

Création et gestion de documents (C2i)

TD 7/12 : HTML

Ce TD est une initiation à la création de pages web. Vous allez découvrir comment fonctionne le web, créer un document HTML de base, apprendre à structurer votre contenu, ajouter des liens, des images et des tableaux. Enfin, vous vérifierez la validité de votre code.

1. Introduction

1.1. Le web, comment ça marche ?

Aller sur le web, tout le monde sait le faire désormais. Expliquer ce qui se passe lorsqu'on tape l'adresse d'un site, ou lorsqu'on clique sur un lien, c'est déjà plus compliqué ! Et s'il s'agit de comprendre comment procéder pour afficher du contenu sur le web, là, il faut commencer à parler d'informatique, de serveurs et de programmation.

Le grand principe de base est le suivant : les internautes sont des **clients** et les sites web des **serveurs**. Le but d'un serveur est, bien évidemment, d'apporter un service au client. Dans notre cas, lui fournir du contenu.

1.1.1 Serveurs

Vous avez tous vu des images de salles machines, avec des allées entières d'armoires remplies de boîtes qui clignotent. Ce sont des serveurs. Mais il n'est pas nécessaire d'avoir une telle infrastructure pour déployer du contenu sur le web.

En effet, un serveur, c'est simplement un logiciel, qui *écoute* sur Internet, en attendant qu'un client lui demande quelque chose.

Et comme c'est un logiciel, rien n'empêche de l'installer sur un simple PC, un Mac, ou même un appareil mobile ! La seule vraie contrainte c'est que l'appareil doit être connecté au réseau Internet.

Dans notre cas, le serveur sera appelé **serveur web**, il en existe de nombreux sur le marché, mais les plus célèbres sont Apache (Apache Software Foundation) et IIS (Microsoft).

Le rôle d'un serveur web est de *servir* (livrer) des documents web aux clients. Ceux-ci sont notamment : les pages web (**HTML**, PHP, ...), les images, les vidéos, mais également les scripts JavaScript, les feuilles de style **CSS**, les animations Flash, etc.

Notes :

- les fichiers que l'on installe sur un serveur web pour qu'ils soient accessibles sur le web respectent les mêmes principes de dossiers, noms de fichiers, extensions que vous avez l'habitude d'utiliser. La seule différence, c'est que les chemins sont affichés comme sous Unix/Linux, donc sous la forme dossier/sous_dossier/fichier.extension et non pas (comme sous Windows) de la forme dossier\sous_dossier\fichier.extension
- évitez vous de nombreux problèmes : n'utilisez jamais de nom de fichier avec des accents ou des espaces !
- un fichier accessible/visible sur le web l'est complètement. Inutile d'essayer de le rendre impossible à copier ou à modifier. Si on peut le voir publiquement sur le web, alors les utilisateurs pourront le sauvegarder chez eux, d'une manière ou d'une autre.

1.1.2 Le réseau et les protocoles

Si je vous demande sur quel réseau vous accédez aux sites, vous le savez tous : **Internet** !

Mais lorsqu'on parle d'un site, on parle plus généralement d'un site **Web**.

Quelle est la différence entre Internet et le Web ?

Au passage, que signifient les WWW en début d'une adresse de site Internet (Internet, vraiment ?)

Création et gestion de documents (C2i)

TD 7/12 : HTML

Et que signifie HTTP ? Et HTML ?

Voyons cela :

- WWW = World Wide **Web**. En français : la toile mondiale.
 - Toile ? Parce que les liens entre les pages web permettent de tisser un maillage semblable à une toile.
- HTTP = *Hyper Text* Transfer **Protocol**. En français : protocole de transfert de l'hyper texte. C'est ce qui définit le format de communication (échanges) entre le client et le serveur.
 - Hyper texte ? Ce sont des informations (du texte) liées à d'autres, via des liens (hyper liens), qui permettent de passer d'une information à l'autre.
 - Le protocole HTTP définit également les erreurs de transfert, par exemple, la fameuse erreur 404 ! (404=document introuvable).
- HTML = *Hyper Text* Markup **Language**. En français : langage de balises hyper texte.
 - Balise ? C'est une entité qui sert à structurer du texte (de l'information donc).

Enfin, revenons à notre question initiale : internet ou web ? **Le web est un système que les serveurs web utilisent, en s'appuyant sur le réseau des réseaux (en anglais : Internet !), via le protocole HTTP (notamment), pour livrer des pages HTML (ou autre, compatible avec le protocole HTTP) aux clients.**

Mais Internet, si vous l'avez bien compris, c'est beaucoup plus que le web. C'est une infrastructure réseau mondiale, qui sert au web, mais également à d'autres activités fondamentales, comme échanger des mails, faire de la visioconférence, de la téléphonie sur IP, le transfert de fichier...

Note : sur le web, on trouve souvent le protocole HTTPS qui est la même chose que le HTTP, mais avec de la Sécurité des échanges. C'est notamment utilisé dans les services bancaires.

1.1.3 Les navigateurs

Les navigateurs, vous vivez quotidiennement avec. Les 5 plus populaires ? Chrome, Firefox, Internet Explorer, Safari, Opera.

Ce sont nos **clients** web. Ils formulent des requêtes (en HTTP) pour demander (en passant par Internet) des pages web aux serveurs, qui renvoient les pages demandées.

Pour le serveur, le client importe peu : à partir du moment où la requête HTTP est bien formulée, il la comprend et répond correctement (ou pas, d'où les erreurs 404 et autres).

Mais pour un développeur web, le problème des navigateurs est un véritable casse tête. En effet, HTML est un langage, qui répond à une « norme », et normalement (sic!), si le code ne respecte pas la norme, cela devrait générer des erreurs.

Or, ce n'est pas le cas ! Les navigateurs sont « gentils » avec les internautes et « essaient » coûte que coûte, d'afficher quelque chose à l'écran, même si la page HTML a été mal codée.

Plus grave encore, certains navigateurs ne respectent pas correctement la norme, et affichent du contenu d'une façon qui n'est pas celle qui est définie dans la page HTML ni dans la norme !

C'est comme cela que les web-développeurs sont obligés d'utiliser des astuces pour essayer d'avoir le bon affichage, quel que soit le navigateur, et de tester dans les différentes configurations des clients (navigateurs, système d'exploitation, versions, ...)

Création et gestion de documents (C2i)

TD 7/12 : HTML

À tout cela, ajoutez l'arrivée massive des smartphones et tablettes sur le web, et vous allez comprendre qu'en plus, il faut adapter le contenu des sites à la taille des (petits) écrans de nos appareils mobiles. Et je ne parle même pas des environnements IOS/Android, encore des problèmes de compatibilité en vue !

Remarque : Nous sommes dans un TD d'informatique, certes, mais pas pour faire de vous des web-développeurs chevronnés. Aussi, nous allons partir du principe que nous utilisons tous Firefox. Pourquoi ? Parce qu'il est libre, gratuit, populaire et disponible dans tous les systèmes d'exploitation courants. Ainsi, nous allons faire abstraction de tous ces soucis, et nous focaliser sur l'essentiel : la syntaxe HTML et la création de contenu structuré.

1.2. L'éditeur de texte

Vous allez le voir très rapidement, mais le code informatique, de manière générale, se matérialise par du texte simple. Il est donc inutile d'avoir un logiciel de traitement spécifique pour faire du code, et tout programme capable de sauvegarder un document « .txt » fera l'affaire.

Ceci étant dit, le meilleur outil, c'est celui que vous maîtrisez le mieux. Donc, si vous avez un éditeur de prédilection, utilisez-le ! Sinon, je vous recommande Brackets, Gedit ou Notepad++.

2. HTML, c'est quoi ?

Comme nous l'avons déjà vu, HTML signifie Hyper Text Markup Language (langage de balises hyper texte).

Concrètement, c'est du texte, avec des balises (éléments de syntaxe et de structuration) pour délimiter les différentes zones de texte et définir leur « catégorie ».

HTML permet de créer des pages complètes, avec tout le contenu textuel (et sa structuration en titres, paragraphes, tableaux, ...), les images, les liens...

Par contre, idéalement, HTML ne s'occupe jamais de la mise en forme. C'est le rôle de CSS.

Par conséquent, les documents que nous allons traiter ici seront un peu (beaucoup !) austères. Patience ! Il s'agit de structurer correctement votre contenu afin de ne pas avoir à y revenir lors de l'application de la mise en forme via CSS.

2.1. Exercice n°1 : Mon premier document HTML

1. A l'aide de votre outil d'édition favori, écrivez le code suivant :

```
1 |<!DOCTYPE html>
2 |<html>
3 |   <head>
4 |     <meta charset="utf-8" />
5 |     <title>Hello World !</title>
6 |   </head>
7 |   <body>
8 |     Ceci est ma première page HTML !
9 |   </body>
10|</html>
```

2. Sauvegardez votre document avec le nom 01-hello_world.html, sur votre espace DropBox, dans le dossier TD7 que vous créerez pour l'occasion.
3. Ouvrez votre fichier avec votre navigateur.
4. Bravo :-)

Création et gestion de documents (C2i)

TD 7/12 : HTML

2.2. La syntaxe

Analysons le contenu de votre première page web, et avançons un peu sur la syntaxe HTML.

2.2.1 Le doctype

En première ligne, on voit apparaître le **doctype**. Il est obligatoire pour gérer correctement le contenu de sa page et qu'elle s'affiche correctement. C'est ce qui définit officiellement que nous sommes en train de travailler avec un document de type HTML (et cette syntaxe de doctype indique également implicitement que nous sommes en HTML5, nous y reviendrons).

2.2.2 Les balises

Une balise est un mot entouré par les caractères « < » et « > ». Elle sert à définir un contexte (titre, paragraphe, image, lien, cadre, tableau, ...).

2.2.2.1 l'élément racine

En 2^e ligne, apparaît notre première **balise**, qui est aussi *l'élément racine*, **<html>**. C'est l'élément qui va contenir toute *l'arborescence* de la page web.

2.2.2.2 La structuration arborescente et les balises fermantes

Les balises sont structurées de manière arborescente. A part l'élément racine, chaque balise est contenue dans une autre balise, formant une arborescence. Et l'enchaînement des ouvertures et fermetures de balises doit être correctement coordonné : Une balise ouvrante démarre un contexte, une balise fermante le clôture. Une **balise fermante** utilise le même mot clé que la balise ouvrante, mais en ajoutant un « / » immédiatement avant le mot.

Très important : il est donc impossible de fermer une balise contenante avant d'avoir fermé toutes les balises contenues.

Ex : les lignes de code suivantes sont invalides :

- `<head><title>exemple</head>` : il manque la balise `</title>` !
- `<head><title>exemple</head></title>` : la balise `</title>` apparaît après `</head>` alors qu'elle a été ouverte à l'intérieur, et doit donc se fermer à l'intérieur également !

Le code correct est donc : `<head><title>exemple</title></head>`.

Les autres balises présentes dans cette première page sont `<head>`, `<meta>`, `<title>` et `<body>`, avec leur balise fermante respective.

2.2.2.3 Les balises « spéciales »

Il existe, en plus de la balise racine (`<html>`), 2 autres balises spéciales : `<head>` et `<body>`.

`<head>` sert à définir les entêtes de la page. Les éléments qui y sont contenus ne sont pas affichés directement dans la page web, mais ils servent à intégrer du contenu et spécifier des éléments. C'est un peu comme si c'était une section de paramétrage de la page. On peut définir son titre, son encodage de caractères, inclure des scripts, des polices, ajouter des mots clés et une description du contenu de la page, ...

Tout ceci sera utile pour intégrer du contenu ou pour donner des indications aux navigateurs (et aux moteurs de recherche).

Création et gestion de documents (C2i)

TD 7/12 : HTML

`<body>` sert à définir le contenu de la page. Ce qui sera immédiatement affiché dans les navigateurs. C'est le corps (body) de la page. Tout ce qui est inséré dans cette balise sera (par défaut) visible à l'écran.

2.2.3 Les attributs

Les balises peuvent être simplement formées par leur nom (comme `<body>` dans notre exemple) ou avoir des attributs qui précisent des propriétés.

La syntaxe est celle que nous voyons avec la balise `<meta>`. Cette balise n'a pas de signification si elle n'est pas accompagnée d'attributs et de valeurs associées à ces attributs.

Exemple : dans le code : `<meta charset="utf-8" />`, le mot *charset* est l'**attribut** sert à spécifier le type d'encodage (le *charset* pour character set : le jeu de caractères utilisés) de la page, suivi de « = » puis, obligatoirement entre guillemets, de la **valeur** associée à l'attribut (ici, `utf-8`).

Note : nous n'allons pas étudier les jeux de caractères, mais sachez qu'ils sont fondamentaux et servent notamment à définir comment sont représentés les caractères accentués dans les fichiers. Ainsi, un jeu de caractères mal défini par rapport à ce qui est saisi dans la page web engendre des erreurs d'affichage, qui font apparaître des caractères spéciaux. (ex : Å au lieu de è)

2.2.4 Les entêtes (contenu de la balise `<head>`)

Comme nous l'avons vu implicitement dans l'exemple, il existe deux éléments importants à définir dans le cadre de l'entête. Il s'agit du jeu de caractères et du titre de la page.

- Il faut donc utiliser les balises `<meta charset="utf-8" />` (ou autre charset en fonction des besoins) pour définir l'encodage de la page.
- Le titre de la page n'est pas celui qui s'affichera à l'écran mais celui qui sera visible dans la barre de titre du navigateur. Pour le définir, il suffit d'utiliser la balise `<title>`.

2.2.5 Norme et conventions

2.2.5.1 Définition et conformité à la norme

Tout langage respecte une grammaire (ou syntaxe, ou norme). En HTML, elle est dictée par le W3C (World Wide Web Consortium). Le site <http://www.w3.org> vous donne accès à tous les travaux du consortium, les normes des langages et des outils pour webmasters, notamment le célèbre (et très utile) **validator**.

2.2.5.2 Exercice n°2 : validation de page web.

Allez sur <http://validator.w3.org/> et validez que votre première page HTML est correctement structurée.

2.2.5.3 Éléments principaux de la norme

Voici les principaux éléments de la norme qu'il faut respecter à tout prix, afin de s'éviter un maximum de problèmes de compatibilité et d'affichage :

- Nous avons déjà parlé des balises fermantes. Toute balise ouvrante doit avoir sa balise fermante associée. Et les balises fermantes doivent être fermées dans le bon ordre. Ainsi, la dernière balise ouverte doit être la première fermée, puis l'avant dernière et ainsi de suite.
 - Il existe des balises particulières qui sont entièrement définies avec leur balise ouvrante et qui ne nécessitent pas de balise fermante. Pour ces cas spéciaux, on ferme la balise implicitement en remplaçant « > » par « /> ». Ex : `<meta charset="utf-8" />`
- les noms de balises et d'attributs sont forcément en minuscule.
- Les valeurs des attributs sont forcément entourées par des guillemets.

Création et gestion de documents (C2i)

TD 7/12 : HTML

2.2.5.4 Conventions de codage

Ici, il ne s'agit pas d'éléments obligatoires pour respecter le langage, mais de bonnes pratiques afin de mieux lire (et relire) son code. Un bon code est avant tout un code lisible. Pour qu'il le soit, nous allons appliquer 3 grands principes :

- l'indentation du code. C'est à dire les espaces en début de ligne. Comme vous avez pu le voir dans le premier exemple, à chaque fois qu'une balise est ouverte, nous ajoutons une tabulation. Cela permet de rapidement voir dans quelle balise est placé chaque contenu et de bien visualiser l'arborescence des balises.
- Les sauts de ligne : après chaque balise (ouvrante ou fermante), nous allons à la ligne, sauf si le contenu de la balise est suffisamment petit pour tenir sur une seule ligne (voir `<title>` dans notre exemple)
- lorsqu'on fait quelque chose d'un peu compliqué, ou pour ajouter de la clarté dans le code HTML (changement de section, objet particulier...), on ajoute des commentaires !
 - La syntaxe des commentaires est la suivante : pour démarrer un commentaire, on utilise `<!--` et pour le terminer, on utilise `-->`
 - un commentaire peut donc s'étendre plusieurs ligne, tant qu'on n'a pas rencontré `-->`

3. HTML : Mise en pratique

Après cette très longue (mais nécessaire) présentation des concepts et de la syntaxe, nous allons (enfin !) passer aux exercices pratiques.

Vous allez vite constater qu'il existe énormément de balises. Il est donc utile de les manipuler pour voir leur fonctionnement et mieux mémoriser ce qu'elles permettent.

L'objectif va être de vous faire découvrir les balises par vous même, et qu'à la fin des exercices vous disposiez d'une page de référence contenant toutes les balises utiles et un exemple d'utilisation pour chacune.

Pour cela, vous aurez sûrement besoin d'une référence (un manuel) en matière de balises HTML afin de vérifier la syntaxe et la signification des balises que vous allez devoir utiliser. A toute fin utile, utilisez ceci : <https://developer.mozilla.org/fr/docs/Web/HTML/Element>

Et n'oubliez pas de regarder le résultat de votre travail dans un navigateur, après chaque modification de votre fichier HTML !

3.1. exercice n°3 : Mise en forme du texte

Commençons par le plus simple, mais aussi le plus riche en balises : la mise en forme du texte.

1. Vous allez créer un nouveau document vide, y placer correctement les balises spéciales requises (`html`, `head`, `meta`, `body`), et sauvegarder votre fichier avec le nom `02-refcard.html`.
2. Partant de cette page vide, et comme vu précédemment, vous allez commencer par ajouter un titre pour la page : « ma carte de référence sur les balises HTML »
3. Vous allez utiliser les chapitres pour structurer votre contenu. Donc ajoutez une balise `<h1>` avec pour contenu « La structuration de données en HTML ».
4. Ajoutez une balise `<h2>` avec pour contenu « La mise en forme du texte ».
5. Essayez de saisir le texte suivant, sur 2 lignes :
« nous allons découvrir ici quelles balises il faut utiliser pour mettre en forme du texte dans un document HTML. »
6. Découvrez la balise `
`, son utilité, et n'hésitez pas à l'utiliser à chaque fois que nécessaire !
 - (a) Vous remarquez que `
` est une balise qui se ferme automatiquement. On peut voir `
` seul, mais cela signifie qu'il manquera une balise fermante. C'est accepté en HTML5, mais je préfère que vous fermiez toutes vos balises, donc utilisez `
`. On y reviendra.

Création et gestion de documents (C2i)

TD 7/12 : HTML

7. Découvrez la balise `<p>` et écrivez un exemple de son utilisation : « ceci est mon premier paragraphe, j'utilise la balise spécialement prévue pour mettre en forme ce genre de contenu et automatiquement sauter les lignes adéquates en début et en fin de paragraphe. Une balise pratique ! Elle se distingue de la balise br par le fait qu'elle ne sert pas à aller à la ligne, mais bien à changer de paragraphe. C'est sémantiquement différent. ».
8. Découvrez la balise `<abbr>` et servez vous en pour définir le mot « HTML ».
9. Découvrez la balise `<dd>` et celles qui lui sont nécessaire, et définissez ce qu'est HTTP.
10. Découvrez la balise `` et saisissez le texte : « Internet et le web, c'est pareil ! ».
11. Découvrez la balise `<ins>` et saisissez le texte : « Internet c'est le réseau, le web c'est les sites ! ».
12. Découvrez les balises ``, `` et `<mark>`. Créez un nouveau paragraphe et insérez la phrase suivante : « En HTML₅, de nouvelles balises sont apparues, comme mark par exemple. Elles ne servent qu'à définir la sémantique du contenu (mise en valeur, importance, surlignage). L'apparence change par défaut, mais on pourra la modifier en CSS. » et faites en sorte que « HTML₅ » soit mis en valeur, « balises » soit important et « mark » soit surligné.
13. Découvrez la balise `` et celles qui lui sont associées, et créez une liste formatée ainsi :
 1. HTML c'est bien
 - mais HTML₅ c'est mieux
 2. CSS c'est encore mieux !
 - mais c'est beaucoup plus compliqué !
14. Découvrez les balises `<sub>` et `<sup>` et insérez le texte « c'est incroyable, H₂O et M^{lle} sont correctement affichés en HTML. »
15. Enfin, découvrez l'élément `<cite>` et insérez le texte suivant : « Parce qu'il n'y a pas d'apostrophe à Polytech, la brigade de l'apostrophe sanctionne ! ». Que ceux qui ne savent pas de quoi il est question aillent voir Lucas !
16. Modifiez votre page pour qu'avant l'apparition de chacun des exemples de balise que vous avez utilisé, vous ayez un titre correspondant, avec le bon niveau de hiérarchie (`<h3>` donc, si vous suivez !) et le nom de la balise utilisée en contenu. N'hésitez pas à reformater votre page, et supprimer des paragraphes (balise `<p>` si besoin !)
17. Sauvegardez votre document sur votre espace DropBox.

Remarque importante : vous constatez que ces balises modifient la mise en forme du texte. Ce n'est pas leur objectif premier. En effet, cette mise en forme est complètement modifiable en CSS. Par contre, ces balises servent à enrichir, préciser le contexte et ajouter de la sémantique à votre contenu. C'est là, l'utilité fondamentale. Plus votre document est bien structuré, plus il sera lisible, et plus facilement il sera modifiable avec CSS.

3.2. Les liens

Nous entrons enfin dans la dimension web. Sans les liens, le web n'est qu'une série de pages contenant du texte sans aucun moyen de faire référence de l'une à l'autre.

La balise `<a>` pour *anchor* (ancree) fait partie de celles qui n'ont pas de signification sans attribut. Le plus important, le plus connu (et le seul vraiment utile d'ailleurs) est l'attribut « `href` », qui spécifie la destination du lien.

3.2.1 Les différents types de liens

Il existe au moins 3 types de liens :

- ceux qui envoient sur une autre page de votre site.
- ceux qui envoient sur une page d'un autre site.
- Ceux qui envoient vers une autre section de votre site.

Création et gestion de documents (C2i)

TD 7/12 : HTML

Les trois utilisent la même balise : `<a>`, mais avec un attribut `href` différent :

- le premier indique tout simplement le nom (et si besoin, le chemin) du fichier à ouvrir
- le deuxième indique l'adresse complète, en commençant impérativement par le protocole (ex : `http://...`)
- le troisième utilise le nom de l'ancre interne à la page, vers laquelle il doit pointer. Les noms d'ancres commencent par « # »
 - pour définir une ancre dans le contenu d'une page, il faut utiliser la syntaxe suivante : ``. Puis on peut créer un lien qui permettra d'accéder à cette ancre rapidement, ainsi : `Retourner à mon ancre`.

3.2.2 Exercice n°4 : mise en place d'hyperliens.

Dans le même fichier `02-refcard.html` :

1. Ajouter un titre de niveau 2 et indiquer « La gestion des liens »
2. Faites ce qu'il faut pour créer un lien qui renvoie en haut de la page avec pour texte « Retour au début ». Si vous lisez bien la doc, vous gagnez du temps...
3. créez un lien vers votre première page `01-hello_world.html`, avec le texte « Redécouvrez ma première page ! »
4. créez un lien vers le validator `w3c` avec pour texte « Accès au W3C validator ».
5. ajouter les titres de niveau 3 adéquats pour ces différents exemples.
6. Sauvegardez votre document sur votre espace DropBox.

3.2.3 Les liens avancés.

- Lorsque qu'une page souhaite accéder à une autre page dans le même site, mais dans un autre dossier, il faut utiliser les chemins des dossiers dans les liens. Par exemple, une page qui souhaite accéder à la `page2.html` située dans le dossier `test` devra écrire un lien de la forme `voir la page 2`
- le système d'ancre interne à une page fonctionne également avec les liens vers d'autres pages. Il suffit d'utiliser une syntaxe du type :
 - `voir mon ancre dans la page 2`.
- il est possible d'envoyer un e-mail avec un lien ! Pour cela, il faut utiliser la syntaxe :
 - `envoyez moi un mail !`.

3.2.4 Exercice n°5 : mise en place d'hyperliens avancés.

1. Créez un nouveau dossier nommé « niveau_2 », et créez à l'intérieur un fichier nommé `03-test_liens.html`.
2. Dans ce fichier, qui sera syntaxiquement correct, mettez un titre au document : « Test des liens vers ancres et chemins ».
3. Ajoutez une ancre nommée « ancre_test » avec pour texte : « Ceci est une ancre pour faire un test ».
4. Toujours dans le dossier `niveau_2`, créez une nouvelle archive (un dossier compressé si vous préférez) nommée `04-test_archive.zip`.
5. Dans le fichier `02-refcard.html` :
 - (a) Ajoutez un lien pour accéder à `ancre_test` dans la page `03-test_liens.html`, avec le texte « Lien vers une ancre dans une page d'un sous dossier ».
 - (b) Ajoutez un lien permettant de vous envoyer un e-mail, avec pour sujet « yeah » et comme texte « Spammez-moi ! »

Création et gestion de documents (C2i)

TD 7/12 : HTML

- (c) Ajoutez un lien pointant sur l'archive 04-test_archive.zip, avec pour texte « Téléchargez mon .zip ! ». Testez ce lien ! Magique !
 - (d) Ajoutez les titres de niveau 3 adéquats pour nos nouveaux liens.
6. Sauvegardez vos documents et dossiers sur votre espace DropBox.

3.3. Les images

La balise `` fait également partie de celles qui n'ont pas de signification sans attribut. Le plus important est l'attribut « `src` », qui spécifie la *source* de l'image (le nom du fichier, et son éventuel chemin pour y accéder). L'attribut « `title` » est très utile pour afficher un texte lorsqu'on passe sur l'image avec la souris (« *tooltip* »).

L'attribut « `alt` » est également très utile car il permet d'afficher un texte lorsque l'image n'a pas pu être chargée, ou lorsque le navigateur souhaite décrire l'image (description/accessibilité aux mal-voyants par exemple).

Il existe 3 types d'images particulièrement adaptés et bien supportés par le web :

- PNG : bonne qualité et gestion de la transparence, mais mal supporté par les vieux navigateurs et peu adapté pour les photos.
- GIF (voire le GIF animé) : format historique, gestion de la transparence, mais 256 couleurs uniquement, moins performant que le PNG mais mieux supporté par les navigateurs. Et permet des animations sans utiliser de vidéo ou de Flash.
- JPEG : bonne qualité d'image pour un poids assez faible. Le format de prédilection sur le web.

3.3.1 Exercice n°6 : ajout d'images

1. Créez un dossier nommé « images », insérez-y le logo de l'université, que vous nommerez 05-logo_uns avec l'extension correspondante à son format.
2. Dans le fichier 02-refcard.html :
 - (a) Ajoutez un nouveau titre de niveau 2 et indiquer « Les images »
 - (b) Insérez l'image 05-logo_uns et ajoutez un texte alternatif et un titre à l'image : « Le logo de Université Nice Sophia »
3. Sauvegardez votre document sur votre espace DropBox.

Remarque : avez vous pensé à fermer correctement vos balises `` ? Eh oui, c'est comme pour la balise `
` vue précédemment !

3.4. Les tableaux

Les tableaux sont assez pénibles à écrire en HTML, du fait de l'arborescence des balises. Le principe de fonctionnement est le suivant : on déclare une balise `<table>`, puis une (ou plusieurs) balises `<tr>` (table row : les lignes), qui elles-mêmes contiendront une (ou plusieurs) balises `<td>` représentant cellules (cases) du tableau.

Ajoutez à cela des balises supplémentaires pour faire des entêtes de colonnes (`<th>`), des légendes (`<caption>`), et il vous manquera encore des balises avancées (non étudiées ici) telles que `<colgroup>`, `<thead>`, `<tbody>` ou `<tfoot>`.

3.4.1 Exercice n°7 : ajout de tableau

1. Dans le fichier 02-refcard.html :
 1. ajoutez un titre de niveau 2 nommé « Les tableaux ».
 2. créez un tableau à 3 colonnes, avec pour entête « Nom », « Prénom » et « Age », et que vous remplirez avec des données de votre choix (au moins 2 lignes).
 3. Ajoutez la légende : « L'age de certaines personnes ».

Création et gestion de documents (C2i) TD 7/12 : HTML

4. Validez que votre document est validé par le W3C. Si ce n'est pas le cas, faites ce qu'il faut pour le valider !
5. Sauvegardez votre document sur votre espace DropBox.

Sources bien utiles pour créer ce TD, ainsi que des références très intéressantes pour les webmasters :

<http://validator.w3.org/>

<http://fr.openclassrooms.com/informatique/html/cours>

<http://fr.openclassrooms.com/informatique/cours/apprenez-a-creeer-votre-site-web-avec-html5-et-css3>

<http://www.w3schools.com/html/default.asp>

<http://www.alsacreations.com/tutoriels/>

<http://www.html5rocks.com/en/resources>

<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>

<http://diveintohtml5.info/>

<http://html5doctor.com/>