

EXERCICES EN CLASSE

Développement d'applications de serveur Web 420-DW3-AS I

CLASSE 3

Indications

1. Ajoutez des commentaires et indentations dans vos programmes.
2. Soumettez le code de votre programme (.php fichiers).
3. Soumettez ce formulaire avec les captures d'écran correspondantes (plus d'une, le cas échéant) sous chaque exercice pour prouver que vous avez créé le programme approprié et qu'il fonctionne.

SOLUTION

Exercice 1 –

Gestion de la collecte de données via un formulaire

Ci-dessous se trouve le code d'un formulaire HTML qui permet à un utilisateur d'écrire son prénom et nom afin de recevoir et utiliser ces informations dans un script PHP.

```
<!--Form-->
<form id="form1" method="post" action="class3a_response.php" >
  <!--Form fields to input data-->
  <label for="inputfname">First Name|Prénom</label>
  <input id="inputfname" type="text" name="fname" placeholder="John" required>
  <label for="inputlname">Last Name|Nom</label>
  <input id="inputlname" type="text" name="lname" placeholder="Doe" required>
  <!--Submit button to send form data-->
  <input id="submitbutton1" type="submit" name="send" value="SEND|ENVOYER" />
</form>
```

Répondez aux questions ci-dessous en vous référant au code HTML ci-dessus.

- 1- Pour chacun des éléments nécessaires (ex. form, input), écrivez ci-dessous tous les attributs qui sont essentiels pour la réception des informations soumises dans le formulaire dans un script PHP.

Élément form :

```
method="post"  
action="class3a_response.php"
```

Élément Input

```
name="fname"  
name="lname"  
name="send"
```

- 2- Écrivez ci-dessous le rôle de chacun des attributs identifiés dans la réponse à la question 1 pour la réception des informations soumises dans le formulaire dans un script PHP.

Élément form :

`method="post"` : la méthode http utilisée pour envoyer les informations soumises via le formulaire; correspond aussi à la variable superglobale et tableau associatif utilisée pour les stocker (\$_POST ou \$_GET)

`action="class3a_response.php"` : le fichier que le serveur doit ouvrir lorsqu'on clique sur le bouton de soumission, qu'on soumet donc le formulaire

Élément Input

`name="fname"` : l'index utilisé pour stocker et récupérer l'information soumise dans le champ de saisie identifié par id="inputfname" dans la variable superglobale et tableau associatif (\$_POST ou \$_GET). C'est-à-dire `$_POST["fname"]`.

`name="lname"` : l'index utilisé pour stocker et récupérer l'information soumise dans le champ de saisie identifié par `id="inputlname"` dans la variable superglobale et tableau associatif (`$_POST` ou `$_GET`). C'est-à-dire `$_POST["lname"]`.

`name="send"` : l'index utilisé pour stocker et récupérer l'information soumise dans le champ de saisie identifié par `id="submitButton1"` dans la variable superglobale et tableau associatif (`$_POST` ou `$_GET`). On peut donc l'utiliser pour tester si le formulaire est déjà soumis ou pas. C'est-à-dire `if (isset($_POST["send"]))` ou `if (!(isset($_POST["send"])))`.

- 3- Peut-on ajouter la même valeur pour l'attribut `name` de 2 éléments `input` différents? Justifiez votre réponse.

Non. La valeur doit être unique parce qu'on l'utilise comme index utilisé pour stocker et récupérer l'information soumise dans le champ de saisie. Donc si on a plusieurs champs, on doit avoir des index différents pour chacun d'eux.

Oui. Si on doit mettre la même valeur, il doit être une liste (ou tableau) afin que l'information soumise dans chaque champ de saisie puisse être stockée et récupérée avec un index différent. Par exemple :

```
<input id="inputfname" type="text" name="one_name[]"  
<input id="inputlname" type="text" name="one_name[]"
```

```
$first_last_names = $_POST["one_name"];
```

`$first_last_names` est une liste qui contient le prénom puis le nom, stocké et donc accessible chacun avec un index différent.

- 4- Écrivez ci-dessous le nom du script PHP qui sera ouvert lorsque l'utilisateur soumet le formulaire.

```
class3a_response.php
```

- 5- Écrivez ci-dessous le code à utiliser dans le script PHP identifié dans la réponse à la question 4 pour tester si l'utilisateur a soumis le formulaire via l'élément input de type submit.

```
if (isset($_POST["send"]))
```

- 6- Écrivez ci-dessous le code à utiliser dans le script PHP identifié dans la réponse à la question 4 pour stocker le prénom puis le nom soumis dans le formulaire dans 2 variables PHP.

```
$prenom=$_POST["fname"]
```

```
$nom=$_POST["lname"]
```

- 7- Écrivez ci-dessous l'information qui sera soumise via l'élément input de type submit lorsque l'utilisateur soumet le formulaire; et qui pourra donc être récupérer dans le script PHP identifié dans la réponse à la question 4 à partir de la variable superglobale \$_POST.

```
SEND|ENVOYER
```

Voilà le contenu du script `class3a_response.php` ci-dessous créer à partir des réponses précédentes :

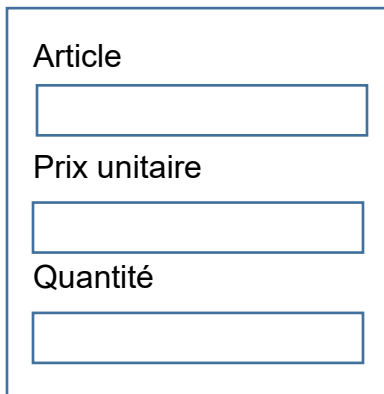
```
<?php
if (isset($_POST["send"])){
    //Assigner des valeurs
    $prenom=$_POST["fname"];
    $nom=$_POST["lname"];
    $boutonSoumission=$_POST["send"];
    //Afficher
    echo "Prenom : $prenom <br/>";
    echo "Nom : $nom <br/>";
    echo "Valeur Attribut value Input de type Submit : $boutonSoumission <br/>";
}
?>
```

Exercice 2 –

Gestion de la collecte de données via un formulaire

Un vendeur vous a demandé d'écrire un module d'une page web qui comprend un formulaire (voyez l'exemple visuel ci-dessous) permettant aux acheteurs de taper:

1. Le nom de l'article acheté;
2. Le prix unitaire de cet article;
3. Et la quantité de l'article acheté.



The image shows a web form with a blue border. Inside the border, there are three input fields stacked vertically. The first field is labeled 'Article', the second is labeled 'Prix unitaire', and the third is labeled 'Quantité'. Each label is to the left of its corresponding input field.

Une fois le formulaire soumis, le programme affiche des résultats (voyez l'exemple visuel ci-dessous) dont :

1. Le prix d'achat sans taxes qui est égal au produit du prix unitaire par la quantité de l'article acheté (sous-total);
2. Le montant de la TVQ qui est égal à 9.975% du sous-total;
3. Le montant de la TPS qui est égal à 5% du sous-total;
4. Le prix d'achat avec taxes (total) qui est égal à la somme du prix d'achat, du montant de TVQ et du montant de la TPS.

Article : ???

Prix unitaire : ??? CA\$

Quantité : ???

Sous-total : ??? CA\$

TVQ (9.975%) : ??? CA\$

TPS (5%) : ??? CA\$

Total : ??? CA\$