

## Unix et Programmation Web

### Cours 8

kn@lri.fr  
<http://www.lri.fr/~kn>

## Champs de texte

On peut placer un champ de texte dans un formulaire au moyen de la balise `<input type="text" />`.

```
<input size="10" name="montexte" type="text" value="abc" />
```

Apperçu:

Attributs:

- **size**: donne la largeur en caractères
- **name**: le nom de l'élément (pour y accéder depuis PHP)
- **value**: la valeur initiale du champ (si on veut le pré-remplir)

Accès à la valeur depuis PHP se fait via `$_GET["montexte"]` ou `$_POST["montexte"]` selon la valeur de l'attribut `method` de l'élément `form` englobant.

## Plan

- 1 Réseaux, TCP/IP ✓
- 2 Web et HTML ✓
- 3 CSS ✓
- 4 PHP : Introduction ✓
- 5 PHP : expressions régulières, fichiers, sessions ✓
- 6 Formulaires
  - 6.1 Formulaires (utilisation avancée)
  - 6.2 Variables super-globales (rappels)
  - 6.3 Conception d'une application simple (étude de cas)

## Champs de texte long

Pour des textes multilignes, on utilise la balise `<textarea />`.

```
<textarea name="montexte" rows="3" cols="10">du texte!</textarea>
```

Apperçu:  Attributs:

- **rows**: donne le nombre de lignes
- **cols**: donne le nombre de colonnes (largeur en nombre de caractères)
- **name**: le nom de l'élément (pour y accéder depuis PHP)

On peut pré-remplir la zone en mettant du texte entre la balise ouvrante et fermante. Accès à la valeur depuis PHP se fait via `$_GET["montexte"]` ou `$_POST["montexte"]` selon la valeur de l'attribut `method` de l'élément `form` englobant.

## Listes déroulantes

On peut définir une liste déroulante avec les balises `<select>` et `<option>`

```
<select name="maliste">
  <option value="bleu" >Bleu</option>
  <option value="vert" >Vert</option>
  <option value="rouge" >Rouge</option>
  <option value="jaune" selected="selected" >Jaune</option>
</select>
```

Apperçu:  Attributs:

- **name (dans select):** le nom de l'élément (pour y accéder depuis PHP)
- **value (dans option):** donne la valeur du choix
- **checked (dans option):** pré-sélectionne cette option

Le contenu de chaque élément option est celui affiché dans la liste. Accès à la valeur depuis PHP se fait via `$_GET["maliste"]` ou `$_POST["maliste"]` selon la valeur de l'attribut `method` de l'élément `form` englobant.

## Boutons à choix unique

On peut définir un choix avec la balise `<input type="radio">`

```
8h<input type="radio" name="h" value="matin"/>
12h<input type="radio" name="h" value="midi"/>
20h<input type="radio" name="h" value="soir"/>
```

Apperçu : 8h  12h  20h

Accès à la valeur depuis PHP se fait via `$_GET["h"]` ou `$_POST["h"]` selon la valeur de l'attribut `method` de l'élément `form` englobant.

**Attention** il faut réutiliser la même valeur de l'attribut `name` pour les boutons du même groupe.

## Cases à choix multiples

On peut définir des cases à cocher à choix multiple avec la balise `<input type="checkbox">`

```
8h<input type="checkbox" name="tab[]" value="matin"/>
12h<input type="checkbox" name="tab[]" value="midi"/>
20h<input type="checkbox" name="tab[]" value="soir"/>
```

Apperçu: 8h  12h  20h

Accès à la valeur depuis PHP se fait via `$_GET["tab"]` ou `$_POST["tab"]` selon la valeur de l'attribut `method` de l'élément `form` englobant.

**Attention** il faut rajouter `[]` dans l'attribut `name` pour que PHP crée un tableau. `tab[i]` contient la valeur de la *i*<sup>ème</sup> cases cochée (à partir de 0).

## Plan

- 1 Réseaux, TCP/IP ✓
- 2 Web et HTML ✓
- 3 CSS ✓
- 4 PHP : Introduction ✓
- 5 PHP : expressions régulières, fichiers, sessions ✓
- 6 Formulaires
  - 6.1 Formulaires (utilisation avancée) ✓
  - 6.2 Variables super-globales (rappels)
  - 6.3 Conception d'une application simple (étude de cas)

## Variables super-globales

PHP définit un certain nombre de variables dites « super-globales ». Ce sont des variables accessibles depuis n'importe où dans un programme PHP, y compris depuis des fonctions, sans adjonction du mot clé `global`.

```
$TOTO = 42;
function f() {
    global $TOTO; // obligatoire, sinon $TOTO n'est pas visible

    echo $TOTO;
    echo $_GET["champ"]; //fonctionne toujours, même sans 'global'
                        //au début de la fonction.
}
}
```

## Variable `$_GET` et `$_POST`

Ces variables contiennent les valeurs des champs d'un formulaire. Si l'attribut `method` de l'élément `form` vaut `get` la variable `$_GET` contient les valeurs. Si il vaut `post` la variable `$_POST` est à utiliser.

## Variable `$_FILES`

Cette variable contient les informations relatives aux fichiers envoyés par le client.

Si le nom du champ est `fichier`, alors:

`$_FILES["fichier"]["error"]`: Code d'erreur (0 si tout c'est bien passé, > 0 si une erreur s'est produite. Les autres champs ne sont définis que si "error" vaut 0).

`$_FILES["fichier"]["tmp_name"]`: Nom du fichier temporaire sur le serveur où a été sauvegardé le contenu du fichier envoyé

`$_FILES["fichier"]["name"]`: Nom original du fichier

`$_FILES["fichier"]["size"]`: Taille du fichier

`$_FILES["fichier"]["type"]`: Le type MIME du fichier

## Variable `$_SESSION`

Contient les **variables de sessions**. La variable n'est accessible qu'après un appel à `session_start()`. Ces le contenu de cette variable persiste durant toute la durée de la session même sur des pages différentes. La variable `$_SESSION` est un tableau.

## Variable \$\_COOKIE

Contient les **cookies** que le client à envoyé au serveur. La variable \$\_COOKIE est un tableau.

## Plan

- 1 Réseaux, TCP/IP ✓
- 2 Web et HTML ✓
- 3 CSS ✓
- 4 PHP : Introduction ✓
- 5 PHP : expressions régulières, fichiers, sessions ✓
- 6 Formulaires
  - 6.1 Formulaires (utilisation avancée) ✓
  - 6.2 Variables super-globales (rappels) ✓
  - 6.3 Conception d'une application simple (étude de cas)

## Cahier des charges

On souhaite créer une application Web simple de gestion de QCM. L'application doit répondre au contraintes suivantes :

- Un QCM est un ensemble de questions, suivies d'une liste de choix. Il peut y avoir plusieurs choix possible. La réponse à la question est correcte si uniquement les bons choix sont sélectionnés.
- Un QCM doit être facilement modifiable/éditable
- Après la réponse à la dernière question, le score de l'utilisateur est affiché, ainsi que les bonnes réponses
- Il y a un temps limite par question

## But du jeu

Nous allons concevoir, le plus précisément possible l'application (prenez des notes) :

- Rafiner le cahier des charges
- Prévoir plusieurs scénarios d'utilisation
- Concevoir l'application (données, interface graphique, code)
- Tester/vérifier la robustesse

Vous devrez écrire le code correspondant en TP