

# Développement Web

Internet, xHTML, CSS

# SOMMAIRE

- \* Internet et web : présentation
- \* W3C : Standard et validation
- \* xHTML : Introduction
- \* xHTML : Les tableaux
- \* xHTML : Les formulaires
- \* xHTML et CSS

# Internet et Web

- \* Objectifs :

- \* Définir ce qu'est internet
  - \* Définir ce qu'est le Web
  - \* Définir le modèle Client/serveur.

# Internet : C'est quoi?

- \* Réseaux informatique mondial
- \* IP
- \* Non-centralisé : « Réseau de réseaux »
- \* Ce n'est pas le Web!

# World Wide Web

- \* 1 application d'internet
- \* Liens HyperTextes
- \* HTTP(s)
- \* Texte & divers Médias

# xHTML : langage web

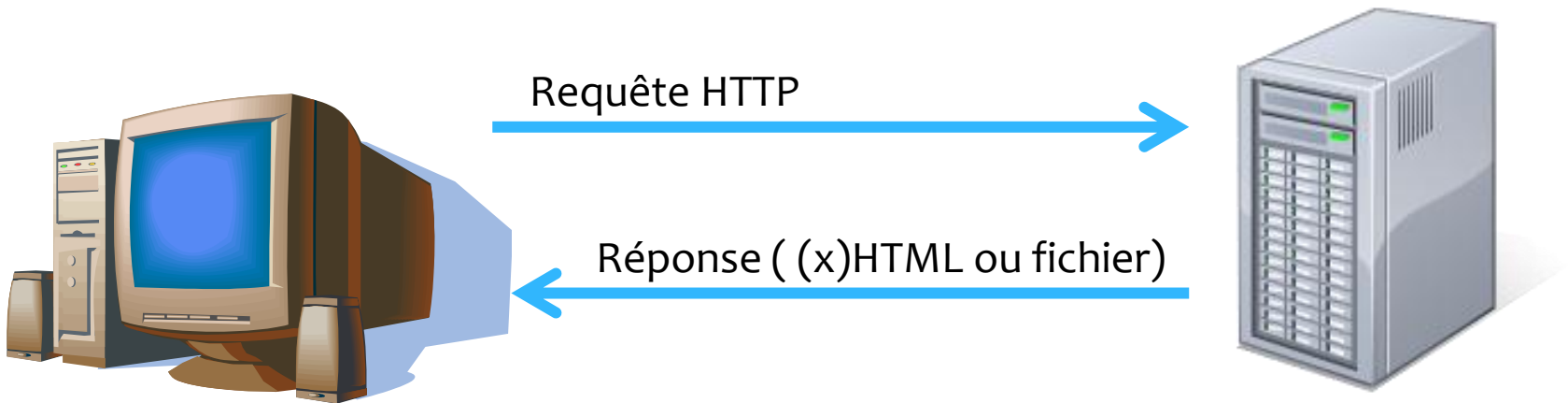
- \* eXtensible HyperText Markup Language
- \* Mise en forme d'un document
- \* Langage de description
- \* Contexte hyperTexte
- \* Facile à utiliser
- \* Extension possible

# Modèle Client/Serveur

- \* HTTP : HyperText Transfer Protocol
  - \* Développé pour le web
  - \* Client (Browser)
  - \* Serveur (Web Server)
  - \* Transfert de donnée entre le browser et le serveur Web
- \* Autres protocoles:
  - \* FTP, SMTP, POP ...

# Modèle Client/Serveur

\* HTTP procédure de communication:





# Modèle Client/serveur

- \* Serveur Web

- \* Reçoit les requêtes HTTP et envoie un fichier en retour (xHTML ou autre)
- \* Exécute les programmes chargés.
- \* 2 serveurs dominant le marché:
  - \* Apache
  - \* IIS

# Modèle Client/serveur

## \* Répartition des Web Browser (mars 2010)



# W3c Standard et validation

- \* Objectifs:
  - \* Découvrir ce qu'est le W3c
  - \* Découvrir la validation
  - \* Apprendre à valider un site
  - \* Connaître les outils de validation

# Qu'est ce que le W3c?

- \* World Wide Web Consortium
- \* Organisme de normalisation
- \* Créé en 1994
- \* Géré par le MIT, l'ERCIM et l'université Keio (Japon)
  - \* MIT: Massachusetts Institute of Technology
  - \* ERCIM: European Research Consortium for Informatics and Mathematics



# Que fait le W3c?

- \* Promeut

La compatibilité des technologies Web (recommandation)

- \* Crée des standards comme :

- \* HTML, XHTML, CSS, SVG, XML, WSDL ...

- \* Plus d'informations sur : [www.w3.org](http://www.w3.org)

# La Validation

- \* Pourquoi faire?
  - \* Vérifier le respect des normes
- \* Pourquoi respecter les normes?
  - \* Une accessibilité universelle
  - \* La portabilité
  - \* La rapidité de production
  - \* Un contrôle facilité
  - \* Des documents durables
  - \* ...

# Les types de validation

- \* Pour le xHTML :
  - \* Transitional.
  - \* Strict.
  - \* Autre ...

# Validation « transitional »

- \* Utilisation de la normalisation ... progressive
- \* Accepte quelques balises bannies
  - \* Comme
    - \* `<font />`
    - \* `<center />`
- \* Non recommandé



# Validation « strict »

- \* Respect total de la norme
- \* N'accepte pas les balises de style
  - \* Exemple:
    - \* `<font />`
    - \* `<center />`
- \* Recommandé!

# Validation: Comment procéder?

- \* <http://validator.w3.org>
- \* Possibilités de validation:
  - \* Liens direct
  - \* Upload de fichier
  - \* Copie du code source
- \* Limitation :
  - \* Ne valide qu'une page à la fois!
- \* Avertissement:
  - \* Si vous avez plusieurs erreurs, corrigez la premier et revalidez!

# Validation: autres outils

- \* <http://jigsaw.w3.org/css-validator/> Validation des CSS
- \* <http://www.w3.org/Status.html>
  - \* Regroupe divers outils comme:
    - \* Vérificateur de liens
    - \* Validation pour les Mobiles
    - \* ...

# xHTML

- \* Objectifs :

- \* Être capable d'écrire un site avec xHTML.
- \* Présenter des données dans des tableaux.
- \* Envoyer des données via un formulaire.

# Fichier xHTML

- \* Extension de fichier : .html ou .htm (hérité 8.3)
- \* Construit par des balises prédéfinis
- \* Exemple:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>CSS Example</title>
  </head>
  <body>
    <h1>Shopping List</h1>
    <p>Please <strong>don't</strong> forget the following:</p>
    <ul>
      <li>Large tomatoes</li>
      <li>Red bell peppers</li>
      <li>Eggplant</li>
    </ul>
  </body>
</html>
```

# Les balises

- \* Entourées par < et >
- \* 2 types de balises
  - \* Les balises par paires:
    - \* `<h1> Un titre très important </h1>`
  - \* Les balises vides:
    - \* `<br />`
- \* Règle importante : les balises sont toujours en minuscule.

# Les balises

- \* Balises « block » :
  - \* S'affiche les uns sous les autres.
  - \* Utilisé pour structuré la page en différent « block ».
  - \* Peut contenir des éléments « block » et/ou « inline ».
- \* Balises « inline » :
  - \* Doit être contenu dans un « block »
  - \* Mise en forme d'un groupe de mots
  - \* S'affiche les uns à côté des autres
  - \* Peut contenir d'autre « inline » mais pas de « block »

# Structure

1. `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
2. `<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">`
3. `<head>`
4. `</head>`
5. `<body>`
6. `</body>`
7. `</html>`



# Explication L.1

- \* Le « DOCTYPE » indique au navigateur à quelle DTD (Document Type Definition), il est conforme
- \* Le « DOCTYPE » doit être placé avant l'élément « racine » du document
- \* Nous choisirons toujours une validation « strict »

# Explication L.2 et L.7

- \* « Racine » du document
- \* La balise html est de type block
- \* Contient un entête « <head> » et un corps « <body> »
- \* 2 attribut sont requis
  - \* Xmlns
  - \* Xml:lang
- \* En xHtml 1.1, il est obligatoire de déclarer le namespace utilisé et la langue.

# Explication L.3 et L.4

- \* < head >, balise de type « block »
- \* Non affiché
- \* Peut contenir:
  - \* Le titre du document : <title> ... </title>
  - \* Du css : <style>... </style> ou <link />
  - \* Du javascript : <script>... </script> ou <link />
  - \* Des aides au référencement : <meta />

# Explication L.5 et L.6

- \* < body >, balise de type « block »
- \* Contient les éléments décrivant le contenu du document.

# xHTML: Les titres

\* `<hn> ... </hn>` , n compris entre 1 et 6

**C'est un titre h1**

**C'est un titre h2**

**C'est un titre h3**

**C'est un titre h4**

**C'est un titre h5**

**C'est un titre h6**

# xHTML: Les paragraphes

- \* `<p>... </p>`
- \* Définit un paragraphe
- \* Retour à la ligne automatique.
- \* Exemple :
  1. ...
  2. `<body>`
  3. `<p>Un premier paragraphe</p>`
  4. `<p>Un second paragraphe</p>`
  5. `</body>`
  6. `</html>`

# xHTML: les listes

## Listes numérotés

Définit par <ol>... </ol>

Chaque élément est défini par: <li>... </li>

```
<body>  
  <ol>  
    <li>HTML</li>  
    <li>Javascript</li>  
  </ol>  
</body>
```

1. HTML
2. Javascript

## Listes à puces

Définit par <ul>... </ul>

```
<body>  
  <ul>  
    <li>HTML</li>  
    <li>Javascript</li>  
  </ul>  
</body>
```

- HTML
- Javascript

# xHTML : Images

- \* `<img />` , balise vide
- \* 2 attributs requis :
  - \* `src="url"`
  - \* `alt= "texte alternatif"`
- \* Attribut `title` : info-bulle.
- \* Exemple : ``
- \* Règles sur les attributs:
  - \* Toujours dans la balise ouvrante.
  - \* Nom des attributs en minuscule.



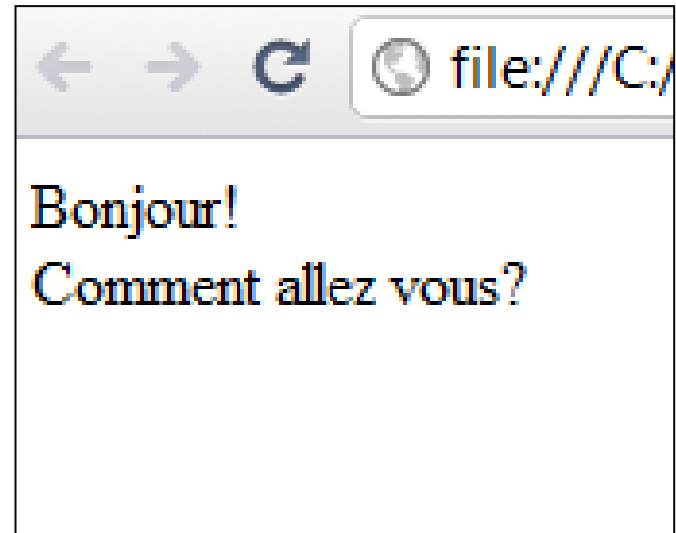
# xHTML: saut de ligne

- \* Balise de saut de ligne : `<br />`

- \* Balise vide.

- \* Exemple:

1. `<body>`
2. `<p>`
3. `Bonjour!<br/> Comment`
4. `allez vous?`
5. `</p>`
6. `</body>`



# xHTML: Liens et ancre

- \* `<a>...</a>`
- \* Utiliser pour créer:
  - \* Un lien à un autre document.
  - \* Un saut à une section du document.
- \* Exemple :
  1. `<p>`
  2. `<a href="http://site.fr/chemin/fichier.html">Texte du liens</a>`
  3. `</p>`

# xHTML: Liens et ancre

- \* Allez à une ancre :
  - \* Utilisation d'un liens avec #NomAncre
  - \* Exemple : `<a href="index.html#Chap2">Chapitre 2</a>`
- \* Création d'un ancre :
  - \* Avec la balise `<a>` :
    - \* `<h1><a name="Chap2"> Chapitre 2</a></h1>`
  - \* Avec l'attribut `id` :
    - \* `<h1 id="Chap2"> Chapitre 2</h1>`

# xHTML : Commentaire

- \* Non affiché par le Web Browser
- \* Utilités:
  - \* Reprise du projet après une longue pause.
  - \* Reprise du projet par un autre développeur.
  - \* Travail en équipe
  - \* ...
- \* <!-- Ceci est un commentaire qui vous sera sans doute utile -->

# xHTML: les caractères spéciaux

- \* Utilisation principale :

- \* &# suivit de la valeur Unicode

Copyright ©

- <p> je veux des espaces! </p>

je veux des espaces!

je veux des espaces!

# xHTML: Les accents

- \* Cas spécial pour les accents
- \* « & »+la lettre utilisé + abréviation + « ; »

Description	Abréviation
Accent grave	grave
Accent aigu	acute
Accent circonflexe	circ
Cédille	cedil
Tréma (ex: ï, ä ...)	uml
Tilde (ex: ã)	tilde

# xHTML : Les accents

## \* Exemples :

Code	Résultat
&eacute;	é
&acirc;	â
&ccedil;	ç
&auml;	ä
&tilde;	~

# xHTML: Les tableaux

- \* À utiliser pour :
  - \* Classer des données
  - \* Afficher des données en tableau
- \* Mauvais usage :
  - \* Utilisation pour le design.
  - \* Utilisation d'attribut interdit. (border, ...)



# xHTML: Les tableaux

- \* Structure principale :
  - \* `<table>... </table>`
  - \* Diviser en ligne : `<tr>... </tr>` (« table rows »)
  - \* Chaque ligne peut contenir :
    - \* Des entêtes : `<th>... </th>` (« table header »)
    - \* Des données: `<td>... </td>` (« table data »)
  - \* Les balises `<th>` et `<td>` forment des cellules.

# xHTML: Les tableaux

## \* Exemple:

```
1.  <table>
2.    <tr>
3.      <th>Ligne 1, Entête 1</th>
4.      <th>Ligne 1, Entête 2</th>
5.    </tr>
6.    <tr>
7.      <td>Ligne 2, Cellule 1</td>
8.      <td>Ligne 2, Cellule 2</td>
9.    </tr>
10.   <tr>
11.     <td>Ligne 3, Cellule 1</td>
12.     <td>Ligne 3, Cellule 2</td>
13.   </tr>
14. </table>
```

# xHTML: Les tableaux

- \* Fusionner des cellules
  - \* colspan: largeur en nombre de cellules.
  - \* rowspan: hauteur en nombre de lignes.
- \* Exemple:

```
1. <table>
2.   <tr>
3.     <th colspan="2">Entête</th>
4.   </tr>
5.   <tr>
6.     <td>1</td>
7.     <td rowspan="2">Fusionne 2 lignes</td>
8.   </tr>
9.   <tr>
10.    <td>2</td>
11.  </tr>
12.  <tr>
13.    <td colspan="2">Fusionne 2 cellules</td>
14.  </tr>
15. </table>
```

# xHTML: Les tableaux

- \* Attribut interdit :

- \* border

- \* align

- \* valign

- \* width

- \* height

- \* ...

# xHTML: Les formulaires

- \* Objectifs:

- \* Découvrir ce qu'est les formulaires
- \* Créer des formulaires

# xHTML: Les formulaires

- \* Utilisation
- \* Important pour du web dynamique
- \* Permet au utilisateur d'insérer des données

# xHTML: Les formulaires

- \* Structure principale :

1. `<form id="myForm" action="page.html" method="post">`
2. `<!-- Form codes -->`
3. `</form>`

- \* action: page à laquelle on va envoyer les données
- \* method: méthode d'envoi
- \* L'attribut « id » est obligatoire.
- \* L'attribut « name » est interdit.

# xHTML: Les formulaires

- \* Méthode d'envoi « GET »:
  - \* Les données sont passées par l'URL.
  - \* Problèmes de sécurité.
  - \* Autorise seulement 255 caractères.
- \* Exemple d'url avec données GET :
  - \* `monsite.fr/login.php?pseudo=michu&password=123456`



# xHTML : Les formulaires

- \* Méthode d'envoi « POST » :
  - \* Les données sont passé par l'header http.
  - \* Les données sont invisible pour l'utilisateur.

# xHTML: Les formulaires

- \* `<input />` type « text »
  - \* Affiche un champs de saisie de texte.
  - \* Attributs optionnels : size, maxlength

- \* Exemple:

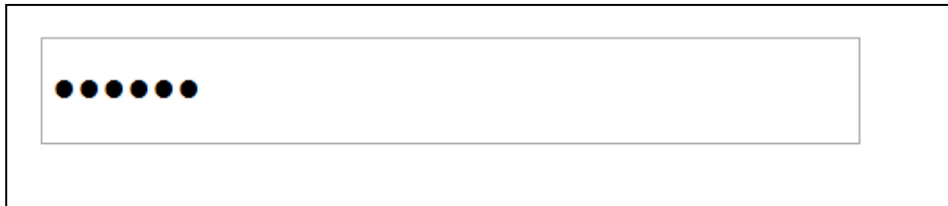
- \* `<input type="text" name="prenom" value="champs texte" />`

champs texte

# xHTML: Les formulaires

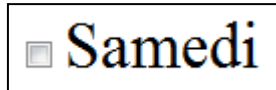
- \* `<input />` type « password »
  - \* Les caractères saisis sont masqués par des \*.
- \* Exemple:

```
* <input type="password" name="pass" value="123456" />
```



# xHTML: Les formulaires

- \* `<input />` type « checkbox »
  - \* Affiche une case cochable/décochable
  - \* Attribut optionnel « checked »
- \* Exemple :
  - \* `<input type="checkbox" name="jour" value="6" checked="checked" />`Samedi



# xHTML: Les formulaires

- \* `<input />` type « radio »
  - \* L'utilisateur pour choisir une possibilité parmi plusieurs.
  - \* Attribut optionnel « checked »

- \* Exemple :

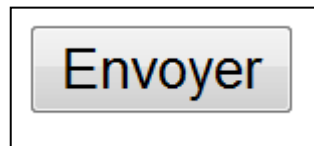
1. `<input type="radio" name="sex" value="m" />` Homme
2. `<input type="radio" name="sex" value="f" checked="checked" />` Femme

☐ Homme      ☒ Femme

# xHTML: Les formulaires

- \* `<input />` type « submit »
  - \* Boutton.
  - \* Envoie les données du formulaire.
- \* Exemple :

\* `<input type="submit" value="Envoyer" />`



# xHTML: Les formulaires

- \* `<input />` type « reset »
  - \* Bouton.
  - \* Réinitialise toute les valeurs du formulaire.
- \* Exemple :
  - \* `<input type="reset" value="reset" />`



# xHTML: Les formulaires

- \* `<input />` type « button »
  - \* Bouton.
  - \* N'implémente pas de fonctionnalité spécifique.
- \* Exemple :

1. `<input type="button" name="" value="Pas d'action" />`





# xHTML: Les formulaires

- \* `<input />` type « image »
  - \* Affiche une image.
  - \* Fonctionne comme un bouton « submit ».

- \* Exemple :

- \* `<input type="image" src="go.png" alt="Go" />`



- \* Note: Envoie aussi les coordonnées du clic sur l'image.

# xHTML: Les formulaires

- \* `<input />` type « hidden »

- \* N'est pas affiché.

- \* Exemple :


1. `<input type="hidden" name="cache" value="invisible" />`

# xHTML: Les formulaires

- \* `<input />` type « file »
  - \* Sélection de fichier.
  - \* Requièrè la méthode POST et l'attribut « enctype ».

- \* Exemple :

```
1. <form id="form1" method="post" enctype="multipart/form-data">
2.     <input type="file" name="file_send" />
3. </form>
```

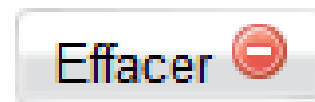
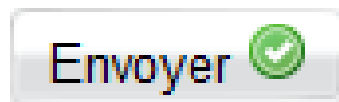
A screenshot of a web form element. It consists of a rectangular box with a thin black border. Inside the box, on the left, is a small button with the text "Choisissez un fichier". To the right of this button, the text "Aucun fi... choisi" is displayed, where "fi..." is truncated.

# xHTML: Les formulaires

- \* La balise <button>
- \* Accepte du code HTML à l'intérieur

- \* Exemple :

```
1. <button name="send" type="submit">
2.     Envoyer 
3. </button>
4. <button name="delete" type="reset">
5.     Effacer 
6. </button>
```



# xHTML: Les formulaires

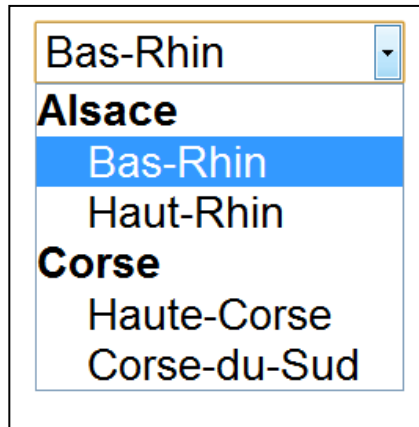
- \* La balise `<select>`
  - \* Permet de sélectionner un élément dans une liste
  - \* Les items sont dans des balises `<option>`
  - \* qui peuvent être groupé avec `<optgroup>`

# xHTML: Les formulaires

## \* Exemple :

```
<select name="FrenchDepartments">
  <optgroup label="Alsace">
    <option value="67">Bas-Rhin</option>
    <option value="68">Haut-Rhin</option>
  </optgroup>
  <optgroup label="Corse">
    <option value="2B">Haute-Corse</option>
    <option value="2A">Corse-du-Sud</option>
  </optgroup>
</select>
```

## \* Résultat :



Bas-Rhin
<b>Alsace</b>
Bas-Rhin
Haut-Rhin
<b>Corse</b>
Haute-Corse
Corse-du-Sud

# xHTML: Les formulaires

- \* La balises `<label>`
  - \* Permet d'identifier un contrôle du formulaire

- \* Implicite :

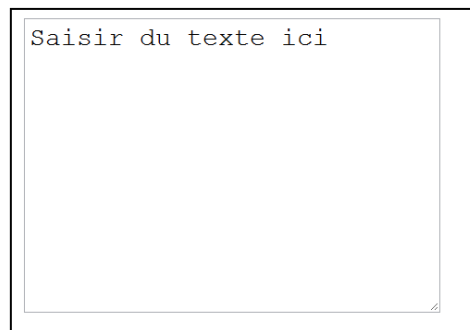
```
* <label>  
*   <input type="radio" name="sexe" value="h" /> Homme  
* </label>
```

- \* Explicite

```
* <input type="radio" name="sexe" value="f" id="femme" />  
* <label for="femme">Femme</label>
```

# xHTML: Les formulaires

- \* La balise <textarea>
- \* Champs texte pouvant contenir plusieurs lignes
- \* Attribut : cols , rows
- \* Exemple:
  - \* `<textarea cols="25" rows="10" name="comments">`
  - \* *Saisir du texte ici*
  - \* `</textarea>`



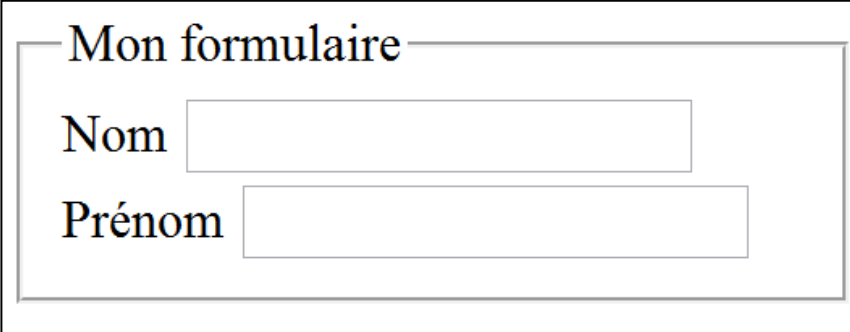


# xHTML: Les formulaires

- \* <fieldset> et <legend>
  - \* Regroupe les contrôles par thèmes

- \* Exemple :

```
<fieldset>  
  <legend>Mon formulaire</legend>  
  <label>Nom<input type="text" name="nom" /></label>  
  <label>Prénom<input type="text" name="prenom" /></label>  
</fieldset>
```



Mon formulaire

Nom

Prénom

The image shows a visual rendering of the xHTML code provided. It features a rectangular frame containing a legend at the top with the text 'Mon formulaire'. Below the legend, there are two labels: 'Nom' and 'Prénom'. Each label is followed by a text input field. The 'Nom' field is a single-line text box, and the 'Prénom' field is a single-line text box. The entire form is enclosed in a double-line border.

# xHTML: Les formulaires

- \* Quelques attributs spécifiques
  - \* type="hidden"
  - \* readonly="readonly"
    - \* Sur <input /> ou <textarea />
  - \* disabled="disabled"
    - \* Sur <input />, <textarea />, <select >, <button />, <option> ou <optgroup />

# CSS

- \* Objectifs:

- \* Apprendre ce qu'est le CSS
- \* Apprendre comment utiliser le CSS

# CSS

- \* CSS : Cascading Style Sheet
- \* Langage différent de (x)HTML
- \* Permet de modifier l'apparence d'un page (x)HTML

# CSS

- \* Avantages:

- \* Centralise les éléments de mise en page dans un fichier externe
- \* Sépare le design du document
- \* Simplifie la maintenance des sites web
- \* Réduit la taille de la page
- \* Mise en cache du CSS possible.

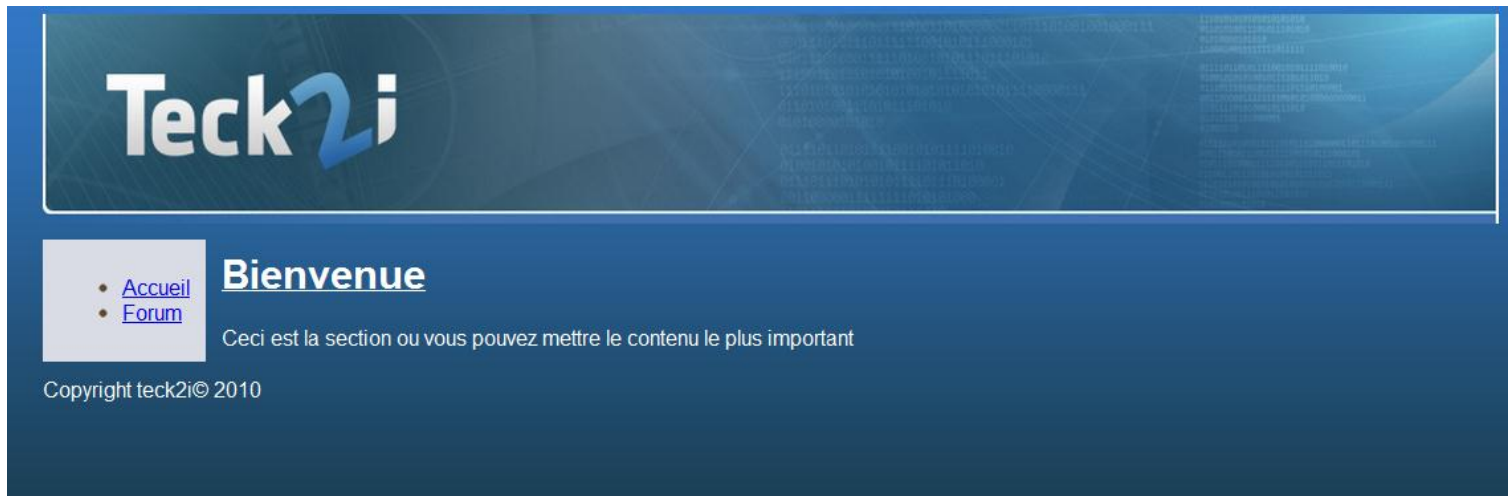
# CSS

- \* `<div>...</div>`
- \* Définit une section (division) de la page

```
* <body>
*   <div id="banner"></div>
*   <div id="menu">
*     <ul>
*       <li><a href="home.html">Accueil</a></li>
*       <li><a href="forum.html">Forum</a></li>
*     </ul>
*   </div>
*   <div id="main">
*     <h1>Bienvenue</h1>
*     <p>Ceci est la section ou vous pouvez mettre le contenu le
plus important</p>
*   </div>
*   <div id="footer">
*     Copyright teck2i© 2010
*   </div>
* </body>
```

# CSS

\* Nous donne avec un peu de CSS



# CSS

- \* Où le placer?

- \* Dans un fichier externe:

  - \* `<link href="main.css" media="all" rel="stylesheet" type="text/css" />`

- \* Dans la balise `<style>`:

  - \* `<head>`
  - \* `<style type="text/css">`
  - \* `p { color: blue; }`
  - \* `</style>`
  - \* `</head>`

- \* Dans les balises :

  - \* `<span style="color: red;">Texte rouge... </span>`



# CSS

- \* Syntaxe:

```
sélecteur{  
    Propriété : valeur;  
}
```

- \* Exemple:

```
a {  
    margin:5px 5px 10px 5px;  
    padding:6px;  
    border:solid 1px #b8b8b8;  
    background-color:#f4f5f5;  
}
```

# CSS

- \* Grouper plusieurs sélecteurs
- \* Exemple:
  - \* `h1 { background-color: white; color: red; }`
  - \* `p, h1 { font-family: Verdana, "Sans serif"; }`
- \* Le style est appliqué dans l'ordre de définition
- \* En cas de conflit, la dernière propriété écrase la précédente.

# CSS

- \* Les commentaires : `/* un commentaire */`

- \* Exemple :

```
/* Document font family */
```

```
body { font-family: Verdana, "Sans serif"; }
```

# CSS

## \* Quelques sélecteurs:

Sélecteur	sélectionne
*	tout les éléments
E	tout les éléments du type E
E F	tout les éléments F dans un élément E
E > F	tout les éléments F directement enfant de E
.maClasse	tout les éléments de classe « maClasse »
#monID	l'élément avec l'id « monID »
E:hover	l'élément E quand il est survolé par la souris

# CSS

- \* Attribut class
  - \* Applique un style à un groupe défini d'élément
- \* Exemple
  - \* CSS : `h2.red{ background-color: red; }`
  - \* Est appliqué au élément :
    - \* `<h2 class="red">Mon Titre</h2>`

# CSS

- \* Attribut id
  - \* Applique un style à 1 et 1 seul élément
- \* Exemple
  - \* CSS : `#red{ background-color: red; }`
  - \* Est appliqué a l'élément :
    - \* `<div id="red">Mon texte</div>`

# CSS

- \* Couleurs:

- \* Nom

- \* `rgb(x,x,x)`                      x entre 0 et 255

- \* `Rgb(x%,x%,x%)`                x entre 0 et 100

- \* `#rrggbb`                        entre 00 et FF

- \* Exemple:

- \* Red

- \* `rgb(255,0,0)`

- \* `Rgb(100%,0,0)`

- \* `#FF0000 (= #Foo)`

# CSS

- \* Unité de mesure

- \* %

- \* in, cm, mm

- \* em, ex

- \* pt, pc

- \* px



# CSS

- \* La liste des attributs est dans le fichier pdf
- \* Vous y trouver les attributs, les valeurs possibles et leurs fonctions.