

# HTML5 : Formulaires

- Objectifs
- Construction d'un formulaire
- Élément `<form>`
- Méthodes GET et POST
- Élément `<input>`, ses types et attributs
- Les autres éléments

# Objectif des formulaires

- La plupart des éléments HTML permettent d'**afficher** sur le navigateur **une information reçue** du serveur. Cela suffit avec les sites statiques
- Mais **comment envoyer une demande ?**  
Comment interagir avec le site ?  
Comment avoir une réponse personnalisée ?
- Les **formulaires** permettent au visiteur du site d'**entrer des informations** qui, une fois validées, sont **envoyées au serveur**. La réponse est alors plus ciblée à la demande.

# Construction d'un formulaire

- Un **formulaire** est contenu dans un élément **<form>** et souvent est composé de **champs** **<input>**, **<textarea>**,... et d'un **bouton d'envoi**
- HTML5 permet, via un formulaire, d'**entrer des données** mais son but n'est **pas de les traiter** (le serveur peut le faire avec PHP, MySQL,...) **ni de les valider** (le client peut le faire avec JavaScript, s'il n'est pas désactivé)
- On définit **quelles sont les données** à envoyer au serveur, **où** et **par quelle méthode**

# Élément <form>

- L'élément **<form>** contient les autres éléments du formulaire. Il est défini par ses attributs :
  - **action** : URL où sont envoyées les données
  - **method** : méthode d'envoi HTTP au serveur
  - **name** : nom du formulaire
  - **enctype** : type MIME associé au contenu lors de l'envoi au serveur
  - **novalidate** : le navigateur ne doit pas valider les données avant l'envoi au serveur
  - Et aussi : **accept-charset, autocomplete, target**

# Méthodes GET et POST

- La méthode par défaut est **GET**
  - Les données envoyées sont **écrites dans l'URL** demandée (via paires de clé/valeur)
  - **Longueur limitée** (varie selon le navigateur ou le serveur : 255, 2048, 64k,...)
  - **A éviter** pour envoyer des **données sensibles**
  - Utile pour **débugger** et les **bookmarks**
- La méthode **POST est privilégiée**
  - N'apparaît pas à l'écran, donc plus sûre et pas de taille limite

# Élément `<input>`

- L'élément `<input>` a plusieurs fonctionnalités et apparences selon son attribut **type** :
  - **text** : champ de texte
  - **password** : champ mot de passe
  - **submit** : bouton d'envoi
  - **reset** : bouton de remise à zéro
  - **radio** : bouton radio (un seul choix)
  - **checkbox** : case à cocher (plusieurs choix)
  - **button** : bouton
  - **hidden** : champ caché (invisible)
  - **image** : bouton graphique cliquable
  - **file** : sélection d'un fichier

# Nouveaux types <input>

- Des nouveaux types ont été ajoutés avec HTML5 (certains navigateurs proposent un affichage spécifique)
  - **tel** : numéro de téléphone
  - **url** : adresse, URL
  - **email** : courriel
  - **search** : champ de recherche
  - **datetime-local** : date et heure
  - **range** : valeur entre 2 bornes
  - **date** : date
  - **time** : heure
  - **month** : mois
  - **week** : semaine
  - **number** : nombre
  - **color** : couleur

# Attributs pour <input>

- Avec l'élément <input>, on a les attributs :
  - **name** : définit le nom du champ
  - **value** : définit la valeur transmise
- L'attribut **for** permet de **lier** un **<input>** avec un **<label>**
- Selon les types, les **autres attributs** sont :
  - multiple** (email, file), **checked** (radio, checkbox),
  - accept** (file), **height/width/alt/src** (image),
  - min/max/step** (date, time, datetime-local, month, week, number, range)



# Exemple

```
<form method="GET" name="login" action="test.php">  
Identifiant : <input type="text" name="nom">  
Mot de passe : <input type="password" name="mdp">  
<input type="submit" value="Se connecter">  
</form>
```

```
<?php  
    if (isset($_GET['nom']) && isset($_GET['mdp'])) {  
        echo "Bienvenue " . $_GET['nom'] . ", sur le site !";  
    }  
?>
```

# Zone de texte `<textarea>`

- Il permet une entrée de **texte multiligne** entre les balises `<textarea>` et `</textarea>`
- Les attributs spécifiques sont :
  - **rows** : nombre de lignes par défaut
  - **cols** : nombre de colonnes par défaut
  - **wrap** : retours à la ligne (hard) ou non (soft)
- Si un style CSS est spécifié, les attributs rows et cols sont ignorés

# Liste de choix

- On a une combinaison de l'élément **<select>** contenant des éléments **<option>**
- L'attribut **size** représente le nombre d'options et on peut en choisir plusieurs avec **multiple**
- L'élément **<option>** est caractérisé par les attributs **value**, **name**, **label** et **selected**
- On peut **regrouper visuellement** des options avec l'élément **<optgroup>**

# Radiobouton et checkbox

- Retour sur certains éléments `<input>`
  - **Radio-bouton** : `<input type="radio">`
  - **Checkbox** : `<input type="checkbox">`
- Souvent, on les utilise **en groupe**
  - Radio-bouton : **un seul coché** par groupe
  - Checkbox : **0, 1 ou plusieurs** cochés
- Pour les grouper, l'attribut **name** doit **être le même**, on les distingue par l'attribut **value** et l'attribut **checked** indique celui qui est coché



