# Book Writing the Advisicon Way

This document contains the styles necessary to write a book manuscript (and examples of their use). Use this document as a template!

## Self-Contained “Chapters”

In the new generation of Advisicon books, we’re aiming for each chapter to exist as a discreet, self-contained component. There are a few reasons for this:

* We recycle our chapters into other books (see the Advisicon Book Daisy Chain to learn the feeding order). By keeping each chapter self-contained, we can keep all books with shared chapters updated as we make changes.
* Self-contained chapters mean that we can offer our content in smaller chunks. This allows our customers to mix and match chapters for a customized learning experience.
* We don’t need to complete a book to be able to put out new content. We can create targeted content as it is needed.

We may want to stop referring to these as chapters and start calling them modules instead.

### Chapter Title

Since each module is self-contained, make sure that each module you write is its own file. Give the module a name with the document’s filename, this will become the chapter name when the module appears in a book. For example, because this document’s filename is book template.docx the chapter will be named “Book Template.”

## Semantic Markup

In order to make our authors’ meaning clear and consistent Advisicon requires its authors to markup their text semantically. To markup your text semantically, you need to use paragraph and character styles that describe the meaning of the marked up text. Marking up text based on looks, presentation, color, size, font, etc. is an inappropriate use of a style.

Some of the reasons we are stressing semantic markup:

* To expedite the proofreading and editing process. Semantic markup will help peer reviewers and editors determine the author’s original intended meaning.
* To ease the writing process. By making proper use of semantic markup, our authors will get to focus on the job of writing instead of following the rabbit trails of presentation.
* To improve the consistency and speed of generation in the final product. Semantic markup will allow us to get further along the path to a final product programmatically.

To learn a little more about the linguistic field of semantics (a subset of Semiotics), check out the following resources:

* [Wiktionary](http://en.wiktionary.org/wiki/semantics)
* [Wikipedia](http://en.wikipedia.org/wiki/Semantics)
* [Dictionary.com](http://dictionary.reference.com/browse/semantics)
* [Merriam-Webster](http://www.merriam-webster.com/dictionary/semantics)

# Heading 1 Example

Use the Heading 1 paragraph style (keyboard shortcut: Ctrl + Alt + 1) to mark a top level heading. This type of heading is for major sections within the chapter.

It can often be helpful to view the structure of the document as you write and edit a module. Do this by opening the navigation pane.

* Via keyboard shortcut:

1. Alt then w, q.

* Via Ribbon navigation:

1. Go to the View tab, Show group.
2. Toggle the checkbox for Navigation Pane to checked.

## Heading 2 Example

Use the Heading 2 paragraph style (keyboard shortcut: Ctrl + Alt + 2) to mark a second level heading.

### Heading 3 Example

Use the Heading 3 paragraph style (keyboard shortcut: Ctrl + Alt + 3) to mark a third level heading.

#### Heading 4 Example

Use the Heading 4 paragraph style to mark a fourth level heading. You should need only very occasional use of a Heading 4. Lower level headings than Heading 4 (like Heading 5 and beyond) should not be used.

If you find that you need to nest content further than in a Heading 4, you should consider breaking up your module/chapter in to multiple modules.

# Other Important Paragraph Level Styles

Everything should be semantically marked up. That means that many paragraphs you might not otherwise consider are going to need to be marked up.

## Normal Paragraphs

For example, this paragraph is marked up with the Normal paragraph style.

The Normal paragraph style is used because the paragraph is just a normal body paragraph.

## Lists

There are two different basic kinds of lists: ordered and unordered.

### Ordered Lists

Ordered lists are lists in which the order is important. Use ordered lists for sequential steps, a prioritized group of items, etc.

There are unique paragraph styles for each level of nesting:

1. Ordered List 1. Use this for the top level ordered list items.
2. Ordered List 2. Use this for the second level ordered list items.

### Unordered Lists

Unordered lists are lists in which the order is not important. Use unordered lists for groups of like items, a collection of options, etc.

There are unique paragraph styles for each level of nesting:

* Unordered List 1. Use this paragraph style for the top level unordered list items.

Unordered List 2. Use this for second level unordered list items.

Unordered List 3. For the third level items.

### Complex Lists

You can mix ordered lists and unordered lists together by swapping out the equivalent list level.

You can also use a list paragraph of the appropriate level for creating paragraphs of content within a list.

1. Pick a French Cheese

Camembert

Brie de Meaux

Roquefort

Boursin

Reblochon

Munster

Pont l’Évêque

Époisses

Chèvre

Tomme de Savoie

Brie de Bourgogne

1. Eat!

Cheese is pretty good just by itself, but you may want to eat it with some awesome bread too.

Here’s a list of the best breads you could eat your cheese with (in order):

1. Baguette

Baguette de tradition française

Bánh mì

Marraqueta

1. Boule
2. Brioche
3. Crêpe

## Figure Captions

When you are indicating a figure in the flow of your text, use the Figure Caption paragraph style.

1. Write the image caption itself and then write the name of the image file within square brackets. If the image is not a screenshot indicate the source of the image as well. Source: My imagination [non-existent-image.tif]

Check out the screen capture guidelines for more information on screen capturing and working with images in your manuscript.

# Character Styles

We also use character styles to indicate inline level semantic elements.

## Important Term

For important terms that you want to call attention to, use the Important Term character style (keyboard shortcut: Ctrl + i).

You’ll notice Important Terms are used for the title of the styles throughout this document when the term is being defined.

## Code

For keystrokes and code to input use the Code character style. The keyboard shortcuts in this document are all marked up with Code.

## UI Interaction

For buttons to click on and items to select, use the UI Interaction character style (keyboard shortcut: Ctrl + b).

## Cite

Use the Cite character style to mark the title of cited work like A Guide to the Project Management Body of Knowledge (PMBOK®)—Fifth Edition.

## Hyperlink

When inserting a url like http://Advisicon.com use the Hyperlink character style.

## Cross-Reference

Use the Cross-Reference character style to mark a reference to another module or another section within the current module. For example a reference to the Troubleshooting section of this document should be marked up as a Cross-Reference.

# Tables

Use a table for your tabular information.

|  |  |
| --- | --- |
| Column 1 | Column 2 |
| Cell 1, Column 1 | Cell 1, Column 2 |
| Cell 2, Column 1 | Cell 2, Column 2 |
| Cell 3, Column 1 | Cell 3, Column 2 |

# Callouts

Within our books we use four different types of callouts to indicate special information: PMBOK, Notes, Tips, and Warnings. Use these paragraph styles to indicate callout content.

## PMBOK

Use the Callout | PMBOK paragraph style when you are discussing where you can find a concept or term in the Project Management Institute’s Project Management Body of Knowledge – The PMBOK.

This is an example of a PMBOK callout.

## Note

The Callout | Note paragraph style is for indicating a key fact or insight to help participants better understand helpful background information, quirks, explanations for the way things work, answers to frequently asked questions, and helpful things to remember.

Callout boxes like this Note callout will help your readers identify important information. While everything you write is important, try to use callout boxes only for the most important information. Over use of callouts will actually make learning more difficult for your readers.

## Tip

To present quick ways to do things faster and impress colleagues use the Callout | Tip paragraph style.

Keep in mind that you can still use the character styles in any callout box. For example, text marked up with Code can exist within a Tip callout: styled = true unless (c\_style.nil? && p\_style.nil?)

## Warning

When you need to draw attention to important risks, pitfalls, potential issues, and alternate concepts that may assist you with managing your project processes use the Callout | Warning paragraph style.

Carriage returns can allow you to create multiple paragraphs within a single callout. However, the semantic meaning here isn’t as clear as we’d like. Multiple paragraphs may also indicate multiple callouts of the same time in sequence.

Be careful to make sure your meaning is clear.

One less than ideal solution to indicate multiple callouts of the same type is to apply the Normal paragraph style to an empty carriage return between callouts.

# Troubleshooting

If your markup is behaving strangely, turn on the Show/Hide ¶ feature (keyboard shortcut: Ctrl + \*).

1. On the Home tab, Paragraph group, click the ¶ button to toggle hidden character visibility throughout your document.

Impress your typography nerd friends and annoy everybody else by calling the ¶ symbol a “Pilcrow.” This is one of several great names for this symbol. Other names include “paragraph mark,” “paragraph sign,” “paraph,” “alinea,” and “blind P.”

# This Document is Evolving

We are continuously improving our book writing process. As we find better ways to do things, we’ll update this document and provide our authors with a new copy.