

SCI6052 Information documentaire numérique

# Cours 10

## Éléments du langage HTML

# Au programme aujourd'hui

---

- ➔ Contexte général de création d'un site Web & cas particulier du TP6
- ➔ Éléments du langage HTML5 vus au cours 10 (et potentiellement 11)
  - Principes de base
    - Balises et éléments
    - Attributs
    - Traitement des espaces et de la casse
  - Composantes d'un document HTML5
    - Titrage
    - Contenu structuré (*partie 1*)
    - *Exercices, démonstration de Notepad++*
    - Validation HTML
    - Contenu structuré (*partie 2*)
    - Structure globale
    - Éléments charpente
    - Regroupements structurels

Cours 10 – Objectifs visés, matériel associé et évaluation

# Examen final



## **OG1 Comprendre les principaux volets de la gestion d'information documentaire numérique**

- OS d) Expliciter les avantages et désavantages de la structuration de l'information dans différents contextes*
- OS i) Expliquer les avantages et les désavantages des sites Web statiques pour la gestion d'information documentaire numérique*

## **OG2 Mettre sur pied des systèmes de gestion d'information documentaire numérique au moyen d'outils représentatifs de la réalité des milieux**

- OS g) Concevoir un site Web de base en HTML et utilisant des feuilles de style CSS*

Compétences à développer (ensemble des acétates, exercices 1 et 2, exercices complémentaires sur StudiUM, tp6) :

- Connaître la syntaxe de base d'un élément HTML (balise d'ouverture, de fermeture, attribut)
- Comprendre la structure logique de base d'une page HTML
- Connaître les éléments présents dans l'entête (title, meta, link)
- Savoir utiliser les principaux éléments HTML (contenu structuré)
- Comprendre la différence entre une adresse absolue et une adresse relative
- Pouvoir « construire » l'adresse absolue et relative d'un fichier sur un serveur Web
- Expliquer l'importance de la validation des pages Web

## Cours 10 – Objectifs visés et activités associées

# Mise en application



### **OG1 Comprendre les principaux volets de la gestion d'information documentaire numérique**

- OS d) *Expliciter les avantages et désavantages de la structuration de l'information dans différents contextes*
- OS i) *Expliquer les avantages et les désavantages des sites Web statiques pour la gestion d'information documentaire numérique*

### **OG2 Mettre sur pied des systèmes de gestion d'information documentaire numérique au moyen d'outils représentatifs de la réalité des milieux**

- OS g) *Concevoir un site Web de base en HTML et utilisant des feuilles de style CSS*

#### Activités :

- TP6
- Exercices 1 et 2
- Exercices complémentaires sur StudiUM (dictées trouées HTML)
- Glossaire des balises HTML

# Contexte général de création d'un site Web

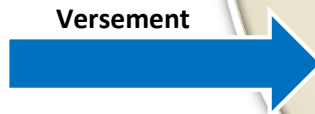
Fichiers images  
Fichiers HTML  
Etc.  
Fichiers sonores

**Site Web**

Création  
Mise à jour



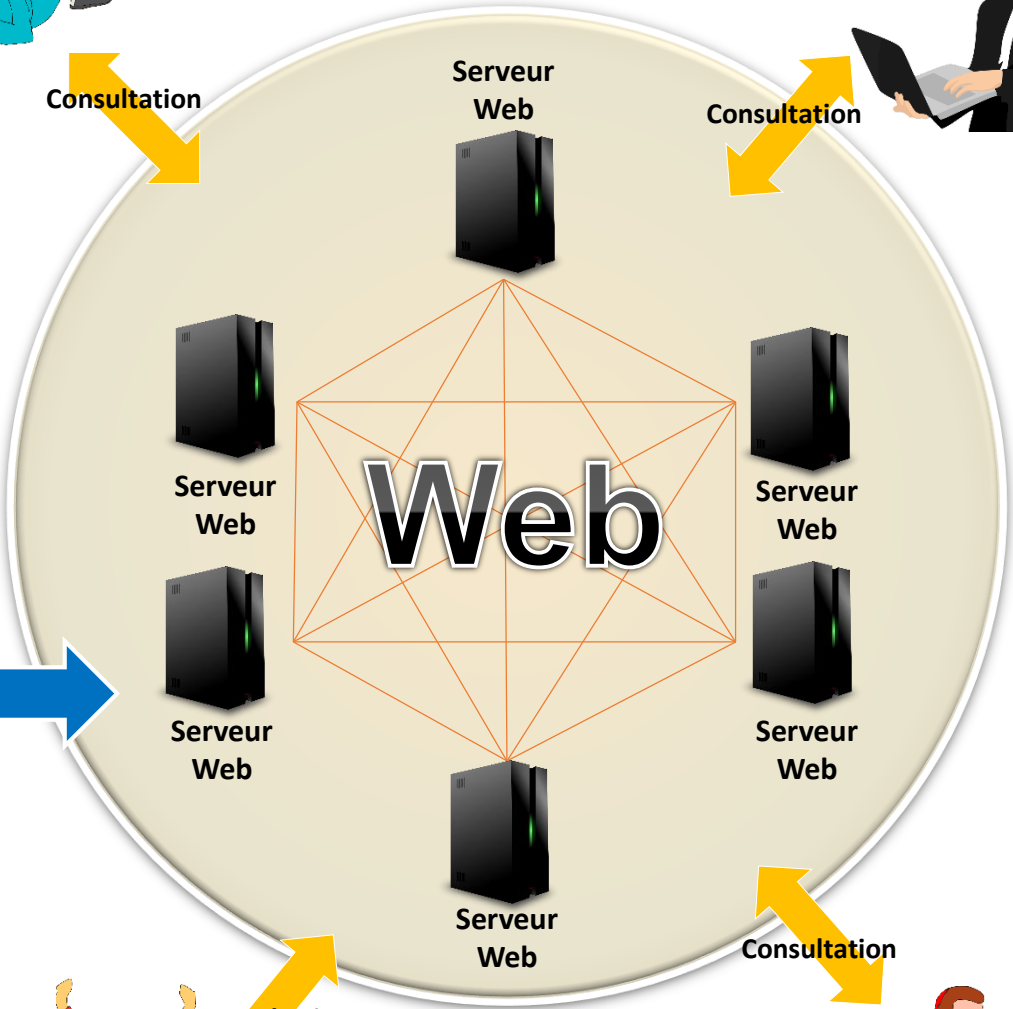
Poste de travail



Consultation



Consultation



Consultation



Consultation

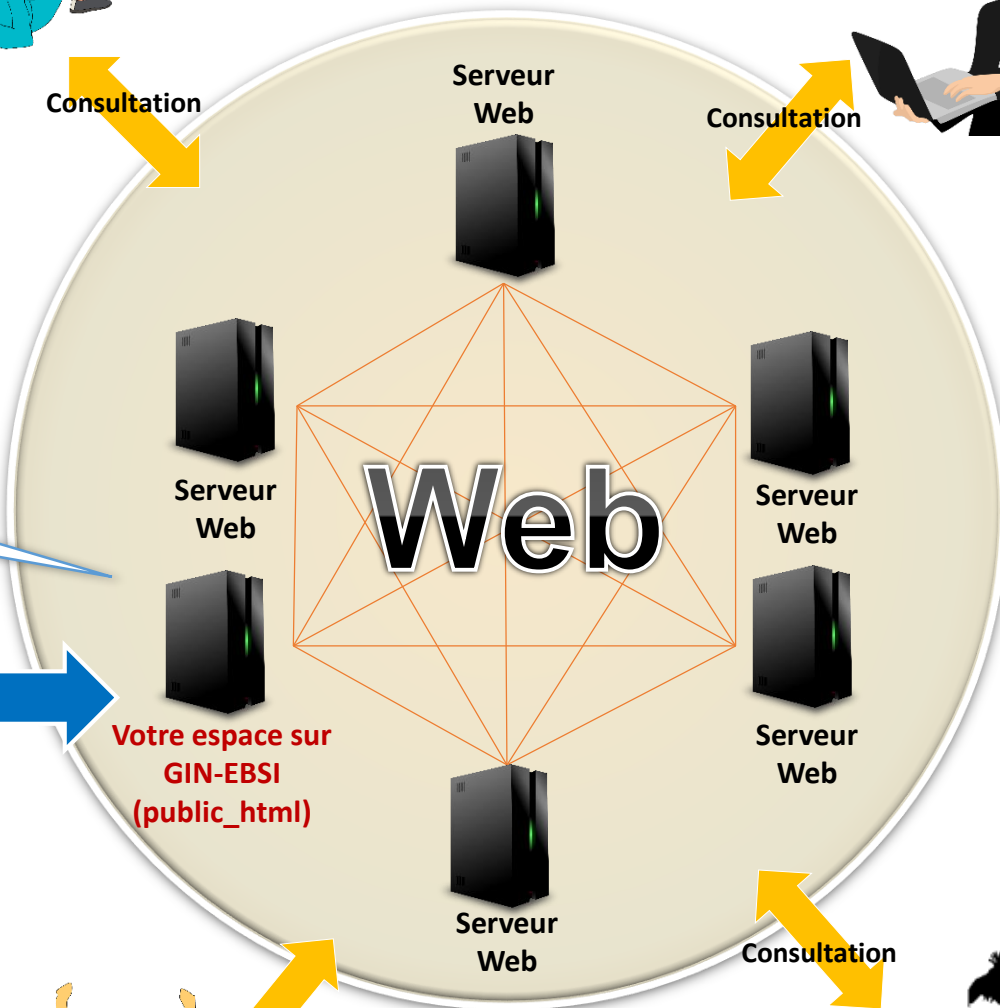
# Contexte particulier du TP6 [1/2]



Consultation

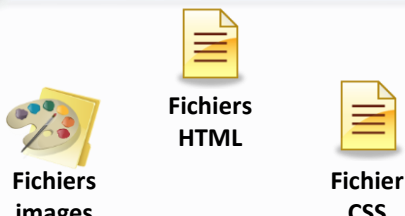
Consultation

Au C2027-35-43: Lecteur réseau (R:)  
De la maison : via WebDavSSL ou  
VPN + connexion lecteur réseau  
(voir <http://www.gin-ebis.umontreal.ca/>)



Versement

Votre espace sur  
GIN-EBIS  
(public\_html)



Votre site Web

Création  
Mise à jour



Consultation

Consultation



**Vous**  
(C2027-35-43, maison, bureau)

# Contexte particulier du TP6 [2/2]

---

## ➔ But et objectifs du TP6

- Vous aider à comprendre les principes de la création d'un site Web en
  - Mettant en application les notions fondamentales du langage HTML et des feuilles de styles CSS
  - Vous familiarisant avec l'utilisation d'un éditeur HTML par la réalisation d'un petit site Web

## ➔ Tâche

- Développer les pages en HTML5 et la feuille de styles CSS3 pour répliquer l'exemple fourni
  - Pages HTML et feuille de styles doivent être valides

## ➔ *TP non évalué*

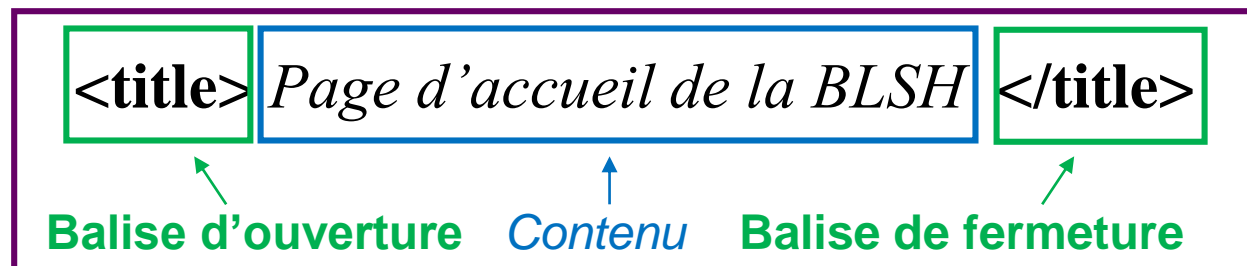
- Acquisition des connaissances reliées évaluée à l'examen final

Principes de base HTML

## Balises et éléments [1/3]

*Rappel* : HTML est un **format de documents structurés**; il en possède donc les caractéristiques

- ➔ **Balises (étiquettes, *tags*)** : délimitent les éléments pour indiquer la nature de l'information balisée et faire ressortir la structure logique
  - Nom de la balise en minuscule\* et encadré par < et >
- ➔ Présence d'une balise d'ouverture et d'une balise de fermeture\* (dans la majorité des cas)



\* HTML5 permet une syntaxe plus laxiste que d'autres formats de balisage pour les pages Web comme XHTML qui adoptent la syntaxe plus stricte XML. En HTML5, par exemple, les balises peuvent être écrites en majuscule. **Toutefois, nous privilégierons dans le SCI6052 une syntaxe stricte même pour HTML5.** Une astérisque (\*) indiquera les éléments de syntaxe plus stricts.

## Principes de base HTML

# Balises et éléments [2/3]

---

### ➔ Cas particulier des **éléments vides**

› **Éléments vides** = Éléments sans contenu (c'est-à-dire sans rien à « encadrer » par les balises d'ouverture et de fermeture)

- Exemples d'éléments vides : **br** (retour de ligne), **hr** (ligne horizontale), **img** (image), **meta** (metadonnées)

› Syntaxe : trois options

1. **juxtaposer** la balise d'ouverture et la balise de fermeture\*. Par ex. :

Les formats de documents structurés : `<br></br>` de SGML à HTML

2. utiliser une **balise d'élément vide** qui est une contraction des balises d'ouverture et de fermeture d'un élément\*. Par ex. :

Les formats de documents structurés : `<br />` de SGML à HTML

*Note : cette option sera privilégiée dans le matériel du cours*

3. utiliser uniquement la **balise d'ouverture**. Par ex. :

Les formats de documents structurés : `<br>` de SGML à HTML

## Principes de base HTML

## Balises et éléments [3/3]

---

### ➔ Possibilité d'imbriquer des éléments...

`<p>`Cet événement marqua le début de la  
`<strong>`Première guerre mondiale`</strong>` qui allait  
durer plus de quatre ans.`</p>`

### ➔ ... mais un élément **ne peut pas** inclure n'importe quel autre élément

`<strong>`Cet événement marqua le début de la  
`<p>`Première guerre mondiale`</p>` qui allait durer plus de  
quatre ans.`</strong>`

Non  
valide!

## Principes de base HTML

# Attributs

---

- ➔ Certains éléments peuvent avoir des **attributs** (parfois obligatoires, d'autres fois facultatifs) dans leur balise d'ouverture, afin d'en préciser certaines caractéristiques

```
<a href="http://www.w3c.org">Page d'accueil du W3C</a>
```

```
<h2 id="cours2">Typologie des logiciels généraux</h2>
```

- ➔ **Syntaxe** de la balise d'ouverture ou de la balise d'élément vide avec un attribut
  - > **nom de l'élément** (a et h2)
  - > **nom de l'attribut** (href et id) suivi du signe « = »
  - > **valeur de l'attribut** (http://www.w3c.org, cours2) entre guillemets\*
- ➔ Il est possible d'avoir **plus** d'un attribut dans une balise d'ouverture
  - > Couples *attribut-valeur* séparés par une espace

```
      attribut 1                attribut 2
      ↙                       ↘
<meta name="author" content="John Smith" />
```

## Principes de base HTML

# Traitement des espaces et de la casse

---

- ➔ En XML et en XHTML, la **casse** est prise en compte
  - En XHTML, les éléments et les attributs doivent toujours être en minuscules. Par ex. :
    - XHTML valide : `<h1>La musique disco du moyen âge à aujourd'hui</h1>`
    - XHTML non valide : `<H1>La musique disco du moyen âge à aujourd'hui</H1>`
- ➔ En HTML5, les éléments et les attributs peuvent être en majuscules ou en minuscules. Toutefois, il est conseillé de conserver la bonne pratique établie de XHTML de les garder en minuscules pour faciliter, au besoin, le passage à XHTML
- ➔ Les espaces, tabulations et sauts de ligne entre les éléments balisés sont ignorés par les navigateurs

Éléments d'un document HTML

# Titrage

---

➔ **<h1> à <h6>** : titres de section et de sous-section

- Le chiffre indique la structure **hiérarchique** des titres, h1 étant le plus haut niveau, h6 le plus petit
- Logiquement, on ne doit retrouver qu'un seul titre de niveau 1 à l'intérieur d'une zone (par exemple, à l'intérieur d'un élément `<article>` ou à l'intérieur d'un élément `<header>`)

Un glossaire interactif des éléments HTML présentés dans le cadre du cours est accessible à l'URL <http://cours.ebsi.umontreal.ca/glossaireweb/index.php?cours=sci6052> et doit être utilisé en complément des notes de cours

## Éléments d'un document HTML

# Contenu structuré [1/2]

---

- ➔ Éléments qui représentent différents types de contenu
- ➔ Certains servent à définir la **structure logique** du contenu du document (=niveau bloc)
  - Conceptuellement bien séparés de ce qui les précède et de ce qui les suit
    - Lors de la restitution (affichage), un saut de ligne est généralement ajouté avant et après l'élément balisé
  - Par exemple
    - **<p>**, **<address>**, **<blockquote>**
    - Éléments pour les *listes*
      - **<ul>**, **<ol>**, **<li>**, **<dl>**, **<dt>**, **<dd>**
    - Éléments pour définir des *tableaux* (vus plus tard)
      - **<table>**, **<caption>**, **<th>**, **<tr>**, **<td>**

## Éléments d'un document HTML

# Contenu structuré [2/2]

---

- ➔ Certains servent à **qualifier le texte** à l'intérieur d'un élément de niveau bloc ou à **insérer du contenu multimédia** (par ex. image) (=niveau texte ou en ligne)
- ➔ Lors de la restitution (affichage), aucun retour de ligne n'est généralement ajouté, ni avant ni après l'élément
- ➔ Utilisables entre autres partout où on peut retrouver du contenu textuel
- ➔ Par exemple :
  - <strong>**, **<em>**, **<img>**, **<cite>**
  - <a>** (vu plus tard)

## Avant d'aller plus loin...

---

- ➔ Exercice de balisage
- ➔ Exercice de validation
- ➔ Démonstration de Notepad++

## Conception des pages Web

# Validation HTML

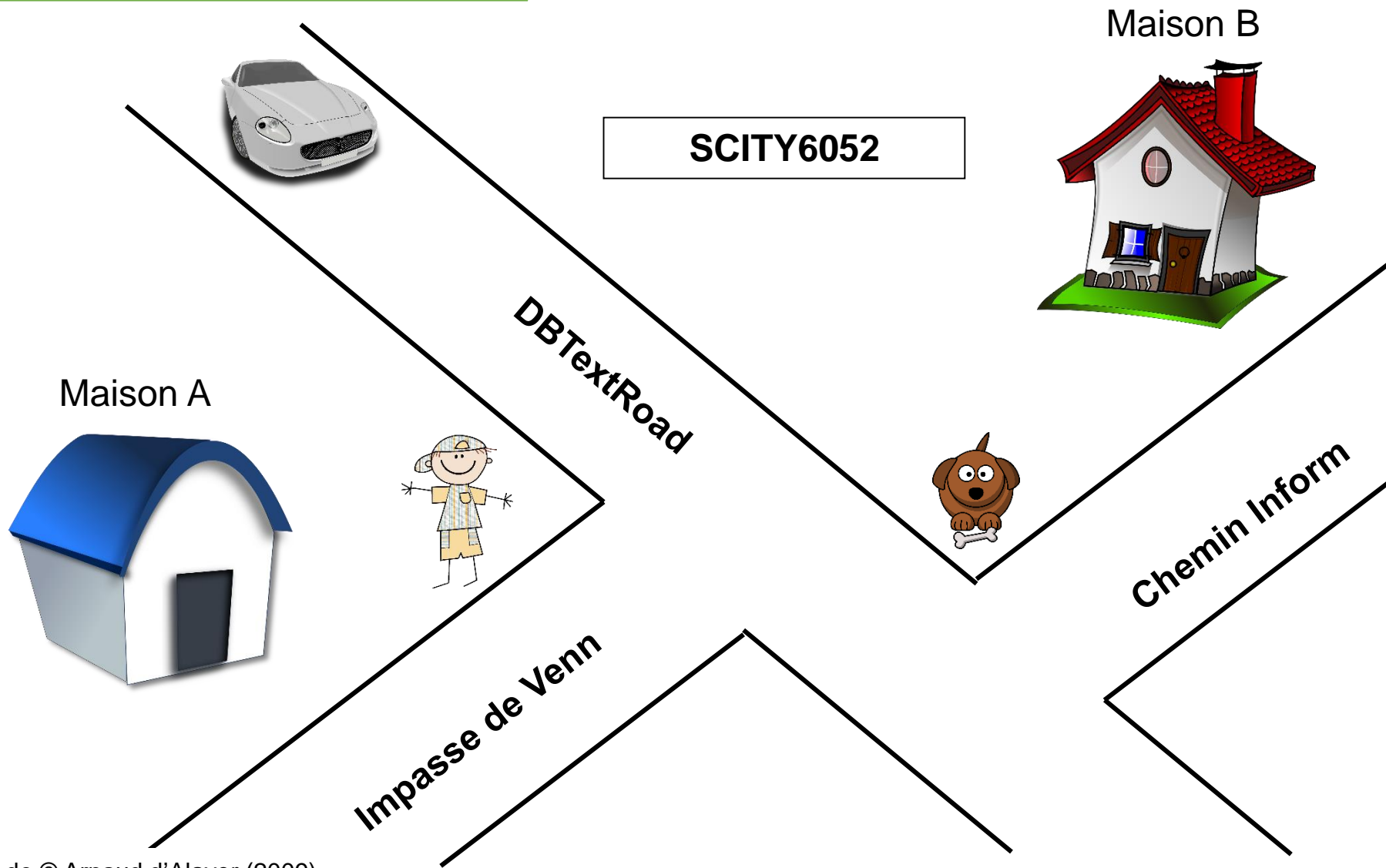
---

- ➔ Pour s'assurer de **respecter les normes** et donc de bénéficier de **tous** les avantages d'utiliser une norme
- ➔ Validation *via* le **logiciel d'édition de pages Web**
  - › Possible pour certains logiciels
  - › Pas toujours efficace à 100%
- ➔ Valideur du **W3C** (*html5 sans validation xml*) : <http://validator.w3.org/>
- ➔ Valideur du service **validator.nu** (*html5 avec validation xml*)\* : <https://validator.nu>
- ➔ Processus recommandé
  - › Valider **au fur et à mesure** de la construction des pages
  - › **Alterner** entre validation et corrections
    - Une erreur peut avoir des incidences multiples

\* Ce valideur, en ajoutant l'attribut `xmlns="http://www.w3.org/1999/xhtml"` dans la balise `html` d'une page, permet de valider les fichiers selon les critères plus stricts de codage présentés en classe. C'est le valideur qui sera utilisé dans le cadre du TP6

# Contenu structuré : <a> (liens hypertextes) [1/14]

Adresse absolue vs Adresse relative



Éléments d'un document HTML

## Contenu structuré : `<a>` [2/14]

---

### Adresse absolue = Adresse complète

➔ **Maison A**

10 Impasse de Venn  
H0H0H0  
SCITY6052  
CANADA



➔ **Maison B**

30 chemin Inform  
H0H0H0  
SCITY6052  
CANADA



Ex. : `<a href="http://cours.ebsi.umontreal.ca/sci6052/index.htm">site SCI6052</a>`

Éléments d'un document HTML

## Contenu structuré : <a> [3/14]

---

➔ Composantes d'une adresse URL absolue :

**protocole://serveur[:port]/[chemin/]fichier[#position]**

- › *Protocole* : par exemple http, ftp, telnet
- › *Serveur* : adresse physique ou logique de l'ordinateur
- › [*Port* : numéro du port où le serveur est en attente (80 par défaut si non spécifié pour le protocole HTTP)] (*facultatif*)
- › [*Chemin* : suite de dossiers pour se rendre au document] (*facultatif*)
- › *Fichier* : nom du document
  - *Facultatif* pour un lien vers une page Web nommée index.htm ou index.html
- › [*Position* : position dans le document] (*facultatif*)

## Éléments d'un document HTML

# Contenu structuré : `<a>` [4/14]

---

➔ Pour déterminer l'adresse URL absolue d'un fichier déposé sur un serveur [1/3]

➤ Si je verse, à partir des laboratoires de l'EBSI, un fichier intitulé `menu.html` dans mon espace sur le serveur gin-ebsi à la racine du dossier `public_html`, celui-ci se retrouvera à l'endroit suivant :

- `R:\public_html\menu.html`

➤ On pourra accéder à ce fichier sur le Web en utilisant l'adresse URL absolue suivante :

**`http://www.gin-ebsi.umontreal.ca/p123456/menu.html`**

- où `p123456` représente votre code d'utilisateur DGTIC

- l'URL **`http://www.gin-ebsi.umontreal.ca/p123456/`** pointe vers le dossier `public_html` dans votre espace sur gin-ebsi; c'est ce que l'on appelle l'URL de base d'un site Web

## Éléments d'un document HTML

# Contenu structuré : `<a>` [5/14]

---

- ➔ Pour déterminer **l'adresse URL absolue** d'un fichier déposé sur un serveur [2/3]
  - Si je verse, à partir des laboratoires de l'EBSI, un fichier d'image `logo.gif` dans le dossier `images` créé dans mon espace sur le serveur gin-ebis dans le dossier `public_html`, celui-ci se retrouvera à l'endroit suivant :
    - `R:\public_html\images\logo.gif`
  - On pourra accéder à ce fichier sur le Web en utilisant l'adresse URL absolue suivante :  
**`http://www.gin-ebis.umontreal.ca/p123456/images/logo.gif`**
    - où `p123456` représente votre code d'utilisateur DGTIC

## Éléments d'un document HTML

# Contenu structuré : <a> [6/14]

---

➔ Pour déterminer l'**adresse URL absolue** d'un fichier déposé sur un serveur [3/3]

➤ En d'autres mots, l'adresse URL absolue d'un fichier se construit en **ajoutant** le **chemin d'accès du fichier** à partir du dossier **public\_html** (en remplaçant les « \ » par des « / ») au bout de l'**URL de base de votre espace Web** sur le serveur (<http://www.gin-ebsi.umontreal.ca/p123456/>)

### ➤ Exemples additionnels

- **R:\public\_html\recettes\tarte.html**
  - <http://www.gin-ebsi.umontreal.ca/p123456/recettes/tarte.html>
  
- **R:\public\_html\musique\fete.mp3**
  - <http://www.gin-ebsi.umontreal.ca/p123456/musique/fete.mp3>

Éléments d'un document HTML

## Contenu structuré : `<a>` [7/14]

---

### ➔ Adresse URL absolue : en résumé

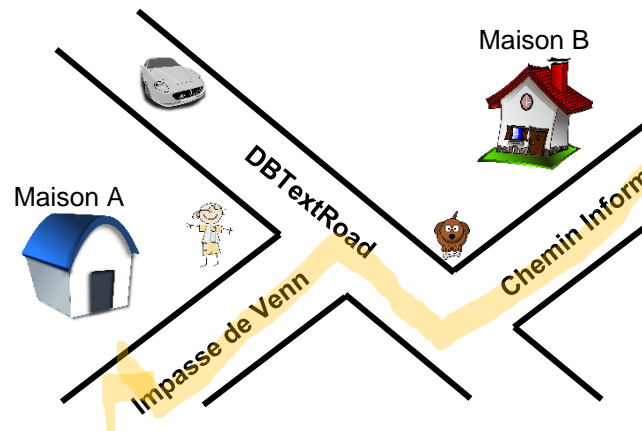
- Les adresses URL absolues donnent le **chemin d'accès complet** au document, incluant le nom du serveur, le chemin et le nom du fichier (à noter que le nom du fichier peut être omis s'il s'agit de `index.html` ou `index.htm` qui seront repêchés par défaut si rien n'est spécifié)
- **Problème des adresses URL absolues** : si l'adresse URL de base du site change (par exemple parce qu'on a changé de serveur), les URL utilisées pour pointer vers d'autres fichiers du site ne seront plus bonnes
- **Solution** : utiliser des adresses URL **relatives**!

Éléments d'un document HTML

## Contenu structuré : `<a>` [8/14]

**Adresse relative = adresse par rapport à un point de référence**

- ➔ Chemin vers la Maison A à partir de la Maison B
  - > À droite sur « Chemin Inform »
  - > À droite sur « DBTextRoad »
  - > À gauche sur « Impasse de Venn »
  - > Première maison  
à droite (no 10)



Éléments d'un document HTML

## Contenu structuré : `<a>` [9/14]

---

- ➔ Les adresses URL relatives donnent le chemin d'accès au fichier **par rapport à l'endroit** où se trouve le fichier qui contient le lien (fichier source)
  - › Exemples d'adresses URL relatives
    - **index.html**
    - **images/logo.gif**
    - **../recettes/tarte.html**
  - › **Avantage** : si vous changez votre site Web de serveur mais que la structure demeure la même, les liens entre les pages du site demeurent ainsi fonctionnels

## Éléments d'un document HTML

# Contenu structuré : <a> [10/14]

- ➔ Pour déterminer l'**adresse URL relative** d'un fichier, on spécifie le chemin d'accès du fichier en fonction de l'emplacement du fichier source, en omettant l'adresse URL de base de l'espace Web sur le serveur

Par exemple, pour le  
fichier source

**tarte.html**

public\_html



Même logique que les chemins sur un disque dur vus en début de session

## 3 cas possibles

1. Pour pointer vers un fichier situé *dans le même dossier* (par ex.: `gateau.html`) on donne seulement le **nom du fichier** :  
`<a href="gateau.html">Recette de gâteau blanc</a>`
2. Pour pointer vers un fichier situé dans un *dossier parent* (par ex. : `index.html`), on donne le **nom du fichier, précédé de « ../ »** :  
`<a href="../index.html">Retour à la page d'accueil</a>`
3. Pour pointer vers un fichier situé dans un *dossier placé dans le même dossier* (par ex.: `ingredients.jpg`), on donne le **nom du dossier suivi de « / » et du nom du fichier** :  
``

Éléments d'un document HTML

## Contenu structuré : `<a>` [11/14]

---

### ➔ Adresses URL relatives vs absolues

#### > Quand utiliser une **adresse URL absolue**?

- Quand le fichier n'appartient pas à notre site

Consultez le site `<a href="http://www.recettes.com/tarte_sucre.htm">`Mes meilleures recettes`</a>` pour une autre version de la recette traditionnelle de tarte au sucre.

- Quand le protocole est différent (rare)

#### > Quand utiliser une **adresse URL relative**?

- Le reste du temps... c'est-à-dire pour pointer vers une page Web qui appartient à notre site Web
- Facilite le déménagement d'un site Web

*Note* : si vous avez de la difficulté à construire des URLs absolues ou relatives, retournez faire les exercices sur StudiUM sur les arborescences de dossiers. Il s'agit en effet de la même logique ici qui s'applique.

## Éléments d'un document HTML

# Contenu structuré : `<a>` [12/14]

---

➔ Pour créer un **lien interne** (c'est-à-dire un lien à l'intérieur d'une même page Web)

1. On localise d'abord l'endroit auquel on veut que le lien mène (la destination du lien ou ancre). Dans l'exemple précédent :

```
<h2>Section 3</h2>
```

2. On « nomme » l'élément où l'on veut se rendre avec un identifiant unique par l'utilisation d'un attribut *id* ajouté à sa balise d'ouverture

```
<h2 id="sect3">Section 3</h2>
```

3. On ajoute un élément `<a>` à l'endroit d'où part le lien qui aura comment valeur pour l'attribut **href** la valeur de l'attribut *id* de l'ancre, précédée de « # ». Dans l'exemple précédent

```
<p>Consultez la <a href="#sect3">Section 3</a> du présent document.</p>
```

## Éléments d'un document HTML

# Contenu structuré : <a> [13/14]

---

### ➔ Autre usage de l'attribut **id**

- Une fois qu'on a placé un attribut **id** dans un document HTML, la valeur de cet attribut (précédée de « # ») peut être ajoutée au bout de l'adresse URL du document pour pointer à ce point précis dans le document, plutôt qu'au début du document

`<p> Voir la sect3 Section 3 du rapport annuel </a> . </p>`

- Ce lien mènera directement à l'élément portant l'attribut **id="sect3"** du document auquel on réfère, et non au début du document

Éléments d'un document HTML

## Contenu structuré : `<a>` [14/14]

---

### ➔ Hyperliens vers une adresse de courrier électronique

- › l'attribut **href** est utilisé pour spécifier l'adresse à laquelle on veut qu'un courriel soit envoyé
- › l'adresse doit être précédée de **mailto:**

```
<a href="mailto:john.smith@gmail.com">John Smith</a>
```

## Éléments d'un document HTML

# Contenu structuré : Tableaux

---

- ➔ Pour représenter de l'information « tabulée »
  - **À ne pas utiliser pour de la mise en page** (par ex. une présentation en colonne)
- ➔ On doit utiliser un grand nombre d'éléments, imbriqués les uns dans les autres, pour définir un tableau
  - **<table>** : tableau (balise de plus haut niveau)
  - **<caption>** : titre du tableau
  - **<tr>** : rangée du tableau (*table row*)
  - **<td>** : cellule simple (*table data*)
  - **<th>** : cellule d'entête (*table header cell*)
- ➔ Un tableau se construit par ligne et, ensuite, par cellules (qui correspondront aux colonnes). Par ex., pour un tableau ayant une ligne et deux colonnes

```
<table>
```

```
<tr>
```

```
<td>contenu de la cellule</td>
```

```
<td>contenu de la cellule</td>
```

```
</tr>
```

```
</table>
```

## Éléments d'un document HTML

# Entités [1/2]

---

- ⇒ Pour coder
  - › les **caractères spéciaux ou réservés** (par ex. ©, &)
  - › les **caractères accentués** (*facultatif*) si on veut que le fichier soit codé selon le jeu de caractères ASCII pur
    - De moins en moins utilisés
- ⇒ Chaque entité est précédée de **&** et suivie de **;**
- ⇒ Exemples
  - › **&copy;** pour le symbole ©
  - › **&amp;** pour le symbole &

## Éléments d'un document HTML

# Entités [2/2]

---

➔ Entité **&nbsp;** (*non-breaking space*) = espace insécable

➤ Espace insécable = espace qui ne peut éventuellement être remplacée par un saut de ligne. Par ex. :

La journée «**&nbsp;**Portes ouvertes**&nbsp;**» aura lieu le 12 novembre 2017.

Ne pourra pas s'afficher de la façon suivante :

La journée « Portes ouvertes  
» aura lieu le 12 novembre 2017.

Comme les espaces entourant **Portes ouvertes** sont insécables

## Éléments d'un document HTML

# Commentaires

---

⇒ **Annotations ignorées par le navigateur** lors de la visualisation incluses par exemple pour documenter la page HTML

‣ Précédées de `<!--` et suivi de `-->`. Par ex. :

`<!-- Début du pied de page -->`

⇒ Peuvent également être utilisés pour du texte qu'on ne désire pas afficher immédiatement. Par ex. :

`<!-- <h2>Matériel du cours 11</h2> -->`

‣ Au moment où on souhaite afficher l'information, il faut simplement retirer les caractères indiquant qu'il s'agit d'un commentaire. Par ex. :

`<h2>Matériel du cours 11</h2>`

# Structure d'un document HTML

## Structure logique globale [1/3]

```
<!DOCTYPE>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
<meta />
```

```
<link />
```

```
</head>
```

```
<body>
```

```
<header>[titrage/contenu structuré]</header>
```

```
<section>
```

```
<article>[titrage/contenu structuré]</article>
```

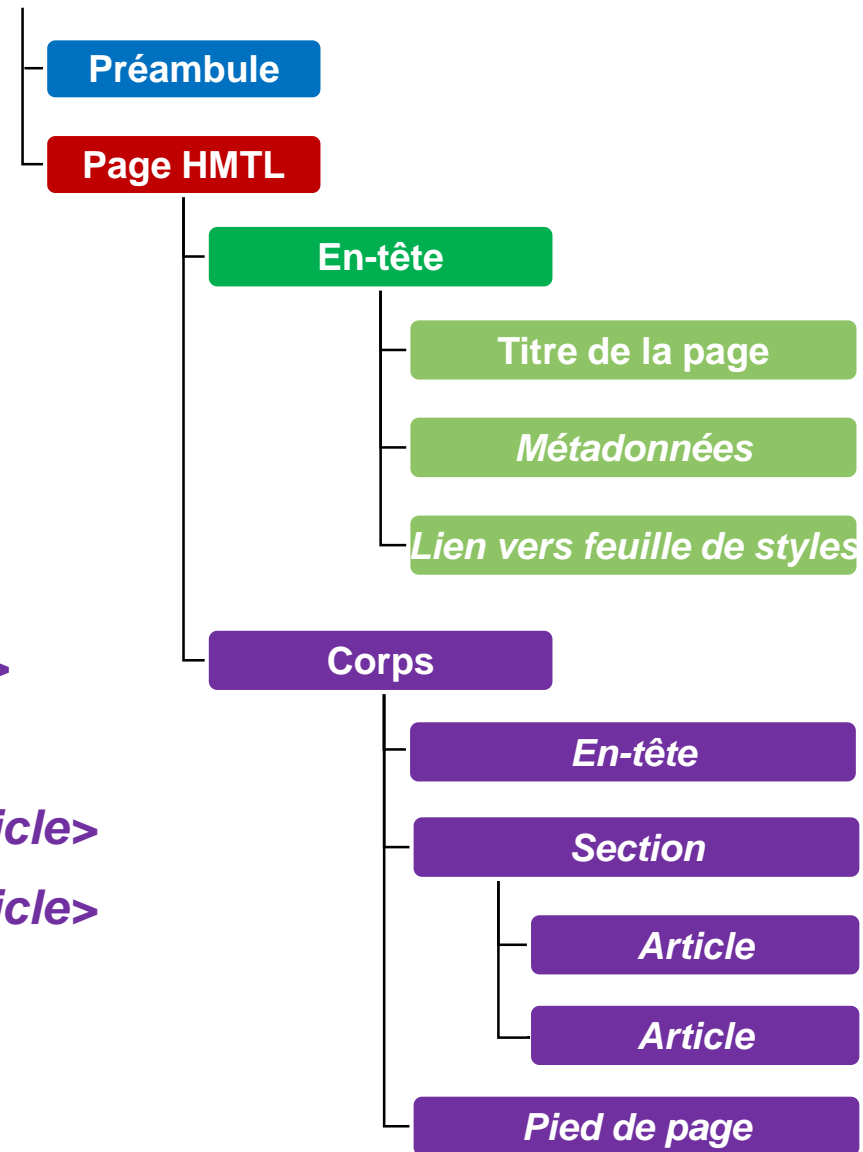
```
<article>[titrage/contenu structuré]</article>
```

```
</section>
```

```
<footer>[titrage/contenu structuré]</footer>
```

```
</body>
```

```
</html>
```



*Italique = facultatif*

# Structure logique globale [2/3]

```
<!DOCTYPE html>
```

Préambule

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr-ca">
```

Élément  
HTML

```
<head>
  <title>Module de formation continue</title>
</head>
```

Entête

```
<body>
  <header><h1>Modules de formation continue</h1></header>
  <section>
    <article>
      <p><strong>Responsable :</strong> John Smith</p>
      <p><strong>Sujet :</strong> Macramé</p>
    </article>
    <article>
      <p><strong>Responsable :</strong> Marie Poppins</p>
      <p><strong>Sujet :</strong> L'art du biscuit</p>
    </article>
  </section>
  <footer><p>Mise à jour le 13 novembre 2013</p></footer>
</body>
```

Corps

```
</html>
```

Structure d'un document HTML

## Structure logique globale [3/3]

---

### ➔ **Grandes catégories d'éléments et exemples**

- **Éléments charpente** : `<html>`, `<head>`, `<body>`
- **Métadonnées** (*dans `<head>`*) : `<title>`, `<meta>`, `<link>`
- **Regroupements structurels** (*dans `<body>`*) : `<header>`, `<footer>`, `<section>`, `<article>`, `<nav>`
- **Titrage** (*dans `<body>`, dans les regroupements structurels s'il y a lieu*) : `<h1>` à `<h6>` (*vu précédemment*)
- **Contenu structuré** (*dans `<body>`, dans les regroupements structurels s'il y a lieu*) : `<p>`, `<strong>`, etc. (*vu précédemment*)

## Structure d'un document HTML

# Éléments charpente : `<html>`

---

➔ L'élément `<html>` doit chapeauter tout le document

➤ Ex. de balise d'ouverture pour l'élément `<html>` :

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr-ca">
```

- L'attribut « `xml:lang` » est *facultatif* mais recommandé. Il indique dans quelle langue est le document.

- Langue identifiée par son code selon la spécification RFC3066 :  
<http://www.ietf.org/rfc/rfc3066.txt>

- <http://www.i18nguy.com/unicode/language-identifiers.html> (Using Language Identifiers)

- L'attribut « `xmlns` » est facultatif; il permettra, lors de la validation, de s'assurer de respecter le balisage plus strict privilégié

Structure d'un document HTML

## Éléments charpente : `<head>` [1/2]

---

- ➔ L'élément `<head>` peut contenir (entre autres) les éléments suivants (métadonnées)
  - › L'élément `<title>` (**obligatoire**)
  - › Un ou des éléments `<link>` (*facultatif*)
  - › Un ou des éléments `<meta>` (*facultatif*)
  
- › *Attention : Les éléments `<link>` et `<meta>` sont facultatifs dans le format HTML mais **obligatoires** pour le TP6!*

Exemple d'une entête complète pour une page Web comportant **3 métadonnées** (2 auteurs et une description) et un lien vers une feuille de styles.

Structure d'un document HTML

## Éléments charpente : <head> [2/2]

Élément <title> (obligatoire)

<head>

Métadonnées

```
<title>Site Web de John Smith et Bill Brown</title>
```

```
<meta name="author" content="John Smith" />
```

```
<meta name="author" content="Bill Brown" />
```

```
<meta name="description" content="Site Web  
professionnel de John Smith et Bill Brown,  
consultants en information." />
```

```
<link rel="stylesheet" type="text/css"  
href="css/styles.css" />
```

</head>

Lien vers la feuille de styles

Structure d'un document HTML

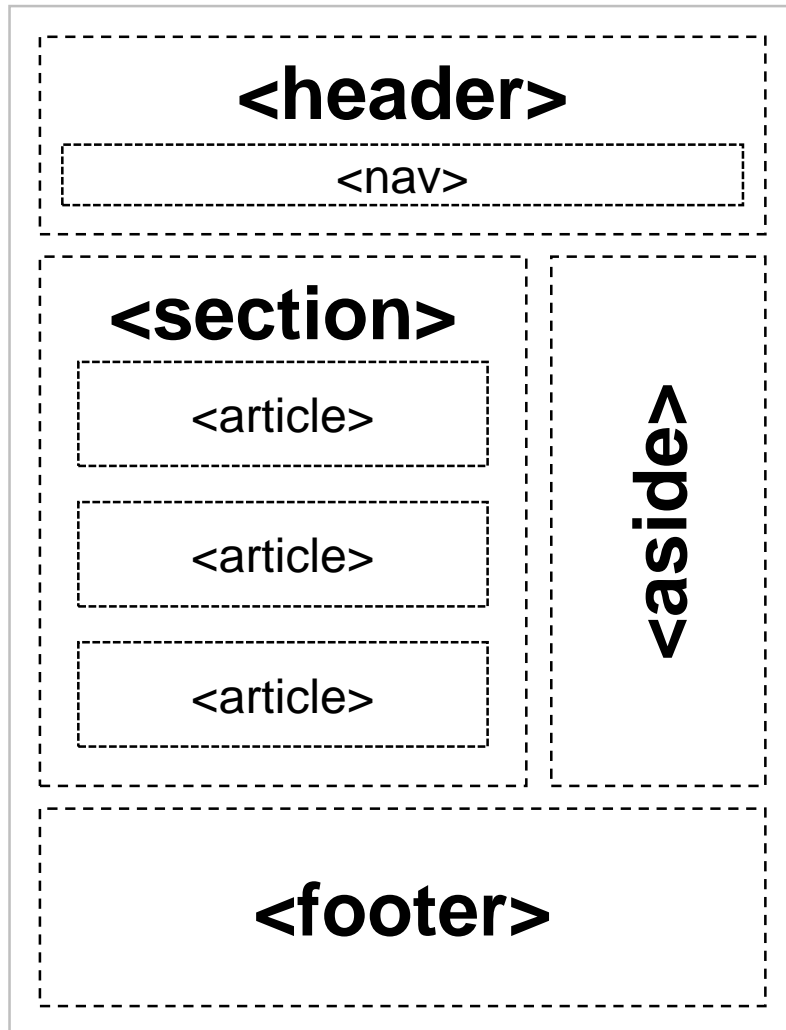
## Éléments charpente : `<body>`

---

- ➔ L'élément `<body>` inclut le **corps du document**, c'est-à-dire tout ce qui est visible dans la fenêtre du navigateur
- ➔ Inclut différents types d'éléments représentant
  - › Regroupements structurels
  - › Titrage (*vu précédemment*)
  - › Contenu structuré (*vus précédemment*)

Structure d'un document HTML

# Regroupements structurels



= zones possibles  
(*facultatives*) à  
l'intérieur du corps du  
document

Source : Rimelé, Rodolphe. 2011. HTML5 :  
Une référence pour le développeur Web.  
Paris: Eyrolles. Page 92

## Conception de pages Web

# Pour en savoir plus ...

---

### ➔ Normes du W3C

- › HTML5 : Recommendation <<https://html.spec.whatwg.org/multipage/>> (consultée le 4 novembre 2019)
- › Feuilles de style en cascade, niveau 3 <<http://www.w3.org/TR/2001/WD-css3-roadmap-20010523/>> (consultée le 4 novembre 2019)

### ➔ Autres sites Web

- › *Top Ten Mistakes in Web Design* de Jakob Nielsen (1<sup>er</sup> janvier 2011). <<http://www.nngroup.com/articles/top-10-mistakes-web-design/>> (consultée le 4 novembre 2019)
- › *Top Ten Enduring Web-Design Mistakes* de Jakob Nielsen (30 octobre 2016). <<https://www.nngroup.com/articles/top-10-enduring/>> (consultée le 4 novembre 2019)
- › *Apprenez à créer votre site web avec HTML5 et CSS3* de Mathieu Nebra (24 octobre 2019). <<https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3>> (consultée le 4 novembre 2019)