par un lien, de retourner à la page **liste.php** une fois l’action terminée.