

Premiers pas avec Emacs-Muse

John Wiegley et Michael Olson (traduction Jean Magnan de Bornier)

28 juillet 2007

| | | |
|------|---------------------------------------------|----|
| 1 | Ce document | 2 |
| 2 | Pour commencer | 2 |
| 3 | Création d'un projet Muse | 2 |
| 4 | Règles de marquage | 3 |
| 4.1 | Paragraphe | 3 |
| 4.2 | Paragraphe centré et citations | 3 |
| 4.3 | En-têtes | 3 |
| 4.4 | Lignes horizontales | 3 |
| 4.5 | Emphase ou magnification du texte | 4 |
| 4.6 | Insérer des notes de bas de page | 4 |
| 4.7 | Les vers | 4 |
| 4.8 | Paragraphe sous forme littéraire | 5 |
| 4.9 | Listes | 6 |
| 4.10 | Tableaux | 7 |
| 4.11 | Balises et liens délimités | 8 |
| 4.12 | URLs et adresses électroniques | 8 |
| 4.13 | Images | 8 |
| 4.14 | Liens | 9 |
| 4.15 | Code source | 9 |
| 4.16 | Lisp, Perl, Ruby, Python, ou Shell embarqué | 10 |
| 5 | Styles de publication | 10 |
| 5.1 | Dériver à partir d'un style existant | 10 |
| 5.2 | Créer un nouveau style | 11 |
| 6 | Licence | 11 |

Emacs-Muse est un environnement d'écriture et de publication pour Emacs. Il simplifie le processus d'écriture et de publication de documents sous divers formats.

Muse consiste en deux parties principales: un mode d'édition de texte pour un écriture de documents et la navigation à l'intérieur des projets Muse, et un ensemble de styles de publication pour créer différentes sortes de produits finals.

1 Ce document

Ce document fournit un exemple de marquage de texte par Muse et fonctionne aussi comme une initiation à Muse.

Pour visualiser à quoi qu'il ressemblera publié, saisissez `make examples`. Vous obtiendrez alors un document info, un document HTML, et un document PDF (à condition qu'une distribution de LaTeX soit installée avec les fontes nécessaires).

2 Pour commencer

Pour utiliser Muse, on ajoutera le répertoire contenant les fichiers muse à la variable `load-path` de votre fichier `.emacs`. Ensuite on charge le `muse-mode` et les styles dans lesquels on souhaite publier. Par exemple :

```
(add-to-list 'load-path "<chemin où est Muse>")
```

```
(require 'muse-mode) ; chargement du mode écriture
```

```
(require 'muse-html) ; chargement des styles de publication qu'on utilise
```

```
(require 'muse-latex)
```

```
(require 'muse-texinfo)
```

```
(require 'muse-docbook)
```

```
(require 'muse-project) ; publication par projets
```

Une fois que ceci aura été chargé, la commande `M-x muse-publish-this-file` publiera le document saisi dans n'importe quel des styles disponibles. Si vous validez `muse-mode` à l'intérieur d'un tampon (buffer), en tapant `M-x muse-mode`, cette commande sera accessible avec `C-c C-t`.

3 Création d'un projet Muse

On veut souvent publier de manière automatique tous les fichiers d'un certain dossier avec un ensemble particulier de styles. Muse permet pour cela la création de « projets ». Voici un exemple de projet à définir dans votre fichier `.emacs` :

```
(setq muse-project-alist
  '(("website" ("~/Pages" :default "index")
    (:base "html" :path "~/public_html")
    (:base "pdf" :path "~/public_html/pdf"))))
```

Par les lignes ci-dessus en définit un projet nommé "website", dont les fichiers se trouvent dans le dossier `~/Pages`. La page à visiter par défaut est `index`. Quand ce projet sera publié, chaque page sortira en format HTML dans le dossier `~/public_html`, et comme PDF dans le dossier

~/public_html/pdf. Dans une page appartenant à un projet, vous pouvez créer un lien à d'autres pages en utilisant la syntaxe `[[nom_de_la_page]]`.

4 Règles de marquage

Un document Muse utilise des règles spéciales de marquage contextuel pour déterminer comment formater le résultat final. Par exemple, si un paragraphe est indenté, Muse suppose qu'il doit être mis en citation.

Il n'y a pas énormément de règles de marquage, et on a essayé qu'elles soient aussi simples que possible pour que l'utilisateur puisse se concentrer sur la création de son document plutôt que sur sa mise en forme.

4.1 Paragraphes

Dans Muse on sépare les paragraphes par une ligne blanche.

Par exemple, le texte de cette section est saisi comme ceci:

Dans Muse on sépare les paragraphes par une ligne blanche.

Par exemple, le texte de cette section est saisi comme ceci:

4.2 Paragraphes centrés et citations

Une ligne qui commence par six ou plus colonnes ou espaces (qu'il s'agisse de tabulations ou d'espace) indique un paragraphe centré.

Ceci est centré

« Mais quand une ligne commence par une espace de moins de six colonnes, cela indique un paragraphe de citation. »

4.3 En-têtes

Un en-tête devient le titre d'un chapitre ou d'une section dans le document final—en fonction du style. Pour indiquer un en-tête, on commence un nouveau paragraphe par un, deux ou trois astérisques, suivi d'une espace et du titre. Puis on commence un autre paragraphe pour saisir le texte de cette section.

* Premier niveau

** Second niveau

*** Troisième niveau

4.4 Lignes horizontales

Quatre tirets ou plus indiquent une ligne horizontale. Attention à ne pas oublier de les entourer de lignes blanches, faute de quoi ils seraient considérés comme appartenant au paragraphe précédent ou suivant!

Le séparateur ci-dessus a été produit en saisissant ceci :

4.5 Emphase ou magnification du texte

Pour mettre l'emphase sur du texte, on l'entoure de certains caractères spécifiques :

```
*emphase*
**emphase forte**
***emphase très forte***
_soulignage_
=verbatim et monospace =
```

La liste ci-dessus apparaîtra comme ceci :

```
emphase
emphase forte
emphase très forte
soulignage
verbatim et monospace
```

4.6 Insérer des notes de bas de page

Une note de bas de page sera appelée simplement par le nombre entre les parenthèses carrées [1].¹. Pour saisir la note, placer son texte à la fin de votre fichier. `footnote-mode` peut être utilisé pour faciliter grandement la création de ce type de note.

Footnotes:

```
[1] Des notes de bas de page sont identifiées par le même nombre entre
des parenthèses carrées en début de ligne. On peut utiliser la commande
C-c ! a du footnote-mode pour insérer des notes de bas de page en
saisissant du texte. C-x C-x permet de revenir au point d'insertion.
```

4.7 Les vers

Pour la poésie il faut que les espaces soient conservées, sans pour autant utiliser le monospace. Pour indiquer cela, on utilise le marquage suivant, qui rappelle les citations dans le courrier électronique :

```
>           HERNANI, *faisant un pas*
>
>                               Sire,
> La chose est toute simple, et l'on peut vous la dire.
> Nous gravions la sentence au mur de Balthazar.
>           *Il tire un poignard et l'agite*
```

¹ Ceci est une note.

```

> Nous rendions à César ce qu'on doit à César.
>     DON CARLOS
> Bien!
>     *À don Ruy Gomez*
>     Vous traître, Silva!
>     DON RUY GOMEZ
>
>                                     Lequel de nous deux, Sire?

```

Le paragraphe ci-dessus apparaîtra ainsi:

```

HERNANI, faisant un pas
Sire,
La chose est toute simple, et l'on peut vous la dire.
Nous gravions la sentence au mur de Balthazar.
    Il tire un poignard et l'agite
Nous rendions à César ce qu'on doit à César.
    DON CARLOS
Bien!
    À don Ruy Gomez
    Vous traître, Silva!
    DON RUY GOMEZ
    Lequel de nous deux, Sire?

```

(Victor Hugo: **Hernani**, IV, 4)

On peut aussi utiliser le délimiteur `<verse>` si on préfère :

```

<verse>
    HERNANI, *faisant un pas*
Sire,
La chose est toute simple, et l'on peut vous la dire.
Nous gravions la sentence au mur de Balthazar.
    *Il tire un poignard et l'agite*
Nous rendions à César ce qu'on doit à César.
    DON CARLOS
Bien!
    *À don Ruy Gomez*
    Vous traître, Silva!
    DON RUY GOMEZ
    Lequel de nous deux, Sire?
</verse>

```

4.8 Paragraphes sous forme littérale

Le délimiteur `<example>` est utilisé pour des exemples, quand on doit conserver les espaces, avec texte en monospace et où les caractères spéciaux ne doivent pas être interprétés .

Il y a aussi le délimiteur `<literal>` qui fait qu'un bloc marqué sera laissé entièrement de côté. On peut utiliser cela pour insérer des blocs de code HTML quand le format final est le HTML.

Si l'on veut qu'une partie de texte ne soit insérée que quand on publie dans un format particulier on peut utiliser l'attribut `style` du délimiteur `<literal>`. Voici quelques exemples.

Vous lisez la version
`<literal style="html">HTML</literal>`
`<literal style="pdf">PDF</literal>`
`<literal style="info">Info</literal>`
de ce document.

Produit ceci:
Vous lisez la version de ce document.

`<literal style="context">`
ConTeXt a été utilisé pour publier ce document.
`</literal>`

Produit:
ConTeXt a été utilisé pour publier ce document.

4.9 Listes

Les listes seront obtenues en utilisant des caractères spéciaux en début de ligne. Une espace doit être insérée avant les éléments d'une liste numérotée ou non, pour éviter la confusion avec l'utilisation habituelle de ces caractères.

Les types de listes reconnus sont les suivants:

- premier élément d'une liste non numérotée
 - deuxième élément d'une liste non numérotée
1. premier élément d'une énumération
 2. deuxième élément d'une énumération

Terme1 :: Description d'un premier élément

Terme2 :: Description d'un deuxième élément

Ces listes apparaîtront ainsi:

Liste non numérotée:

- premier élément d'une liste non numérotée
- deuxième élément d'une liste non numérotée

Énumération:

1. premier élément d'une énumération
2. deuxième élément d'une énumération

Définition:

- Terme1
Description d'un premier élément

- Terme2

Description d'un deuxième élément

Les listes peuvent être imbriquées l'une dans l'autre. Le niveau de l'imbrication est indiqué par la quantité d'espace en tête de ligne.

- Premier niveau, élément non numéroté 1
 1. Deuxième niveau, élément numéroté 1
 2. Deuxième niveau, élément numéroté 2
- Premier niveau, élément non numéroté 2

Ce qui apparaîtra ainsi :

- Premier niveau, élément non numéroté 1
 1. Deuxième niveau, élément numéroté 1
 2. Deuxième niveau, élément numéroté 2
- Premier niveau, élément non numéroté 2

4.10 Tableaux

Muse peut traiter des tableaux simples. La syntaxe est la suivante :

Double barre verticale || Séparation des champs d'en-tête

Barre verticale unique | Séparation des champs du corps du tableau
D'autres | Éléments du corps

Triple barre verticale ||| Séparation des champs du pied du tableau

Ce qui sera rendu ainsi²:

| | |
|------------------------|-------------------------------------------|
| Double barre verticale | Séparation des champs d'en-tête |
| Barre verticale unique | Séparation des champs du corps du tableau |
| D'autres | Élément du corps |

« Triple barre verticale| Séparation des champs du pied du tableau »

On peut aussi présenter les tableaux comme ceci :

| | | |
|------------------------|-------------------------------------------|--|
| Barre verticale unique | Séparation des champs du corps du tableau | |
| D'autres | Éléments du corps | |

Ce qui sera rendu ainsi :

| | |
|------------------------|-------------------------------------------|
| Barre verticale unique | Séparation des champs du corps du tableau |
| D'autres | Élément du corps |

² Les trois barres verticales pour le bas de tableau ne sont pas fonctionnelles dans les styles `latex` ou `context` et leurs dérivés.

Pour ceux qui sont familiers avec les tableaux dans le style Org-mode, des variantes simples (sans colonnes groupées ni formules autogénérées) seront prises en compte correctement. De même, les tableaux du style table.el seront publiés à condition que le format de publication soit connu de table.el.

4.11 Balises et liens délimités

Si vous commencez une ligne par “#balise”—où “balise” peut-être n’importe quel mot ne contenant pas d’espace—cela définit une balise à ce point du document. Ce point pourra ensuite être référencé en utilisant “page# balise” comme un cible dans un lien Muse (voir plus bas).

Cliquer [ici](#) pour revenir au paragraphe précédent.

4.12 URLs et adresses électroniques

Une URL ou une adresse e-mail rencontrée dans le texte est publiée comme un lien hypertexte si le style utilisé pour la publication le permet. Par exemple, la dernière version de Muse peut être téléchargée à l’adresse <http://download.gna.org/muse-el> et du courrier peut être envoyé à mwolson@gnu.org.

4.13 Images

Si une URL a comme extension celle d’un format d’image connu, elle sera affichée si possible. Pour créer un lien à une image plutôt que de l’afficher, mettre le texte “URL:” immédiatement devant le lien.

Voici un exemple d’image affichée :

Produit avec `[[muse-made-with.png]] Emacs Muse.`

Ce sera publié ainsi :

Produit avec



Figure 1

Emacs Muse.

Voici un exemple d’image avec légende :

`[[emacs-muse.png] [Muse, l’outil de publication (d’un sous-ensemble) des grands penseurs]]`

Qui sera publié ainsi :



Figure 2 Muse, l’outil de publication (d’un sous-ensemble) des grands penseurs

Ce qui suit sera publié uniquement sous forme de lien.

Le logo Muse: `[[URL:http://mwolson.org/static/logos/emacs-muse.png]]`.

Le logo Muse: <http://mwolson.org/static/logos/emacs-muse.png>.

4.14 Liens

Un hyperlien peut référencer une URL, ou une autre page à l'intérieur d'un projet Muse. De plus, un texte descriptif peut être spécifié, il sera affiché plutôt que le lien pour les styles qui acceptent les descriptions de lien. La syntaxe est la suivante :

```
[[lien cible][description]] [[lien cible sans description]]
```

Donc le texte:

Muse peut être téléchargé[[<http://download.gna.org/muse-el/>][ici]], ou ici :
[[<http://download.gna.org/muse-el/>]].

Sera publié comme ceci :

Muse peut être téléchargé [ici](http://download.gna.org/muse-el/)³, ou ici : <http://download.gna.org/muse-el/>.

4.15 Code source

Si `htmlize.el` (version 1.34 ou postérieure) est installé sur votre machine, il est possible de publier du HTML colorisé pour le code source de tout mode majeur supporté par Emacs en utilisant le délimiteur `<src>`. Si le style de publication n'est pas le HTML, le texte entre les délimiteurs est traité comme s'il y avait un délimiteur `<example>`.

Voici un exemple de code C. Muse cheche l'élément `lang` et lui ajoute `"-mode"` pour déterminer quel mode majeur doit être utilisé pour la colorisation du code.

```
<src lang="c">
#include <stdlib.h>

char *unused;

int main (int argc, char *argv[])
{
    puts("Hello, world!\n");
}
</src>
```

Voici le résultat colorisé. Son apparence peut varier si vous avez configuré certaines faces.

```
#include <stdlib.h>

char *unused;
```

³ <http://download.gna.org/muse-el/>

```
int main (int argc, char *argv[3])
{
    puts("Hello, world!\n");
}
```

4.16 Lisp, Perl, Ruby, Python, ou Shell embarqué

Des types de marquage arbitraires peuvent être réalisés grâce au délimiteur `<lisp>`, le seul des délimiteurs Muse supporté dans le texte d'en-tête ou de bas de page d'un style. Avec le délimiteur `<lisp>`, on peut générer n'importe quel texte final. Le produit du code sera marqué si le délimiteur `<lisp>` apparaît dans le corps du texte.

```
<lisp>(concat "Cette forme sera " "insérée")</lisp>
```

Ce code sera rendu ainsi:

Cette forme sera insérée

On peut aussi traiter le résultat comme s'il était à l'intérieur d'un des délimiteurs `<example>`, `<src>`, ou `<verse>`, en spécifiant "example", "src", ou "verse" comme attribut markup du délimiteur.

Par exemple:

```
<lisp markup="example">
(concat "Inserez" " moi")
</lisp>
```

Le résultat est:

Cet attribut markup peut aussi être passé aux délimiteurs `<perl>`, `<ruby>`, `<python>`, and `<command>`, qui interprètent respectivement les codes Perl, Ruby, Python, Shell.

5 Styles de publication

Une des caractéristiques majeures de Muse est sa capacité à publier un simple texte de départ dans une variété de styles à l'arrivée. Muse permet aussi facilement de créer de nouveaux styles, ou d'en dériver à partir de ceux qui existent.

5.1 Dériver à partir d'un style existant

Pour créer un nouveau style à partir d'un style existant, on utilise la fonction `muse-derive-style`:

```
(muse-derive-style NOM-DÉRIVÉ NOM-DE-BASE PARAMÈTRES-DU-STYLE)
```

Le nom dérivé est une chaîne de caractères définissant le nouveau style, comme "mon-html". Le nom de base doit identifier une style existant, comme "html"—si on a chargé `muse-html`. Les paramètres de style sont les mêmes que ceux qu'on utilise pour créer un style, et ils remplacent les définitions présentes dans le style de base.

Le plus souvent, cela sera utile pour définir un en-tête, un pied de page, une feuille de style, pour un projet particulier. En voici un exemple.

```
(setq ma-feuille-de-style
  (concat "<link rel=\"stylesheet\" type=\"text/css\""
    " charset=\"utf-8\" media=\"all\""
    " href=\"/different.css\" />"))

(muse-derive-style "mon-xhtml" "xhtml"
  :header "~/.emacs.d/muse/different-header.html"
  :footer "~/.emacs.d/muse/different-footer.html"
  :style-sheet ma-feuille-de-style)
```

De nombreux paramètres supportent d'être ainsi remplacés partiellement. Par exemple: avec `:functions`, si une fonction de markup n'est pas trouvée dans la liste de fonctions du style dérivé, c'est la fonction du style de base qui sera appelée.

Si un paramètre prend le nom d'une fonction, alors si non-nil est retourné par cette fonction, aucun autre processus ne sera appelé. Si la fonction retourne nil, tout autre fonction dans le style de base sera appelée.

5.2 Créer un nouveau style

Pour créer un nouveau style, utiliser `muse-define-style`:

```
(muse-define-style NOM PARAMETRES-DE-STYLE)
```

Quand on veut créer un nouveau style, le mieux est d'examiner d'abord le code source d'autres styles, pour avoir une idée de ce qui est nécessaire. Chaque format de sortie devrait avoir un fichier propre, contenant tous les styles ayant la même base. Par exemple, les styles `latex`, `latex-slides`, `pdf` et `latex-slides-pdf` sont tous inclus dans le fichier `muse-latex.el`.

Si vous désirez être l'auteur de documents avec copyright pour la Free Software Foundation (nous vous aiderons dans cette démarche), les auteurs de Muse pourraient être intéressés à inclure votre travail dans les versions futures de Muse.

6 Licence

Ce document "premiers pas" peut être redistribué et/ou modifié selon les termes de la GNU General Public License telle que publiée par la Free Software Foundation; la version 2, ou (suivant votre choix) toute version postérieure.

« Une autre note, non référencée »