TD 9 : formulaires HTML et traitement PHP

L2 I : Développement Web

DÉPARTEMENT Sciences Informatiques

CERGY PARIS UNIVERSITÉ

Environnement de travail

N'oubliez pas de tester vos pages avec les différents navigateurs disponibles sur les postes (Firefox, Opera, Chrome).

Pensez à valider les codes **HTML** et **CSS** à l'aide des validateurs du W3C et pensez à la validation XML de vos résultats.

Pour ce TD, tous les exercices doivent être traités sur une unique page du TD9.

Exercice 1 : formulaire de recherche en utilisant un site externe (ici, un moteur de recherche)

À partir des exemples de liens paramétrés du TD précédent, testez les URL suivantes :

- https://fr.search.yahoo.com/search?q=PHP+8
- https://www.google.fr/search?q=PHP+8
- https://www.bing.com/search?q=PHP+8

Créez un formulaire avec un champ de saisie et un bouton « rechercher » : associez l'action à **un** des moteurs de recherche ci-dessus : compte tenu des URL, vous utiliserez impérativement la méthodes « get ».

N.B. : cet exercice doit être traité en HTML seul (i.e. pas de PHP).

Vous pouvez utiliser un « `<input type="search">` » pour cet exercice.

Exercice 2 : test simple de la méthode get

Proposez un formulaire (code HTML) avec un champ de type texte permettant à l'utilisateur de saisir une chaîne de caractère (en minuscules) et un bouton de validation.

Dans une page PHP de traitement **provisoire** (utilisez efficacement header.inc.php et footer.inc.php pour construire rapidement une nouvelle page fonctionnelle, valide et qui s'intègre à votre site), affichez la valeur saisie en majuscules en utilisant la fonction strtoupper() par exemple.

Exercice 2bis : test des méthodes get et post et type de données

Complétez le formulaire précédent avec un second champs de type texte correspondant à une

valeur numérique en décimal.

Dans la page PHP de traitement, convertissez la **valeur** décimale en sa **représentation** hexadécimale en respectant les types (int → string) et affichez-la.

Testez les méthodes « get » puis « post » pour les échanges entre le navigateur et le serveur web et comparez.

Remarque : pour la majorité des situations vous utiliserez de préférence la méthode « get » qui est la plus répandue et qui permet plus simplement le « *debug* » de vos pages.

Exercice 2 ter : formulaire et traitement sur la même page

Rassemblez le formulaire et son traitement sur la même page (du TD 9) : vous modifierez l'exercice 2 initial afin de ne conserver qu'une version de cet exercice : vous veillerez au cas où le formulaire n'a pas encore été rempli.

Exercice : conversion et CSS

Créez un formulaire avec un champ de saisie permettant de saisir une couleur web sur 6 chiffres hexa et affichant la conversion de cette valeur sous la forme de 3 entiers en respectant la syntaxe CSS : exemple : si la valeur saisie est #FFFF00, le résultat sera : rgb(255, 255, 0)

Exercice 3 : bouton-radio

Créez un formulaire avec un champ de saisie de type string permettant de saisir une valeur décimale ou hexadécimale et un bouton-radio permettant de préciser la base de numération retenue : afficher ensuite comme résultat les nombres à partir de 0 jusqu'à la valeur saisie (vous afficherez l'ensemble des valeurs dans un tableau dans les 4 bases de numération déjà étudiées) : donc si l'utilisateur saisie 10 en base 10, il y aura 11 lignes de résultat et si il saisie 10 en base 16, il y aura 17 lignes de résultat.

Exercice 4 : formulaire de saisie d'URL

À partir d'une URL quelconque (ex. http://www.cyu.fr, https://www.cyu.fr/) saisi par l'utilisateur dans un champ (unique) de type url, réutilisez la fonction du TD précédent pour extraire :

1) le protocole (ex. http, https)
2) le TLD (Top Level Domain) (ex. : France) à partir d'un tableau associatif php : vous pourrez vous limiter aux TLD suivantes : com, org, net, fr
3) le sous-domaine (ex. : cyu).
4) le nom du serveur (host[1]) sur ce domaine (ex. : www).
5) Vous utiliserez également la fonction PHP « dns_get_record » avec la valeur « DNS_A » pour le paramètre « $type » afin de récupérer l'adresse IP correspondant au nom de

1 Le nom complet du serveur (host) reste néanmoins dans cet exemple : « www.cyu.fr ».

domaine.

Vous rassemblerez le formulaire et son traitement sur la même page : l'affichage du formulaire et le traitement seront donc réalisés sur une page web unique : cf. recommandations à la fin de ce sujet pour éviter les messages d'avertissement.

Remarque : il existe des URL différentes (ex. http://www.education.gouv.fr/ ou http://ratp.fr/) qui ne sont pas à prendre en compte dans cet exercice.

Exercice 5 : contrôle des données en provenance d'un formulaire

Affichez un formulaire avec une saisie d'une valeur numérique comprise entre 2 et 16 (cf. input type = « number »), puis affichez la table de multiplication correspondante : comme les données en provenance d'un formulaire peuvent être facilement modifiées, prévoyez **côté serveur** la vérification de la validité de cette valeur (i.e. une valeur entière comprise entre 2 et 16).

Exercice 6 : formulaire avec cases à cocher.

À l'aide d'un formulaire composé de 9 cases à cocher (*user*, *group*, *others* et r, w, x), reconstituez la valeur octale de la commande « chmod ». Vous afficherez le formulaire de saisie sous la forme d'un tableau 3 x 3.

Exercice 7 : formulaire avec bouton radio

On considère à présent, sur la même page, un champ de saisie de la valeur octale de « chmod » et 2 boutons-radio pour fichier ou répertoire. Afficher sur la même page le formulaire et le résultat (au format « d/- » et « rwx »). La valeur octale et le type d'élément seront **ré-affichés** dans le formulaire pour conserver la correspondance d'affichage.

```
marc@marc-HP-ZBook-15u-G2:~/public_html/continuite_pedagogique.d$ ls -al
total 76
drwxrwxr-x  3 marc marc  4096 mars  20 22:30 .
drwxr-xr-x 25 marc marc  4096 mars  17 10:46 ..
-rw-rw-r--  1 marc marc  2886 mars  20 22:30 index.php
-rw-rw-r--  1 marc marc  3032 mars  20 22:28 index.php~
-rw-rw-r--  1 marc marc 16188 mars  20 22:08 liste_mails_l2i.csv
drwxrwxr-x  2 marc marc 20480 mars  20 22:30 trombinoscope.d
```

Illustration 1: exemple d'affichage en mode console : « d » = directory ; « - » = regular file

Amélioration : indiquer les équivalences textuelles pour r (lecture), w (écriture) et x (traversable pour un dossier vs exécutable pour un fichier).

Exercice 8 : formulaire généré par des boucles PHP avec liste d'options

Rappel : on considère sur une même page (unique pour le TD #9) le formulaire et son traitement :

le formulaire proposera une liste déroulante d'options pour choisir une année entre 1950 et 2050 : cette liste est générée par une boucle ! La page de traitement indiquera si il s'agit d'une année bissextile et précisera le jour de la semaine du 1er janvier (cf. fonction date en PHP) : vous prévoirez une fonction de traitement.

Quelques conseils complémentaires

Ajoutez les balises « <fieldset> » et « <legend> » (dans « form ») autour des éléments de chaque formulaires afin de pouvoir regrouper l'ensemble des formulaires sur la même page.

Pour le traitement, il faut s'assurer de la validité des variables de formulaire en utilisant les fonctions « isset() » et « empty() » avant le traitement.

Ajoutez des champs de type « <label> » avec l'attribut « for » pour les associer aux différents éléments du formulaire (via leur « id »).

Lorsque la page s'actualise (après la validation d'un des formulaires), replacez les valeurs de l'utilisateur (i.e. de l'internaute) dans les champs du formulaire : cf. le comportement dans un moteur de recherche : les mots recherchés se ré-affichent dans le champs de saisie de la page résultat.

Vous pouvez aussi ajouter par exemple l'attribut *placeholder*.