

TP n° 1

Premiers Pas en HTML5

1 Moodle

Le cours d'IO2 est présent sur Moodle (Cours "Internet et outils" à chercher dans "Sciences/UFR d'informatique/L1" ou directement <http://moodlesupd.script.univ-paris-diderot.fr/course/view.php?id=6734>). Connectez-vous sur ce cours avec vos identifiants ENT et inscrivez-vous y comme élève du cours.

Là, vous pourrez :

- trouver les feuilles de TP et les fichiers à télécharger ;
- accéder à un forum dédié au cours où poser vos questions, auxquelles les enseignants comme les étudiants pourront répondre ;
- rendre vos projets ;
- recevoir des messages d'information (pensez à bien rediriger vos mails qui arrive sur Moodle sur votre boîte mail habituelle !) ;
- etc.

Ce document pdf, ainsi que les feuilles de TP suivantes et les documents à télécharger pour faire certains exercices, seront toujours à télécharger sur le Moodle du cours.

2 Débuter en HTML.

Le but de cette partie est :

- de comprendre et écrire des documents HTML élémentaires.
- d'apprendre à se servir des ressources (logiciels, documentation) pour le développement du HTML.

L'accent est mis sur la version 5 de HTML.

2.1 Analyse d'une première page HTML 5

Exercice 1 Visitez le Moodle du cours IO2, "TP/TP1/documents", et affichez dans votre navigateur le fichier `accueil.html`.¹

Regardez le code source de la page (on explique sur cette même page comment faire). Observez en particulier :

- les *balises* (*tag* en anglais), qui commencent par `<` et terminent par `>`, décrivent la structure de la page HTML.
- la balise de déclaration du `DOCTYPE`, obligatoire pour que le document soit reconnu en tant qu'HTML version 5² ;
- les balises *ouvrantes* et *fermantes* qui déterminent le début et la fin des *éléments* (comme `<h1>` et `</h1>`, ou `<p>` et `</p>`) ;
- les éléments `html`, `head`, `title` et `body`, présents dans tout document HTML ;
- l'*attribut* (*attribute* en anglais) `lang` de la balise `<html>`, qui possède la valeur `"fr"` ;
- les éléments *vides*, qui n'ont pas de balise fermante (par exemple, `<meta>`). (En l'occurrence, la balise `<meta charset=...>` indique l'encodage, on en dit davantage plus loin.)

1. Pour afficher dans le navigateur une page html, deux solutions : utilisez votre gestionnaire de fichiers et cliquez sur le fichier ou dans Firefox cliquez à droite et choisissez "Open File". (Si l'option n'y figure pas, choisissez customize et ajoutez l'option.)

2. Sans cela, les navigateurs modernes activent un mode de rendu qui cherche à imiter de vieilles version du navigateur Internet Explorer, mais qui n'est pas conforme à la spécification de HTML. On peut voir le mode de rendu sous firefox : voir "Informations de la page" > "Mode de rendu".

— la balise de commentaire `<!-- ... -->` dont le contenu est ignoré par le navigateur.

2.2 Ma première page HTML

On vous rappelle que le nom d'un fichier HTML prend l'extension `.html`. Le document HTML 5 minimal (conseillé) est constitué de ce qui suit :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Un Titre</title>
</head>
<body>
</body>
</html>
```

On peut bien sûr remplacer “Un Titre” par ce qu'on veut, seul un titre vide est considéré comme une erreur.

Les balises telles que celles qui indiquent les titres (`<h1>...</h1>`, `<h2>...</h2>`, `<h3>...</h3>`, etc.) ou un paragraphe (`<p>...</p>`), ou similaire, doivent être placées entre les balises `<body>` et `</body>`.

Exercice 2 Dans votre répertoire de travail, créez un fichier html appelé `page1.html`, par exemple avec un titre (title) “Une page minimaliste” et contenant le texte “Bonjour tout le monde !” en titre de niveau 1 (`h1`). Vous pouvez également expérimenter les autres balises que vous connaissez.

Nous vous recommandons fortement d'indenter votre code, comme sur la page `accueil.html`. C'est une question de lisibilité ! Le plus facile est d'utiliser `emacs` qui indente automatiquement chaque ligne écrite. Si vous avez besoin de réindenter un bloc : sélectionnez-le à la souris puis appuyez sur la touche de tabulation.

2.2.1 Mes premiers liens

Regardez dans le cours comment insérer un lien (balise `<a>` avec l'attribut `href`).

Exercice 3 Créez une deuxième page similaire à la première dans un fichier appelé `page2.html` dans le même répertoire de travail. Changez le titre (title) en : “Un lien vers ma page minimaliste”. Changez “Bonjour” en “Rebonjour”. Ajoutez un paragraphe “Allez voir ma page 1.” Insérez un lien vers votre première page sur les mots “ma page 1”. Ajoutez de la même manière un lien de votre page 1 vers votre page 2.



2.2.2 Ma première page web

Le SCRIPT dispose d'un serveur web sur lequel vous allez publier les pages ci-dessus. Votre site sera alors accessible à l'adresse : `http://pams.script.univ-paris-diderot.fr/~votre_login`. Ce site n'est accessible que depuis le réseau interne de l'université, il est malheureusement impossible de le voir depuis l'extérieur.

Ce qui est accessible sur votre serveur correspond au contenu du répertoire `public_html` qui se trouve à la racine de votre répertoire principal.

Exercice 4 — Utiliser le serveur web du SCRIPT

1. Il faut tout d'abord s'assurer que le répertoire `public_html` est accessible : créez-le s'il n'existe pas, et donnez les droits d'accès (`chmod a+x ~/public_html`) à ce répertoire ainsi qu'à votre répertoire principal.
2. Mettez-y `page1.html` et `page2.html`, donnez leur des droits en lecture : (`chmod a+r page1.html` et idem pour la page 2). Affichez ces pages dans votre navigateur, à partir de l'adresse en `http` du serveur du SCRIPT. Par exemple, `http://pams.script.univ-paris-diderot.fr/~votre_login/page1.html`. En règle générale, il faudra mettre après votre login, le chemin vers la page **depuis `public_html` non compris**.
Si ça ne marche pas, vérifiez l'adresse, vérifiez les droits... et, si besoin, appelez votre chargé de TP.

2.3 Outils et ressources sur Internet

2.3.1 Documentation

- On pourra se référer (y compris durant les TPs) aux sources suivantes :
- le cours bien sûr ;
 - le site https://www.w3schools.com/html/html5_intro.asp contient des tutoriaux (en anglais, mais assez simple à comprendre), des exemples, ainsi qu'une applet³ qui permet de tester du code HTML directement dans le navigateur. Cependant, comme d'autres sites Internet, c'est quelquefois à prendre avec des pincettes ;
 - pour mémoire, le manuel de référence à destination des développeurs web du W3C⁴ : <https://dev.w3.org/html5/spec-author-view/>.

2.3.2 Le validateur

Le W3C fournit un validateur de pages HTML que l'on trouve à l'adresse <http://validator.w3.org/>. On peut y entrer une URL, ce qui ne fonctionnera pas avec le serveur du SCRIPT qui est isolé de l'extérieur.

On peut aussi copier-coller une page HTML sous l'onglet "Validate by Direct Input", ou télécharger un fichier à partir du disque sous l'onglet "Validate by File Upload". Choisissez la solution que vous préférez.

Une page est valide si le validateur affiche "This document was successfully checked as HTML5 !". (Il est normal qu'il y ait des "warnings".)

Exercice 5 — Validation Vérifiez la validité de la page `accueil.html` de l'exercice 1 en copiant-collant le code source de la page.

1. La page est-elle valide ?
2. Que se passe-t-il si on échange l'ordre des balises `` et `` juste avant **Indice**, sans changer l'ordre des balises fermantes ? Le document HTML ainsi modifié reste-t-il valide ? Pourquoi ?
3. Vérifiez que la plus petite page HTML 5 valide est la suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Un Titre </title>
  </head>
</html>
```

c'est-à-dire qu'on ne peut rien y enlever. Essayez sans titre par exemple.

3. https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

4. Le World Wide Web Consortium est l'organisme chargé de développer les standards pour le web.

Exercice 6 — *Invalidation*

1. Vérifiez la validité de vos pages `page1.html` et `page2.html`.
2. Corrigez votre code HTML jusqu'à ce qu'elles soient valides. Vous pouvez vous aider des indications d'erreur que vous fournit le validateur, des sources ci-dessus, de votre browser habituel, etc.

2.3.3 Inspection du Document Object Model ou DOM

Pour l'instant, on a aperçu nos documents HTML de deux façons :

- comme un fichier texte, à travers le code source de la page, lorsqu'on l'écrit à la main ou qu'on le génère ;
- comme une page web telle que rendue par le navigateur ;

En réalité, le navigateur se représente la page HTML de façon bien plus complexe, conformément au Document Object Model (modèle d'objets de document). Le DOM décrit la structure (arborescente) de la page, mais aussi la façon dont on peut interagir avec la page pour la rendre dynamique (à l'aide d'un langage tel que le JavaScript).⁵

On peut accéder à cette représentation interne dans les versions plus actuelles des principaux navigateurs. Ça permet entre autre d'inspecter l'arbre, d'aider à la mise au point des feuilles de style (dont on parle plus loin), de déboguer du code JavaScript et d'analyser les performances d'un site.

Exercice 7 — *Mes premières inspections*

- Visualisez dans Firefox ou Chrome la page `accueil.html`. Faites un clic droit et choisissez la bonne option (le nom peut varier selon le navigateur, mais en général c'est `Inspect` ou `Inspect élément`). Regardez rapidement comment le navigateur interprète le fichier HTML sous forme d'arbre et modifiez-le (changez une partie du texte, supprimez une branche de l'arbre, etc.). Vérifiez que le changement se fait immédiatement sur la page web.
- Rafraîchissez ensuite votre navigateur. Que se passe-t-il ? Pourquoi ?
- Allez maintenant sur un site un peu plus imposant (comme <http://www.lemonde.fr>), jetez un coup d'œil à l'arbre.

2.4 Éléments de style

Comme on vous l'a expliqué en cours, HTML applique le principe de séparation du *style* et du contenu. Pour décrire le style, ou la façon dont les éléments sont dessinés, on a recours à un autre langage (CSS), souvent dans un fichier séparé.

Les navigateurs utilisent des styles par défaut, par exemple en écrivant les titres en plus gros, le contenu de la balise `<code>` avec des caractères de largeur constante, ou celui de `` en gras. Nous verrons dans quelques semaines comment faire nos propres styles.

Cette semaine le but est de bien comprendre qu'une même page web, avec le même code source donc, peut s'afficher de deux façons bien différentes si on lui applique deux styles différents.

Exercice 8 — *Attacher une feuille de style*

- Reprenez le fichier `accueil.html` et téléchargez les fichiers `style1.css` et `style2.css` dans le même répertoire. Visualisez la page `accueil.html`.
- Rajoutez dans l'entête (entre les balises `<head>`) la balise `<link>` suivante :

```
<link rel="stylesheet" href="style1.css">
```

- Essayez maintenant avec `style2.css`.

5. Le DOM est une recommandation du W3C, intégrée à la recommandation HTML depuis la version 5.

2.5 Quelques balises

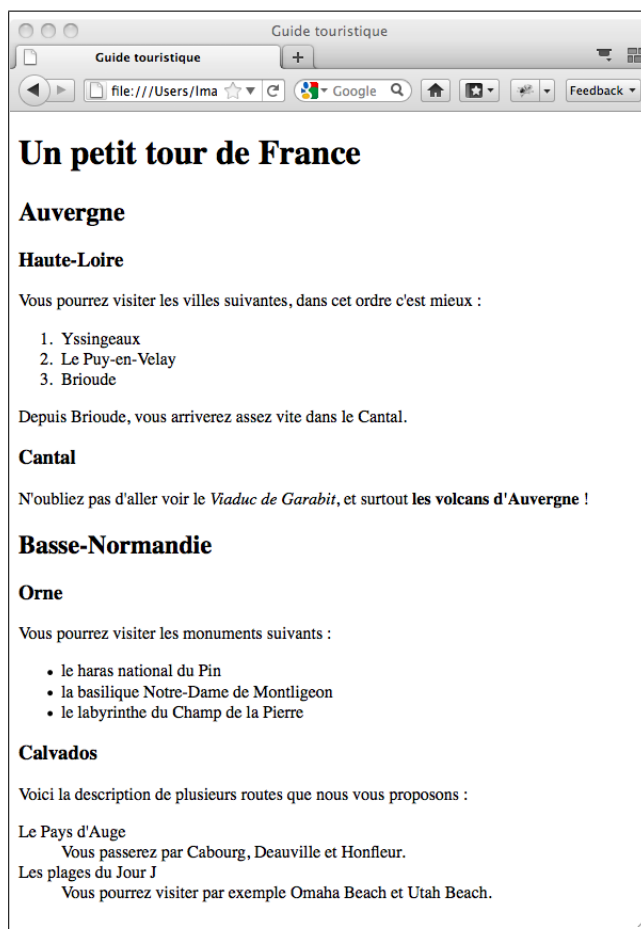
Encore une fois, les balises servent à **structurer** la page web en fonction du **sens** qu'ont les différents éléments. Pour la mise en page, on utilisera les feuilles de style CSS.

Pensez aux personnes aveugles quand vous écrivez une page web. Leur interface vocale ne leur décrit pas la mise en page, mais la structure du document.

Pensez à la façon dont les moteurs de recherche vont vous référencer le jour où vous écrirez de vraies pages web. Là aussi, c'est la structure du document qui prime sur l'affichage...

En plus des éléments d'en-tête, de liens et de paragraphes déjà rencontrés dans les exercices précédents, nous allons utiliser ici les différents éléments de listes, de mise en valeur et d'insistance.

Exercice 9 Téléchargez le fichier `tourisme.txt`, il contient le texte brut de la page web ci-dessous. Utilisez-le pour reproduire cette page web dans un fichier appelé `tourisme.html`, en respectant la structure du document (Indice : allez regarder les balises de liste). Vous pouvez aussi lier une feuille de style à l'aide de la balise `<link>` comme dans l'exercice précédent. Assurez-vous que `tourisme.html` est valide grâce au validateur.



Exercice 10 — balises sémantiques Nous allons ajouter des balises sémantiques (cf. cours) : Ajoutez à `tourisme.html` un titre général ("Le site de [votre nom]"...) grâce des balises `<header>` et `<h1>`. Rajoutez également une liste de deux liens vers <https://fr.wikipedia.org/wiki/Auvergne> et <https://fr.wikipedia.org/wiki/Basse-Normandie>, au sein d'un élément `<nav>`. Le contenu initial sera identifié par une balise `<article>` qui l'entoure.

Assurez-vous que la page est valide grâce au validateur.

Observer que les balises `<header>`, `<nav>`, `<article>` ne changent rien à l'affichage de la page sous Firefox (en l'absence d'une définition explicite de leur style).

Nous verrons plus tard comment exploiter ces balises sémantiques pour améliorer la mise en page.

2.6 Images et liens

Exercice 11 — Images Téléchargez le fichier `Puy_des_Goules1.jpg` disponible sur le Moodle du cours et insérez cette image dans le fichier `tourisme.html` de l'exercice précédent, à l'endroit qui vous semble le plus approprié. Pensez à donner une taille (la hauteur ou la largeur suffira) et à donner un texte alternatif.

Faites de cette image un lien vers le site <http://www.lesvolcansdauvergne.com/>.

Comme toujours, assurez-vous que votre page est valide.

Exercice 12 — Liens internes Regardez dans le cours comment on fait un lien interne (ou ancre). Ajoutez un identifiant à chacun des titres des départements. Ajoutez au début du document, dans le `<nav>`, une liste de liens vers ces départements. Observez l'effet des liens (quitte à réduire la taille de la fenêtre de votre navigateur). Assurez-vous que votre page est valide.

2.7 Encodage

Quand vous enregistrez un document, l'éditeur de texte traduit les caractères en chiffres en fonction d'un *encodage* spécifique. Par exemple : `iso-8859-1` (latin 1) pour les langues occidentales, `iso-8859-15` (latin 9) pour le français, `koi8-r` pour le russe. Ces différents encodages ne sont pas inter-traductibles.

Pour pallier ce problème, un jeu de caractères unifié a été défini, il s'agit d'Unicode. Unicode permet d'écrire tous les caractères actuels (lettres avec accents, chinois, arabe, hébreux, grec, etc.), mais aussi un énorme ensemble de symboles comme des notes de musique ou ceux utilisés en mathématiques.

Unicode définit plusieurs encodages dont `utf-8` dont on va se servir par défaut, car c'est le plus répandu.

Malgré le progrès que constitue Unicode, un document texte n'a pas de sens si l'on ne précise pas l'encodage des caractères. Aussi, il faut faire attention quand on utilise des caractères qui ne font pas partie des 128 caractères du standard américain ASCII : accents, cédilles, signe euro, lettres dans d'autres alphabets (grecs, cyrilliques, etc.). Si l'encodage n'est pas précisé, la page web peut mal s'afficher.

Pour indiquer au navigateur quel est l'encodage utilisé, on a utilisé dans `accueil.html` la balise :

```
<meta charset="utf-8">
```

Bien sûr, il faut veiller à enregistrer le fichier avec le même encodage que celui que l'on annonce ! **Nous vous recommandons fortement de configurer votre éditeur afin qu'il utilise UTF-8 comme encodage par défaut.** Pour les utilisateurs d'emacs, `utf-8` devrait être déjà configuré. Vérifiez que l'unicode est bien activé en bas à gauche de la fenêtre, où un U doit apparaître. Si ce n'est pas le cas, il suffit de rajouter dans le fichier `.emacs` :

```
(prefer-coding-system 'utf-8)
(set-terminal-coding-system 'utf-8)
(set-keyboard-coding-system 'utf-8)
```

Les autres éditeurs doivent être facilement configurables à partir des menus.

Exercice 13 — Réparer les caractères cassés Téléchargez le fichier `codage.html` et observez l'affichage en fonction de l'encodage choisi dans votre navigateur. Ici, il faut choisir le codage unicode, contrairement à ce qui est écrit dans la balise `meta`. (Avec Firefox : dans le menu "View/Text Encoding".)

Il arrive parfois sur Internet d'avoir une page qui est servie avec le mauvais encodage. Vous savez maintenant comment demander à votre navigateur de le corriger.

Pour éviter ce problème, il aurait suffi de donner à la balise `meta` la valeur adaptée.

Exercice 14 Créez un fichier accents.html contenant le titre de niveau 1 “Quelques accents et caractères spéciaux en français” et le paragraphe suivant : “On ne l’admire peut-être pas, mais elle a quelque chose d’idéal et de charmant qui force à s’en occuper. Juliane de Krüdener, Valérie, 1804”. Tapez les caractères accentués au clavier.

- Dans les options de votre éditeur de texte préféré, modifiez l’encodage à l’enregistrement, mettez-le en utf-8 afin d’enregistrer accents.html en utf-8. Par ailleurs, insérez la balise meta annonçant l’encodage utf-8.
- Copiez accents.html dans accents2.html. Essayez d’enregistrer accents2.html en koi8-r. Votre éditeur devrait râler, car vous allez corrompre vos données.
- Revenez maintenant à accents.html. Dans l’en-tête de votre page, modifiez l’attribut charset en remplaçant utf-8 par koi8-r. Que se passe-t-il dans le navigateur ?
- Si jamais vous voulez que vos pages soient lisibles dans n’importe quelle encodage, vous pouvez utiliser les références d’entité de caractères, en écrivant, par exemple ´ au lieu de é. Vous pouvez trouver sur <https://dev.w3.org/html5/html-author/charref> une liste de références d’entité de caractères. Encore avec le charset koi8-r remplacez les caractères spéciaux pour les références d’entité de caractères correspondants. Qu’est-ce que vous voyez comme résultat ?
- Refaites la même manipulation avec iso-8859-7 et iso-8859-15...

Vous savez maintenant faire des pages qui peuvent contenir n’importe quel caractère à l’exception de &, <, et > qui sont spéciaux en HTML. (Nous apprendrons comment faire ces caractères dans un autre TP.)

3 Partie optionnelle

3.1 Tableaux, navigation sur un site web

Le but de cet exercice est de créer un site web sur le cinéma. Vous allez voir comment organiser votre site et naviguer à l’intérieur. Vous allez également voir comment faire des tableaux.

Exercice 15 — Liens relatifs Pour qu’un site soit bien organisé, on peut créer des sous-dossiers dans lesquels seront stockés les différentes pages du site et les documents (images, vidéos, sons...) qui y apparaissent.

Créez un répertoire Films-IO2.

Téléchargez l’archive Films-IO2.zip que vous trouvez sur le Moodle. Il y a tous les documents dont vous aurez besoin :

- Les pages du site : index.html, films.html, stars.html ;
- Les affiches des films ;
- Les photos des acteurs et actrices apparaissant dans ces films.

Organisez tous ces documents dans des sous-dossiers de votre répertoire public_html : Films et Stars. Mettez le fichier index.html dans le dossier Films-IO2.

Visualisez la page index.html dans votre navigateur, à partir de l’adresse du serveur du script. Quelle est son URL complète ?

Corrigez les liens dans les trois pages du site en tenant compte de la nouvelle arborescence.

Exercice 16 — Tableaux Complétez les tableaux contenus dans les pages films.html, stars.html en vous aidant des données contenues dans description.txt. Enfin, validez vos trois pages.

3.2 Médias

La gestion des médias en HTML5 se fait via les balises <audio> et <video>. Tout comme , ces balises prennent en attribut une source «src». Si l’attribut «autoplay» est présent, le média se joue automatiquement. Si l’attribut «loop» est présent le média boucle. L’attribut «controls» ajoute des contrôles sur la vidéo.

Dans le cadre de la vidéo, l’attribut «poster» spécifie une image à placer avant de commencer la lecture.

Exercice 17 — Vidéo Prenez le fichier technician.mp4 sur Moodle. Faites le jouer - automatiquement ou non, avec ou sans contrôle.

3.2.1 Sous-titre

Il est possible d'ajouter des sous-titres au vidéo, en ajoutant la balise <track>. Cette balise à un attribut source dont la valeur est un fichier au format «vtt».

Un exemple de fichier «vtt» ce qui suit :

WEBVTT

```
1
00:00:01.000 --> 00:00:02.000
Texte 1
```

```
2
00:00:02.200 --> 00:00:03.000
Texte 2
```

Chaque passage a la forme :

```
numero du sous-titre
Temps de debut --> temps de fin
Texte du sous-titre (peut etre sur plusieurs lignes)
```

[ligne vide]

où le temps est indiqué de la manière suivante : hh:mm:ss.millièmes.

Exercice 18 — Sous-titre Inventez des sous-titres et mettez-les sur la vidéo.

3.3 DNS

Pour traduire les noms de domaines en adresse IP, on utilise le service DNS.

Exercice 19 Parcourez l'article wikipédia sur DNS :

https://fr.wikipedia.org/wiki/Domain_Name_System

Pour utiliser le DNS, on doit utiliser la commande host dans un terminal. La commande host permet de convertir un nom de domaine en une adresse IP et vice versa.

- Exercice 20**
1. Dans un terminal, regardez le manuel de la commande host.
 2. Dans le terminal, tapez `host www.univ-paris-diderot.fr`. Qu'est-ce qui s'affiche ?
 3. Tapez maintenant la commande `host 81.194.16.89`, que répond le code cette fois-ci ?
 4. Essayez maintenant `host http://www.google.com`. Quelle différence voyez vous ?