Text Encoding Initiative

ENC, M2, 2017-2018
vincent.jolivet@enc-sorbonne.fr

1. Du texte brut aux balises

Les états du texte numérique ?

“type” de fichier	formats	usages
image		
texte brut		
traitement de texte		
texte balisé		
format détachable		

Oppositions :

- image / texte
- texte brut / texte enrichi

Texte brut – exercice (1/2)

Comparer différents fichiers du début du *Poète assassiné* d'Apollinaire.
http://corpus.enc.sorbonne.fr/cours/m2/apollinaire/

- apollinaire1.txt = apollinaire2.txt ?
- apollinaire.txt / apollinaire.docx, +/- ?
- apollinaire.docx est-il bien structuré ?

Éditeurs de texte

- Gedit (Linux, éditeur par défaut)
- Notepad++ (Windows)
  https://notepad-plus-plus.org/fr/
- Komodo Edit (Linux, Windows, OS X)
  https://www.activestate.com/komodo-ide/downloads/edit
- Atom (Linux, Windows, OS X)
  https://atom.io/

texte brut / texte enrichi

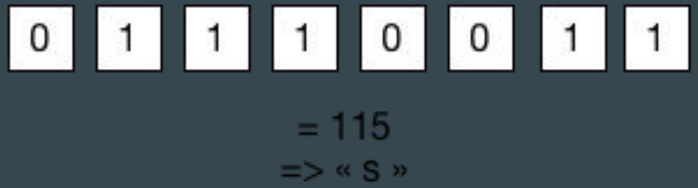
- **Text brut** (*plain text*) = une chaîne de caractères : “bonjour” / “Bonjour” / “BONJOUR”
- **Texte enrichi** (*fancy text*) = le texte de nos traitements de textes
  Texte + mise en forme (à l’affichage) :
  - **Bonjour** / *bonjour* / bonjour / Bonjour / `bonjour`

Le standard Unicode définit le texte brut.

Le texte brut représente le contenu basique, échangeable et inter-opérable du texte.

Le texte brut représente seulement les caractères contenus, sans leur apparence (ceci signifie que seule une numérotation des caractère est utilisée, la police de caractères étant fournie par un mécanisme indépendant).

Texte brut. Codage des caractères



- **ASCII** (*American Standard Code for Information Interchange*)
  Les caractères latins non accentués (écrire en anglais)
  codés sur 7 bits. 2^7 = 128 possibilités.
- **ISO 8859-1 (Latin 1)**
  Les 191 caractères de l'alphabet latin ; conçu comme une extension de l'ASCII
  codés sur 1 octet (8 bits). 2^8 = ? possibilités.
- **UTF-8** (***U****niversal Character Set* ***T****ransformation* ***F****ormat - **8** bits*)
  L'ensemble des caractères du « répertoire universel de caractères codés »
  codés sur 1 à 4 octets (compatible Unicode et ASCII)
  L'UTF-8 est utilisé par **86 % des sites web** en 2016.

USASCII code chart

b_7 b_6 b_5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Row ↓ Column →	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Texte brut – exercice (2/2)

- convertir apollinaire1.txt en UTF-8 ;
- restructurer apollinaire1.txt en paragraphes (connaissez-vous les regex ?) ;
- cette structuration en paragraphe est-elle explicite ?
- le texte reste-t-il le même ?
- trouver une manière de caractériser (dans le fichier) les éléments éditoriaux :
  - le titre principal
  - les titres hiérarchiques
  - les paragraphes
  - la pagination
  - la mise en valeur typographique (comment désambiguïser la sémantique de l'italique ?)
  - ?

De quoi avons-nous besoin ?

Les expressions régulières (regex)

- dans — toutes les occurrences de “dans”
- ^dans — “dans” en début de ligne – ancre
- qui$ — “qui” en fin de ligne – ancre
- ^$ — ?
- de|du — toutes les occurrences de “de” et de “du”
- d[eu] — toutes les occurrences de “de” et de “du” – **les** classes de caractères
- [a-z] — “a”, ou “b”, ou “c”, ..., ou “y”, ou “z” – intervalle (dans une classe)
- [^a-z] — n’importe quel caractère sauf “a”, ou “b”, ou “c”, ..., ou “z”
  classe complémentée **(tout caractère qui n’est pas énuméré).**

Les quantificateurs

- s**?** — **facultatif** (reconnaît zéro ou une occurrence de “s”)
- [a-z]***** — **facultatif** (reconnaît zéro, une ou plusieurs occurrences de la classe [a-z])
- e**+** — **obligatoire** (reconnaît une ou plusieurs occurrences de “e”)
- [a-z]**{n}** — **obligatoire restrictif** (reconnaît **n** occurrences de la classe [a-z])
- [a-z]**{n,m}** — **obligatoire restrictif** (reconnaît **n** à maximum **m** occurrences de la classe [a-z])
- [a-z]**{n,}** — **obligatoire non restrictif** (reconnaît au moins **n** occurrences de la classe [a-z])

Capture de sous chaînes

- ()

Langages à balises

```
$$contenu divers$$

<balise>contenu divers</balise>
```

- Texte brut : « qui signifie parce que »
- Résultat attendu : « qui signifie *parce que* »
- Balises :
  - qui signifie <italique>parce que</italique>
  - qui signifie $$parce que$$

Des balises

Word, Writer balisage *ad hoc* pour l’italique.

LaTeX	`balisage \emph{ad hoc} pour l’italique.`
wikicode	`balisage ''ad hoc'' pour l’italique.`
DocBook	`balisage <emphasis>ad hoc</emphasis> pour l’italique.`
HTML5	`balisage <i>ad hoc</i> pour l’italique.`
XML	`balisage <italic>ad hoc</italic> pour l’italique.`
XML	?
TEI	`balisage <hi rend="i">ad hoc</hi> pour l’italique.`
XML	`balisage <locutionEtrangere>ad hoc</locutionEtrangere> pour l’italique.`

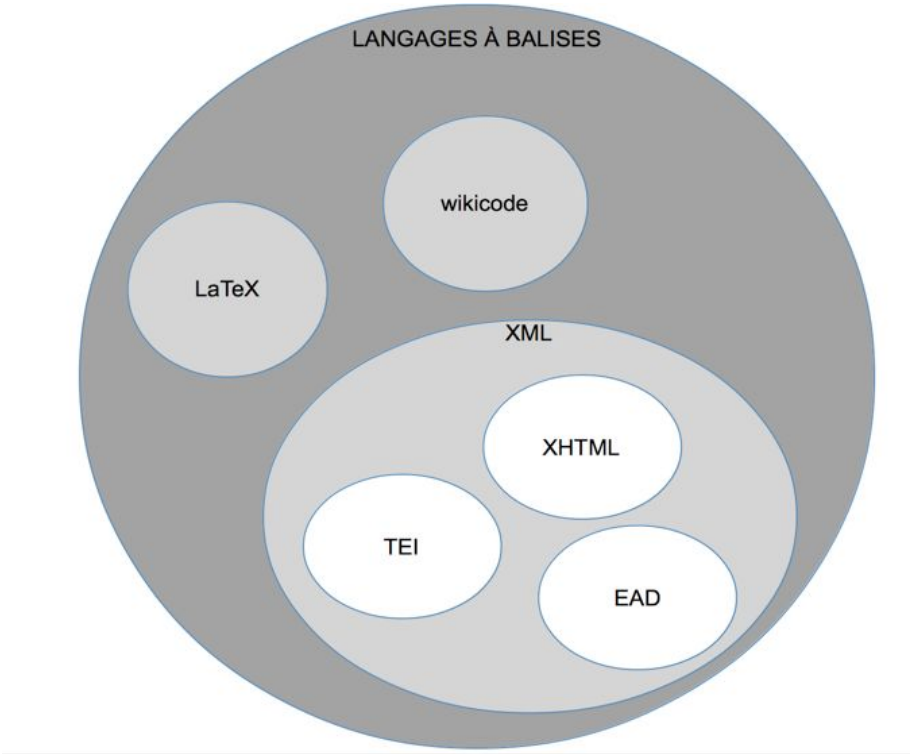
Mise en valeur (typographique) vs sémantique
1/2. Typographie

LaTeX	`\emph{ad hoc}`	pour mettre en valeur **certains mots**, les titres d'ouvrages en particulier.
wikicode	`"ad hoc"`	Il est possible de mettre le texte en gras, en italique, etc., pour **mettre en valeur les informations d'un texte** ou pour écrire le titre d'une œuvre selon les conventions (par exemple, un titre de film doit être en italique).
TEI	`<hi rend="i">ad hoc</hi>`	**distingue un mot ou une expression** comme graphiquement distincte du texte environnant, **sans en donner la raison**.
HTML5	`<i>italique</i>`	The i element represents a span of text in an alternate voice or mood, or otherwise offset from the normal prose in a manner **indicating a different quality of text, such as** a taxonomic designation, a technical term, an idiomatic phrase from another language, transliteration, a thought, or a ship name in Western texts.

Mise en valeur (typographique) vs sémantique
2/2. Balisage sémantique

LaTeX	`\selectlanguage{latin}{ad hoc}`	Pour alterner entre les langues, on utilise la commande en spécifiant entre accolades la langue demandée.
wikicode	`{{Langue|la|texte=ad hoc}}`	Ce modèle a pour but d'indiquer la langue d'un texte pour les synthétiseurs vocaux et l'indexation
TEI	`<foreign xml:lang="la">ad hoc</foreign>`	(étranger) reconnaît un mot ou une expression comme appartenant à une langue différente de celle du contexte.
HTML5	`<i lang="la">ad hoc</i>`	The lang attribute (in no namespace) specifies the primary language for the element's contents and for any of the element's attributes that contain text.

XML et les autres



exercice
modèles éditoriaux

Comparer les fichiers balisés (sémantique et modèle documentaire)

- LaTeX : apollinaire.tex
- DocBook : apollinaire_db.xml
- HTML5 : apollinaire.html
- XML/TEI : apollinaire_tei.xml

2. XML

eXtensible Markup Language

- X**ML** : **M**arkup **L**anguage, XML est un langage à balises.
- **X**ML : XML est e**X**tensible
  → permet de définir différents “espaces de noms” (*namespace*)
- XML ne propose pas un jeu prédéfini et fermé de balises, mais des règles sur ce que doit être un document **bien formé** et **valide**.
- Objectifs : faciliter l’échange de contenus
- Principes :
  - la structure d’un document XML est définie et validable par un **schéma** ;
  - un document XML est **transformable** en un autre document XML.

Structurer un document

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <métadonnées>
    <auteur>Vincent</auteur>
    <date>11 octobre 2017</date>
    <titre>XML. Introduction</titre>
  </métadonnées>
  <texte>
    <partie>
      <titre>Première partie</titre>
      <paragraphe>La première phrase <locEtrangere xml:lang="lat">ad hoc</locEtrangere> du
premier paragraphe.</paragraphe>
      <paragraphe>…</paragraphe> …
    </partie>
    <partie>
    …
    </partie>
  </texte>
</document>
```

Balise, élément

- Caractère Unicode (UTF-8)
- Balise
  - balise ouvrante : `<titre>`
  - balise fermante : `</titre>`
  - NB : une balise peut contenir d'autres balises.
- Élément
  - `<titre>Première partie</titre>`
  - = balise ouvrante + contenu + balise fermante.
- Nœud de type texte : “Première partie”

Attribut

```
<locEtrangere xml:lang="lat">
```

- Une paire nom="valeur", intégrée à la balise
- Séparé du nom de la balise par une espace
- Valeur de l'attribut entre *double quotes* : "valeur"

LE

MISANTHROPE.

ACTE PREMIER.

SCÈNE I.

PHILINTE, ALCESTE.

PHILINTE.

Qu'est-ce donc? qu'avez-vous?

ALCESTE, assis.

Laissez-moi, je vous prie.

PHILINTE.

Mais encor, dites-moi, quelle bizarrerie....

ALCESTE.

Laissez-moi là, vous dis-je, et courez vous cacher.

PHILINTE.

Mais on entend les gens au moins sans se fâcher.

ALCESTE.

Moi, je veux me fâcher, et ne veux point entendre.

PHILINTE

Dans vos brusques chagrins je ne puis vous comprendre,
Et, quoique amis enfin, je suis tout des premiers....

ALCESTE, se levant brusquement.

Moi, votre ami? Rayez cela de vos papiers.
J'ai fait jusques ici profession de l'être,
Mais, après ce qu'en vous je viens de voir paroître,

exercice
encoder cette page du Misanthrope

- De quels éléments avons-nous besoin pour encoder le début de cette scène ?
- Comment ces éléments vont-ils s'imbriquer ?

Le Misanthrope

Conformité : un document bien formé

```xml
<?xml version="1.0" encoding="UTF-8"?>
<piece xml:lang="fr">
  <titre>LE MISANTHROPE.</titre>
  <acte>
    <titre>ACTE PREMIER.</titre>
    <scene>
      <titre>SCÈNE 1.</titre>
      <personnages>PHILINTE, ALCESTE</personnages>
      <tourDeParole>PHILINTE.</tourDeParole>
      <vers>Qu'est-ce donc? qu'avez-vous ?</vers>
      <tourDeParole>ALCESTE<didascalie>, assis</didascalie>.</tourDeParole>
      <vers aligner="droite">Laissez-moi, je vous prie.</vers>
      <tourDeParole>PHILINTE.</tourDeParole>
      <vers>Mais encor, dites-moi, quelle bizarrerie....</vers>
      <tourDeParole>ALCESTE.</tourDeParole>
      <vers>Laissez-moi là, vous dis-je, et courez vous cacher.</vers>
      <tourDeParole>PHILINTE.</tourDeParole>
      <vers>Mais on entend les gens au moins sans se fâcher.</vers>
      <tourDeParole>ALCESTE.</tourDeParole>
      <vers>Moi, je veux me fâcher, et ne veux point entendre.</vers>
      <tourDeParole>PHILINTE</tourDeParole>
      <vers>Dans vos brusques chagrins je ne puis vous comprendre,</vers>
      <vers>Et quoique amis enfin, je suis tout des premiers....</vers>
      <tourDeParole>ALCESTE<didascalie>, se levant brusquement</didascalie>.</tourDeParole>
      <vers>Moi, votre ami? Rayez cela de vos papiers.</vers>
      <vers>J'ai fait jusques ici profession de l'être,</vers>
      <vers>Mais, après ce qu'en vous je viens de voir paraître,</vers>
    </scene>
  </acte>
</piece>
```

EXERCICE

Ouvrir le fichier dans Oxygen.

Faire des erreurs de syntaxe.

Comprendre ce qu'est un fichier **bien formé** :

- ?
- ?
- ?

http://corpus.enc.sorbonne.fr/cours/m2/moliere/

Validité : un document conforme au schéma

- XML ne propose pas de balises prédéfinies.
- La structure d'un document doit être spécifiée :
  - quelles balises ? quels attributs ?
  - quelles sont les règles d'imbrication de ces balises ?

Langages de schéma

- DTD (*Document Type Definition*)
  https://fr.wikipedia.org/wiki/Document_Type_Definition
- Relax-NG
  https://fr.wikipedia.org/wiki/Relax_NG
- XML Schéma (XSD), Schematron, ODD...

exercice
XML, schéma (validation)

Rédiger un petit paragraphe libre pour spécifier le schéma que nous avons défini intuitivement pour notre scène.

- formaliser une DTD ;
- ajouter cette DTD à notre transcription ;
- modifier le nom / la casse d'un élément ;
- ajouter une scène, un acte ;
- quels bénéfices tirons-nous de la validation ?
- Pouvons-nous échanger nos fichiers ?

```
<!DOCTYPE piece [
  <!ELEMENT piece (titre, acte+)>   // Une pièce a un titre, un ou plusieurs actes.
  <!ATTLIST piece xml:lang CDATA #REQUIRED>  // Une pièce a un attribut langue
  <!ELEMENT acte (titre, scene+)>  // Un acte a un titre, une ou plusieurs scènes
  <!ELEMENT scene (titre | personnages? | vers | tourDeParole)*>   // Une
scène a un titre, zéro à plusieurs personnages, des vers, des tours de paroles, et ce, à l'infini.
  <!ELEMENT titre (#PCDATA)>  // Le titre, c'est du texte
  <!ELEMENT personnages (#PCDATA)>  // Le personnage, c'est du texte
  <!ELEMENT tourDeParole (#PCDATA | didascalie)*>  // Le tour de parole, c'est du
texte ou un élément didascalie
  <!ELEMENT didascalie (#PCDATA)>  // La didascalie c'est du texte
  <!ELEMENT vers (#PCDATA)>  // Un vers, c'est du texte
  <!ATTLIST vers aligner (droite|centre) #IMPLIED>  // On peut spécifier
l'alignement d'un vers, sinon, c'est implicite.
]>
```

Misanthrope, DTD

Conclusion

- Document **bien formé** :
  document conforme aux règles XML (les balises ouvertes sont fermées, pas d'espace en trop, ?)
- Document **valide** :
  document conforme au schéma défini.

3. TEI

Partager des balises, échanger des fichiers

- 1987 : établissement de la Text Encoding Initiative.
- 1990 : TEI P1 (proposal 1), dir. Michael Sperberg-McQueen et Lou Burnard.
- 1992-1993 : TEI P2, expansion.
- 1994 : TEI P3, première version complète.
- 2000 : naissance du TEI Consortium.
- 2001-2004 : TEI P4, introduction du XML.
- 2007-... : TEI P5, abandon de SGML.
  Mises à jour fréquentes.

Board of directors, Technical Council, workgroups, SIG, près de 600 éléments.

Les “principes de Poughkeepsie” (1987)

Proposer des *Guidelines* (recommandations) avec pour objectifs :

1. Provide a standard format for data interchange in humanities research.
2. Suggest principles for the encoding of texts in the same format.
3. Define (a) a recommended syntax for the format, (b) a metalanguage for the description of text-encoding schemes, (c) describe the new format and representative existing schemes both in that metalanguage and in prose ;
4. propose sets of coding conventions suited for various applications.
5. include a minimal set of conventions for encoding new texts in the format.
6. The guidelines are to be drafted by committees on text documentation, text representation, text interpretation and analysis, metalanguage definition and description of existing and proposed schemes, coordinated by a steering committee of representatives of the principal sponsoring organizations.
7. Compatibility with existing standards will be maintained as far as possible.
8. A number of large text archives have agreed in principle to support the guidelines in their function as an interchange format. We encourage funding agencies to support development of tools to facilitate this interchange.
9. Conversion of existing machine-readable texts to the new format involves the translation of their conventions into the syntax of the new format. No requirements will be made for the addition of information not already coded in the texts.

Guidelines

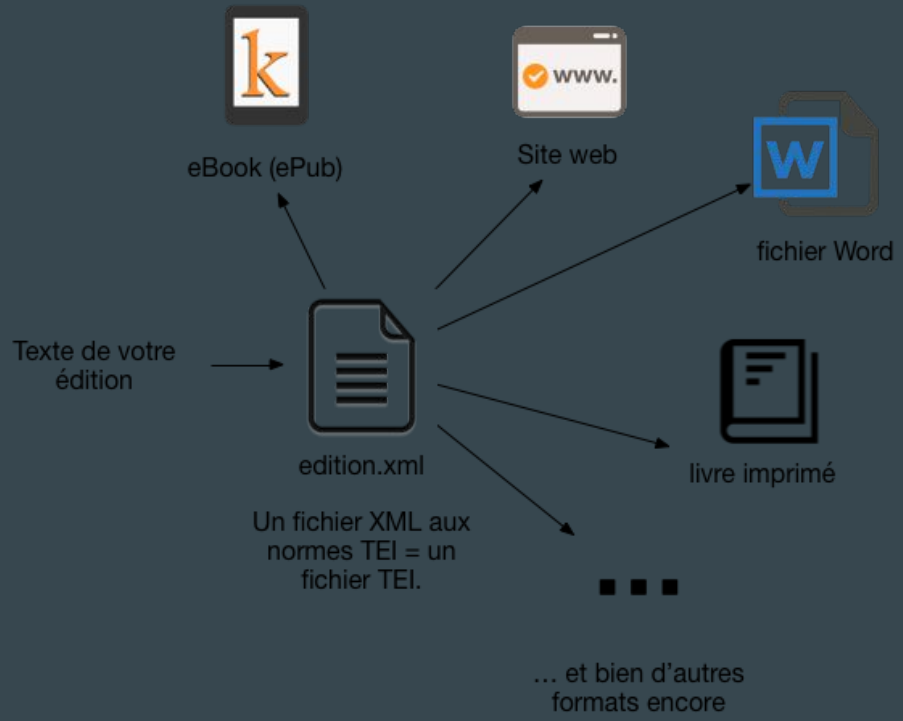
Un réservoir d'éléments et une logique pour décrire votre texte.

- Des modules, dont 4 obligatoires (communs à tous les docs TEI) :
  - `tei` : définition des classes, macros et types de données
  - `textstructure` : éléments de base pour structurer un texte de type livre
  - `core` : éléments disponibles dans tous les documents TEI
  - `header` : en-tête TEI (métadonnées du document)
- Plus de 569 éléments

TEI (All) n'est pas un schéma à proprement parler.
Mais plutôt un *framework*, utile à la conception de son propre schéma.

- Pérennité du format
- Séparer source et vue (possibilité de supports multiples en sortie)
- Un format pivot ?
- Interopérabilité ?

TEI, un format pivot ?



Guidelines, exercice

En consultant les *Guidelines*, proposer un encodage TEI pour nos 3 premières pages d'Apollinaire :

- Dans Oxygen, reprendre l'encodage de apollinaire.html
- Comment automatiser ces reprises (changement du nom des éléments, etc.) ?
- Comment lire notre fichier TEI ?
- Comment produire un fichier texte brut à partir d'un tel balisage ?
- Reprendre la DTD pour valider notre document TEI.

Personnaliser et documenter

Il est fortement déconseillé d'utiliser un schéma englobant l'intégralité de la TEI : une phase importante d'un projet est la conception d'un modèle adapté aux données et au projet, à l'exploitation des documents.
Documenter pour rendre ces choix lisibles et réexploitables par un groupe plus large ou d'autres chercheurs.

Outil

Roma : application de génération de schémas TEI, permettant d'opérer les choix principaux de modèle et de générer un fichier utilisant la syntaxe ODD (*One Document Does it all*, fichier TEI utilisant le module `tagdocs`), ainsi que le schéma demandé au format souhaité.