

# Fountain Mode

---

GNU Emacs major mode for screenwriting in Fountain markup

Paul W. Rankin

---

Copyright © 2021–2023 Paul W. Rankin

The following people contributed to this documentation:

Paul W. Rankin, Kohl Sudduth, Sandra Snan

# Overview

Fountain Mode combines the simplicity of Fountain syntax with the extensibility of Emacs. Fountain Mode is a major mode for Emacs, which is to say it is a program that runs inside Emacs — the extensible, customizable, free/libre text editor. It enables the writer to engage with all aspects of realizing a screenplay — story development, outlining, drafting, and rewriting.

To quickly get up to speed with Fountain syntax, memorize the syntax for the six most used elements of the screenplay format: [Scene Headings], page 3, [Action], page 3, [Characters], page 4, [Dialogue], page 4, [Parentheticals], page 4, and [Transitions], page 5.

Then learn how to outline a script. See [Section Headings], page 2. This will be familiar for anyone who has used Markdown, as the syntax is the same. Sections allow you to easily show or hide and reorder large parts of a script.

There are additional Fountain syntax elements to allow for brainstorming, capturing ideas and omitting a part of the script without deleting it. See [Synopses], page 5, [Notes], page 5, and [Comments], page 6. These elements are not usually included in the formatted output.

All of Fountain Mode’s capabilities are accessible via the menu, and all customization is performed via the **Fountain** option group. See [\[\(emacs\) Customization Groups\]](#), page [\[undefined\]](#).

## Script Elements

The central idea of writing a script in Fountain is that you should be able to just start writing — as long as your text looks like a script, you should get correctly formatted output.

**n.b.:** Fountain Mode uses the Fountain 1.1 syntax specification. While some programs use custom additions to the specification, for the greatest cross-compatibility, stick with the 1.1 spec.

## Metadata

A Fountain script usually starts with some metadata stating the basic information about the script. These are colon-separated key-value pairs at the start of a file:

`key: value`

Other Fountain-compatible programs may refer to this as “title page” info, but metadata can store information not related to or present on the title page.

Calling `auto-insert` will guide you through adding the most common Fountain metadata, prompting with likely defaults:

- `title` is pretty obvious, and will default to base-name of the current buffer.
- `credit` is not actually your name, but the type of credit given to the `author`, e.g. `written by` or `screenplay by`.
- `author` is you or your writing team, and defaults to the value of variable `user-full-name`.
- `format` will override the value of `fountain-default-script-format` for the current script. Accepted values are: `screenplay`, `teleplay`, `stageplay`.
- `date` defaults to the current date in your locale’s “preferred” format.
- `source`, for if you’re writing an adaptation.
- `contact` is useful if the writer wants readers to ever get in touch to make the film/play!

All metadata keys can take either a single line of input following the colon, or a newline followed multiple lines indented with whitespace:

```
author:
  Mary Maryland
  and
  Alan Smithee
```

Metadata keys are case-insensitive, e.g. ‘Title’ is the same as ‘title’.

## Section Headings

Section headings are lines beginning with `#` (number sign). There are five levels of section headings, with additional `#` characters demoting the outline level of a heading.

`# Top-Level Heading`

`## Sub-Heading`

`### Sub-Sub-Heading`

See [Outlining], page 7.

## Scene Headings

Scene headings begin with a prefix, specified in `fountain-scene-heading-prefix-list`, which defaults to:

INT, EXT, EST, INT./EXT., INT/EXT, I/E

Each prefix may be followed by a dot and/or a space, so the following are equivalent:

INT HOUSE - DAY

INT. HOUSE - DAY

`fountain-scene-heading-prefix-list` [User Option]

This options allows setting your own scene heading prefixes. These entries are case-insensitive.

`fountain-scene-heading-suffix-list` [User Option]

This options allows setting your own scene heading suffixes (or time-of-day). These entries are case-insensitive.

`fountain-scene-heading-suffix-separator` [User Option]

This option allows setting the separator (a regular expression) between the scene location and time of day. The default ‘ --? ’ allows either one or two dashes.

`fountain-auto-upcase-scene-headings` [User Option]

Enabling this option will automatically upcase text after a scene heading prefix as you type.

`fountain-double-space-scene-headings` [User Option]

Enabling this option will display two blank lines above scene headings, regardless of what’s in the file. n.b. This option does not affect calculation of pagination.

See [Scene Heading Completion], page 13.

## Action

Action is the easiest Fountain element — anything that isn’t parsed as another element is considered action.

Sometimes you may write some action that will be unintentionally parsed as dialogue, for example:

Jack examines his shopping list...

BLACK SHIRT

BLACK PANTS

EXPLOSIVES

MAP

Here **BLACK SHIRT** would be parsed as a character, who then shouts “**BLACK PANTS!...**” We don’t want that. To prevent this, prefix the first line with **!** (exclamation mark). This is known as a “forced” element.

**!BLACK SHIRT**

BLACK PANTS  
EXPLOSIVES  
MAP

## Characters

To write a character name to cue some dialogue, type the name in uppercase (ALL-CAPS).

JOLENE

The next line will be parsed as dialogue.

Sometimes you might want to include lowercase letters in a character name. To force a line as a character name, prefix it with @ (at sign).

@JOLENE McCLOGGS

Some Fountain tools may not parse a line as a character name if the extension contains lowercase letters, requiring the @ prefix.

@JOLENE (cont'd)

If you are just writing the character's name within action, type the character's name normally.

Jolene

If you prefer to write character names within action in uppercase, that's fine too. The following will still be parsed as action.

JOLENE throws the chair at PHILIP.

**fountain-character-extension-list** [User Option]  
This option allows setting your own character extensions, e.g. "(O.C.)".

See [Character Name Completion], page 13.

## Dialogue

Dialogue is any text following [Characters], page 4. Just enter a newline and the next text entered will be parsed as dialogue.

JOLENE  
Have you seen trouble? I seem to have misplaced it.

## Parentheticals

Any text inside parentheticals and within dialogue is parsed as a parenthetical element, e.g.

JOLENE  
(concerned)  
Have you seen trouble? I seem to have misplaced it.

See [Autocompletion], page 13.

## Transitions

A transition is any line ending with **TO:**, or you can force a transition by prefixing it with **>**.

**fountain-trans-suffix-list** [User Option]

This option allows setting your own transition suffixes, so that lines ending in these will be parsed as transitions. These entries are case-insensitive.

## Notes

Text enclosed in **[[ ]]** (double brackets) is parsed as a note, and is generally not included in export. Use these for keeping draft notes, research, feedback, etc.

**C-c C-z** This will insert a note based on **fountain-note-template** (see below) beneath the current element, or if the region is active, surround the region with note delimiters (**fountain-insert-note**).

**fountain-note-template** [User Option]

Template for inserting a new note. Passed to **format-spec** with the following specification:

- **%u** user-login-name
- **%n** user-full-name
- **%e** user-mail-address
- **%x** date in locale's preferred format
- **%F** date in ISO format
- **%P** leave point here

The default **%P - %n %x** inserts something like:

```
[[ | - Alan Smithee 12/31/2017 ]]
```

## Synopses

A synopsis is a handy way to detail what a scene or section is about. A synopsis element is simply a line beginning with **=** (equals sign).

```
INT. FISHING TRAWLER - DAY
= The men eat the shark they caught.
```

Synopses are not included by most export tools.

## Center Text

Any text surrounded with **>** and **<** (greater and less than signs) will be centered in output, e.g.:

```
> Once Upon a Time... <
```

## Comments

Text that are in C-style comments, `/* like this */`, is not exported by export tools. It's text that's "in the boneyard". Unlike other Fountain syntax, these comments can span newlines and other formatting; structurally, it's as if the text were removed.

This "boneyard" is for text that you're on the brink of removing, or drafts that you aren't ready to "uncomment" and include yet, or just anything else you want to ignore.

If you instead just want to annotate your screenplay, [Notes], page 5, or [Synopses], page 5.



## Outlining

Acts and sequences can be given meaningful section titles, giving you an overview of the story structure.

There are five levels of section headings. Scene headings count as the sixth level headings.

Cycle an individual subtree visibility with *TAB*. Cycle global outline visibility with *S-TAB* (shift-tab) or *C-u TAB*.

**fountain-insert-section-heading** [Command]  
Bound to *M-RET*, insert an empty section heading at the current outline level.

**fountain-outline-to-indirect-buffer** [Command]  
If you want to focus on discrete sections of your script you can open these in indirect buffers. Bound to *C-c C-x b*, this command clones the current section or scene to indirect buffer.

See [\(undefined\)](#) [(emacs) Indirect Buffers], page [\(undefined\)](#),

**fountain-pop-up-indirect-windows** [User Option]  
Set this to control how indirect buffer windows are opened. Sometimes you might want to limit your focus to one sequence, other times you might want to look at two scenes in windows side-by-side. Set this option to spawn a new window.

## Navigation

Because a Fountain script is structured text, there are plenty of ways to quickly move to where you want to be.

<i>C-M-n</i>	Move to the next visible scene or section heading ( <code>fountain-outline-next</code> ).
<i>C-M-p</i>	Move to the previous visible scene or section heading ( <code>fountain-outline-previous</code> ).
<i>C-M-f</i>	Move forward to the next visible scene or section heading at same outline level ( <code>fountain-outline-forward</code> ).
<i>C-M-b</i>	Move backward to the previous visible scene or section heading at the same outline level ( <code>fountain-outline-backward</code> ).
<i>C-M-a</i>	Move to the beginning of the current scene or section ( <code>fountain-outline-beginning</code> ).
<i>C-M-u</i>	Move up to the parent section heading ( <code>fountain-outline-up</code> ).
<i>M-n</i>	Move to next character name ( <code>fountain-forward-character</code> ).
<i>M-p</i>	Move to previous character name ( <code>fountain-backward-character</code> ).
<i>C-x ]</i>	Move to next page. This command does not move by one page but actually moves to the next page-break point in the text pagination properties ( <code>fountain-forward-page</code> ).
<i>C-x [</i>	Move to previous page ( <code>fountain-backward-page</code> ).
<i>M-g s</i>	Prompt for a scene number string and move to the specified scene. If scenes are numbered, this command will take you to the scene matching that string, otherwise it will count the scenes from beginning of the accessible buffer, recalculating when encountering a numbered scene ( <code>fountain-goto-scene</code> ).
<i>M-g p</i>	Prompt for a number and move to the specified page. This command does not actually count pages but uses the text pagination properties ( <code>fountain-goto-page</code> ).

## Syntax Highlighting

In Fountain Mode, each Fountain element can have Font Lock syntax highlighting toggled individually — or none at all. See [\(emacs\) Font Lock](#), page [\(undefined\)](#).

In addition to faces for specific elements, Fountain Mode uses an Emacs feature called face remapping, which means the `default` face is displayed using the `fountain` face. This allows you to customize the `fountain` face just for Fountain Mode buffers. See [\(emacs\) Face Customization](#), page [\(undefined\)](#).

**fountain-highlight-elements** [User Option]

This option is a list of Fountain elements to be highlighted.

**fountain-toggle-hide-emphasis-markup** [Command]

Bound to `C-c C-x *` this command will toggle the display of text emphasis markup. See [\[Text Emphasis\]](#), page 11.

**fountain-toggle-hide-element-markup** [Command]

Bound to `C-c C-x !`, this command, like above, will toggle the display of element markup. This means the leadings characters in section headings and forced elements.

## Element Aligning

In addition to syntax highlighting, Fountain Mode has the ability to automatically align, or indent, script elements, so your script looks more like real script. This is display only and does not change the file contents, and can be turned off if you don't like it.

**fountain-align-elements** [User Option]  
Whether to automatically align elements.

Different script formats have different alignment, so each element can be aligned on a per-format basis — see the Customize group **fountain-align**.

## Text Emphasis

Text can be underlined, italic, bold, or a combination thereof. Demarcate emphasized text with:

- `_underscores_` for underline,
- `*single asterisks*` for *italic*,
- and `**double asterisks**` for **bold**.

For the writer persists who want to work the way our ancestors did on typewriters, stick to underlining.

## Do What I Mean

Like many screenwriting programs, in Fountain Mode pressing **TAB** will do the most convenient thing based on context.

The most common use is triggering autocompletion. If the point is at a blank line or the end of a non-blank line, pressing **TAB** will call **completion-at-point**. See [Autocompletion], page 13.

In Fountain Mode, **TAB** is also used to control outline visibility. So if point is at a scene or section heading, it will cycle visibility of that scene or section between collapsed and expanded. To allow for more control over outline cycling, if **TAB** is prefixed with **ARG**, call **fountain-outline-cycle** and pass **ARG**. See [Outlining], page 7.

**TAB** also helps working with parentheticals. If the point is at a blank line within dialogue, it will insert a parenthetical; if the point is inside an empty parenthetical, it will delete it, or if inside a non-empty parenthetical, move to a newline.

If the point is at or inside a note, **TAB** will cycle visibility of that note between collapsed and expanded.

This all might seem complicated, but the idea is by covering all the cases you don't have to think about it.

**fountain-dwim**

[Command]

This is the command you'll use the most. Bound to **TAB**, it will perform the most convenient action based on the current context.

## Autocompletion

One of the nice things about using a dedicated screenwriting program is that it helps you type less of the things you need to type a lot. Fountain Mode provides autocompletion for scene headings and character names.

**fountain-completion-update** [Command]

This command, bound to `C-c C-x a` will update the completion candidates for current buffer.

## Scene Heading Completion

If the line has a partial scene heading, i.e. it begins with a prefix from **fountain-scene-heading-prefix-list** like so:

INT. |

*TAB* will offer completions of previously used locations.

If the cursor is at the time-of-day, like so:

INT. SUBMARINE - |

*TAB* will offer completions from **fountain-scene-heading-suffix-list**.

## Character Name Completion

The most basic use of this is when pressing *TAB* on an empty line. If there's an empty line above, this will offer to autocomplete a character name. Character names are suggested in the order:

1. the second-to-last previously speaking character within the current scene, i.e. a character's conversational partner;
2. the last speaking character within the current scene, i.e. a character continuing speaking;
3. the remaining characters in the script in order of frequency (default if there are not yet speaking characters in the current scene).

*TAB* will also offer character name completion if a line has a partial character name.

When the cursor is after a character name and opening parenthesis, *TAB* will offer completions from **fountain-character-extension-list** plus **fountain-continued-dialog-string**.

MARY (|

When the cursor is at an empty line within dialogue, *TAB* will add an empty parenthetical.

MARY

|

I'm hungry.

Likewise, if the cursor is within an empty parenthetical, *TAB* will remove the parenthetical.

MARY

(|)

I'm hungry.

When the cursor is at the end of a non-empty parenthetical, either inside or outside the closing parenthesis, *TAB* will move to the beginning of the next line if the next line is non-empty, otherwise it will insert a newline.

MARY

(angry|)

I'm hungry.

When the cursor is at the end of a non-empty line of dialogue, and the value of `fountain-dwim-insert-next-character` is non-nil, *TAB* will insert an empty line and the second-to-last previously speaking character.

MARY

(angry)

I'm hungry. |

The cursor will be left at the end of the next character, allowing successive presses of *TAB* to cycling through additional character completion candidates.

MARY

(angry)

I'm hungry.

JOHN |



## Scene Numbering

Scene numbers are appended to scene headings between `# #` (hash). Fountain Mode can automatically add scene numbers for you.

**fountain-add-scene-numbers** [Command]  
 Bound to `C-c C-x #`, add scene numbers to scene headings in current buffer. Remove scene numbers by prefixing with `C-u`.

Whether scene numbers are applied or not, you can still navigate by scene number. See [Navigation], page 8.

Scene numbers can be displayed in the window margins.

**fountain-display-scene-numbers-in-margin** [User Option]  
 Display scene numbers in left and right margins.

## Pagination

Fountain Mode features internal pagination. This is achieved by periodically calculating where page breaks occur in the whole buffer and applying pagination properties to the text. When the text is changed enough to invalidate these properties they are recalculated.

**fountain-pagination-max-change** [User Option]

Maximum change in page characters before invalidating pagination. Set this to a lower value to get more accurate pagination, high if you find the updates annoying.

**fountain-page-size** [User Option]

An important factor in calculating pagination is the page size. This is the target page size. Valid options are **letter** or **a4**.

**fountain-page-max-lines** [User Option]

This option specifies how many lines fit on each of the page sizes.

**fountain-pagination-break-sentences** [User Option]

Whether pagination will break pages within sentences, or only at sentence ends.

You can optionally show the current page and total page count in the mode line by enabling **which-function-mode**, See [\(emacs\) Which Function](#), page [\(undefined\)](#). This will display something like this:

[Page 31 of 108]

**fountain-count-pages** [Command]

Bound to **C-c C-p**, return both the current page and the total page count of the current buffer.

You can also navigate by page number. See [\[Navigation\]](#), page 8.

## Exporting

Fountain Mode can export to PostScript or PDF using the troff format as an intermediary. This requires only a **troff** program on your system, such as GNU roff (<https://www.gnu.org/software/groff/>). One advantage of this is that the internal pagination will match the output exactly.

Alternatively you can export using an external command-line program, which is covered further below.

**fountain-export-troff** [Command]

This command, bound to `C-c C-e t`, will export using the internal exporter and typeset using a **troff** program.

**fountain-export-view** [Command]

This command, bound to `C-c C-v`, attempts to open the last exported file. This works by finding the other most recently modified file in the current directory matching the current file base-name.

**fountain-export-format** [User Option]

This is the target format for internal troff export. Valid options are Postscript (**ps**) or PDF (**pdf**).

**fountain-export-troff-command** [User Option]

The name of the **troff** program on your system.

**fountain-export-troff-extra-options** [User Option]

Extra option flags passed to the **troff** program. The default tells **troff** the to use UTF-8.

**fountain-export-title-page** [User Option]

Whether to include a title page in export. This only affects the internal troff export.

**fountain-export-number-first-page** [User Option]

Whether to include a page number on the first page in export. This only affects the internal troff export.

**fountain-export-scene-heading-format** [User Option]

A list specifying the export scene heading format. Available options are zero or more of **bold**, **double-space**, and **underline**. This option only affects the internal troff export.

## Exporting Using an External Program

As an alternative to the internal export system you can use an external command-line program.

Here are some recommended export programs:

- Afterwriting (external site) (<https://github.com/ifrost/afterwriting-labs>)
- Wrap (external site) (<https://github.com/Wraparound/wrap>)
- Screenplain (external site) (<https://github.com/vilcans/screenplain>)

- Textplay (external site) (<https://github.com/olivertaylor/Textplay>)<sup>1</sup>

By defining an “export profile” you can easily interface with an external program from within Emacs. A profile is essentially a shell command, interpolated with a selection of values:

- **%b** is the **buffer-file-name**
- **%B** is the **buffer-file-name** sans extension
- **%n** is the **user-full-name**
- **%t** is the title (See [Metadata], page 2.)
- **%a** is the author (See [Metadata], page 2.)
- **%F** is the current date in ISO format
- **%x** is the current date in your locale’s “preferred” format

The default command profiles are only intended as examples. You are encouraged to edit/replace these to suit your own needs.

**fountain-export-command** [Command]

This command, bound to **C-c C-e e**, will prompt for an export profile. The first profile is considered default, so just hitting RET is a quick way to export in your usual way.

**fountain-export-command-profiles** [User Option]

This option contains the list of shell command profiles for exporting. The first profile is considered the default.

---

<sup>1</sup> Requires PrinceXML (<https://www.princexml.com>) for PDF export (external site).

## Bugs and Feature Requests

To report bugs and feature requests that affect the Debian package, please use the following method:

```
$ sudo apt install reportbug  
$ reportbug elpa-fountain-mode
```

## Contact

If you run into any trouble using Fountain Mode, or you have a feature request, you can email the maintainer Paul W. Rankin directly at [hello@paulwrankin.com](mailto:hello@paulwrankin.com).

For bugs, please ensure you can reproduce with:

```
$ emacs -Q -l fountain-mode.el
```

You can also try the `#emacs` IRC channel on Libera.chat (external site) (<https://libera.chat>) where Fountain Mode's maintainer uses the nickname `rnkn` or `pwr`, but please note that most other users on the channel are unlikely to be screenwriters.

Financial reward is not a consideration in maintaining Fountain Mode, but if you would like to support development, donations are graciously accepted via GitHub(external site) (<https://github.com/sponsors/rnkn>) or Liberapay(external site) (<https://liberapay.com/rnkn/>).

# Indexes

## Key Index

### C

C-c C-e e.....	18
C-c C-e t.....	17
C-c C-p.....	16
C-c C-v.....	17
C-c C-x #.....	15
C-c C-x a.....	13
C-c C-x b.....	7
C-c C-z.....	5
C-M-a.....	8
C-M-b.....	8
C-M-f.....	8
C-M-n.....	8
C-M-u.....	8
C-x [.....	8
C-x ].....	8

### M

M-g p.....	8
M-g s.....	8
M-n.....	8
M-p.....	8
M-RET.....	7

### S

S-TAB.....	7
------------	---

### T

TAB.....	7, 12, 13
----------	-----------

## Index of Commands

### B

backward-character.....	8
backward-page.....	8

### C

completion-update.....	13
count-pages.....	16

### D

dwim.....	12
-----------	----

### E

export-command.....	18
export-view.....	17

### F

forward-character.....	8
forward-page.....	8
fountain-add-scene-numbers.....	15
fountain-completion-update.....	13
fountain-count-pages.....	16
fountain-dwim.....	12
fountain-export-command.....	18
fountain-export-troff.....	17
fountain-export-view.....	17

fountain-insert-section-heading.....	7
fountain-outline-to-indirect-buffer.....	7
fountain-toggle-hide-element-markup.....	9
fountain-toggle-hide-emphasis-markup.....	9

### G

goto-page.....	8
goto-scene.....	8

### I

insert-note.....	5
insert-section-heading.....	7

### O

outline-backward.....	8
outline-beginning.....	8
outline-forward.....	8
outline-next.....	8
outline-previous.....	8
outline-to-indirect-buffer.....	7
outline-up.....	8

## Index of User Options



fountain-align-elements.....	10	fountain-highlight-elements .....	9
fountain-auto-upcase-scene-headings .....	3	fountain-note-template .....	5
fountain-character-extension-list.....	4	fountain-page-max-lines .....	16
fountain-display-scene-numbers-in-margin .	15	fountain-page-size .....	16
fountain-double-space-scene-headings .....	3	fountain-pagination-break-sentences .....	16
fountain-export-command-profiles.....	18	fountain-pagination-max-change .....	16
fountain-export-format .....	17	fountain-pop-up-indirect-windows .....	7
fountain-export-number-first-page .....	17	fountain-scene-heading-prefix-list .....	3
fountain-export-scene-heading-format .....	17	fountain-scene-heading-suffix-list .....	3
fountain-export-title-page.....	17	fountain-scene-heading-suffix-separator....	3
fountain-export-troff-command .....	17	fountain-trans-suffix-list.....	5
fountain-export-troff-extra-options .....	17		