

Table des matières

1.	Contexte et historique de HTML	1
2.	Un langage de balises	1
3.	Exercice 01 – Mise en route	2
4.	Le langage HTML	2
5.	Exercice 02 : Contenu de la première page.....	3
6.	Exercice 03 : Design de la page	5
7.	Exercice 04 : Style du bandeau.....	6
8.	Exercice 05 : Style du menu.....	7
9.	Exercice 06 : Style de la « présentation du site ».....	7
10.	Exercice 07 : Créer des liens	8
11.	Exercice 08 : Liens vers d'autres pages	8
12.	Exercice 09 : Créer une autre feuille de style pour le même fichier html.....	9
13.	Quelques principes css	9
14.	Pour aller plus loin.....	10

1. Contexte et historique de HTML

Le langage HTML (Hyper Text Markup Language) fait partie d'une classe plus vaste de langages (SGML, XML, XHTML,...) destinés à décrire des documents et leur mise en page tout en distinguant la forme et le fond.

Il a été inventé par Tim Berners-Lee en 1991 et a d'abord été utilisé par des physiciens du CERN (Organisation européenne pour la recherche nucléaire) afin de communiquer leurs travaux.

Il est à la base du WEB, la partie actuellement la plus utilisée d'internet.

Tim Berners-Lee suit encore aujourd'hui avec attention l'évolution du Web et a créé le World Wide Web Consortium (W3C) qui définit les nouvelles versions des langages et les normes à mettre en œuvre. Il a par ailleurs créé plus récemment la World Wide Web Foundation, qui analyse et suit l'évolution du Web.

2. Un langage de balises

HTML est un langage de balises (appelées aussi tags) permettant de décrire la mise en page et la forme d'un contenu rédigé en texte simple.

Une balise est un nom encadrée par le caractère inférieur (<) et le caractère supérieur (>).

Exemple : <H1>.

Les balises HTML fonctionnent par paire afin d'agir sur les éléments qu'elles encadrent. La première est appelée « *balise d'ouverture* » (parfois *balise ouvrante*) et la seconde « *balise de fermeture* » (ou *fermante*). La balise fermante est précédé du caractère / (*barre oblique*) :

Exemple : Ce texte est en gras

Certaines balises peuvent parfois être uniques (les balises orphelines)

Exemple : la balise
 représente par exemple un retour à la ligne.

Liste des balises existantes : <https://www.codeshttp.com/toutes-les-balises-html-liste-complete-avec-syntaxe.html>

3. Exercice 01 – Mise en route

1. Ouvrez le logiciel de traitement de texte libre office (**LibreOffice Writer** (**obligatoirement**)) et tapez un texte comprenant un titre (**Option Style > Titre 1**) et deux paragraphes, l'un comprenant un mot en rouge et l'autre comprenant un mot en gras.
2. Faites **Fichier > enregistrer sous** en **choisissant le format ".html" (option Type)**
3. Chargez le fichier HTML sauvegardé dans un navigateur.
4. Faites un clic-droit sur le texte et sélectionnez "**Afficher le code source de la page**". Les éléments de texte et de formatage sont délimités par des balises.

Explication ci-dessus A REFAIRE avec des explications détaillées car peu savent se servir de libre office !!!

Q1 : Quelles balises délimitent votre titre ?

Q2 : Quelles balises délimitent vos paragraphes ?

Q3 : Comment est codée la couleur rouge de votre texte ?

4. Le langage HTML

Un fichier au format HTML (version 5) est avant tout destiné à être lu à l'aide d'un navigateur internet mais peut-être souvent en partie interprété par un traitement de texte comme nous venons de le constater.

La structure typique d'un document HTML (version 5) est la suivante :

L'en-tête

<!DOCTYPE html >

<html lang="fr">

<head>

```
<title>Exemple de HTML</title >
</head>
<body>
  <p>Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.</p>
  <p>Ceci est un paragraphe où il n'y a pas d'hyperlien.</p>
</body>
</html>
```

La première ligne est une description du format du document.

Le reste du document séparé en un en-tête et un corps encapsulés par les balises ou tags `<html>` et `</html>`.

Les balises les plus courantes sont `<p>` et `</p>` qui délimitent un paragraphe et les plus importantes `<a>` et `` qui définissent un lien hypertexte c'est-à-dire un renvoi vers une autre partie du document courant ou vers un autre document.

Bien qu'il soit possible de transformer un document HTML en une véritable soupe de tags assez indigeste (comme le font souvent les traitements de texte), de bonnes habitudes et de la rigueur permettent d'écrire des documents lisibles dont le fond est relativement séparé de la forme.

On écrira ainsi dans des fichiers HTML le contenu et les tags relatifs à la structure, et dans un fichier CSS (Cascading Style Sheet) les informations relatives au style. On ne se lancera donc pas dans la rédaction d'un document HTML sans avoir réfléchi avant à sa structure et son découpage en sous-partie. Cette réflexion pourra être conduite au brouillon et à l'aide de schémas. Dans la suite nous construirons petit à petit les bases d'un site internet présentant des recettes de cuisine.

5. Exercice 02 : Contenu de la première page

Structurer un document

Nous allons respecter une structure extrêmement classique pour un site internet : un bandeau horizontal en haut donnant l'identité du site (titre, logo), un menu latéral sur le côté gauche (pour la navigation dans la base de données des recettes) et un bloc droit destiné à recevoir les informations plus variables (ici les recettes mais aussi les informations de présentation du site). Il faut donc prévoir dès l'écriture d'une page sa structure en différentes **divisions** : en-tête, menu, section, article, etc...

Approfondir : <https://www.alsacreations.com/article/lire/1376-html5-section-article-nav-header-footer-aside.html>

Espace de travail

Vous aurez besoin de trois « espaces de travail » :

- Un éditeur de texte pour coder vos pages (Bloc-note, **Notepad ++**, ...)
- Un navigateur pour voir le résultat de votre code

- Un explorateur pour voir vos différents fichiers (Tous vos fichiers devront être préfixés de votre trigramme).
1. Récupérez le fichier index.txt.
 2. Renommez-le TP01index.html (préfixé de votre trigramme : TRI_TP01index.html)
 3. Ouvrez ce fichier dans votre éditeur de texte (notepad++) et encapsulez le texte par les balises <body> et </body>.
 4. Récupérez le fichier entete.txt.
 5. Ouvrez ce fichier dans votre éditeur, inclure son contenu en tête du fichier TP01index.html et ajouter la balise </html> en fin de fichier.
 6. Mettre un titre entre les balises <title> et </title> de l'en-tête. C'est sous ce titre que votre document apparaîtra dans l'onglet de votre navigateur.
 7. L'en-tête étant complet, travaillons sur **le corps** (body) du document.
 1. Encapsuler le bandeau (Pour l'instant il n'y a que le titre « Cuisine du XXI^e siècle ») par les balises <header> et </header>. Ceci est notre première division.
 2. Encapsuler le menu par les balises <nav> et </nav>.
 3. Encadrer la présentation du site par les balises <article> et </article>.
 4. Sauvegarder votre fichier et chargez-le dans un navigateur pour tester votre code.

Si vous n'avez pas fait d'erreurs les trois divisions devraient apparaître, chacune débutant sur une ligne.

8. Encapsuler le texte du bandeau (header) par les balises <h1> et </h1> pour en faire un titre de niveau 1.
9. Mettre le "e" de "XXI^e" entre les balises ^{et} afin d'indiquer le « e ».
10. Sauvegarder et tester dans votre navigateur (il suffit de rafraîchir la page avec la petite flèche en haut à droite de la barre URL ou F5 ou Ctrl+F5 pour « forcer » le cache)
11. Dans la présentation du site, mettre un titre de niveau 2 pour le titre « Présentation du site » et des balises <p> et </p> à chacun des deux paragraphes. Tester.
12. Occupons-nous du menu à présent.
 1. "Au menu" doit devenir un titre de niveau 1, "Entrées/Plats/Desserts" des titres de niveau 2. Sauvegarder et tester.
 2. Encadrer les items de chaque menu par des balises et (Soient 3 balises et au total) pour en faire des listes non ordonnées (Unordered List) :


```
<ul>
  Oeufs d'Alien façon moléculaire
  Bouchées de courgettes à la confiture de parmesan
</ul>
```
 3. Et chacun des six noms de recettes par des balises et pour en faire des items de liste (List Item) :

Oeufs d'Alien façon moléculaire

4. Tester.

Le W3C (World Wide Web Consortium), est un organisme international qui développe des standards pour le Web afin que les gens puissent communiquer efficacement à travers Internet, autour de formats ouverts garantissant une meilleure **interopérabilité** (c'est-à-dire une meilleure compréhension des systèmes hétérogènes à travers des données et langages standardisés).

Le consortium existe depuis 1994 et est dirigé par l'inventeur du Web, Tim Berners-Lee.

Il est composé d'une équipe fixe et des membres (dont plusieurs centaines d'entreprises partenaires). Les membres délèguent des ingénieurs au sein du W3C et participent ainsi à l'élaboration des spécifications techniques pour les technologies du Web. De nombreux membres actifs font partie des équipes de développement des navigateurs (Microsoft, Mozilla, Apple, Opera, Google, etc).

Le W3C offre un outil important permettant de « valider » les pages HTML et les feuilles de style CSS.

13. Taper l'adresse "http://validator.w3.org/" dans un nouvel onglet du navigateur et faire valider la page par la méthode "file upload".

Si votre document contient des erreurs corrigez-les. Sinon vous aurez éventuellement un avertissement concernant la première balise h1, mais votre page sera correcte et validée par le W3C :

The document validates according to the specified schema(s).

On notera que la mise en forme actuelle est minimale. Nous n'avons indiqué que la nature des choses mais donné aucune véritable instruction de mise en forme. Toute la mise en page a été décidée entièrement par le navigateur à partir de nos instructions de fond. Une bonne page HTML doit toujours être ainsi conçue.

6. Exercice 03 : Design de la page

Créer une feuille de style et mettre en page

Les instructions de mise en page sont enregistrées dans une feuille de style au format CSS. La nôtre s'appelle "TP01Style.css" ce qui n'est pas original mais informatif.

1. Récupérez le fichier TP01Style.css. et renommez-le afin de le préfixer par votre trigramme.
2. Ouvrez le fichier. Il contient des indications de mises en forme pour le "body". Etudiez les différents attributs.
3. Enlever les marques de commentaires <!-- et --> de l'en-tête de votre fichier HTML et modifiez le nom du fichier référencé pour que la feuille de style s'applique. Tester → Les couleurs ont changé.

4. Créer une division <div id="page"> pour encapsuler vos trois divisions. Tester.
Q4 : Comment sont codées les couleurs noir et blanc ?

7. Exercice 04 : Style du bandeau

Au fur et à mesure que vous avez compris une propriété CSS commentez la dans le fichier TP01Style.css avec /* et */ (Vous pouvez aller chercher sur internet)

1. Mettons le bandeau en forme en ajoutant une par une les instructions suivantes et en testant et **commentant à chaque fois** :

```
header
{
    margin:2em 3% 1% 3%; /* commentaire à faire */
    padding:0.05em 0 0.05em 0; /* commentaire à faire */
    font-size:200%; /* commentaire à faire */
    background-color:#ffd000; /* commentaire à faire */
    border-radius:25px; /* commentaire à faire */
}
```

2. Rajouter, commenter et tester :

```
header h1
{
    font-family:"Comic Sans MS",Arial,Helvetica,sans-serif;
    text-align:center;
}
```

3. Il manque un logo pour notre bandeau.

Insérer dans la division du bandeau une division <div class="logo"> (au-dessus du titre). On notera l'utilisation de class plutôt que id. Contrairement aux id, les class ne sont pas uniques et peuvent apparaître plusieurs fois avec le même nom dans un même fichier. **Chercher des explications sur le Web.**

Récupérer le fichier logo.png.

Insérer dans la nouvelle division. Tester.

4. Pour positionner le logo utiliser le style :

```
header .logo
{
    float:left;
    margin-top:20px;
    margin-bottom:5px;
    margin-left:1%;
    display:block;
}
```

8. Exercice 05 : Style du menu

1. Positionnons le menu par :

```
left:5%;  
position:absolute;  
width:15%;  
overflow:hidden;
```

Q5 : Que signifie `position:absolute` ?

2. Tester. On se rend compte qu'il faut positionner immédiatement la présentation du site. Le faire avec une instruction `margin-left` provisoire. Inutile de faire quelque chose de très précis, nous y reviendrons en son temps.
3. Choisissez des couleurs de fond et d'avant plan pour le menu.
4. Puis donner les attributs suivants aux titres de niveau 2 du menu (vous pouvez changer la couleur) :

```
border-color:#f09000;  
border-style:solid;  
border-width:0 0 3px 0;  
font-size:medium;  
font-weight:700;  
margin:0;  
padding-left:0em;  
text-transform:uppercase
```

Q6 : Que signifie `em` ?

Q7 : Que signifie `font-weight` ?

5. Changer la couleur de la première lettre des titres de niveau 2 en utilisant `nav h2:first-letter`.
6. Terminer par la mise page des listes du menu, avec pour `` :

```
margin-bottom:1.5em;  
margin-top:0.5em;
```

Et pour `` :

```
list-style:none;
```

7. Tester.

9. Exercice 06 : Style de la « présentation du site »

1. Positionner cette présentation plus précisément par :

```
background-color:#f5f8fc;
```

```
border:1px solid #6090c0;
line-height:150%;
margin:0 3% 2em 21%;
padding:0 2% 3em 2%;
color:#000000;
```

et changer éventuellement les couleurs.

Q8 : Que signifie line-height ?

2. Choisir une mise en forme pour les titres de niveau 2.

Vérifier que chaque ligne de votre feuille de style a bien été commentée. Essayez d'organiser intelligemment les styles.

10. Exercice 07 : Créer des liens

La vocation d'une page HTML est d'être liée à d'autres pages.

Ainsi nous souhaitons qu'un clic sur un item de menu renvoie à la recette correspondante.

1. Modifier les items de menu comme par exemple `Oeufs d'Alien façon moléculaire`.
2. On constate que le menu n'est plus aussi joli. On va corriger cela dans la feuille de style.
3. Ajouter et tester

```
nav a
{
    display:block;
    text-decoration:none;
}
```

Vous pouvez changer d'autres propriétés à votre guise.

Q9 : Que signifient les deux propriétés précédentes ?

4. Changer la couleur du fond pour `nav a:hover` et tester.
5. Valider la page maintenant complète par le W3C.

11. Exercice 08 : Liens vers d'autres pages

1. Maintenant que la première page est achevée nous allons nous en servir comme modèle pour créer les suivantes. Faire une copie de cette page sous le nom "modele.html".
2. Transformer le logo afin qu'en cliquant dessus on retourne en page d'accueil (le logo est en fait un lien vers la page d'accueil).
3. Récupérer le fichier `oeuf_alien.txt`.

4. Insérer le contenu du document `oeuf_alien.txt` dans le modèle (dans la `div article`) et sauvegarder sous le nom `oeuf_alien.html` (Attention de ne pas oublier de préfixer tous vos fichiers par votre trigramme).
5. Mettez à jour les valeurs pour l'auteur et la date de mise à jour.
6. Formatez le reste de la page à votre guise.
7. Modifiez tous les items du menu afin qu'ils pointent vers la recette correspondante.

12. Exercice 09 : Créer une autre feuille de style pour le même fichier html

Pour bien comprendre la différence entre le fond (le texte en dehors des balises) et la forme (défini par le fichier CSS) nous allons travailler sur un autre fichier CSS.

Cette fois-ci nous voulons mettre le menu à l'horizontale, et ne faire apparaître à l'écran que les mots Entrées, Plats et Desserts. Lorsque la souris survole un de ces mots, apparaissent alors les items correspondants.

1. Reprendre le fichier `TP01index.html`
2. Encapsuler chaque titre `h2` et la liste `ul` associée par une balise `<div class="container">` qui va nous aider à placer les items de la liste `ul` bien au-dessous les uns des autres et du titre `h2`.
3. Récupérer le fichier `TP01Style02.css`
4. Faites la modification nécessaire afin que `TP01index.html` fasse appel au nouveau fichier `css`.
5. Tester.
6. Etudiez ce nouveau fichier `css` et approfondissez sa compréhension en vous aidant ici www.w3.org/Style/css3-selectors-updates et ici [Fonctionnement de float](#)

13. Quelques principes css

- Un fichier CSS est une suite d'instructions de la forme

```
Sélecteur
{
    attribut1:valeur1;
    attribut2:valeur2;
    etc...
}
```

- Les balises sont rangées en deux grandes catégories:
- Les blocks: les balises en double comme : `<p>` et `</p>` ou `` et ``
 - les inline: les balises simples comme : `` (pour insérer une image) et `
` (pour sauter de ligne)

- Avec l'attribut `display` on peut cacher ou montrer les balises à l'écran, ce qui est pratique pour faire un menu déroulant:

```
ul
{
    display:none;/*on cache la liste ul à l'écran*/
}

ul
{
    display:block;/*on montre la liste ul à l'écran*/
}
```

- Avec l'attribut `float` on peut mettre les blocks à l'horizontale.
- Les attributs `margin` et `padding` sont importants, ils permettent de positionner les blocks les uns par rapport aux autres.
- Les sélecteurs peuvent être un peu plus complexes comme `nav .container h2:hover +ul` qu'on lit de la droite vers la gauche :
 - Les attributs qui vont suivre concernent donc l'ul qui suit immédiatement un h2 qui a été survolé par la souris (`hover`) dans le block `container` qui se trouve dans le block `nav`.
 - Concrètement cela permettra de montrer à l'écran l'ul lorsque le mot encadré par le block h2 a été survolé à l'écran, par exemple Entrées, ou Plats ou Desserts. Evidemment il ne faut pas que la liste des Desserts soit révélée lorsque le mot Entrées est survolé par la souris.
 - Pour écrire des sélecteurs plus complexes, il faut donc bien analyser la structure en **arbre** du fichier html.
- Pour se repérer précisément dans un arbre il y a des sélecteurs comme `.container:nth-child(n+2)`, qui sélectionne tous les enfants des containers à partir du deuxième inclus.

14. Pour aller plus loin

Les pages HTML sont faites pour être lues par des navigateurs, des robots de recherche et bien sûr ceux qui les écrivent et les maintiennent.

La multiplication des types d'usage différents et l'automatisation des processus a conduit à abandonner les pratiques des années 1990 consistant à mélanger la forme et le fond. On peut produire de mauvaises pages en suivant ces anciens principes à l'aide d'un traitement de texte.

Au contraire, l'écriture de bonnes pages faciles à lire, à maintenir et à manipuler par des robots demande une structuration proche de la programmation. Il faut hiérarchiser les contenus, découper en blocs logiques, séparer la structure (HTML proprement dit), de la mise en page (reléguée dans les feuilles de style CSS).

Ces bonnes pratiques ont aussi l'avantage d'augmenter la vitesse de chargement des pages (les feuilles de style utilisées plusieurs fois ne sont chargées qu'une fois) et d'optimiser l'espace de stockage de l'information.

Vous avez pu constater qu'utiliser du HTML/CSS pur ne permettait toutefois pas d'éviter la répétition de certains contenus (bandeau, menu) ce qui ne facilite pas la maintenance des pages.

On peut éviter cela par la division d'une page en plusieurs fichiers envoyés séparément aux navigateurs. Cela correspond à l'utilisation de `frame`. Cette technique ancienne est déconseillée puisqu'elle rend difficile l'exploitation de liens profonds par les moteurs de recherche et risque de causer des incohérences et des pertes de contenu.

La bonne solution consiste à faire construire automatiquement les pages **du côté serveur WEB** pour qu'elles soient ensuite envoyées complètes au navigateur client. Cela peut se faire par l'utilisation de langages de programmation comme **JavaScript** ou **PHP** si l'on souhaite que cette construction soit interactive (les pages sont générées à la demande du client).