

## TP 9

# PHP et formulaires

Ce TP est dédié à PHP. Sauf mention contraire, aucune modification des fichiers HTML et CSS fournis n'est autorisée.


Si vous utilisez les machines du département :


- démarrer sous Windows
- les fichiers `.php` doivent être placés dans le répertoire suivant :  
C:\wamp64\www\
  - depuis le navigateur, charger `localhost:8080/votrefichier.php`

### EXERCICE

**Exercice 9.1** (Traitement d'un formulaire). Cet exercice se concentre exclusivement sur PHP. L'objectif est de compter les points d'un quiz, dont le code HTML (et CSS) est fourni. Chacune des trois questions au quiz vaut un point en cas de bonne réponse.

*NB : le bouton « Pré-vérifier » n'a aucune utilité dans cet exercice (il sera utilisé à l'exercice suivant dans le cadre de vérifications avancées avec JavaScript).*

 **Question 1 :** Renommer la page `quiz.html` en `quiz.php`. Vérifier que `quiz.php` se charge correctement en demandant l'URL `localhost:8080/votrefichier.php` dans votre navigateur. Nous travaillerons désormais uniquement dans `quiz.php`.


 **Question 2 :** Ajouter un morceau de code PHP dans `quiz.php` afin que, juste en-dessous du titre `<h1>`, la date courante s'affiche. Compléter le morceau de code ci-dessous, et chercher la documentation de la fonction `date()` :

```
1 <h1>Quiz avec PHP</h1>
2
3 <?php
4     echo '<p>Bonjour, nous sommes le ' . date('quelque chose') .
5     '&nbsp;';
6 ?>
```

Ce qui doit afficher le rendu suivant dans le navigateur :

### Quiz avec PHP

Bonjour, nous sommes le 13/04/2023!

 **Question 3 :** Ajouter le code suivant dans la balise `<form>` pour que le contenu du formulaire soit envoyé vers la page elle-même :

```
1 action="<?php echo $_SERVER['XXXX']; ?>"
```

Par quoi faut-il remplacer XXXX? (voir le cours)

**Question 4 :** Créer un fragment de code PHP *avant* le début du code HTML. L'objectif va être de récupérer les valeurs transmises par le formulaire via la méthode `POST`. La structure devra donc être :

```

1 <?php
2 if ($_SERVER["REQUEST_METHOD"] == "POST") {
3
4 // Récupération des données
5 $nom = $_POST['nom'];
6 // TODO
7 }
8 // le code HTML commence après
9 ?>
```

Compléter les déclarations de variables afin de récupérer les valeurs des 4 champs du formulaire.

**Question 5 :** Compléter le code PHP pour que, après envoi du formulaire, une phrase contenant le contenu du champ « nom » s'affiche à l'écran sous le formulaire. Par exemple :

Bonjour Jade, voici vos résultats :

Astuce : déclarer une variable

```

1 $texte_reponses = '';
```

au tout début du fichier PHP (avant le premier `if`), puis l'afficher dans la page HTML sous le formulaire, avec :

```

1 <?php
2 echo $texte_reponses;
3 ?>
```

Ce sera donc cette variable qui permettra tout l'affichage des réponses.

Ne pas oublier qu'il faut générer du code HTML, et non du simple texte.

**⚠ Validation du code HTML**

Attention, le code HTML généré par PHP doit être *valide*!

**Question 6 :** Compléter le code PHP pour tester les réponses de l'internaute, et compter ses points. Actualiser la variable `$texte_reponses` en conséquence. Il faut le faire pour chacune des trois questions du quiz.

Ci-dessous le code correspondant à la première question :

```

1 $points = 0;
2 // Question 1
3 if (!empty($annee) && $annee == 1969) {
4     $points++;
5     $texte_reponses .= 'Vous avez réussi la première question,
6     bravo&nbsp;  !<br>';
7 }else{
8     $texte_reponses .= 'Vous avez échoué à la première question.<br>';
9 }
```


NB : pour les cases à cocher, le point est accordé si et seulement si exactement les trois bonnes cases sont cochées (non, on n'enseigne pas (encore) de saut en parachute à Sorbonne Paris Nord). Il faut donc vérifier que exactement trois cases sont bien cochées, puis vérifier que chacune des trois bonnes réponses apparaît dans le tableau de résultats récupéré.

Exemple d'affichage dans la page HTML pour deux bonnes réponses :

Bonjour Emma, voici vos résultats :  
 Vous avez réussi la première question, bravo!  
 Vous avez réussi la deuxième question, bravo!  
 Vous avez échoué à la troisième question.  
 Score total : **2 points**.

Exemple d’affichage dans la page HTML pour trois bonnes réponses :

Bonjour Léo, voici vos résultats :  
 Vous avez réussi la première question, bravo!  
 Vous avez réussi la deuxième question, bravo!  
 Vous avez réussi la troisième question, bravo!  
 Vous avez tout bon, bravo!  
 Score total : **3 points**.


 **Question 7 :** Modifier le code PHP pour que, en cas de score maximal (toutes les réponses correctes), le formulaire ne se réaffiche pas du tout, c’est-à-dire que seul le contenu de `$texte_reponses` s’affiche.


Une solution possible :

- déclarer une variable booléenne testant si toutes les réponses sont correctes ;
- si cette variable est à `true`, ne pas afficher le formulaire, grâce à l’ajout (exceptionnel) d’un champ `style` (avec la bonne propriété CSS) dans la balise `<form>`.


(Il existe d’autres solutions aussi.)


Une fois de plus, s’assurer que le code HTML ainsi généré est entièrement valide.

 **Question (optionnelle) 8 :** Noter de façon plus fine la question sur les disciplines : une bonne case cochée vaut 1 point, et une mauvaise vaut  $-0,5$  point (sans possibilité d’avoir une note  $< 0$  à cette question). NB : la note maximale au quiz devient donc 5.


 **Question (optionnelle) 9 :** Afficher toutes les réponses de l’internaute en donnant aussi les bonnes réponses en cas d’échec. Exemple d’affichage :

Bonjour Louise, voici vos résultats :  
 Vous avez répondu « 1969 » à la première question, bravo!  
 Vous avez répondu « Droit », « Informatique », « Médecine » et « Saut en parachute à la deuxième question », aïe aïe aïe, c’est raté! La bonne réponse était « Droit », « Informatique » et « Médecine ».  
 Vous avez répondu « Un scientifique » à la troisième question, bravo!  
 Score total : **4,5 points**.

 **Question (optionnelle) 10 :** Remplacer « 2023 » par l’année courante de façon automatique. Regarder la documentation de la fonction `date()` en PHP.

 **Question (optionnelle) 11 :** Rappel : si l’internaute n’a pas tout bon, le formulaire se réaffiche... mais ses précédentes réponses sont perdues!


Réintroduire toutes ses réponses dans le formulaire en cas d’échec, afin qu’il soit plus facile de les resoumettre après correction.


 **Question (optionnelle) 12 :** Modifier le code HTML et le code PHP afin que la première question (sur l’année) s’affiche sous la forme d’une glissière (élément HTML `<input type="range">`).





## EXERCICE

**Exercice 9.2** (Pré-vérifications avec JavaScript (pour les plus rapides)).


 **Question (optionnelle) 1 :** Créer un fichier JavaScript `quiz.js` et le lier depuis la page HTML. Cet exercice se fera exclusivement dans le fichier `quiz.js` sans modification des fichiers HTML et CSS.


 **Question (optionnelle) 2 :** Faire en sorte que, lors du clic sur le bouton « Pré-vérifier », si strictement moins de deux cases ont été cochées à la question sur les disciplines, alors une alerte « Merci de cocher au minimum 2 cases » s’affiche (avec `window.alert`).

 **Question (optionnelle) 3 :** Faire en sorte que, lors du clic sur le bouton radio « 2023 » à la question sur l’année de création, une alerte « You must be kidding? » s’affiche. Le bouton radio ne doit alors *pas* être coché. Utiliser un attribut `checked` quelque part.


 **Question (optionnelle) 4 :** Faire en sorte que, si moins de deux cases ont été cochées à la question sur les disciplines, alors les deux boutons « Réinitialiser » et « Envoyer » sont désactivés (attribut `disabled`) — auquel cas le formulaire ne peut évidemment pas être validé.

Vérifier que, initialement, les deux boutons sont bien désactivés. Vérifier qu’ils deviennent bien activés lors d’un clic sur « Pré-vérifier » avec au moins deux cases cochées. Vérifier que, lors d’un clic sur « Réinitialiser », les deux boutons sont bien de nouveau désactivés.

 **Question (optionnelle) 5 :** Faire en sorte que, si le champ « nom » est vide, alors les deux boutons « Réinitialiser » et « Envoyer » sont désactivés.

 **Question (optionnelle) 6 :** Effectuer avec JavaScript une pré-validation dynamique des deux questions du formulaire avec des couleurs :

- si la bonne réponse à la première question est sélectionnée, celle-ci s’affiche en vert ; sinon en rouge ;
- pour la deuxième question, chaque case cochée à raison s’affiche en vert ; chaque case cochée à tort s’affiche en rouge ; les autres restent affichées en noir.

 **Question (optionnelle) 7 :** Ajouter en bas de la page HTML (toujours uniquement avec JavaScript) un comptage *dynamique* des réponses : à chaque actualisation d’une réponse par un clic, le score se modifie dynamiquement.

Exemple d’affichage possible :

Prénom : Mia  
 Score courant :  
 Question 1 : 1  
 Question 2 : 1,5  
 Question 3 : 0  
 Score total : **2,5 points.**

