Introduction à Emacs et Org mode pour les universitaires

Formation Urfist Bordeaux

Frédéric Santos*

4 octobre 2021

Table des matières

*frederic.santos@u-bordeaux.fr

1. Pourquoi utiliser Emacs ?

GNU Emacs est probablement l'un des plus anciens logiciels encore en cours d'utilisation (il a été créé en 1976 !), et sa popularité ne faiblit pas. Outre les programmeurs qui sont son coeur de cible, il dispose d'une communauté d'utilisateurs extrêmement large :

— des chercheurs en sciences humaines ou en littérature ;
— des écrivains (voir notamment `https://www.youtube.com/watch?v=FtieBc3KptU`);
— des lycéens (voir notamment `https://emacsconf.org/2020/talks/26/`);
— des musiciens (voir notamment `https://emacsconf.org/2020/talks/05/`);
— et encore bien d'autres spécialités plus ou moins inattendues (y compris les jardiniers et les culturistes !).

Comment expliquer ce succès depuis plus de 40 ans, et quelles sont les grandes forces d'Emacs ?

1.1. La force du texte brut

Emacs est avant tout un *éditeur de texte brut* [1], à ne pas confondre avec des *traitements de texte* comme LibreOffice Writer ou Microsoft Word. Travailler en texte brut uniquement possède des atouts évidents, en particulier pour le personnel académique.

— Le format texte brut est durable dans le temps : un fichier texte (par exemple, un .txt) s'ouvre aujourd'hui de la même manière qu'il y a 40 ans, et s'ouvrira encore de la même manière dans 40 ans. Par opposition, songez à la durabilité des formats .doc ou .docx ne serait-ce que sur des périodes de 5 ans…
— Le format texte brut est parfaitement portable d'un système d'exploitation à l'autre, et est totalement ouvert (il ne requiert aucun logiciel propriétaire pour être lu).
— Il favorise les bonnes pratiques telles que le versionnement, chose beaucoup moins aisée quand on travaille avec du .docx.

Ces raisons expliquent également pourquoi le texte brut est le format le plus adapté à la pratique de la science ouverte et reproductible (Desquilbet et al., 2019), et pourquoi il est le plus adapté à la rédaction de documents longs, y compris hors du champ scientifique [2].

Paradoxalement, on peut soutenir que les traitements de texte avancés comme Microsoft Word, bien qu'apparus bien plus tard que les éditeurs de texte brut comme Emacs, constituent une façon *moins moderne* de travailler, et en tout cas indiscutablement moins efficace, moins ouverte, moins portable, et beaucoup moins compatible avec les exigences de la science ouverte et reproductible. Il est vraisemblable que l'avenir de la science ouverte soit assez étroitement lié à la généralisation de l'adoption d'habitudes de travail en texte brut, quelle que soit la discipline.

1.2. Une documentation très fournie

GNU Emacs est indiscutablement l'un des logiciels les mieux documentés au monde, avec une documentation complète qui se compte en milliers de pages (entre Emacs lui-même et l'ensemble de ses extensions courantes).

1. Cette définition est d'ailleurs limitative et assez contestée. Il est assez courant d'entendre dire qu'Emacs est plutôt *une machine Lisp disposant de fonctionnalités avancées d'édition de texte*.
2. Un point méconnu : en France, de nombreux rapports parlementaires sont *in fine* mis en forme avec LaTeX.

— Les manuels sont disponibles au téléchargement sur le site officiel d'Emacs (`https://www.gnu.org/software/emacs/documentation.html`) et sont également en vente en version papier.
— Emacs dispose en outre d'un tutoriel intégré pour les nouveaux venus : tapez simplement `C-h t` [3] après ouverture du logiciel.

1.3. Des extensions qui changent la vie

À l'image des navigateurs modernes (Firefox, Opera, ...) qui disposent tous d'une myriade d'extensions (ou *plugins*), GNU Emacs dispose de près de 5200 extensions (au 1er juillet 2021) sur son dépôt officiel MELPA. Bien que le Emacs « de base » soit déjà extrêmement polyvalent et performant, ces extensions ajoutent de nouvelles fonctionnalités ou modifient le comportement par défaut d'Emacs. Parmi les extensions très connues :

— AUCTEX offre un environnement très puissant d'édition de fichiers LaTeX ;
— Emacs Speaks Statistics (ESS) offre aux utilisateurs du langage R un environnement de développement intégré similaire à Rstudio ;
— Magit offre un client Git directement intégré à Emacs ;
— pdf-tools offre une visionneuse PDF performante intégrée dans Emacs, que l'on peut combiner à org-noter pour annoter les PDF dans des fichiers texte séparés ;
— Projectile offre des outils de gestion de projet pour les programmeurs ;
— Org Roam offre un système de gestion de notes conceptuelles et bibliographiques ;
— YASnippet offre un système de *snippets*, i.e. de templates pré-remplis qui peuvent être insérés par des raccourcis clavier au choix (très utile aux programmeurs, mais pas seulement eux, pour des gains de temps considérables).

D'autres extensions ont des objectifs plus légers ; elles offrent par exemple des *thèmes* d'apparence (couleurs, polices, ...), ou même des jeux [4] !

1.4. Un environnement totalement personnalisable

Emacs n'impose aucune vue particulière à ses utilisateurs : au contraire, ils sont encouragés à *hacker* en profondeur le logiciel et à redéfinir le moindre de ses comportements pour obtenir un environnement de travail conforme à leurs préférences. Il n'existe pratiquement aucun comportement d'Emacs qui ne puisse être redéfini par l'utilisateur.

Pour ce faire, il est possible de choisir entre deux approches :

1. Emacs dispose d'un menu de *customisation* (`Options > Customize Emacs`), voir section 4.1. Ce menu permet de redéfinir « à la souris » un grand nombre de comportements et d'en ajouter de nouveaux. Toutefois, les possibilités sont un peu plus limitées si l'on choisit cette voie.
2. L'utilisateur plus avancé choisira généralement d'écrire un *fichier d'initialisation* (voir section 4.2) pour Emacs, c'est-à-dire de fournir *en lignes de code* de nouvelles instructions pour modifier le comportement d'Emacs. Il n'y a alors plus aucune limite aux possibilités de personnalisation du logiciel.

3. Cela signifie « Control-H puis T » ; voir plus loin pour les conventions d'écriture des raccourcis clavier avec Emacs.
4. En particulier, les grands classiques tels que Tetris, Pacman ou le 2048 sont jouables directement dans Emacs.

1.5. Un folklore et une communauté

Selon les proverbes en usage, « *Emacs is the editor of a lifetime* », ou mieux encore, « *Emacs is not an editor, it is a way of life* ».

À l'origine, Emacs et vi étaient les deux grands éditeurs de texte des systèmes Unix, et tout un folklore s'est développé autour de leur opposition (voir l'article *Editor war* sur Wikipedia : `https://en.wikipedia.org/wiki/Editor_war`), chaque « camp » s'amusant à exhiber une détestation surjouée des utilisateurs de l'éditeur rival. En conséquence, les utilisateurs d'Emacs et de vi se sont chacun organisés dans des communautés soudées et folkloriques. Dans le cas d'Emacs, cela a notamment abouti à voir son créateur, Richard Stallman, être intronisé Saint de l'Église d'Emacs (Fig. 1), titre revendiqué lors de nombreuses apparitions publiques.

FIGURE 1 – Richard Stallman apparaissant en *Saint IGNUcius*.

En plus des amusements issus de l'*editor war*, le nouvel entrant dans l'univers d'Emacs pourra profiter du sérieux et de la compétence d'une communauté d'utilisateurs, qu'il pourra notamment retrouver :

— sur le forum francophone Emacs Doctor (`https://emacs-doctor.com/forum/`), qui organise régulièrement des rencontres à Paris, et plus occasionnellement dans d'autres villes françaises ;
— sur Emacs StackExchange (`https://emacs.stackexchange.com/`) ;
— sur Reddit (`https://www.reddit.com/r/emacs/`) ;
— sur Telegram (channels « Emacs » et « Emacs Stories »).

Il existe encore d'autres endroits où demander de l'aide aux *Emacsiens* confirmés, comme sur IRC (Emacs possède d'ailleurs un client IRC intégré), ou encore sur la mailing list Emacs.

2. Installation

Installer Emacs lui-même ne présente aucune difficulté particulière, mais pour illustrer un usage académique, de nombreux autres logiciels seront utilisés au cours de la formation. Des instructions d'installation détaillées peuvent être trouvées sur le dépôt en ligne de la formation : `https://gitlab.com/f-santos/formation-orgmode-2021`

3. Concepts et commandes basiques

Dans cette section, nous essaierons d'exposer quelques notions générales pour débuter dans l'univers Emacs, sans être trop redondant par rapport au tutoriel intégré (que chaque nouveau venu devrait *vraiment* effectuer en entier !) ni aux premières pages du manuel.

3.1. Conventions de notation des raccourcis clavier

Emacs est un éditeur de texte très centré sur l'utilisation du clavier plutôt que de la souris. En particulier, utiliser efficacement Emacs implique de mémoriser un certain nombre de raccourcis clavier, qui peuvent tous être (re)définis à votre convenance.

Historiquement, la documentation d'Emacs utilise certaines conventions pour noter ces raccourcis clavier, héritées de l'époque des machines LISP sur lesquelles Emacs a été créé :

— la touche Ctrl est simplement notée `C` ;
— la touche Alt est notée `M` (pour « Meta » : c'était son équivalent sur les machines LISP) ;
— la touche Maj est notée `S` (pour « Shift ») ;
— la touche Entrée est notée `RET` (pour « Return ») ;
— on utilise le trait d'union `-` pour indiquer une combinaison de touches.

Quelques exemples

— La notation `C-g` veut dire « appuyer simultanément sur Ctrl et G ».
— La notation `M-w` veut dire « appuyer simultanément sur Alt et W ».
— La notation `S-RIGHT` veut dire « appuyer simultanément sur Maj et sur la flèche directionnelle droite ».
— La notation `C-h t` veut dire « appuyer simultanément sur Ctrl et H, relâcher, puis appuyer sur T ».
— La notation `C-x C-b` veut dire « appuyer simultanément sur Ctrl et X, relâcher, puis appuyer simultanément sur Ctrl et B ».

Ces notations seront utilisées dans tout ce qui suit, conformément à l'usage. Pour un aide-mémoire compact, voir par exemple Cameron (2002).

3.2. Tout est fonction

Dans Emacs, *tout est fonction*. Entendons par là que chaque touche activée par l'utilisateur appelle en réalité une *fonction* interne d'Emacs, reliée à cette touche. Par exemple, lorsqu'un utilisateur se trouve dans une fenêtre d'édition de texte (un *buffer*) d'Emacs et appuie sur la flèche directionnelle « ↓ » de son clavier, le curseur d'Emacs descend d'une ligne. En réalité, la flèche directionnelle « ↓ » est reliée par défaut à la fonction interne `forward-line`, qui a pour effet de faire passer le curseur à la ligne suivante.

La plupart des commandes d'Emacs peuvent être appelées ou bien par un raccourci clavier, ou bien explicitement par leur forme *verbeuse*, i.e. à l'aide de leur nom. Pour exécuter « verbeusement » une commande, presser `M-x` puis taper le nom de la commande dans le *minibuffer*, et valider par Entrée.

D'après ce qui précède, il existe donc au moins deux façons de déplacer le curseur vers la ligne suivante avec Emacs :

— ou bien appuyer sur la flèche directionnelle « ↓ » ;
— ou bien faire `M-x forward-line RET`.

Cela correspond simplement aux deux manières possibles (verbeuse, ou par un raccourci clavier) d'appeler la fonction `forward-line`. Cela se généralise à toutes les autres fonctions d'Emacs.

3.3. Un peu de terminologie

On liste ci-dessous quelques termes courants dans la documentation et l'usage d'Emacs.

Mode

À chaque type de fichier correspond un *mode* d'édition Emacs, c'est-à-dire un ensemble de fonctionnalités qui sont activées pour ce type de fichier. Par exemple, le mode `pdfview-mode` est activé pour lire des PDF (et est déclenché dès que l'on ouvre un fichier dont l'extension est `.pdf`), le mode `text-mode` est activé pour lire ou écrire des fichiers texte, etc.

On distingue les *modes majeurs* (un seul peut être activé en même temps), et les *modes mineurs* facultatifs, que l'on peut combiner à volonté avec autant d'autres modes (majeurs ou mineurs) que l'on souhaite.

Buffer

On appelle *buffer* (ou *tampon* en français) une fenêtre d'édition Emacs. Chaque fichier s'ouvre dans un buffer, mais il est possible d'écrire dans un buffer sans qu'il corresponde à un fichier particulier [5].

Il est possible d'avoir beaucoup de buffers ouverts dans une même session Emacs : on peut voir un buffer comme l'équivalent Emacs des *onglets* dans les navigateurs web. D'ailleurs, depuis la version 27 d'Emacs, il est possible de représenter les buffers ouverts comme des onglets, en activant le mode mineur `tab-bar-mode` (cf. Fig. 2).

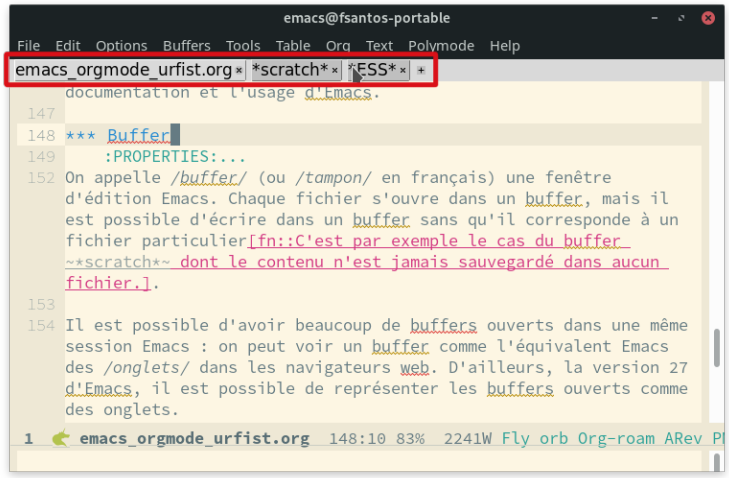

FIGURE 2 – Représentation des buffers Emacs comme des onglets affichés dans une barre supérieure (ici encadrée en rouge).

5. C'est par exemple le cas du buffer `*scratch*`, qui s'ouvre par défaut au démarrage d'Emacs, et dont le contenu n'est lié à aucun fichier.

3.4. Commandes basiques

La table 1 fournit un ensemble très minimal de commandes pour débuter avec Emacs, ainsi que les raccourcis clavier qui leur sont associés par défaut [6].

Opération	Nom de la commande	Raccourci par défaut
Ouvrir ou créer un fichier	`find-file`	`C-x C-f`
Sauver le buffer	`save-buffer`	`C-x C-s`
Couper la région active	`kill-region`	`C-w`
Copier la région active	`copy-region-as-kill`	`M-w`
Coller	`yank`	`C-y`
Se déplacer au début du buffer	`beginning-of-buffer`	`M-<`
Se déplacer à la fin du buffer	`end-of-buffer`	`M->`, i.e. `M-S-<`
Annuler la dernière opération	`undo`	`C-_`
Rechercher (en avant)	`isearch-forward`	`C-s`
Rechercher et remplacer	`query-replace`	`M-%`
Afficher la liste des buffers actifs	`list-buffers`	`C-x C-b`
Changer de buffer	`switch-to-buffer`	`C-x b`

TABLE 1 – Quelques premières commandes utiles.

3.5. Obtenir de l'aide

Emacs dispose d'une aide (technique) interne très détaillée.

— Si vous avez oublié la signification d'une *fonction*, faites le raccourci clavier `C-h f` suivi du nom de la fonction dont vous voulez consulter l'aide. Par exemple, `C-h f insert-file RET` affichera la page d'aide de la fonction `insert-file`.

— Si vous avez oublié à quoi sert exactement un raccourci clavier, taper `C-h k` suivi du raccourci clavier en question. La fonction à laquelle il est rattaché vous sera rappelée.

3.6. Installer de nouveaux packages

Pour accéder à la liste complète des packages Emacs disponibles sur le dépôt officiel Melpa, il suffit de taper `M-x list-packages RET`. Dans le buffer qui s'ouvre, il est possible de naviguer tout aussi bien au clavier qu'à la souris, et d'installer très aisément (à l'aide d'un bouton cliquable) de nouveaux packages.

EXERCICE 1 (Installer et utiliser un nouveau package). — Pour se détendre, installons un petit jeu…

1. Ouvrir la liste des packages Emacs.
2. Naviguer vers le package `2048-game`, et installez-le en cliquant sur le bouton `Install`.
3. Une fois cette opération effectuée, vous pouvez fermer les différentes fenêtres qui se sont ouvertes par un simple appui sur la touche `q`.
4. Tester le nouveau package installé : lancez-le avec `M-x 2048-game RET`. Vous pourrez également quitter le jeu par la touche `q`.

6. Insistons sur le fait qu'ils peuvent être redéfinis suivant votre bon vouloir !

4. Personnaliser Emacs

Emacs a une capacité illimitée de personnalisation. On décrit dans cette section comment le *hacker*.

4.1. Via l'interface graphique

On peut accéder au menu graphique de personnalisation d'Emacs via `Options > Customize Emacs`. Donnons ci-dessous un premier exemple basique de personnalisation : comment supprimer la barre supérieure d'icônes qui est affichée par défaut au démarrage d'Emacs.

EXERCICE 2 (Supprimer la barre d'icônes). — À l'aide du menu `Options > Customize Emacs > All Settings Matching...`, désactiver l'affichage de la barre d'outils d'Emacs. Cette barre d'outils est affichée lorsque le mode mineur `tool-bar-mode` est activé : il suffit donc le désactiver pour la faire disparaître. Après avoir sauvegardé ce réglage, redémarrer Emacs et constater le changement.

De très nombreux paramètres peuvent être redéfinis en suivant la même démarche.

4.2. Via le fichier d'initialisation

Si on essaie de comprendre un peu mieux l'étape ci-dessus : comment Emacs *sait-il* qu'il ne doit désormais plus afficher la barre d'outils au démarrage ? Où est stocké/écrit ce réglage que l'on vient de définir ? Quand et comment est-il lu par Emacs ?

Emacs est programmé pour exécuter au démarrage tout le contenu d'un script d'initialisation, appelé `init.el` ou `.emacs`. Ce script (écrit en langage Emacs Lisp) contient toutes les instructions destinées à modifier son comportement. Par défaut[7], il figure :

— sous Linux, dans le répertoire `~/.emacs.d/` ;

— sous Windows, dans le répertoire `C:/Users/yourusername/AppData/Roaming`.

Lorsque l'on utilise le menu graphique de personnalisation comme dans la section 4.1, Emacs *traduit* nos réglages en lignes de code (en langage Emacs Lisp), et les *écrit* automatiquement pour nous dans le fichier d'initialisation. Ces lignes de code seront ensuite exécutées à chaque démarrage d'Emacs : c'est ainsi qu'Emacs mémorise et prend en compte nos demandes de personnalisation.

Plutôt que de passer par le menu graphique, il est possible (et même recommandé !) d'éditer nous-mêmes, « à la main », ce fichier d'initialisation. Il est donc certes utile de connaître quelques rudiments de langage Emacs Lisp, mais une excellente documentation existe à ce sujet (Chassell, 2009). Toutefois, la plupart du temps, il vous suffira en réalité de copier-coller dans ce fichier des extraits de code Emacs Lisp mis à disposition par la communauté. Beaucoup d'utilisateurs d'Emacs n'écrivent pas eux-mêmes de lignes de code en Lisp dans leur fichier d'initialisation, mais y insèrent des extraits de code trouvés ça et là sur Internet — notamment sur les sites décrits en section 1.5.

Pour cette formation, un exemple de fichier d'initialisation est fourni, afin que chaque participant dispose de la même configuration d'Emacs. Téléchargez et placez dans le dossier adéquat le fichier disponible à l'adresse `https://link.infini.fr/init-file-urfist`.

7. Cela peut néanmoins varier en fonction de la version d'Emacs que vous utilisez. Si vous voulez vous rendre directement à votre fichier d'initialisation, taper `M-: (find-file user-init-file) RET`.

EXERCICE 3 (Éditer votre fichier d'initialisation). — On se propose ici d'éditer directement votre fichier d'initialisation « à la main », afin d'y ajouter un élément très simple de personnalisation.

1. Ouvrez votre fichier d'initialisation, par exemple en faisant `C-x C-f ~/init.el RET` (vous pouvez aussi utiliser le menu `File > Open File` à la souris).
2. Ajoutez l'instruction suivante sur une nouvelle ligne [8] :

```
(setq inhibit-splash-screen t)
```

3. Sauvegardez le fichier, par exemple avec `C-x C-s`, puis redémarrez Emacs. Pouvez-vous trouver ce qu'il y a de nouveau à présent ?

4.3. Utiliser des *starter kits*

Pour ceux qui souhaitent prendre un raccourci dans l'aventure de personnalisation d'Emacs, sachez qu'il existe des configurations toutes prêtes (appelées *starter kits*) un peu partout sur la toile, pour chaque grande catégorie d'usage. On pourra notamment citer :

— Scimax, un starter kit bien adapté aux ingénieurs, programmeurs et chercheurs en sciences dures. Un focus particulier est fait sur l'utilisation de Python dans Emacs.
— Prelude, un starter kit beaucoup plus généraliste, modifiant simplement quantité de comportements par défaut d'Emacs afin de les moderniser et de les rendre plus accessibles.

L'utilisation de *starter kits* peut incontestablement aider les débutants, car le Emacs « *out of the box* » (i.e., sans aucune personnalisation) peut sembler rebutant. Ces *starter kits* permettent de débuter avec des paramétrages plus agréables et une expérience déjà très satisfaisante. Toutefois, il ne faut pas perdre de vue que la grande force d'Emacs est la capacité à être personnalisé sans limite, et à se plier à la moindre de vos préférences ou habitudes de travail. Ces *starter kits* ne le pourront pas : seul un paramétrage par vos soins fera d'Emacs un compagnon vraiment utile pour vous. « L'esprit Emacs » consiste plutôt à considérer ces *starter kits* comme des points de départ à retravailler, et pas comme des produits finis.

5. Utiliser Org mode pour rédiger des documents

5.1. Un couteau suisse pour l'écriture académique

Org mode n'a pas été initialement créé pour l'écriture d'articles de recherche, ni même nécessairement pour la rédaction de rapports académiques longs. Il s'agit pourtant aujourd'hui de deux cas d'utilisation fréquents d'Org mode, dans lesquels il révèle toute sa puissance.

Org mode est particulièrement convenable pour l'écriture académique pour de nombreuses raisons :

1. La légèreté de sa syntaxe permet de se concentrer sur le contenu du document plutôt que de batailler sur des détails de mise en forme.

8. Par exemple à la fin du fichier, mais cela n'a aucune importance : vous pouvez placer cette instruction à n'importe quel endroit.

2. Son moteur d'export universel (en particulier, export PDF via LaTeX) permet d'obtenir aisément des manuscrits mis en forme selon les normes des grands éditeurs scientifiques [9].
3. Son interface de programmation lettrée (Babel) permet d'effectuer toutes les analyses statistiques directement dans le document Org.
4. Ses fonctionnalités de gestion de tâches et de suivi de projet se combinent à toutes les fonctionnalités précédentes, afin d'offrir un moyen simple de planifier l'écriture d'un article, et de suivre précisément son état d'avancement.

Il existe aujourd'hui des interfaces de programmation lettrée (Jupyter, Rmarkdown, ...), des logiciels de gestion de tâches (TaskPaper, ToDoist, ...), et des logiciels de conversion et d'export de fichiers vers d'autres formats (Pandoc), mais Org mode est pour l'heure l'unique solution logicielle à offrir une approche totalement intégrée et unifiée couvrant l'ensemble de ces aspects.

5.2. Rédiger un document au format Org

5.2.1. Métadonnées

Rédiger un fichier Org mode commence généralement par la saisie de quelques *métadonnées* du document : son titre, le nom de son auteur, la date, ainsi que d'autres données plus accessoires (l'adresse électronique de l'auteur, etc.). La saisie de ces données est totalement facultative, mais il est largement préférable de le faire à chaque fois.

Ces métadonnées doivent être indiquées tout en haut du fichier Org, et commencent toutes par l'élément de syntaxe #+ suivi d'un des mots-clés de la table 2.

Mot-clé	Signification
`#+TITLE:`	Titre du document
`#+SUBTITLE:`	Sous-titre (totalement facultatif !)
`#+DATE:`	Date à afficher lors de l'export. Si absent, la date du jour sera utilisée.
`#+EMAIL:`	Adresse électronique de l'auteur
`#+OPTIONS:`	Précise des options pour l'export dans d'autres format (cf. manuel Org)
`#+STARTUP:`	Précise des options pour l'affichage du document dans Emacs

TABLE 2 – Principaux mots-clés pour les métadonnées d'un document Org.

Par exemple, pour le document que nous avons à reproduire, les métadonnées sont les suivantes :

```
#+TITLE: Utiliser GNU Emacs et Org mode à l'université
#+SUBTITLE: Rédiger des documents aisément, s'organiser au quotidien
#+AUTHOR: Frédéric Santos
#+DATE: 4 octobre 2021
#+EMAIL: frederic.santos@u-bordeaux.fr
#+OPTIONS: email:t toc:t
```

Adaptez simplement l'auteur et l'adresse email à votre cas.

9. Tous les grands éditeurs (Elsevier, Plos, Springer/Nature, Wiley, ...) fournissent aujourd'hui des classes LaTeX qu'il suffit d'appliquer lors de l'export du document Org en LaTeX.

5.2.2. Arborescence d'un document

Org mode encourage vivement à structurer son document en sections, même pour les documents courts. Les titres des sections doivent être précédés de une ou plusieurs astérisques, en fonction de leur niveau : une astérisque pour les titres principaux, deux pour les sous-sections, trois pour les sous-sous-sections, etc.

Le travail sur les sections est très facile en Org grâce aux fonctionnalités suivantes :

— créer une nouvelle section : faites simplement `C-RET` ;
— changer le niveau d'une section : placez le curseur sur le titre d'une section, puis faites `M-RIGHT` ou `M-LEFT` pour changer le niveau de la section ;
— *cycle visibility* : placez votre curseur sur un titre, et appuyez sur `TAB` : le contenu de cette section est alors affiché ou masqué (utile pour masquer des sections sur lesquelles on a fini de travailler) ;
— navigation entre sections : placez le curseur sur le titre d'une section, puis faites `C-c C-n` pour naviguer vers la section suivante (*next*), ou `C-c C-p` pour naviguer vers la section précédente (*previous*) ;
— permuter aisément des sections : placez le curseur sur le titre d'une section, puis faites `M-UP` ou `M-DOWN` ;
— *refile subtree* : il est aisé de déplacer tout le contenu d'une sous-section vers une autre section principale, sans avoir le moindre copier-coller à faire. Faites simplement `C-c C-w` sur le titre d'une sous-section.

Un exemple minimal d'arborescence est fourni ci-dessous :

```
* Un titre de niveau 1
Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

** Une sous-section
Donec vitae dolor. Nullam tristique diam non turpis. Cras placerat accumsan nulla.

** Une autre sous-section
*** Une section de niveau 3
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec hendrerit tempor
tellus.

*** Une autre section de niveau 3
Donec pretium posuere tellus.  Proin quam nisl, tincidunt et, mattis eget,
convallis nec, purus.
```

5.2.3. Listes

Dans un document Org, les listes à puces s'indiquent simplement par des tirets initiaux :

```
- Un premier item
- Un second item
- Un troisième item
```

En particulier :

— il suffit de faire `M-RET` pour passer automatiquement à l'item suivant ;
— faire `S-RIGHT` sur n'importe quel endroit de la liste permet d'en changer le type (liste à puces vs. liste numérotée).

5.2.4. Liens

La façon la plus simple d'insérer un hyperlien dans un document Org est d'utiliser l'interface de saisie offerte via le raccourci clavier `C-c C-l` (ce raccourci clavier est rattaché par défaut à la fonction `org-insert-link` : il est donc équivalent à l'instruction `M-x org-insert-link RET`). Emacs demandera alors d'écrire successivement la cible (i.e., l'adresse) du lien, puis sa description (i.e. le texte cliquable qui s'affichera).

Il reste possible de les saisir directement à la main, avec une syntaxe proche de celle du langage Markdown :

```
[[http://un-exemple-de-lien.com][Le texte de description]]
```

5.2.5. Tables

Org mode permet de mettre en forme rapidement et efficacement des tables, ainsi que de les numéroter automatiquement lors de l'export du document. En particulier, la saisie d'une table Org est incomparablement plus simple que de coder un tableau en HTML ou en LaTeX, et est également plus commode qu'en Markdown. Outre les fonctionnalités d'édition de tables « statiques », Org mode propose également quelques outils équivalents au concept de *formules* dans les tableurs (Excel, LibreOffice Calc), permettant ainsi de créer des tables au contenu dynamique. Cela ne sera néanmoins pas couvert dans ce document : le lecteur pourra se reporter directement au manuel d'Org mode à ce sujet.

Un exemple. Voici comment est saisie la table 3 en syntaxe Org :

```
#+NAME: tab-villes
#+CAPTION: Quelques villes françaises.
|------------+------------+--------------------|
| Ville      | Population | Région             |
|------------+------------+--------------------|
| Angers     |    155.000 | Pays de la Loire   |
| Poitiers   |     90.000 | Nouvelle-Aquitaine |
| Strasbourg |    285.000 | Alsace             |
|------------+------------+--------------------|
```

Ville	Population	Région
Angers	155.000	Pays de la Loire
Poitiers	90.000	Nouvelle-Aquitaine
Strasbourg	285.000	Alsace

TABLE 3 – Quelques villes françaises.

Principes d'édition

— Les colonnes d'une table sont séparées par le caractère `|`

— Pour démarrer la saisie d'une table, saisissez simplement tout le contenu de la première ligne en séparant les colonnes par `|`, et par la suite, un simple appui sur `TAB` prépare la saisie de toute la ligne suivante et ré-aligne le contenu des cellules.

— Les lignes horizontales de séparation s'ajoutent par le raccourci `C-c -`, ou en insérant `|-` au début d'une ligne puis en appuyant sur `TAB`.
— Les lignes peuvent être permutées en utilisant le raccourci `M-UP` ou `M-DOWN`.
— Les colonnes peuvent être permutées en utilisant le raccourci `M-RIGHT` ou `M-LEFT`.

Il est à noter que, pour l'heure, il n'est pas possible d'utiliser le concept de *cellules fusionnées* via la syntaxe Org. (Il ne s'agit de toute façon pas, en général, d'une excellente pratique.)

Légende et label. L'item `#+CAPTION:` gère l'affichage de la légende lors de l'export, et l'item `#+NAME` permet d'attribuer un label arbitraire à la table, qui servira plus tard à l'appeler au sein du document via une référence croisée (voir section 5.2.7). Ces deux items sont très utiles, mais facultatifs.

5.2.6. Figures

Pour insérer une figure (par exemple au format jpeg ou png) dans un document Org, il suffit d'insérer l'élément de syntaxe suivant à l'endroit du document où vous souhaitez la voir apparaître :

```
[[./chemin/vers/la/figure.png]]
```

Il suffit donc d'indiquer le chemin (absolu ou relatif) de la figure entre une paire de doubles crochets. Mieux encore, l'ajout des éléments suivants (facultatifs) permet de gérer l'affichage d'une légende ainsi que d'effectuer plus loin un appel à cette figure [10] :

```
#+NAME: ma-figure
#+CAPTION: Texte de légende
[[./chemin/vers/la/figure.png]]
```

Une remarque importante : par défaut, une fois ces éléments de syntaxe saisis, Org peut les interpréter en affichant directement dans le document Org l'image cible. Il s'agit d'un réglage que l'utilisateur peut modifier de trois manières :
— ou bien globalement via le fichier d'initialisation, en donnant la valeur `t` à la variable `org-startup-with-inline-images` [11] ;
— ou bien localement dans chaque document Org, en ajoutant dans l'entête du document la ligne `#+STARTUP: inlineimages` ;
— ou bien au cas par cas : effectuez le raccourci clavier `C-c C-x C-v` pour visualiser directement toutes les images appelées dans le document Org (ce raccourci est lié à la fonction `org-toggle-inline-images`). Vous pouvez refaire ce raccourci une seconde fois pour cacher toutes les images.

5.2.7. Références croisées

Org mode gère pour nous, lors de l'export du document, la numérotation automatique des tables et des figures. Comme nous l'avons vu en sections 5.2.5 et 5.2.6, il est possible d'attribuer un label (i.e., une étiquette) à chaque table ou figure, de façon à pouvoir ensuite

10. Il s'agit donc d'une syntaxe totalement identique à celle des tables sur ce point.
11. En clair : il faudra ajouter l'instruction `(setq org-startup-with-inline-images t)` dans votre fichier d'initialisation.

y faire appel au sein du document. Ce nom est à indiquer au niveau de l'attribut `#+NAME:` placé avant la table ou la figure en question.

Par exemple, la table 3 avait reçu l'étiquette `tab-villes`, via l'attribut `#+NAME: tab-villes`. Pour l'appeler, il suffit d'indiquer son nom entre doubles crochets, comme ci-dessous :

```
La population municipale de quelques villes françaises est fournie en table [[tab-villes]].
```

Ce principe se généralise aux figures, ainsi qu'aux sections d'un document.

Remarque. — Le package org-ref de John Kitchin offre quelques fonctionnalités additionnelles (et très commodes) pour l'étiquetage des figures, tables et sections, ainsi que pour effectuer des références croisées. La syntaxe utilisée est légèrement différente de la syntaxe Org de base présentée ici, mais (pour l'heure) ne fonctionne que pour l'export PDF via LaTeX. Ainsi, les références créées avec org-ref ne fonctionneront pas pour l'export HTML ou ODT. Ceci rend org-ref surtout convenable pour l'écriture d'articles de recherche, l'export PDF étant la voie standard dans ce cas de figure.

5.3. Export (HTML, ODT, PDF, . . .)

Pour diffuser un document Org (et en obtenir un bon rendu esthétique), la voie standard est *l'export* vers un autre format plus riche. Ceci inclut notamment :

— l'export en PDF via LaTeX ;
— l'export en page web HTML ;
— l'export au format OpenDocument (.odt), utilisé par LibreOffice et OpenOffice ;
— l'export sous forme de diaporama, ou bien au format beamer (via LaTeX là encore), ou bien au format reveal.js (diaporamas HTML [12]).

Pour exporter un document Org :

1. Effectuer le raccourci clavier `C-c C-e`.
2. Dans la liste des formats possibles, sélectionner le format désiré. Par exemple, l'instruction complète pour exporter au format PDF/LaTeX sera `C-c C-e l p` ; tandis que l'instruction complète pour exporter au format HTML sera `C-c C-e h h`.

Au cas par cas, la mise en forme de chaque document exporté peut être personnalisée : très efficacement pour l'export HTML (via une feuille de style CSS par exemple) ou PDF ; de manière plus limitée pour l'export au format OpenDocument.

5.4. Références bibliographiques

5.4.1. Avertissement

À ce jour (octobre 2021, ce qui correspond à la version 9.4.6 d'Org mode), l'insertion, la gestion et l'export de références bibliographiques dans un document n'est *pas* une fonctionnalité de base d'Org mode. Un package spécifique, org-ref, doit être utilisé pour cela. À compter de la version 9.5 d'Org mode, à paraître très prochainement, Org mode offrira nativement des fonctionnalités de gestion bibliographique. Une partie de ce qui est exposé dans cette section sera donc remplacé par un système différent et plus efficace d'ici quelques semaines.

12. Ce dernier format n'est pas nativement supporté par Org, mais est offert par le package ox-reveal.

Actuellement, le package org-ref offre d'excellentes fonctionnalités de gestion bibliographique *lorsque la visée finale est d'exporter le document en PDF*. Org mode, dans ce cas, utilise LaTeX et BibTeX pour produire le document final. En revanche, l'export de références bibliographiques fonctionne encore assez mal pour les formats html et odt [13]. Cette limitation sera corrigée lors de la sortie de la version 9.5 d'Org mode.

5.4.2. Fonctionnement (via org-ref)

1. La première étape est de disposer d'un fichier de références bibliographiques *au format .bib*, i.e. au format BibTeX. Notez que certains logiciels (dont Zotero) peuvent exporter directement une base de données bibliographique dans ce format : vous n'aurez donc pas à le créer à la main.
2. Il sera ensuite nécessaire d'indiquer, de préférence à la fin de votre document, le chemin absolu de ce fichier BibTeX sur votre disque dur. Pour ce faire, on utilise le mot clé `bibliography:`. Par exemple (sous Windows), si votre fichier s'appelle `biblio.bib` et se trouve à la racine du répertoire `C:/Documents/`, il sera nécessaire d'ajouter la ligne suivante à la fin de votre fichier :

   ```
   bibliography:C:/Documents/biblio.bib
   ```
3. Il sera ensuite nécessaire de choisir un style bibliographique particulier, en l'indiquant à l'aide du mot-clé `bibliographystyle`. Ces styles bibliographiques sont ceux proposés par BibTeX ; on pourra donc par exemple spécifier :

   ```
   bibliographystyle: plain
   ```

   pour des références simplement numérotées (i.e., qui ne seront pas du genre auteur-année).
4. Pour insérer une citation dans le document, il suffit de placer son curseur à l'endroit désiré et d'effectuer le raccourci clavier `C-c ]`. La liste de toutes les références s'affiche alors, et on peut sélectionner la citation souhaitée avec la touche Entrée. Un lien de citation, de la forme `cite:clédecitation` est alors inséré dans le document.
5. Pour terminer, exporter le document *en PDF* avec `C-c C-e l o` : c'est tout !

5.5. Pile logicielle pour rédiger des articles scientifiques

En résumé, outre Emacs et Org mode eux-mêmes, on peut conseiller la pile logicielle suivante pour être capable de rédiger à 100% en Org mode un article de recherche reproductible.

— Zotero [14] permettra la collecte aisée de références bibliographiques via son intégration à votre navigateur web.

— Le plug-in BetterBibTeX de Zotero facilitera l'export de votre base bibliographique en un fichier BibTeX, dont les clés de citation auront un format harmonisé. Il est alors

13. La démarche standard pour exporter correctement en odt un document org contenant des références bibliographiques serait d'exporter d'abord le document org en document LaTeX, puis d'utiliser le logiciel tiers Pandoc pour convertir le document LaTeX en document odt. Ceci est assez fastidieux, mais ne sera donc plus nécessaire dans un avenir proche.

14. On pourrait tout aussi bien conseiller JabRef, ou encore d'autres logiciels équivalents, mais Zotero offre d'excellentes extensions qui facilitent grandement le *workflow* général détaillé ici.

conseillé de donner l'adresse de ce fichier aux variables `org-ref-default-bibliography` et `bibtex-completion-bibliography` dans Emacs [15].

— Le plug-in ZotFile de Zotero facilitera la gestion des pièces jointes (au format PDF) des références de votre base bibliographique, grâce à leur renommage selon une nomenclature harmonisée et leur stockage dans un dossier commun [16]. Il est alors conseillé de donner le chemin complet de ce dossier aux variables `org-ref-pdf-directory` et `bibtex-completion-library-path` dans Emacs.
— Le package org-ref d'Emacs facilitera la gestion et l'insertion de références bibliographiques.
— Le package org-noter d'Emacs permettra quant à lui l'annotation des PDF des articles grâce à des notes saisies au format Org.
— (Facultatif) Un langage de programmation adapté à l'analyse de données (Julia, Python, R, ...) permettra d'effectuer vos analyses statistiques directement dans votre manuscrit Org mode et d'obtenir ainsi un article totalement transparent, et plus aisément reproductible.
— Si nécessaire, le logiciels tiers Pandoc (ainsi, peut-être, que le package Emacs ox-pandoc) permettra d'exporter vos documents dans encore plus de formats que ceux initialement proposés par Org mode.

6. Utiliser Org mode comme planificateur du quotidien

Dans ce document, Org mode a jusqu'ici été surtout décrit et utilisé comme un langage de balisage léger similaire à Markdown, disposant de son propre moteur d'export. Toutefois, Org mode était originellement surtout un outil d'organisation au quotidien (d'où le « org » de « Org mode »), de gestion d'agenda, de planification et de suivi de tâches (Dominik, 2010). Grâce à ces fonctionnalités, Org mode est un des outils permettant d'utiliser au quotidien les principes GTD (*Getting Things Done*) de David Allen (2003).

Dans cette section, nous aborderons deux approches complémentaires :

1. Comment créer et maintenir un fichier d'agenda global (ou « principal ») en Org mode afin d'y centraliser toutes vos tâches et d'en suivre la progression.
2. Comment suivre spécifiquement l'avancée d'un projet correspondant à un fichier Org donné.

6.1. Prologue : instructions pour le fichier d'initialisation

Comme précédemment, toutes les instructions spécifiques permettant de plier Emacs selon vos besoins devront être indiquées dans le fichier d'initialisation (ou bien via l'interface graphique, ou bien directement en lignes de code). Tout au long de cette section, nous ajouterons petit à petit de nouvelles instructions dans ce fichier, ou nous commenterons les instructions déjà existantes dans le fichier d'initialisation qui vous est proposé lors de cette formation.

Toutefois, une instruction est à ajouter par vous-même dès à présent.

15. Autrement dit, si l'adresse de ce fichier est `~/Documents/biblio.bib`, ajouter l'instruction `(setq bibtex-completion-bibliography "~/Documents/biblio.bib")` dans votre fichier d'initialisation.

16. Par défaut, Zotero stocke les pièces jointes des références dans une arborescence assez obscure, avec un dossier par référence, ce qui rend très délicate leur exploitation via Emacs, Org mode et Org Roam. ZotFile est donc un ajout indispensable.

EXERCICE 4. — Nous allons ici créer et configurer votre agenda global.

1. Vous trouverez l'instruction suivante en ligne 6 du fichier d'initialisation de la formation :

   ```
   (setq my-todo-file "/path/to/todo.org")          ; PERSONNALISER ICI
   ```

   Dans cette instruction, remplacez l'élément `"/path/to/todo.org"` par l'adresse où vous souhaitez avoir votre fichier d'agenda global sur votre ordinateur. Attention : ce fichier devra donc impérativement demeurer à cet endroit une fois cette variable définie [17]. Emacs saura désormais que votre fichier d'agenda global se situe à cet emplacement, et pourra y ajouter ou en extraire des tâches automatiquement.
2. Sauvegardez votre fichier d'initialisation (`C-x C-s`).
3. Il reste encore à créer votre fichier d'agenda global : pour l'instant, Emacs en connaît l'adresse, mais ce fichier n'existe pas ! Créez donc un fichier d'agenda à l'emplacement que vous avez défini (`C-x C-f`, ou menu `File > Visit New File`). Ajoutez-lui quelques métadonnées sympathiques selon la syntaxe Org (e.g., un titre et un auteur), puis sauvegardez-le (`C-x C-s`, ou menu `File > Save`). Laissez vide le reste du fichier pour l'instant.
4. Fermez et relancez Emacs pour que les modifications soient prises en compte, puis rouvrez votre fichier d'agenda pour ce qui suit.

6.2. Tâches

Tout le système de planification du quotidien d'Org mode repose sur la définition de *tâches*. Une tâche est simplement *un titre de section associé à un statut*, i.e. un intitulé précédé de une ou plusieurs astérisques `*` ainsi que d'un mot-clé donnant le statut de la tâche en cours.

6.2.1. Statuts des tâches

Les statuts possibles pour une tâche sont stockés dans la variable `org-todo-keywords`. Par défaut, Org mode ne connaît que deux statuts possibles pour une tâche (i.e., la variable `org-todo-keywords` ne contient que deux valeurs) :

— `TODO` : la tâche est à accomplir ;
— `DONE` : la tâche a été accomplie.

Ainsi, voici deux exemples de tâches en syntaxe Org [18] :

```
* TODO Corriger les copies des Master 1
* DONE Remplir dossier d'évaluation
```

Les deux statuts `TODO / DONE` étant un peu limitatifs en pratique, de nouveaux statuts sont définis dans le fichier d'initialisation de la formation :

17. Plus exactement, il ne s'agit pas ici d'une variable « officielle » Emacs, mais d'une variable que je répercute à différents autres endroits du fichier d'initialisation. La « vraie » variable Emacs indiquant l'emplacement du fichier d'agenda est `org-agenda-files`.
18. Rappelons ici qu'une tâche est donc *nécessairement* un titre de section (ou sous-section, etc.), suivi d'un mot-clé indiquant son statut actuel.

— `IN-PROGRESS` : la tâche a été débutée mais n'est pas totalement finie (par opposition à `TODO` qui désignera désormais les tâches dont l'exécution n'est pas *du tout* commencée) ;
— `WAITING` : la tâche est actuellement bloquée pour une raison qui ne dépend pas de nous (e.g., attente du retour d'un collaborateur) ;
— `SOMEDAY` : la tâche sera peut-être à effectuer un jour, mais la nécessité de son exécution n'est pas claire à ce stade.

Bien sûr, le nom et le sens donné à ces nouveaux mots-clés n'a rien de standard : n'hésitez pas à en changer le nom et/ou à en définir d'autres plus adaptés à votre quotidien et vos pratiques. Il suffira pour cela de modifier les deux variables `org-todo-keywords` et `org-todo-keywords-faces` dans le fichier d'initialisation.

EXERCICE 5 (Créer et mettre à jour manuellement vos premières tâches). — Ouvrez votre fichier d'agenda global (`C-x C-f`).

1. Dans le corps du fichier, ajoutez une tâche « Révision de mon article », et attribuez-lui initialement le statut `TODO`.
2. Notez que changer de statut est aisé : positionnez le curseur sur la tâche et faites `M-RIGHT`, puis `M-LEFT`. Que notez-vous ?
3. Mieux encore : toujours en étant positionné sur la tâche, faites `C-c C-t`, et passez la tâche au statut `DONE`.

6.2.2. Division en sous-tâches

Dans la philosophie *Getting Things Done* (Allen, 2003), une tâche est un énoncé général, qui doit si possible être découpé en sous-tâches plus précises et directement *activables*. Org mode offre cette possibilité, que l'on va illustrer sur l'exemple suivant :

```
* TODO Préparer ma formation [/]
- [ ] Installer les logiciels requis
- [ ] Imprimer la documentation
```

EXERCICE 6 (Découpage en sous-tâches). — Les opérations suivantes sont à réaliser dans votre fichier d'agenda global.

1. Saisir manuellement la tâche « Préparer ma formation ». L'élément de syntaxe « `[/]` » doit être ajouté manuellement après l'intitulé de la tâche, et servira de compteur (automatiquement actualisé) pour les sous-taches réalisées par la suite [19].
2. Retourner à la ligne, puis ajouter une sous-tâche débutant par l'élément de syntaxe « `- [ ]` », symbolisant une case à cocher (*checkbox*).
3. Pour créer plus facilement de nouvelles sous-tâches, vous pouvez utiliser le raccourci clavier `M-S-RET` plutôt que de saisir « `- [ ]` » à la main. Utilisez ce raccourci pour créer la seconde sous-tâche.
4. Lorsque l'on veut indiquer qu'une sous-tâche a été effectuée, il suffit de placer le curseur sur cette sous-tâche et d'utiliser le raccourci clavier `C-c C-c`. Noter alors l'actualisation automatique du compteur de sous-tâches !

19. On peut aussi utiliser l'élément de syntaxe « `[%]` » si vous préférez un compteur en pourcentage d'avancement.

5. Que se passe-t-il lorsque vous avez coché (i.e., effectué) toutes les sous-tâches [20] ?

6.2.3. Planification

Définir des tâches n'a pas nécessairement en soi un grand intérêt si cela ne vous aide pas à agir concrètement au quotidien. Il est donc généralement utile de leur associer une date à laquelle vous commencerez le travail (*schedule*) et/ou une date limite (*deadline*). Nous verrons plus loin comment obtenir une visualisation synthétique de toutes ces dates afin d'obtenir un planning de travail élégant (cf. section 6.3).

EXERCICE 7 (Attribuer des dates à vos tâches). — Les opérations suivantes sont à réaliser dans votre fichier d'agenda global.

1. Revenir sur la tâche « Préparer ma formation » précédemment créée, et décocher toutes les sous-tâches éventuellement cochées. Positionner le curseur sur cette tâche, puis effectuer le raccourci clavier `C-c C-d`. Dans le calendrier qui s'ouvre, choisir la date du 6 octobre 2021, et valider par `RET`. Que constatez-vous ?
2. Ajouter une nouvelle tâche « `* TODO RDV chez le dentiste` ». Positionner le curseur sur cette tâche, puis effectuer le raccourci clavier `C-c C-s`. Dans le calendrier qui s'ouvre, choisir la date du 5 octobre 2021, et valider par `RET`. Que constatez-vous ?

6.2.4. Tags

Dans votre fichier d'agenda global, il est possible que vous souhaitiez mélanger des tâches personnelles et des tâches professionnelles. Un système de *tags* vous permet par exemple de différencier ces deux types d'environnements. Par la suite, vous pourrez ainsi filtrer aisément les tâches personnelles uniquement, ou les tâches professionnelles uniquement.

EXERCICE 8 (Attribuer un tag à une tâche). — Dans votre fichier d'agenda global, revenir sur la tâche « RDV chez le dentiste » précédemment définie. Placer le curseur sur cette tâche, puis effectuer le raccourci `C-c C-q`. Attribuer le tag « `perso` », puis valider par `RET`.

6.2.5. Priorités

Certaines tâches peuvent être plus urgentes ou plus importantes que d'autres : il est alors important de pouvoir les identifier rapidement parmi l'océan de tâches que constituera peut-être votre fichier d'agenda global. Par défaut, Org offre trois degrés de priorités, classés de `A` (tâche urgente) à `C` (tâche très secondaire) [21].

EXERCICE 9 (Attribuer une priorité à une tâche). — Dans votre fichier d'agenda global, revenir sur la tâche « Préparer ma formation » précédemment définie. Placer le curseur sur cette tâche, puis effectuer plusieurs fois le raccourci clavier `S-UP`. Que constatez-vous ?

20. Ce que vous observerez n'est pas le comportement par défaut d'Org mode, mais est une personnalisation ajoutée dans le fichier d'initialisation de la formation.

21. Il est possible de définir des degrés plus fins de priorité : on pourra notamment personnaliser les variables `org-highest-priority` et `org-lowest-priority`.

6.2.6. Notes rapides

Lorsqu'une tâche est complexe, par exemple divisée en plusieurs sous-tâches, et qu'elle est amenée à durer dans le temps, il peut être utile d'y associer des notes au fur et à mesure de son avancement. Org mode offre un système de notes rapides (*quicknotes*), permettant d'écrire des notes horodatées qui seront conservées dans un « tiroir » (*drawer*) sous l'intitulé de la note en question [22], que l'on peut déplier ou replier avec la touche `TAB`.

EXERCICE 10 (Prendre des notes sur une tâche). — Dans votre fichier d'agenda global, revenir sur la tâche « Révision de mon article » précédemment définie. Placer le curseur sur cette tâche, puis effectuer un `C-c C-t w` pour la passer en statut `WAITING`. Que constatez-vous ?

Noter qu'il est possible de déclencher automatiquement la saisie de notes rapides dans certains cas. Par exemple, passer une tâche en statut `WAITING` suppose qu'il se passe quelque chose de particulier dans le déroulement de cette tâche : attendez-vous le retour d'un collègue, attendez-vous d'avoir fini une expérience au préalable, autre chose ? Il peut sembler logique de saisir une note rapide *à chaque fois* qu'une tâche passe en attente, pour expliquer *pourquoi* vous ne pouvez plus avancer. Ceci s'indique au moment de la définition des différents statuts (`TODO`, `WAITING`, `...`), en spécifciant lesquels doivent déclencher automatiquement une note rapide : consulter le manuel d'Org mode pour plus d'information à ce sujet.

EXERCICE 11 (Notes rapides déclenchées automatiquement). — Dans votre fichier d'agenda global, revenir sur la tâche « Préparer ma formation » précédemment définie. Placer le curseur sur cette tâche, puis effectuer un `C-c C-z`. Entrer une note dans le buffer qui s'ouvre à l'occasion, et valider par `C-c C-c`. Que constatez-vous ?

6.2.7. Que faire des tâches clôturées ?

Au fur et à mesure que vous terminez vos tâches, de plus en plus d'items de votre fichier d'agenda global recevront le statut `DONE`. Pour autant, ils vont demeurer dans votre fichier d'agenda : que faut-il alors en faire ?

— Si le fait de conserver une trace de ces tâches passées présente un intérêt (par exemple parce qu'on y avait saisi beaucoup de notes rapides que l'on souhaite conserver), il existe alors dans Org mode un système d'*archivage* des tâches, consistant à déplacer les tâches terminées dans un fichier d'archive afin de ne pas encombrer inutilement le fichier d'agenda principal. Ce système ne sera pas abordé dans ce document.

— Si le fait de conserver une trace d'une tâche donnée ne semble pas pertinent (e.g., il est douteux que créer une archive pour la tâche « RDV chez le dentiste » présente un intérêt), le plus simple est alors de la supprimer manuellement du fichier d'agenda.

6.2.8. Pour aller plus loin

Org mode possède d'autres fonctionnalités de planification plus avancées, qui ne seront pas abordées dans cette formation d'initiation. On peut néanmoins en énumérer quelques unes :

22. Ce n'est pas une fonctionnalité uniquement reliée aux tâches, mais à toute section Org. Le contenu du *drawer* n'est pas exporté lors de l'export PDF (ou html, etc.), ce qui en fait des notes totalement « privées ».

— Org mode offre la possibilité de définir des *tâches récurrentes*, qui reviendront à intervalles réguliers (e.g., quotidiennement, hebdomadairement, mensuellement), dans votre système. Voir par exemple le tutoriel de Rainer König à ce sujet (`https://www.youtube.com/watch?v=nbC-gL5wcf4`).
— Il est également possible de définir des tâches récurrentes comme étant des *habitudes*. Org mode en suit alors précisément l'exécution, gardant en mémoire la date à laquelle vous les accomplissez, le temps total que vous y consacrez, etc. Ceci est particulièrement utile pour faire un retour critique sur vos habitudes de travail. Là encore, la chaîne Youtube de Rainer König en offre une explication très claire (`https://www.youtube.com/watch?v=acj3NhXlnnk`).

6.3. Vue agenda

Pour l'heure, nous avons juste créé des tâches « en vrac » dans un fichier d'agenda global. Cela ne peut pas réellement être d'une grande aide au quotidien : nous avons besoin d'un meilleur moyen d'extraire et de visualiser les tâches pertinentes dans cette masse d'informations. Quelles sont les tâches à effectuer aujourd'hui ? Quelles sont les tâches urgentes, ou les tâches actuellement bloquées et en attente ? Les différentes *vues agenda* d'Org mode vous permettront de répondre aisément à ces questions et de planifier votre journée.

Certaines de ces vues, assez basiques, sont intégrées par défaut dans Org mode. Il est néanmoins possible de construire soi-même de meilleures vues agenda, affichant les informations dont vous avez réellement besoin pour vous organiser au quotidien.

6.3.1. Vue par défaut

En vous situant *n'importe où* dans Emacs (il n'est absolument pas nécessaire d'avoir actuellement le fichier d'agenda ouvert : c'est un point capital !), effectuer le raccourci clavier `C-c a` vous ouvre un *menu agenda*. Presser encore `a` vous ouvrira la vue agenda par défaut, qui extrait toutes les tâches courantes de votre fichier d'agenda pour la semaine en cours ou le jour en cours. Pour basculer entre la vue « semaine » et la vue « jour », presser simplement les touches `d` (pour *day*, qui ouvre la vue du jour) ou `w` (pour *week*, qui ouvre la vue de la semaine).

Concrètement, Org mode parcourt le fichier d'agenda qui lui a été indiqué, recherche toutes les tâches ayant une date de début (schedule) ou une date limite (deadline), et les affiche dans la vue quotidienne ou hebdomadaire. Le raccourci `q` (*quit*) permet de refermer la vue agenda.

6.3.2. Filtrage de tâches

Au lieu de la vue globale, il est également possible d'extraire uniquement les tâches répondant à des critères particuliers : par exemple, toutes les tâches ayant un tag précis, ou toutes les tâches ayant un certain statut, etc. Org mode effectue alors une requête sur le fichier d'agenda, et affiche les tâches correspondantes.

EXERCICE 12 (Filtrer des tâches selon un tag). — On avait précédemment défini une tâche de type « privé » (personnelle) dans notre fichier d'agenda global, en lui donnant le tag `:perso:`. Vérifiez que le raccourci clavier `C-c a m perso RET` permet d'afficher les tâches perso figurant dans votre fichier d'agenda.

6.3.3. Vues personnalisées

Des vues répondant spécialement à vos besoins peuvent aussi être définies, afin de n'extraire et de n'afficher que les informations qui vont sont véritablement utiles. Le package org-super-agenda permet même de les classer en différentes catégories beaucoup plus lisibles. À titre d'exemple, le fichier d'initialisation de la formation propose une vue personnalisée affichant :

— les tâches du jour (schedule et/ou deadline fixées aujourd'hui) ;
— les tâches urgentes (priorité `[#A]`) ;
— les tâches bloquées / en attente (`WAITING`).

Vous pouvez y accéder par le raccourci clavier `C-c a d`. La documentation d'Org mode et celle de org-super-agenda montrent comment définir ces vues personnalisées.

6.4. Utiliser le système de capture

Jusqu'ici, nous avons saisi nos tâches en les éditant manuellement dans le fichier d'agenda global. Il s'agit là d'une pratique vraiment sous-optimale, que l'on évitera autant que possible. La plupart du temps, les tâches seront ajoutées dans ce fichier d'agenda à l'aide d'un système de *capture templates* personnalisables.

6.4.1. Intérêt

Le principe de *capture* est un élément fondamental de la méthode Getting Things Done (Allen, 2003). Disposer d'un outil de capture consiste à pouvoir noter très rapidement toute information, sollicitation ou tâche qui se présente (à la suite d'un coup de téléphone reçu, d'un mail annonçant un événement, d'une visite impromptue d'un collègue, ou tout simplement d'une idée qui nous traverse la tête). En particulier :

— cela doit pouvoir se faire *avec une interruption nulle ou négligeable de notre travail en cours* ;
— ces notes doivent être brèves ; elles peuvent être augmentées et détaillées dans un second temps, dans une phase dite de clarification.

6.4.2. Pratique avec Org mode

EXERCICE 13 (Capturer rapidement un rendez-vous). — Nous allons ici montrer comment ajouter beaucoup plus aisément des tâches dans votre agenda.

1. Placez-vous dans le buffer `*scratch*`. (Rappel : cela peut se faire ou bien par le raccourci clavier `C-x C-b`, ou bien à la souris par le menu `Buffers`.)
2. Vous êtes en train de travailler dans ce buffer et d'y écrire du texte lorsqu'un collègue vient vous demander un rendez-vous. Sans interrompre votre travail ni changer de buffer, effectuez le raccourci `C-c c` pour ouvrir le menu de *capture*, puis pressez `r` pour choisir l'option « Rendez-vous ».
3. Dans le buffer qui s'ouvre, ajoutez une date (mettons, le 4 octobre à 18 heures) ainsi qu'une description pour votre rendez-vous, puis validez par `C-c C-c`.
4. Vous voilà de retour dans le buffer `*scratch*`. À présent, rouvrez votre vue agenda avec `C-c a d` : que constatez-vous ?

5. Pour comprendre, retournez dans votre fichier d'agenda global. Quel élément y a été ajouté ?

Voilà essentiellement comment vous pourrez ajouter différents nouveaux événements (rendez-vous, réunions, projets d'articles, tâches scientifiques, ...) très aisément dans votre agenda, sans jamais sortir significativement du travail que vous étiez en train d'effectuer. Plus généralement, des *capture templates* peuvent être définis pour des catégories plus variées : des idées à développer plus tard, des articles que vous aimeriez relire plus tard, etc. Des packages permettent même de relier votre fichier d'agenda à votre navigateur web afin d'ajouter automatiquement une entrée de type `TODO` vous incitant à relire plus tard cette page. Par ce genre d'astuces, vous tirerez pleinement profit du système de capture d'Org mode.

6.5. Fonctionnalités « en local » dans un fichier Org

Au quotidien, c'est précisément la *combinaison* des deux aspects de Org (l'aspect planification de tâches, et l'aspect langage de balisage léger) qui le rend extrêmement utile dans le milieu académique, et en particulier dans le cadre de la science ouverte et reproductible.

Les fonctionnalités de planification que nous venons de voir (ajout de `TODOs`, de deadlines, de tags, de priorités, etc.) ne sont pas restreintes à un seul fichier défini comme « agenda global » : elles peuvent être utilisées dans *tout* fichier Org. En particulier, un *workflow* inspiré des principes suivants se révèlera très efficace en pratique lorsque l'on rédige un article scientifique.

— Utiliser un fichier Org pour écrire *la totalité* du texte de votre manuscrit. Il pourra ensuite être exporté en LaTeX, en utilisant la classe fournie par l'éditeur du journal [23].
— Pour vos analyses et modélisations, utiliser le système de programmation lettrée d'Org mode (Babel), de telle sorte que vous n'aurez aucun autre logiciel qu'Emacs à utiliser tout au long de la rédaction de l'article et de la production de vos résultats.
— Utiliser des statuts (`TODO`, `IN-PROGRESS`, `DONE`, ...) sur chaque section de votre article pour suivre l'avancement de votre travail ; attribuer également des dates de début de fin pour le travail sur chaque section.
— Pour chaque section, utiliser (peut-être ?) des tags correspondant aux noms des différents auteurs censés travailler dessus, afin de répartir explicitement et lisiblement le travail de tous.
— Pour certaines sections, utiliser des *quicknotes* pour indiquer toute information utile et censée rester privée (contretemps, incertitude, désaccord entre auteurs, etc.).
— Et bien sûr, versionner le tout avec Git pour suivre de manière encore plus lisible et transparente les progrès effectués sur l'article, et en garder un historique complet.

Noter qu'il est possible d'obtenir des vues agenda relatives à *un seul fichier Org précis*. Par exemple, lorsque vous ouvrez le menu agenda avec `C-c a`, vous pouvez voir que Org vous propose l'option en utilisant l'option d'une restriction au buffer courant avec la touche `<`. Ainsi, effectuer le raccourci `C-c a < a` vous affichera une vue agenda qui correspond uniquement aux entrées actives (`TODOs`) du fichier courant. Cela facilitera le suivi du travail sur un article précis, par exemple.

23. Un exemple de template Org prêt à l'emploi pour l'éditeur Wiley est disponible sur le dépôt `https://github.com/frederic-santos/org-template-ijoa`

Références

Allen, D. (2003). *Getting things done : The art of stress-free productivity*. New York : Penguin.

Cameron, D. (2002). *Emacs : précis & concis*. Paris : O'Reilly.

Chassell, R. J. (2009). *An introduction to programming in Emacs Lisp* (3rd éd.). GNU Press.

Desquilbet, L., Granger, S., Hejblum, B., Legrand, A., Pernot, P., & Rougier, N. (2019). *Vers une recherche reproductible*. Bordeaux : Unité régionale de formation à l'information scientifique et technique de Bordeaux. Consulté sur `https://hal.archives-ouvertes.fr/hal-02144142`

Dominik, C. (2010). *Org Mode 7 Reference Manual : Organize your life with GNU Emacs, release 7.3*. Bristol : Network Theory.