II – Créer son édition scientifique numérique

Dans cette deuxième partie du cours, on va s'intéresser à notre édition scientifique numérique avec l'encodage TEI en lui-même. Cela va se diviser en plusieurs parties. Tout d'abord, je vous ferai une extensive présentation du contenu d'un arbre XML-TEI, afin qu'il soit correctement formé, mais également assez complet aux vues de nos besoins. Ensuite, j'expliquerai l'importance de la construction d'un schéma d'encodage bien défini, ce qui passe notamment par sa documentation, afin de garantir une facilité de travail et une homogénéité dans la mise en place d'un corpus. Enfin, on passera à l'aspect pratique de la formation et vous aurez l'occasion d'encoder vous-même un texte, suivant les différents éléments que je vous aurais montrés.

A – La TEI au service de son édition

Nous allons voir tout d'abord que la TEI est là pour servir votre édition puisqu'elle se compose de multiples éléments et attributs qui donnent la possibilité d'encoder tout petit détail présent sur le facsimilé ou autre document que vous cherchez à traiter avec la TEI. Il y a généralement une balise pour chaque élément à recenser et c'est ce que je vais présenter maintenant, de façon détaillée.

a) Contenu de l'arbre XML-TEI

Tout d'abord, on retrouve une balise unique, la balise racine <TEI>. Elle doit impérativement être accompagnée d'un espace de nom (*xmlns="http://www.tei-c.org/ns/1.0"*). Cela permettra ensuite de vérifier que l'arbre est bien en TEI, que vous utilisez les bonnes balises et spécialement cela pourra aider l'autocomplétion, (disponible sur Oxygen par exemple). Un document TEI se divise ensuite en plusieurs parties, dans lequel chacune contient des informations spécifiques. Il y a tout d'abord deux grandes parties :

- Le <teiHeader>, qui est là pour encoder les métadonnées associées au document travaillé
- Le <text>, qui représente le contenu du texte, qui sera divisé lui-même en trois parties distinctes : le <front>, le <body> et le <back>. Le *front* et le *back* contiennent respectivement les annexes d'avant et d'après texte et le *body* contient le corps du texte même.

Au sein même de toutes ces parties, on pourra trouver des balises d'encodage assez spécifiques pour encoder des informations particulières.

b) Les métadonnées : le <teiHeader>

Toutes les informations d'encodage pour le <teiHeader> se trouveront dans le chapitre 2 des guidelines : https://tei-c.org/release/doc/tei-p5-doc/fr/html/HD.html

<teiHeader> (en-tête TEI) fournit des informations descriptives et déclaratives qui constituent une page de titre électronique au début de tout texte conforme à la TEI.

Le header est composé de quatre grandes parties, qui apporte chacune des informations différentes sur le contenu du fichier électronique qui sera traité dans la partie texte :

- **<fileDesc>** (description bibliographique du fichier) contient une description bibliographique complète du fichier électronique.
- **<encodingDesc>** (description de l'encodage) documente la relation d'un texte électronique avec sa ou ses sources.
- **<profileDesc>** (description du profil) fournit une description détaillée des aspects non bibliographiques du texte, notamment les langues utilisées et leurs variantes, les circonstances de sa production, les collaborateurs et leur statut.
- **<revisionDesc>** (descriptif des révisions) fournit un résumé de l'historique des révisions d'un fichier.

Sur toutes ces parties, seule la première, le <fileDesc>, est obligatoire. Les autres sont optionnels.

1. *<fileDesc>*

Le <fileDesc> est la partie du header qui est souvent la plus longue, puisqu'en tant que description complète du fichier électronique, elle peut contenir de très nombreuses informations, parfois très détaillées, sur la composition du fichier. Il y a six parties :

- **<titleStmt>** (mention de titre) regroupe les informations sur le titre d'une œuvre et les personnes ou institutions responsables de son contenu intellectuel.
- **<editionStmt>** (mention d'édition) regroupe les informations relatives à l'édition d'un texte.
- **<publicationStmt>** (mention de publication) regroupe des informations concernant la publication ou la diffusion d'un texte électronique ou d'un autre type de texte.
- **<seriesStmt>** (mention de collection) regroupe toute information relative à la collection (si elle existe) à laquelle appartient une publication.
- **<notesStmt>** (mention de notes) rassemble toutes les notes fournissant des informations sur un texte, en plus des informations mentionnées dans d'autres parties de la description bibliographique.
- **<sourceDesc>** (description de la source) décrit la source à partir de laquelle un texte électronique a été dérivé ou produit ; habituellement une description bibliographique pour un texte numérisé, ou une expression comme "document numérique natif " pour un texte qui n'a aucune existence précédente.

À l'intérieur du <titleStmt>, on trouvera tout d'abord le titre de l'œuvre (ou de l'extrait mentionné si l'œuvre n'est pas complète) et les personnes impliquées dans la création de l'œuvre (un auteur <author>, un éditeur <editor>, un commanditaire <sponsor>, un financeur <funder> ou un chercheur principal <principal>. On trouvera aussi des informations des personnes impliquées dans tout ce qui aura trait au traitement de l'œuvre, informatique notamment, tel que son encodage, sa numérisation, sa transcription, etc. (<respStmt>, <resp> et <name>).

Dans <editionStmt>, on trouve les informations liées à l'édition du texte, ce qui sera généralement encodé à l'aide d'une balise <edition> et de <respStmt>, comme ceux de la partie précédente.

La partie <publicationStmt> vise à donner les informations sur la publication du fichier électronique actuellement traité dans l'encodage. Il peut contenir divers types de responsables (un éditeur <publisher>, un diffuseur <distributor> ou un responsable de publication <authority>) et un lieu de publication <pubPlace>. Il y a aussi la disponibilité du fichier <availability>, habituellement associé à une licence <licence> et enfin une date, qui ne sera pas la date de création du fichier, mais celle de publication.

Il est possible que l'œuvre ne soit pas un élément unique et indépendant, mais une partie appartenant à une collection plus large. Dans ce cas, la balise <seriesStmt> aura comme objectif de mentionner cette collection, en donnant son titre <title>, son identifiant <idno> s'il en a un et encore une fois le responsable de la collection même avec un <respStmt>.

La balise <noteStmt> a pour vocation de rassembler toutes les notes qui fournissent des informations sur le texte, ce qui est encodé grâce à la balise <note>.

Enfin, la balise <sourceDesc> aura pour objectif de décrire la source à l'origine du texte électronique. Cela peut être une simple balise paragraphe qui mentionne « document numérique natif » dans les cas où le fichier n'existait pas avant, comme lorsque l'on crée un index. Cela peut aussi se composer d'informations bien plus étendues, telles qu'une description bibliographique <bibl>, simple ou détaillé. Dans certains autres cas, on utilisera la balise <msDesc> qui sert à la description de manuscrit (voir Chapitre 10), avec laquelle on peut décrire la source physique, son lieu d'origine et autres informations complémentaires.

2. *<encodingDesc>*

L'<encodingDesc> a pour objectif de décrire l'encodage du fichier électronique.

Tout d'abord, l'une des balises les plus importantes sera <projectDesc> qui est là pour décrire le projet. On peut ainsi présenter en détail le but et l'objectif visé dans l'encodage d'un fichier électronique, ainsi que tout autre détail qui pourrait être pertinent sur la manière dont le fichier a été travaillé.

Il peut aussi contenir un certain nombre de balises, de déclarations (de balisages, de classifications, de coordonnées géographiques). Les deux plus importantes sont <samplingDecl> et <editorialDecl> :

- **<samplingDecl>** (déclaration d'échantillonnage) contient une description en texte libre du raisonnement et des méthodes utilisés pour l'échantillonnage des textes dans la création d'un corpus ou d'une collection.

  Il peut ainsi être mentionné la taille des échantillons, les méthodes de sélections et ce qui a été échantillonné

  <editorialDecl> (déclaration des pratiques éditoriales) donne des précisions sur les pratiques et les principes éditoriaux appliqués au cours de l'encodage du texte.

  Cela peut concerner les règles de correction <correction>, de normalisation <normalization>, de ponctuation <punctuation>, de citation <quotation> ou encore de césurage <hyphenation>.

3. *<profileDesc>*

Le <profileDesc> est là pour fournir la description détaillée des aspects non bibliographiques du texte. Cela peut contenir la date de création du texte <creation> (et non de publication), les langues utilisées dans le texte <langUsage>, un résumé du texte fait par l'encodeur <abstract>, ainsi que des informations un peu plus précises. Si l'on veut fournir des détails sur les entités mentionnées et annotées dans le texte, on peut rajouter un <particDesc> et un <settingDesc> qui sera là pour décrire les participants et le contexte du texte (lieux et organisations). Si on travaille avec une correspondance, on peut fournir des informations sur les correspondants et les temps d'écriture avec un <correspDesc>. Si le texte semble avoir été écrit à plusieurs mains, on peut présenter ces différentes mains avec <handNotes> et plusieurs <handNote> imbriqués. Cependant, cela ne fonctionne que dans le cas où nous ne sommes pas avec un manuscrit et donc un <msDesc>, ce qui impliquerait que les mains ont déjà été décrites dedans avec un <handDesc>.

4. *<revisionDesc>*

Enfin, le <revisionDesc> fournit un résumé de l'historique de révisions d'un fichier, c'est-à-dire qu'ils encodent toutes les modifications effectuées sur un fichier électronique

depuis sa création. Cela se fait à l'aide de la balise <change>, qui sera généralement accompagnée d'un attribut @who pour spécifier la personne responsable du changement et @when pour connaître la date de changement. Il est recommandé d'encoder ces changements dans l'ordre chronologique inversé, avec le plus récent en premier.

c) Les annexes, avec <front> et <back>

Outre la balise <body> qui contient le corps du texte, on peut trouver dans la balise <text> des annexes, qui se divisent en deux types : les annexes d'avant texte ou <front> et les annexes d'après texte ou <back>. Ils peuvent contenir divers éléments complémentaires au texte et ils seront encodés de la même manière que le corps du texte dans le <body>.
Ces éléments sont spécifiés dans le chapitre 4 des guidelines :

- 4 Default Text Structure https://tei-c.org/release/doc/tei-p5-doc/fr/html/DS.html

Le texte préliminaire ou <front> peut contenir les éléments suivants :

- Une préface ou un avant-propos adressé aux lecteurs par l'auteur
- Des remerciements rédigés par l'auteur
- Une ou des dédicace(s) faites par l'auteur pour des personnes en particulier ou une/des institution(s).
- Un résumé ou court extrait du contenu du texte
- Une table des matières, qui présente le contenu et la manière dont il est structuré au sein du texte
- Un frontispice avec la description, s'il y en a une, de l'illustration présente avec la page de titre.

Le texte annexe ou <back> peut contenir les éléments suivants :

- Une annexe contenant des sections auxiliaires du texte avec des informations complémentaires
- Un glossaire avec une liste de termes associée à leurs définitions
- Un regroupement de notes textuelles ou autre genre de notes contenues dans le texte
- Une liste des citations bibliographiques qui se retrouvent dans le texte
- Toute forme d'index associé à l'œuvre
- Un colophon, soit une note finale précisant les conditions physiques de production de l'ouvrage

d) Le corps du texte : le <body>

Une fois les métadonnées encodées, on va s'intéresser au contenu du texte, qui sera mis dans la balise <text>. La partie la plus importante sera le corps du texte, qui devra être contenu dans la balise <body>. Les différentes règles d'encodage de cette partie se trouvent dans les chapitres suivants :

- 3 Elements Available in All TEI Docs https://tei-c.org/release/doc/tei-p5-doc/fr/html/CO.html#COPU
- 4 Default Text Structure https://tei-c.org/release/doc/tei-p5-doc/fr/html/DS.html

Au sein de la balise <body>, il est généralement préférable d'encoder le texte à l'aide de divisions et subdivisions, qui seront faites à l'aide de la balise . Il existe deux types de , une normale et des divisions numérotés (<div1>, <div2>, etc.). L'utilisation de ces dernières n'est pas recommandée et il sera plus encouragé d'utiliser la balise en lui ajoutant des attributs de spécifications.

Ainsi, il y a trois principaux attributs :

- @*xml:id* pour donner un attribut bien spécifique à la division, qui pourra être référencé par la suite.
- @*type* cela servira à spécifier le type de divisions que l'on a (par exemple, tome, chapitre, poème, lettre, etc.)
- @*n* cela sert de numérotation et sera généralement associé au type que l'on aura défini

Une fois dans la division et avant de mettre le texte même, on peut donner un titre, s'il y en a un, à l'aide d'un <head> qui contient tout type d'en-tête (titre de section, intitulé de liste, description de manuscrit).

Les autres balises très importantes qu'il faut retenir sont les suivantes :

- Si un texte passe à la ligne sans terminer le paragraphe, le vers, ou tout autre type de division, il faudra y mettre la balise <lb>, qui permet d'encoder les sauts de ligne. C'est une balise vide, auto fermante.
- Dans le cas d'un fichier d'encodage qui reprend un document paginé, le saut de page sera marqué par la balise <pb>. Il peut être mis après la fin d'un paragraphe et le début d'un autre ou même avant la mise en place d'un . Ses attributs permettent notamment de numéroter les pages, mais aussi d'appeler le facsimilé pour avoir l'image en face, lorsqu'elle est disponible numériquement.

Ensuite, pour le texte, il y aura diverses balises selon le type de texte que l'on encode :

- La balise <p> pour marquer les paragraphes dans des textes en prose
- Les balises <lg> (groupe de vers) et <l> (vers) pour encoder les poèmes ou les pièces de théâtre
- Les balises <list> (liste) et <item> (composant) pour encoder une liste et ses composants
- Les balises <table> (tableau), <row> (ligne) et <cell> (cellule) pour encoder le contenu d'un tableau
- La balise <figure> (figure) qui est là pour encoder une illustration. Elle s'accompagne généralement d'une balise <graphic> qui contiendra le lien URL vers l'image. Elle peut aussi être accompagnée d'un <figureDesc> qui servira à décrire l'image.

Il y a bien d'autres balises qui permettent d'encoder un texte, de manière plus ou moins détaillée, en fonction du type de texte sur lequel on travaille (une pièce de théâtre, un roman, un poème, un discours, etc.) et qui se trouvent facilement dans les guidelines, grâce à des chapitres dédiés exclusivement à eux.

e) L'encodage d'éléments spécifiques dans les textes

Enfin, une fois les métadonnées et le corps du texte encodé, on pourra aller un peu plus en profondeur en annotant le texte par le biais de balises bien plus spécifiques. Pour ce faire, on peut suivre les différentes balises présentées dans les chapitres suivants :

- 3 Elements Available in All TEI Docs
  https://tei-c.org/release/doc/tei-p5-doc/fr/html/CO.html#COPU
- 13 Names, Dates, People, and Places
  https://tei-c.org/release/doc/tei-p5-doc/fr/html/ND.html

Tout d'abord, il y a l'encodage d'entités nommées, c'est-à-dire tout ce qui aura trait aux personnes, lieux et organisations qui seront cités dans le texte (mais aussi parfois dans les métadonnées). Il peut être intéressant d'encoder ces données pour faire un relevé des personnes, lieux et autres entités que l'on retrouve dans le texte. Il pourra ensuite être possible d'aller chercher plus d'informations sur ces entités et de les décrire plus amplement dans une liste, comme celle que l'on a brièvement présenté dans les <particDesc> et <settingDesc> que l'on trouve dans le <profileDesc>.
Il existe plusieurs techniques pour encoder ces entités. Tout d'abord, il y a une version assez générique avec les balises <rs> et <name>.
<rs> représente une chaîne de référence qui contient un nom générique ou une chaîne à laquelle on devra apposer un attribut @type (person, place ou org) et surtout un @ref pour identifier la chaîne. Par exemple, si dans un texte, un auteur mentionne « le Président », on

encodera cette partie avec la balise <rs> et ensuite, avec le contexte de l'écrit, on pourra faire une entrée dans un index pour préciser quel président est référencé et mettre cette référence comme attribut pour la balise. La balise <name> fonctionne de manière assez similaire, à l'exception que <name> ne doit être utilisé que pour des noms propres.

Si l'on veut être plus précis dans notre encodage, il est possible d'utiliser des balises plus spécifiques, qui n'auront plus besoin d'un attribut @type puisque la précision sera dans la balise :

- **<persName>** (nom de personne) contient un nom propre ou une expression nominale se référant à une personne, pouvant inclure tout ou partie de ses prénoms, noms de famille, titres honorifiques, noms ajoutés, etc.
- **<placeName>** (nom de lieu) contient un nom de lieu absolu ou relatif.
- **<orgName>** (nom d'organisation) contient le nom d'une organisation.

Comme on peut le voir, tous les éléments n'ont pas été remplacés par les balises spécifiques, puisque « sa sœur » et « l'hôtel » ne sont pas des entités assez précises pour être encodé avec les balises <persName> et <placeName>

B – Construire et documenter son schéma d'encodage (et s'y tenir !)

Comme on vient de le voir pendant cette présentation du contenu d'un arbre XML-TEI et la brève introspection dans les TEI Guidelines, la *Text Encoding Initiative* est très fournie et contient un très grand nombre d'éléments. De plus, comme on l'a vu juste avant, il est également possible d'avoir deux balises pour un même élément de texte. Dans ce cas, il sera nécessaire de faire un choix et de s'y tenir. C'est alors que la construction et la documentation du schéma d'encodage entre en scène. La TEI a créé un type particulier de fichier qui permet de faire explicitement cela : une ODD pour *One Document Does it all*, dans le sens que dans le même document, on va avoir la possibilité de définir notre schéma d'encodage, mais aussi de le documenter.

a) Construire son schéma d'encodage

Afin de ne pas créer notre ODD en partant de rien, mais également de plus facilement mettre en place notre schéma d'encodage, nous allons utiliser *Roma*, qui a été créée spécifiquement par le consortium pour cette tâche. Cet outil va nous permettre de rapidement remplir les métadonnées de notre fichier ODD et de s'intéresser plus rapidement ensuite au contenu de notre arbre TEI, et notamment de ce que l'on veut et ne veut pas dedans. Cet outil recense tous les éléments contenus dans la TEI et catégorisé par module (header, core, manuscript description, etc.). Il y aura quatre type d'affichage des modules :

- Si la classe n'est absolument pas utilisée, elle n'est pas mentionnée dans l'ODD et dans ce cas, l'ODD saura que tous les éléments contenus dedans seront à exclure de l'encodage. ;
- Si la classe n'est utilisée que pour un nombre limité d'éléments, elle sera déclarée et accompagnée d'un attribut @include qui contiendra ces éléments ;
- Si la classe est utilisée pour un très grand nombre d'éléments, elle sera déclarée et accompagnée d'un attribut @except qui contiendra les éléments non retenus ;
- Enfin, si la classe est utilisée avec tous ses éléments, elle sera déclarée simplement.

Une fois que les déclarations ont été faites, il est également possible de personnaliser son schéma de quatre manières : addition/suppression/changement d'éléments et personnalisation d'attributs et de leurs valeurs. L'addition implique de créer un élément ou attribut qui n'appartient pas au domaine de la TEI mais qui serait jugé nécessaire dans le cadre du corpus. La suppression peut permettre notamment d'enlever la possibilité d'utiliser certains attributs dans des éléments donnés. Le changement peut permettre la personnalisation de certains éléments ou attributs. On peut décider de changer ce qui peut être contenu dans certains éléments pour donner plus d'informations, par exemple, et on peut aussi l'utiliser pour définir un certain nombre de valeurs pour un attribut donné.

Exercice ROMA

- Aller sur Roma : https://romabeta.tei-c.org/
- Choisir "TEI ALL" et "START"
- Mettre comme titre "ODD Formation TEI" et mettre votre nom dans "Author" puis cliquer sur "--> CUSTOMIZE ODD"
- Classer les éléments par module avec "⇅ by module"
- Désélectionner "*analysis*", "*certainty*", "*corpus*", "*dictionaries*", "*drama*", "*gaiji*", "*iso-fs*", "*linking*", "*nets*", "*spoken*", *'tagdocs*" et "*textcrit*"
- Avec l'outil de recherche, taper "*div*" et désélectionner les div numérotés
- Avec l'outil de recherche, taper "*name*" et désélectionner l'élément "*name*"
- Avec l'outil de recherche, taper "*def*", cliquer sur l'élément puis "*Attributes*", modifier l'attribut "*rend*", mettre l'option "Closed" et ajouter les valeurs "erasure", "overwritten" et "strikethrough"
- Cliquer sur "Download" et choisir "Customization as ODD"
- Modifier l'extension du fichier pour mettre un ".xml"
- Ouvrir l'ODD nouvellement créée et opérer la transformation en schéma Relax NG (qui est un type de fichier utilisé pour valider la conformité à la TEI)
    - Appuyer sur le bouton "Appliquer les scénarios de transformation"

    - Choisir “TEI ODD to RELAX NG XML”
    - Appuyer sur le bouton “Appliquer”
- Dans le dépôt du cours, télécharger le fichier “orgueil_et_prejuges.xml” et ensuite l’ouvrir
    - Appuyer sur le bouton “Associer schéma”
    - Dans “URL”, utiliser le bouton “Naviguer” et choisir le fichier RNG nouvellement créé et situé dans le dossier “out”
    - Dans “Type de schema”, sélectionner “RELAX NG XML Syntax” puis appuyer sur “OK”
- Relever les erreurs qui se présentent alors et modifier selon ce qui est nécessaire, pour permettre de rendre le fichier conforme
    - Ajouter “particDesc” et “settingDesc” du module “corpus” dans l’ODD
    - Regénérer le schéma Relax NG
    - Mettre en commentaires le qui contient les entités nommées encodées avec <name>
    - Modifier la valeur de l’élément <del> pour qu’il corresponde à une des valeurs imposées

→ Le fichier devrait enfin être conforme

b) Documenter son schéma d’encodage

Une fois que l’on a créé notre schéma d’encodage et que l’on a décidé quels éléments garder et quels éléments supprimés, on a la possibilité de documenter tout le processus.
L’intérêt de cette documentation est double. Tout d’abord, c’est un moyen personnel de préciser les modalités d’encodage de son corpus, afin de toujours savoir quelles balises utiliser à quel moment, notamment si on travaille sporadiquement sur son encodage. Après l’avoir abandonné quelque temps, en y revenant en ayant accès à une documentation d’encodage, on saura d’emblée ce que l’on doit faire. Cette documentation peut également être utile pour d’autres qui pourraient travailler sur le même type de fichier que vous, mais avoir des doutes sur le choix de l’une ou l’autre des balises pour un élément donné. La documentation fournira un élément de réponse.
Il y a plusieurs choix pour cette documentation:

- exhaustive → on présente au cas par cas, pour chaque élément, le choix qui a été fait et pourquoi, ou simplement, on présente la balise et comment elle s’utilise
- partielle → on évoque simplement les cas d’utilisation de balise qui nécessite de plus amples informations.

Cette documentation se fait également en XML. Elle reprend tout d'abord les éléments standards de la hiérarchie XML, tel que , <head> et <p>. Mais on retrouve également de nouveaux éléments dédiés à la documentation, tel que <gi> pour mentionner un élément, <att> pour mentionner un attribut et <eg> pour donner un exemple ou <egXML> pour partager un bout d'arbre XML en guise d'exemple.

Exercice de documentation → Faire écrire un paragraphe qui documente le choix d'utilisation des balises spécifiques d'entités nommées, plutôt que de la balise <name> avec @type="person" ou "place" ou "organisation", en mentionnant les noms de balises, d'attributs et leurs valeurs ; fournir un exemple tiré du texte de démonstration

C – Travaux pratiques : définir un premier balisage pour son corpus

a) Oxygen XML Editor

Cet outil permet de travailler à l'encodage de documents en XML et autres types de langages (XSLT, Markdown, EAD, etc.). Il est notamment pratique pour sa fonction d'autocomplétion, qui permet, lorsque le *namespace* est déclaré et que l'outil sait quel vocabulaire est utilisé (TEI, EAD, etc.), d'offrir des suggestions d'utilisation de balises. Cela est particulièrement utile pour savoir quelles balises sont autorisés à certains endroits ou non. En effet, une fois une balise ouverte, les propositions d'ouverture de nouvelles balises suivront les règles établies dans les guidelines, à savoir la hiérarchie TEI, puisque les balises ne peuvent pas toujours être utilisées. Cela dépendra de l'endroit où on se trouve dans l'arbre TEI.

Pour vous aider aux exercices d'encodage que vous allez faire à partir de maintenant, voici quelques petits raccourcis pour l'utilisation d'Oxygen, ainsi que quelques expressions régulières qui pourraient être utiles dans l'utilisation de l'option "Cherche/Remplace". Les expressions régulières sont une chaîne de caractères qui décrit, selon une syntaxe précise, un ensemble de chaînes de caractères possibles.

b) Encoder les métadonnées

Exercice :

Créer un fichier XML-TEI que vous enregistrerez sous le nom de "Le_tour_du_monde_en_80_jours.xml" dans un endroit facile d'accès (sur le Bureau ou dans les Téléchargements par exemple)

En utilisant les métadonnées de l'ouvrage *Le Tour du monde en quatre-vingts jours* présent dans le dossier "texte_de_travail" du dépôt du cours et sur la page Gallica de l'ouvrage, ainsi qu'en vous aidant de ce que vous pourrez, dans les TEI Guidelines et dans les textes

d'exemple, encoder le <teiHeader> de l'ouvrage utilisé comme texte de travail.

c) Encoder le corpus du texte

Exercice :

En utilisant le fichier texte présent dans le dossier "texte_de_travail" du dépôt du cours, ainsi que les ressources que vous pouvez trouver dans les TEI Guidelines, encoder le <body> du chapitre 2 du *Tour du monde en quatre-vingts jours* de Jules Verne, afin de représenter tous les éléments que l'on peut trouver dans cet extrait de l'ouvrage (numéro de chapitre, titres, pagination, figures, etc.)

Astuce pour encoder rapidement les linebreaks (après avoir encodé les paragraphes, titres, pagination, etc.):

- ❏ Sélectionnez votre texte et faites chercher/remplacer
- ❏ Vérifier que "Seulement les lignes sélectionnées" et "Expressions régulières" sont cochés
- ❏ Dans "Rechercher", rentrez "-$" puis faites "Chercher tout" pour vérifier ce que l'on vous propose
- ❏ Une fois que vous êtes sûrs, dans "Remplacer", rentrez "-<lb break="no"/>" et faites "Tout remplacer"
- ❏ Refaites l'expérience en changeant "-$" par "[^>]$" et "<lb/> " (ne fonctionnera que si les fins des lignes sont précédés d'espace, sinon cela remplacera un caractère dans les mots de fin de ligne, d'où la nécessité de vérifier ce qui va être remplacé)

d) Encoder les entités nommées

Exercice:

En reprenant votre texte nouvellement encodé et en vous aidant de ce qui a précédemment été cité en exemple, ainsi que des TEI Guidelines, encoder les noms des deux personnages principaux du chapitre (pour chacune de leurs mentions), tous les lieux que vous trouverez, ainsi que l'organisation qui est mentionnée à quelques reprises.

Vous devrez ensuite créer un index dans le header, où vous regrouperez, par type, les différentes entités que vous aurez trouvées.

Lors de cet encodage, les entités qui devront être reconnus sont les suivantes:

- Phileas Fogg et Jean Passepartout (personnes)
- Reform-Club (organisation)
- Mme Tussaud, Royaume-Uni, Londres, Paris, Angleterre, Maison de Saville-Row (lieux)