

www

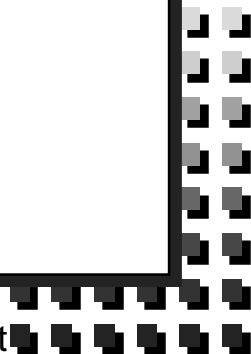
# Programmation sur Internet

## Cours : documents HTML et leur style CSS



Jean-Michel Ilié

IUT Paris 5 - département informatique



# Éléments de spécification HTML

- ☞ Un élément est repéré par une balise ou un jeu de balises (ouvrante-fermante)
- ☞ La spécification devient hiérarchique quand les éléments sont imbriqués
- ☞ De nombreux attributs sont disponibles pour paramétrer les éléments
- ☞ Les attributs manquants auront une valeur par défaut .

Element avec contenu html  
(spécification html hiérarchique)

```
<tag attr1="val1" ... attrn="valn"> ...  
        contenu textuel (html...)  
</tag>
```

Element vide  
(exple : un bouton, une image, ... )

```
<tag attr1="val1" ... attrn="valn" /> ...  
        <-- pas de contenu -->
```

# Documents HTML bien structuré

## Document HTML bien structuré

- ⇒ règles à respecter pour l'utilisation des balises
- ⇒ séparation de la gestion des styles (CSS) et du contenu HTML

## Apport

- ⇒ validateurs automatiques (html, css, ...)
- ⇒ meilleur partage du travail (séparation du travail sur le contenu et le graphisme).

# Règles d'or

1. écrire en minuscule
2. valeur d'attributs est mis entre "" ou ''
3. pas d'attribut déclaré sans valeur

KO `<input type="checkbox" checked/>`

OK `<input type="checkbox" checked="checked"/>`

4. imbrication des balises sans chevauchement

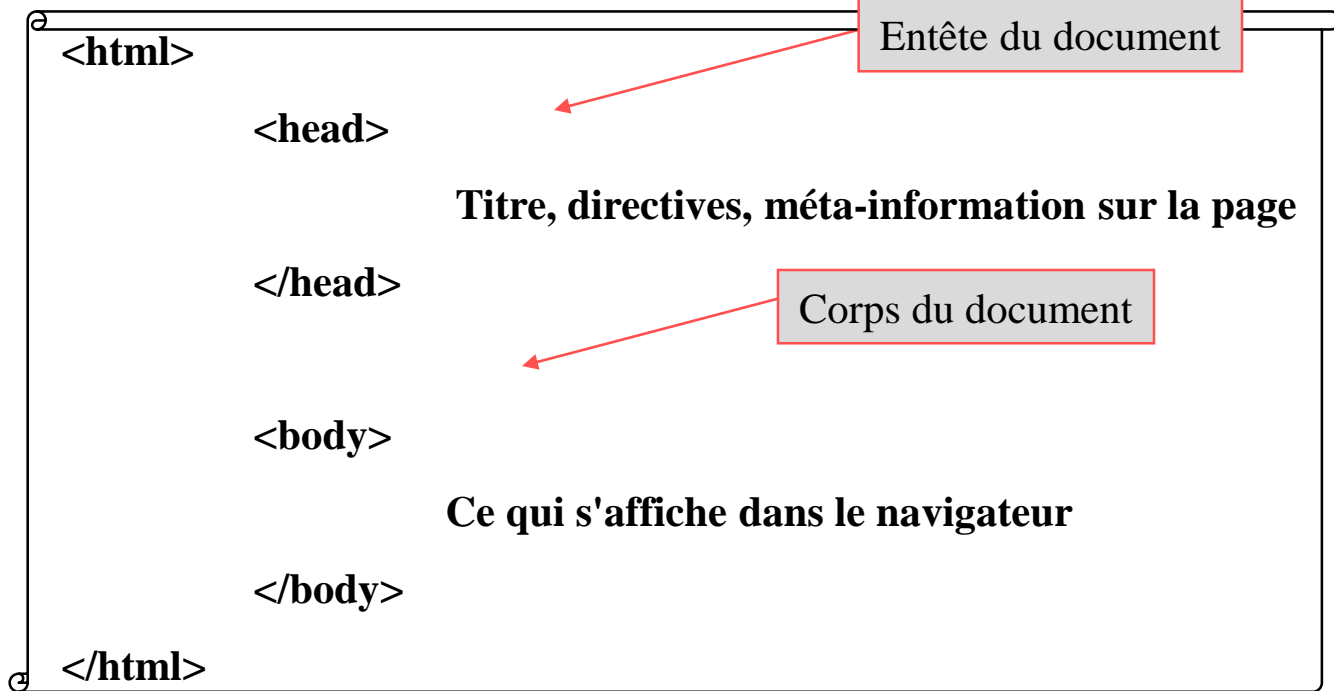
KO

```
<b>  
  <i> text  
</b>  
</i>
```

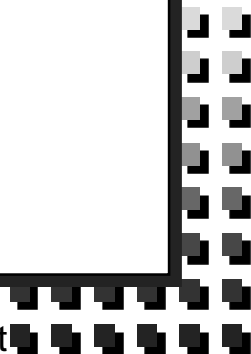
OK

```
<b>  
  <i> text  
  </i>  
</b>
```

# Structuration générale pour un document HTML



Commentaires : <- - commentaire d'un fichier HTML -->



# les balises META de l'entête

```
<meta name="nom-de-la-propriété" content="valeur" />
```

author  
keywords  
description

- ☞ Permet de décrire des propriétés générales du document
- ☞ Exploiter par les moteurs de recherche pour indexer les pages et connaître les changements

```
<meta http-equiv="refresh | expires" content="précision"/>
```

- ☞ Force un changement de page : 'slide show', redirection de pages ...
- ☞ refresh : sur délai d'échéance
- ☞ expire sur date fixée

# Exemple d'entête

```
<html><head>
<title> mon entête préférée</title>

<meta name= "keywords" content="entête méta"/>
<meta name = "authors" content="jmi"/>
<meta name = "description" content="essais des tags de type méta"/>

<meta http-equiv= "refresh" content="5 ; url://www.autre-page.html"/>
<meta http-equiv = "expires" content="Tues, 26 Sept 2008 09:30:02 gmt"/>
</head><body></body></html>
```

Les moteurs de recherche  
utilisent les mots-clefs pour indexer les pages

Après la date d'expiration (26 sept2008) :  
oblige le gestionnaire du cache à recharger la  
page à partir du serveur web.

Après expiration (5s) :  
Oblige le navigateur à redemander  
une URL (autre page à charger, ...)

# Corps du document HTML

`<body> ...corps... </body>`

Nombreux attributs optionnels pour la balise body :

- ☞ bgcolor : couleur de fond
- ☞ text : couleur du texte
- ☞ background : URL de l'image en fond d'écran
- ☞ link : couleur des liens non encore visités
- ☞ vlink : couleur des liens déjà visités
- ☞ alink : couleur des liens lorsque l'on clique dessus

```
<body  bgcolor="#FFFFFF"  
        background="images/FondIUT.jpg">  
...  
</body>
```

# Spécification HTML des couleurs

- ☞ 3 codes hexa ou octal #RGB (Rouge\*Vert\*Bleu)
- ☞ ou un nom symbolique : green, yellow, purple, red, blue, ...  
(plus simple mais le rendu est dépendant du navigateur !)

**Black** = "#000000"  
 Silver= "#C0C0C0"  
 Gray= "#808080"  
 Maroon= "#800000"  
 Red= "#FF0000"  
 Purple= "#800080"  
 Fuschia= "#FF00FF"  
**White**= "#FFFFFF"

Green= "#008000"  
 Lime= "#00FF00"  
 Olive= "#808000"  
 Yellow= "#FFFF00"  
 Navy= "#000080"  
 Blue= "#0000FF"  
 Teal= "#008080"  
 Aqua= "#00FFFF"

on peut aussi utiliser  
un logiciel de gestion d'image  
pour connaître les valeurs  
de couleur RGB  
pour un objet prédéfini

## Utilisations diverses :

```

<body bgcolor = "#00FF00 " ...> ... </body> [corps fond vert]
<font color = "#FF0000 " ...>... </font> [texte rouge]
<td bgcolor = "#0000FF" ...>...</td> [cellule fond bleu]
  
```

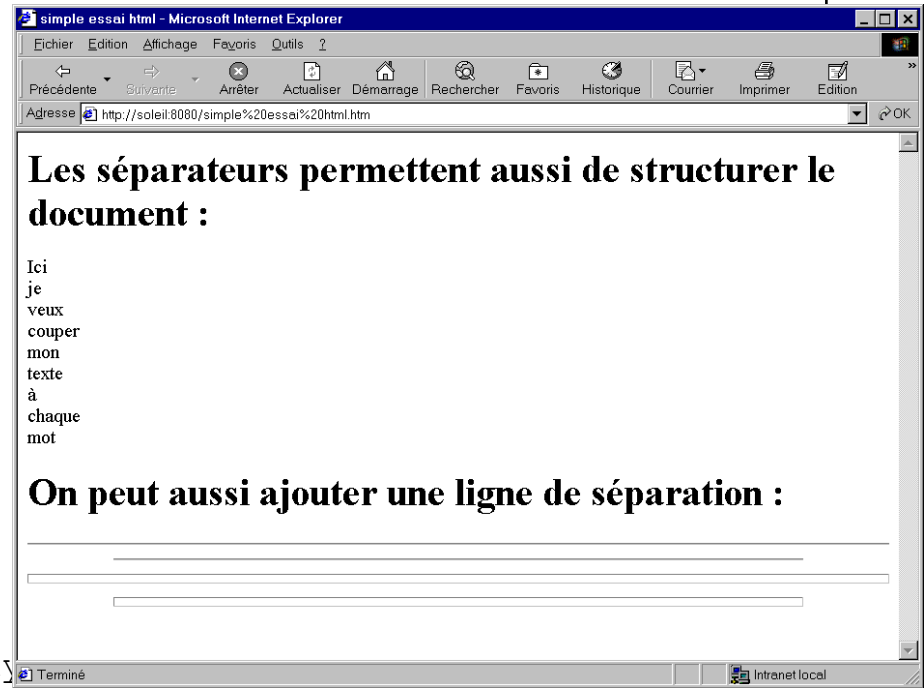
# Structuration du contenu HTML au sein du body

- ☞ `<h1>...</h1>` niveau de titre (entête) : en fait, 6 niveaux différents (h1..h6)
- ☞ `<p>` paragraphe `</p>`      délimiteur de paragraphe  
pour formater chaque paragraphe en modifiant l'alignement  
`<p align="left">... </p>` (*idem avec "center", "right"*)
- ☞ `<hr />` insère une ligne de séparation
- ☞ Autres :
  - `<br />`                    coupure forcée d'une ligne (au sein d'un paragraphe)
  - `<blockquote>` mise en retrait du texte `</blockquote>`
  - `<center> ...texte... </center>`            centrage d'un texte html quelconque
- ☞ Il existe aussi des présentations prédéfinies :  
encadrés par les balises `code`, `definition`, ou `citation`

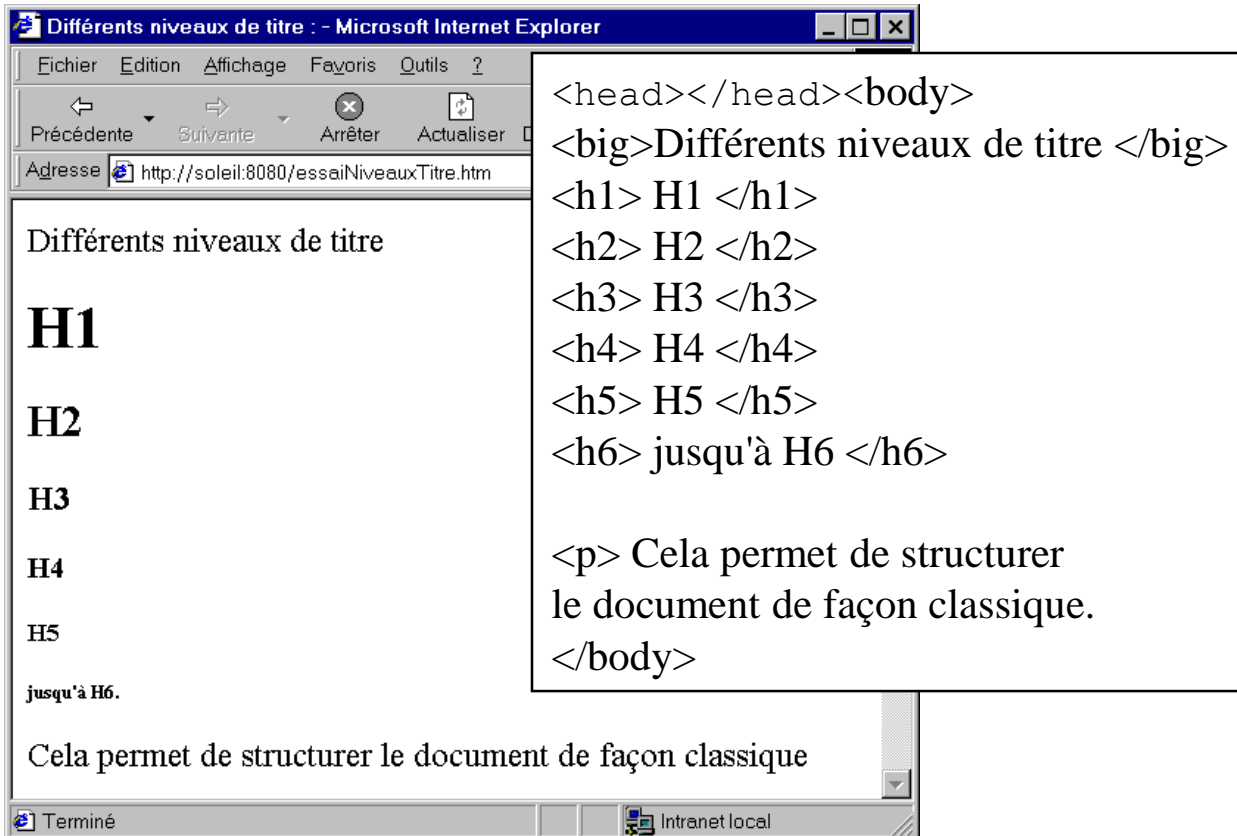
# Exemples de séparateurs

```
<html><head></head><body>
<h1>Les séparateurs permettent aussi de structurer le document :</h1>
Ici <br />
je <br />
veux <br />
couper <br />
mon <br />
texte <br />
à <br />
chaque <br />
mot <br />

<h1>On peut aussi ajouter
une ligne de séparation : </h1>
<hr />
<hr width="80%" />
<hr size="10" />
<hr width="80%" size="10" /></body>
```



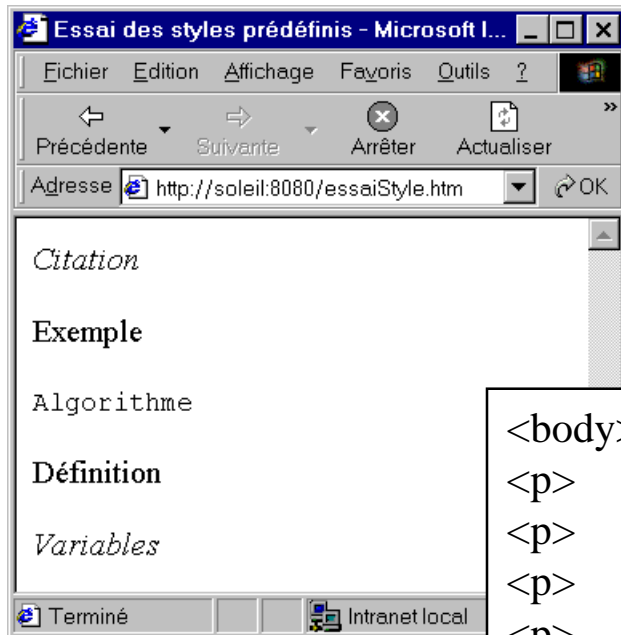
# Exemples de niveaux de titre



The screenshot shows a Microsoft Internet Explorer window titled "Différents niveaux de titre : - Microsoft Internet Explorer". The address bar contains "http://soleil:8080/essaiNiveauxTitre.htm". The page content displays several heading levels: H1, H2, H3, H4, H5, and H6. A text block follows, stating "Cela permet de structurer le document de façon classique". A code window is overlaid on the right side of the browser, showing the HTML code for the page content.

```
<head></head><body>  
<big>Différents niveaux de titre </big>  
<h1> H1 </h1>  
<h2> H2 </h2>  
<h3> H3 </h3>  
<h4> H4 </h4>  
<h5> H5 </h5>  
<h6> jusqu'à H6 </h6>  
  
<p> Cela permet de structurer  
le document de façon classique.  
</body>
```

# Exemples de styles prédéfinis



```
<body>
<p>    <cite>    Citation </cite> </p>
<p>    <sample> Exemple </sample> </p>
<p>    <code>    Algorithme </code> </p>
<p>    <definition> Définition </definition> </p>
<p>    <var> Variables </var> </p>
</body>
```

# Mise en valeur HTML d'un texte


Les plus courantes :

- ☞ `<b> ... gras... </b>`,
- `<i> ... italique... </i>`,
- `<u> ... souligné... </u>`
- `<pre> ...texte préformaté (à ne pas reformater)... </pre>`

Autres possibilités :

- ☞ `<big> ...texte... </big>` "grande" police
- `<small> ...texte... </small>` "petite" police
- ☞ `<font size="x">... </font>` taille de police x de 1 à 7 (7 est la plus grande)
- ☞ `<basefont size="x">` initialise la taille courante (par défaut `size="3"`)
  - `<font size="+x">... </font>` taille relative par rapport à la taille courante
  - `<font size="-x">... </font>`

# Les images



```

```

- ☞ src spécifie l'URL de l'image à charger
- ☞ alt sert d'explication pendant le chargement, ou pendant le survol de l'image.

```
      Insère l'image maphoto.gif
```

## Autres attributs optionnels

- ☞ align = "top", "middle", "bottom", "left" ou "right"  
aligne l'image par rapport à la ligne
- ☞ height
- ☞ width    hauteur, largeur de l'image

# Les liens et les ancres HTML

## Forme générale

```
<a href="url" title="un titre">  
    texte html cliquable  
</a>
```

## Cas d'url munie d'une ancre

```
<a href="xxx#ancre" title="un titre"> texte html cliquable  
</a>
```

- *xxx* est une url:    lien vers un document d'url *xxx*
- *xxx* absent        lien vers le même document
- *#ancre*            lien vers une ancre dans le document courant
- *"#" seul*         lien sans changement visuel

Spécification des ancres (*autre termes : bookmark ou signet du document*)

```
<a name="ancre"> texte </a>
```

# Exemple d'image cliquable (et autre texte HTML cliquable)

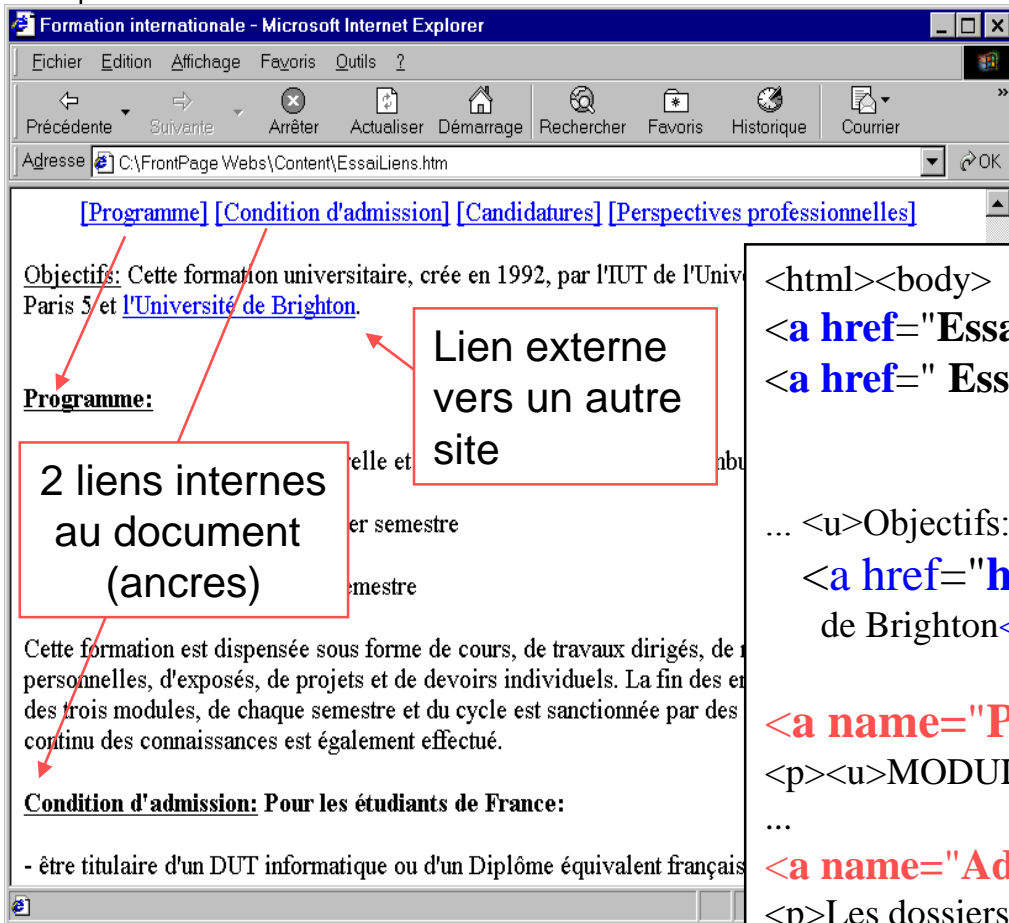
```
<html> <body>
<a href="general/factualites.htm">
  
</a><br /><br />
<a href="mailto:bedos@iut.univ-paris5.fr">
  
</a>
</body> </html>
```

Note :

*le second lien lance le gestionnaire de mail*



## Exemple de liens et d'ancres



EssaiLiens.htm

```
<html><body>
<a href="EssaiLiens.htm#Progr"> [Programme]</a>
<a href="EssaiLiens.htm#Adm"> [Condition
d'Admission]</a>

... <u>Objectifs:</u> Cette formation universitaire, ...
<a href="http://www.bton.ac.uk/">l'Université
de Brighton</a>.

<a name="Progr"> Programme </a> <br>
<p><u>MODULE1:</u><br>
...
<a name="Adm"> Condition d'admission </a> <br>
<p>Les dossiers de candidatures seront disponibles,
au département informatique </p>
... </body></html>
```

# Structuration sous forme de listes

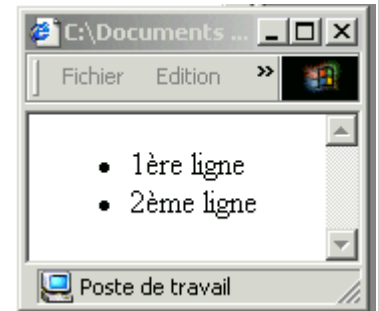
☞ liste non ordonnée: `<ul> ...</ul>`

☞ liste ordonnée: `<ol> ... </ol>`

☞ Élément de liste `<li> item </li>`

☞ Attribut optionnel pour les listes

- (ul) type= "disc", "square" ou "circle"
- (ol) type= "1" (décimal),  
"a" (minuscule), "A"(majuscule),  
"i" ou "I" (romain)



```
<ul>
  <li>1ère ligne </li>
  <li>2ème ligne </li>
</ul>
```

# Structuration sous forme de tableaux

Décrire  
pour chaque ligne  
(`<tr></tr>`)

la série des cellules :

- `<td>` cellule standard`</td>`
- `<th>` cellule-entête  
(surlignée)`</th>`

```
<table>
  <tr>
    <th>contenu de la cellule 1-1</th>
    <td>contenu de la cellule 1-2</td>
  </tr>
  <tr>
    <th>contenu de la cellule 2-1</th>
    <td>contenu de la cellule 2-2</td>
  </tr>
</table>
```

*Ligne 1* |

*Ligne 2* |

# Structuration sous forme de tableaux (...)

## Autres attributs des tableaux

- ☞ `border` définit l'épaisseur du cadre
- ☞ `width` largeur du tableau (pixel ou %)

## Balise optionnelle des tableaux

`<table> <caption> ... La légende ... </caption> </table>`  
pour décrire de façon succincte ce que le tableau contient.

## Autres attributs des cellules:

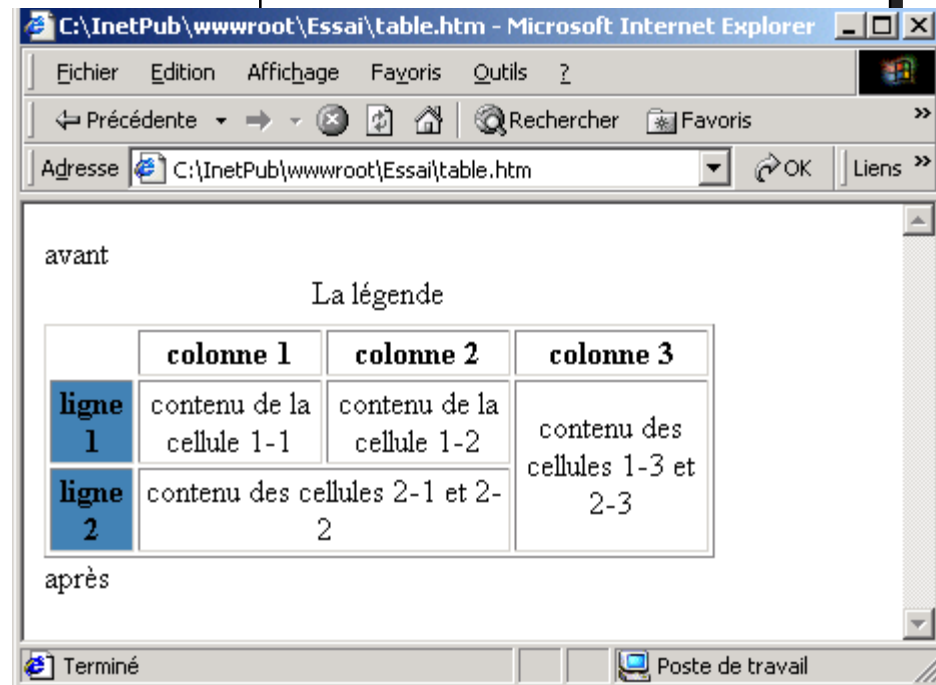
- ☞ `<th>` *contenu cellule* `<th>` *cellule d'entête (style surligné).*
- ☞ `colspan, rowspan:` *débordement d'une cellule sur la colonne ou la ligne suivante*
- ☞ `cellspacing, cellpadding` *détermine l'espace entre le texte et le bord*
- ☞ `bgcolor` *couleur du fond*

# Exemple de tableaux

```

<html><body>avant
<table border="1" width="80%" >
  <caption> La légende </caption>
  <tr>
    <th ></th>
    <th>colonne 1</th>
    <th>colonne 2</th>
    <th>colonne 3</th></tr>
  <tr align="center">
    <th bgcolor="steelblue">ligne 1</th>
    <td>contenu de la cellule 1-1</td>
    <td>contenu de la cellule 1-2</td>
    <td rowspan="2" align="center">
      contenu des cellules 1-3 et <br /> 2-3
    </td></tr>
  <tr align="center">
    <th bgcolor="steelblue">ligne 2</th>
    <td colspan="2">contenu des cellules 2-1 et 2-2</td></tr>
</table>
après</body></html>

```



# Division du document en blocs



balise **<div...> texte HTML du dloc </div>**

## Attribut courant :

☞ **id** = "*identifiant distinguant le bloc div*"

- Définit la page en blocs de spécifications HTML.
- Des styles pourront être appliqués pour :
  - décorer,
  - dimensionner et
  - positionner le bloc.
- L'imbrication des blocs est possible.

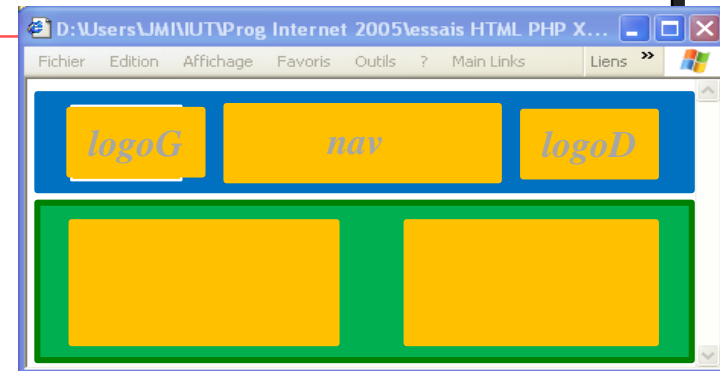
# Division du document en blocs

(voir cours sur les styles CSS)

```

<html> <head>
  <style>      #entete div, #presentation div { float:left; }
              #presentation { clear:left }
  </style>
</head>
<body>
  <div id="entete">
    <div id="logoG"></div> <!-- sous divisions de l'entete visible-->
    <div id="nav"></div>
    <div id="logoD"></div>
  </div> <!--fin entete-->
  <div id="presentation">
    <div></div>          <!-- sous divisions de la presentation-->
    <div></div>
  </div> <!--fin presentation-->
</body></html>

```



# Structuration des Formulaires en HTML

Zone HTML regroupant un ensemble de champs de contrôle  
Interprétable par un script ou un exécutable  
Réponse : une nouvelle page sur le navigateur client

```
<form action="url" target="nom_fen">  
... le formulaire... </form>
```

- ➡ Plusieurs fomulaires "form" possibles, entre <body> et </body>
- ➡ **action** : attribut d'url du script pouvant traiter le formulaire
- ➡ **target** : nom de la fenêtre pour afficher la réponse  
(défaut : fenêtre courante)

# Exemple de formulaire

Boutons prédéfinis au sein d'un formulaire

- ☞ bouton **submit** dans le formulaire
- ☞ bouton **reset** : réinitialise localement le formulaire

CGI4VB Mailing List Subscription Form - Micros...

Echier Edition Affichage Favoris Outils ?

← Précédente → Suivante × Arrêter Actualiser

Adresse C:\inetpub\wwwroot\cgi4v...

## Liste d'insc

Saisir votre "nom" et votre "e

Nom:

Email:

Go Reset

Terminé Poste de travail

```
<form action="inscription.php"> saisir ...  
Nom: <input type="text" name="Nom" maxlength="20" size="20"/>  
Email:  
  <input type="text" name="Adresse" maxlength="29" size="20"/>  
<input type="submit" value="Go"/>  
<input type="reset" value="Reset"/>  
</form>
```

# Champs de contrôle (1)

## Zones de saisies et boutons élémentaires

### balise `<input ...>`

#### Les attributs :

☞ **type** = text | password | // saisie classique ou mot de passe,  
checkbox | radio | // case à cocher, bouton radio,  
submit | image // bouton de soumission du formulaire  
(pour image, préciser l'url → src="url"),  
reset | // réinitialisation des champs  
file // sélection de fichier

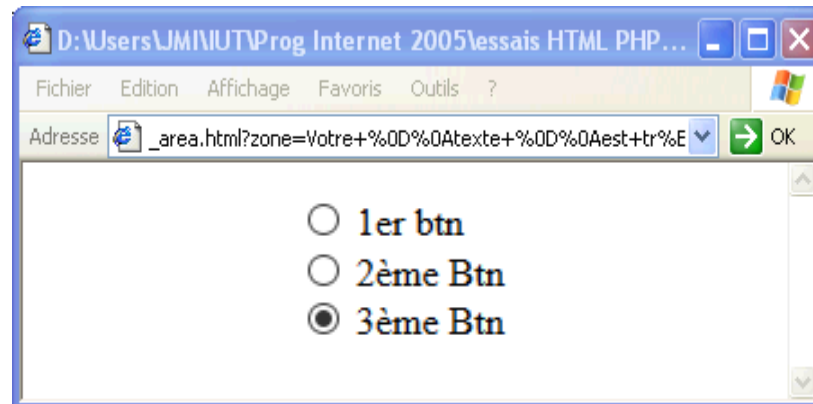
#### ☞ Autres attributs :

name = // nom interne  
value = // valeur  
size = // taille visible  
Maxlength = // taille maximale allouée  
checked="checked" // coche initiale pour les boutons radios et case à cocher  
disabled="disabled" // grisé de zone (non sélectionnable)

Les boutons d'un même groupe ont la même valeur pour name.

# Spécificités des boutons

☞ les boutons d'un même groupe ont la même valeur pour name



# Champs de contrôle HTML (2)

## Liste déroulante select et ses option(s)

**<select>** ...les options... **</select>**

- ☞ name = *nom interne*
- ☞ size = *nombre d'options visibles*
- ☞ multiple *1 seule ou plusieurs options sélectionnables*

**<option>** texte de l'élément

- ☞ texte : contenu html visible
- ☞ selected : *option sélectionnée par défaut*
- ☞ value = *valeur interne non visible sert d'index (dans les programmes) Peut-être un identifiant, une URL...*

```
<select>
  <option > texte </option >
  <option selected="selected">
    texte2 </option >
</select>
```

(touche contrôle :  
pour effectuer  
une sélection multiple)

```
<select multiple="multiple">
  <option selected="selected"> texte </option>
  <option > texte2 </option>
  <option selected="selected">texte3 </option>
</select>
```

# Champs de contrôle (3)

## Zones de saisie multi-lignes

### Balise `<textarea>`

#### Les attributs :

- ☞ `name` : nom interne
- ☞ `rows` : nb de lignes
- ☞ `cols` : nb de colonnes
- ☞ `maxlength` : nb de lignes affichées (ascenseur)

```
<textarea rows="2" cols="80" >  
    texte html  
    multiligne  
</textarea>
```

#### Autres attributs de `textarea`

- ☞ `wrap`: détermine la césure des mots en fin de ligne
- ☞ `readonly="readonly"` lecture seule
- ☞ `disabled="disabled"` désactivé

# Spécificités des zones de texte multilignes (textarea)

☞ attribut wrap = hard | soft | off

```
<html><body><form>
<textarea  name="zone"
           wrap="hard" cols="10" >
  votre texte est très long
</textarea>
<input type= "submit">
</form></body></html>
```



- 'off' pas de retour automatique à la ligne (ascenseur horizontal)
- 'soft' retours automatiques mais non-transmis au serveur :  
     **zone**=Votre+texte+est+tres+long
- 'hard' retours automatiques transmis au serveur :  
     **zone**=Votre+%0D%0Atexte+%0D%0Aest+tres+%0D%0Along

retour  
CR LF

# Exemples de champs de contrôle

**Nom** `<br /><input name="ctrl1" type="text" value="Votre nom" /><br />`

**Mot de passe** `<br /><input type="password" name="pass" value="xxxx" /> <br />`

**Couleur** `<br />`

`<input type="radio" name="coul" value="0" /> Vert`

`<input type="radio" name="coul" value="1" checked="checked" /> Bleu`

`<input type="hidden" name="coul" value="pas vu" />`

`<input name="foncé" type="checkbox" checked="checked" />foncé<br />`

`<p>Age <select name="age"> <option value="0"> 17 </option>  
<option value="a1" selected="selected"> 18 </option>  
<option value="a2"> 19 </option> </select><br />`

**Commentaire**`<br />`

`<textarea name="comment" rows="2" size="18" maxlength="250">  
c'est tout ! </textarea>`

Mozilla Fir... [min] [max] [fermer]

Fichier Édition Affichage Historique

...

Nom  
Votre nom

Mot de passe  
xxxx

Couleur  
 Vert
  Bleu
  foncé

Age 18 ▼

Commentaire  
c'est tout !

# Améliorer l'ergonomie d'un formulaire

## Fixer l'ordre de saisie des champs (input, select, ...) du formulaire

`<input tabindex="un numéro" ... />`

- >> attribuer un numéro différent par champs
- >> les différents *numéros* fixent l'ordre du "focus" sur les champs, sous l'action d'une série de tabulations.

## Dessiner un cadre décorable autour d'un groupe de champs de contrôle

`<fieldset>` *groupe de champs de contrôle* `</fieldset>`

Un fieldset peut être grisé : `disabled=disabled`

## Associer un label à un champs de contrôle, appelé ici "*nom*"

`<label for="nom">` **le texte du label** `</label>`

- >> un clic sur le label donne le focus sur le champs
- >> les séquences de tabulations ne prennent pas en compte les labels

# Améliorer la structuration d'un formulaire

## Encadrement de zones (fieldset) et labels des champs (label)

The screenshot shows a web browser window with the following content:

```

D:\Users\UMINUT\Prog Internet 2005\essais HTML PHP X...
Fichier  Edition  Affichage  Favoris  Outils  ?  Main Links  Liens >>
Vous êtes :
Votre nom : [input type="text" value=""]
Adresse : [input type="text" value=""]
[s'identifier] [annuler]
  
```

```
<html><body><form>
```

```
<fieldset align="center">
```

```
<legend>Vous êtes :</legend>
```

```
<br/>
```

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

```
<input name="Nom" type="text" maxlength="20" size="20" tabindex="1" />
```

```
<br/>
```

```
<label for="Adresse">Adresse :</label>
```

```
<input name="Adresse" type="text" maxlength="29" size="20" tabindex="2"/>
```

```
<input type="submit" value="s'identifier" />
```

```
<input type="reset" value="annuler"/>
```

```
</fieldset>
```

```
</form> </body></html>
```

*note :*

*alignement*

*complet*

*à faire en CSS*

# Alignement CSS du formulaire

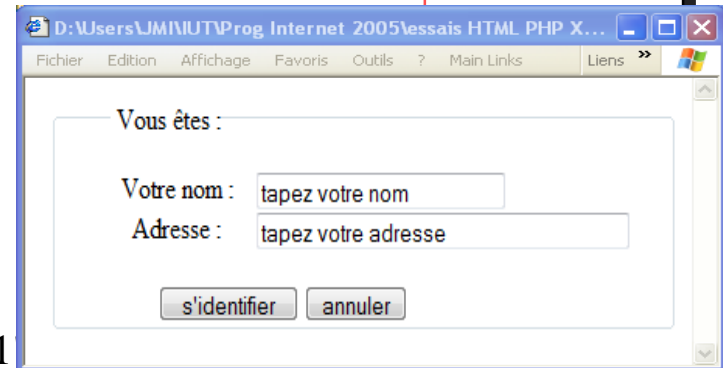
(voir cours sur les styles CSS)

```
<html><head><style>
  form{margin:auto 10% }
  fieldset {padding-left:30; text-align:left; width: 50em}
  label {clear:left; float:left; width:12em; text-align:center}
  input {display:block; height:1em}
    input[type=Nom] { width:12em; }
    input[name=Adresse] { width:22em; }
```

```
</style></head> <body><form>
<fieldset >   <legend>Vous êtes :</legend>
  <label for="Nom" > Votre nom : </label>
  <input name="Nom"      maxlength="20" tabindex="1"

  <label for="Adresse">Adresse :</label>
  <input name="Adresse" maxlength="30"  tabindex="2" >

  <input type="submit" value="s'identifier" />
  <input type="reset"  value="annuler" />
</fieldset>
</form> </body></html>
```



# Attributs html5 et les formulaires nouveaux types de saisie



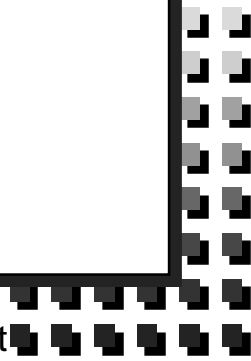
```
<input ... type="....." />
```

**date**

**number**

**url**

**tel**



# Attributs html5 et les formulaires

## directives de saisie

### pattern

#### Vérification d'un champs de saisie

```
<input ... pattern=" [A-F][0-9]{5}" />
```

Format donné par une  
expression régulière

### placeholder

#### Sollicitation de saisie

```
<input ... placeholder="infoSaisie" />
```

### required

#### Indication des saisie obligatoire

```
<input ... required="required" />
```

# Construire un modèle de chaîne

☞ Indication de positionnement :

**^**m      *modèle commençant par ch*

**\$**m      *modèle finissant par ch*

**^m\$**    *modèle ch exactement*

*ch1 | ch2* *choix entre 2 modèles*

☞ Classes pour un (type) de caractère

**.**                      // décrit un caractère quelconque

[a-zA-Z\_.\-] | [0-9] // un caractère parmi les lettres et {'\_', '.', '-'} **ou** parmi les chiffres.

[**^**a;\|]                // **^** pour la négation : un caractère sauf 'a' ';' '\' et '/'

\d et \D                // équiv à [0-9] et et l'opposé d'un chiffre

\w et \W                // équiv à [a-zA-Z0-9\_] et l'opposé

\s et \S                //(espace, retour, tabulation) et l'opposé.

☞ Précision d'une multiplicité

**.**{nb}                  //il faut une succession de 'nb' caractères quelconques

\.<sup>{inf,sup}</sup> // une série de (vrai) '.' en quantité au moins 'min' et au

[0-9]{inf,} //au moins 'inf' chiffres

Modèle de chaîne :

- succession de caractères et classes de caractères
- multiplicité de répétition
- groupement de sous-série entre parenthèses

**^**\-?[0-9]+\$

**^**(.+ )@(.+ )\.(.+ )\$

**abréviations : ?**={0,1}, **+**={1,} et **\***={0,}