

HTML : Mini tutoriel

I. Introduction

HTML est un langage de formatage de texte adapté à la conception de pages web.

Le HTML est un langage *balisé*, fondé en particulier sur la notion de lien hypertexte, qui permet de "surfer" d'une page à l'autre.

HTML continue à évoluer : en 2009 le W3C décide de créer HTML5. Ce langage n'est pas encore complètement finalisé, mais c'est sur cette dernière version qui est aujourd'hui le langage d'avenir que nous allons travailler.

Le HTML5 définit la structure logique d'un document, permettant ensuite à un navigateur (Firefox, Mozilla, Netscape, Opera, Internet Explorer, etc.) de rendre visuellement (mais aussi auditivement ou tactilement, pour les non-voyants) le document.

J'ai essayé de réduire au maximum ce document, vous trouverez plus d'explications sur le site d'openclassrooms, si nécessaire.

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

Je vous recommande tout particulièrement le memento :

<http://fr.openclassrooms.com/informatique/cours/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-1>

et le site w3school : <http://www.w3schools.com/tags> qui détaille chaque balise, et vous permet de faire des tests

II. Le langage HTML

Définition

HTML est un langage pour **décrire** des pages web.

HTML est le sigle pour Hyper Text Markup Language (langage de balisage d'hypertexte)

Voici un exemple de code HTML :

```
<html>
  <head>
    <title>Le titre de ma page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <strong> paragraphe </strong><p>
  </body>
```

Figure 1.1 – Un exemple de code HTML

Lecture du code HTML

Les navigateurs Internet (comme Internet explorer, Firefox ou Chrome par exemple) sont capables de lire les documents HTML et de les afficher, sous forme de pages web. Certains éléments, comme <html>, <body>, etc... ne sont pas affichés sur la page web. Ces éléments donnent des indications au navigateur sur la valeur (ceci est un titre, ceci est un sous titre, ceci est un paragraphe, ceci est une partie importante...) de chaque partie du texte)

Le code HTML ci-dessus, interprété par un navigateur, donne le résultat suivant :

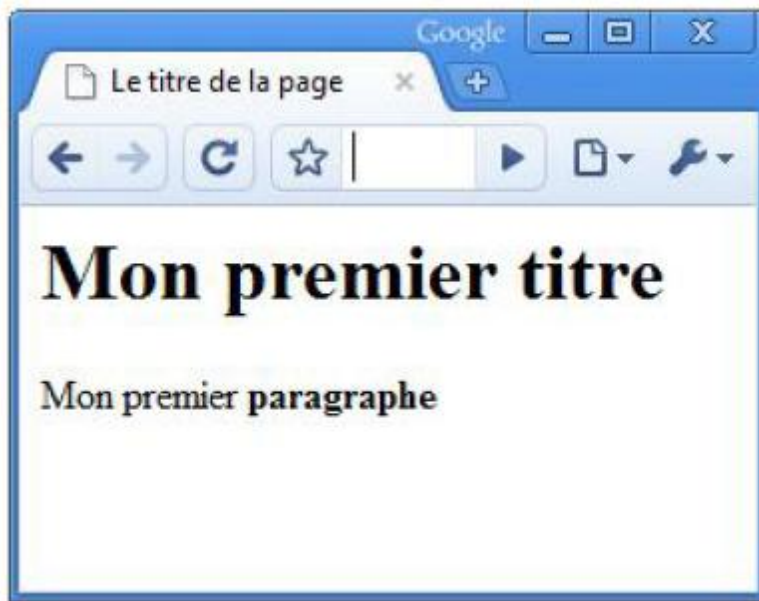


Figure 1.2 – Le code HTML de la figure 1.1 interprété par Google Chrome

1. Reconnaître une balise

Document HTML = textes + balises

Pour composer une page web, on écrit donc du texte, et on y ajoute **des balises** qui permettront au navigateur (par exemple Firefox ou Internet Explorer), de mettre en forme ce texte.

Un document HTML est donc composé de :

- textes
- balises

Sur la figure 1.1, les balises sont :

<html>, <head>, <title>, </title>, </head>, <body>, <h1>, </h1>, <p>, , , </p>, </body>, </html>

2. Balises ouvrantes et balises fermantes

Les balises vont par paire : une balise ouvrante et une balise fermante. Les balises fermantes s'écrivent avec un slash (/). Voici les balises ouvrantes et les balises fermantes de l'exemple ci-dessus :

Balises ouvrantes : <html>, <head>, <title>, <body>, <h1>, <p>,

Balises fermantes : </title>, </head>, </h1>, , </p>, </body>, </html>

Le texte écrit entre la balise ouvrante et la balise fermante est le contenu de la balise. Dans l'exemple suivant

```
Mon premier <strong >paragraphe</strong >
```

le terme paragraphe est le contenu de la balise .

Sur la figure 1.1 voici le contenu de certaines balises :

– contenu de la balise <head> : <title>Le titre de la page</title>

– contenu de la balise <h1> : Mon premier titre</h1>

– contenu de la balise <body> : <h1>Mon premier titre</h1><p>Mon premier paragraphe</p>

Les balises agissent sur leur contenu. Il existe de très nombreuses balises. Pour l'instant, nous allons uniquement décrire la signification des balises utilisées dans l'exemple de la figure 1.1.

Structure de base

La structure de base de tout document HTML est la suivante :

```
<!DOCTYPE html>

<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

<!DOCTYPE html>

La toute première ligne s'appelle le **doctype**. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML.

Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation).

Rq : Cette ligne du doctype était autrefois incroyablement complexe. Il était impossible de la retenir de tête. Pour XHTML 1.0, il fallait écrire :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Dans le cadre de HTML5, il a été décidé de la simplifier, pour le plus grand bonheur des webmasters. Quand vous voyez une balise doctype courte (<!DOCTYPE html>), cela signifie que la page est écrite en HTML5.

<html>

Cette balise englobe tout le contenu de la page.

<head>

Entre les balises <head> et </head> se trouvent des informations supplémentaires, qui n'apparaissent pas directement sur la page, comme par exemple les styles utilisés sur la page, l'encodage des caractères, le nom de l'auteur...

Pour que tous les caractères de votre texte s'affichent correctement, il **FAUT** ajouter :

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

pour préciser la table de caractère utilisée (Nous y reviendrons plus tard dans l'année, mais si vous êtes curieux vous pouvez chercher par vous-même !!)

Les informations qui apparaîtront directement sur la page sont écrites entre les balises <body> et </body>

<title>

La balise <title>, inscrite dans la partie head correspond en fait au texte qui sera affiché comme titre de l'onglet (voir figure 1.2). C'est aussi le titre qui apparaît dans la page de résultat des moteurs de recherche (Google...).

<h1>

La balise <h1> signifie heading 1. Cette balise permet d'indiquer que le contenu de la balise est un titre. Les caractères entre <h1> et </h1> sont donc affichés, par défaut, avec une taille de police plus grande que les autres caractères de la page. (modifiable avec le CSS)

Il est également possible de définir des sous-titres, des titres de section, etc... en utilisant les balises suivantes : <h1>, <h2>, <h3>, <h4>, <h5>, <h6>. <h1> sera utilisée pour les titres principaux, <h2> pour les sous-titres, <h3> pour les titres de section, etc...

<p>

La balise <p> permet de définir les paragraphes. A la fin du paragraphe, le navigateur fait un retour à la ligne, pour commencer une autre partie.

< strong >

La balise < strong > indique que le texte est important, par défaut, il s'affichera en gras dans les navigateurs, mais sa fonction est surtout d'indiquer que c'est un texte important.

3. A VOUS !

Nous allons utiliser Notepad ++, comme éditeur de texte. (Chaque éditeur a ses avantages et ses inconvénients, il fallait faire un choix !)

- 1) Ouvrez Notepad ++
- 2) Il faut indiquer à votre éditeur de texte d'enregistrer le fichier en UTF-8, pour cela dans Notepad++, allez dans le menu Encodage > Encoder en UTF-8 (sans BOM) pour que votre fichier soit enregistré en UTF-8 dès le début. Cela ne s'applique qu'au fichier actuellement ouvert.
Pour ne pas avoir à le faire pour chaque nouveau fichier, je vous conseille d'aller dans le menu Paramétrage > Préférences, onglet Nouveau document/Dossier. Sélectionnez UTF-8 sans BOM dans la liste.

3) Indiquer à l'éditeur que vous écrivez en HTML : Langage > H > HTML . Lorsque vous utiliserez le logiciel, il colorera votre code, ce qui vous permettra de vous repérer plus facilement. Une manière de faire que je recommande, est d'enregistrer tout de suite votre fichier même vide, Fichier/Enregistrer Sous choisissez un repertoire et indiquer dans le menu déroulant : Hyper Text Markup Language Files ... **RQ : raccourci clavier pour enregistrer ctrl + S**

4) Ecrivez votre première page en HTML :

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

  <body>

  </body>
</html>
```

Vous reconnaissez déjà certaines balises !!!

Vous devez être capable de mettre un titre à votre page, et d'écrire un petit paragraphe, entre les balises body.

5) Enregistrez votre page ou mieux Ctrl+S.

Puis, ouvrez le navigateur Firefox, et ouvrez votre page (Fichier puis Ouvrir...) Contemplez votre œuvre, à chaque modification dans Notepad enregistrez les modifications (Ctrl+S) puis actualisez la page dans le navigateur (touche F5)

4. Imbriquer des balises

Attention les balises doivent s'imbriquer correctement, par exemple :

<p>Mon premier < strong > paragraphe</ strong ></p> est correct, mais,

<p>Mon premier < strong > paragraphe</p></ strong > ne l'est pas.

CONSEIL : Dans l'éditeur de texte, à chaque fois que vous ouvrez une balise, écrivez tout de suite la balise fermante, pour ne pas l'oublier.

5. Attributs des balises

Pour expliquer le rôle d'un attribut d'une balise, prenons l'exemple suivant :

```
<a href="http://fr.openclassrooms.com/informatique/cours/apprenez-a-creer-votre-site-web-avec-html5-et-css3"> Tutoriel HTML5 d'OpenClassrooms</a>
```

L'attribut de la balise <a> est *href* .

Essayer de rajouter ce code dans votre page pour voir.

Remarquez la syntaxe : <balise attribut="valeur">

Voici quelques caractéristiques d'un attribut :

- Pour affecter une valeur à un attribut, on utilise le signe = (égal)
- Les attributs s'insèrent après l'élément de la balise et avant le chevron (>). Les attributs ne se placent que dans la balise d'ouverture, jamais dans celle de fermeture. Quand il y a plusieurs attributs, ils sont placés les uns à la suite des autres, dans un ordre quelconque et séparés par des espaces.
- La valeur de l'attribut est entourée par des guillemets "

6. Les liens

La balise <a>, avec son attribut href permet de créer un hyperlien vers une autre page. Voici un exemple :

```
<a href="http://www.google.fr">GOOGLE</a>
```

Quelques explications sur cet exemple :

- Dans cet exemple, le terme GOOGLE apparaît sur la page. Lorsque l'utilisateur clique sur le lien, la page <http://www.google.fr> apparaîtra.
- Un lien, nommé GOOGLE, et pointant vers <http://www.google.fr> est créé.
- href est l'attribut de la balise a
- la valeur de l'attribut href donne l'adresse vers laquelle l'utilisateur sera dirigé
- Le texte qui est placé entre la balise <a...> et apparaît sur la page.

7. Les images

La balise permet d'insérer une image dans un document HTML. Cette balise possède un attribut nommé src qui indique où prendre l'image à afficher.

Par exemple, l'instruction suivante

```

```

affiche l'image nommée mon_img.jpg sur la page.

Il faut ajouter l'attribut alt pour être conforme aux recommandations du W3C. Cet attribut permet d'afficher un texte alternatif, si l'image ne s'affiche pas (mal voyant par exemple)

L'attribut title permet d'afficher un texte dans une bulle, lorsque l'on passe la souris sur l'image.

Rq : Il est courant de créer un répertoire « image » pour stocker les images d'un site, c'est ce que je vous recommande. On indique alors le chemin comme ceci :

Rq : Evitez à tout prix les accents, majuscules et espaces dans vos noms de fichiers et de dossiers. Voici un chemin qui va poser problème :

"Images du site/Image toute bête.jpg".

Il faudrait supprimer les espaces (ou les remplacer par le symbole « _ »), supprimer les accents et tout mettre en minuscules comme ceci :

"images_du_site/image_toute_bete.jpg".

Sachez donc que, si votre image ne s'affiche pas, c'est très certainement parce que le chemin est incorrect ! Simplifiez au maximum vos noms de fichiers et de dossiers, et tout ira bien.

8. Combiner les balises et <a>

Nous avons vu que la balise <a> (et son attribut href) permettent de créer un lien Hypertext. En plaçant une balise d'image à la place du texte, nous pouvons faire en sorte que l'image devienne un lien.

Par exemple :

9. A vous.

Insérez une image sur votre page, insérez un lien (n'oubliez pas de créer un répertoire image).

10. Structurer le texte.

On doit indiquer au navigateur, la valeur de chaque partie du texte, on peut par exemple, indiquer que ceci est une abbréviation (<abbr>) ceci est une partie très importante ()

Attention : Le rôle des balises est d'indiquer le *sens* du texte. Ainsi, indique à l'ordinateur « Ce texte est important ». C'est tout.

Et pour *montrer* que le texte est important, l'ordinateur décide de le mettre en gras (mais il pourrait aussi bien l'écrire en rouge !). La plupart des navigateurs affichent les textes importants en gras, mais rien ne les y oblige.

Pour plus de détails : <http://fr.openclassrooms.com/informatique/cours/apprenez-a-creer-votre-site-web-avec-html5-et-css3/la-mise-en-valeur>

Voici quelques balises pour structurer votre texte :

Balise	Description
<abbr>	Abréviation
<blockquote>	Citation (longue)
<cite>	Citation du titre d'une œuvre ou d'un évènement
<q>	Citation (courte)
<sup>	Exposant
<sub>	Indice
	Mise en valeur forte (par défaut en gras)
	Mise en valeur normale (par défaut en italique)
<mark>	Mise en valeur visuelle (par défaut en surligné)
<h1>	Titre de niveau 1
<h2>	Titre de niveau 2
<h3>	Titre de niveau 3

<h4>	Titre de niveau 4
<h5>	Titre de niveau 5
<h6>	Titre de niveau 6
	Image
<figure>	Figure (image, code, etc.)
<figcaption>	Description de la figure
<a>	Lien hypertexte
 	Retour à la ligne
<p>	Paragraphe
<hr />	Ligne de séparation horizontale
<address>	Adresse de contact

Rq : L'élément abbr représente un acronyme ou une abréviation... Comme pour une image, l'attribut title crée une infobulle pour l'utilisateur lorsqu'il survole l'élément <abbr> en question.

Il faut aussi indiquer au navigateur, où sont **les titres, les sous-titres...** pour cela on utilise les balises <h1>, <h2>, <h3>...

11. Les commentaires

Très important les commentaires, pour vous, et pour ceux qui regardent votre code source (sur n'importe quelle page clic droit : afficher le code source de la page). Utiliser les pour bien faire apparaître, les différentes parties de votre code !!

```
<!-- Ceci est un commentaire -->
```

12. Les tableaux

Les tableaux sont créés à l'aide des balises suivantes :

- <table> : définit le tableau
- <tr> : définit une ligne dans le tableau (table row)
- <td> : définit une cellule dans le tableau (table data)

Par exemple, le code suivant produit un tableau de 2 lignes et 3 colonnes (voir 1.4)

```
<table border="1">
  <tr>
    <td>aaa</td>
    <td>b</td>
    <td>ccc ccc</td>
  </tr>
  <tr>
    <td>abcde</td>
    <td>b</td>
    <td>ccc</td>
  </tr>
</table>
```

Figure 1.4 – Un tableau

13. Balises sans contenu

Certaines balises ont un contenu (par exemple une `belle` ville, le contenu de `` étant belle), d'autres n'ont pas de contenu (par exemple `<hr>` (permet de tracer une ligne horizontale), ou encore ``). Pour ces balises sans contenu, plutôt que d'écrire la balise ouvrante et la balise fermante (`<hr></hr>`), on fusionne ces deux balises, pour n'écrire qu'une balise : `<hr/>`, ou `` (où le slash se situe à la fin de la balise)

14. Les balises `` et `<div>`

Ces balises sont importantes pour la mise en forme, nous y reviendrons en lien avec le CSS. Elles s'utilisent avec les attributs `id` et `class`.

On distingue les balises "*block*" qui structurent la page et les balises "*inline*" qui sont plutôt de la mise en forme du texte. `<p>` est une balise « block », `<a>` est une balise « inline »

La balise `span` est une balise « inline », elle n'a aucun effet ! Par exemple, écrire

Un peu de texte

ou

Un `` peu de `` texte

produira exactement le même résultat. Cependant, une telle balise est utile pour délimiter une portion du texte, à savoir le texte présent entre la balise ouvrante `` et la balise fermante ``. Et modifier le style d'affichage dans le CSS. Par exemple :

Un `` peu de `` texte


Et en définissant dans la feuille de style, un style d'affichage particulier pour la class `texte_petit`

De même la balise `<div>` est une balise « block » permet de délimiter une section dans la page (`` à l'intérieur d'un texte, `<div>` pour une partie de la page). Ceci peut-être utile, en liaison avec le CSS.

15. Structurer sa page : (copie du site du zéro)

Ces balises permettent de construire le squelette de notre page web.


Balise	Description
<code><header></code>	En-tête
<code><nav></code>	Liens principaux de navigation
<code><footer></code>	Pied de page
<code><section></code>	Section de page
<code><article></code>	Article (contenu autonome)
<code><aside></code>	Informations complémentaires



ZoZor
Carnets de voyage


<nav>
ACCUEIL BLOG CV CONTACT

<header>



<div>

Retour sur mes vacances aux États-Unis...
Voir l'article ▶



JE SUIS UN GRAND VOYAGEUR


<article>

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris metus massa, luctus in tincidunt a, porttitor ac leo. Maecenas at mi feugiat turpis elementum ornare. Sed tempor rutrum lorem, in vestibulum felis elementum ac. Fusce purus orci, scelerisque ut tincidunt in, dignissim vel augue. Nulla laculis ultrices sagittis. Nulla vitae neque dignissim enim tempor scelerisque at quis tellus. In hac habitasse platea dictumst. Aenean elit elit, pellentesque nec venenatis ut, convallis eu sem. Mauris eu leo nec arcu volutpat euismod nec eu dolor. Morbi aliquet, mauris quis porttitor dapibus, odio enim viverra eros, quis interdum massa urna at velit. Integer tempor facilisis libero non accumsan. Aliquam diam felis, dapibus sed condimentum quis, molestie vel odio. Maecenas eget ante massa, a sagittis quam. Cras posuere magna ac urna molestie vitae luctus lacus lobortis. Quisque leo neque, vulputate at semper non, varius porta enim.

Praesent sit amet lectus eros, ac pellentesque nisi. Donec consequat magna sed libero condimentum vitae aliquet elit ornare. Nunc at nulla purus. Aliquam sit amet sapien sit amet nisi aliquet rutrum vel nec mi. Mauris ultricies felis egestas mi varius molestie molestie sapien tristique. Cras lacus lacus, rutrum id sagittis sit amet, malesuada nec odio. Nulla consectetur lobortis libero, ac convallis massa consectetur in. Nam facilisis posuere sagittis. Sed a ligula id dui vulputate congue quis at tortor. Nunc pellentesque faucibus felis, eu venenatis massa interdum in. Donec venenatis lacus id tortor vestibulum id accumsan est lobortis. Morbi turpis quam, tincidunt in accumsan quis, ullamcorper quis orci. Quisque nisi magna, egestas eget consectetur non, mollis ac ante. Donec elit felis, blandit at auctor in, lacinia et dolor.






Ut blandit, diam id aliquam volutpat, quam libero euismod neque, ut volutpat nunc ipsum a magna. Donec hendrerit sem in dolor egestas lobortis. Etiam bibendum lobortis interdum. Etiam ac felis vitae neque sodales sodales. Nunc tempus dignissim dapibus. Duis sit amet tellus vitae elit suscipit convallis. Sed et tincidunt velit. Donec congue elementum ante eu consectetur. Morbi lectus mauris, sodales a euismod id, dapibus sollicitudin urna. Sed sagittis sagittis placerat. Etiam at lorem risus. Quisque imperdiet elementum tortor nec viverra. tempus dignissim dapibus. Duis sit amet tellus vitae elit suscipit convallis. Sed et tincidunt velit. Donec congue elementum ante eu consectetur. Morbi lectus mauris, sodales a euismod id, dapibus sollicitudin urna. Sed sagittis sagittis placerat.

À PROPOS DE L'AUTEUR



Laisse-moi le temps de me présenter : je m'appelle ZoZor, je suis né un 23 Novembre 2005.

Bien maigre, n'est-ce pas ? C'est pourquoi, aujourd'hui, j'ai décidé d'écrire ma biographie (ou zBiographie, comme vous voulez !) afin que les zéros sachent enfin qui je suis réellement.

<aside>

<section>

MON DERNIER TWEET <div>

Hii haaaaaan !

le 12 mai à 23h12

MES PHOTOS



MES AMIS

→ Pupi le lapin

→ Mr Baobab

→ Kaiwall

→ Perceval.eu

→ Belette

→ Le concombre masqué

→ Ptit prince

→ Mr Fan

<footer>

16. Remarques

Les commentaires

Un commentaire est une balise HTML dont la syntaxe est : `<!-- commentaire -->`

Vous pouvez le mettre où vous voulez dans votre code source : il n'apparaît pas sur votre page mais il vous sert à vous repérer dans votre code. Il est également très utile à ceux qui lisent ledit code.

Règles à respecter

Avec les navigateurs actuels (Internet explorer, Firefox, Chrome, etc...), même lorsque le code HTML contient certaines erreurs, la page s'affichera malgré tout correctement. Cependant, certains supports n'ont pas les ressources nécessaires pour interpréter du code HTML contenant des erreurs. Il s'agit par exemple des pages destinées à être vues sur un téléphone portable.

Voici donc certaines règles qu'il convient de respecter, même si les navigateurs actuels afficheront correctement le résultat :

- l'imbrication des balises doit être correct
- les balises doivent toujours être fermées : `` et pas ``

- les noms des balises et leurs attributs s'écrivent en minuscule : `` et pas ``
- les valeurs des attributs sont toujours encadrées par des guillemets ou des apostrophes : `` et pas ``

17. Compléments sur HTML

Lors de la création d'un page Internet en HTML, quelques règles s'imposent quand à la présentation du code HTML et à la marche à suivre pour créer cette page.

Marche à suivre

- la structure de la page doit être définie AVANT de commencer à écrire du code. Cela ne veut pas dire que l'on ne peut pas changer cette structure par la suite pour y apporter des améliorations.
- avant de mettre du texte et des images sur la page, commencer par créer un squelette, c'est-à-dire par écrire les tags HTML sans texte ainsi que les informations de taille pour que cela ressemble le plus possible à la structure de la page finale.

Par exemple :

- le code doit être présenté de manière lisible. Une bonne habitude à prendre est d'indenter, c'est à dire décaler, le code comme ci-dessus. Ainsi les balises ouvrantes et fermantes se trouvent alignées dans une même colonne ce qui rend le code plus lisible et donc permet de trouver plus facilement les erreurs. Cela devient particulièrement important lorsque vous avez des milliers de lignes de code et que plusieurs personnes travaillent sur un même projet !
- une fois la structure de la page définie, introduire le texte et les images.
- le style vient en dernier. Les couleurs, les types de polices de caractère, leur taille, etc se met tout à la fin. Nous verrons d'ailleurs qu'avec CSS, le style est en général stocké dans un fichier séparé qu'on ne remplit qu'une fois le fichier html terminé.
- par convention, les noms de fichiers contenant du code html se terminent par l'extension .html. Par exemple : index.html.
- Stocker les fichiers dans un répertoire dont le nom décrit le site sur lequel vous travaillez. Par exemple pour l'exercice HTML sur le site du CERN :

CERN_html/

Il est courant de stocker les images dans un répertoire séparé, souvent nommé images :

CERN_html/images/

Ainsi, quand vous utilisez une balise image, il vous faudra écrire, par exemple :

```

```

17) La validation par le W3C

Pourquoi faut-il valider sa page web?

Lorsqu'il rencontre une erreur, un navigateur la "corrige", mais rien n'assure que d'autres navigateurs n'en auront pas une autre interprétation.

De plus un code correct rend votre page accessible, par exemple aux mal-voyants.

Un site sur ce sujet : [AccessiWeb](#)

Vous avez fini votre première page HTML. Il vous reste à valider votre travail sur le site du W3C. Le W3C propose un programme qui analyse votre code source et vous indique s'il est correct ou s'il comporte des erreurs. Vous le trouvez à cette adresse :

<http://validator.w3.org>

Si tout va bien, le validateur vous affichera :

A green rectangular box with white text that reads "This document was successfully checked as HTML5!".

Mais c'est rarement le cas dès le départ : il reste toujours une balise mal fermée, une structure à revoir ...

Si le code de votre page n'est pas correct, le validateur vous précisera quelles sont les lignes à corriger, ainsi que la nature de l'erreur. Vous pourrez ainsi corriger votre code pas à pas, jusqu'à obtention de la ligne verte ci-dessus.