Université de Jijel
2ème Année Licence Informatique

Cours Développement d'application Web

Chapitre I : HTML et CSS

Historique
Arpanet

- 1959-68 : le programme ARPA naît pendant la guerre froide
  La peur d'une guerre nucléaire
  - Faiblesse du système centralisé *versus* distribué
  - Proposition d'un maillage d'ordinateurs (1964, P. Baran)
  - 1ère communication téléphonique entre 2 machines en 1965
- 1969 : ARPANET
  1969 : 4 noeuds, 1971 : 15 nœuds, 1972 : 37 nœuds
- 1970-82 : ouverture sur le monde
  - Apparition du courrier électronique
  - Communications internationales (Angleterre, Norvège)
  - Apparition de TCP/IP (1974) plus puissant que NCP
- 1983 : TCP/IP adopté comme standard ARPANET ⇒ Internet

Historique
Internet/World Wide Web

- 1983-89 : expansion du réseau (*autoroutes de l'information*)
  - La NSF [1] effectue des progrès importants (réseau NFSNET)
  - Utilisation importante par les scientifiques
  - Réseaux hétérogènes (NCP et TCP/IP)
  - Fin officielle de ARPANET en 1989 (TCP/IP)
- 1990-97 : explosion d'internet
  - 1990, le physicien Tim Berners Lee (CERN) étend le concept de lien hypertexte à Internet
  - HyperText Markup Language (HTML) et HyperText Transfer Protocol (HTTP)
  - 1^{er} navigateur : NCSA Mosaic
  - 1995 ouverture au grand public (Netscape et Internet Explorer)
  - 1997 des dizaines de milliers de nœds dans plus de 42 pays

1. National Science Foundation

Généralités

- Qu'est-ce que le Web (ou World Wide Web, Toile, WWW, W3) ?
  - Système hypertexte public : système contenant des documents liés entre eux par des hyperliens permettant de passer automatiquement d'un document à l'autre.
- Différence entre le Web et Internet ?
  - Internet : réseau mondial d'ordinateurs permettant aux utilisateurs de communiquer (courrier électronique), de publier des informations (Web), de transférer des données (FTP), de travailler à distance (telnet et ssh)...
  - Web : un aspect d'Internet.

Architecture client-serveur

Le client (navigateur : Internet Explorer, Firefox, Safari...)

- demande au serveur des informations
- affiche des pages pour l'utilisateur

Le serveur (Apache, Microsoft IIS...)

- reçoit en permanence les requêtes des clients
- renvoie les documents correspondants

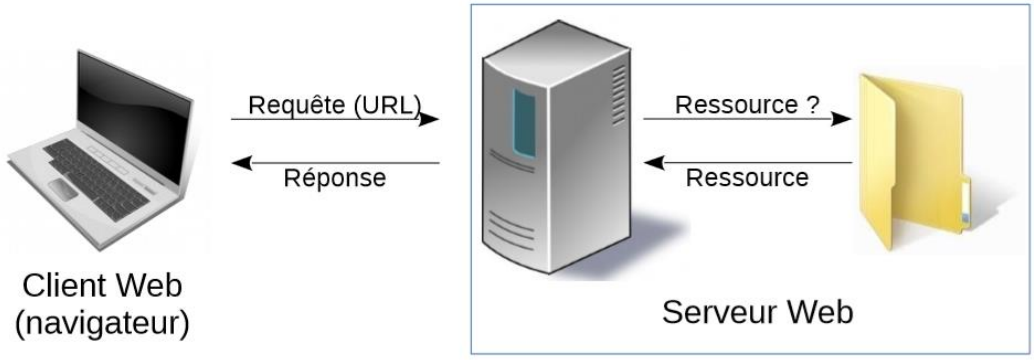
Vocabulaire

serveur Web : soit le logiciel qui répond aux requêtes d'un client, soit la machine sur lequel fonctionne ce serveur.

site Web : ensemble de pages Web et d'éventuelles autres ressources, liées dans une structure cohérente, publiées par un propriétaire (une entreprise, une administration, une association, un particulier, etc.) et hébergées sur un ou plusieurs serveurs Web.

hébergeur Web : entreprise de services informatiques hébergeant (mettant en ligne) sur ses serveurs Web les ressources constituant les sites Web de ses clients.

Principes de fonctionnement
La base du Web



- Architecture Client/Serveur
- Nécessité d'un protocole de communication : Http

Principes de fonctionnement (2/7)
URL, URN et URI

- URL : Uniform Ressource Locator
  - Spécification de la localisation d'une ressource de manière unique
- URN : Uniform Ressource Name
  - Mécanisme de nommage des ressources
  - `urn:<Namespace>:<SpecificString>`
  - `Namespace` : identificateur de nommage (ex : isbn)
  - `SpecificString` : chaîne de caractères spécifique désignant la ressource de manière unique
- URI : Uniform Resource Identifier
  - `URI = URL + URN`
  - En pratique, la forme d'URI la plus utilisée est l'URL

URL

- URL : *Uniform Resource Locator*
- Identifie l'endroit où se trouve une ressource sur le Web.
- Dans le cas du Web, ressource = document ou fragment
- `http://lea-linux.org:80/reseau/secu/firewall.html#intro`
  protocole machine port répertoire fichier fragment
- Principaux protocoles utilisés dans les URL :

  http, https deux protocoles du Web, sans et avec chiffrement et authentification ;

  ftp protocole de transfert de fichier, parfois utilisé sur le Web pour le téléchargement de gros fichiers ;

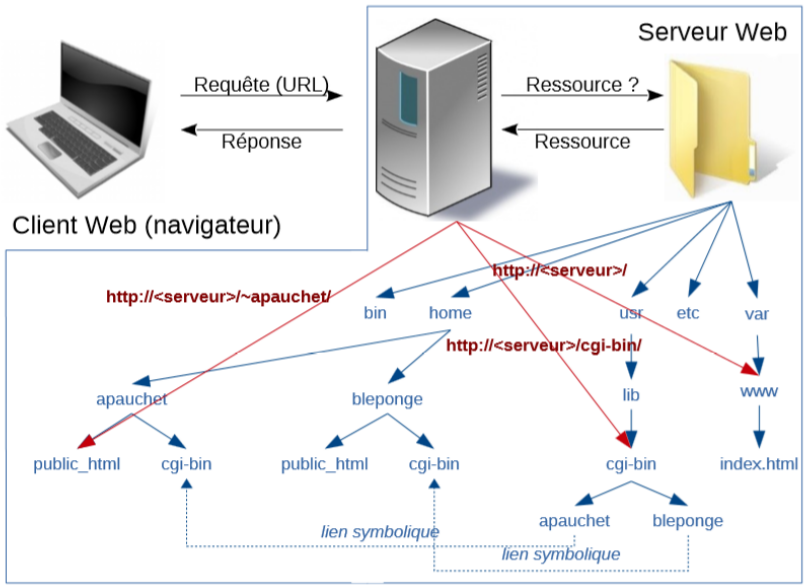
  mailto pseudo-protocole dénotant une adresse e-mail.
- Port par défaut : 80 pour HTTP, 443 pour HTTPS

HTTP

Protocole Ensemble normalisé de règles décrivant la manière de transmettre des informations, par exemple sur un réseau comme Internet entre un client et un serveur.

HTTP HyperText Transfer Protocol, le plus utilisé des protocoles de communication sur le World Wide Web. Permet à un client Web d'indiquer quelle page il veut obtenir, et au serveur Web de lui répondre en lui donnant cette page.

Principes de fonctionnement
Exemple : système de fichiers Apache HTTP



Limites
Http sécurisé : Https

Le protocole Http n'est pas sécurisé
⇒ Alternative à Http : **Https**

- 's' pour secured
- Combinaison de Http avec SSL ou TLS
- Vérification de l'identité d'un site par un certificat d'authentification
- Garantie confidentialité et intégrité des données envoyées par l'utilisateur (ex : formulaires)

HyperText Markup Language
Historique

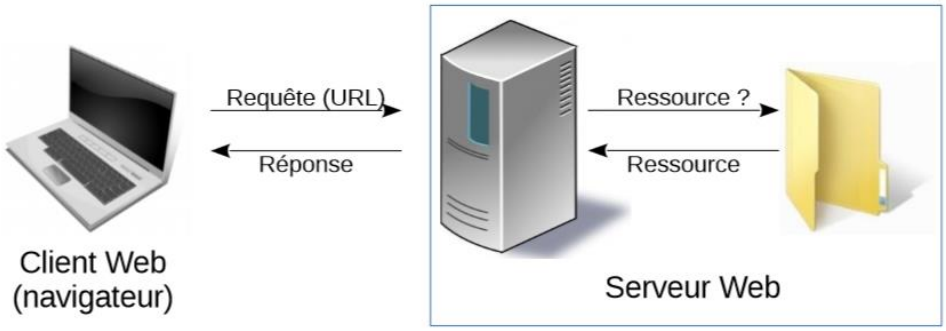
- Années 1990 : HTML est créé par Tim Berner-Lee au Centre Européen de Recherche Nucléaire (CERN)
- 1995 : HTML 2.0 normalisation par l'IETF [1]
- 1996 : HTML 3.2 ajout des tables, des applets (Java), *etc.*
- 1998 : HTML 4.01 ajout des feuilles de styles, des frames, *etc.*
- 2000 : XHTML 1.0 reformulation de HTML 4 en XML 1.0
- 2002-2006 : XHTML 2.0 en cours de spécification
- 2007-maintenant : HTML5

Normalisation par le W3C [2] depuis 1996.

HyperText Markup Language

- normalisé par le W3C (World Wide Web Consortium) regroupant industriels (Microsoft, Google, Apple...) et académiques (INRIA, MIT...)
- format ouvert : lecture possible dans des conditions correctes sans contrainte matérielle ou logicielle
- un fichier texte avec des balises
- description de la structure et du contenu d'un document, accent sur l'accessibilité
- on ne décrit pas la présentation (ce sera le rôle de CSS)
- on ne décrit pas de comportement dynamique (ce sera le rôle de JavaScript et des langages côté serveur)

HyperText Markup Language
Principe de fonctionnement



1. Le navigateur effectue une requête spécifiée à travers l'URL
2. Le serveur retourne un flot typé de données
3. Le navigateur interprète le flot de données et l'affiche

HyperText Markup Language
Langages à balises

- Un fichier HTML/XHTML est un fichier **texte** (*cf.* protocole HTTP) contenant des **balises** appelant des commandes, dont l'action est limitée au texte contenu entre la **balise de début** et la **balise de fin**.
- Extension HTML : `.htm` ou `.html` ; XHTML : `.xhtml`
- Balise de début : `<commande>`
- Balise de fin : `</commande>`
- Balise auto-fermante : `<commande/>`
- Commentaires : `<!--Ceci est un commentaire-->`

- HTML est un langage qui alterne texte et balises (`<blabla>` ou `</blabla>`)
  - Les balises permettent de structurer chaque partie du document et servent par exemple au navigateur pour réaliser la mise en page du document.
- Les fichiers HTML
  - sont structurés en deux parties principales : l'en-tête `<head> ... </head>`) et le corps `<body> ... </body>`)
- En HTML, les blancs (espace, tabulations, retours à la ligne) sont en général équivalents et servent juste à délimiter mots, balises... Leur nombre n'a pas d'importance.

Balises

- Leur syntaxe est (balises ouvrante et fermante)

```
<balise attributs>contenu</balise>
```

ou (balise sans contenu)

```
<balise attributs>
```

balise : mot clé désignant un élément particulier

contenu : peut contenir du texte ou d'autres balises

attributs : représente les différents paramètres associés à l'élément, sous la forme d'une liste de `nom="valeur"` ou `nom='valeur'`, séparés par des espaces (les guillemets ne sont pas toujours indispensables, mais elles le deviennent dès que `valeur` contient des caractères exotiques)

Balises

- Les noms des éléments et des attributs sont souvent écrits en minuscule, mais `<head>` et `<HeAd>` sont équivalents.
- Les balises sont ouvertes et refermées dans l'ordre (`<b><i></i></b>` et non `<b><i></b></i>`).
- Des règles strictes déterminent quelles balises peuvent être mises à l'intérieur de quelles balises.
- Sous certaines conditions, une balise peut être implicitement refermée, mais ces conditions sont assez complexes à décrire.

Balises : exemples

Exemples

- `<title>coucou</title>` pour attribuer le titre *coucou* au document
- `<em>cuicui</em>` pour mettre en *emphase* le texte *cuicui* (cela sera rendu, le plus souvent, par une mise en italique).
- `<strong>cuicui</strong>` pour indiquer que le texte **cuicui** est important (cela sera rendu, le plus souvent, par une mise en gras).

Contre-exemple

```
<strong><em>bouh</strong></em>
```

Structure d'un document

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html lang="fr">
  <head>
    <!-- En-tête du document -->
  </head>
  <body>
    <!-- Corps du document -->
  </body>
</html>
```

- La déclaration `<!DOCTYPE ...>` précise la version d'HTML utilisée.
- La langue du document est précisée avec l'attribut `lang` de la balise principale `<html>`.

En-tête

- L'en-tête du document est délimitée par les balises `<head>...</head>`.
- L'en-tête contient des méta-informations concernant le document telles que son titre, son encodage, les fichiers annexes, etc. Les deux informations les plus importantes sont :
  - Le jeu de caractères de la page, à mettre tout au début de l'en-tête

    ```
    <meta http-equiv="Content-Type"
          content="text/html; charset=utf-8">
    ```

  - Le titre de la page (la seule information obligatoire à préciser) ; celui-ci sera par exemple affiché dans la barre de titre du navigateur, il n'apparaît pas dans la page elle-même.

    ```
    <title>Mon super site</title>
    ```

- -Les balises méta.htm

- Les balises `<body> ... </body>` délimitent le corps du document.
- Le corps est structuré en sections, paragraphes, listes, etc.
- Il existe 6 balises permettant de représenter les titres de sections, par importance décroissante :
  - `<h1>Titre de la page</h1>`
  - `<h2>Titre de section principale</h2>`
  - `<h3>Titre de sous-section</h3>`
  - `<h3>Titre de sous-sous-section</h3>`
  - ...
- Les balises `<p> ... </p>` permet de délimiter un paragraphe. Tous les paragraphes de texte doivent être balisés ainsi.
- La balise `<hr>` (horizontal rule) indique une séparation majeure dans le document (rendue par exemple graphiquement par une ligne horizontale).
- Directement à l'intérieur de `<body> ... </body>` n'apparaissent que des balises de bloc : `<p>`, `<h1>`, `<form>`, `<hr>`, `<ul>`, `<table>` ainsi que la balise `<div>` qui dénote un bloc sans sémantique particulière.

Listes

- XTHML possède plusieurs balises permettant de présenter le texte sous forme de listes.
- On en distingue trois types :
  - les listes non numérotées,
  1. les listes numérotées,

  les listes de définitions (ou lexiques)
- Ces listes peuvent être emboîtées les unes à l'intérieur des autres.

- Les listes classiques :
  - Les listes non numérotées délimitées par les balises `<ul> ... </ul>` (unordered list).
  - Les listes numérotées délimitées par les balises `<ol> ... </ol>` (ordered list).
  - Tous les éléments d'une liste numérotée ou non sont délimités par les balises `<li> ... </li>` (list item)
- Les lexiques sont délimités par les balises `<dl> ... </dl>` (definition list) et leurs entrées par les balises `<dt> ... </dt>` (term) et `<dd> ... </dd>` (definition).

Exemples

```
<ol> <li>un</li> <li>deux</li> </ol>

<dl> <dt>lapin</dt> <dd>rongeur à oreilles</dd> </dl>
```

Tableaux

- Les tableaux sont délimités par les balises `<table>... </table>`.
- Les balises `<tr> ... </tr>` (table row) délimitent les lignes.
- Les balises `<td> ... </td>` (table data) délimitent les cellules.
- Attention ! On déclare les lignes à l'intérieur du tableau, les cellules à l'intérieur des lignes.

Exemple

```
<table>
  <tr> <td> l1, c1 </td> <td> l1, c2 </td> </tr>
  <tr> <td> l2, c1 </td> <td> l2, c2 </td> </tr>
</table>
```

Ajouter de la structure à un tableau en :

- donnant une légende au tableau avec les balises `<caption>... </caption>` juste après la balise ouvrante `<table>`.
- remplaçant les `<td> ... </td>` qui contiennent des en-têtes (de ligne, de colonne) par des `<th> ... </th>` (table header).

Images

- Pour insérer une image dans un document HTML, on utilise la balise `<img>`.
  - L'attribut `src` permet de préciser où se trouve l'image.
  - L'attribut `alt` permet de remplacer l'image par un texte quand elle n'est pas disponible. Il est obligatoire de l'utiliser, pour que tout agent (malvoyants, navigateur texte, incidents techniques, robots) ne pouvant voir votre image puisse avoir un texte alternatif.

```
<img src="http://www.lri.fr/images/lri.png" alt="LRI">
<img src="../images/montblanc.png" alt="Mont Blanc">
```

- Les formats d'images utilisables pour le Web sont :
  - Le JPEG (.jpg), un format adapté aux photos.
  - Le GIF (.gif) et le PNG (.png), des formats adaptés aux autres types d'image ; le PNG est à préférer dans tous les cas (transparence, profondeur de couleurs...) sauf besoin d'images animées (à utiliser avec parcimonie !).

Liens

- Ce qui différencie une page Web (page HyperTexte) d'un banal document : ce sont les liens !
- Ils sont introduits par la balise `<a> ... </a>` (cette balise est une balise en ligne, il faut la placer à l'intérieur d'un bloc).
- En cliquant sur un lien, on peut se déplacer vers :
  - un autre serveur ou un fichier du même serveur
  - une autre partie du même document

- Pour faire un lien, on utilise l'attribut `href` de la balise `<a>` dont le contenu formera le lien :

```
<a href="http://www.cnrs.fr/">
  <img src="images/cnrs.gif" alt="CNRS">
</a>

<a href="bio/indexbioinfo.html">Bioinformatique</a>
```

- Les ancres servent à atteindre un endroit précis dans le document.
  - On commence par définir les ancres, soit sur une balise existant déjà grâce à l'attribut `id`, soit avec un `<a id="...">` :

```
<h3 id="tutorials">Tutorials</h3>
<a id="tutorials">
```

  - Ensuite, on fait le lien avec cette ancre.

```
<a href="#tutorials">tutorials</a>
<a href="http://www.w3.org/#tutorials">tutorials</a>
```

  - On voit parfois l'ancienne syntaxe `<a name="...">`.

Et aussi...

- De nombreuses autres balises en ligne : `<abbr>`, `<cite>`, `<var>`...
- Quelques autres balises blocs : `<blockquote>`, `<address`...
- Représentation des caractères spéciaux (ex., « é ») :
  - directement dans le codage du document (utf-8 de préférence, pour représenter tous les caractères possibles), à privilégier ;
  - par leur code en décimal (é) ou en hexadécimal (é) ;
  - par des entités caractères nommées (é)

Formulaires

- Permettent d'interagir avec l'utilisateur en lui proposant d'entrer des informations
- En HTML : uniquement l'interface de formulaire
- L'essentiel du travail sera fait par le script qui traitera la soumission du formulaire

Balise <form>

- Un formulaire HTML est placé à l'intérieur d'une balise `<form>`.
- Celle-ci prend les attributs suivants :

action	URL du script auquel sera soumis le formulaire.
method	Méthode HTTP, valant soit `"get"` soit `"post"`.
enctype	Encodage HTTP. Peut valoir `"application/x-www-form-urlencoded"` (valeur par défaut) ou `"multipart/form-data"`.

Example (Formulaire élémentaire)

```
<form action="action.php" method="get">
  <div><input type="submit"></div>
</form>
```

Ensembles de champ

En HTML, il est interdit de mettre des champs de formulaire directement à l'intérieur d'un `<form>`. Il faut d'abord les regrouper :

- Dans des paragraphes `<p>` si les champs de formulaires sont à l'intérieur de paragraphes de textes (rare).
- Dans des ensembles de champ `<fieldset>` pour regrouper des champs de formulaire de sémantique proche. On pourra alors donner une légende à l'ensemble de champs avec la balise `<legend>`.
- Dans des divisions `<div>` sans contenu sémantique sinon.

Exemple (Ensemble de champ)

```
<fieldset>
  <legend>Mensurations</legend>
  <input type="text" name="taille">
  <input type="text" name="poids">
</fieldset>
```

Étiquettes

- La plupart des champs sont naturellement accompagnés d'une étiquette (`<label>`).
- On peut la placer où on veut, en général juste à gauche ou à droite du champ.
- Son attribut `for` référence l'attribut `id` du champ correspondant.
- Dans les navigateurs graphiques, un clic sur l'étiquette d'un champ permet en général de sélectionner le champ.

Example (Étiquette)

```
<label for="taille">Taille :</label>
<input type="text" name="taille" id="taille">
```

Saisie d'une ligne de texte

- `type ="text"` est utilisé pour la saisie d'un texte dont la taille est inférieure à une ligne.
- L'attribut `value` permet de préciser la valeur par défaut (facultatif).
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut `maxlength` (facultatif).

Exemple

```
<input type="text" name="prenom" value="Jordy" maxlength="50">
```

Choix multiples parmi une liste

- `type ="checkbox"` permet de choisir plusieurs éléments parmi une liste de possibilités.
- Cela se matérialise sous forme de cases à cocher.
- La valeur retournée est obligatoirement précisée à l'aide de l'attribut `value`.
- L'attribut `checked ="checked"` permet de cocher la case par défaut.
- Certains langages côté serveurs imposent que les noms de champs de formulaire à choix multiples se terminent par [].

Exemple

```
<input type="checkbox" name="pub[]" value="site"
  checked="checked" id="pub-site">
<label for="pub-site">Recevoir des offres de notre site</label>

<input type="checkbox" name="pub[]" checked="checked"
  value="externe" id="pub-externe">
<label for="pub-externe">Recevoir des offres externes</label>
```

Choix unique parmi une liste

- `type ="radio"` permet de choisir un seul élément parmi une liste de possibilités.
- Cela se matérialise sous forme de boutons radio.
- La valeur retournée est obligatoirement précisée à l'aide de l'attribut `value`.
- L'attribut `checked ="checked"` permet de préciser la valeur par défaut.

Exemple

```
Recevoir de la pub:
<input type="radio" name="pub" value="oui" id="pub-oui"
  checked="checked">
<label for="pub-oui">oui</label>

<input type="radio" name="pub" value="non" id="pub-non">
<label for="pub-non">non</label>
```

Fichiers joints

- `type ="file"` permet de joindre au formulaire un fichier.
- À cause de la taille de la requête due au téléchargement (upload) du fichier, il faut impérativement utiliser la méthode POST et l'encodage `multipart/form-data`.

Exemple

```
<label for="fichier">Fichier:</label>
<input type="file" name="fichier" id="fichier">
```

Champs cachés

- `type ="hidden"` permet de cacher des champs au client mais leur contenu est envoyé avec le formulaire.
- Ceci permet de préciser des informations, en utilisant l'attribut `value`, concernant l'interaction client/serveur.
- C'est à utiliser avec précaution car cela peut être à l'origine de problèmes de sécurité assez graves : ne pas oublier que le client peut éditer la page à la main pour changer la valeur de ces champs !

Exemple

```
<input type="hidden" name="monnaie_utilisee" value="EUR">
<input type="hidden" name="customerCB" value="c2415-345-8563">
```

Ré-initialisation d'un formulaire

- `type ="reset"` permet de réinitialiser le formulaire en affectant aux différents champs leur valeur par défaut.
- L'attribut `value` permet de changer le texte du bouton correspondant.

Exemple

```
<input type="reset" value="Tout effacer">
```

Soumettre le formulaire

- `type ="submit"` permet de soumettre le formulaire.
- Le client envoie le contenu du formulaire à l'adresse précisée par l'attribut `action` de la balise `<form>`.
- L'attribut `value` permet de changer le texte du bouton correspondant.

Exemple

```
<input type="submit" value="Envoyer">
```

Saisie de plusieurs lignes de texte

- Pour les saisies multiligne, on utilise la balise `<textarea>`.
- Le texte délimité par cette balise permet d'initialiser la valeur par défaut du champ.
- La balise fermante est obligatoire même si le champ est vide.
- Les attributs `rows` et `cols` (obligatoires) permettent de spécifier la taille en lignes et colonnes de la fenêtre de saisie.

Exemple

```
<textarea name="bio" cols="40" rows="5">
  Fille de Josiane Balasko,
  Marilou Berry fait ses premiers pas au cinéma à 8 ans...
</textarea>
```

Menus de sélection

- La balise `<select>` permet d'ajouter une liste de sélection :
  - L'attribut facultatif `size` permet de préciser le nombre de choix apparaissant sur la page Web. Par défaut, ce nombre est initialisé à 1.
  - L'attribut `multiple ="multiple"` permet d'autoriser des réponses multiples. Dans ce cas, pour PHP, on donnera toujours un nom se terminant par `[]`.
- Les choix du menu sont indiqués à l'aide de la balise `<option>` :
  - L'attribut `selected ="selected"` permet de spécifier le(s) choix par défaut.
  - L'attribut `value` permet de spécifier la valeur associée au choix.

Exemple

```
<select name="age">
  <option value="20">Moins de 20 ans</option>
  <option value="35" selected="selected">21 à 35 ans</option>
  <option value="50">36 à 50 ans</option>
  <option value="51">Plus de 51 ans</option>
</select>
```

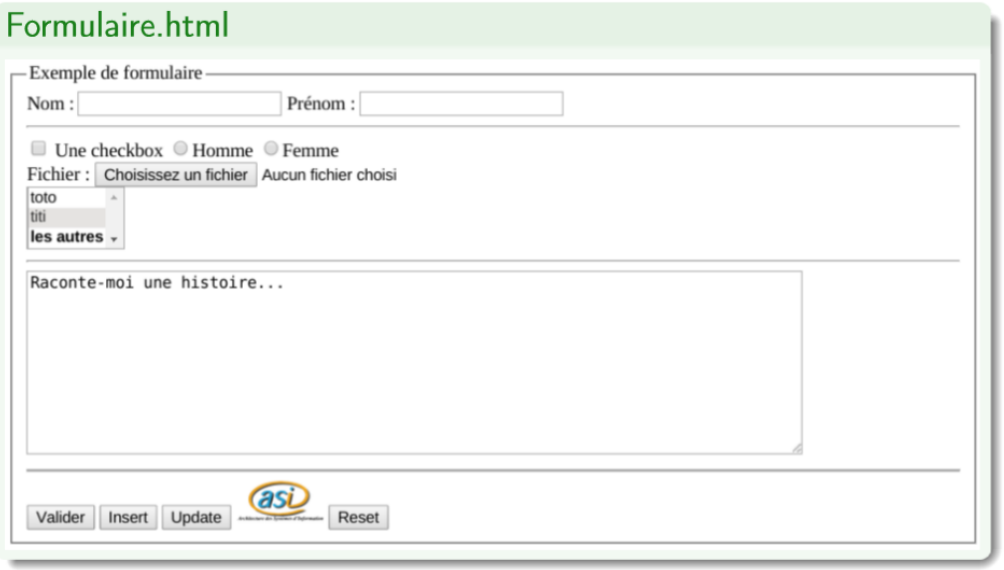
Les formulaires
Exemple de formulaire

Formulaire.html

```
<form action="GET">
   <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/logoasi.png" alt="logoasi.png" name="action" width="75"/>
      <input type="reset" value="Reset"/>
   </fieldset>
</form>
```

Les formulaires
Exemple de formulaire

Formulaire.html



Les cadres

- Création des cadres indépendant à l'intérieur d'une fenêtre
- Accélération du chargement de la page
  - seule le cadre qui change est chargé
- Une page HTML est souvent divisée en :
  - un petit cadre contenant des index (liens)
  - un grand cadre contenant les données à afficher

Définition des cadres : balise <FRAMESET>

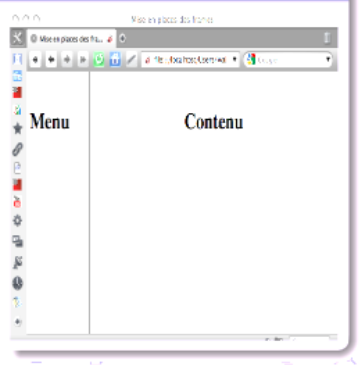
Pour définir des cadres on utilise la balise
<FRAMESET>...</FRAMESET>

- elle contient la description de chacun des cadres présents
- pas de balise <BODY>

Un exemple

```
<HTML>
<HEAD>
   <TITLE> Mise en place des frames
</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <frame src="menu.html" name="menu">
  <frame src="cont.html"  name="cont">
</FRAMESET>
</HTML>
```

Cela donne



Attributs de la balise <FRAMESET>

- ROWS : division en *zones horizontales*
  - ROWS="h1,h2,...,hn"
  - hi peut être
    - hi=n : hauteur du cadre en pixels
    - hi=n% : hauteur du cadre en % de la taille de la zone mère
  - ex : pour avoir 2 frames horizontales de taille identique :
    - <frameset rows="50%,50%">
- COLS : division en *zones verticales*
  - COLS="l1,l2,...,ln"
- FRAMEBORDER : frontières avec effet 3D
  - valeurs : yes (1) ou no (0), 1 par défaut
- BORDERCOLOR : couleur des bordures
- BORDER : largeur de la bordure
  - BORDER="0" $\Longrightarrow$ cadres invisibles

Attributs de la balise <FRAME>

- src="URL de la page à afficher dans le cadre"
- name="nom du cadre"
  pour référencer le cadre afin qu'il devienne la cible d'un autre lien hypertexte
- MARGINWIDTH="n" (15 par défaut)
  - nombre de pixels entre les frontière gauche et droite de la zone et le document HTML à afficher dans la zone
- MARGINHEIGHT="n"
  - nombre de pixels entre les frontière haute et basse de la zone et le document HTML à afficher dans la zone
- NORESIZE : pas de valeur
  - le navigateur empêche le redimensionnement de la zone à l'aide de la souris.

Les attributs de la balise <FRAME>

SCROLLING="yes / no / auto"

- indique si la zone doit posséder une barre de défilement
- yes : affiche une barre même si le document est plus petit que le cadre
- no : n'affiche jamais la barre (des parties de la page peuvent ne pas être atteinte)
- auto : la barre apparaît si nécessaire (valeur par défaut)

Utilisation des cadres : attribut target

target="nom du cadre" : attribut de <A> et <FORM>

- désigne un cadre cible du lien
- nom : nom local du frameset

Fichier *index.html*

```
<HTML>
<HEAD>
   <TITLE> Mise en place des frames
</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <frame src="menu.html" name="menu">
  <frame src="cont.html"  name="cont">
</FRAMESET>
</HTML>
```

Fichier *menu.html*

```
<html> <head>
<title> mise en place de frames </title>
</head> <body> Menu
<a href="c1.html" target="cont"> Cours 1
</a>
<a href="c2.html" target="cont"> Cours 2
</a>
<a href="c3.html" target="cont"> Cours 3
</a>

</body></html>
```

Utilisation des cadres : attribut target

Il faut créer *cont.html*, *c1.html* ...

Fichier *cont.html*

```
<HTML>
<HEAD>
   <TITLE> Mise en place des frames
</TITLE>
</HEAD>
<BODY>
   ici apparaitra le contenu
</BODY>
</HTML>
```

Fichier *c1.html*

```
<HTML>
<HEAD>
<TITLE> Cours 1 </TITLE>
</HEAD>
<BODY>
   Outis Web 1 ...
</BODY>
</HTML>
```

Les liens et les cadres : attribut target

L'attribut target :

- contient le nom d'un cadre existant
  (la cible est affiché dans le cadre nommé)
- si target n'est pas renseigné alsors
  la cible est affiché dans le cadre courant
- si target contient le nom d'un cadre inexistant alors
  la cible est affichée dans une nouvelle fenêtre
- target peut contenir une valeur réservée :
  - _self : la cible est affichée dans le cadre où est définie le lien
  - _parent : la cible affiché dans le cadre *père* (celui qui l'a créé)
  - _blank :la cible est affichée dans une autre fenêtre sans structure de cadre
  - _top : la cible affichée replace la fenêtre courante
    ($\Longrightarrow$) suppression de tous les cadres
  - utiliser _top dès qu'un lien pointe sur une page extérieure

Imbrication de cadres

```
<HTML> <HEAD>
<TITLE> Imbrication des cadres </TITLE>
</HEAD>
<FRAMESET row="25%,75%" frameborder="yes">
  <frame src="haut.html" name="haut" >
  <FRAMESET cols="70%,30%" border="10">
    <frame src="bas_gauche.html" name="basgauche">
    <frame src="bas_droit.html" name="basdroit">
  </FRAMESET>
</FRAMESET>
</html>
```

HTML contre XHTML

- XHTML : une application d'XML
- Les balises sans contenus `<hr>`, s'écrivent `<hr />` en XHTML.
- Certains éléments peuvent ne pas être refermées en HTML (`<ol> <li> un <li>deux </ol>`), mais la fermeture est obligatoire en XHTML.
- Les valeurs des attributs peuvent ne pas être entre guillemets (`<img src=toto.png alt=toto>`) en HTML, guillemets obligatoires en XHTML.
- Les noms des éléments et des attributs sont insensibles à la casse en HTML (`<HTMl laNg=fr>`), contrairement à XHTML ou tout doit être en minuscule.
- Attributs `xmlns` et `xml:lang` sur la balise `<html>` en XHTML.
- Et quelques autres petites subtilités. . .

Avantages respectifs de HTML et XHTML

Avantages de HTML 4.01

- Meilleur support par les navigateurs (Internet Explorer 6/7 comprennent mal XHTML).
- Moins de contraintes...
- Beaucoup plus utilisé sur le Web.

Avantages de XHTML 1.0

- Plus de contraintes... (donc plus simple !).
- Syntaxe claire, sans ambiguïté.
- Familiarité avec XML, utile dans d'autres contextes.
- Facilité d'utilisation dans des contextes XML (p.ex. XSLT).

Règles de compatibilité à respecter pour du XHTML

- Théoriquement, `<p>` et `<p></p>` sont synonymes en XML, donc en XHTML. En pratique, on utilisera la notation `<balise>` uniquement pour les balises n'ayant jamais de contenu (p.ex. `<br>`, `<hr>`...).
- Théoriquement, `<br>` et `<br/>` sont synonymes. En pratique, on utilisera toujours `<br>`.
- Théoriquement, un document XHTML peut commencer par une ligne `<?xml version="1.0"encoding="utf-8"?>`. En pratique, on l'omettra.
- cf. `http://www.w3.org/TR/xhtml1/#guidelines`
- On peut utiliser `http://qa-dev.w3.org/~bjoern/appendix-c/validator/` pour vérifier le respect de ces quelques règles.

XHTML 2.0 contre HTML 5

- XHTML 2.0 : initiative du W3C, incompatible avec HTML 4.01/XHTML 1.0, changements majeurs
- HTML 5 : initiative des développeurs de navigateurs, compatibilité avec HTML 4.01/XHTML 1.0, changements incrémentaux mais nombreux
- XHTML 2.0 abandonné en juillet 2009
- Des fonctionnalités de HTML 5 font leur apparition dans les navigatuers récents (excepté Internet Explorer, qui a cependant annoncé en septembre 2009 son intérêt)
- HTML 5 laisse le choix entre les conventions HTML 4.01 ou XHTML
- Nouvelles fonctionnalités : dessin 2D (`<canvas>`), multimédia (`<audio>`, `<video>`), meilleurs éléments structurants (`<section>`, `footer`), etc.

Outils

- N'importe quel éditeur de texte (par exemple le bloc-notes de Windows, emacs, vim...). Privilégier les éditeurs avec coloration syntaxique. Bon choix : Scite (libre, léger, simple d'utilisation, multi-plateforme, coloration syntaxique pour HTML/CSS/PHP/JavaScript/...)
- N'importe quel navigateur, et plusieurs navigateurs avec un moteur de rendu différent
- Utiliser la fonction « Afficher la source » des navigateurs Web
- Firefox présente de nombreux avantages pour le développement/analyse de sites Web
- En particulier, extension Firebug de Firefox, excellent outil
- Pour vérifier qu'une page se conforme bien aux normes de HTML, utiliser le validateur du W3C : `http://validator.w3.org/`

Les CSS (1/12)
Description

Cascading Style Sheets
Feuilles de style en cascade

- Un document en mode texte
- 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
- 3 normes : CSS 1, CSS 2, CSS 3

Les CSS (2/12)
Inclusion d'une CSS

Trois possibilités d'inclusion :

- Directement dans les balises (à éviter)

```
<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"/>
</head>
```

Les CSS (3/12)
Déclaration d'une règle

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

- On peut mettre autant de couples propriété/valeur que voulu, séparés par des ;
- Commentaire : `/* Commentaire */`

Exemple

```
h2 {color:red;}
```

Les CSS (4/12)
Déclaration d'une règle (suite)

- 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>`

Remarque

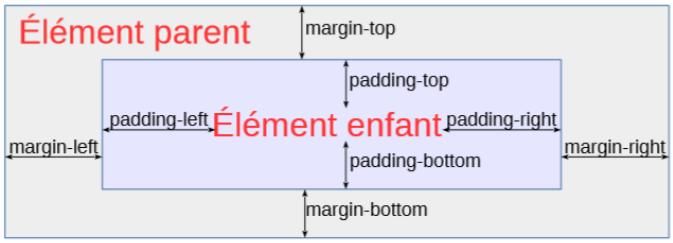
Les id sont uniques sur une même page.
Les class s'appliquent à plusieurs balises.

Les CSS (5/12)
Déclaration d'une règle (suite)

- 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}`

Les CSS (6/12)
Éléments et propriétés

- Propriétés (`http://www.w3schools.com/cssref/`)
  - 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, margin-bottom, padding-right, border-style, border-top-width, ...`



Les CSS (7/12)
Cascade

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;
}
```

Les CSS (8/12)
Cascade (suite)

Tous les styles applicables sont appliqués.

```
h1 {color : red;}
p { backgroung-color:grey; }
.untitre { font-size : 64px; }
```

+

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

=

Le titre qui a du style

Les CSS (9/12)
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 relatif (par rapport à sa position dans le flux) :
  `position: absolute; top: x px; left: y px;`

`http://www.w3schools.com/css/css_positioning.asp`

Le sélecteur d'élément

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Every paragraph will be affected by the style.

Me too!

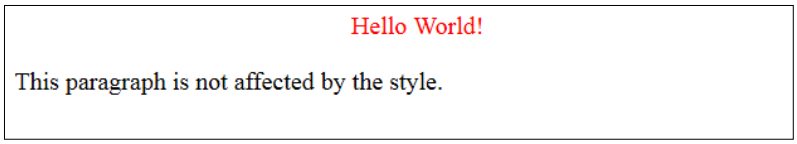
And me!

L'id sélecteur

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```



71

Le sélecteur de classe (pour tous les éléments)

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

72

Le sélecteur de classe (pour seulement <p> éléments)

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-
aligned.</p>

</body>
</html>
```

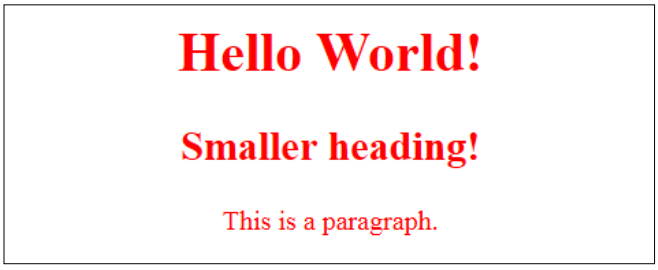
73

sélecteurs de regroupement

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>

</body>
</html>
```



74

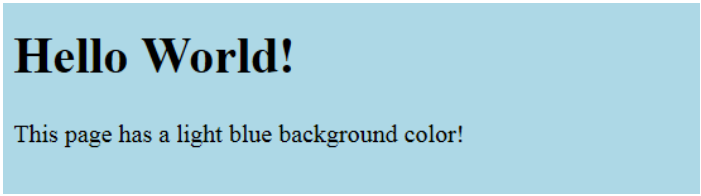
Définissez la couleur d'une page d'arrière - plan

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has a light blue background color!</p>

</body>
</html>
```



75

Définissez la couleur des différents éléments d'arrière - plan

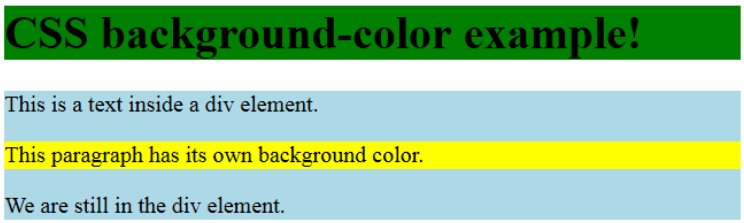
```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    background-color: green;
}

div {
    background-color: lightblue;
}

p {
    background-color: yellow;
}
</style>
</head>
<body>

<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>

</body>
</html>
```

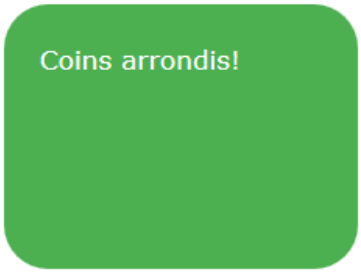


76

CSS3 coins arrondis

- Avec le CSS3, la propriété **border-radius**, vous pouvez donner tout élément "coins arrondis".
- Ex: coins arrondis pour un élément avec une couleur de fond spécifiée:

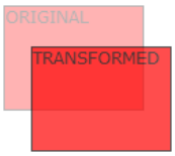
```
#rcorners1 {
  border-radius: 25px;
  background: #73AD21;
  padding: 20px;
  width: 200px;
  height: 150px;
}
```



Le translate () Méthode & La rotation () Méthode

- La **translate()méthode** déplace un élément à partir de sa position actuelle (en fonction des paramètres indiqués pour l'axe X et l'axe Y).
- L'exemple suivant déplace l'élément 50 pixels à droite et 100 pixels vers le bas de sa position actuelle:

```
div {
  -ms-transform: translate(50px, 100px); /* IE 9 */
  -webkit-transform: translate(50px, 100px); /* Safari */
  transform: translate(50px, 100px);
}
```



- Le **rotate()** procédé dans le sens horaire fait tourner un élément ou dans le sens antihoraire , selon un degré donné.
- L'exemple suivant fait pivoter l'élément dans le sens horaire avec 20 degrés:

```
div {
  -ms-transform: rotate(20deg); /* IE 9 */
  -webkit-transform: rotate(20deg); /* Safari */
  transform: rotate(20deg);
}
```

Bouton Couleurs

vert | Bleu | rouge | Gris | Noir

Utilisez la `background-color` propriété pour changer la couleur d'un bouton de fond:

Exemple

```
.button1 {background-color: #4CAF50;} /* Green */
.button2 {background-color: #008CBA;} /* Blue */
.button3 {background-color: #f44336;} /* Red */
.button4 {background-color: #e7e7e7; color: black;} /* Gray */
.button5 {background-color: #555555;} /* Black */
```

CSS auto

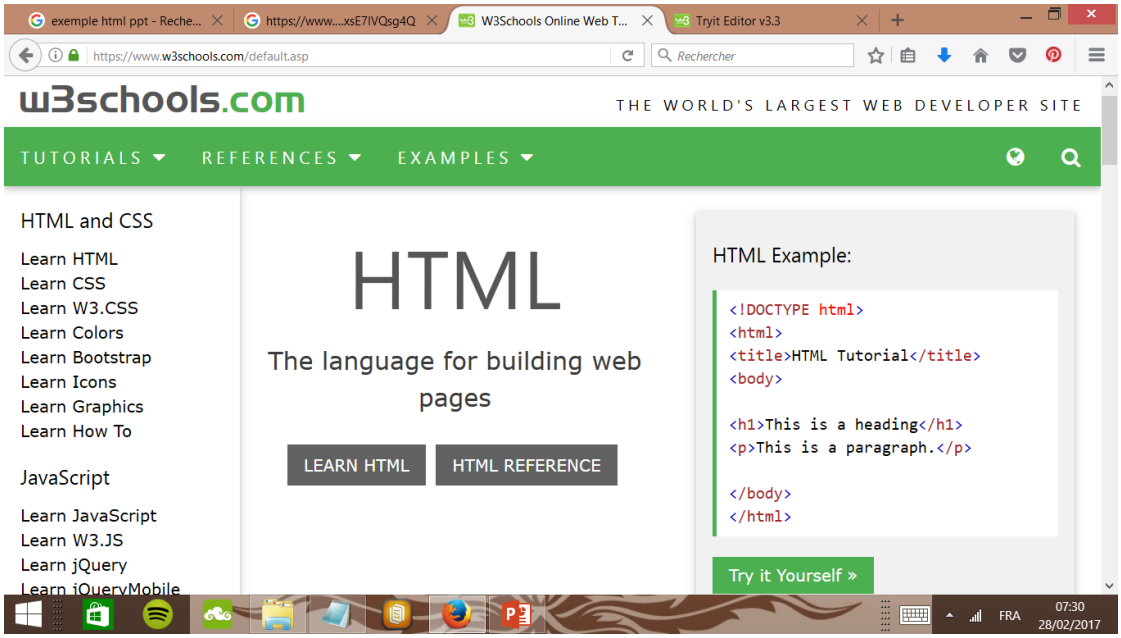
- http://www.elzedo.com/?page=css

- Spécification de CSS
- http://dicolive.media-box.net/docCSS/css.php?orderByType=1

- http://www.w3schools.com/

- http://css.alsacreations.com/

- code de couleurs
- http://www.docmemo.com/internetwebmasters/codescouleurs.php


exemple html ppt - Reche...
https://www....xsE7IVQsg4Q
W3Schools Online Web T...
Tryit Editor v3.3
https://www.w3schools.com/default.asp
Rechercher
w3schools.com
THE WORLD'S LARGEST WEB DEVELOPER SITE
TUTORIALS
REFERENCES
EXAMPLES
HTML and CSS
Learn HTML
Learn CSS
Learn W3.CSS
Learn Colors
Learn Bootstrap
Learn Icons
Learn Graphics
Learn How To
JavaScript
Learn JavaScript
Learn W3.JS
Learn jQuery
HTML
The language for building web pages
LEARN HTML
HTML REFERENCE
HTML Example:
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>
<h1>This is a heading</h1>
This is a paragraph.
</body>
</html>
Try it Yourself »
FRA
07:30
28/02/2017

81


exemple html ppt - Reche...
https://www....xsE7IVQsg4Q
W3Schools اون لاين الو...
Tryit Editor v3.3
https://www.w3schools.com/default.asp
Rechercher
Google
Traduit en : Arabe
Afficher l'original
Options
w3schools كوم
WEB DEVELOPER أكبر موقع لل في العالم
دروس
المراجع
أمثلة
Arabe
HTML و CSS
تعلم HTML
تعلم CSS
تعلم W3.CSS
تعلم الألوان
تعلم التنسيق
تعلم الرموز
تعلم الرسومات
تعلم كيفية
جافا سكريبت
تعلم جافا سكريبت
تعلم W3.JS
HTML
اللغة لبناء صفحات الويب
تعلم HTML
مرجع HTML
HTML مثال:
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>
<h1>This is a heading</h1>
This is a paragraph.
</body>
</html>
FRA
07:31
28/02/2017

82

83

Références

- Spécification de HTML 4.01
  `http://www.w3.org/TR/REC-html40/`
- Spécification de XHTML 1.0
  `http://www.w3.org/TR/xhtml1/`
- Brouillon de XHTML 2
  `http://www.w3.org/TR/xhtml2/`
- **HTML 5**
  w2schools : `http://www.w3schools.com/html5/`
  Toutes les balises :
  `http://www.w3schools.com/html5/html5_reference.asp`
- **CSS**
  w2schools : `http://www.w3schools.com/html5/`
  Balises :
  `http://www.w3schools.com/html5/html5_reference.asp`