

HTML 5 et CSS 3 (partie 1)

Objectifs

- Connaître les différences au niveau des balises et des attributs,
- Créer un document HTML5,
- Définir une feuille de style.

Quelques règles simples qu'il faut respecter lors de l'écriture des balises HTML5 :

- En HTML5, les balises ne sont pas sensibles à la casse. On peut écrire indifféremment <BALISE>, <Balise> ou <balise>.
- La règle générale veut que toute balise ouverte <balise> doit être fermée </balise>. Les libertés prises dans l'écriture du HTML 3.2 à cause du fonctionnement plus ou moins permissif des navigateurs ne sont plus d'actualité. La rigueur apportée par le HTML 4.0 strict et le XHTML 1.0 doit continuer à guider votre écriture.
- Les balises doivent être correctement imbriquées. Lorsqu'on affecte plusieurs balises à un élément, l'ordre de fermeture de celles-ci est essentiel. La première balise de fermeture doit correspondre à la dernière balise d'ouverture non fermée.
- Les valeurs des attributs doivent toujours figurer entre des guillemets.

Les balises et attributs HTML 4.0 disparus :

- Disparition complète des cadres. Les balises <frame>, <frameset> et <noframes> ont définitivement disparu.
- Disparition de certaines balises de présentation comme <big>, <center>, , <strike>, <tt> et <u>. Ici aussi, pas de grande surprise car le principe de la séparation de contenu et de la présentation est à présent bien établi. Ces balises de présentation doivent maintenant être prises en charge par des feuilles de style CSS.
- Disparition de tous les attributs relatifs à l'alignement, la largeur, les arrière-plans (de couleur ou avec image), les bordures et la numérotation des listes qui sont passés à la trappe. Tous ces attributs doivent être pris en charge par les feuilles de style qui deviennent ainsi indissociables du code HTML5.

Exemples :

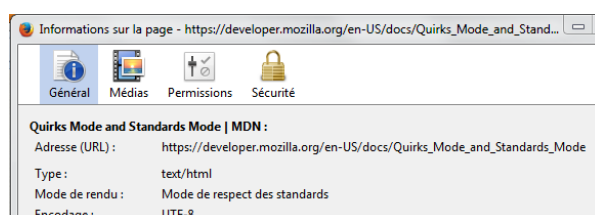
- align dans les balises <caption>, <iframe>, , <input>, <object>, <legend>, <table>, <hr>, <div>, <hx>, <p>, <col>, <colgroup>, <tbody>, <td>, <tfoot>, <th>, <thead> et <tr>.
- alink, link, text et vlink dans la balise <body>.
- background dans la balise <body>.
- bgcolor dans les balises <table>, <tr>, <td>, <th> et <body>.

Le document HTML5 minimum

Tout document HTML doit commencer par un **doctype**. Le HTML5 propose un doctype unique et simplifié :
(<!DOCTYPE html>)

Le doctype, aussi appelé DTD sert à indiquer au navigateur à quelles règles d'écriture obéit le code source de la page HTML ou XHTML. Il utilisera celles-ci pour afficher la page selon les standards du W3C. Tous les navigateurs appliquant la même règle, vous êtes ainsi assuré d'un rendu identique entre les différents navigateurs.

En l'absence d'un doctype, le navigateur ignore selon quelles règles il doit traiter la page. Il se rabat ainsi sur des procédures qui lui sont propres pour afficher votre page HTML. Ce qui peut entraîner des différences de restitution sensibles entre les différents navigateurs. Ce mode bancal est appelé **mode compatibilité** ou **quirks mode**. Avec Firefox, cliquer sur le **bouton droit** de la souris puis **informations sur la page**.



La balise `<html>` indique au navigateur qu'il s'agit d'un document Html. La balise `<html>` est l'élément le plus haut ou l'élément racine du document. Elle prend donc place juste après la déclaration de doctype.

```
<!DOCTYPE html>
<html lang="fr">
...
</html>
```

La balise `<html>` peut prendre comme attribut `lang="fr"` qui spécifie que le document est en langue française. Cette information est fort prisée par les moteurs de recherche comme Google ainsi que par les synthèses vocales utilisées par les personnes malvoyantes.

La balise `<head> ... </head>` qui se positionne juste après le doctype et la balise `<html>` contient ce qu'on appelle l'en-tête du document. Dans cet en-tête du document se situe toute une série d'informations relatives au document lui-même, indépendamment du contenu de celui-ci. Ces informations peuvent être de natures très diverses :

- Le titre du document : la balise `<title> ... </title>`.
- Des déclarations ou des appels de feuilles de style CSS.
- Des fonctions ou des liens vers des fichiers JavaScript.
- Des informations à l'intention des moteurs de recherche pour le référencement de la page.
- Des informations à l'intention des navigateurs.
- Des balises méta qui reprendront quant à elles l'encodage, la description de la page, les mots-clés associés à la page, le nom de l'auteur, des mentions de copyright, etc.

L'encodage du document

La notion du jeu de caractères utilisé (**charset**) est une notion essentielle dans le développement de pages Web. Le jeu de caractères désigne la façon dont les caractères d'un alphabet donné sont convertis en octets dans un fichier informatique (et vice-versa). Certaines méthodes d'encodage sont spécifiques à un environnement informatique dans une langue ou un alphabet donné, d'autres sont multiplateformes et multilingues.

ASCII	Jeu de caractères basique mais relativement ancien qui a permis l'échange de textes en anglais à un niveau mondial. Il a été repris à l'origine de la toile mais comme l'anglais ne comporte pas d'accents, il fallait encoder les caractères accentués et spéciaux par des entités du genre ´ (pour le é) ou € (pour l'€).
ISO-8859-1	Extension de l'ASCII, l'ISO-8859-1, souvent appelée Latin-1 ou Europe occidentale, est le codage fréquemment utilisé dans notre région du monde. Il ajoute de nombreux caractères latins ainsi que quelques autres signes de ponctuation ou symboles.
ISO-8859-15	Version actualisée de l'ISO-8859-1 qui comporte entre autres le signe €.
UTF-8	C'est un format de codage de caractères (charset) conçu pour coder l'ensemble des caractères internationaux d'Unicode de plusieurs langues, contrairement aux formats plus vieux comme l'iso-8859-1 qui requiert une forme d'encodage afin d'accepter, par exemple, des lettres accentuées ou des caractères spéciaux. Son principal avantage pour le web est qu'il permet de représenter les milliers de caractères des différentes langues et permet donc de créer des sites Web qui puissent être multilingues (français, chinois ou arabe,...) sans avoir à créer de multiples versions du site dans l'encodage du pays concerné.

Depuis fin 2011, le WC3 recommande le charset UTF-8 pour l'édition de pages Web en grande partie pour son universalité. Les navigateurs possèdent bien une fonction de détection automatique du jeu de caractères.

Voici donc le document HTML5 minimum :

```
1 <!doctype html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8">
5   <title>Titre de la page</title>
6   <link rel="stylesheet" href="style.css">
7   <script src="script.js"></script>
8 </head>
9 <body>
10  ...
11  <!-- Le reste du contenu -->
12  ...
13 </body>
14 </html>
```

La validation du code HTML5

Les avantages d'un code parfait et de la validation sont :

- Une vérification en profondeur du HTML5 pour les codeurs débutants (et confirmés).
- L'assurance que votre page sera toujours correctement affichée par la plupart des navigateurs. Une page invalide peut amener les navigateurs à interpréter celle-ci de manière très différente.
- La démonstration de vos capacités professionnelles en fournissant un code de qualité, conforme au standard du HTML5.
- Les synthèses vocales et autres aides techniques à l'attention des personnes moins valides se basent sur un code valide pour restituer les pages de la toile.

Si le document soumis au validateur n'est pas correct, celui-ci vous fournira une analyse des problèmes qui vous permettra de corriger votre code. Les premières validations peuvent être très frustrantes mais constituent un excellent apprentissage du code source. Le rapport d'erreur est cependant souvent très technique.

- Le **W3C** lui-même présente un validateur de code (<http://validator.w3.org/>).
- Le site **validator.nu** propose également au stade expérimental un validateur HTML5 (<http://html5.validator.nu/>).

Exercice 1 – Validation d'une page HTML5

1. Ouvrir la page **erreur.html** et repérer les erreurs de codage.
2. Valider la page **erreur.html** sur le site du **W3C** puis sur le site **validator.nu**.
3. Les validateurs détectent-ils les erreurs ?
4. Corriger les erreurs.

Exercice 2 – Le texte

Pour mettre du texte en gras (bold en anglais), il suffit d'inclure celui-ci entre les balises ` ... `.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <meta charset="UTF-8">
6 </head>
7 <body>
8 <b>Du texte en gras</b>
9 </body>
10 </html>
```

1. Mettre en gras avec une feuille de style CSS sans passer par la balise ``.
2. Procéder de même pour du texte en italique, la taille et la couleur des caractères, l'alignement et le soulignement d'un mot.

Exercice 3 - La structuration d'un document

Le langage HTML5 propose six grandeurs de caractères pour les titres (balise `h<x>`, `x` varie de 1 à 6). Le W3C insiste pour qu'en HTML5, les balises de titre `<h<x>` retrouvent toutes leurs qualités de structuration des pages Web. L'usage des balises de titres doit donc être réservé à leur fonction originelle. Il importe de ne pas les détourner de leur but premier pour, par exemple, mettre du texte en gras ou avec une taille de caractères déterminée. Les feuilles de style CSS sont là pour réaliser ces présentations.

1. Ecrire le code pour indenter les titres de niveau 1 à 3.
2. Valider le code.

Exercice 4 – Les divisions de page

Il est possible de diviser en bloc et en ligne des éléments. Il existe deux grands groupes de balises : les balises de rendu **CSS bloc** (block) et les balises de rendu **CSS en ligne** (inline).

Les balises de rendu CSS bloc occupent par défaut toute la largeur de la fenêtre du navigateur. Elles se placent ainsi les unes au-dessus des autres. Généralement, le navigateur insère automatiquement un espace au-dessus et en dessous du contenu de la balise. C'est le cas de la balise de titre `<h1>` et de toutes les balises de ce chapitre. Elles sont toutes des balises de type bloc.

Une balise de titre ou de paragraphe occupe toute la largeur de la page et est séparée par un espace plus ou moins important.

Le code `<p>Paragraphe 1</p><p>Paragraphe 2</p>` va s'afficher sur deux lignes car la balise `<p>` est une balise de type bloc. Chaque paragraphe va occuper une ligne.

Les principaux éléments de division de type bloc sont :

- Les balises de titres `<h1> ... </h1>`.
- Les balises de paragraphe `<p> ... </p>`.
- Les balises de citation `<blockquote> ... </blockquote>`.
- La balise `<hr>` qui introduit une ligne horizontale.
- Les balises ` ... ` des listes ordonnées.
- Les balises ` ... ` des listes à puces.
- Les balises `<dl> ... </dl>` des listes de définitions.
- Les balises de tableaux `<table> ... </table>`.
- Les balises de déclaration de formulaires `<form> ... </form>`.
- La balise `<div> ... </div>` spécialement conçue pour introduire une division (div pour division) sans passer par une des balises citées ci-avant. Cette balise est particulièrement utile pour appliquer une déclaration de feuilles de style CSS.

Au contraire des éléments de type bloc, les éléments en ligne se placent toujours l'un à côté de l'autre afin de rester dans le flux du texte. Ainsi, le code texte en `gras` et en `<i>italique</i>` est restitué sur une seule ligne, sans interrompre le flux du texte : texte en gras et en italique.

Les principales balises en ligne sont :

- La balise ` ... ` de mise en gras du texte.
- La balise `<i> ... </i>` de mise en italique du texte.
- La balise `
` de passage à la ligne.
- La balise `<a>` pour les liens.
- La balise `` pour les images.
- Les balises de champs de formulaires `<input>`, `<select> ... </select>` et `<textarea> ... </textarea>`.
- La balise `` pour introduire une division en ligne dans le texte. Cette balise est particulièrement utile pour appliquer une déclaration de feuilles de style CSS.

Une balise de type bloc peut contenir une (ou plusieurs) balise(s) de type bloc. Ainsi, par exemple, une citation incluse dans un paragraphe. L'extension **Web Developer** de Firefox permet de visualiser les différents éléments bloc d'une page et par conséquent la structure de celle-ci (<https://addons.mozilla.org/fr/firefox/addon/web-developer/>).

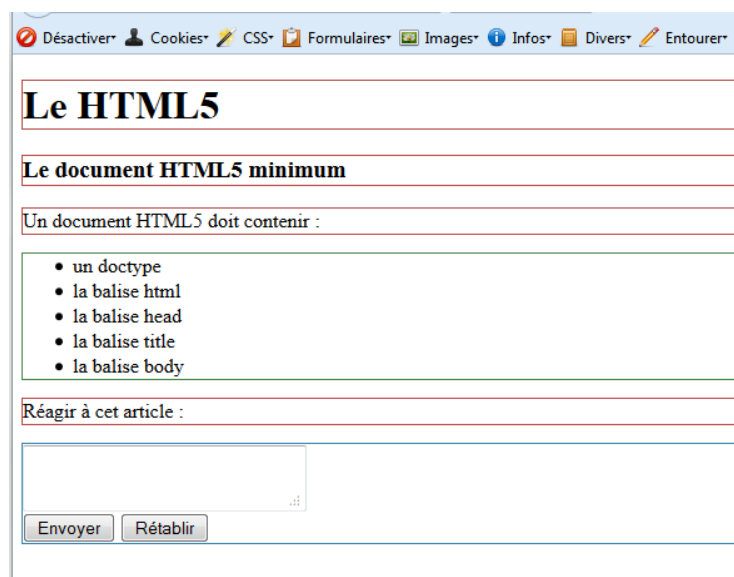
TP4 – HTML 5 et CSS 3 (partie 1)

1. Reproduire la page HTML ci-dessous :

```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4  <title>Html5</title>
5  <meta charset="UTF-8">
6  </head>
7  <body>
8  <h1>Le HTML5</h1>
9  <h3>Le document HTML5 minimum</h3>
10 <p>Un document HTML5 doit contenir :</p>
11 <ul>
12 <li>un doctype</li>
13 <li>la balise html</li>
14 <li>la balise head</li>
15 <li>la balise title</li>
16 <li>la balise body</li>
17 </ul>
18 <p>Réagir à cet article :</p>
19 <form action="">
20 <textarea rows="2" cols="25"></textarea><br>
21 <input value="Envoyer" type="submit">
22 <input value="Rétablir" type="reset">
23 </form>
24 </body>
25 </html>

```

2. Installer l'addon **Web-Developer** de Firefox et entourer les différents éléments de type blocs de la page.

Exercice 5 – Les liens

Les liens constituent l'essence du langage Html et des pages Web. La richesse de la fonction hypertexte permet de tisser cette toile gigantesque qu'est le Web.

La balise `<a> ... ` introduit un lien. Sa syntaxe de base est : `Texte de lien`

La destination du lien peut être :

- Un endroit de la page en cours.
- Une autre page du site.
- Un endroit dans une autre page du site.
- Une page d'un autre site situé sur le Web.
- Une adresse électronique.
- Un fichier à télécharger.

La balise de lien possède différents attributs :

- href
L'attribut **href** définit l'adresse (url) de la cible du lien.
- hreflang
Précise la langue du document cible si celle-ci est différente de celle du document de départ.

TP4 – HTML 5 et CSS 3 (partie 1)

- ping
Nouveau en HTML5. L'attribut ping peut contenir une liste d'adresses url (séparées par un espace) qui recevront une notification lorsque l'utilisateur suit le lien.
`Html5`
 Le navigateur va alors envoyer une requête POST à l'adresse url spécifiée dans l'attribut. Ce nouvel attribut sera particulièrement utile pour les statistiques d'un site. Dans le cas présent, il permettra de connaître le nombre de visiteurs de la partie consacrée au Html5.
- rel
Nouveau en HTML5. Spécifie la relation entre le document de départ et le fichier cible du lien.
 Les valeurs sont nombreuses : alternate, archives, author, bookmark, contact, external, first, help, icon, index, last, license, next, nofollow, noreferrer, pingback, prefetch, prev, search, stylesheet, sidebar, tag, up.
 Cet attribut n'est que très partiellement implémenté dans les navigateurs récents.
- target
 Spécifie au navigateur la façon d'afficher la cible du lien. Cela peut être dans une nouvelle instance du navigateur ou un nouvel onglet du navigateur (target="_blank"), dans la même fenêtre que la page de départ du lien (target="_self") et dans la même fenêtre mais en occupant la totalité de la fenêtre du navigateur.
- type
 Indique au navigateur le type MIME¹ de la cible si celui-ci n'est pas un document HTML5, par exemple un fichier son ou une image.

Les attributs HTML 4.0 suivants ont disparu de la spécification HTML5 : charset, coords, name, rev et shape.

Liens vers une ancre située dans la même page

Après avoir défini l'ancre, il faut effectuer le lien vers celle-ci. Pour créer ce lien, il suffit de reprendre dans l'attribut href le nom de l'ancre directement précédé du signe dièse (#) : `Texte du lien`

```

1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4  <title>Html5</title>
5  <meta charset="UTF-8">
6  </head>
7  <body>
8  <a href="#chapitre2">Aller au chapitre 2</a>
9  <h3>Chapitre 1</h3>
10 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
11 non risus. Suspendisse lectus tortor, dignissim sit amet,
12 adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
13 diam...</p>
14 <h3>Chapitre 2</h3>
15 <a id="chapitre2"></a>
16 <p>Contenu du chapitre 2</p>
17 </body>
18 </html>

```

1. Réaliser un lien vers une ancre situé dans une seconde page.
2. Réaliser un lien pour télécharger le fichier pdf de ce TP.
3. Réaliser un lien pour ouvrir le site <https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5> dans un nouvel onglet.
4. Ajouter une infobulle sur le lien précédent.
5. A l'aide d'une feuille de style, enlever le soulignement des liens (text-decoration: none) et uniformiser la couleur des liens (propriété de style color).

Les liens peuvent maintenant porter sur n'importe quel élément de type bloc comme les divisions (<div>) et items de liste ().

6. Réaliser le code ci-dessous et modifier-le pour que le lien s'ouvre dans un nouvel onglet. Conclure.

¹ Un Internet media type1, à l'origine appelé type MIME ou juste MIME ou encore Content-type2, est un identifiant de format de données sur internet en deux parties. Un type et un sous-type et d'un ou plusieurs autres champs au besoin. Par exemple, les sous-types du type text ont un champ optionnel charset indiquant le codage des caractères ; ou encore, les sous-types du type multipart définissent une option boundary entre les parties.

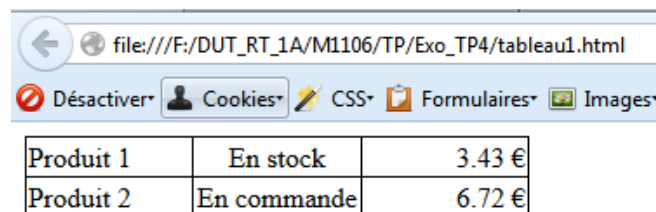
```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <title>Html5</title>
5 <style>
6 div { width: 280px;
7       height: 180px;
8       border: 1px solid black;
9       padding-left: 3px;}
10 a { text-decoration: none;
11    color: black;}
12 </style>
13 <meta charset="UTF-8">
14 </head>
15 <body>
16 <p>Cliquez sur l'extrait pour le texte complet.</p>
17 <a href="division.html">
18 <div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
19 non risus. Suspendisse lectus tortor, dignissim sit amet,
20 adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
21 diam...</div>
22 </a>
23 </body>
24 </html>
```

Exercice 6 – Les tableaux

Un tableau (<table>) comporte des lignes <tr>. Ces lignes contiennent des cellules <td>. Les cellules peuvent contenir tous les éléments définis par le HTML, soit du texte, des images, des liens, des arrière-plans et même des tableaux.

En HTML5, les attributs width, height, align, valign, abbr, axis et scope du HTML 4.0 ont disparu.

1. Réaliser le tableau ci-dessous. La mise en forme du tableau est réalisée par une feuille de style interne.



Produit 1	En stock	3.43 €
Produit 2	En commande	6.72 €

2. Ajouter un arrière-plan de couleur pour les cellules des prix.

La **balise <caption>** permet d'associer un titre ou une légende au tableau. La balise <caption> doit être placée juste après la balise ouvrante <table> et ne peut apparaître qu'une fois dans le tableau. Par défaut, la plupart des navigateurs affichent le contenu de la balise <caption> de façon centrée au-dessus du tableau. Il est possible de mettre cette légende en dessous du tableau à l'aide de la propriété de style CSS caption {caption-side: bottom;}.

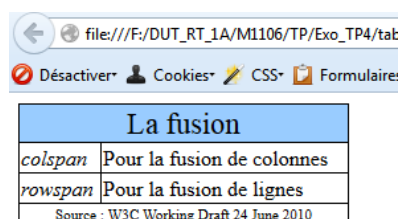
3. Insérer la légende suivante « Tarifs des produits festifs » en-dessous du tableau.

Le HTML5 prévoit des balises pour structurer de façon logique le contenu d'un tableau. Ces balises sont :

- **<thead>** pour regrouper les informations concernant l'en-tête du tableau comme par exemple le titre et l'intitulé des colonnes.
- **<tbody>** pour le corps du tableau, soit l'ensemble des données de celui-ci.
- **<tfoot>** pour le contenu de bas de page comme par exemple des remarques ou une légende.

Ces divisions logiques n'affectent en rien la présentation par défaut du tableau mais elles peuvent être reprises par des propriétés de style pour agrémenter la présentation du tableau. La balise <tfoot> doit se situer avant la balise <tbody> de sorte que le navigateur puisse prévoir le pied de page avant la réception des lignes de données.

4. Construire le tableau ci-après en utilisant les balises précédentes.



La fusion	
colspan	Pour la fusion de colonnes
rowspan	Pour la fusion de lignes
Source : W3C Working Draft 24 June 2010	

Exercice 7 – Les images et arrière-plans

Pour réduire au maximum le temps de téléchargement, le Web réclame des formats de fichier particuliers ; le format GIF, JPEG et PNG. Ces trois types de format comportent tous un algorithme de compression.

▪ Le format GIF

Le **format GIF** (pour **Graphics Interchange Format**) a connu son heure de gloire aux débuts du Web. Il était alors parfaitement adapté aux cartes graphiques de l'époque (souvent maximum 256 couleurs) et aux faibles capacités des lignes de téléphone analogiques (taux de compression élevé).

En raison de sa limitation à 256 couleurs, le format GIF n'est pas du tout approprié aux graphiques à haute résolution comme les photos, les images nuancées et les dégradés. Par contre, il excelle encore pour les petits logos, boutons, puces, barres, symboles, et autres cliparts fréquents pour certains graphiques du Web qui ne nécessitent pas une palette de couleurs très étendue. Il est cependant de plus en plus supplanté par le format PNG.

▪ Le format JPEG

Le **format JPEG** figure pour **Joint Photographic Expert Group**, donc non pas pour le nom du format graphique mais pour le nom de la corporation qui a développé ce format. C'est le type de format le plus répandu dans l'univers de la photo numérique.

Le JPEG est un excellent format pour les photos, spécialement celles où il y a des tonalités diverses et des dégradés de couleurs. Mais il n'est pas exempt de défauts avec ses pertes de données et ses déformations.

À cause des pertes de données et des déformations, il est impératif de respecter la règle de sauvegarder l'image originale et de travailler uniquement sur des copies de celle-ci.

▪ Le format PNG

Le **format PNG**, pour Portable Network Graphic, est présenté comme le format du futur. Il s'agit d'un format graphique conçu spécialement pour la mise en œuvre sur le Web. La recommandation du W3C concernant le PNG remonte à 1996. Le PNG doit théoriquement rassembler tous les avantages de GIF et du JPEG. Il commence à être largement retenu par les concepteurs d'applications Web.

Le format PNG est appelé à être de plus en plus utilisé à l'avenir car il représente un excellent compromis entre le format GIF et JPEG. Les professionnels l'adoptent largement en lieu et place du GIF et pour les petites photos.

▪ Le format WebP

Le **WebP** est un nouveau format d'image qui risque de prendre de l'importance dans les prochaines années. Ce format est développé par Google à partir d'un des formats de la nouvelle balise <video> du Html5 (le WebM). Selon les premiers essais, le WebP réduirait la taille des fichiers de 39 % par rapport aux formats JPEG, PNG et GIF, sans perte de qualité perceptible.

Le poids des photos

Il semble illusoire de charger directement une photo de votre appareil photo numérique sur un site. Certaines "pèsent" jusqu'à 4 Mo, ce qui, même avec une connexion rapide, prendra un temps certain de chargement.

On peut conseiller :

- De recadrer les images pour ne recueillir que ce qui est utile.
- De redimensionner cette image recadrée.
- D'adopter un taux de compression d'environ 15 à 20 % pour les images JPEG. La perte de qualité n'est quasiment pas perceptible à l'œil normal.
- De diminuer le nombre de couleurs de l'image, si l'opération s'y prête.

Ces différentes manipulations devraient ramener la taille du fichier à une dimension beaucoup plus raisonnable et améliorer sensiblement le temps de téléchargement.

L'insertion d'une couleur d'arrière-plan


Depuis la disparition de l'attribut bgcolor, il n'est plus possible d'ajouter un arrière-plan de couleur à un élément. Il faut impérativement passer par les propriétés de style CSS, **background-color** en l'occurrence.

L'insertion d'une image d'arrière-plan

L'attribut background étant déclaré obsolète en HTML5, il n'y a d'autre solution que de passer par la propriété de style **background-image** pour insérer une image d'arrière-plan.

1. Réaliser le tableau ci-dessous.
2. Positionner le logo HTML5.
3. Bonus : insérer en arrière-plan le logo DUT Réseau-Télécom

Module M1106

UE 11	Développement et administration des services réseaux	Volume horaire : 30 h
Référence du module M1106 (T3-12)	Nom du module Initiation au développement Web	Semestre 1
Objectif du module : <ul style="list-style-type: none"> • Homogénéiser les notions abordées pour le Bac • Découvrir le contenu et le contenant d'une page Web qui présente des informations sur Internet pour PC, Tablette et Smartphone • Découvrir les notions de base sur le fonctionnement du modèle client-serveur • Savoir écrire et modifier des pages Web dans un langage normalisé de description de contenus • Savoir modifier les attributs de présentation d'une page Web • Savoir adapter un contenu Web pour navigateur mobile 		
Compétences visées : <ul style="list-style-type: none"> • Être capable de formater et de modifier des documents dans un langage normalisé de description de contenus • Être capable de mettre en œuvre des feuilles de styles • Être capable de mettre en œuvre le Responsive Design 		
Pré requis : <ul style="list-style-type: none"> • Bac ou équivalent 		
Contenus : <ul style="list-style-type: none"> • Langage normalisé de description de contenus Web <ul style="list-style-type: none"> ○ Un document minimal – structure et gestion de l'encodage ○ Les éléments structurant un document ○ Les éléments multimédias ○ Les formulaires: types d'acquisition; attributs ○ Les attributs événements • Feuilles de styles • Responsive Design 		
Modalités de mise en œuvre : <ul style="list-style-type: none"> • Pédagogie par projet 		
Prolongements possibles : <ul style="list-style-type: none"> • Web dynamique (M2105) • Application informatique dédiée aux R&T (M4207C) 		
Source : PPN DUT R&T 2013		

Complément sur les unités

Les unités **em** et **ex** dépendent de la police de caractères et peuvent être différentes pour chaque élément du document. L'unité **em** est simplement la taille de la police de caractères. Pour un élément dont la police est de taille 2in, 1em signifie 2in. Exprimer des tailles, comme les **margins** et les **padding**s, en **em** signifie qu'elles sont relatives à la taille de la police, et si l'utilisateur a une large police de caractères (sur un grand écran par exemple) ou au contraire une petite police (sur un smartphone par exemple), les tailles seront en proportion.

<http://www.w3.org/Style/Examples/007/units.fr.html>

Références

<https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5>

<http://www.w3.org/TR/html51/>