

## Conception de Sites Web Dynamiques : TD 5

Formulaires HTML5 - Tableaux associatifs - Récupération de paramètres POST/GET

<http://www.labri.fr/perso/preuter/cswd2022>

### A respecter tout au long de ce TD :

- Si vous ne travaillez pas sur les postes du CREMI, vérifiez que votre logiciel de serveur Apache (MAMP ou Wampserver64 ou Xampp) est bien démarré..
- Les extensions des fichiers doivent être toujours `.php`
- Dans le navigateur, l'URL commence toujours par `http://` et non pas par `file://`.

Vous disposez d'une nouvelle Cheatsheet pour les formulaires, disponible sur le site du cours !

### Exercice 5.1 *Votre premier formulaire HTML5 : addition de deux nombres*

1. Ouvrir votre éditeur de texte. Créer un fichier `formulaire.php` dans votre dossier `td5`. N'oubliez pas de travailler dans votre dossier `c:\MAMP\htdocs` ou `/Applications/MAMP/htdocs` ou `c:\wamp64\www` ou similaire.

Votre arborescence sera donc :

```
cswd/  
├── td1/  
│   └── ...  
├── td2/  
│   └── ...  
├── td3/  
│   └── ...  
├── td4/  
│   └── ...  
├── td5/  
│   └── formulaire.php  
└── ressources/  
    └── ...
```

2. Dans ce fichier, mettez les balises pour un document HTML5 simpliste (donc avec les balises DOCTYPE, `html`, `head`, `body`, ...) et enregistrez-le. Dans le corps du document (donc après l'ouverture du `<body>`), créez un formulaire qui renvoie vers la page `addition.php` avec la méthode `get` comme suit :

```
<form action="addition.php" method="get">  
...  
</form>
```

3. A l'intérieur de la balise `<form>`, créez deux champs `<input />` de type `text` avec les noms de valeur `nombre1` et `nombre2` dans les propriétés `name`, qui permettent de saisir deux nombres, ainsi qu'un bouton `<input type="submit" />` qui permet de soumettre votre formulaire.

Nombre 1 :

Nombre 2 :

4. Ouvrez votre page dans votre navigateur. Si vous travaillez sur les ordinateurs du CREMI, ce sera

```
http://prenom-nom.emi.u-bordeaux.fr/cswd/td5/formulaire.php
```

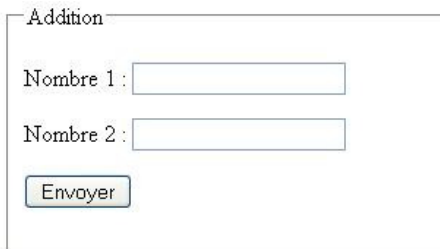
Si vous travaillez sur vos propres ordinateurs, vous passez par votre serveur Apache (MAMP ou Wampserver ou Xampp), p.ex. :

```
http://localhost/cswd/td5/formulaire.php
```

5. Pour expérimenter avec les champs HTML5, changez le type des deux balises `<input/ >` en y mettant l'attribut `type="number"` au lieu de `type="text"`. Comparez.

6. Que se passe-t-il lorsque vous cliquez sur le bouton "Soumettre" ? Regardez bien l'URL, et essayez de la comprendre intégralement. Bien sûr, comme la page `addition.php` n'existe pas encore, il y a un message d'erreur. Souvenez-vous comment accéder à ces plusieurs paramètres passés par l'URL (et qui sont séparés par des `&`) ?

7. Il est possible de regrouper des champs dans un formulaire avec la balise `<fieldset>`, et de mettre des étiquettes avec la balise `<label>` permettant d'activer le champ par clic de souris. Essayez d'utiliser ces deux balises afin d'obtenir le rendu suivant (n'oubliez pas de consulter le Cheatsheet sur les formulaires) :



The image shows a web form with a title "Addition". It contains two input fields, one labeled "Nombre 1" and another labeled "Nombre 2". Below these fields is a button labeled "Envoyer". The form is enclosed in a box with a thin border.

Attention : lorsque vous apportez des changements dans votre page du formulaire (ici `formulaire.php`), n'oubliez pas de recharger la page dans le navigateur (`ctrl-R` ou `cmd-R`) pour que les modifications soient prises en compte !

### Exercice 5.2 *Evaluation d'un formulaire*

1. Maintenant, ouvrez votre éditeur de texte et créez un nouveau document HTML5 (donc avec les balises `DOCTYPE`, `html`, `head`, `body`, ...) dans le fichier `addition.php`.

2. En tout début du fichier, ajoutez l'instruction PHP suivante :

```
<?php print_r($_GET); ?>
```

- L'instruction `print_r` vous permet de connaître le contenu de la variable `$_GET`. Quel est le type de données de cette variable ? Notez que cette instruction ne sert que pour vous aider et donc de simuler un mode *debug*.

3. En tout début du fichier, récupérez les deux paramètres `nombre1` et `nombre2` avec la méthode `$_GET`, et stockez les dans les deux variables `$nombre1` et `$nombre2`, respectivement.

4. Ensuite, dans la partie `<body>` affichez les deux nombres ainsi que leur somme, comme par exemple "Le résultat de l'addition de 47+95 est égal à 142". Vous pouvez d'abord déclarer une variable `$resultat`, et l'affecter à la somme des deux nombres.

5. Mettez chaque nombre et le résultat dans des balises `<strong>...</strong>`.

6. Une fois tout fonctionne bien, vous pouvez bien évidemment commenter (ou supprimer) l'instruction `print_r` (qui ne servait qu'à simuler un mode *debug*, c'est à dire pour faciliter de trouver des erreurs).

7. Si vous ne l'avez pas encore fait, protégez votre page au cas où les paramètres `nombre1` et `nombre2` ne seraient pas renseignés, en utilisant des conditions avec la fonction `isset`.

8. Testez votre formulaire en laissant vos champs de textes vide. Essayez de comprendre le comportement. Pour éviter d'accepter des paramètres qui existent mais sans contenu (vide, en anglais `empty`), vous pouvez également essayer la fonction `empty` : `if (!empty($_GET[...]))`, sachant que l'opérateur `!` fait une négation logique. Vous pouvez également tester l'existence des deux paramètres en utilisant l'opérateur booléen `and` (ou bien `&&`).

### Exercice 5.3 CSS

1. Dans votre dossier `td5`, créez un dossier `css`. Dans ce dossier, créez un fichier `formulaire.css`, et intégrez-le dans votre page `formulaire.php` (via la balise `link` dans la partie `head`, comme d'habitude).
2. Changez la taille de la police des deux champs `<input/>` en définissant une classe `nombre`, comme expliqué sur le Cheatsheet.

### Exercice 5.4 Formulaire pour un commentaire (difficulté : facile)

Dans cet exercice, nous allons faire principalement la même chose que dans l'exercice précédent, et vous allez donc voir si vous avez tout compris.

1. Créer une nouvelle page HTML5 dans le fichier `recette.php` avec un titre `<h1>` de votre choix (par exemple culinaire) et créez les 3 paragraphes (avec la balise avec `<p>...</p>`) avec le texte issu de la version populaire de "Lorem ipsum" issue de la page <https://fr.wikipedia.org/wiki/Faux-texte>.
2. A la fin de votre page `recette.php`, créez un titre `<h3>` avec le texte "Laissez votre commentaire". Ajoutez un formulaire qui renvoie vers la page `recap.php` avec la méthode `get`, et qui permet de renseigner un champs d'entrées du texte (`input`, appelé *nom*), d'une zone de texte (`textarea`, appelée *texte*) ainsi qu'un bouton pour soumettre le formulaire.
3. Créer la page `recap.php` qui récupère et qui affiche les données du formulaire dans une liste `ul`, avec deux éléments "Votre nom : " et "Votre commentaire : ", suivi du nom et du texte du commentaire, respectivement. Pour vous aider, vous pouvez temporairement toujours afficher tous le paramètres GET pendant la phase de développement, avec l'instruction :  
`<?php print_r($_GET); ?>`

Une fois votre page fonctionne parfaitement, enlevez cette instruction !

4. A la fin de votre page `recap.php`, créez un lien hypertexte avec le texte "retourner à la recette", et qui permet de renvoyer à la page `recette.php` !
5. Réfléchissez si vous êtes d'accord que ce serait une bonne idée que vos commentaires apparaissent maintenant sur la page de recette. A votre avis, qu'est-ce qu'on a besoin pour faire ceci ?

### Exercice 5.5 Formulaire pour un commentaire avec POST (difficulté : facile)

Avec la méthode GET, les informations saisies dans les formulaires sont visibles dans l'URL, ce qui pose notamment un problème pour des paramètres long tel que des commentaires, ou alors les mots de passe. Pour palier à ce problème, nous allons changer l'exercice précédent et utiliser la méthode POST au lieu de la méthode GET :

1. Changez la méthode de passage de paramètres de votre formulaire `recette.php` en méthode `post` en modifiant la propriété `method` de la balise `<form>` comme suit :

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

Pour récupérer les paramètres, dans votre page `recap.php`, utilisez donc `$_POST` (*Remarque:* Remplacez donc toutes les occurrences de `$_GET` par `$_POST` ). Pour vous aider, vous pouvez temporairement toujours afficher tous le paramètres POST pendant la phase de développement, avec l'instruction :

```
<?php print_r($_POST); ?>
```

2. Testez votre formulaire. N'oubliez pas de recharger la page `recette.php` qui contient votre formulaire dans votre navigateur avec `ctrl-R` ou `cmd-R` afin que les modifications soient prises en compte ! Lors de la soumission du formulaire, qu'observez-vous dans l'URL ?

## Exercice 5.6 Formulaire de connexion (difficulté : moyenne)

### Connexion

Adresse email :

Mot de passe :

Se souvenir de moi :

1. Créer une page PHP `connexion.php` avec les en-têtes HTML5. Ajoutez un titre titre h2 "Connexion" ainsi qu'un formulaire permettant à un membre de se connecter au site. Cette page devra renvoyer à une page `verifier-connexion.php` avec la méthode POST.

Cette page contiendra :

- un champs de type email s'appelant `email` permettant de renseigner une adresse mail,
  - un champs de type mot de passe s'appelant `motdepasse`,
  - une case à cocher permettant de spécifier si l'utilisateur souhaite être reconnu,
  - et un bouton de type submit permettant de se connecter.
2. Testez votre formulaire dans un navigateur.
3. Créez la page `verifier-connexion.php`. Cette page devra afficher le texte "Votre email est ..." suivi par l'adresse email que l'utilisateur a renseigné.
4. Complétez votre page `verifier-connexion.php` pour qu'elle affiche "Bienvenue, vous êtes désormais connecté" si l'email connecté correspond à `mail@monmail.fr` et le mot de passe correspond à `f2aL+RSw`. Vous pouvez de nouveau utiliser l'opérateur booléen `and (&&)`. Référez-vous, comme d'habitude, au Cheatsheet PHP pour vous rappeler de la syntaxe de la structure de contrôle `if`. N'oubliez pas que les comparaisons en PHP se font avec `==` et non pas `=`.

Vous avez certainement remarqué :

1. Bien évidemment, il peut y avoir plusieurs pairs d'email et de mot de passe qui correspondent à différents utilisateurs. Dans un prochain TD, nous allons vérifier les codes d'accès stockés dans une table `utilisateur` d'une base de données.
2. Comme les informations saisies dans les formulaires contiennent des mots de passe, nous avons utilisé la méthode POST afin de ne pas les rendre visibles dans la barre d'adresse de votre navigateur.
3. Cet exercice ne respecte pas une règle de sécurité informatique, car dans votre page PHP, vous avez accès à un mot de passe stocké "en clair", à savoir d'une manière non-crypté. Quand nous allons stocker les mots de passe dans une table d'une base de données, nous allons respecter cette règle de sécurité informatique, et nous allons crypter les mots de passe, car même un créateur ou administrateur de site n'a pas le droit de connaître les mots de passe ...

Vérifier tous vos fichiers HTML5 et CSS avec les validateurs (pour les fichiers avec du code PHP, attention : passer par "afficher source" dans votre navigateur").

<http://validator.w3.org/>

<http://jigsaw.w3.org/css-validator/>

# 1 Exercices optionnelles 1

## Exercice 5.7 Préparation d'un sondage (plutôt difficile)

Dans cet exercice, nous allons préparer un site web permettant de créer une question, et proposez différentes réponses pour un sondage.

1. Créez une page `sondage-creer-question.php` avec les balises HTML simpliste. Cette page contiendra un formulaire qui revoie vers la page `sondage-creer-reponses.php` avec la méthode GET. Ce formulaire contient un champ texte permettant de saisir une question et qui s'appelle `question`, un champ nombre s'appelant `nombre` qui permet de spécifier le nombre de réponses, ainsi qu'un bouton pour soumettre le formulaire.
2. Créez la page `sondage-creer-reponses.php` qui doit récupérer et afficher la question posé, ainsi que le nombre de réponses.
3. Cette page doit également elle-même avoir un formulaire qui renvoie vers la page `sondage-recap.php`. Elle doit afficher plusieurs champs de texte (autant de champs que renseigné dans `nombre`), s'appelant `reponse1`, `reponse2`, `reponse3`, etc., ainsi qu'un bouton soumettre.
4. Dans votre formulaire, ajoutez également deux champs `<input/ >` de type caché (avec la propriété `type="hidden"`) et avec la propriété `name` s'appelant `question` et, respectivement, de `nombre`, en renseignant leurs valeurs respectives (avec la propriété `value=""`). Ces champs serviront à la page du récapitulatif pour avoir les informations respectives.
5. Créer la page `sondage-recap.php` qui doit afficher toutes la question et les réponses possibles.

## Exercices optionnels 2

Dans les exercices suivants, nous souhaitons créer un site web permettant d'effectuer des calculs sur des matrices. Nous commençons par l'opération la plus simple, l'addition de deux matrices **A** et **B** de taille fixe, à savoir ( $m = 2, n = 2$ ). Par la suite, nous étendons cette opération à des matrices de taille généralisée ( $m, n$ ).

### Exercice 5.8 *Addition de Matrices 2x2 (difficulté : simple)*

1. Créer un fichier `form-addition-matrices.php` dans votre dossier `td5`, et créez un titre `h1`, deux titres `h2`, et 8 champs `input` de type `number`, ainsi qu'un formulaire englobant le tout et renvoyant vers le fichier `resultat-addition-matrices.php` avec la méthode `GET`, comme suit :

## Addition de matrices

### Matrix A

<input type="number"/>	<input type="number"/>
<input type="number"/>	<input type="number"/>

### Matrix B

<input type="number"/>	<input type="number"/>
<input type="number"/>	<input type="number"/>

Pour les propriétés `name` des champs `input`, choisissez `A1-1`, `A1-2`, `A2-1`, et `A2-2` pour la matrice **A**, et des noms similaires pour la matrice **B**.

2. Ouvrez votre page dans votre navigateur.  
3. Créez le fichier HTML5 `resultat-addition-matrices.php` permettant d'afficher un récapitulatif des matrices **A** et **B**, ainsi que le résultat de l'addition de matrices.

### Exercice 5.9 *Addition de Matrices nxm (difficulté : moyen)*

1. Modifiez votre fichier `form-addition-matrices.php` pour qu'il prenne deux paramètres  $m$  et  $n$  depuis l'URL, et qu'il permet de renseigner des matrices **A** et **B** avec  $m$  lignes et  $n$  colonnes. Bien évidemment, il faut utiliser deux boucles imbriquées, de type `for` ou `while`.  
2. Ouvrez votre page dans votre navigateur, et expérimentez avec des valeurs différents pour  $n$  et  $m$ .  
3. Dans votre formulaire, ajoutez deux champs `<input />` de type caché (avec la propriété `type="hidden"`) et avec les propriétés `name` de `n` et `m` en renseignant leurs valeurs respectives. Ces champs serviront à la page du résultat de savoir la taille de la matrice.

### Exercice 5.10 *Faire choisir les tailles des matrices (difficulté : simple)*

1. Créez un fichier `form-taille-matrices.php` qui contient un formulaire avec deux champs `input` de type `number` permettant de saisir la taille de matrice, ainsi qu'un champs pour soumettre le formulaire. Le formulaire renverra à la page `form-addition-matrices.php`.