

HTML

INTRODUCTION

WWW = World Wide Web

- Créé en 1989 au CERN par Tim Berners-Lee
 - ▣ Objectif initial : mettre en ligne de la documentation (initialement technique pour physiciens)
- 1993 : Mosaic 1er navigateur graphique (NCSA)
- 1994 : World Wide Web Consortium (W3C)
 - ▣ créé par le CERN et le MIT
 - ▣ organisme de standardisation du Web
 - ▣ administré par le MIT et l'INRIA
- 1994 : Netscape Mozilla navigateur (Win, Unix, Mac)
 - ▣ développé notamment par Marc Andreessen (un des auteurs de Mosaic)
- 1996 : Microsoft Internet Explorer
- 1996 : 1ère version d'Opera en téléchargement (1994 Telenor R&D, Norway)
- 2003 : création de la Mozilla Foundation (Firefox, Thunderbird)

WWW = World Wide Web

□ Services de l'Internet

- ▣ www
- ▣ messagerie
- ▣ transfert de fichiers (FTP)
- ▣ forum de nouvelles (news), de discussion (chat)
- ▣ peer-to-peer (Gnutella, Freenet, ...), diffusion de flux audio et vidéo (voip, vod, streaming)

□ Principes du Web

- ▣ hypertexte : HTML
- ▣ client/serveur : HTTP
- ▣ schéma de désignation : URL

World Wide Web - Hypertexte

- Texte "normal" :
 - ▣ Organisation linéaire.
 - ▣ Eventuellement renvois sous forme de sommaires, index, notes de bas de page.

- Hypertexte :
 - ▣ Organisation pas forcément linéaire.
 - ▣ Texte enrichi de liens :
 - renvoi vers un document.
 - renvoi vers une partie du même document.
 - renvoi vers une partie d'un autre document.

World Wide Web - Client/serveur

- Client : le navigateur (Internet Explorer, Firefox, Chrome, Opera, Safari...)
- Serveur : le serveur Web (Apache, Microsoft IIS...)

- 1. Le client émet une requête
- 2. Le serveur répond en fournissant le document demandé ou un message d'erreur si le document n'existe pas

World Wide Web - URL

- Schéma de désignation
- Uniform Resource Locator (URL)
 - ▣ Permet de désigner une page Web
 - ▣ Chaque page a un nom unique : pas d'ambiguïté possible
 - protocole://serveur/page
 - <http://www.lip6.fr/index.html>
- Organisation hiérarchique possible pour les pages
 - ▣ protocole://serveur/répertoire/.../page
 - ▣ <http://www.lip6.fr/congres/2002/index.html>

World Wide Web - standards

□ HTTP

- ▣ 0.9 : version de base avec requête/réponse
- ▣ 1.0 : version standardisée IETF (RFC 1945)
- ▣ 1.1 : version étendue (connexions persistantes)

□ HTML

- ▣ 1.0 : version initiale
- ▣ 2.0 : version standardisée W3C
- ▣ 3.2 : version étendue (tableaux, images cliquables, applets)
- ▣ 4.0 : version étendue (frames, feuilles de style) : version basique et stricte
- ▣ 5.0 : en cours de normalisation (déjà utilisé par la plupart des navigateurs)

□ URL

- ▣ format stable depuis 1989

(x)HTML

HyperText Markup Language

- HTML : langage de balisage issu de SGML
- Document HTML contient :
 - ▣ du texte
 - ▣ des balises (ou tags) : structuration ou mise en forme
 - `<i>Ce texte est en italique</i>`
 - ``
 - les balises peuvent être imbriquées
 - `<p>texte en <i>italique</i> !</p>`

HyperText Markup Language

- Structure d'un document HTML :
 - Déclaration de la version HTML utilisée
 - En-tête
 - Corps du document.
- HTML Strict : tout est obligatoire.
- Absence de corps = document vide.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Première page web</title>
</head>

<body>
  <p>Ceci sera affiché dans le navigateur</p>
</body>
</html>
```

Règles de balisage

- 2 types de balises
 - ▣ autonome `< ... >` (exemple : `<! DOCTYPE >`)
 - ▣ délimitant une zone
 - balise de début de zone `< ... >` (exemple : `<html>`)
 - balise de fin de zone `</ ... >` (exemple : `</html>`)
 - `` pour les balises sans contenu
 - Attention `</...>` est différent de `<.../>`
- Toutes les balises doivent être fermées (XHTML1.0 strict)

Règles de balisage

- Identifiants de balise non sensibles à la casse :
 - ▣ `<BODY>` est similaire à `<body>`
 - ▣ Les noms de balises doivent être en minuscules

- Des attributs peuvent être associés aux balises :
 - ▣ Un attribut à un identifiant et une valeur `id="valeur"`
 - ▣ ``
 - ▣ Les noms d'attributs doivent être en minuscules
 - ▣ Les valeurs d'attributs doivent être entre guillemets
 - ▣ Tous les attributs doivent avoir une valeur
 - ▣ Pas de `<input checked>`

Règles de balisage

□ Commentaires

- `<!-- ceci est un commentaire -->`

□ Tous les caractères spéciaux peuvent/doivent être encodés :

- format : `"&xxx;"` avec xxx variable

- `< > & : < > &`

- caractères accentués : `&` lettre accent ;

 - acute, grave, circ, uml, tilde, ring

- `é à ô : é à ô`

STRUCTURE D'UN DOCUMENT



En-tête des documents HTML

- Langage et version utilisés par le document :
 - ▣ les navigateurs savent s'ils pourront afficher le document.
 - ▣ `<!DOCTYPE langage statut version >`
 - ▣ <http://validator.w3.org>

- Exemple
 - ▣ `<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>`
 - ▣ `<!DOCTYPE html>` pour HTML5

Balise html

- La balise html englobe tout le document (Sauf doctype)
 - ▣ Uniquement 2 balises filles possibles : head et body
- Attributs possibles :
 - ▣ xmlns : <http://www.w3.org/1999/xhtml>
 - ▣ dir : rtl ou ltr – indique le sens de lecture du document
 - ▣ xml:lang : indique le langage utilisé dans le document.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>... </head>
```

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

```
</html>
```


Balise head

- Informations sur le document :
 - Titre, auteur, description, mots clefs
 - Inclusion d'autres fichiers (javascript, feuilles de style, ...)

- `<title>un titre</title>`
 - ▣ Titre du document
 - ▣ Apparaît notamment dans la barre supérieur du navigateur.
 - ▣ Utilisé par les moteurs de recherche.
 - ▣ Obligatoire !

```
<head>
  <title>Ma page web</title>
  ...
</head>
```

Balise head – les metas

- Informations supplémentaires sur la page
 - ▣ Propriété du document / non affichée.
 - ▣ `<meta name="propriété" content="valeur">`
 - ▣ Content est obligatoire
 - ▣ Valeurs possibles
 - ▣ author, description, keywords, generator, revised
 - ▣ Autres valeurs possibles, potentiellement utilisées mais pas normalisées :
 - ▣ Robots : indexer la page ? suivre les liens ?

```
<head>
  <title>Ma page web</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta name="author" content="moi" />
  <meta name="description" content="cours web" />
  <meta name="mise a jour" content="tous les ans" />
  <meta name="robots" content="index, follow" />
</head>
```

Balise head – les link

- Créer un lien avec un autre document :
 - Surtout utilisé pour inclure une feuille de style.
 - En général on utilise :
 - href : localisation du fichier
 - rel : définition du type de lien défini – stylesheet, home, ...
 - Pas forcément utilisé par les navigateurs
 - type : type (MIME) du fichier
 - media : indique si l'inclusion doit être faite pour certains media

```
<head>
  <title>Ma page web</title>
  <link rel="stylesheet" type="text/css" href="x.css" />
</head>
```

Balise head – les scripts

- Définition d'un script exécuté en local :
 - ▣ type : définition du type de script (obligatoire)
 - ▣ src : url du script si défini dans un fichier externe
 - ▣ Attention à fermer avec `</script>`

```
<head>
  <title>Ma page web</title>
  <script type="text/javascript">
    alert("Bonjour !")
  </script>
  <script type="text/javascript" src="menu.js"></script>
</head>
```

Corps des documents HTML

□ `<body attr1="val1" ... attrn="valn">`

▣ Attributs possibles :

- bgcolor : couleur de fond.
- text : couleur du texte.
- background : URL de l'image de fond d'écran.
- link : couleur des liens non encore visités.
- vlink : couleur des liens visités.
- ...

▣ Normalement on utilise les CSS, mais en attendant...

Attributs

- Les différentes balises peuvent avoir :
 - ▣ Un identifiant : `id="monid"`
 - ▣ Un nom : `name="monnom"`
 - Majoritairement pour les formulaires
 - ▣ Une ou plusieurs classes : `class="classe1 classe2"`.
- Un seul ID par fichier HTML :
 - ▣ Permet d'identifier un objet en particulier
 - Par exemple le menu.
- Plusieurs objets peuvent avoir la même classe :
 - ▣ Utilisées pour dire que plusieurs objets sont similaires.
 - Principalement utilisé pour le rendu visuel (CSS).

Autres attributs classiques

- title :
 - ▣ Donne un titre à la balise concernée.
 - ▣ Peut-être utilisé ou pas.
- style :
 - ▣ Donne un style à la balise concernée.
 - ▣ Plus utilisé, on fait des feuilles de style à la place.
- xml:lang :
 - ▣ Langue du contenu de la balise.
- dir :
 - ▣ Sens dans lequel la balise va être lue.

Les types de balises

- Deux types de balise :
 - ▣ Bloc (peut contenir des balises blocs et en-ligne, sauf exception)
 - Exemples : div, p, h1, h2, ul, table
 - ▣ En-ligne (ne contient que des balises en-ligne, sauf exception) :
 - Exemples : span, em, strong, img, br, input
 - ▣ En HTML5 c'est moins simple
- Principe de base :
 - ▣ Les blocs sont les uns aux dessus des autres.
 - ▣ Les éléments en-ligne sont les uns à coté des autres.
 - ▣ Les éléments sont placés les uns à la suite des autres dans la page (le flux).
- On peut modifier le type des balises.

BALISES EXISTANTES

Balises classiques

□ Titres :

- ▣ `<h1>Titre</h1>`
- ▣ `<h2>` à `<h6>` pour les niveaux inférieurs
- ▣ Attribut possible `align="left | center | right"`

□ Paragraphes :

- ▣ `<p>contenu du paragraphe</p>`

□ Autres :

- ▣ `
` passage à la ligne
- ▣ `<hr />` trait horizontal
- ▣ `` = gras, `<i>` = italique, `<u>` = souligné, ...

Liens hypertextes

- Portion de texte permettant d'atteindre un document désigné par une URL
 - ▣ `texte du lien`
- URL absolue
 - ▣ `LIP6`
- URL relative `sommaire`
 - ▣ `plan d'accès`
 - ▣ `accueil`
- Désignation de signets à l'aide du caractère # dans les URLs
 - ▣ `conclusion du rapport`
 - ▣ `conclusion du rapport`
 - ▣ Crée un lien vers la balise dont l'id est conclusion dans le fichier mentionné.

Images

- Insertion d'une image dans un document :

``

- Attributs possibles

- `height="99" / width` : hauteur/largeur en pixels (ou en pourcentage) de l'image

- `alt=" ... "` : légende associée à l'image, utilisé en remplacement de l'image si elle n'est pas affichée.

- `hspace="99" / vspace` : espacement autour de l'image

- `align="bottom | middle | top | left | right"` : alignement de l'image / au texte

- Le navigateur redimensionne l'image si les tailles indiquées ne correspondent pas.

Listes

- Listes numérotées () ou non (), et définitions (<dl><dt><dd>)
- Attributs possibles
 - ▣ <ul type="disc|circle|square"> : type de puce
 - ▣ <ol type="1|i|i|A|a"> : type de numérotation
 - ▣ start="99"> : départ de la numérotation
 - ▣ <li type="disc|circle|square"> : type de puce (liste ul)
 - ▣ <li type="1|i|i|A|a"> : type de numérotation (liste ol)

```
<ul>
  <li>rouge</li>
  <li>vert</li>
</ul>

<dl>
  <dt>Titre</dt>
  <dd>définition</dd>
</dl>
```

Listes

□ Imbrication possible des listes :

```
<ol start="3">
  <li type="1"> élément 1,1 </li>
  <li> élément 1,2 </li>
  <li><ul type="a">
    <li type="square"> élément 2,1 </li>
    <li type="circle"> élément 2,2 </li>
  </ul></li>
  <li>élément 1,3 </li>
</ol>
```

Tableaux

□ Tableaux à 2 dimensions (<table> ... </table>)

```
<table>                                <!-- balise de début de tableau -->
<tr>                                    <!-- balise de début de ligne -->
  <td> cellule </td>                    <!-- balise de cellule -->
  ...
</tr>                                    <!-- balise de fin de ligne -->
<tr>
  <td>
  ...
</tr>
</table>                                <!-- balise de fin de tableau -->
```

Tableaux

- Possibilité de laisser des cellules vides
 - ▣ A ne pas encourager.

```
<table border="1">
<tr>
  <td> case 1,1 </td>
</tr>
<tr>
  <td> case 2,1 </td>
  <td> case 2,2 </td>
</tr>
<tr>
  <td></td>
  <td> case 3,2 </td>
</tr>
</table>
```


Tableaux

- `border="99"` :
 - ▣ Epaisseur en pixels des bordures du tableaux.
- `align="left | right"` (obsolète)
 - ▣ Alignement du tableau par rapport au texte.
- `width="99 | 99%"` :
 - ▣ Largeur du tableau en pixels ou en % de la largeur de la fenêtre.
- `cellspacing="99"` :
 - ▣ Espacement en pixels entre les cellules.
- `cellpadding="99"` :
 - ▣ Marge intérieure en pixels des cellules.

Tableaux

- Attributs possibles pour `<tr>`
 - ▣ `align="left | right | center | justify"` : alignement horizontal du texte (\neq `<table>`)
 - ▣ `valign="middle | top | bottom"` : alignement vertical du texte
- Attributs possibles pour `<td>`
 - ▣ `align/valign` : idem `<tr>`, supplante `<tr align/valign >`
 - ▣ `width="99 | 99%"` : largeur de la colonne en pixels | en % largeur tableau (obsolète)
- Balise `<caption>` légende associée au tableau
 - ▣ `align="top | bottom"` : placement de la légende

Tableaux - remarque finale

- Les tableaux servent à :
 - Représenter des données tabulées.

- Ils ne devraient pas servir à :
 - Faire de la mise en page de données non tabulées.
 - Par exemple pour créer un menu il faut plutôt utiliser une liste d'objets ``

- En pratique on se sert trop des tableaux !
 - Pas durant le module

Blocs génériques

- Pour agencer des éléments on utilise les blocs génériques :
 - ▣ `<div>` : balise de type bloc générique
 - ▣ `` : balise de type en-ligne générique
- Uniquement de la structuration

```
<ul id="menu">
...
</ul>
<div id="main">
  <p>Ceci est très <span class="important">important</span></p>
</div>
<div id="footer">
...
</div>
```

Autres balises

- A tester dans un navigateur pour voir le rendu :
 - ▣ En général on utilise plutôt des CSS.

```
<em>Texte mis en avant</em><br />
<strong>Texte important</strong><br />
<dfn>Définition</dfn><br />
<code>Code source</code><br />
<samp>Exemple de code source</samp><br />
<kbd>Texte clavier</kbd><br />
<var>Variable</var><br />
<cite>Citation</cite><br />
<sub>Indice</sub><br />
<sup>Exposant</sup>
```

Autres balises

- ❑ Balises supprimées/déconseillées dans HTML 5 :
 - ❑ Mise en forme pure.
 - ❑ Éviter de les utiliser même si ça fonctionne.

```
<tt>texte teletype</tt>  
<i>texte en italique</i>  
<big>Texte en gros</big>  
  
<b>Texte en gras</b>  
<small>Texte en petit</small>
```

FORMULAIRES

Formulaires

- HTML 1.0 essentiellement "mono"-directionnel :
 - ▣ Informations fournies par le serveur (suite à une commande client)
- Utilisation professionnelle
 - ▣ Nécessité de flux d'information bi-directionnels (client ↔ serveur)
 - ▣ HTML 2.0 introduit les formulaires.
 - ▣ Permettent aux clients de saisir des informations à envoyer au serveur.
- Un formulaire est défini en HTML et peut contenir :
 - ▣ Zones de saisie de texte.
 - ▣ Boîtes à cocher.
 - ▣ Boutons radio.
 - ▣ Menus déroulants.
 - ▣ Boutons.

Formulaires

- Déclaration d'un formulaire :
 - ▣ Balises `<form>` et `</form>`
- Attributs principaux
 - ▣ `<form action="..." method="..." ... >`
 - ▣ `action` : url vers laquelle envoyer les données saisies.
 - ▣ `method` : commande http (get ou post) à utiliser pour effectuer l'envoi.
- Toutes les balises HTML sont permises entre `<form>` `</form>`
 - ▣ Images, tableaux, ... peuvent être inclus dans un formulaire.
 - ▣ Les formulaires peuvent être imbriqués.

Formulaire – champs input

- Déclaration des champs de saisie
 - ▣ exclusivement entre `<form>` `</form>`
- `name` = nom du champ de saisie (unique dans un formulaire)
 - ▣ Permet de récupérer les valeurs soumises
- `type` = type du champ de saisie :
 - ▣ `text` : zone de saisie texte (type par défaut en cas d'omission de type)
 - `size` : taille apparente `maxlength` : taille max
 - ▣ `radio` : bouton radio
 - tous les boutons ayant même nom (`name`) ∈ au même groupe
 - dans ce cas, les attributs (`value`) permettent de les différencier
 - ▣ `checkbox` : boîte à cocher
 - ▣ `submit` : bouton d'envoi des données au serveur
 - ▣ `reset` : bouton d'effacement du formulaire

Exemple

...

```
<form action="prog.php" method="get">
  <p>nom <input name="client" type="text" size="4" /></p>
  <p>rue <input name="rue" type="text" size="40" /></p>
  <p>ville <input name="ville" type="text" size="20" /></p>
  <p>code postal <input name="cp" type="text" size="5" /></p>
  <p>carte de crédit no <input name="carte" type="text" size="10" /></p>
  <p>expire <input name="expire" type="text" size="4" /></p>
  <p>m/c <input name="cc" type="radio" value="mc" checked="checked" />
    visa <input name="cc" type="radio" value="vis" /></p>
  <p>contre remboursement <input name="cr" type="checkbox" /></p>
  <p><input type="submit" value="envoi" /></p>
  <p><input type="reset" value="remise à zéro" /></p>
</form>
```

...

Labels

- Intitulé associé à un champ de formulaire
 - ▣ Le clic sur l'intitulé donne la main sur le champ

```
<label for="id">Label :</label>  
<input id="id" type="text" />
```

```
<label for="id">Label :  
  <input id="id" type="text" />  
</label>
```

Envoi des données au serveur

- Lorsque l'utilisateur appuie sur le bouton SUBMIT, le navigateur :
 - ▣ Construit une chaîne de caractères contenant toutes les données du formulaire.
 - ▣ Envoie cette chaîne au serveur.

- Chaîne :
 - ▣ Ensemble de couples séparées par le caractère &
 - ▣ Chaque couple est de la forme nom de champ = valeur saisie
 - ▣ Les espaces sont remplacés par le caractère + (ou %20)
 - ▣ Les caractères + & = sont encodés %2B %26 %3D

- Exemple :
 - ▣ `client=Jean+Vier&ville=Paris&cp=75001&carte=0123456789&cc=vis&cr=on`
 - ▣ Pour les boîtes à cocher : « on » si cochée, rien sinon

Conclusion

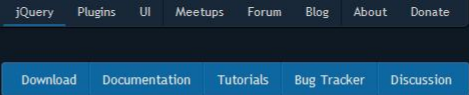
- De nombreuses balises pour faire ce qu'on veut :
 - ▣ Utiliser les balises dans le bon objectif
 - Une table et une liste n'ont pas le même sens.
- La présentation se fait ensuite avec les feuilles de style.
- Ecrire du code valide :
 - ▣ Doc : <http://www.w3schools.com/tags/default.asp>
 - ▣ Validateur : <http://validator.w3.org/>

Et HTML 5 ?

- En cours de normalisation, support variable selon les navigateurs :
 - <http://caniuse.com/>
 - Plutôt tester sur Chrome ou Opéra
- Nouveautés :
 - Code light / moins strict
 - Nouvelles balises sémantiques / multimédia / dessin
 - Formulaire améliorés
 - Drag and drop et édition de contenu
 - Notifications
 - Géolocalisation
 - Création de site hors ligne (avec base de données)
 - ...

CSS





14 Days of jQuery *jQuery*
January 14th through 28th, 2010

<http://jquery14.com>

- ✓ Lightweight Footprint
- ✓ CSS3 Compliant
- ✓ Cross-browser

Download(*jQuery*);

Google DELL NBC CBS NETFLIX Technorati mozilla.org

RUN CODE

- ◆ [Mailing List](#)
- ◆ [Source code / SVN](#)
- ◆ [Plugin Authoring](#)
- ◆ [Submit a New Bug Report](#)

BOOKS ABOUT JQUERY

version sans CSS



- [jQuery](#)
- [Plugins](#)
- [UI](#)
- [Meetups](#)
- [Forum](#)
- [Blog](#)
- [About](#)
- [Donate](#)
- [Download](#)
- [Documentation](#)
- [Tutorials](#)
- [Bug Tracker](#)
- [Discussion](#)

New Year, New Release!

14 Days of jQuery *jQuery*

January 14th through 28th, 2010

Celebrating
jQuery 1.4

<http://jquery14.com>

jQuery is a new kind of JavaScript Library.

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. **jQuery is designed to change the way that you write JavaScript.**

- [Lightweight Footprint](#)
- [CSS3 Compliant](#)
- [Cross-browser](#)

Grab the latest version!

Choose your compression level:

☒ [jquery-1.4.2.min.js](#) Production (24KB, Minified and Gzipped) ☐ [jquery-1.4.2.js](#) Development (155KB, Uncompressed Code)

[Download](#)

Current Release: v1.4.2

Who's using jQuery?

Cascading Style Sheets

- Objectifs :
 - ▣ Définir la présentation de plusieurs pages.
 - ▣ Dissocier contenu et présentation
 - C'est loin d'être toujours possible simplement

```
<head>
<style>
  h1 {color: blue;}
</style>
</head>
<body>
  <h1>titre</h1>
</body>
```

Cascading Style Sheets

- Inclusion de fichier css externe :
 - ▣ Même fichier pour plusieurs pages
 - ▣ `<link rel="stylesheet" href="...">`
 - ▣ Méthode à privilégier

```
<head>  
  <link rel="stylesheet" href="f.css">  
</head>  
<body>  
  <h1>titre en ... </h1>  
</body>
```

Cascading Style Sheets

- Plusieurs feuilles de styles peuvent être chargées :
 - ▣ Plusieurs balises `<link ... >`
 - ▣ En cas de conflit sur le style d'une balise, c'est la dernière définition qui l'emporte !

```
<link rel="stylesheet" href="xx.css">
<style>
h1 {
  color: blue;
}
</style>

...
```

Définir des styles

- On peut définir des styles pour :
 - ▣ Des balises classiques
 - `<h1>`
 - ▣ Des balises avec un identifiant donné
 - `<div id="menu">`
 - ▣ Des balises d'une classe donnée
 - `<p class="first">`
 - ▣ Des balises imbriquées dans d'autres
 - `...`

Définitions de styles

- Définition pour des balises standards :
 - ▣ balise { prop1:val1; ... ; propn:valn; }

```
/* style attribué à tous les titres 1 */  
h1 {  
    text-align: center;           /* texte centré */  
    color: blue;                  /* texte en bleu */  
    font-size: 16px;              /* police de taille 16 pixels */  
}
```

Styles d'identifiants

- Un seul identifiant par document :
 - ▣ Objet uniques (en-tête, menu, bloc droit, ...)
- Style : utilisation du symbole "#"
 - ▣ #identifiant {prop1:val1; ... }

```
/* définition du style de la balise d'identifiant test */  
#test {  
    color: red;           /* texte en rouge */  
}  
  
<div id="test">test</div>
```


Styles de classes

- Plusieurs objets peuvent être de la même classe
- Style : utilisation du symbole "."
 - ▣ .classe { prop1:val1; ... }

```
/* style pour les balises de classe equation */  
.equation {  
    color: blue;          /* texte en bleu */  
}
```

```
<p class="equation"> a = a+b </p>
```

```
<span class="equation"> c = d*a </span>
```

Définition contextuelle

- Style de balises imbriquées
 - ▣ balise1 balise2 { prop1:val1; ... }
 - ▣ Style appliqué à balise2 si elle est descendante de balise1

```
/* définition du style pour les li dans des ol dans des ol */  
ol ol li {  
    list-style: upper-roman;  
}  
  
<ol>  
<li>le style ne s'applique pas</li>  
<li><ol>  
    <li>le style s'applique</li>  
    </ol>  
</li>  
</ol>
```

Combinaison de définitions

- On peut combiner autant que l'on veut :
 - Style pour les balises de base : balise { }
 - Style pour identifiants : #id { }
 - Style pour classe : .classe { }
 - Imbrication de styles : balise1 balise2 { }
- Permet d'atteindre n'importe quelle balise du document

```
h1.impair {...}  
h1.impair span#logo {...}  
ul.menu#premiermenu ul.sousmenu li { }
```

Regroupement

- Définition du même style pour plusieurs balises.
 - ▣ balise1, balise2 {...}
 - ▣ Evite la répétition, mais attention à l'excès

```
/* style classe texte */  
.texte {  
    color: blue;  
}  
/* style p d'id premier */  
p#premier {  
    color: blue;  
}
```

```
.texte, p#premier {  
    color: blue;  
}
```

Règle de la dernière définition

- En cas de conflit, la dernière définition l'emporte
 - ▣ Attention aux redéfinitions
 - ▣ Attention en cas d'inclusion de plusieurs fichiers css

```
p, h1 {  
    color: blue;  
}  
p {  
    color: red;  
}
```

Règles supplémentaires

- Importer un fichier css dans un fichier css
 - ▣ `@import "tableau.css";`
 - ▣ Avant toute autre déclaration, sinon c'est ignoré

Fonds

- Possibilité de définir :
 - ▣ Couleur de fond
 - ▣ Image de fond
 - ▣ Position de l'image de fond
 - ▣ Répétition de l'image de fond
 - ▣ Défilement de l'image de fond

Couleurs

- 2 solutions :
 - ▣ Identificateurs prédéfinis : green, yellow, purple, blue, red, ...
 - ▣ Nombre de 6 chiffres hexadécimaux (ex. : #1A7FC0) codant les intensités de rouge/vert/bleu.

- Equivalence entre les 2 :
 - ▣ blue \equiv #0000FF
 - ▣ purple \equiv #800080
 - ▣ Mais très peu de couleurs avec identificateurs.

- Utiliser des palettes de couleurs pour trouver les codes hexa.

Police de caractères

- Possibilité de choisir :
 - ▣ Type de police
 - font-family:"Times New Roman", Times, serif;
 - ▣ Style de la police
 - font-style:normal; font-style:italic; font-style:oblique;
 - ▣ Taille de la police
 - font-size:40px; font-size:2.5em;
 - ▣ Police normale ou en gras
 - font-weight: bold;
- Version courte :
 - ▣ p { font: bold italic large Palatino, serif; }

Texte

- Possibilité de choisir :
 - ▣ Couleur du texte : color
 - ▣ Alignement du texte : text-align
 - left, right, center, justify
 - ▣ Décoration : text-decoration
 - none, overline, line-through, underline, blink
 - ▣ Transformation : text-transform
 - uppercase, lowercase, capitalize
 - ▣ Indentation : text-indent

Liens

- Les liens peuvent changer d'état :
 - ▣ État de base
 - ▣ État du lien déjà visité (dans l'historique du navigateur)
 - ▣ État du lien quand la souris est dessus (survol)
 - ▣ État du lien quand on clique dessus

- Chaque état peut être stylé :
 - ▣ a:link - état normal
 - ▣ a:visited - état visité
 - ▣ a:hover - état survolé
 - ▣ a:active - état actif/cliqué

Liens

```
/* lien de base : gris non souligné */
a:link, a:visited {
    text-decoration: none;
    color: #9c9c9c;
}

/* lien survolé : noir, souligné, bordure et couleur de fond */
/* a éviter en vrai */
a:hover, a:active {
    background-color: #00F;
    text-decoration: underline;
    color: #000000;
    border: 1px solid red;
}
```

Listes

- Possibilité de choisir :
 - ▣ Type de liste selon ul ou ol : `list-style-type`
 - circle, square, upper-roman, lower-alpha
 - ▣ Image pour les puces : `list-style-image`
 - `url('fleche.gif')`
 - Ne fonctionne pas de manière identique sur tous les navigateurs
- En général on style le contenu de la liste

Type de média

- Possibilité de définir des feuilles de style pour :
 - ▣ Moniteur, imprimantes, portables, lecteurs braille ou audio...
 - ▣ Utilisation de la règle `@media {styles}`

```
@media print {body { font-size: 10pt; }}
```

```
@media screen {body{ font-size: 12px; }}
```

```
@media screen, print {body{ line-height: 1.2; }}
```

```
@media (min-width: 768px) and (max-width: 979px) {}
```

```
//CSS 3
```

HTML 5



LI385 - Nouvelles Technologies de Web

HTML5

- Evolution (compatible) de HTML4
 - ▣ Les balises HTML4 restent valables en HTML5
 - Quelques notions obsolètes
- Nouvelles fonctionnalités :
 - ▣ Du langage HTML (balises, formulaires)
 - ▣ De Javascript
 - ▣ De CSS
- Compatibilité :
 - ▣ <http://caniuse.com/>

CODE LIGHT ET NOUVELLES BALISES



Simplification du code

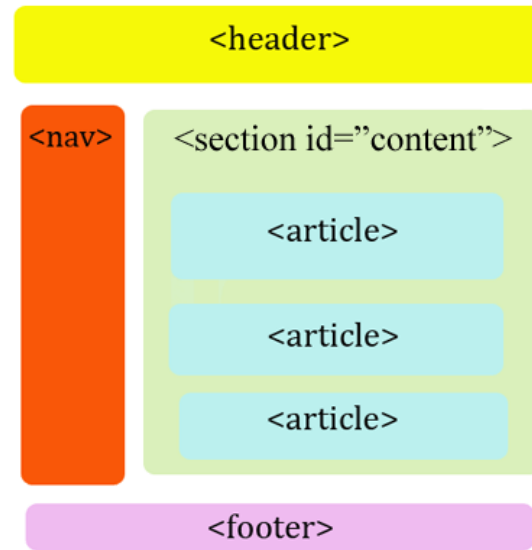
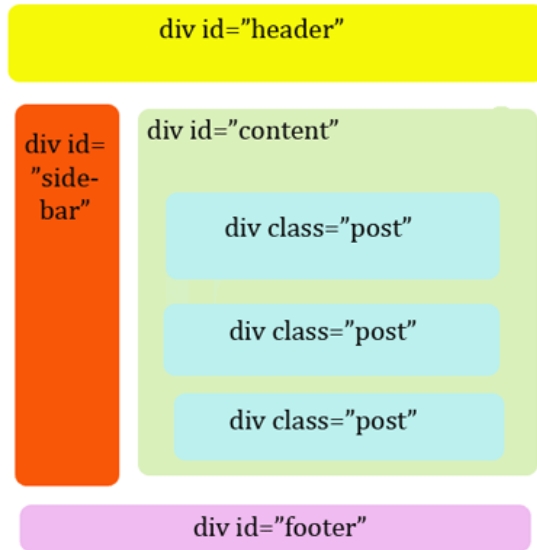
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtdhttp://www.w3.org/1999/xhtml" xml:lang="fr" >
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <link rel="stylesheet" type="text/css" href="design.css" />
    <script type="text/javascript" src="script.js"></script>
</head>
```

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="design.css" />
    <script src="script.js"></script>
</head>
```

Nouvelles balises

- Pourquoi `<div id="xxx" />` ne suffit pas ?
- Balises sémantiques :
 - ▣ `<header>` : en-tête
 - ▣ `<footer>` : pied-de-page
 - ▣ `<nav>` : navigation, type menu
 - ▣ `<aside>` : zone secondaire non liée au contenu principal de la page
 - ▣ `<article>` : portion de la page qui garde un sens même séparée de l'ensemble de la page (comme un article de blog par exemple)
 - ▣ `<figure>` : une illustration, par exemple image+légende+...

Structure globale



Balises multimédia

- Audio :
 - ▣ Lecture de fichiers audio
 - ▣ MP3, Ogg, Wav

- Video :
 - ▣ Lecture de fichiers vidéo
 - ▣ MP4, Ogg, WebM

- Canvas :
 - ▣ Dessin

- Éventuellement prévoir des fallbacks (flash)

Balises audio/video

- Attributs audio :
 - autoplay
 - controls
 - loop
 - muted
 - preload
 - src
- Attributs video supplémentaires :
 - height, width
 - poster
- Événements associables :
 - onpause, onvolumechange, onseeking, ...
 - http://www.w3schools.com/tags/ref_eventattributes.asp

FORMULAIRES

Nouveaux champs

- ❑ search : champ de recherche
- ❑ email : un ou plusieurs email
- ❑ url
- ❑ tel : pas de validation particulière
- ❑ datetime :
- ❑ date : champ de choix de date
- ❑ month
- ❑ week
- ❑ time
- ❑ datetime-local
- ❑ number
- ❑ range : choix d'un nombre
- ❑ color : color picker
- ❑ keygen : génération de clé publique

Nouveaux attributs

- Autocomplete : formulaire ou champ
 - ▣ Propose l'autocomplétion
 - ▣ `<input type="text" id="name" autocomplete="on" />`
- Novalidate : formulaire
 - ▣ La validité du champ n'est pas vérifiée.
 - ▣ `<form action="" novalidate>`
- Autofocus : champ
 - ▣ Donne le focus par défaut à l'unique champ de ce type
 - ▣ `<input type="email" id="email2" autofocus>`
- Required : champ
 - ▣ Champ obligatoire
 - ▣ `<input type="email" id="email" required>`

Nouveaux attributs

- Placeholder : input
 - ▣ Champ prérempli mais avec contenu qui disparaît
 - ▣ `<input type="email" placeholder=mail@domain.com`
- Readonly : input
- Multiple : input
 - ▣ Renseigner plusieurs valeurs (pas que pour un select)

Datalist

- Champ texte avec valeurs déroulantes prédéfinies
 - ▣ Possibilité de taper autre chose

```
<input list="students"/>
<datalist id="students">
  <option value="Laurent"/>
  <option value="Marc"/>
  <option value="Maurice"/>
</datalist>
```

Soumissions spécifiques

- Création de plusieurs boutons avec des comportements différents :
 - ▣ formaction
 - ▣ formmethod
 - ▣ formnovalidate
 - ▣ formtarget

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

```
<input type="submit" formaction="admin.php"  
value="Admin submit">
```