

Небольшая шпаргалка по Muse (Emacs Muse)

Клавиатурные команды

- **C-c C-a** ('muse-index')—список всех известных muse-страниц
- **C-c C-b** ('muse-find-backlinks')—поиск всех страниц ссылающихся на эту
- **C-c C-e** ('muse-edit-link-at-point')—редактировать ссылку под курсором
- **C-c C-f** ('muse-project-find-file')—открыть другую muse-страницу
- **C-c C-i l**, **C-c TAB l** ('muse-insert-relative-link-to-file')—относительная ссылка на другой файл
- **C-c C-i t**, **C-c TAB t** ('muse-insert-tag')—вставка тега
- **C-c C-i u**, **C-c TAB u** ('muse-insert-url')—вставка URL
- **C-c C-l** ('font-lock-mode')—переключить режим подсветки синтаксиса
- **C-c C-p** ('muse-project-publish')—публикация всех изменённых страниц в проекте
- **C-c C-s** ('muse-search')—поиск текста во всех файлах проекта
- **C-c C-t** ('muse-project-publish-this-file')—публикация текущего файла в проекте
- **C-c C-S-t**, или **C-c C-M-t** ('muse-publish-this-file')—публикация текущего файла
- **C-c C-v** ('muse-browse-result')—показать результаты публикации, если нужно, то запустить процесс публикации
- **C-c =** ('muse-what-changed')—показать изменения, по сравнению с последней резервной копией
- **TAB**—переместиться к следующей wiki-ссылке
- **S-TAB**—переместиться к предыдущей wiki-ссылке
- **M-TAB**—автодополнение имени страницы для текущего проекта
- **M-RET**—вставка нового элемента списка
- **C-<**—уменьшить отступ текущего элемента списка (изменить уровень вложенности)
- **C->**—увеличить отступ текущего элемента списка (изменить уровень вложенности)
- **M-x muse-colors-toggle-inline-images RET**—переключить режим отображения ставленных изображений
- **M-x muse-update-values RET**—обновить различные автосгенерированные переменные, например, после модификации muse-project-alist.

Язык разметки

Абзацы

- Должны быть разделены пустой строкой (собственно как у всех Wiki)
- 6 или более пробелов в начале строки или тег `<center>`—центровка абзаца
- менее 6 пробелов или тег `<quote>`—режим цитирования
- `<example>` тег используется для публикации контента “как есть”, шрифт выбирается моноширинный
- `<literal>` для вставки html кода буквально
- `
`—для ручного разрыва строки

Заголовки

Всего 4 уровня начинаются с соответствующего количества звездочек (*) в начале строки

Преамбула документа

Или директивы в начале документа. Директивы начинаются с символа решётка (#) в начале строки.

- #author—автор документа
- #date—подставляет дату последней модификации документа
- #desc—короткое описание документа
- #title—заголовок документа

Начертание шрифта

- *курсив*—*курсив*
- **жирный**—**жирный**
- ***жирный курсив***—***жирный курсив***
- подчёркнутый—подчёркнутый
- =код= или тег <code>код</code>—код, вторая форма позволяет включать '=' в текст

Сноски

Сноска—просто цифра в квадратных скобках ([1]). Сами сноски нужно располагать в самом низу документа—начинаются с того же номера в квадратных скобках, но с самого начала строки. Удобно использовать режим `footnote-mode` и клавиатурные комбинации:

- C-c ! a—для вставки сноски
- C-c ! d—для удаления сноски
- C-c ! g—для перехода к сноске
- C-c ! r—перенумеровать сноски
- C-c ! s—установить стиль сносок (по умолчанию - арабские цифры)
- C-x C-x—для возврата в точку вставки

Стихи

В начале строки ставим '>' и дальше пишем стих, пробелы сохраняются. Альтернативно— использовать тег <verse>

Списки

- пробел, '-' и пробел—простой нумерованный список, для вложенного списка - добавить ещё пробел
- пробел, цифра, точка и пробел—нумерованный список, для вложенного списка требования аналогичные как у нумерованного.
- Определения терминов:

Определение1 :: формулировка

Определение2 ::

формулировка

формулировка

Таблицы

Самые простейшие.

```
Заголовок1 || Заголовок2
Col1      | Col2
Подвал1   ||| Подвал2
```

Выключить генерацию таблиц, добавить в заголовке команду `#disable-tables t`

Есть возможность использовать таблицы из `org-mode` (точнее подрежима `tbl-mode`), за подробностями в документацию: [Generation of data tables](#)¹

Гиперссылки

Тут всё просто:

- `[[url]]`—ссылка без подписи
- `[[url] [подпись]]`—ссылка с подписью

За дополнительными сведениями: [Bare URLs, WikiNames, and InterWiki links](#)²

Изображения

- `[[url] [путь к локальной картинке]]`—картинка-ссылка
- `[[URL:ссылка]]`—вариант вставки картинки с удалённого сайта
- `[[путь к картинке]]`—вариант вставки картинки с диска
- `[[путь к картинке] [заголовок]]`—отцентрирует картинку и выведет подпись к ней

Горизонтальные линии и якоря

Нарисовать линию: набрать 4 или более минусов----

Поставить якорь: ввести `#метка_без_пробелов`, сослаться: `page#метка`

Выполнение lisp-кода

Для этого служит тег `<lisp>`

Результат работы можно обрмить в окружения `example`, `src`, `verse`, передавая их имена как значение опциональному параметру—`markup`.

Цитирование сторонних источников

Не хочу особо расписывать, отошлю к основной документации: [Support for citing other resources](#)³

Комментарии

Если не хотим, что бы строчка появилась в результатах, поставим в самом начале символ ; (точка с запятой)

Известные теги

Выше уже упоминались некоторые специальные теги, посмотреть все можно в документации: [Tags that Muse recognizes](#)⁴

Ниже небольшой список оных.

`
`

разрыв строки

¹<http://mwolson.org/static/doc/muse.html#Tables>

²<http://mwolson.org/static/doc/muse.html#Implicit-Links>

³<http://mwolson.org/static/doc/muse.html#Citations>

⁴<http://mwolson.org/static/doc/muse.html#Tag-Summary>

<cite>
вставка цитаты другого источника

<class>
при публикации в HTML вставит блок span, имеет опциональный аргумент—name— который задаёт css-class для span

<code>
блок кода

<command>
отправляет данные блока интерпретатору, имеет опциональный аргумент—interp—, который указывает тип интерпретатора (по умолчанию—shell). Опциональный аргумент—markup—управляет, как результаты будут обработаны:

- аргумент пропущен—будет выведено как есть, muse обработает результаты
- nil—будет выведено как есть, но muse обрабатывать не будет
- example, src, verse—будет окружено соответствующим тегом

<comment>
выделить область как комментарий, что бы предостеречь от публикации и/или вывода на экран

<contents>
опубликовать содержание, опциональный аргумент—depth—задаёт уровень оглавления (по заголовкам)

<div>
при публикации в HTML, вставляет div блок, имеется два опциональных аргумента: style и id, смысл которых аналогичен оным в HTML

<example>
примеры кода

<include>
включить указанный файл в текущую позицию при публикации, обязательный аргумент— file— указывающий файл для включения. Опциональный аргумент—markup— говорит как представлять вставленный блок (аналогично тегу <command>)

<lisp>
выполнить lisp-команды, принимает опциональный аргумент—markup—аналогичный оному в теге <command>

<perl>
аналогичен <lisp>, только вызывается perl

<python>
даже повторяться не хочется :)

< quote>
цитирование (временами работает некорректно, так вот тут сейчас пришлось поставить пробел, ибо ломало форматирование всего)

<ruby>
выше было уже про перлы и лиспы. . .

<src>
регион подвергается HTMLизации (htmlize) и вставляется, принимает опциональный аргумент— lang—задающий язык разметки

<verbatim>
предостерегает muse от каких либо действий над блоком

<verse>
стихи