

<b>BTS SIO</b>
<b>TC5 – Développement Web</b>
<b>TP - Traiter les données d'un formulaire avec PHP</b>

**Ce travail est à rendre à l'emplacement désigné par l'enseignant. Il sera noté.**

## Contexte

En période de Pâques, il est de tradition d'offrir du chocolat aux enfants. M. Lepoulain, directeur de la chocolaterie artisanale Choco & Co ne le sait que trop bien. Les œufs, les poules et les lapins sont les objets chocolatés qui ont le plus de succès, mais il vend aussi toutes sortes d'objets ou d'animaux en chocolat. Cette période faste ne dure qu'une quinzaine de jours, mais elle constitue une partie importante de son chiffre d'affaires annuel.

M. Lepoulain s'est souvent interrogé sur la façon dont il pourrait optimiser ses ventes de chocolats de Pâques. Il réunit donc son équipe de direction pour définir une stratégie. Après cette réunion fructueuse, et suivant une suggestion de sa directrice marketing, M. Lepoulain décide de lancer une campagne de promotion spéciale sur le web intitulée « Joyeuses Pâques avec Choco & Co ! ».

Dans le cadre de cette campagne, une sélection de produits Choco & Co seront proposée à la vente à prix réduits sur un site Internet construit pour l'occasion ([chocopaques.fr](http://chocopaques.fr)). Une campagne de publicité dans la presse écrite et à la radio sera mise en place pour faire venir les visiteurs sur le site web.

M. Lepoulain fait appel à l'entreprise CoolWebDesign pour réaliser le site web.



## Travail à faire

Vous travaillez chez CoolWebDesign et vous faites partie de l'équipe chargée de développer le site web. Certaines tâches de développement vous sont donc confiées.

Un formulaire web permettant aux visiteurs du site de commander les chocolats et de payer en ligne a été créé par l'un de vos collègues (vous le trouverez dans le dossier « chococo » accompagnant ce TP).

**Votre mission est de récupérer les données de ce formulaire de commande et de créer une page web affichant le récapitulatif de la commande d'un client.** Cette page permettra au client de vérifier sa commande avant de passer au règlement. Le travail sera réalisé en PHP.

Suivez les instructions pas à pas données dans ce TP pour réaliser cette page.

Le résultat final de votre travail devra ressembler à ceci (sans images ni feuille de style css):

### Votre adresse de livraison

Isidore Miraplu  
34, Impasse de la Mule qui Braie  
67890 Saint-Gontrant-sur-Pelisse

Paquet cadeau gratuit !

### Vos coordonnées (pour contact)

- Téléphone : 0605040302
- Email : imiraplu@gogole.com

### Votre commande

#### Oeufs en chocolat

Article	Quantité	Prix TTC
Paquet moyen (300 g)	1	27
Total :	27	

#### Animaux en chocolat

Article	Quantité	Prix TTC
Lapin	2	21.98
Total :		

#### Assortiment personnalisé

Article	Quantité	Prix TTC
Assortiment de fritures	2	48

#### Contenu de l'assortiment

- tulipe
- crevette

Remarque : si vous ne désirez pas personnaliser votre assortiment, nous vous enverrons l'ensemble des figures.

---

Total de la commande : 118.96 €

## 1. Adaptation du formulaire

Ouvrez le formulaire HTML **commande\_chocolats.html** dans Notepad++ et prenez connaissance de son contenu.

Ouvrez-le également dans un navigateur web (Firefox ou Chrome) pour voir à quoi il ressemble du point de vue du client.

Pour que les données du formulaire HTML puissent être traitées par votre futur programme php, il faut commencer par y ajouter les deux informations suivantes :

- Dans quel fichier se trouve le programme php qui va traiter les données du formulaire.
- Quelle méthode devra être utilisée pour envoyer les données au serveur web.

Il existe en effet deux méthodes HTTP possibles pour récupérer les données d'un formulaire HTML (**GET** et **POST**). Nous allons utiliser la méthode **POST**.

Vous remarquez que le formulaire HTML est délimité par les balises **<form></form>** que vous connaissiez déjà. Nous allons modifier ces balises pour y ajouter les deux informations dont nous venons de parler.

Modifiez la balise **<form>** du formulaire HTML comme indiqué ci-dessous.

```
<form method="post" action="recap.php">
```

Il va de soi que le fichier php que vous allez créer doit obligatoirement s'appeler **recap.php**. Sinon vous aurez une belle erreur.

## 2. Définition des variables

Nous allons maintenant nous concentrer sur le fichier qui contiendra le programme php.

Dans votre éditeur de texte Notepad++, créez un nouveau fichier. Appelez-le **recap.php**.

Tapez les lignes de code qui débutent et terminent toujours un programme php.

```
<?php  
?>
```

Sauvegardez le fichier dans le même répertoire que le formulaire HTML.

Vous rappelez-vous comment on peut récupérer les données d'un formulaire ?

Non ? Eh bien je vais vous le rappeler. Quand l'utilisateur clique sur le bouton **submit** du formulaire, le contenu des champs est placé automatiquement dans des espaces mémoires (ce sont donc des variables !) qui portent un nom particulier. Ce sont les fameuses **variables superglobales** comme **\$\_POST['xxx']**.

Exemple :

Si dans un formulaire HTML nous avons le champ suivant :

```
<input type="text" name="nom"/>
```

Et si l'utilisateur clique sur le bouton **submit** du formulaire, alors le contenu du champ « nom » sera affecté à une variable qui s'appelle `$_POST['nom']`.

Vous pouvez récupérer le contenu de cette variable dans votre propre variable en faisant une simple affectation :

```
$nom = $_POST['nom']; //récupérer le contenu de la superglobale $_POST[]
```

### 3. Traitement de l'adresse de livraison

Nous voulons que dans la page récapitulative de la commande, l'adresse de livraison apparaisse sous cette forme :

Votre adresse de livraison :

Jacques Ceptepas  
34, rue des Platanes  
Résidence des Ormes, Esc 2, Porte 4  
34560 Saint Rémy Du Champ d'En Bas

Pour cela vous devez :

- Récupérer les données du formulaire concernant l'adresse de livraison.
- Afficher les informations en PHP. Vous devez savoir le faire (sinon, revoir cours), soit avec l'instruction `echo`, soit en utilisant une expression PHP.

Tous les clients n'ont pas forcément besoin de renseigner le complément d'adresse (la ligne « Résidence des Ormes, Esc 2, Porte 4 » dans l'exemple ci-dessus.). Cette ligne correspond au code suivant dans le formulaire html.

```
<p>  
  <label for "complement">Complément d'adresse :</label>  
  <input type="text" name="complement" id="adresse" size="80" maxlength="100" value=""/>  
</p>
```

Quand il n'y a pas de complément d'adresse, on ne veut pas qu'il y ait une ligne vide qui s'affiche comme ci-dessous car c'est inesthétique.

Votre adresse de livraison :

Marie Toivite  
12, chemin des Alouettes  
  
21670 La Pierre sur Crouzon

Il faut donc mettre en place la condition suivante :

```
Si le client a saisi un complément d'adresse,  
    afficher le complément d'adresse  
Sinon  
    ne rien faire
```

Comment savoir si le client a saisi quelque chose ou pas dans le champ **complément** ? Il suffit de faire un test sur le contenu de la variable, comme dans l'exemple ci-dessous.

Exemple :

```
if(!empty($nom)){ //c'est-à-dire si la variable $nom n'est pas vide  
    echo "Merci d'avoir donné votre nom" ;  
}  
else{  
    echo "Vous avez oublié de donner votre nom" ;  
}
```

J'insiste pour dire que c'est un **EXEMPLE**. Ne reproduisez pas bêtement ce code tel quel.

Voilà. Vous savez tout ce qu'il y a à savoir pour développer cette partie du code ! Sinon n'hésitez pas à relire le cours.

Maintenant, Ecrivez le code php attendu et testez-le.

## 4. Traitement du paquet cadeau

Vous avez remarqué la case à cocher dans le formulaire HTML ?

Emballage cadeau gratuit : ☐

Nous voulons développer le comportement suivant :

```
si la case est cochée  
    afficher le rappel « Paquet cadeau gratuit » juste après l'adresse de livraison.  
sinon  
    ne rien faire
```

Comment savoir si la case est cochée ? Grâce à la fonction `isset()`.

Exemple :

```
if(isset($_POST['accepter'])){  
    echo "Vous avez accepté le contrat" ;  
}  
else{  
    Echo "Vous avez oublié d'accepter le contrat" ;  
}
```

Maintenant, écrivez le code php (et testez-le bien sûr).

## 5. Affichage des coordonnées

Vous devez afficher les coordonnées sous la forme suivante :

Vos coordonnées :

- Téléphone : 06 07 78 56 98
- Email : monemail@truc.machin

Il faut donc créer une liste à puce HTML ! Cela ne représente aucune difficulté pour vous. Il faut juste se rappeler comment on peut écrire du HTML en PHP. Voir les exemples ci-dessous.

Façon de faire 1 :

```
<?php
    echo "<H1>Ceci est un titre de niveau 1</H1>";
    echo "<p>Ceci est un paragraphe.</p>";
?>
```

Façon de faire 2 :

```
< ?php
//ici on écrit du PHP
?>

<!-- Ici on intercale du HTML entre le code PHP -->
<H1>Ceci est un titre de niveau 1</H1>
<p>Ceci est un paragraphe.</p>

<?php //ici on se remet à écrire en PHP...
```

Ecrivez le code php nécessaire.

## 6. Affichage des articles commandés

Ici nous nous heurtons à plusieurs problèmes.

Tout d'abord, tous les clients ne commandent pas forcément tous les articles. Par exemple ils peuvent très bien commander des œufs en chocolat et un assortiment, mais pas d'animaux en chocolat.

Concernant l'assortiment personnalisé, Il faut également rappeler quelles figures le client a sélectionnées (poissons, crevettes, tulipe, ...).

Enfin il faut calculer le total par article, le total par rubrique et le total global pour que le client vérifie qu'il ne s'est pas trompé dans sa commande et qu'il sache ce qu'il va devoir payer.

Nous allons prendre ces problèmes un par un.

1. Dans un premier temps nous allons réaliser un programme qui va afficher les quantités commandées pour chaque article (que l'article soit commandé ou pas. S'il n'est pas commandé, la quantité sera 0).
2. Ensuite nous allons développer le calcul nécessaire pour obtenir les totaux et nous allons afficher ces totaux.

3. Enfin nous allons masquer tous les articles non commandés, c'est-à-dire dont la quantité commandée est 0 (zéro).

#### a) Afficher les quantités commandées

Cette partie du programme ne présente pas de difficulté sauf pour le cas de l'assortiment sur lequel nous allons nous attarder. En effet, nous voulons certes afficher combien de paquets d'assortiment sont commandés (facile !), mais aussi la sélection de figures (poissons, crevettes, requins, ...) choisies par le client.

Pour afficher la sélection de figures, nous nous trouvons devant un problème. En effet, voyons ci-dessous comment se présente le code HTML pour effectuer ce choix (j'ai volontairement supprimé les balises de `<TABLE>` pour rendre le code plus lisible.

```
Saumon <input type="checkbox" name="figures[]" value="saumon">
Coquillage <input type="checkbox" name="figures[]" value="coquillage">
Masque<input type="checkbox" name="figures[]" value="masque">
Tulipe<input type="checkbox" name="figures[]" value="tulipe">
Crevette<input type="checkbox" name="figures[]" value="crevette">
Epée<input type="checkbox" name="figures[]" value="pee">
Requin<input type="checkbox" name="figures[]" value="requin">
```

Et nous voyons qu'il ne s'agit pas d'une valeur unique (comme par exemple pour le champ « nom »), mais d'une liste de valeurs pour un même élément (`name="figures[]"`). Nous ne pouvons donc pas stocker cette liste de valeurs dans une variable élémentaire qui ne peut prendre qu'une seule valeur à la fois, comme vous le savez (j'espère). Nous avons besoin d'un **tableau indexé** qui permet de stocker plusieurs valeurs à la fois (rappelez-vous du cours !).

Eh bien ça tombe très bien car en réalité les variables `$_POST[]` sont des variables structurées et vous ne le saviez pas ! Ces variables peuvent donc très bien stocker des tableaux. D'ailleurs vous voyez que cette fois-ci le paramètre `name="figures[]"` contient des crochets []. Cela indique que nous nous attendons à récupérer ici un tableau et non pas une variable élémentaire.

Exemple :

Si on a le code HTML suivant :

```
Rouge<input type="checkbox" name="couleurs[]" value="rouge"><br />
Vert<input type="checkbox" name="couleurs[]" value="vert"><br />
Jaune<input type="checkbox" name="couleurs[]" value="jaune"><br />
Bleu<input type="checkbox" name="couleurs[]" value="bleu"><br />
```

Qui correspond à ceci :

Rouge ☐  
Vert ☐  
Jaune ☐  
Bleu ☐

Et si on a coché les cases Vert, Jaune et Bleu comme ceci :

Rouge ☐  
Vert ☒  
Jaune ☒  
Bleu ☒

Puis on a cliqué sur le bouton **submit** du formulaire HTML, alors les valeurs « Vert », « Jaune » et « Bleu » ont été allouées dans une variable `$_POST['couleurs']` contenant les trois valeurs « Vert », « Jaune » et « Bleu » (mais pas « rouge » car elle n'a pas été cochée).

Là vous me dites « super ! mais comment je fais pour récupérer ces trois valeurs dans cette variable `$_POST['couleurs']` et pour les afficher ? Un simple `echo "$_POST['couleurs']"` ne permet d'afficher que la dernière couleur de la liste, c'est-à-dire « Bleu » (essayez si vous ne me croyez pas !).

Eh bien il existe plusieurs façons, mais je vais vous en indiquer une qui est très efficace et rapide à écrire. Il s'agit d'utiliser une structure itérative que nous avons vue en cours et qui s'appelle `foreach()`. Voici le code PHP correspondant.

```
$couleurs = $_POST['couleurs']; //allocation de la variable $couleurs
foreach($couleurs as $value){
    echo $value."<br/>" ;
}
```

Ce programme affiche les lignes suivantes :

```
vert
jaune
bleu
```

Utilisez cette même structure pour récupérer et afficher le choix de figures dans votre programme php.



## b) Calculer les totaux

Pour l'instant nous avons affiché les quantités commandées. Avant de calculer le total global de la commande, il nous faut enrichir notre page récapitulative avec les prix par article puis calculer les totaux intermédiaires (prix \* quantité pour chaque article).

Où allons-nous trouver le prix par article ? Nous n'avons pas de base de données dans laquelle nous pourrions aller chercher les prix des articles, mais heureusement il y a un autre moyen de conserver des données. Il suffit de créer un champ caché dans le formulaire HTML (Le visiteur du site le verra s'il regarde dans le code source de la page). Ce champ contient une valeur que l'on peut récupérer dans une variable `$_POST[]`.

Ouvrez le formulaire HTML dans Notepad++ et examinez-le. Vous constatez que votre collègue qui a créé le formulaire a été perspicace. Il a mis les prix par articles dans le formulaire. Vous devriez voir plusieurs balises de type « hidden » (« caché » en français) comme celle-ci.

```
<input type="hidden" name="px_assort" value="24 €">
```

Récupérez les valeurs de ces champs en les affectant à des variables dans votre fichier php. Vous pourrez alors effectuer les calculs de prix.

Vous devez calculer :

- Le total par article (quantité \* prix unitaire)
- Le total par rubrique (œufs, animaux, assortiment)
- Le total global

**Etudiant :** *Mais Monsieur, on n'a pas appris à faire des calculs en PHP !*

**Prof :** *Il faut vraiment tout vous mâcher !*

*3 secondes de recherche sur Gogole plus tard :*

$\$a + \$b$	Addition	Somme de $\$a$ et $\$b$ .
$\$a - \$b$	Soustraction	Différence de $\$a$ et $\$b$ .
$\$a * \$b$	Multiplication	Produit de $\$a$ et $\$b$ .
$\$a / \$b$	Division	Quotient de $\$a$ et $\$b$ .

Ecrivez le code PHP de calcul des prix et testez-le.

## c) Affichage des prix sur le récapitulatif

Vous allez maintenant afficher les prix de chaque article à côté de la quantité commandée. Vous avez défini toutes les variables nécessaires à l'étape précédente et vous savez déjà les afficher. Cette étape de devrait donc pas poser de problème pour vous.

Pour que les informations soient organisées et faciles à lire pour le client, il est fortement conseillé de les intégrer dans un tableau HTML (<table>). Celles-ci n'ont normalement pas de secret pour vous.

N'oubliez pas d'ajouter les sous-totaux et le total général.

Modifiez le code PHP d'affichage des informations en y ajoutant le calcul des prix.

#### d) Masquer les produits non commandés

Les produits non commandés ne doivent pas apparaître dans le récapitulatif. Il faut donc les masquer. Comment faire ? Une simple structure `if () {}` suffit.

##### Exemple

```
if ($quantite != 0){ //Donc, uniquement si la quantité commandée est différente de 0
    echo "l'article coûte $prix." ;
}
```

La phrase « l'article coûte \$prix ». Ne s'affiche que si la quantité est différente de 0 (c'est-à-dire si l'article a été commandé).

Il faut également éviter d'afficher les rubriques vides. Je vous laisse trouver une méthode pour le faire.

Modifiez votre code php pour masquer les rubriques vides et les produits non commandés.

## 7. Embellir votre page (Points bonus !)

La page de récapitulatif n'est vraiment pas esthétique. Vous pouvez l'améliorer de plusieurs de façons :

- Tracer des lignes autour des tableaux
- Ajouter des images
- Modifier la police de caractère, les couleurs...
- Etc

Il va de soi que vous pouvez faire tout cela en ajoutant des images et en créant une feuille de style css. **Le bonus ne sera attribué que si vous avez terminé toutes les étapes précédentes.**

Si vous êtes arrivé jusqu'ici, félicitations ! Vous êtes en bonne voie dans l'apprentissage de PHP.

