

RAPPORT

PROJET R.102

Noé VENGEON | TP1I
R.102 | DÉVELOPPEMENT D'INTERFACES WEB

Table des matières

PARTIE 1 – EPUB	1 à 2
Préparatifs et mise en place	1 à 2
Création des pages HTML et CSS.....	3 à 4
PARTIE 2 – SITE STATIQUE	4 à 5
Préparatifs et mise en place	4 à 5
Travail de conversion	5 à 6

PARTIE 1 – EPUB

Préparatifs et mise en place

ORGANISATION DES RÉPERTOIRES

Tout d'abord, toute cette partie concernant le ePub se trouve dans un gros répertoire que je nomme : « EPUB ». Pour bien commencer, de la même façon que vu dans le *Travail Pratique (TP) n°3 : première création d'un ebook*, j'ai créé deux répertoires pour permettre la création du ePub : « META-INF » qui contient le fichier « container.xml » avec le même script que dans le TP3. Puis un répertoire : « OEBPS » qui contient « content.opf » qui reprend la même base que dans le TP3 mais que l'on modifiera en temps voulu pour bien importer les bons fichiers CSS, HTML et images pour notre ePub. Enfin, il ne faut pas oublier le fichier « mimetype » (sans extension) qui doit se trouver dans le même répertoire où se trouve déjà « OEBPS » et « META-INF ».

Toujours de façon similaire au TP3, je rajoute trois répertoires dans « OEBPS » qui se nomme :

1. « html » où l'on trouvera les fichiers .xhtml
2. « css » où l'on trouvera les fichiers .css
3. « img » où l'on trouvera les images (.png .jpg)

Maintenant, qu'on a organisé nos répertoires on pourrait se dire qu'on peut commencer le travail du ePub mais il faut qu'on définisse quelles sections du livre on va traiter pour ce projet.

SÉLECTION DES 30 SECTIONS A TRAITER

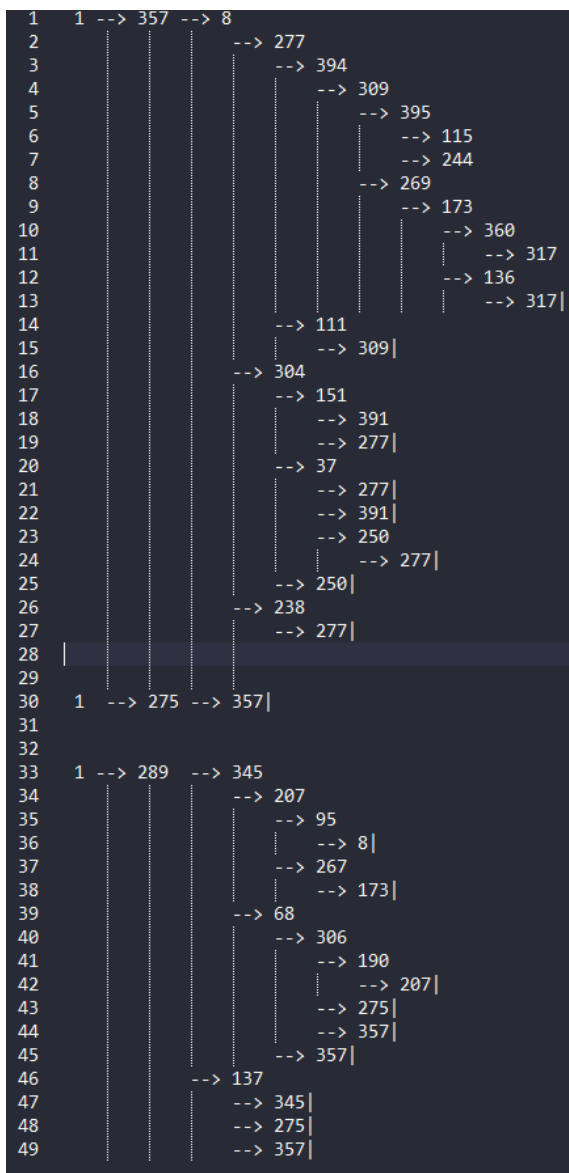
Il est indiqué dans le sujet que l'on doit réaliser **au moins 30 sections**¹.

Mais lesquelles choisir ? Pour savoir quelles sections j'allais devoir créer, j'ai eu l'idée de concevoir une **arborescence** sur un fichier texte à l'aide de l'éditeur *Notepad++*. Créer une arborescence m'a été utile pour y voir plus clair sur la manière dont ont été organisés les pages dans ce livre particulier. Mais aussi et surtout, cela permet à l'utilisateur d'interagir un bon nombre de fois avec les liens cliquables en partant des pages du début. Contrairement au cas où si j'avais sélectionné aléatoirement 30 sections à traiter, cela aurait été de la chance si je traite une section qui se trouve être un lien cliquable dans une autre page.

Cette arborescence correspond aux pages accessibles dans une page. C'est-à-dire, dans ce cas-ci : lorsqu'on est dans la page 1 on peut accéder à la page 357, 275 ainsi que 289. Et ainsi de suite...

(cf capture d'écran)
(à la page suivante)

¹section = 1 partie numérotée de l'histoire = lots de paragraphes
(voir PDF du livre)



Cette arborescence contient :

Sections uniques : 1, 8, 37, 68, 95, 111, 115, 136, 137, 151, 173, 190, 207, 238, 244, 250, 267, 269, 275, 277, 289, 304, 306, 309, 317, 345, 357, 360, 391, 394, 395

Total : 31

Légende : --> = accès à la page n°X

| = page traité précédemment

EXTRACTION DES IMAGES ET DU TEXTE

D'entrée de jeu, j'ai eu la liberté de la manière d'extraction des images et du texte. J'ai fait le choix pour l'extraction des images d'ouvrir le PDF du livre (fourni dans le sujet) sur Word qui peut le convertir en fichier *PDF modifiable* et ainsi donc faire Clique-droit, Enregistrer l'image, dans mon répertoire créé au préalable : « EPUB\OEBPS\img ». Évidemment j'enregistre uniquement les images liées à mes sections que je dois traiter (d'où l'utilité de les sélectionner en amont). Enfin, pour l'extraction du texte liés aux sections j'ai trouvé ça peu utile de créer des fichiers texte vierges et de copier-coller le texte dans un fichier qu'on aurait appelé « txt » dans « OEBPS » similairement aux images car au moment où je dois copier-coller mon texte dans ma page HTML c'est du pareil au même de copier-coller le texte depuis le PDF que depuis des fichiers textes. Sans parler du fait qu'il faudrait ouvrir tous les blocs-notes et donc qu'au final cela serait même moins rapide.

Création des pages HTML et CSS

Pour ce qui est partie technique je code sur le logiciel *Visual Studio Code*. Je mets à profit toutes mes connaissances que j'ai pu acquérir en Interfaces Web jusqu'ici. Je ne vais pas rentrer dans les détails de chacune de mes pages HTML et CSS, détaillé le choix de chacune balises etc. Je vais donc parler de mes choix et procédés plus en généralités. Avant ça, dès lors que j'ai rencontré des difficultés j'ai regardé sur Internet sur des sites comme : MDN Web Docs, Stack Overflow ou W3Schools, etc. Sinon en dernier recours pour ne pas me retrouver bloqué, j'ai demandé de l'aide à mes camarades de classe de BUT sur la façon de faire la chose qui me pose problème. Je n'ai eu recours qu'une fois à cela, j'expliquerais les circonstances prochainement (partie 2 - site statique).

Rentrons donc dans les généralités pour la création des pages HTML et CSS :

Pour créer les pages HTML et CSS je décide de suivre l'ordre du PDF du livre c'est-à-dire : couverture, page de garde, mécanismes, indications, intro-histoire, histoire (toutes les pages), copyright et 4ème de couverture. J'ai donc créé mes pages HTML suivis du CSS correspondant pour faciliter les intégrations du CSS dans le fichier HTML.

Après chaque création d'une page HTML et de son CSS je testais le résultat de mon code en ouvrant le .xhtml de la page pour voir les potentielles corrections à apporter. Ensuite j'ajoute chacune de mes nouvelles pages créer dans le « EPUB\OEBPS\content.opf » en respectant la syntaxe permettant la bonne intégration. En vérifiant que l'étape précédente été fonctionnelle, je testais mon fichier ePub à intervalles réguliers en le recréant et supprimant l'ancien. J'ai utilisé la procédure vue au TP3 pour créer mon fichier ePub c'est-à-dire ce qui est appelé « Emballage » dans la question 4 du TP3 (cf TP3). Enfin, comme stipulé dans l'énoncé du sujet, j'ai dû créer des éléments par moi-même qui étaient manquants au PDF, il s'agissait en fait de : la page de copyrights ainsi que la quatrième de couverture. Pour la page de copyrights j'ai été cherché de vraies informations pour la rendre la plus réaliste possible. J'ai pris les informations sur ce site : [Lien](#). Quant à la 4^{ème} page de couverte j'ai voulu aussi la rendre la plus réaliste possible. J'ai cherché des modèles en rapport avec *Le Manoir de l'Enfer* sur Internet et j'ai essayé d'en reproduire un qui me plaisait.

Durant cette première partie pour la création du ePub je n'ai rencontré que des difficultés mineures qui se sont vite réglé. J'ai été voir sur Internet des solutions à mes problèmes et j'ai pu vite comprendre et régler ces petits problèmes.

Pour conclure cette partie j'ai donc créé mon ePub cela marque la fin de cette première partie : « *PARTIE 1 – EPUB* », on peut désormais passer à la deuxième partie de ce projet que je nomme donc : « *PARTIE 2 – SITE STATIQUE* ».

PARTIE 2 – SITE STATIQUE

Préparatifs et mise en place

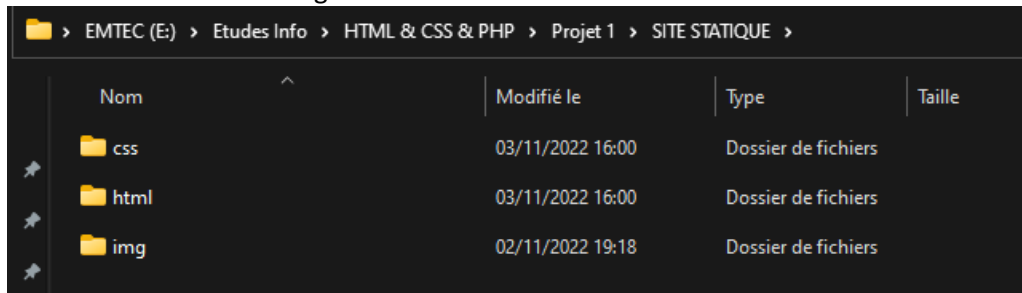
ORGANISATION DES RÉPERTOIRES

Pour l'organisation des répertoires je décide de prendre la même base que la partie ePub. En effet ayant déjà effectué le ePub c'est quasiment un travail de conversion en site statique que nous devons effectuer. Je copie-colle donc **le contenu** de mon répertoire « EPUB » qui contient mon « META-INF », ... dans un répertoire que je nomme « SITE STATIQUE ».

Ensuite, je le réhabilite dans le but de transformer ou plutôt convertir ce répertoire afin qu'il accueille parfaitement le site statique. Je procède donc :

1. **SUPPRESSION** → « *SITE STATIQUE\le_manoir_de_l_enfer.epub* »
2. **SUPPRESSION** → « *SITE STATIQUE\mimetype* »
3. **SUPPRESSION** → « *SITE STATIQUE\OEBPS\content.opf* »
4. **TRANSFERT** DU CONTENU DE « *SITE STATIQUE\OEBPS* » **DANS** « *SITE STATIQUE* »
5. **SUPPRESSION** → « *SITE STATIQUE\META-INF* »

Je me retrouve avec cette configuration-ci dorénavant :



The screenshot shows a Windows File Explorer window with the address bar displaying the path: EMTEC (E:) > Etudes Info > HTML & CSS & PHP > Projet 1 > SITE STATIQUE >. The main area shows a table of files and folders:

Nom	Modifié le	Type	Taille
css	03/11/2022 16:00	Dossier de fichiers	
html	03/11/2022 16:00	Dossier de fichiers	
img	02/11/2022 19:18	Dossier de fichiers	

Par la suite,

je change toutes les extensions des pages web qui étaient en « .xhtml » du répertoire « *SITE STATIQUE\html* » à « .html »

Travail de conversion

Après quoi, je change les balises XHTML :
Pour mettre les balises HTML :

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<html>
```

(quand je parle des balises il va sans dire que je parle de la balise ouvrante et fermante).

Il faut que j'extraie les sections du « SITE STATIQUE\html\histoire.html » et que je crée une page HTML par section.

Ici, ce n'est que du travail d'extraction aucune difficultés c'est juste fastidieux.

En même temps que l'étape précédente d'extraction, je change mon « href= "..." » dans mes balises « <a> » pour permettre d'accéder aux autres pages en .html et non plus des id définis dans une et même page en .xhtml, à la manière de « histoire.xhtml » de la partie ePub.

Exemple :

On passe de ça :

```
<a href="#357">357</a>
```



à ça :

```
<a href="p357.html">357</a>
```

J'effectue ceci dans toutes mes sections.

Enfin, nos pages web sont terminés mais les consignes demandent de **remplacer le travail d'une liseuse** c'est pourquoi je crée un fichier « liseuse.css » dans mon répertoire « SITE STATIQUE\css ». L'utilité de créer un CSS « liseuse.css » est comme de rajouter une « surcouché » pour que mes pages webs apparaissent à la manière de la liseuse. C'est à dire : bonne taille des pages mais surtout les boutons pour pouvoir tourner les pages. C'est à ce moment-ci que j'ai rencontré un problème que je n'ai pas pu résoudre avec des recherches Internet j'ai donc demandé à un camarade à moi sa façon de créer cet effet liseuse : avec les boutons, etc. Il m'a montré ce qu'il avait fait, m'a expliqué et j'ai pu recréer en m'inspirant de lui mon « liseuse.css » que j'ai d'ailleurs pu me réapproprier en le personnalisant le plus possible. Le CSS va de pair avec les boutons présents dans les pages HTML. J'ai donc dû rajouter tous les boutons à l'aide de balise « <a> » et rajouter les balises « <main> » dans toutes mes pages HTML.

De plus, j'ai créé un CSS que j'ai nommé « global » étant donné qu'il sera sur toute ma partie histoire et certaine autre page comme « copyrights.css ». Il a pour utilité de rajouter un *padding* pour être le plus fidèle au PDF, chose que le ePub Reader faisait automatiquement.

J'ai ajouté les deux nouveaux CSS que je venais de créer dans toutes mes pages HTML en respectant un ordre précis de priorité pour pas qu'il y est de conflit entre les éléments de mes CSS. Ce dernier est :

1. Global
2. Liseuse
3. Style de page

Mais le projet n'est pas encore fini ! Il est marqué dans le sujet qu'il fallait trouver des idées d'améliorations. J'ai trouvé comme idée de faire un « Dark Mode ». Il s'agit concrètement d'un changement d'affichage (qu'on appelle « SWITCH » en anglais de page) qui permettent une meilleure lecture selon certaines circonstances. Concrètement, j'ai créé un bouton « Changer le thème » réalisé

avec une balise « <a> » via le fichier HTML. Et j'ai rajouter dans les fichiers HTML de toutes les pages, un script en JavaScript (abrégé : JS) permettant l'action effectuant l'échange du thème.

Exemple : Le thème par défaut est *Sombre*, alors si on appuie pour la première fois sur le bouton « Changer le thème » le thème sera *Clair*.



Etant donné que c'est ma première fois que je code du JavaScript je n'ai pas réussi à faire en sorte que le switch de thème se prolonge lorsqu'on change de page web. J'ai quand même essayé de regarder sur Internet ou même à un camarade, j'ai essayé pleins de corrections mais je n'ai jamais réussi.

```
49
49  <script> /* PARTIE JAVA SCRIPT */
50
51  const switchThemeBtn = document.querySelector("#switch-theme");
52
53  function darkTheme() {
54    document.documentElement.style.setProperty('--ecriture', '#f1f1f1');
55    document.documentElement.style.setProperty('--background', '#1c1c1c');
56    document.documentElement.style.setProperty('--backgroundbis', '#2f2f2f');
57  }
58
59  function lightTheme() {
60    document.documentElement.style.setProperty('--ecriture', '#1c1c1c');
61    document.documentElement.style.setProperty('--background', '#a7a7a7');
62    document.documentElement.style.setProperty('--backgroundbis', '#f1f1f1');
63  }
64
65  let currentTheme = window.localStorage.getItem('theme');
66
67  if(currentTheme == "0") {
68    lightTheme();
69  } else if(currentTheme == "1") {
70    darkTheme();
71  } else {
72    lightTheme();
73    window.localStorage.setItem('theme', '1');
74  }
75
76  switchThemeBtn.addEventListener('click', () => {
77    currentTheme = window.localStorage.getItem('theme');
78    if(currentTheme == "0") {
79      lightTheme();
80      window.localStorage.setItem('theme', '1');
81    } else {
82      darkTheme();
83      window.localStorage.setItem('theme', '0');
84    }
85  })
86
87  </script>
```

Exemple de tentative de correction

J'ai donc décidé laisser comme au début car le système fonctionne en soi.

C'est ainsi que je conclue le projet R102 où l'on devait réaliser un Site Statique et un ePub à partir d'un PDF de livre interactif.