RÉSUMÉ DU COURS DE HTML ET CSS

— Architectures web —

URL : Uniform Resource Locator

`protocole://machine/repertoire/fichier#fragment`

ex : `http://limsi.fr/individu/enseignement.html#pw`

Client : ordinateur utilisé (souvent via le navigateur web).

Serveur : ordinateur qui stocke et fournit le site web.

Site statique (HTML) : contenu fixe (*pas d'interaction*).

Site dynamique (PHP) : contenu variable (*interaction*).

Le client envoie une *requête* (demande de page).

Le serveur *génère* la page.

Le serveur envoie une *réponse* (renvoie la page).

Client : présentation, interactions utilisateur (*HTML*, *CSS*, *JavaScript*).

Application : traitement des données, validation des requêtes (*PHP*).

Ressource : stockage, accès aux données (*PostgreSQL*).

— HTML —

HTML concerne la **structure** et le **contenu** de la page web.

Permet de créer des documents reconnus et mis en forme par tous les navigateurs.

```
<balise [attributs]>contenu</balise>
<balise [attributs] />
```

> *Les deux balises ont le même nom, sensible à la casse (toujours en minuscule).*
>
> *Les balises doivent être **fermées** et l'**imbrication** doit être correcte.*
>
> *Les valeurs d'attributs sont entre guillemets.*

Un exemple de page HTML basique :

```
<!DOCTYPE html>
<!-- Commentaire -->
<html>
    <head>
        <meta charset="UTF-8" />
        <title>Le titre de la page</title>
    </head>
    <body>
        <h1 class="attribut">Un titre</h1>
        <p>Un paragraphe</p>
    </body>
</html>
```

Pour faire valider une page HTML : http://validator.w3.org/

— Balises HTML —

Mise en forme physique *OBSOLÈTE* **:** `<b>` (gras), `<i>` (italique), `<pre>` (texte préformaté) ...

Mise en forme logique : `<strong>` (mise en valeur), `<em>` (fait ressortir), `<code>` (encadre du code) ...

Éléments non sémantiques : pas d'information sur le contenu (*`div`* , *`span`* ...)

Éléments sémantiques : définit le contenu (*`table`* , *`form`* , *`img`* ...)

`<span>` : définit une partie de texte

`<div>` : définit une section d'un document

`<h1>, ..., <h6>` : titre de niveau n

`<p>` : paragraphe (saut de ligne avant et après)

`<ul>` , `<ol>` : liste d'items (*ordonnée pour ol*)

`<li>` : élément d'une liste d'items

`<table>` , `<tr>` , `<th>` , `<td>` : tableau avec lignes et cellules

`<hr/>` : insère une ligne horizontale

`<br/>` : passage à la ligne

`<a href="url">texte</a>` : lien hypertexte (vers un document externe/page du site)

`<img src="url" alt="si image non affichée" />` : affiche une image

`<header>` : définit l'en-tête

`<nav>` : définit la zone de navigation

`<section>` : définit une section

`<aside>` : définit une zone de contenu à part

`<article>` : définit un article

`<footer>` : définit le pied

`<audio>` : affichage d'éléments audio sur une page

`<video>` : affiche d'éléments vidéo sur une page

— CSS —

CSS permet la mise en forme du contenu de la page web.

Cette norme n'est pas supportée de façon uniforme sur tous les navigateurs ...

Bonne pratique : mise en forme hors de la mise en page.

```
h1 {
    color: blue;
    font-size: 12px;
}
```

> `h1` *est le sélecteur (élément auquel le style s'applique).*
>
> `color: blue;` *est une déclaration.*
>
> `color` *est une propriété.*
>
> `blue` *est une valeur.*

`<p id="unique">` : **id** définit un élément unique dans le document

`<p class="groupe">` : **class** définit un groupe d'éléments

`#id` pour sélectionner l'élément portant cet *id*.

`.class` pour sélectionner les éléments portant cette *classe*.

Pour **importer une feuille de style** dans son document, dans l'en-tête (`<head>`) :

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

`span` : permet d'appliquer un style à une partie de texte (*inline*)

`div` : permet d'appliquer un style à un bloc de texte (*block*)

Pseudo-classe : définir un état particulier d'un élément (`selector:pseudo-class`)

ex : `a:hover` *(passage de la souris sur le lien),* `a:visited` *...*

Pseudo-élément : définir le style de certaines parties d'un élément (`selector::pseudo-element`)

ex : `p::first-line` *(première ligne),* `h1::before`

Responsive Web Design : pages web adaptées à tout type d'écran (*media queries*).