

INF2005 – Programmation web

Formulaires

Jacques Berger

# Objectifs

Comprendre les formulaires HTML

# Prérequis

HTML

PHP

# Utilité

Saisir des données

Interaction avec l'utilisateur

Les données seront envoyées au serveur web ou manipulées par une application Javascript

# Attribut name

Les champs de formulaire ont un attribut optionnel nommé name

Le name est le nom du champ d'un point de vue de programmation

# Attribut name

Lorsque le formulaire sera envoyé au serveur, l'attribut name d'un champ deviendra le nom de la variable pour ce champ au niveau du serveur

C'est aussi le nom de la variable dans la requête HTTP

# Contrôles

## Boîte de texte

Un champ texte (textbox)

Sur une seule ligne

```
<input type="text" name="prenom" size="40">
```

# Contrôles

## Boîte de mot de passe

Un champ texte identique au précédent  
Sauf que les caractères sont cachés

```
<input type="password" name="pwd" size="30">
```



# Contrôles

## Zone de texte

Sur plusieurs lignes

Pour les descriptions et longs textes

```
<textarea name="description" cols="60" rows="5">Un long  
texte.</textarea>
```

# Contrôles

## Bouton

2 types de bouton avec 2 balises

Une forme de bouton pour les formulaires  
(input)

Une forme de bouton plus flexible  
(button)

# Contrôles

## Bouton pour formulaire

3 sortes de boutons pour formulaires

Avec l'élément input

L'attribut name est moins important dans  
ce cas

```
<input type="reset" value="Vider">  
<input type="submit" value="Envoyer">  
<input type="button" value="Cliquez ici">
```

# Contrôles

## Boutons pour formulaire

### Type reset

Vide le formulaire

### Type submit

Envoie le formulaire

### Type button

Un simple bouton cliquable

# Contrôles

## Bouton plus flexible

- Utilise la balise button
- Permet les images

```
<button type="button" value="calculate">Calculer le  
remboursement</button>
```

```
<button type="button" value="follow">  
    
</button>
```

# Contrôles

## Case à cocher

Uniquement le contrôle, n'inclut pas  
la description

```
<input type="checkbox" name="promo" checked="checked" value="1">  
<span>Cochez cette case pour recevoir des offres  
promotionnelles</span>
```

# Contrôles

## Bouton radio

Choix à cocher

Un seul bouton radio peut être  
sélectionné à la fois

L'attribut name doit avoir la même valeur  
pour tous les choix

# Contrôles

```
<span>Votre salaire annuel :</span><br>
<input type="radio" name="salary" value="1" checked="checked">
<span>Moins de 20 000$</span><br>
<input type="radio" name="salary" value="2">
<span>20 000$ à 29 999$</span><br>
<input type="radio" name="salary" value="3">
<span>30 000$ à 39 999$</span><br>
<input type="radio" name="salary" value="4">
<span>Plus de 40 000$</span><br>
```



# Contrôles

Pour afficher un bouton radio ou une case à cocher comme étant coché, on utilise l'attribut `checked`

```
<input type="checkbox" checked="checked" name="publicity" value="1">
```

# Contrôles

Liste déroulante

Combo Box

Choisir une valeur parmi une liste

Liste : élément select

Élément dans la liste : élément option

```
<select name="province">  
  <option value="qc">Québec</option>  
  <option value="on">Ontario</option>  
</select>
```

# Contrôles

## Liste déroulante

### Regroupement de valeur : optgroup

```
<select name="language">
  <optgroup label="Web">
    <option value="html4">HTML 4.01</option>
    <option value="html5">HTML 5</option>
    <option value="css2">CSS 2.1</option>
    <option value="css3">CSS 3</option>
  </optgroup>
  <optgroup label="Dynamique">
    <option value="ruby">Ruby</option>
    <option value="groovy">Groovy</option>
  </optgroup>
</select>
```

# Contrôles

## Liste déroulante avec contenu réutilisable

```
<input name="language" list="progweb">
```

```
<datalist id="progweb">  
  <option value="PHP">  
  <option value="Perl">  
  <option value="Python">  
</datalist>
```

# Contrôles

## Label (étiquette)

Texte associé à un champ

Cliquer sur le label active le champ associé

L'association est faite avec l'attribut for

```
<label for="nom">Nom :</label>
```

```
<input type="text" id="nom" name="name" size="40">
```

# Contrôles

## Champ caché

Envoyer de l'information au serveur que l'utilisateur n'a pas besoin de voir ou de connaître

```
<input type="hidden" name="rating" value="good">
```

# Contrôles

D'autres types d'élément input :  
Le support varie selon les fureteurs

color  
date  
datetime  
datetime-local  
email  
month  
number  
range  
search  
tel  
time  
url  
week

# Fonctionnalités

Certains contrôles peuvent être en lecture seule  
Attribut readonly

```
<textarea cols="60" rows="5" readonly="readonly">Celui-ci est en  
lecture seule.</textarea>
```



# Fonctionnalités

Les contrôles peuvent être désactivés  
Attribut disabled

```
<button type="button" value="off"  
disabled="disabled">Désactivé</button>
```

# Formulaire

Le formulaire HTML permet d'envoyer les valeurs des différents contrôles au serveur web de façon automatique

On crée un formulaire avec la balise form

# Formulaire

## On place les contrôles à l'intérieur du formulaire

```
<form action="address.php" method="get">
  <div>
    <p>Entrez votre adresse</p>
    <label for="adresse">Adresse : </label>
    <input type="text" id="adresse" name="adresse" size="50"><br>
    <label for="ville">Ville : </label>
    <input type="text" id="ville" name="ville" size="35"><br>
    <label for="province">Province : </label>
    <select id="province" name="province">
      <option value="qc">Québec</option>
      <option value="on">Ontario</option>
    </select><br>
    <label for="cp">Code postal : </label>
    <input type="text" size="6" id="cp" name="cp"><br>
    <button type="reset">Vider</button>
    <button type="submit">Soumettre</button>
  </div>
</form>
```

# Formulaire

Si les contrôles ne sont pas dans un formulaire (form), il faut gérer leur contenu et la requête au serveur nous-même (ex. avec Javascript)

# Formulaire

Des fois, il est difficile de placer tous les input dans l'élément form, on peut donc les dissocier

```
<form action="test.php" method="post" id="principal">  
  ...  
</form>
```

```
<input type="text" name="prenom" form="principal">
```

# Formulaire

Lorsqu'on soumet un formulaire (bouton submit), le formulaire va construire une URL pour envoyer une requête au serveur web

Les valeurs du formulaire seront ajoutées comme paramètres à l'URL ou à la requête HTTP

# Balise form

2 attributs importants

action

method

action : l'adresse où le formulaire doit envoyer ses données

# Balise form

method : deux valeurs possibles

get

post

Avec get, les valeurs du formulaire seront visibles dans l'URL

Avec post, les valeurs du formulaire ne seront pas visibles pour l'utilisateur



# Balise form

Note : si vous envoyez des données qu'il vaut mieux ne pas montrer à l'utilisateur (ex. mot de passe), utilisez post

# Paramètres de l'URL

Voici comment les paramètres sont encodés dans l'URL :

```
http://www.jberger.org/get_address.php?  
adresse=2685+Bellerose&ville=Longueuil&province=qc&cp=J4L+9R4
```

# Paramètres de l'URL

Le caractère ? sépare l'adresse des paramètres

Le caractère & sépare chaque paramètre

Les paramètres possèdent un nom et une valeur  
ville=Longueuil

# Paramètres de l'URL

Les attributs name des contrôles sont les noms des paramètres

Les attributs value des contrôles sont les valeurs des paramètres

# Côté serveur

Pour accéder aux données d'un formulaire dans un script PHP, on utilise les variables suivantes :

`$_GET`

`$_POST`

# Côté serveur

On utilise `$_GET` lorsque la méthode du formulaire est get

On accède au paramètre par son nom (attribut name)

```
$ville = $_GET["ville"];  
$adresse = $_GET["addr"];
```

# Côté serveur

On utilise `$_POST` lorsque la méthode du formulaire est post

On accède au paramètre par son nom (attribut name)

```
$password = $_POST["motdepasse"];  
echo $_POST["firstname"];
```

# Côté serveur

Lorsque vous ne savez pas si on vous appelle avec get ou post, vous pouvez utiliser `$_REQUEST` qui contient les valeurs de `$_GET`, `$_POST` et `$_COOKIE`

```
$password = $_REQUEST["motdepasse"];
```



# Côté serveur

Si on veut utiliser une variable dans `$_GET` ou `$_POST` mais que le navigateur ne l'a pas envoyé, cela provoquera une erreur

La fonction `isset` sert à vérifier si la variable existe avant de l'utiliser