

Créer des styles bibliographiques personnalisés CSL

1. Introduction

Rappel 1 : le répertoire **styles**

Où puis-je trouver les styles CSL dans Zotero ? Localiser le répertoire de données **zotero** , puis répertoire **styles** .

Rappel 2 : qualité des données

Tous les problèmes de mise en forme ne proviennent pas des styles bibliographiques, vérifiez la qualité et la complétude des données de votre bibliothèque Zotero en premier lieu !

Rappel 3 : clarté des consignes

Le plus difficile n'est pas forcément d'encoder le style, mais de disposer de consignes claires et précises, traduisibles dans un langage informatique.

Un exemple à suivre : la page de documentation du style Infoclio. Cette page précise en outre de façon détaillée les champs Zotero utilisés pour chaque type de document.

Style de citation [infoclio.ch](https://www.infoclio.ch) - Instructions pour Zotero. (n.d.). [infoclio.ch](https://www.infoclio.ch). Repéré à <https://www.infoclio.ch/fr/node/139551>

Rappel 4 : les types de style

Un style bibliographique définit la façon dont les éléments bibliographiques d'un document :

- appels de citation,
- **notes** si applicable,
- **bibliographie**,

doivent être organisés et formatés.

On distingue différents types de style bibliographiques, correspondant à des formats de style Zotero différents. On trouve ainsi dans [l'entrepôt des styles Zotero](#), près de 9000 styles regroupés en 2 grands ensembles :

- les styles **in-text**, dans lesquels un appel de citation dans le texte renvoie à une entrée de la bibliographie,
- les styles **note**, dans lesquels un marqueur (symbole, nombre, etc.) pointe vers une note de bas de page ou de fin, qui permet d'identifier le document cité.

Chaque type de style a sa logique propre, il peut être tentant de les mélanger, mais cela peut (souvent) aboutir à un résultat incohérent ou difficile à comprendre.

Les styles *in-text*

Les styles numériques

Dans le texte : chaque référence citée est appelée par un **numéro**

Yeast cells were grown at 25°C in batch cultures on 0.5% methanol for 36 hours [21, 22].

Dans la bibliographie : les références sont classées par **ordre d'apparition dans le texte**.

21. Zwart KB, Veenhuis M, Harder W (1983) Significance of yeast peroxisomes in the metabolism of choline and ethanolamine. *Antonie Van Leeuwenhoek* 49: 369-385.

22. van der Klei IJ, Harder W, Veenhuis M (1991) Methanol metabolism in a peroxisome-deficient mutant of *Hansenula polymorpha*: a physiological study. *Arch Microbiol* 156: 15-23.

Les styles numériques composites

Les styles numériques composites, dans lesquels une entrée bibliographique peut contenir plusieurs références, sont très utilisés en chimie. **Ce type de style n'est pas pris en charge par CSL.**

Dans le texte :

Yeast cells were grown at 25°C in batch cultures on 0.5% methanol for 36 hours (1)

Dans la bibliographie :

1. a) Zwart KB, et al. (1983) *Antonie van Leeuwenhoek* 49: 369-385, b) van der Klei IJ, et al. (1991) *Arch Microbiol* 156: 15-23.

Les styles auteur et auteur-date

Dans le texte : chaque référence citée est appelée par le **nom de l'auteur** ou le **nom de l'auteur et la date de publication**.

Yeast cells were grown at 25°C in batch cultures on 0.5% methanol for 36 hours (van der Klei et al. 1991; Zwart et al. 1983)

Yeast cells were grown at 25°C in batch cultures on 0.5% methanol for 36 hours (van der Klei et al.; Zwart et al.)

Dans la bibliographie : les références sont classées par **ordre alphabétique d'auteur**.

van der Klei IJ, Harder W, Veenhuis M (1991) Methanol metabolism in a peroxisome-deficient mutant of *Hansenula polymorpha*: a physiological study. *Arch Microbiol* 156: 15-23.

Zwart KB, Veenhuis M, Harder W (1983) Significance of yeast peroxisomes in the metabolism of choline and ethanolamine. *Antonie Van Leeuwenhoek* 49: 369-385.

Les styles label

Dans le texte : chaque référence citée est appelée par un **code**.

Yeast cells were grown at 25°C in batch cultures on 0.5% methanol for 36 hours [ZwVH1983; vaHV1991].

Dans la bibliographie : les références sont classées par **ordre d'apparition dans le texte**.

[ZwVH1983] Zwart KB, Veenhuis M, Harder W (1983) Significance of yeast peroxisomes in the metabolism of choline and ethanolamine. *Antonie Van Leeuwenhoek* 49: 369-385.

[vaHV1991] van der Klei IJ, Harder W, Veenhuis M (1991) Methanol metabolism in a peroxisome-deficient mutant of *Hansenula polymorpha*: a physiological study. *Arch Microbiol* 156: 15-23.

Un seul style correspond à ce modèle dans l'entrepôt des styles, le style [DIN 1505-2 \(alphanumeric, German\)](#). Il présente 2 limites :

- il est monolingue en allemand,
- le schéma de construction du label ne peut être modifié.

Un label personnalisé peut être défini, mais cela devra être fait manuellement pour chaque référence.

- Pour chaque référence le label à utiliser devra être indiqué dans le champ *Extra* de la notice Zotero, sous la forme : `citation-label: label`.
- Par conséquent, il faudra **désambigüiser** manuellement des labels qui seraient identiques mais se rapporteraient à des articles différents (même premier auteur et même année).

Les styles *note*

Dans le texte :

"In the Island of St. Kilda, according to Martin, [*] the men do not acquire beards until the age of thirty or upwards, and even then the beards are very thin. "

[*]'Voyage to St. Kilda' (3rd edit. 1753), p. 37.

Dans la bibliographie : certains styles *note* incluent une bibliographie de tous les documents cités. Les références sont en général classées **par ordre alphabétique d'auteur**.

Le format du marqueur doit être paramétré dans le **traitement de texte**.

Exemple dans Word 2010

□

2. Principes généraux de CSL

Principes du langage XML : 6 basiques de XML à connaître

Principe 1 - Prologue XML

C'est la première ligne du fichier CSL. Il contient la déclaration XML et spécifie le codage des caractères. Il se présentera ainsi dans la plupart des cas.

```
<?xml version="1.0" encoding="utf-8"?>
```

Principe 2 - Eléments et hiérarchie

Les éléments sont les blocs de base avec lesquels un fichier XML est construit. Ils peuvent être imbriqués hiérarchiquement : on parle d'éléments **parent** et d'éléments **enfant**. Le premier élément est l'élément **racine** (**style** dans le langage CSL), duquel tous les éléments dépendent. On **indente** généralement les éléments enfant par des espaces ou des tabulations pour faciliter la compréhension.

Chaque élément est introduit par une balise **<ouvrante>** et clos par une balise **</fermante>**, ou une barre oblique s'il n'a pas de contenu. Toute balise ouverte doit impérativement être fermée.

Ex:

```
<element parent>
  <element enfant1 attribut="valeur"/>
  <element enfant2> text </element enfant2>
</element parent>
```

Principe 3 - Attributs et contenu de l'élément

Un élément peut être qualifié et contenir des informations de deux manières :

- soit par du **contenu textuel** inséré entre la balise ouvrante et la balise fermante, ex:

```
<author>
  <name>Anton Perdoncin</name>
  <email>anton.perdoncin@gmail.com</email>
</author>
```

- soit par des **attributs**. Si l'ordre des attributs importe peu, une valeur doit obligatoirement être renseignée, entre guillemets. Ex :

```
<link href="http://traces.revues.org" rel="documentation"/>
```

Principe 4 - Echappement

Pour éviter toute ambiguïté dans l'écriture XML, certains caractères significatifs pour la syntaxe XML doivent être substitués par d'autres lorsqu'ils sont utilisés dans un attribut ou dans le contenu textuel d'un élément. Les séquences d'échappement sont les suivantes :

- `<` pour <
- `>` pour >
- `&` pour &
- `'` pour '
- `"` pour "

Principe 5 - Commentaires

Des commentaires pour expliciter des choix d'écriture ou clarifier des points peuvent être introduits sous la forme suivante : `<!-- commentaire libre à rédiger -->`. Ils seront reconnus par les applications utilisant le fichier comme du commentaire et non du code.

Principe 6 - Fichier bien formé et valide

Contrairement à HTML, XML ne pardonne aucune erreur de syntaxe. Toute erreur (oubli d'une balise fermante, échappement incorrect, etc.) empêchera le fichier XML de fonctionner. Il convient donc de s'assurer que le fichier CSL fonctionne correctement en vérifiant qu'il est :

- **bien formé**, i. e. qu'il respecte les règles de XML et ne contient pas d'erreur d'encodage,
- **valide**, i. e. qu'il est conforme aux règles du schéma CSL, qui décrit tous les éléments CSL, leurs attributs et leurs règles d'utilisation.

Adapté de : Zelle, R. M. (n.d.). Primer - An Introduction to CSL: Understanding CSL Styles: XML Basics. *Citation Style Language 1.0.1-dev documentation*. Repéré à <http://docs.citationstyles.org/en/stable/primer.html#xml-basics>

Principes du langage CSL : le jeu des 7 erreurs

□

3. Structure d'un style CSL

Structure générale d'un style CSL

Un style CSL est structuré en plusieurs éléments.

- **style** : élément racine - précise notamment la version de CSL, le type de style et permet des paramétrages globaux : gestion des noms à particules avec l'attribut `demote-non-dropping`, de l'indication du nombre de pages, abréviation des prénoms composés. [Spécification CSL : élément style](#)
 - **info** : métadonnées décrivant le style (nom, auteur, etc.) - il faut notamment modifier les éléments **title**
-

et `id` du nouveau style que l'on crée à partir d'un autre, sinon le style créé sera écrasé lors des mises à jour du style duquel il est dérivé. [Spécification CSL : élément `info`](#)

- `citation` : décrit la façon dont les appels de citation ou les notes (pour les styles **note**) sont formatés. [Spécification CSL : élément `citation`](#)
- `bibliography` : décrit la façon dont la bibliographie est formatée. [Spécification CSL : élément `bibliography`](#)
- `locale` : permet de spécifier des termes, formats de dates et options de formatage différents de ceux prévus par défaut pour la langue. [Spécification CSL : élément `locale`](#)
- `terms` : permet la modification de chaînes de caractères spécifiques (ex. remplacer « edited by » par « ed. by »). [Spécification CSL : élément `terms`](#)
- `macro` : permet la réutilisation des règles de formatage dans `citation` et `bibliography` et des styles plus compacts. [Spécification CSL : élément `macro`](#)

Une `macro` CSL permet de définir des règles et d'attribuer à cet ensemble de règles un nom : appeler la macro en indiquant son nom exécutera la commande et appliquera les règles définies. On peut ainsi écrire une seule fois une longue séquence de paramétrage pour un élément donné, et y faire ensuite référence lorsque l'on souhaite l'appliquer.

Les `macro` économisent du code, et surtout facilitent la réutilisation et la modification des styles. Pour modifier le formatage des noms d'auteur par exemple, il suffit ainsi de modifier la `macro` correspondante, ou encore de copier-coller la `macro` d'un autre style.

L'utilisation des macro fait partie des bonnes pratiques recommandées par Sebastian Karcher (plus connu des utilisateurs de Zotero sous son pseudonyme "**adamsmith**") :

Bonne pratique 2 : Utilisez abondamment les macros. Gardez les éléments `citation` et `bibliography` brefs et ne leur intégrez qu'au minimum des éléments `choose`.

Source : adamsmith. (2013, 28 octobre). Writing CSL - Features and Best Practices. *The Zoteroist*. Repéré à <https://zoteromusings.wordpress.com/2013/10/28/writing-csl-features-and-best-practices/>

Retenons également la bonne pratique 3 recommandées dans ce billet par adamsmith, nous reviendrons sur la bonne pratique 1 un peu plus tard.

Bonne pratique 3 : Utilisez des `terms` et des `labels`. N'ajoutez pas de termes ou d'expressions dans des affixes ou en utilisant `text value=`.

Anatomie des éléments `style` et `info`

Type de style : l'attribut `class`

Style agnostique du point de vue de la **langue** vs. style localisé

On en parlera dans les *Miscellanées* ; pour rendre agnostique du point de vue de la langue un style localisé, il suffit de supprimer l'attribut `default-locale`.

Style dépendant vs. style indépendant.

Dans un souci d'économie de code, certains styles sont dépendants d'un autre style. Dans ce cas là, le fichier CSL se réduit à un élément info. La référence au style principal est indiquée par l'attribut

`rel="independent-parent"`. Exemple du style [Loisir et Société / Society and Leisure](#)

```
<?xml version="1.0" encoding="utf-8"?>
<style xmlns="http://purl.org/net/xbiblio/csl" version="1.0" default-locale="en-US">
  <!-- Generated with https://github.com/citation-style-
  language/utilities/tree/master/generate_dependent_styles/data/taylor-and-francis -->
  <info>
    <title>Loisir et Société / Society and Leisure</title>
    <id>http://www.zotero.org/styles/loisir-et-societe-society-and-leisure</id>
    <link href="http://www.zotero.org/styles/loisir-et-societe-society-and-leisure" rel="self"/>
    <link href="http://www.zotero.org/styles/apa" rel="independent-parent"/>
    <link href="http://www.tandfonline.com/action/authorSubmission?journalCode=RLES20&page=instructions"
    rel="documentation"/>
    <category citation-format="author-date"/>
    <category field="political_science"/>
    <issn>0705-3436</issn>
    <eissn>1705-0154</eissn>
    <updated>2014-05-17T16:39:46+00:00</updated>
    <rights license="http://creativecommons.org/licenses/by-sa/3.0/">This work is licensed under a Creative Commons
    Attribution-ShareAlike 3.0 License</rights>
  </info>
</style>
```

Les principaux types d'éléments

On manipule dans les appels de citation, les notes et la bibliographie des données de nature différente, auxquelles des paramétrages spécifiques peuvent être appliqués. A chaque type de donnée correspond ainsi un type d'élément CSL.

D'autres éléments, `group` et `choose`, par exemple, sont davantage des éléments de **rédaction** que de **contenu**.

- `text` : le texte à afficher peut être celui d'une `variable`, d'une `macro`, d'un `term` ou d'une `value`
- `date` : date de publication, date de consultation, etc.
- `names` : noms des auteurs, des éditeurs scientifiques, des éditeurs commerciaux, etc.
- `number` : volume, numéro, numéro dans la série, etc.
- `choose` : créer des conditions - voir *infra* les différents types de condition
- `label` : étiquette de certaines données, par exemple page ou p. pour page, vol. ou volume pour volume, sous la dir. de pour l'éditeur scientifique, etc.
- `group` : simplifie l'écriture d'un style en :
 - définissant la ponctuation au niveau d'un groupe d'éléments plutôt qu'élément par élément,
 - conditionnant l'affichage de certaines données à la présence d'autres.

C'est le moment de revenir sur les bonnes pratiques d'écriture recommandées par adamsmith, avec la bonne

pratique 1.

Source : adamsmith. (2013, 28 octobre). Writing CSL - Features and Best Practices. *The Zoteroist*. Repéré à <https://zoteromusings.wordpress.com/2013/10/28/writing-csl-features-and-best-practices/>

Bonne pratique 1 : Préférez les `group` et les `group delimiters` aux affixes pour la ponctuation et les espaces entre les objets.

Et rappelons la bonne pratique 3.

Bonne pratique 3 : Utilisez des `terms` et des `labels`. N'ajoutez pas de termes ou d'expressions dans des affixes ou en utilisant `text value=`.

4. Les outils

□

Importez dans votre bibliothèque le [fichier zotero-csl.rdf](#).

Installez le style [Elsevier - Harvard \(with titles\)](#).

Téléchargez les [consignes du style Garni](#).

5. L'éditeur de styles Zotero

Un billet récent du blog Zotero francophone [Quel outil pour éditer des styles CSL?](#) détaille les différents outils disponibles pour l'édition de styles CSL.

Aujourd'hui nous nous limitons :

- à l'éditeur de styles Zotero,
- au validateur en ligne.

Les outils CSL intégrés à Zotero

Zotero possède un outil intégré pour éditer les styles, accessible depuis les *Préférences* > *Citer*.

□

Comme on le voit sur cette copie d'écran, Zotero dispose en fait de deux outils de style, l'*Aperçu des styles* et l'*Editeur de style*. Commençons par une brève présentation du premier.

□

L'*Aperçu des styles* vous permet de comparer rapidement tout ou partie des styles que vous avez installés, en générant des appels de citation et une bibliographie à partir d'éléments sélectionnés dans votre bibliothèque.

Afin que cet aperçu soit significatif et parlant, il importe de s'assurer de deux points.

Tout d'abord, assurez-vous que ces éléments reflètent la **diversité de types de documents** que vous allez citer (article, chapitre, mais aussi thèse ou encore brevet ou film) : vous vérifierez ainsi que tous ces types sont bien pris en compte par les styles que vous comparez. Le style *Nature*, par exemple, ne sera pas le plus adapté si vous citez des documents non publiés comme les thèses. La revue *Nature* demande aux auteurs de citer un nombre restreint de types de document (voir les [instructions aux auteurs](#), rubrique *References*), aussi le style CSL *Nature* encode-t-il le formatage des références uniquement pour ces types de document.

Ensuite, et surtout, assurez-vous que ces éléments sont complets et exacts.

- **S'il manque des informations bibliographiques**, il est logique que le style ne puisse pas les afficher lorsqu'il génère les citations et la bibliographie.
- **Si vous faites un usage inadéquat ou détourné de certains champs**, vous générerez également une bibliographie détournée ou inadéquate, ne correspondant pas au résultat que peut produire le style normalement.

Ces deux points sont tout aussi importants lorsque vous utilisez l'*Editeur de style*.

Si l'*Aperçu des styles* permet de manipuler plusieurs styles à des seules fins de visualisation, l'*Editeur de style* se limite à l'affichage d'un seul style mais en vue de son édition.

□

L'écran est composé de 3 zones, avec, de bas en haut :

- le panneau de **prévisualisation**, dans lequel vous voyez évoluer en temps réel le formatage des citations et de la bibliographie au fur et à mesure des modifications que vous apportez au fichier de style,
- le panneau d'**édition**,
- la barre d'outils.

Pourquoi recommandons-nous cet outil, qui semble rudimentaire et pourvu de fonctionnalités limitées?

Avantages de l'éditeur de styles Zotero

- Toute erreur de syntaxe XML donne lieu à un message d'erreur immédiat : le panneau de prévisualisation affiche le message suivant, aussi voit-on tout de suite que son fichier n'est pas bien formé.

Erreur dans la syntaxe du style :
Error: File is not valid XML

- On prévisualise en direct les modifications de paramétrage du style, et ce à partir des exemples de document de sa bibliothèque.
- On peut tester en direct les paramétrages liés à des situations de citation particulières : ainsi lorsque l'on précise une mention de page ou de paragraphe (menu déroulant *Page*), ou que l'on supprime le nom de l'auteur d'un appel de citation dans un style de type auteur-date. Ou encore, dans un style de type *note*, lorsque l'on cite à plusieurs reprises la même référence (gestion des *ibid*, *op. cit.*, etc. - menu déroulant *Position de la citation*).

□

Limites de l'éditeur de style Zotero

- L'éditeur n'intègre pas d'aide à la saisie comme la **coloration syntaxique** ou l'**indentation** du code.
- Il ne contrôle pas la **validité** CSL : on peut créer un style dont la syntaxe XML est correcte, mais qui ne soit pas conforme aux spécifications CSL et qu'on ne pourra donc pas installer dans Zotero.
- L'écran est partagé en 2 bandeaux horizontaux, ce qui peut occasionner une lisibilité réduite en fonction de la taille et de l'orientation de son écran.

Le validateur en ligne

L'éditeur Zotero n'assure pas la validation CSL, indispensable pour installer et utiliser votre style, et bien sûr pour [le soumettre au dépôt CSL](#) si son champ d'application dépasse un usage personnel ou local.

Le validateur en ligne <http://validator.citationstyles.org/> vérifie la validité de votre fichier, et surtout affiche le cas échéant de façon détaillée les erreurs à corriger.

Valdateur en ligne : vue d'ensemble

□

Valdateur en ligne : focus sur les messages d'erreur

□

Avant de commencer : enregistrer un style créé avec l'Editeur de style Zotero

Une fois votre nouveau style créé, il faut générer le fichier CSL correspondant et installer ce fichier dans Zotero.

Etape 1. Rendre son style unique : se prémunir de l'écrasement de son style

Chaque style est identifié dans l'élément **info** par :

- un nom (élément **title**),
- et surtout un identifiant (élément **id**).

Avant tout travail d'édition sur un style existant, modifiez le contenu de ces éléments pour ne pas confondre votre nouveau style avec le style dont il est dérivé, et surtout pour éviter qu'il ne soit écrasé par une mise à jour de ce dernier.

```
<info>
  <title>Mon style</title>
  <id>http://www.zotero.org/styles/mon_style</id>
  <link href="http://www.zotero.org/styles/nature"/>`
  ...
```

Etape 2. Générer le fichier CSL

Le bouton *Enregistrer sous...* de l'éditeur de style génère un fichier CSL, qu'il suffit d'enregistrer en veillant à bien spécifier l'extension **.csl** (et à ne pas laisser Windows ajouter une extension .txt).

NB : Votre travail d'édition n'est pas sauvegardé ni enregistré tant que vous n'avez pas généré le fichier CSL.

Etape 3. Installer le style dans Zotero

Deux méthodes sont possibles.

- Glisser-déplacer le fichier dans votre bibliothèque : Zotero détecte qu'il s'agit d'un fichier de style et vous propose de l'installer en tant que nouveau style ou d'actualiser le style existant si un style ayant le même `id` est déjà référencé.
- Utiliser le bouton + du *Gestionnaire de styles*.

NB : Les styles ne sont pas synchronisés : si vous utilisez Zotero sur plusieurs ordinateurs, il vous faudra installer le style sur tous ces ordinateurs.

6. Modifier un style : macro simple

Exercice de style 1

□

Modifiez le style *Elsevier - Harvard (with titles)* pour que les **auteurs** soient mis en forme selon les consignes du style Garni.

- Attention, il faut bien considérer à la fois les appels de citation **ET** la bibliographie.
- Attention, certains paramétrages par défaut sont **implicites** : si on souhaite les modifier il ne faut pas seulement modifier la valeur d'un attribut mais ajouter un nouvel attribut avec la valeur appropriée.

Un peu de vocabulaire et de syntaxe pour les noms

Spécifiques aux noms

Nous avons vu les attributs suivants dans notre exemple.

- `sort-separator` : caractère entre le nom et le prénom
- `delimiter` : caractère entre chaque nom d'auteur
- `and` : affichage du *et* entre le pénultième et le dernier nom d'auteur - valeurs `symbol` ou `text`
- `initialize-with` : prénom remplacé par une initiale et éventuellement un caractère - si cet élément n'est pas présent, les prénoms sont restitués en entier
- `delimiter-precedes-last` : caractère de délimitation avant le nom du dernier auteur - valeurs `contextual`, `after-inverted-name`, `always` ou `never`
- `name-as-sort-order` : le nom du ou des auteurs s'affiche selon l'ordre nom-prénom - valeurs `all` ou `first`

L'élément `et-al` définit quant à lui le texte à afficher et la mise en forme à appliquer au *et al.* [Spécification CSL](#)
: élément `et-al`

Non spécifiques aux noms mais utiles pour les noms : grasse, casse, etc.

Les attributs suivants permettent de modifier facilement des caractéristiques telles que la casse ou la graisse.

- `font-style="normal"` , `"italic"` ou `"oblique"`
- `font-variant="normal"` ou `"small-caps"`
- `font-weight="normal"` , `"bold"` ou `"light"`

Modifier une partie du nom

Exemple : nom de famille en petites capitales et prénom en minuscules

```
<names variable="author">
  <name sort-separator=" " name-as-sort-order="all" et-al-min="4" et-al-use-first="3">
    <name-part name="family" font-variant="small-caps"/>
  </name>
  ...
</names>
```

Définir un substitut au nom d'auteur

L'élément `substitute` définit le substitut au nom de l'auteur à utiliser lorsque le champ *Auteur* de la notice Zotero est vide. Il doit faire référence à une `macro` ou à une `variable` , on ne peut pas utiliser un élément de type `text term` ou `text value` . [Spécification CSL : élément `substitute`](#)

Le style *Elsevier-Harvard (with titles)* utilise :

- l'éditeur scientifique,
- à défaut le traducteur,
- à défaut le titre.

```
<macro name="author">
  <names variable="author">
    <name name-as-sort-order="all" sort-separator="," initialize-with="." delimiter="," delimiter-precedes-last="always"/>
    <label form="short" prefix=" (" suffix=")" text-case="capitalize-first"/>
    <substitute>
      <names variable="editor"/>
      <names variable="translator"/>
      <text macro="title"/>
    </substitute>
  </names>
</macro>
```

7. Modifier un style : macro conditionnelle

Exercice de style 2

□

Modifiez le style *Elsevier - Harvard (with titles)* pour que les **titres** soient mis en forme selon les consignes du style Garni.

La `macro name="title"` du style est un peu plus complexe que la `macro name="author"` , car elle définit une

mise en forme conditionnelle en fonction du type de document. Les conditions sont exprimées dans l'élément `choose`.

- `if` définit la 1ère condition : si le document est une thèse ou un rapport, le titre est affiché, suivi entre parenthèses du type de thèse ou de rapport et de son numéro.
- `else-if` définissent les autres conditions : si le document, par exemple, est une page web, le titre est affiché, suivi de la mention entre crochets *WWW Document*.
- `else` définit ce qui s'applique dans tous les autres cas : si le document est d'un autre type que ceux indiqués dans les éléments `if` et `else-if`, le titre est affiché.

Exercice de style 3

□

Modifiez le style *Elsevier - Harvard (with titles)* pour que le formatage des volumes, numéros et pages pour les articles de revues corresponde aux consignes du style Garni.

8. Les différents types de condition

L'élément `choose` peut avoir pour parent un autre élément que `macro` : il peut être élément enfant de l'élément `layout` pour le **paramétrage des notes** par exemple. Les différents types de condition sont exprimés par les attributs possibles pour les éléments `if` et `else-if`.

NB on peut indiquer plusieurs valeurs, sauf pour l'attribut `desambiguate`.

- `desambiguate="true"` (seule valeur possible de l'attribut) : la condition n'est réalisée que si elle permet de désambiguer
- `is-numeric="variable"` valeur de l'attribut = variable qui doit avoir un contenu numérique pour que la condition se réalise
- `is-uncertain-date="date"` valeur de l'attribut = date qui doit être incertaine pour que la condition se réalise [NB Zotero ne gère pas les dates incertaines]
- `locator="XXX"` valeur de l'attribut = `locator` auquel la condition s'applique
- `position="XXX"` cf. infra, utilisées pour les notes de bas de page ou de fin
- `type="XXX"` valeur de l'attribut = type de document auquel la condition s'applique
- `variable="variable"` valeur de l'attribut = variable dont l'absence ou la présence conditionne la réalisation de la condition

L'attribut `match` définit l'opérateur à utiliser.

- `match="any"` = OU = la condition se réalise si au moins un des critères est rempli

ex : `<if type="book thesis" match="any">` la condition se réalise si le document est de type livre OU thèse

- `match="all"` = ET = la condition se réalise si TOUS les critères sont remplis
- `match="none"` = NON = la condition se réalise si AUCUN des critères n'est rempli

ex : `<if variable="volume" match="none">` la condition se réalise si la variable volume est absente, *i. e.* si le

champ *Volume* de la notice Zotero est vide.

Spécification CSL : élément `choose`

9. Les appels de citation et les notes = élément `citation`

Structure de l'élément `citation`

L'élément `citation` permet de paramétrer les appels de citation pour les styles *in-text*, et les notes (de bas de page ou de fin) pour les styles *note*. Il comporte deux éléments :

- `sort` : définit les critères de tri des références multiples,
- `layout` : spécifie les informations à inclure (et leur formatage s'il n'a pas été défini dans des `macro`).

Les attributs et les possibilités de mise en forme diffèrent en fonction du type de style.

Le paramétrage pour les styles **numériques** demeurant simple et sommaire, nous nous concentrons sur les spécificités des styles auteur-date et note.

```
<citation nom_attribut="valeur_à_appliquer"/>
  <sort>
    <key variable="variable_critère_de_tri_pour_les_références_multiples"/>
    <key macro="résultat_de_la_macro_critère_de_tri_pour_les_références_multiples" sort="ascending" ou "descending"/>
  </sort>
  <layout >
    <text variable="variable_à_afficher" préfixe, suffixe, etc./>
    <text macro="résultat_de_la_macro_à_afficher"/>
  </layout>
</citation>
```

Exercice de style 4

□

```
<citation collapse="citation-number">
  <sort>
    <key variable="citation-number"/>
  </sort>
  <layout vertical-align="sup" delimiter=",">
    <text variable="citation-number"/>
  </layout>
</citation>
```

Comment les appels de citation sont-ils formatés dans ce style?

Styles auteur-date

Désambiguïser

Dans les styles auteur-date, l'élément `citation` doit comporter des règles de désambiguïsation des appels de citation (documents distincts ayant le même premier auteur et la même date de publication). CSL envisage

cette désambiguïsation selon 3 étapes, exprimées par des attributs de l'élément `citation`.

Présentons en bref les 3 étapes détaillées dans [Spécification CSL : Disambiguation](#)

Etape 1

`disambiguate-add-names` : par l'ajout d'un nom, sans tenir compte du paramétrage du *et al*

Etape 2

`disambiguate-add-givenname` : par l'ajout d'un prénom, par ex. J. Doe, 2005 et M. Doe, 2005

`givenname-disambiguate-rule` : traitement des prénoms (tous les prénoms, initiales, etc.)

Etape 3

`disambiguate-add-year-suffix` : par l'ajout d'un suffixe à l'année, par ex. : 2007a, 2007b etc.

Grouper

Il s'agit, dans un appel de citation multiple, de regrouper les références par auteur. Ce regroupement est activé par l'attribut `cite-group-delimiter`. Ainsi (Doe 1999; Smith 2002; Doe 2006; Doe et al. 2007) devient (Doe 1999; Doe 2006; Smith 2002; Doe et al. 2007).

[Spécification CSL : Cite Grouping](#)

Compacter

Il s'agit non seulement de regrouper, mais plus encore de compacter les références dans un appel de citation multiple. L'attribut `collapse` active le regroupement et le compactage et peut avoir les valeurs suivantes.

- `"year"` : compacte les références d'un même auteur aux dates, par ex. : (Doe 2000, Doe 2001) devient (Doe 2000, 2001).
- `"year-suffix"` : idem que le précédent, mais compacte aussi des années identiques, par ex. : (Doe 2000a, Doe 2000b) devient (Doe 2000a, b).
- `"year-suffix-ranged"` : idem que le précédent, mais compacte aussi les suffixes, par ex. : (Doe 2000a, b, c, e) devient (Doe 2000a-c, e).

[Spécification CSL : Cite Collapsing](#)

Paramétrer le *et al*

Plusieurs attributs de l'élément `citation` permettent de définir les paramètres d'affichage du *et al*.

- `et-al-min` : nbre d'auteurs minimum pour afficher la mention *et al*
- `et-al-use-first` : nbre d'auteurs à inclure avant *et al*
- `et-al-subsequent-min` et `et-al-subsequent-use-first` : idem supra, pour les références déjà citées
- `et-al-use-last` : remplace *et al* par « ... » et le nom du dernier auteur
- `delimiter-precedes-et-al` : caractère précédant *et al*

Exercice de style 5

□

Modifiez le style *Elsevier - Harvard (with titles)* pour que la mise en forme des appels de citation corresponde aux consignes du style Garni.

Styles note

Pourquoi l'élément `citation` est-il plus complexe dans les style **note**?

D'une part les appels de citation affichent davantage de composants (titre, éditeur, etc.) pour constituer une référence bibliographique abrégée.

D'autre part, les éléments à afficher peuvent différer en fonction de la **position** relative de la référence citée, et de la présence ou non d'une spécification de page, de chapitre, etc., *i. e.* d'un `locator`.

Ibid, op. cit., etc.

On retrouve ici l'élément `choose` et la définition de conditions.

C'est en effet l'attribut `position` d'un élément `if` ou `else-if` qui détermine le contenu à afficher en fonction de la position de la citation par rapport à des occurrences précédentes.

Il peut avoir différentes valeurs.

- `first` : première occurrence de la référence citée,
- `ibid`, `ibid-with-locator`, `subsequent` : une référence citée précédemment à la position `subsequent`. Elle peut aussi avoir la position `ibid` ou `ibid-with-locator` si les deux occurrences sont contiguës. C'est le cas si l'une des deux conditions suivantes est satisfaite.

1. La référence citée est la même que celle qui la précède immédiatement dans la même note.
2. La référence citée est la première de la note, et elle est la même que celle de la note précédente (qui est une citation unique).

C'est alors la présence de `locator` qui détermine la position attribuée.

La citation précédente n'inclut pas de `locator`.

Si la citation comporte un `locator`, la position est `ibid-with-locator`.

Si elle n'en comporte pas, la position est `ibid`.

La citation précédente inclut un `locator`.

Si la citation comporte le même `locator`, la position est `ibid`.

Si la citation comporte un `locator` différent, la position est `ibid-with-locator`.

Si elle ne comporte pas de `locator`, la position est seulement `subsequent`.

- `near-note` : position d'une citation `subsequent` distante d'un nombre *n* de notes de la première occurrence

de la référence citée. L'attribut `near-note-distance` permet en effet de paramétrer la distance maximale entre les 2 notes. Ainsi, une citation très éloignée de la première occurrence de la référence citée pourra comporter une version plus détaillée de la référence.

Exemple de style note

Style [Presses Universitaires de Rennes \(French\)](#)

```
<citation>
  <layout suffix="." delimiter="&#160;; ">
    <choose>
      <if position="ibid-with-locator">
        <group delimiter=",">
          <text term="ibid" text-case="capitalize-first" font-style="italic" suffix="."/>
          <text variable="locator" prefix="p. ">
        </group>
      </if>
      <else-if position="ibid">
        <text term="ibid" text-case="capitalize-first" font-style="italic"/>
      </else-if>
      <else-if position="subsequent">
        <group delimiter=",">
          <text macro="author"/>
          <choose>
            <if type="bill book graphic legal_case motion_picture report song thesis" match="any">
              <text variable="title" form="short" font-style="italic"/>
            </if>
            <else>
              <text variable="title" text-case="capitalize-first" form="short" quotes="true" font-style="normal"/>
            </else>
          </choose>
          <text term="cited" font-style="italic" suffix="."/>
          <text variable="locator" prefix="p.&#160;;"/>
        </group>
      </else-if>
      <else>
        <group delimiter=",">
          <text macro="author"/>
          <text macro="title"/>
          <text macro="translator"/>
          <text macro="edition"/>
          <text macro="pub-place"/>
          <text macro="publisher"/>
          <text macro="collection"/>
          <text macro="yearpage"/>
        </group>
      </else>
    </choose>
  </layout>
</citation>
```

10. La bibliographie

Structure de l'élément bibliography

Comme l'élément `citation`, l'élément `bibliography` a pour élément enfant les 2 éléments suivants :

- `sort` : définit les critères de tri des références dans la bibliographie,
- `layout` : spécifie les informations à inclure (et leur formatage s'il n'a pas été défini dans des `macro`).

Si certaines options (et donc certains attributs) sont les mêmes que pour `citation` (nombre d'auteurs pour l'affichage de *et al*), d'autres sont spécifiques (marges et saut de ligne, formatage des noms d'auteur déjà cités).

Exercice de style 6

□

Voici l'élément `bibliography` du style *Elsevier-Harvard (with titles)*.

- Pouvez-vous comprendre le paramétrage défini par chacun des 3 attributs de `bibliography` ?
- Quels sont les critères de classement des références dans la bibliographie?
- Quel est le dernier caractère affiché dans une référence bibliographique? Est-il toujours identique? Pourquoi?

```
<bibliography hanging-indent="true" entry-spacing="0" line-spacing="1">
  <sort>
    <key macro="author"/>
    <key macro="issued" sort="descending"/>
  </sort>
  <layout>
    <group suffix=".">
      <text macro="author" suffix=","/>
      <text macro="issued" prefix=" "/>
      <group prefix=".">
        <text macro="title"/>
        <text macro="container"/>
        <text macro="locators"/>
      </group>
    </group>
    <text macro="access" prefix="." "/>
  </layout>
</bibliography>
```

Paramètres spécifiques

Gestion du *et al*

On retrouve les mêmes attributs que pour l'élément `citation`.

- `et-al-min` : nombre d'auteurs minimum pour afficher la mention *et al*.
- `et-al-use-first` : nombre d'auteurs à inclure avant *et al*.
- `et-al-subsequent-min` et `et-al-subsequent-use-first` : idem supra, pour les références déjà citées.
- `et-al-use-last` : remplace *et al* par « ... » et le nom du dernier auteur.
- `delimiter-precodes-et-al` : caractère précédant *et al*.

Espacements

Les attributs suivants sont applicables uniquement à l'élément `bibliography` et non également à l'élément `citation`.

- `hanging-indent` : retrait par rapport à la marge.
- `second-field-align` : `flush` ou `margin` (`margin` : permet de placer les numéros de citation dans la marge).
- `entry-spacing` et `line-spacing` : espacement à l'intérieur de la bibliographie.

Spécification CSL : Whitespace

Grouper par nom d'auteur

Il s'agit, lorsque des noms d'auteur apparaissent dans des références adjacentes, de remplacer les occurrences suivantes du nom par un substitut, défini par la valeur de l'attribut `subsequent-author-substitute`.

L'attribut `subsequent-author-substitute-rule` paramètre les règles de substitution qui s'appliquent.

Spécification CSL : Reference Grouping

Créer une bibliographie par type de document

Créer une `macro`

On va assigner à chaque type de document (ou groupe de types de document) un numéro, correspondant à son numéro d'ordre d'apparition dans la bibliographie.

Dans l'exemple suivant, les livres et sections de livres seront tous cités en premier, puis les articles de revues, puis tous les autres types de documents.

```
<macro name="sort-key">
  <choose>
    <if type="book bookSection" match="any">
      <text value="1"/>
    </if>
    <else-if type="article-journal" match="any">
      <text value="2"/>
    </else-if>
    <else>
      <text value="3"/>
    </else>
  </choose>
</macro>
```

Référencer cette `macro` dans l'élément `bibliography`

Dans l'exemple suivant, dans la bibliographie :

- les références sont regroupées par type de document;
- à l'intérieur de chaque regroupement, elles sont classées par auteur, puis par date, puis par titre.

```
<bibliography et-al-min="11" et-al-use-first="10">
  <sort>
    <key macro="sort-key"/>
    <key macro="author"/>
    <key macro="issued"/>
    <key variable="title"/>
  </sort>
```

11. Miscellanées

Date originale

La date originale d'un document est exprimée par une variable de type `date` , `original-date`.

Ex : `<date variable="original-date" date-part="year"/>`

La date originale doit être saisie dans le champ *Extra* de la notice Zotero, sous la forme : `original-date: 1876` .

Exercice de style date originale

□

Modifiez le style *Elsevier Harvard (with titles)* pour afficher la date originale selon les consignes du style Garni.

Langues

Certains styles sont agnostiques du point de vue la langue, d'autres sont "localisés".

Dans un cas comme dans l'autre, ils utilisent par défaut des traductions communes pour les termes bibliographiques, encodées dans les fichiers `locale` . Exemple : [Fichier locale fr-FR](#)

Dans un cas comme dans l'autre, si les traductions par défaut ne correspondent pas aux consignes du style, il est possible de recourir à des termes autres, en les encodant dans un élément `locale` du fichier CSL (voir *supra* exemple de l'abréviation de numéro pour le style Garni).

Dans un cas comme dans l'autre, limiter au maximum le recours à des `text value=" "` pour définir des chaînes de caractère et aux affixes pour paramétrer la ponctuation assure que son style soit le plus **réutilisable** possible.

Styles génériques

Les styles agnostiques du point de vue de la langue ne comportent pas d'attribut `default-locale` . Ils peuvent être utilisés dans toutes les langues.

Styles localisés

Comme nous l'avons vu, la localisation est encodée dans l'élément `style` , par l'attribut `default-locale` .

```
<style xmlns="http://purl.org/net/xbiblio/csl" class="in-text" version="1.0" demote-non-dropping-particle="never"
default-locale="en-US">
```

Les styles non localisés en `en-` comporte en clair dans leur intitulé la langue dans laquelle ils sont utilisables.

Dans l'entrepôt des styles Zotero on peut retrouver les styles localisés dans une langue en indiquant le nom de la langue dans le formulaire de recherche : *french* affichera ainsi tous les styles ayant un attribut `default-locale="fr-XX"` .

Dans quel cas localiser un style?

Limiter l'utilisation d'un style à une langue déterminée permet d'éviter les erreurs de la part des utilisateurs : quelle que soit la langue par défaut de l'utilisateur, ce dernier ne pourra pas rédiger les éléments bibliographiques dans une autre langue que celle prévue par le style bibliographique. Une revue publiant des

articles dans une unique langue se prémunira ainsi des erreurs de langue en localisant son style bibliographique.

Rechercher un style à partir de caractéristiques précises

L'[entrepôt des styles Zotero](#) vous permet de naviguer dans les styles existants par des facettes portant sur :

- le type de style (numérique, auteur-date, etc.),
- le champ disciplinaire.

Vous pouvez également rechercher un style, mais cette recherche porte uniquement sur l'intitulé du style, qui ne comporte pas toutes ces caractéristiques.

Comment procéder si vous souhaitez retrouver tous les styles qui utilisent la mention *Ibid.*?

Il suffit pour ce faire de lancer une recherche dans l'entrepôt [CSL officiel sur GitHub](#).

□