

TP n°1 – environnement de travail, programmation XHTML

Partie 1 – environnement de travail

En vous connectant sur les ordinateurs de l'Université, vous accédez à un ensemble de ressources logicielles et matérielles. Certaines d'entre elles sont locales au réseau universitaire (les fichiers et dossiers de votre espace personnel, le système et les logiciels installés sur les ordinateurs), d'autres sont en ligne sur le Web (votre espace numérique de travail, l'espace pédagogique et toutes les ressources du Web). Cette partie vise à vous fournir quelques informations sur l'utilisation de votre environnement de travail.

Remarque : votre environnement de travail Linux sur le réseau universitaire, ce qui inclut les logiciels et vos documents, est accessible depuis chez vous en utilisant le logiciel « client » gratuit x2goclient. Pour plus d'informations, consulter la page suivante : <https://sif.info-ufr.univ-montp2.fr/?q=content/comment-se-connecter-%C3%A0-distance-%C3%A0-mon-bureau-gnulinux>

1.1. Organiser ses documents en interface graphique

Au fil de votre parcours, vous allez accumuler divers documents (supports de cours et de TP, projets étudiants, ...) que vous allez rapidement avoir besoin d'organiser.

Remarque : vous disposez de 2 Go de stockage sur votre espace personnel sur le réseau de l'université, ainsi que d'un espace de stockage sur l'Espace personnel de votre Espace Numérique de Travail (ENT). Pour ne pas confondre les deux, on appellera **Home** l'espace personnel présent sur le réseau de l'université.

La plupart des espaces de stockage utilisent un système d'organisation par fichiers et dossiers.

Un **fichier informatique** est un ensemble d'informations situé sur un espace de stockage ; son nom est souvent constitué de deux parties séparées par un point. La deuxième partie, appelée **extension** du fichier, donne une information sur la nature du fichier :

txt, doc, odt pour les fichiers texte,
mp4, avi pour les vidéos,
jpg, tiff pour les photos...

Un **dossier informatique** ou **répertoire** est un récipient virtuel, qui peut contenir d'autres dossiers et/ou des fichiers. L'ensemble des dossiers et fichiers peut être représenté sous forme d'un arbre inversé ou **arborescence**. Le point le plus haut de l'arbre correspond à un dossier qui contient tous les autres, appelé **racine** de l'arborescence.

Lorsque vous vous connectez sur les ordinateurs de la faculté, l'écran d'accueil présente le contenu d'un dossier particulier de votre Home : le **bureau** ou **desktop**.

Exercice guidé : Double-cliquez ensuite sur l'icône Dossier personnel. Cette action ouvre un **Gestionnaire de fichiers** qui affiche le contenu de votre Home. Créer, s'il n'existe pas, le dossier public_html . Respecter les minuscules/majuscules.

Exercice : dans le dossier public_html précédemment créé, ajoutez cinq sous-dossiers nommés TP1 à TP5.

Remarque : le plus souvent, en interface graphique, les opérations de base sur les fichiers et les dossiers (ouverture, renommage, suppression etc.) sont accessibles par un clic droit sur l'icône correspondante.

1.2. Mettre ses documents à disposition de ses collègues et professeurs

Il est possible de partager des documents sur le réseau de l'université. Cela permet de ne pas saturer les boîtes de messagerie de vos interlocuteurs et de partager vos documents de manière plus ouverte.

Exercice guidé : Dans votre répertoire personnel, copier un fichier dans le répertoire public_html . Demander à votre voisin(e) de récupérer ce fichier en tapant, depuis un navigateur internet, l'adresse localhost/~votrelogin , où « votrelogin » est votre identifiant de connexion (généralement, 1ère lettre du prénom suivi du nom).

1.3. Rendre un document à son enseignant dans le cadre d'un contrôle sur ordinateur

Certains enseignants demanderont que vous rendiez vos devoirs sur une « plateforme », c'est à dire un service de gestion de documents, comme Moodle.

Exercice guidé : aller sur l'ENT puis Enseignements, Moodle, Faculté des Sciences, Informatique, HLSE305/HLSE401 (si ce n'est pas encore le cas, ajouter ce cours à votre liste), Travaux pratiques, Dépôt de test TP1. Joindre un petit fichier et envoyer la soumission.

Astuce : pour passer rapidement d'une application à l'autre, utiliser la combinaison Alt + Tab



1.4. Travailler avec ses collègues sur un même projet



Certaines applications Web vous permettent de travailler à plusieurs et en même temps sur un même document. La suite proposée par [Framasoft](#), par exemple, fournit un ensemble d'outils collaboratifs sous licence libre :

- Framapad pour prendre des notes à plusieurs
- Framadate pour organiser des rendez-vous
- Framacalc pour gérer des feuilles de calcul
- Framindmap pour structurer les compte-rendus de réunions sous forme de diagrammes
- Framavectoriel pour effectuer des figures
- et beaucoup d'autres.

Exercice : Réaliser un document collaboratif avec plusieurs collègues puis mémoriser l'adresse Web (URL) de votre document dans un fichier séparé. Un seul étudiant donne un titre, qu'il partage avec ses collègues.

1.5. Un peu de ligne de commande (si vous avez le temps)

La **ligne de commande** est une façon d'interagir avec un système informatique en tapant des instructions au clavier. Elle peut sembler austère mais elle permet de gagner un temps considérable pour des tâches complexes.

Pour entrer en mode de ligne de commande, ouvrir un **émulateur de terminal** depuis le menu des applications  , Accessoires, Terminal Xfce, ou en cliquant sur l'icône  en bas de l'écran.

1.5.1. Premières instructions : *pwd*, *tree*, *man*, *cd*, *ls*

Exercice guidé : Depuis le terminal, tapez la commande **pwd** puis valider par la touche d'entrée. Cela affiche le **chemin**, c'est à dire l'emboîtement de sous-dossier en sous-dossier, depuis le répertoire racine jusqu'à votre Home, séparés par le caractère / .

Exercice : taper **tree**. Qu'est-ce que cela affiche ? Taper ensuite puis **tree -d** et **tree -a** (pour arrêter la commande en cours, utiliser la combinaison de touches Ctrl + C) ; à quoi servent les options -d et -a ? Si vous ne trouvez pas, tapez « **man tree** » pour en savoir plus sur la commande tree (ou utilisez un moteur de recherche).

Astuce : pour recopier un texte sur Linux, il suffit de le sélectionner avec la souris pour le mémoriser, puis de cliquer avec le bouton du milieu (molette) pour le recopier.

Exercice guidé : taper `cd public_html/TP1` puis `pwd`. A quoi sert la première commande ? Taper ensuite `cd ../..` (ne pas oublier l'espace après cd) puis `pwd`. A quoi « .. » fait-il référence ?

Exercice : accéder de la même façon au dossier TP2. Revenir au home en tapant `cd ~` (~ est un raccourci désignant le Home).

Exercice : taper `ls` (avec la lettre L, pas le chiffre 1) puis `ls -a`. A quoi cela sert l'instruction ls ? A quoi sert l'option -a ?

Explication : sous Linux, les fichiers de configuration et certains fichiers système sont cachés, pour des raisons de sécurité et de lisibilité.

Remarque : dans le navigateur de fichiers, pour afficher les fichiers cachés, taper Ctrl+h.

1.5.2. Actions par lots

Jusqu'à présent, ce que nous avons fait peut être réalisé facilement en interface graphique. Une possibilité intéressante de la ligne de commande est de faire des actions par lots, comme le montre l'exemple suivant.

Exercice guidé : mkdir est une commande permettant de créer un dossier. Se placer dans le dossier Home avec la commande « cd » puis Taper ou recopier la commande suivante :

```
for lettre in {a..e}; do for annee in {2010..2015}; do mkdir
"archive-$lettre-$annee" ; done; done
```

puis afficher le contenu du dossier Home depuis un navigateur de fichier. Qu'avez-vous fait ?? Combien de temps vous aurait-il fallu pour faire la même chose en interface graphique ?

Exercice : Résoudre le problème en remplaçant, dans la ligne précédente, la commande mkdir

par celle qui permet de supprimer un dossier (utiliser un moteur de recherche).

1.5.3. Comment retrouver un fichier perdu ? Utilisation du pipe.

Vous avez égaré un fichier dans les méandres de votre arborescence. Vous savez que c'est une image .jpg mais vous en avez 300 000. Vous savez que son nom contient « voiture », ce qui réduit la liste à 1500 images, et qu'elle n'est pas dans le répertoire Vacances, ce qui réduit la liste à une seule image.

Voici comment la retrouver :

```
find ~ -name *.jpg | grep voiture | grep -v Vacances
```

Explication : on cherche (find) à partir du répertoire home (~) tous les fichiers ayant l'extension jpg (*.jpg). On obtient la liste des 300 000 images, avec leurs chemins d'accès depuis le Home. Le résultat est ensuite (|) filtré (grep) par le mot voiture, puis (|) filtré (grep) en négatif (-v) par le mot Vacances.

Dans ce contexte, le symbole | est appelé un **pipe** (prononcer à l'anglaise païpe), il permet d'enchaîner plusieurs commandes.

Exercice : quelle commande faut-il taper pour retrouver un fichier commençant par la lettre « a » et n'ayant pas la lettre « e » ? Auriez-vous su faire la même chose en interface graphique ?

Exercice : depuis votre Home, taper `ls -a` puis `ls -a | more` (q pour quitter). A quoi sert la partie qui suit le pipe ?

Partie 2 : Le langage (X)HTML

Source : <http://lemoigno.fr>

Objectif : Apprendre à créer une page Web basique

- Les balises de base du langage (X)HTML
- Insérer des images
- Insérer des liens hypertextes
- Faire des listes
- Faire des tableaux

2.1. Les bases du (X)HTML

Une **page web** est un fichier contenant du texte, généralement dans le langage **HTML** ou **XHTML**.

Remarque sur le XHTML : XHTML est la nouvelle version de HTML.

Pour voir cela, faites un clic droit sur une page web, par exemple <http://lemoigno.fr/Contrôle/TP7/bases.html> et cliquez sur Code source de la page (ou Menu Afficher → Code source de la page). Une fenêtre s'ouvre : elle contient le code de la page en langage HTML. Analysez le contenu du fichier HTML.

Le **document** est la page web telle que nous la visionnons. Il faut donc 2 outils : un outil d'**édition** qui permet d'éditer le texte du fichier et un outil de **visualisation** pour voir le document final.

Il existe différents outils d'édition comme les éditeurs de texte brut (gedit) ou des outils plus dédiés (comme geany ou KompoZer) qui permettent de taper le code HTML ou d'utiliser les assistants graphiques. Les outils de visualisation correspondent aux navigateurs tels que Firefox, Internet Explorer, ...

Récupérez le fichier pp.html sur l'espace pédagogique et copiez-le dans le répertoire TP1.

Il contient certaines des principales balises (ou marqueurs) (X)HTML ainsi qu'un faux texte destiné à remplir cette page de test. Ce fichier vous servira de canevas pour cette partie du TP.

Ouvrez le fichier enregistré dans votre répertoire avec un éditeur de texte (par exemple accesssoires → gedit) et avec le navigateur firefox : l'éditeur vous montre le contenu du fichier et le navigateur affiche le résultat de son interprétation.

Astuce : Pour plus de facilité, vous pouvez afficher les deux fenêtres en plein écran et dans deux bureaux distincts. Pour passer d'un bureau à l'autre : Ctrl + Alt + → .

Remarque : Pour voir l'effet d'une modification du texte (document ou marqueur) dans l'affichage, il faut sauver le fichier .html modifié (CTRL + s) puis rafraîchir l'affichage du navigateur avec la touche F5. De cette façon, il n'y a pas besoin de rouvrir ni l'éditeur, ni le navigateur.

2.1.1. Les principales balises du langage

Le XHTML est un langage « à **balises** ». Un fichier XHTML a pour extension .htm ou .html et est de type « texte ». Les balises ont toujours la forme : `< nom >`. On distingue deux types de balises : les balises locales, indiquant un effet à un endroit donné, qui s'écrivent `< nom / >` (par exemple, `< br / >` pour un saut de ligne) et les balises qui ont une portée et qui ont donc un début et une fin : `< nom >.....< / nom >` (par exemple, `< h1 >...< / h1 >` pour un titre de niveau 1).

Essayez maintenant de répondre aux questions suivantes en utilisant différentes options des animations:

Remarquez la casse (majuscule ou minuscule) des différentes balises (head et body, par exemple): cette casse a-t-elle une importance?

Où retrouve-t-on ce qui est inscrit entre les marqueurs `< title >` et `< / title >`?

Quel est l'effet du marqueur `< br >`?

Que trouve-t-on à l'intérieur du marqueur `< !.... >`?

A quoi correspond `& copy;`?

Enlever les marqueurs fermants `< / html >`, `< / body >` et `< / head >`: quel est l'effet de cette suppression sur l'affichage par le navigateur ? Penser à copier-coller les lignes supprimées dans un autre fichier, afin de pouvoir les remettre lorsque vous aurez terminé.

Enlever les marqueurs ouvrants correspondants aux marqueurs fermants précédents: cela change-t-il l'affichage de la page sur le navigateur?

Enlever à présent le marqueur `< / h2 >`: comment l'affichage est-il modifié? Jusqu'où s'étend cette modification?

À quelle structure correspond le marqueur `< p >`? Que remarquez-vous pour cette balise ?

Modifiez avec la souris la largeur de la fenêtre du navigateur: que constatez-vous ?

Un langage très tolérant : On constate que le langage HTML, contrairement à un langage de programmation comme le JAVA est très tolérant et qu'un navigateur peut afficher correctement un fichier html très mal écrit. Il est bien évident cependant que dans ce cas, l'affichage est aléatoire puisqu'il dépend de la tolérance de chaque navigateur. Il est donc vivement recommandé de toujours respecter au mieux la norme du langage.

2.1.2. Les attributs des balises

Certaines balises possèdent des attributs pour indiquer des paramètres supplémentaires. Les attributs ne se mettent que dans la balise d'ouverture. Ils ont un nom (en minuscule) et une valeur (entre guillemets doubles, valeur « libre »).

Exemple : ``.

Testez les attributs bgcolor et text de la balise body. Les valeurs que peuvent prendre ces deux attributs sont des couleurs écrites sous la forme #HHHHHH, où H est un nombre hexadécimal.

Des attributs dépréciés : les attributs des balises sont donnés à titre de la compatibilité avec les plus anciennes versions de HTML. Elles tombent peu à peu en désuétude et leur rôle est maintenant tenu par les feuilles de style.

2.1.3. Une apparence basée sur le contenu

Si le HTML est si tolérant, c'est qu'il n'est pas un langage de programmation mais un langage de présentation de documents. Il faut toujours avoir à l'esprit quand on écrit du HTML ou plus généralement quand on crée une page Web, que tout doit être basé sur le contenu du document que l'on veut rendre public. En effet, le créateur des documents n'a qu'une influence relative sur la façon dont ses pages seront visualisées puisque cela dépend fortement du navigateur mais aussi du type de matériel de la personne qui visualise les documents.

Pour cela, il existe différents marqueurs qui vont correspondre à une fonction particulières de certaines parties du document (titres, sous-titres, listes, mise en exergue...) et qui vont chercher à mettre en évidence ces parties, par exemple en changeant la fonte (mais pas toujours).

Pour mettre en évidence les titres de différents niveaux (titres, sous-titres,...), les marqueurs sont du type `<hi>...</hi>` où *i* est un entier entre 1 et 6: `<h1>` correspond au niveau le plus élevé (titre principal) et `<h6>` au niveau le plus bas.

`<p>....</p>` délimitent les paragraphes

`
` introduit un saut de ligne sans saut de paragraphe

`... ` sert à mettre en évidence (« un peu » en valeur)

`...` sert à mettre « beaucoup » en valeur

Il existe également des marqueurs qui ne modifient pas l'aspect du contenu sur la base de sa fonction. Ses styles physiques sont déconseillés mais parfois bien pratiques...

`<i> </i>` sert à mettre en italique

`<u> </u>` sert à souligner

` ` sert à mettre en gras

`^{....}` met en exposant

`_{....}` met en indice

Testez ces différentes balises sur la page pp.html en ajoutant les balises de titre de différents niveaux, en mettant en évidence certaines parties du texte (em, strong, gras, italiques...), en écrivant la formule chimique $[\text{NH}_4]^+$.

2.2. Insérer des images et des liens

2.2.1. Insérer des images

Un document HTML est très souvent un document multimédia, c'est à dire qu'il comprend des images et des sons en plus du texte. Cependant, le fichier HTML proprement dit est uniquement un fichier de texte mais qui peut contenir des liens vers des fichiers multimédias.

Le marqueur qui permet d'insérer les images est ****. Ce marqueur comporte un attribut obligatoire qui indique où se trouve le fichier image correspondant, c'est l'attribut **src** = "chemin vers fichier". Ainsi, on aura toujours au moins une syntaxe de ce genre:

Il existe d'autres attributs pour ce marqueur ****, qui sont eux facultatifs:

alt = "texte descriptif" : permet d'afficher un texte décrivant l'image si le navigateur n'arrive pas à l'afficher ou lorsque la souris passe sur l'image. C'est une chaîne de 1024 caractères au maximum.

align=alignement : permet de contrôler l'alignement des images par rapport au texte environnant. Les spécifications HTML définissent 5 valeurs d'alignement: left, right, top, bottom, middle. Les deux premières positionnent l'image sur la marge gauche (left) ou droite (right) du texte qui se place donc sur la partie restante des lignes pour compléter la page. Les trois autres valeurs positionnent verticalement l'image par rapport au texte: top aligne le sommet de l'image sur le plus haut élément du texte environnant, bottom aligne le bas de l'image sur le bas du texte (c'est la valeur par défaut). La valeur middle est traitée différemment par InternetExplorer et Netscape: Netscape aligne seulement le milieu de l'image avec la ligne de base du texte tandis que IE aligne le milieu de l'image avec le milieu de l'élément (lettres ou image) le plus haut de la ligne.

hspace="nombre", **vspace**="nombre" : associés à l'attribut aligne, permettent de laisser un espace libre (exprimé en pixels) entre l'image et le texte.

border="nombre" : lorsqu'une image correspond à un hyperlien (on clique dessus pour appeler un autre document, voir la section 2), elle peut être signalée par une bordure bleue (de la même façon qu'un texte associé à un hyperlien est de couleur différente). L'attribut border précise la taille de la bordure bleue en pixels.

width="nombre", **height**="nombre" : permettent de fixer la largeur (width) et la hauteur (height) de l'image, soit en pixels, soit en pourcentage de la taille de la fenêtre. On peut ainsi éviter le débordement des images trop grandes mais également gagner du temps lors du chargement: le navigateur réserve directement la place demandée pour l'image au lieu de devoir la calculer et l'adapter en fonction de l'image chargée. Attention toutefois de conserver le rapport largeur/hauteur sous peine de distorsion.

Récupérez les photos photo1.jpg et photo2.jpg et insérez-les dans la page, la première dans le premier paragraphe à droite, la deuxième dans le dernier paragraphe en bas.

2.2.2 Insérer des liens hypertextes

Les liens hypertextes sont la principale caractéristique des documents HTML et sont à l'origine de la création du World Wide Web. Ils permettent la navigation à partir du document courant, soit vers d'autres documents de tous types répartis sur le réseau, soit à l'intérieur du document lui-même. Chaque lien est défini par une adresse ou URL (Uniforme Ressource Locator). Il existe plusieurs types d'URL:

- **liens HTTP** (HyperText Transfer Protocol): ils correspondent aux adresses d'autres documents HTML sur le réseau. Ces liens s'écrivent sous la forme `http://serveur:port/chemin` avec "serveur" qui est l'adresse de la machine qui abrite le document demandé. La partie "port" est facultative : elle indique la "porte d'entrée" dans le serveur (en général 80 pour le HTML). Le reste de l'adresse ("chemin") indique la partie de l'arborescence du serveur sur laquelle se trouve la page demandée. Ce chemin peut être d'une longueur quelconque et comporter plusieurs répertoires et sous-répertoire (rep1/rep2/rep3/...). Ce chemin se termine par le nom du document. Si aucun nom de fichier n'est indiqué, le navigateur essaie d'ouvrir un fichier "index.html": s'il n'en trouve pas, il affiche la liste des fichiers du répertoire.
- **les e-mails** (courrier électronique): ces liens appellent en fait un programme de courrier électronique pour envoyer un mail vers l'adresse indiquée. Le programme de courrier est appelé par la commande: `mailto:nom@adresse`

Un lien peut également pointer vers une partie précise d'un document html, si on repère ce point précis par une ancre, qui est en fait un point de repère vers lequel on peut un lien hypertexte.

Une ancre est l'attribut `name="nom"` ajouté à une balise quelconque pour la repérer. Si on a posé une ancre nommée "chap_1" sur le titre principal d'une page :

`<h1> Grand titre </h1>`, on pourra arriver directement à ce titre par le lien `...`.

Si le lien est écrit simplement comme `...` cela signifie qu'il pointe vers l'ancre chap_1 du même fichier : c'est donc un moyen de créer des liens à l'intérieur d'une page un peu longue.

Dans le fichier pp.html, créez un lien vers la page principale de l'Université de Montpellier (<http://www.umontpellier.fr/>).

Faites en sorte que cette adresse soit un lien cliquable qui permet de vous envoyer un mail sur votre adresse institutionnelle.

Remarque : pour pouvoir envoyer vous-même un mail depuis les machines de TP, il faut configurer les paramètres de votre serveur d'envoi (smtp.umontpellier.fr). Les détails de cette configuration dépassent le cadre de ce TP.

Créer un lien depuis les images insérées dans le document vers le site Flickr (<https://www.flickr.com/>) où elles ont été récupérées (sous licence Creative Commons). Vous penserez à créer autour de l'image une bordure d'épaisseur 2 pixels.

2.3. Les tableaux et les listes

2.3.1. Les tableaux

En (X)HTML, les tableaux sont formés de lignes, contenant un certain nombres de cellules.

Un tableau sera compris entre les balises `<table></table>`.
 Chaque ligne est délimitée par `<tr></tr>` (pour Table Row).
 Chaque case est définie par `<td></td>` (pour Table Data) ou par `<th></th>` (pour Table Header) s'il s'agit d'une case de la ligne d'en-tête.
 Le contenu des cases peut être du texte ou des images.

La balise `<table>` admet un certain nombres d'attributs :

align: alignement du tableau par rapport au texte environnant. Il peut prendre les valeurs right, left ou center.
bgcolor : comme pour body, permet de donner la couleur de fond du tableau.
border= taille en pixels : épaisseur de la bordure du tableau (0 par défaut)
bordercolor=couleur : coloration de la bordure.
cellspacing=valeur en pixels : contrôle l'espace entre deux cellules adjacentes ou entre le bord du tableau et les cellules (2 pixels par défaut).
cellpadding=valeur en pixels : contrôle l'espace entre le bord d'une cellule et son contenu (1 pixel par défaut).
width=valeur en pixel ou en % : contrôle la largeur du tableau en % de la largeur de la page.

Récupérez le fichier table.html dans votre répertoire TP1. Affichez-le dans votre navigateur et ouvrez-le avec un éditeur de texte.

Faites-en sorte de donner à votre tableau un fond de couleur "wheat" avec des bordures de 3 pixels de large, de couleur gris foncé (#494949). Centrez le tableau dans la page.

Quelle est, par défaut, la largeur du tableau? Que se passe-t-il si on impose une largeur de 120% ou de 25%?

Quelles sont les deux caractéristiques du contenu des cases th?

Que se passe-t-il si on enlève les caractères dans la première case du tableau? À quoi correspondent-ils?

Les balises `<td></td>` et `<th></th>` admettent des attributs dont :

colspan=nombre de colonnes fusionnées : fusionne les colonnes.
rowspan=nombre de lignes fusionnées : fusionne les lignes

		Préférences	
		Kumquats frais	Kumquats au sirop
sexe	Hommes de moins de 18 ans	45%	55%
	Hommes de plus de 18 ans	51%	49%
	Femmes de moins de 18 ans	56%	44%
	Femmes de plus de 18 ans	63%	37%

Utilisez ces deux attributs pour donner à votre tableau une allure voisine de celle de l'image.

2.3.2. Les listes

Il y a deux types de listes en HTML: les listes ordonnées et les listes non ordonnées.

Comme leur nom l'indique, les **listes non ordonnées** regroupent des éléments qui n'ont pas besoin d'être présentés dans un ordre particulier (par exemple une liste de liens hypertextes vers des pages concernant un même sujet). Ce type de liste est compris entre les marqueurs `` et ``. Chaque élément de la liste est signalé par le marqueur `` et se termine par ``.
Remarque : Le marqueur `` accepte l'attribut **type**="type de puce" qui permet de choisir le motif devant chaque article de la liste entre disc, circle et square.

Par exemple

```
<ul>
  <li> <a href="url1"> adresse1 </li>
  <li> <a href="url2"> adresse2 </li>
  <li> <a href="url3"> adresse3 </li>
</ul>
```

donnera le résultat suivant:

- [adresse1](#)
- [adresse2](#)
- [adresse3](#)

Les **listes ordonnées** sont par exemple utilisées pour faire un index des chapitres d'un document, chaque élément de liste étant un lien vers un chapitre, dans l'ordre du texte bien entendu. Elles sont signalées par les balises `` et ``. Les éléments de listes sont, comme pour les listes non ordonnées signalés par ``.

Par exemple,

```
<ol>
  <li> Chapitre 1 </li>
  <li> Chapitre 2 </li>
  <li> Chapitre 3 </li>
</ol>
```

produira comme résultat:

1. Chapitre 1
2. Chapitre 2
3. Chapitre 3

Utilisez les ancres vues au chapitre précédent et les listes pour ajouter un index au début du fichier pp.html qui pointera vers les débuts de chaque paragraphe.

2.4. A retenir

Le (X)HTML est un langage de présentation de données. C'est un langage utilisant des marqueurs. Il est très tolérant : en cas d'erreur ou d'oubli de certains marqueurs, il affichera tout de même quelque chose mais l'aspect obtenu sera certainement différent de celui souhaité.

Une page (X)HTML n'est qu'une proposition d'affichage : c'est le navigateur de celui ou celle qui lira la page (et son environnement) qui est responsable de l'affichage final. Il faut donc utiliser au maximum des marqueurs basés sur le sens du texte plutôt que des marqueurs physiques qui cherchent à produire un effet visuel.