

# Technologie Web

## *Chapitre 3 : HTML et CSS*

**Cecilia Zanni-Merk**

cecilia.zanni-merk@insa-rouen.fr

Bureau BO B R1 04

# Contenu

- HTML
- CSS

# Références

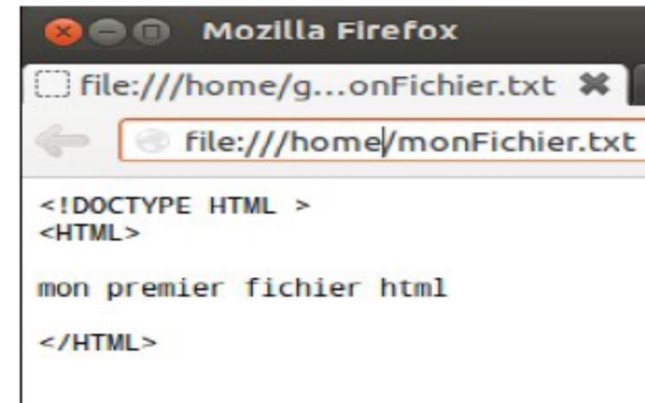
- Cours « Technologies Web » d'Alexandre Pauchet, INSA Rouen Normandie, 2016
- MOOC « Soyez Acteurs du Web », <https://www.fun-mooc.fr/>

# HTML

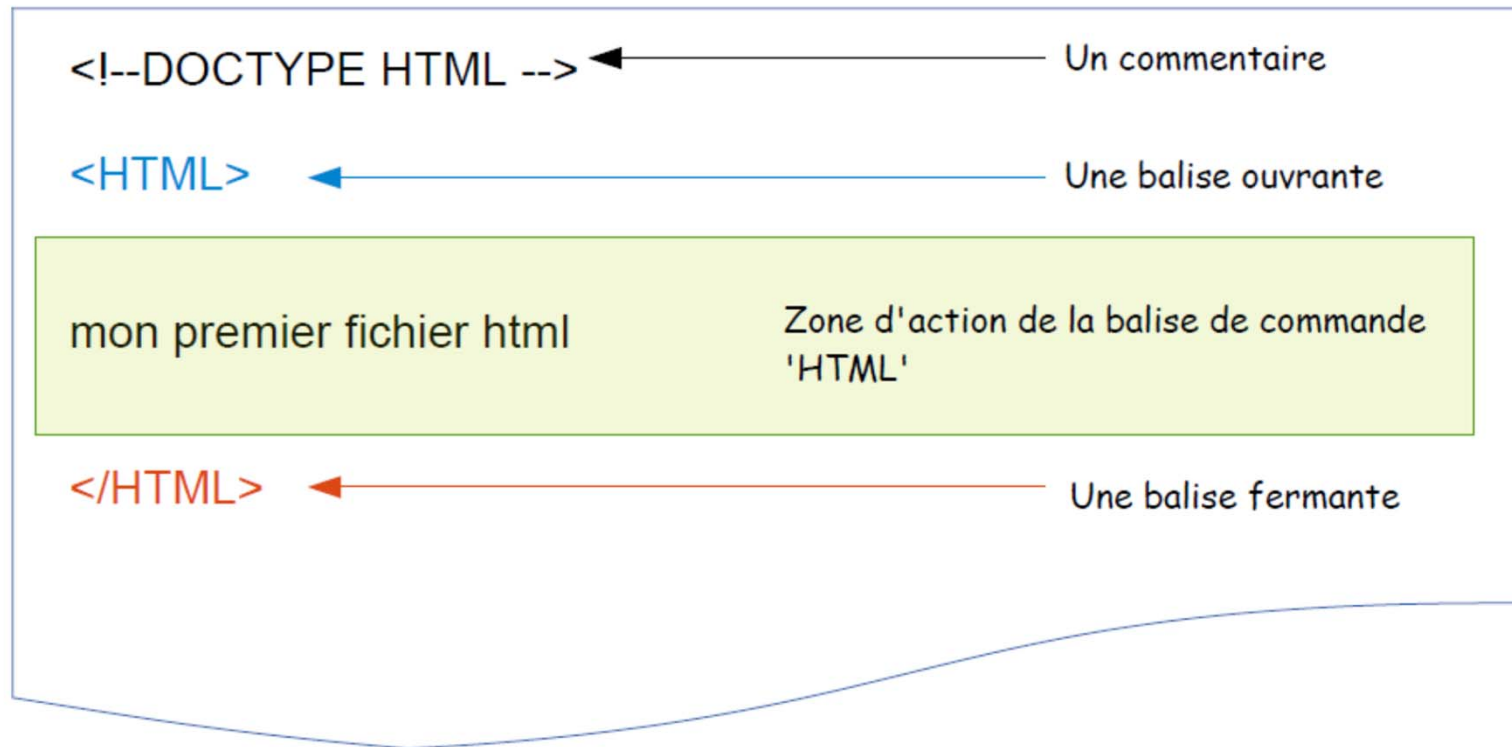
- HTML signifie Hypertext Markup Language, c'est un format conçu pour réaliser des pages web.
- Il permet de structurer sémantiquement une page, de formater le contenu, ajouter des images etc.
- Le HTML est un langage à « balises »
- Plusieurs versions se sont succédées, aujourd'hui : HTML 5

# Mon premier fichier HTML

```
<!--DOCTYPE HTML -->  
<HTML>  
  
mon premier fichier html  
  
</HTML>
```



# Balises



# Balises

- Trois types d'éléments
- **Élément bloc (div)** : élément précédé et suivi d'un saut de ligne. Il forme une boîte dans lequel est inclus du texte ou d'autres éléments.
  - *Exemples* : les paragraphes, les tableaux, ...
- **Élément inline (span)** : élément qui s'insère dans le fil du texte et ne peut contenir que du texte ou d'autres éléments inlines.
  - *Exemples* : les éléments typographiques italique, gras, ...
- **Élément auto-fermant** : élément qui est une balise ouvrante et fermante à la fois. Elle n'a donc pas de contenu. Ce sont soit des balises de type bloc, soit de type inline.
  - *Syntaxe* : `<balise>` ou `<balise/>`
  - *Exemples* : saut de ligne, séparation horizontale ...

# Balises

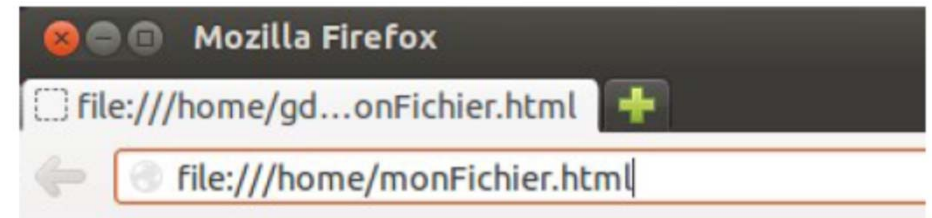
- Balise de début : `<commande>`
- Balise de fin : `</commande>`
- Balise auto-fermante : `<commande />`
- Commentaires : `<!--Ceci est un commentaire-->`
- Remarque : retours chariot, succession d'espaces et/ou de tabulations ne sont pas pris en compte.
- **Principe** : Les balises permettent d'appeler des commandes dont l'action est limitée au texte contenu entre la balise de début et la balise de fin.



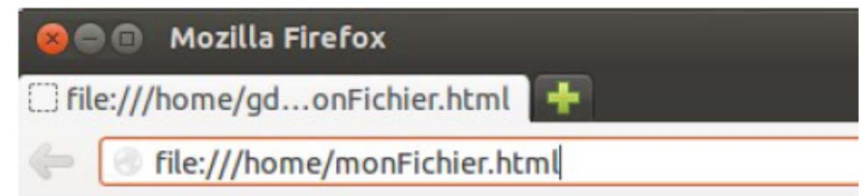
# Balises

```
<!--DOCTYPE HTML -->  
<HTML>  
<em>mon premier fichier html</em> En  
dehors de la balise !  
</HTML>
```

```
<!--DOCTYPE HTML -->  
<HTML>  
<strong>mon premier fichier  
html</strong> En dehors de la balise  
!  
</HTML>
```



*mon premier fichier html* En dehors de la balise !



**mon premier fichier html** En dehors de la balise !

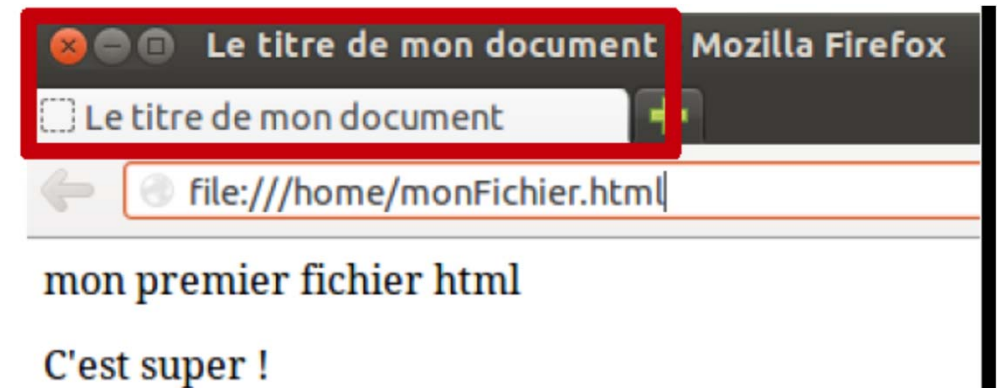
# Balises head & body

```
<!--DOCTYPE HTML -->
<HTML>
<head>
    <title> Le titre de mon document </title>
</head>
<body>
    <p>mon premier fichier html</p>
    <p>C'est super !</p>
</body>
</HTML>
```

Paragraphe

L'attribut align spécifie la justification

L'élément incline <br /> permet de contrôler les sauts de ligne



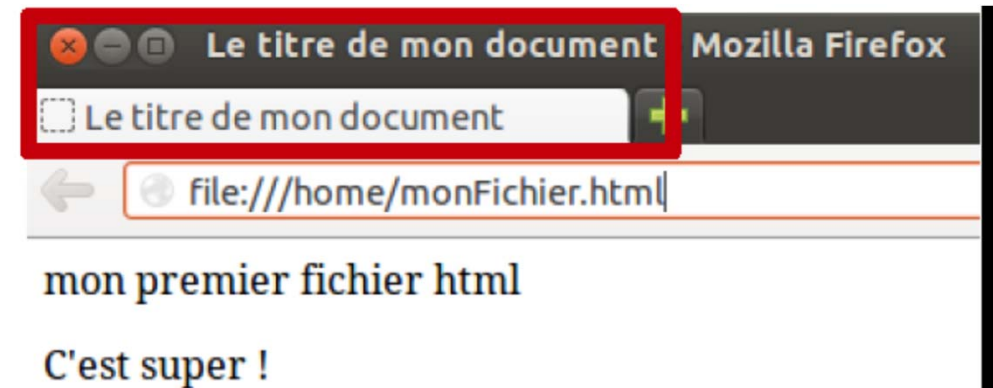
# Balises head & body

```
<!--DOCTYPE HTML -->
<HTML>
<head>
  <title> Le titre de mon document </t
</head>
<body>
  <p>mon premier fichier html</p>
  <p>C'est super !</p>
</body>
</HTML>
```

Paragraphe

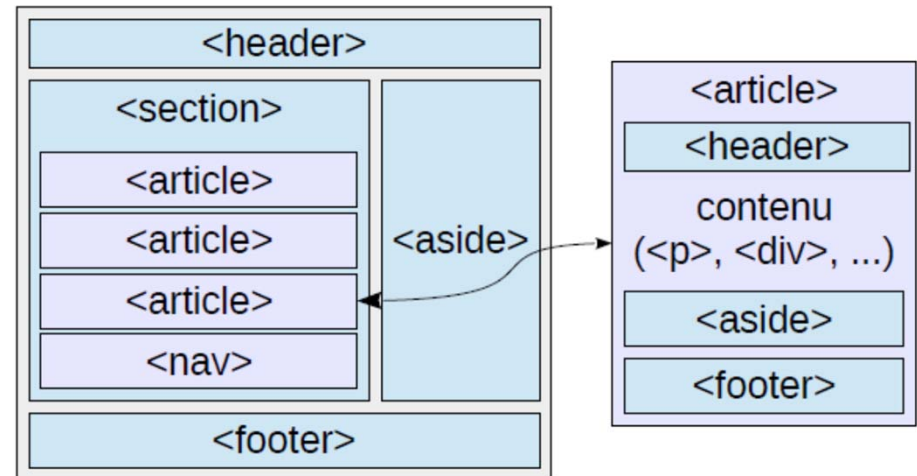
L'attribut align spécifie la justification

L'élément incline <br /> permet de contrôler les sauts de ligne



# Mise en forme

- `<br />` : génère un saut de ligne
- Structuration d'un document : `<section>`, `<article>`, `<header>`, `<footer>`, `<aside>`, `<nav>`
- 6 niveaux de titre
  - `<h1> ... </h1>` jusqu'à  
`<h6> ... </h6>`

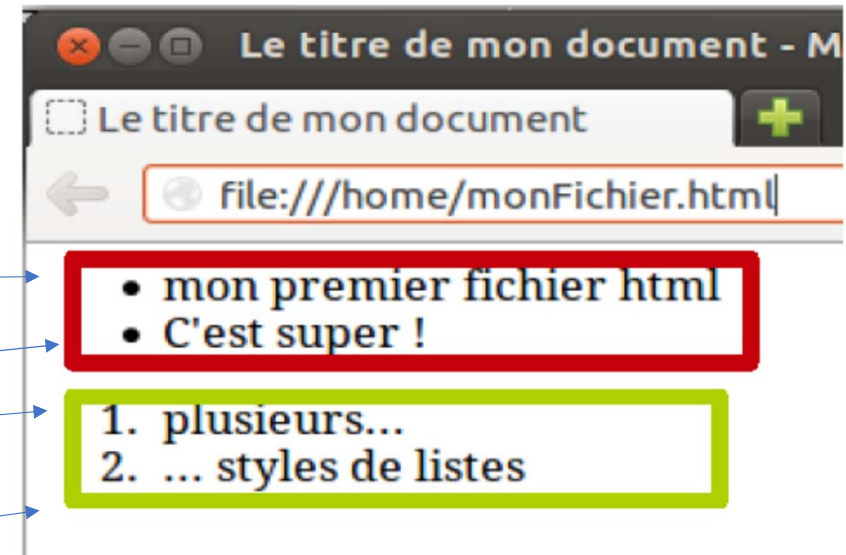


# Notion de flux

- Les balises sont lues séquentiellement, ...
- ... sont affichées au fur et à mesure par le navigateur, ...
- ... les unes en dessous des autres
- Remarque
  - L'affichage se modifie au fur et à mesure du chargement de la page et de ses composants (texte, images, etc.)

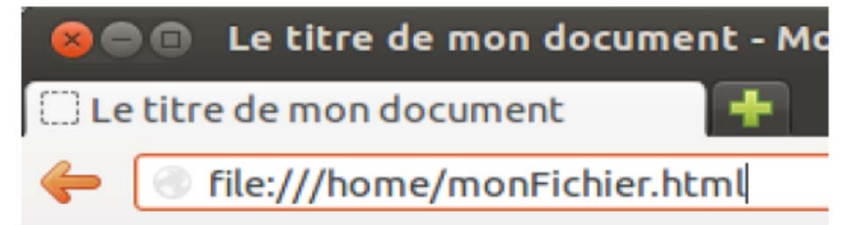
# Listes

```
<!--DOCTYPE HTML -->
<HTML>
<head>
  <title> Le titre de mon document </title>
</head>
<body>
  <ul>
    <li>mon premier fichier html</li>
    <li>C'est super !</li>
  </ul>
  <ol>
    <li>plusieurs...</li>
    <li>... styles de listes</li>
  </ol>
</body>
</HTML>
```



# Tableaux

```
<!--DOCTYPE HTML -->
<HTML><head><title> Le titre de mon document </title></head>
<body>
  <table>
    <tr> <!-- Titres colonnes -->
      <th>Colonne 1</th>
      <th>Colonne 2</th>
    </tr>
    <tr>
      <td>D11</td> <!-- Données tableau -->
      <td>D12</td>
    </tr>
    <tr>
      <td>D21</td>
      <td>D22</td>
    </tr>
    <tr>
      <td></td>
      <td>D32</td>
    </tr>
  </table>
</body>
</HTML>
```



| Colonne 1 | Colonne 2 |     |
|-----------|-----------|-----|
| D11       | D12       | ← 1 |
| D21       | D22       | ← 2 |
|           | D32       | ← 3 |
|           |           | ← 4 |

# Problèmes d'encodage

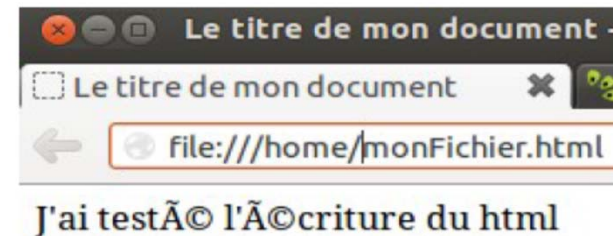
- Encodage : Le jeu de caractères utilisé pour enregistrer un fichier
- Les plus courants : l'ISO-8859-1, l'ISO-8859-15, UTF-8

```
<HTML>

    <head>
        <title> Le titre de mon
document </title>
    </head>

    <body>
        <p>J'ai testé l'écriture du
html </p>
    </body>
</HTML>
```

MonFichier.html



L'encodage par défaut n'est pas  
forcement le plus adapté !



# Problèmes d'encodage

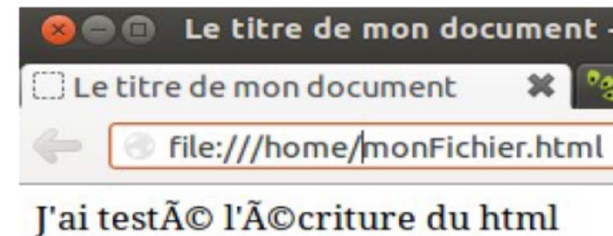
- Encodage : Le jeu de caractères utilisé pour enregistrer un fichier
- Les plus courants : l'ISO-8859-1, l'ISO-8859-15, UTF-8
- UTF-8 est le plus polyvalent
- Ajouter  
`<meta charset="UTF-8" />`  
après dans la partie `<head>` de la page.

```
<HTML>

    <head>
        <title> Le titre de mon
document </title>
    </head>

    <body>
        <p>J'ai testé l'écriture du
html </p>
    </body>
</HTML>
```

MonFichier.html



L'encodage par défaut n'est pas  
forcement le plus adapté !

# Images (éléments inline)

- La balise `<img />` permet d'insérer une image
- Les attributs suivants sont obligatoires :
  - `src` : l'URI où se situe l'image
  - `alt` : courte description de l'image
- Exemple ``
- Remarque : en spécifiant la taille des images, on accélère le chargement (attributs `width` et `height`).

# Liens (éléments inline)

- L'élément `<a href="...">...</a>` permet d'insérer un lien
- Le contenu de l'élément est celui qui sera affiché en tant que lien.
- L'attribut `href` contient l'URI vers laquelle le lien pointe.
  - URL : `http://www.google.com`
  - URL(mail) : `mailto:alexandre.pauchet@insa-rouen.fr`
  - Fichier local avec chemin relatif : `./dossier/autre_page.html`
  - Fichier local avec chemin absolu : `/www/dossier/autre_page.html`
- Exemple  
`<a href="lienversuneautrepage.html">Texte affiché</a>`

# Insertion d'un objet quelconque

- A partir du même principe que les images insérons un objet quelconque : du pdf , un fichier audio, ...
- Pour « visualiser » cette information il faut une application capable de prendre en charge cette nouvelle donnée : un « visualisateur pdf », un « visualisateur » de vidéo, audio, ...
- Principe : similaire aux images
  - Un tag : `object`
  - Une source
  - Un texte alternatif
  - L'application qui permet de gérer l'objet : `type="audio/x-wav"`, `type="application/pdf"`, ...

# Insertion d'un objet quelconque

- A partir du même principe que les images insérons un objet quelconque : du pdf , un fichier audio, ...

- Pour prendre « visu
- Princi
  - Un
  - Un
  - Un
  - L'a typ

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html> <head>
<title>Affichage d'un fichier pdf</title>
</head>
<body>
<h1>Visualisation du pdf</h1>
<object data="visualisation.pdf"
        type="application/pdf"
        width="500" height="500">
  alt : <a href="visualisationt.pdf non disponible">
    visualisation.pdf</a>
</object>
<hr>
<address></address>
<!-- hhmts start --> <!-- hhmts end -->
</body> </html>
```

Un objet à traiter

Une application

# Insertion d'un objet quelconque

- A partir du même principe que les images insérons un objet quelconque : du pdf , un fichier audio, ...

- Pour « prendre » visuellement

- Principi

- Un
- Un
- Un
- L'a typ

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html> <head>
<title>Affichage d'un fichier pdf</title>
</head>
<body>
<h1>Visualisation du pdf</h1>
<object data="visualisation.pdf"
        type="application/pdf"
        width="500" height="500">
  alt : <a href="visualisationt.pdf non disponible">
    visualisation.pdf</a>
</object>
<hr>
<address></address>
<!-- hhmts start --> <!-- hhmts end -->
</body> </html>
```

Un objet à traiter

Une application

L'application est un outil  
intégrée au navigateur qui  
traite l'objet (ii un PDF)

# Gestion des cartes / plans

- Souvent nous souhaitons faire apparaître un plan ou une carte pour aider au repérage, donner une spatialisation d'un processus.
- De nombreux serveurs proposent des cartes/images (Géoportail, Google maps, Mappy, Open Street Map, ...)
- On souhaite avoir une version à jour des informations et donc ne pas stocker en local l'information
- Principe
  - Demander un service à un serveur distant
  - Afficher la carte/plan/image comme une image ou comme un lien externe

# Plus sur les tableaux ... Structuration

```
[...] <body><table>
  <thead> <!-- En-tête du tableau -->
  <tr>
    <th>Col 1 - entête</th>
    <th>Col 2 - entête </th>
  </tr>
</thead>
<tfoot> <!-- Pied de tableau -->
<tr>
  <th>Col 1 - pied</th>
  <th>Col 2 - pied</th>
</tr>
</tfoot>
<tbody> <!-- Corps du tableau -->
<tr>
  <td>Valeur 11</td>
  <td>Valeur 12</td>
</tr>
<tr>
  <td>Valeur 21</td>
  <td>Valeur 22</td>
</tr>
</tbody>
</table></body>
[...]
```

| Col 1 - entête | Col 2 - entête |
|----------------|----------------|
| Valeur 11      | Valeur 12      |
| Valeur 21      | Valeur 22      |
| Col 1 - pied   | Col 2 - pied   |



# Plus sur les tableaux ... Structuration

```
[...] <body><table>
  <caption>La légende du tableau</caption>
  <thead> <!-- En-tête du tableau -->
  <tr>
    <th>Col 1 - entête</th>
    <th>Col 2 - entête </th>
  </tr>
</thead>
<tfoot> <!-- Pied de tableau -->
<tr>
  <th>Col 1 - pied</th>
  <th>Col 2 - pied</th>
</tr>
</tfoot>
<tbody> <!-- Corps du tableau -->
<tr>
  <td>Valeur 11</td>
  <td>Valeur 12</td>
</tr>
<tr>
  <td>Valeur 21</td>
  <td>Valeur 22</td>
</tr>
</tbody>
</table></body>
[...]
```



La légende du tableau

| Col 1 - entête | Col 2 - entête |
|----------------|----------------|
| Valeur 11      | Valeur 12      |
| Valeur 21      | Valeur 22      |
| Col 1 - pied   | Col 2 - pied   |

# Fusion de cellules

- Dans la commande <td>, attributs
  - colspan : pour la fusion selon des colonnes

| Col 1           | Col 2     |
|-----------------|-----------|
| Valeur 11       | Valeur 12 |
| Valeur 21 et 22 |           |

- rowspan : pour la fusion selon des lignes

| Col 1     | Col 2           |
|-----------|-----------------|
| Valeur 11 | Valeur 12 et 22 |
| Valeur 21 |                 |

# Exemple fusion

```
[...] <table border="1">
  <thead> <!-- En-tête du tableau -->
  <tr>
    <th>Col 1 </th>
    <th>Col 2 </th>
    <th>Col 3 </th>
  </tr>
</thead>
<tbody> <!-- Corps du tableau -->
<tr>
  <td>Valeur 11</td>
  <td>Valeur 12</td>
  <td rowspan="2"> Valeur 13 et 23 </td>
</tr>
<tr>
  <td colspan="2">Valeur 21 et 22</td>
  <td rowspan="2"> Valeur 13 et 23 </td>
</tr>
<tr>
  <td colspan="3">Valeur 31 et 32 et 33</td>
</tr>
</tbody>
</table> [...]
```

| Col 1                 | Col 2     | Col 3           |
|-----------------------|-----------|-----------------|
| Valeur 11             | Valeur 12 | Valeur 13 et 23 |
| Valeur 21 et 22       |           | Valeur 13 et 23 |
| Valeur 31 et 32 et 33 |           |                 |

# Les formulaires

- Un formulaire permet la saisi d'informations de la part d'un visiteur, pour être traitée
- Deux aspects à considérer
  - Par quel moyen envoyer le texte saisi par le visiteur ?
  - Une fois les données envoyés, comment les traiter ? Qui ?

# Les formulaires

- L'élément `<form> . . . </form>` déclare un formulaire
- Les attributs :
  - `method` : spécifie la méthode d'acheminement des données (GET par défaut ou POST, qui est la méthode la plus utilisée) ← *premier aspect évoqué*
  - `action` : URL spécifiant le traitement des données (CGI, etc.) ← *deuxième aspect évoqué*

# Contenu d'un formulaire

- L'élément `<fieldset>...</fieldset>` permet de définir un regroupement dans un formulaire.
- L'élément `<legend>...</legend>` permet de donner une légende à un `fieldset`.
- L'élément `<label>...</label>` permet de définir une étiquette.
- L'élément `<input />` contient les attributs suivant :
  - `type` : spécifie le type d'élément à utiliser
  - `name` : donne un nom à l'élément
  - `value` : donne une valeur à l'élément

# Exemple de balises `<input />`

```
<input type="text" name="champs" size="10" value="texte" />
```

```
<input type="password" name="mdp" size="10" maxlength="8" />
```

```
<input type="hidden" name="steak" value="haché" />
```

```
<input type="checkbox" name="chk1" value="ok" />
```

```
<input type="checkbox" name="chk2" value="ok" checked="checked" />
```

```
<input type="radio" name="choix" value="rd1" />
```

```
<input type="radio" name="choix" value="rd2" checked="checked" />
```

```
<input type="radio" name="choix" value="rd3" />
```

- Remarques :

- pour les types checkbox/radio si l'attribut value n'est pas spécifié, la valeur par défaut est on
- Le type hidden permet de passer des valeurs d'une page à une autre

# Boutons

- Un élément `<input />` de type :
  - `submit` : affiche un bouton et permet l'envoi des données du formulaire au serveur
  - `image` : affiche une image et permet l'envoi des données du formulaire au serveur
  - `reset` : affiche un bouton et permet de restaurer les valeurs par défaut du formulaire
  - `file` : affiche un bouton permettant d'ouvrir une boîte de recherche de fichier

- Exemples

```
<input type="submit" name="action" value="Insert" />
```

```
<input type="file" name="unFichier" id="fichier" />
```

```
<input type="image" src="images/croix.jpg" name="action" value="Delete" />
```

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



# Champs texte

- Un élément `<textarea>` permet de créer un champs texte
- Exemple

```
<textarea rows="4" cols="50">
```

Ce texte est éditabile et sera envoyé lors du submit

```
</textarea>
```

# Listes

- La balise `<select>` permet de définir une liste
- Attributs :
  - `multiple` : permet de sélectionner plusieurs éléments de la liste
  - `code` : si `> 2` affiche un tableau, sinon un menu déroulant

- Exemple

```
<select name="laliste" size="3" multiple="multiple">
  <option value="1">toto</option>
  <option selected="selected" value="2">titi</option>
  <optgroup label="les autres">
    <option value="3">tata</option>
    <option value="4">tutu</option>
    <option value="5">tete</option>
  </optgroup>
</select>
```

# Noms des champs

- La balise `<label>` sert à nommer des champs
- Attribut `for` indique le champ décrit (attribut `id`)
- Utile sur les `radio` et `checkbox`
- Exemple

```
<form>
  <label for="h">Homme</label>
  <input type="radio" name="genre" id="h" />
  <br />
  <label for="f">Femme</label>
  <input type="radio" name="genre" id="f" />
</form>
```

# Exemple

```
[...] <form action=« POST">
  <fieldset>
    <legend>Exemple de formulaire</legend>
    <label>Nom :</label> <input type="text" name="monNom" id="nom" />
    <label>Prénom :</label> <input type="text" name="monPrenom" id="prenom" />
    <hr/>
    <input type="checkbox" name="maNewsletter" id="newsletter" /> <label for="newsletter">Une
checkbox</label>
    <input type="radio" name="monSexe" id="homme" /><label for="homme">Homme</label>
    <input type="radio" name="monSexe" id="femme" /><label for="femme">Femme</label><br/>
    <label for="photo">Fichier :</label> <input type="file" name="maPhoto" id="photo" /><br/>
    <select name="laliste" size="3" multiple="multiple">
      <option value="1">toto</option>
      <option selected="selected" value="2">titi</option>
      <optgroup label="les autres">
        <option value="3">tata</option>
        <option value="4">tutu</option>
        <option value="5">tete</option>
      </optgroup>
    </select>
    <hr/><textarea name="texte" rows="10" cols="80">Raconte-moi une histoire...</textarea><hr/>
    <input type="submit" name="maSoumission" id="soumission" />
    <input type="submit" name="action" value="Insert"/>
    <input type="submit" name="action" value="Update"/>
    <input type="image" src="images/ecureuil.png" alt="logoasi.png" name="action" width="75"/>
    <input type="reset" value="Reset"/>
  </fieldset>
</form> [...]
```

# Exemple

Exemple Formulaire

file:///C:/Users/chzm/enseignement INSA Rouen/2011

Rechercher

Les plus visités Débuter avec Firefox Hotel in CARDIFF - Bo... Speedtest.net by Ookl... Bienvenue chez Arpege CNRS - Direction Euro...

Exemple de formulaire


Nom :  Prénom :

☐ Une checkbox ☐ Homme ☐ Femme

Fichier :  Aucun fichier sélectionné.

toto  
titi  
les autres

Raconte-moi une histoire...



# Gestion des styles

# Gestion du style

- Couleur, formatage, positionnement ...
- Deux possibilités
  - Définition locale (de préférence à éviter)
  - Utilisation d'une feuille de style (CSS)



# Div et Span : Balise <div>

- Pour regrouper logiquement des parties de texte
- Applique une représentation (forme) et structure le document (fond)
- Permet de s'affranchir de la mise en page (automatique) du navigateur

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Gestion de la balise div</title>
</head>
<body>
  <h1>Ceci est un essai sans balise div</h1>

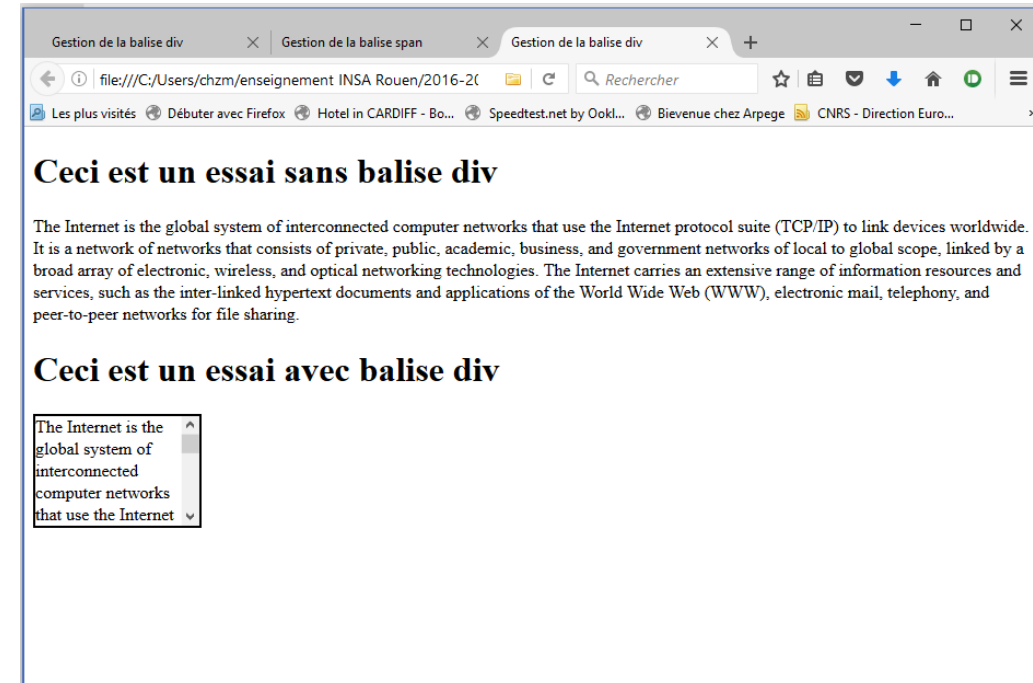
  The Internet is the global system of interconnected
  computer networks that use the Internet protocol suite (TCP/IP) to
  link devices worldwide. It is a network of networks that consists of
  private, public, academic, business, and government networks of
  local to global scope, linked by a broad array of electronic,
  wireless, and optical networking technologies. The Internet carries
  an extensive range of information resources and services, such as
  the inter-linked hypertext documents and applications of the World
  Wide Web (WWW), electronic mail, telephony, and peer-to-peer
  networks for file sharing.

  <h1>Ceci est un essai avec balise div</h1>
  <div style="width:150px; height:100px; overflow:auto;
border:solid 2px black;">
The Internet is the global system of interconnected
computer networks that use the Internet
  </div>

</body>

</html>

```



## Div et Span : Balise <span>

- Pour appliquer une représentation particulière sur une partie du texte
- On traite le texte à l'intérieur d'une organisation logique du document (un paragraphe, une liste ...)
- On peut changer son mode de représentation (font, couleur ...) via une série d'attributs

```
!DOCTYPE html>
<html>

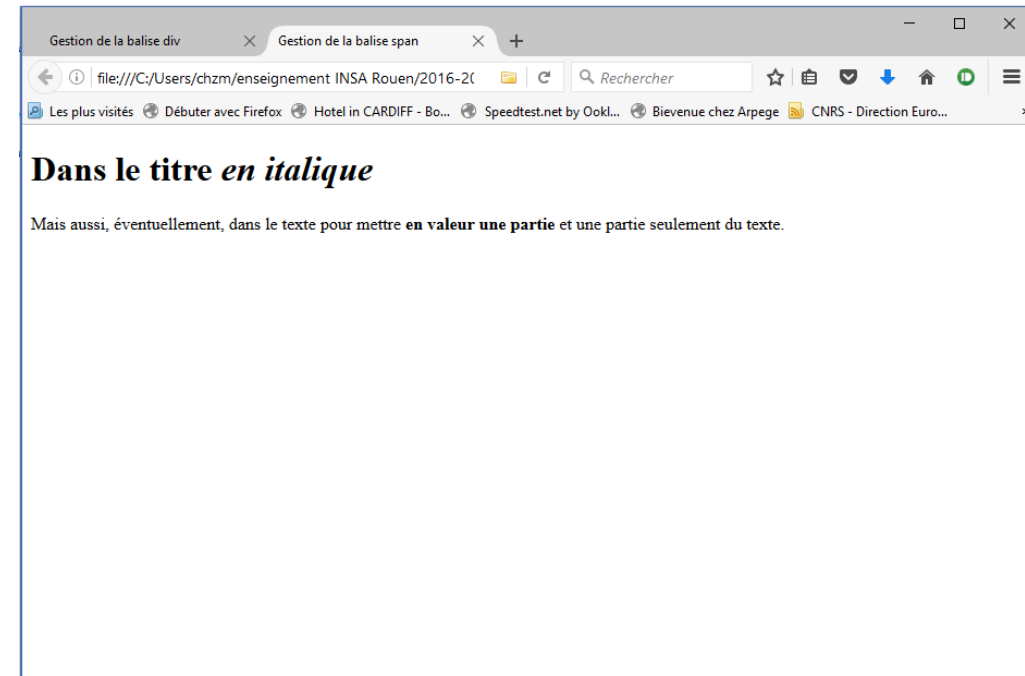
<head>
  <meta charset="utf-8" />
  <title>Gestion de la balise span</title>
</head>

<body>
  <h1>Dans le titre <span style="font-
style:italic"> en italique </span> </h1>

  Mais aussi, éventuellement, dans le texte
  pour mettre <span style="font-weight:bold;
color:red"> en valeur une partie </span> et
  une partie seulement du texte.

</body>

</html>
```



# Éléments manipulables en local

- Le texte standard

```
<body text=black> ... </body>
```

```
<body text="#AC1FBF"> ... </body>
```

- Le fond de la page

```
<body bgcolor=yellow > ... </body>
```

- Une partie spécifique du texte

```
<font color= red> ... </font>
```

```
<!DOCTYPE html>
<html>

<head>

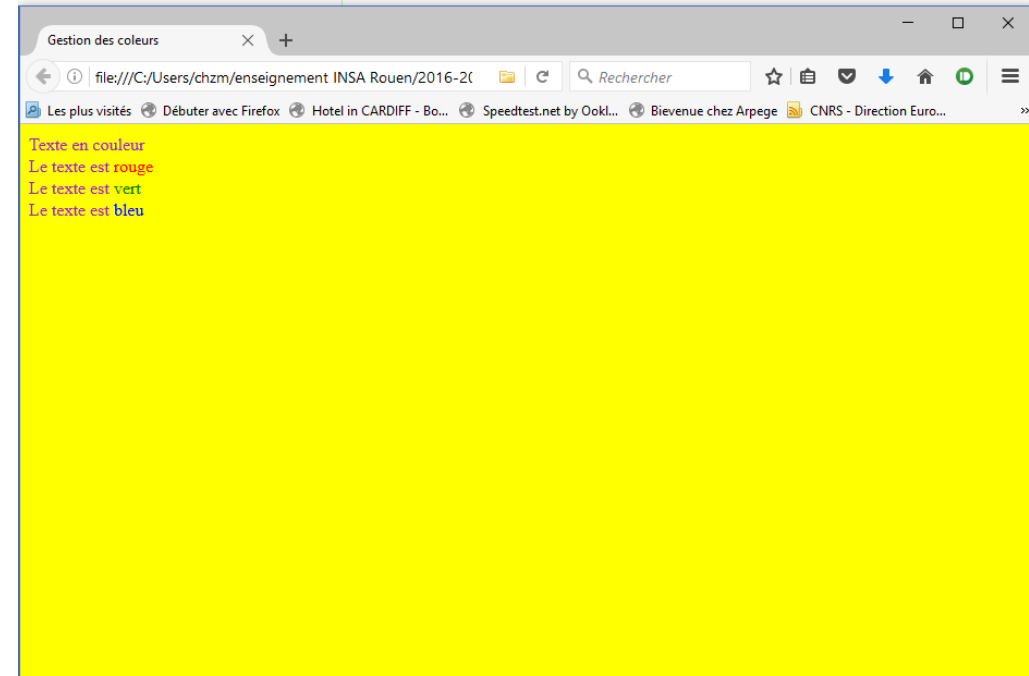
    <meta charset="utf-8" />
    <title>Gestion des couleurs</title>
</head>

<body text="AC1FBF" bgcolor=yellow>

    Texte en couleur <br/>
    Le texte est <font color=red>rouge</font>
<br/>
    Le texte est <font
color=green>vert</font> <br/>
    Le texte est <font color=blue>bleu</font>
<br/>

</body>

</html>
```



# Couleurs en HTML

- Chacune des composantes RVB est codée sur deux « chiffres » dans l'ordre Rouge, Vert, Bleu
- Exemples :
  - Noir : #00 00 00
  - Blanc : #FF FF FF
  - Rouge : #FF 00 00
- Sinon, il y a des noms prédéfinis pour certaines couleurs

# Couleurs en HTML

- Chacune des composantes R, G, B est codée sur 8 bits, dans l'ordre Rouge, Vert, Bleu
- Exemples :
  - Noir : #00 00 00
  - Blanc : #FF FF FF
  - Rouge : #FF 00 00
- Sinon, il y a des noms prédéfinis

### LOS COLORES SEGÚN:



## MUJERES



HOMBRES



PERROS



GEEKS

|               |          |       |         |
|---------------|----------|-------|---------|
| Marrasquino   | Rojo     | Gris  | #F94433 |
| Pimenton      | Rojo     | Negro | #AC203B |
| Granate       | Morado   | Negro | #85343D |
| Ciruela       | Morado   | Negro | #874994 |
| Berenjena     | Morado   | Negro | #663C84 |
| Uva           | Morado   | Negro | #8C2590 |
| Orquidea      | Morado   | Gris  | #A16799 |
| Lavanda       | Morado   | Gris  | #AF99C7 |
| Carnatino     | Rosado   | Gris  | #F38DA3 |
| Frutilla      | Rosado   | Negro | #D2157B |
| Chicle        | Rosado   | Gris  | #EC90B7 |
| Magenta       | Rosado   | Negro | #E90086 |
| Salmon        | Rosado   | Gris  | #F57D7E |
| Mandarina     | Naranja  | Gris  | #F27727 |
| Cantalupo     | Naranja  | Gris  | #FC9B7B |
| Platano       | Amarillo | Gris  | #F7D305 |
| Limon         | Amarillo | Gris  | #F1E311 |
| Miel          | Verde    | Gris  | #CCDF62 |
| Lima          | Verde    | Gris  | #68BD46 |
| Primavera     | Verde    | Gris  | #0AAE4F |
| Trebol        | Verde    | Gris  | #069665 |
| Helecho       | Verde    | Negro | #057054 |
| Musgo         | Verde    | Gris  | #3BA246 |
| Flora         | Verde    | Gris  | #ABCF37 |
| Espuma de mar | Azul     | Gris  | #68C3B2 |
| Rocio de mar  | Azul     | Gris  | #8BCCD0 |
| Real          | Azul     | Gris  | #0687A7 |
| Cielo         | Azul     | Gris  | #078DCA |
| Turquesa      | Azul     | Gris  | #0FB8B5 |



# CSS

- Cascading Style Sheets - Feuilles de style en cascade
- Un document en mode texte (fichier .css)
- Utilisable par des documents HTML ou XML
- Présentation identique de tous les documents HTML ou XML
- Apporte la modularité du formatage
- Séparation du contenu et de la présentation
- Permet du « responsive design »

# Fonctionnalités

- Mise en forme
  - polices de caractères (type, taille, couleur, etc.)
  - paragraphes (alignement, retrait de première ligne, espacement, etc.)
  - cadres et tableaux (bordure, espacement, etc.)
- Positionnement des objets sur l'écran
  - relatif
  - absolu
- Adaptation des média

# Inclusion d'une CSS

- Directement dans les balises (à éviter de préférence)

```
<h2 style="color:red">Titre en rouge</h2>
```

- Définition de la CSS dans le fichier via la balise <style>

```
<head>  
  <style type="text/css">  
    déclaration des styles  
  </style>  
</head>
```

- Déclaration d'un lien vers la CSS via la balise <link>

```
<head>  
  <link href="fichier.css" rel="stylesheet" type="text/css" media="all"/>  
</head>
```

# Déclaration de règles

```
sélecteur {  
    propriété1: valeur1;  
    ...  
    propriétéN: valeurN;  
}
```

- On peut mettre autant de couples propriété/valeur que voulu, séparés par des ;
- Commentaire : /\* Commentaire \*/
- Exemple : `h2 {color:red;}`

# Déclaration de règles

- Sélecteur de type : nom de balise  
`h1 {color: red; background-color: yellow}`
- Sélecteur universel : \*
- Sélecteur d'ID : #  
`#important {color: red}` ou `p#important {font-size: 30pt}`  
Sélection de : `<p id="important">paragraphe</p>`
- Sélecteur de classe : .  
`.special {font-size: 20pt}` ou `h1.special {font-size: 60pt}`  
Sélection de : `<h1 class="special">Titre spécial</h1>`
- Remarques
  - Les id sont uniques sur une même page.
  - Les class s'appliquent à plusieurs balises.

# Déclaration de règles

- Sélecteur de descendant

`p h2 {color: green} : "les h2 qui sont dans un p"`

- Sélecteur d'enfant

`p > h2 {font-size: 30pt} : "les h2 qui sont directement dans un p"`

- Sélecteur d'adjacent :

`p + h2 {font-size: 10pt} : "les h2 qui sont directement après un p"`

- Sélecteur de pseudo-classe

`a:visited {color: brown}`

# Propriétés

- polices de caractères :

`font-size, font-style, font-family, font-weight`

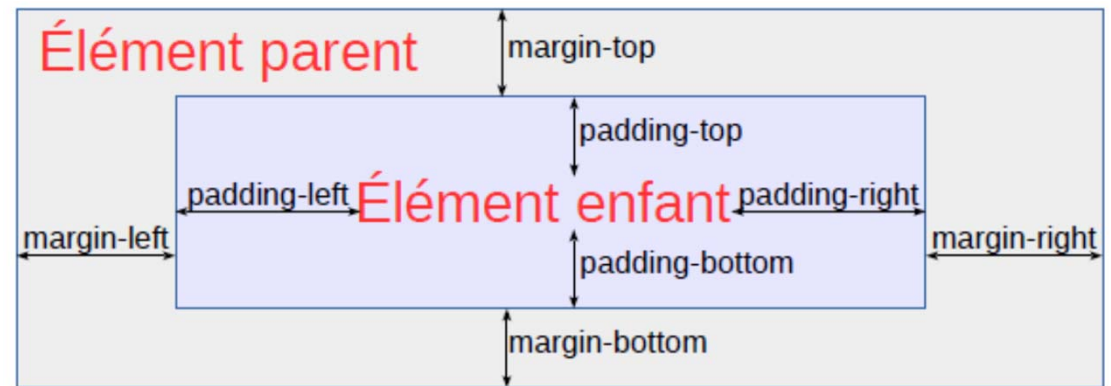
- paragraphes :

`line-height, text-align, text-indent, text-transform`

- blocs :

`height, width, margin-right, margin-left, margin-top, marginbottom, padding-right, border-style, border-top-width, ...`

# Propriétés



- polices de caractères :  
`font-size`, `font-style`, `font-family`, `font-weight`
- paragraphes :  
`line-height`, `text-align`, `text-indent`, `text-transform`
- blocs :  
`height`, `width`, `margin-right`, `margin-left`, `margin-top`, `marginbottom`, `padding-right`, `border-style`, `border-top-width`, ...



# Cascades

- Tous les styles, peu importe où ils sont définis, se **fusionnent** dans **l'ordre de chargement**, en une seule feuille de style avec un système d'héritage et **peuvent s'écraser**.

```
h1 {  
  color : red;  
  font-size : 18px;  
}  
h2 {  
  color: green;  
}  
+  
h1 {  
  color : blue;  
}  
h2 {  
  color: green;  
  font-size : 12px;  
}  
=  
h1 {  
  color : blue;  
  font-size : 18px;  
}  
h2 {  
  color: green;  
  font-size : 12px;  
}
```

# Cascades

- Tous les styles applicables sont appliqués.

```
h1 {color : red;}  
p {background-color:gray;}  
.untitre { font-suze:64px;}
```

+

```
<p> h1 class="untitre">Le titre qui a du style</h1>
```

=

Le titre qui a du style

# Positionnement

- Fonctionnement par défaut :  
Dans le flux, les éléments les uns en dessous des autres
- Positionnement flottant (par rapport au bloc contenant) :  
`float: left;`
- Positionnement absolu (par rapport à la fenêtre, hors flux) :  
`position: absolute; top: x px; left: y px;`
- Positionnement fixe (absolu même en cas de scroll) :  
`position: fixed; top: x px; left: y px;`
- Positionnement relatif (par rapport à sa position dans le flux) :  
`position: relative; top: x px; left: y px;`

# Exemple

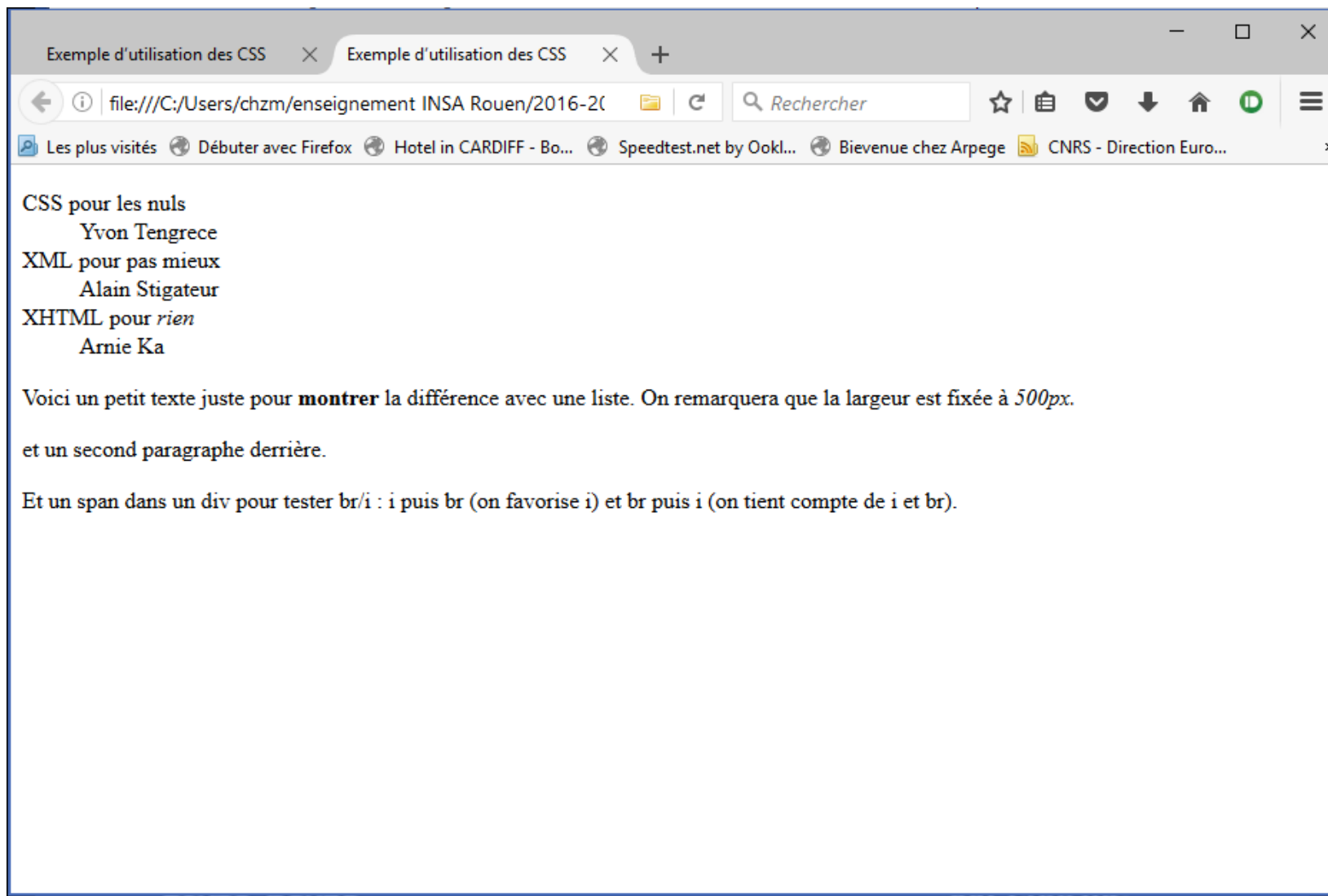
```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple d'utilisation des CSS</title>
  <meta http-equiv="content-type"
content="text/html; charset=utf-8" />
  <link href="test.css" rel="stylesheet" type="text/css"/>
</head>
<body>
  <dl>
    <dt>CSS pour les nuls</dt><dd>Yvon Tengrece</dd>
    <dt>XML pour pas mieux</dt><dd>Alain Stigateur</dd>
    <dt>XHTML pour <em>rien</em></dt><dd>Arnie Ka</dd>
  </dl>

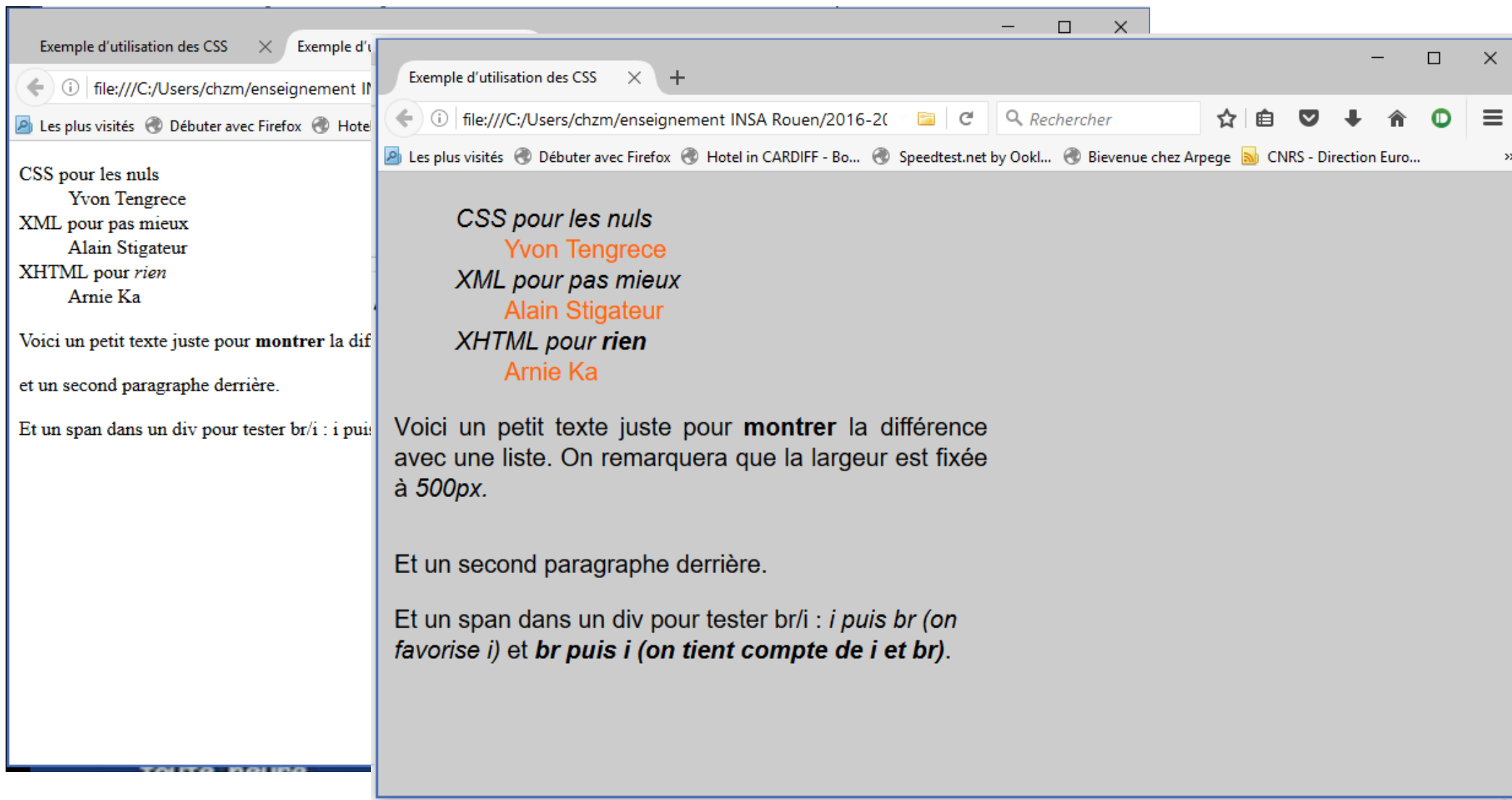
  <p>Voici un petit texte juste pour <strong>montrer</strong>
la différence avec une <span id="
important">liste</span>. On remarquera que
la largeur est fixée à <em>500px.</em></p>

  <p>et un second paragraphe derrière.</p>
  <div>Et un span dans un div pour tester br/i : <span
class="i"><span class="br">i puis br (on
favorise i)</span></span> et <span class="br"><span
class="i">br puis i (on tient compte de i
et br)</span></span>.</div>

</body>
</html>
```

```
html { font-family: arial; font-size: 20px;
font-style: normal; color: #000000;
background: #cccccc; margin: 5px; width:
500px; }
p { text-align: justify }
p:first-letter { text-transform: capitalize
}
p + p { margin-top: 10mm }
dl { margin-left: 50px; }
dt { font-style: italic; }
dt em { font-weight: bold; }
dd { color: #ff6600; }
#important { color: red }
.i>.br { font-weight: normal; font-style:
italic; }
.br>.i { font-weight: bold; font-style:
italic; }
```





# Récapitulatif et Références

# Récapitulatif

- Memento de balises HTML
  - <https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-des-balises-html>
- Memento des propriétés CSS
  - <https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-des-proprietes-css>



# Plus d'information

- <http://www.w3.org/html/>
- <http://www.w3schools.com/>
- <https://html5.validator.nu/> (pour valider le code)