

## **ppp reference**

Svein Inge Albrigtsen

Arendal, March 3, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Macros</b>	<b>4</b>
2.1	Main . . . . .	4
2.2	Language . . . . .	5
2.3	Page Layout . . . . .	5
2.4	Fonts . . . . .	5
2.5	Header and Footer . . . . .	6
2.6	Title Page . . . . .	7
2.7	Listings . . . . .	7
2.8	Sectioning . . . . .	7
2.9	Numbering . . . . .	8
2.10	Notes . . . . .	9
2.11	Bibliography . . . . .	9
<b>3</b>	<b>Extensions</b>	<b>10</b>
3.1	Boxes . . . . .	10
3.1.1	Figures . . . . .	10
3.1.2	Tables . . . . .	11
3.1.3	Code Blocks . . . . .	11
3.1.4	Custom . . . . .	11
3.1.5	In-document References . . . . .	12
3.1.6	Attribute Block . . . . .	13
3.2	Numbered References . . . . .	14

# 1 Introduction

ppp is a pandoc preprocessor and is aimed for writing longer academic texts using markdown and adds some small extensions to the markdown syntax aimed to further this goal.

This document will only cover the usage of ppp and its extensions, for information about the pandoc markdown syntax for which it extends, you can see the pandoc documentation available here<sup>1</sup>.

---

<sup>1</sup><http://johnmacfarlane.net/pandoc/README.html>

## 2 Macros

Most notably ppp adds preprocessing macros for including other files and to set document metadata. Macros begin with the character % and must start on a new line, followed by the macro name. Macros taking an argument must be followed by a : and its argument and for macros taking list of arguments the arguments are separated by either ; or a newline.

```
% MacroName : Argument 1; Argument 2
              Argument 3
```

All macro names are case insensitive and ignore leading and trailing spaces.

### 2.1 Main

**Include** Used to include a separate file into the markdown document. This macro accepts a list of arguments and includes all files in order at the point of entry in the markdown document.

**Type** The document type is used to select which document template is used for the document. The Type macro may only occur once in a document and determines what macros and other markdown extensions are available in the rest of the document.

**Report** (Default) ...

**Article** A condensed version of the report document type.

## 2.2 Language

**Language** Sets the language of the document, controls the names of the generated chapter headings, caption markers and terms in the bibliography. Currently supported languages: English (Default), British, American, Norsk, Nynorsk

**Def** Redefine a list of language terms. The syntax is of the form:

`% Def: termname = New Definition`

## 2.3 Page Layout

**Page** ...

- OneSide (Default)
- TwoSide

**PageCols** Sets the number of columns of the document, default is 1.

**PageSize** Sets the size of the document page, values are specified in ISO paper sizes, i.e.: A0, A1, A2, A3, A4, etc. or B0, B1, B2, etc.. Default is A4.

**PageDiv** Default is 10.

**PageBcor** Sets the binding correction for the document. Default is 0mm.

## 2.4 Fonts

**FontSize** Sets the font size of the document. Default is 12pt.

**FontMain** Sets the main font of the document, the font must be available on the system. Default is Open Sans.

**FontSans** Sets the sans font of the document, the font must be available on the system. Default is Open Sans.

**FontMono** Sets the monospace font for the document. Used by verbatim text and code blocs. The font must be available on the system. Default is Ubuntu Mono.

**FontMath** Sets the math font for the document. Used by formulas and equations encapsulated in \$ or \$\$ . The font must be available on the system. Default is selected by  $\TeX$  and usually is either Computer Modern or Latin Modern Math.

## 2.5 Header and Footer

**Header** The header for document pages. Takes a list of 1, 2 or 3 arguments. Depending on the number of arguments the header are positioned as follows:

1. Centre
2. Left & Right
3. Left & Centre & Right

If a setup other than the above is required, for example a single left entry, an empty dud argument of - can be inserted to fill the empty slot.

On two sided documents the order is reversed for even pages, such that left equates to the outer margin of the page.

If no header is defined the separating horizontal line is also removed from the page. The header is empty by default.

**Footer** Sets the footer of the document. The footer macro takes a list of arguments in the same manner as described in the header macro.

Default the footer put page numbers at the centre of the page on one sided documents, and at the outer margins for two sided documents.

If overridden the page numbers may be manually included using the latex function `\pagemark`. For a more fancy pagination, for example of the style *Page X of X*, one can achieve this using: `Page \pagemark of \lastpage`. These latex functions may also be used in the header if desired.

## 2.6 Title Page

- Titlehead** Sets the header of the title page. The title head macro takes a list of arguments in the same manner as described in header macro.
- Subject** Sets the subject of the document.
- Title** Sets the title of the document.
- Subtitle** Sets the subtitle of the document.
- Author** Sets the author(s) of the document. The author macro takes a list arguments.
- Date** Sets the date of the document. The latex function `\today` may be of use here.
- Publisher** Sets the publisher of the document.
- Keywords** Sets the keywords of the document. The keywords macro takes a list of arguments.
- Abstract** Sets the abstract of the document.
- Dedication** TBI

## 2.7 Listings

The listing macros are all inline macros that places the desired list at the point of entry in the document. Only one of each may be present in a document.

- TableOfContents** Generate a table of content.
- ListOfFigures** Generate a list of figures, all images possessing a caption will be included here.
- ListOfTables** Generate a list of tables, all tables possessing a caption will be included here.
- ListOfPrograms** Generate a list of programs, all code blocks possessing a caption will be included here.
- ListOfExamples** Generate a list of examples.

## 2.8 Sectioning

Documents are divided into 4 main parts:

- Frontmatter** The front matter does not number section headings. Every document starts out in this mode so there is no macro called `Frontmatter`.

**Mainmatter** This part numbers section headings. The `Mainmatter` macro may only occur once in a document, and once activated it cannot be deactivated by any other means than to transition to either the appendix or back matter.

**Appendix** This part changes section numbering to alphabetic and prefixes level 1 headings with the language dependant variant of *Appendices*. Under this style all level 1 headings are treated as separate appendices. The `Appendix` macro may only occur once in a document, and once activated it cannot be deactivated by any other means than a transition to the back matter.

The `Appendix` macro may also be used to include external files similar to the `Include` macro. If doing so the `Appendix` part of the document is activated before the first included document.

**Backmatter** This style does not number section headings. The `Backmatter` macro may only occur once in a document and once activated it lasts through the rest of the document.

## 2.9 Numbering

**TocDepth** Sets the depth to which headings are included in the table of contents. The `TocDepth` macro is an inline function and may be used to selectively change parts of the document. Default behaviour is to not include anything before the toc and everything after the toc down to a default header level of 3. If the `TocDepth` macro appear before the `TableOfContents` macro then the toc will not modify the already set depth otherwise the `TableOfContent` macro will set the depth to 3.

**NumDepth** Sets the depth to which headings are numbered in the document. The `NumDepth` macro is an inline function and may be used to selectively change parts of the document. Only valid under the main matter or appendix part of the document. If the macro occurs before the main matter the depth is applied once the main matter is enabled. If it is applied in the back matter the macro is ignored.

**BmkDepth** Sets the depth to which headings are included in the pdf bookmarks. The `BmkDepth` macro is a global macro and may only occur once in the document. Default depth is 3.



## 2.10 Notes

**LinksAsNotes** Changes all links to footnotes, useful for print.

**Notes** Converts all footnotes to endnotes and prints the resulting list at the macros point of entry in the markdown document.

## 2.11 Bibliography

**Bibliography** The Bibliography macro uses pandoc-citeproc to generate a list of bibliographic entries from a database like file. The macro takes a list of arguments, each a bibliographic database file, and prints the bibliography at the point of entry in the markdown file. No header is generated for this list so you may find it useful to precede the macro with a sensible heading.

**CSL** The CSL macro is used to select what style the bibliography is printed in. The macro takes one argument and it must be a valid .csl file. Default stile is IEEE.

## 3 Extensions

### 3.1 Boxes

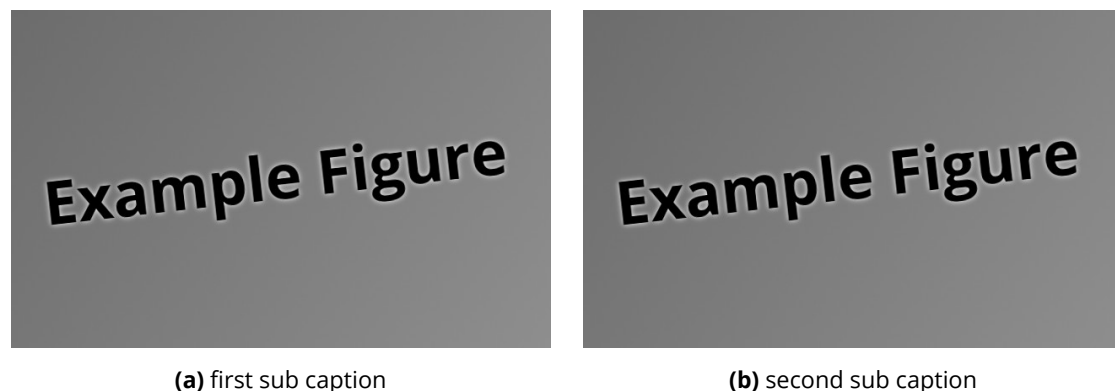
Boxes, traditionally called *floats* in typographical contexts, is a way of placing and grouping elements that should not be split across pages. Figures and tables are the most common examples of such elements.

Traditionally if such an element is placed on a page where there is not enough space for it to fit it might leave half a page empty before appearing at the top of the next page. This is typically an undesired effect as it breaks the flow of text.

Boxes provide a way of positioning the element such that it does not break the flow of text in such a way. This often means that the element will be placed somewhere other than the actual point of entry in the document, but somewhere near where the flow of text is not disrupted.

In ppp all figures, tables and code blocks are wrapped in boxes. These boxed elements will never be split across pages, however, the default behaviour is still absolute positioning at the point of entry in the markdown document, however, the floating behaviour is easily enabled.

#### 3.1.1 Figures



**Figure 3.1:** Example figure

The syntax for figures have been extended to support simple grouping of multiple figures and figure captions. Based on the normal markdown syntax for images the field for alt-text are used for figure captions. This field may be used to specify multiple captions separated by ;.

```
![Example figure; first sub caption; second sub caption](...)
```

The url field of the figure are extended to take a list of urls separated by a space. If the space character is present in the url it may be escaped using the \ character.

```
![...](ex.jpg ex.jpg)
```

At the end of the url field there may also be specified an attribute block. This block is used to set properties to control the behaviour of the box encapsulating the figure. The attribute block will be explained in Section 3.1.6 Attribute Block.

```
![...](... {ATTRIBUTE BLOCK})
```

### 3.1.2 Tables

Table syntax is the same as for normal pandoc tables, except that the table caption is extended to support an attribute block at the end.

### 3.1.3 Code Blocks

Code blocks natively supports attribute blocks in pandoc so this syntax is not extended in any way. However, code blocks do not normally support captions, so to achieve this one may specify a caption attribute in the attribute block.

### 3.1.4 Custom

Custom boxes may be defined using an inline html div. In order for ppp to treat the div as a float it must have the class box. An additional class determines the type of the box which in turn determines the caption prefix and in which listing the element will appear if any. Current supported types are:

Type	Short
figure	fig
table	tab
formula	from
program	prog
example	ex

**Table 3.1:** Box types

Custom boxes were primarily made to be able to group multiple elements together, for example by placing two tables side by side. To accomplish this one may simply wrap the two tables in a box.

```
<div id="my-table" class="table box">
```

```
-----
```

```
a table...
```

```
-----
```

```
-----
```

```
an other table...
```

```
-----
```

```
<span class="caption">My table</span>
```

```
</div>
```

---

```
a table...
```

---



---

```
an other table...
```

---

**Table 3.2:** My table

Custom boxes may be given a caption by adding a span with the class `caption`, in the case above the caption for the custom box, i.e. the wrapper, will be the main caption for the grouping.

### 3.1.5 In-document References

The type of the box also determines the prefix of the in-document reference to the box. All boxes are prefixed with their short names (see Table 3.1 for the full list) followed by `:`. For example the generated reference for the example 'Table 3.2' is `#tab:my-table`.

### 3.1.6 Attribute Block

The attribute block is an element wrapped in curly braces ({}) containing properties to control the behaviour of the box. The syntax for the content of the attribute block are the same as those for pandoc and will not be explained here, for more info see the pandoc documentation<sup>1</sup>.

Class	Description
.float	Enables the floating behaviour for the box.
.span	In a multicolumn document lets the box span the whole page
.wrap	Lets text flow around the document (incompatible with .span)
.long	Overwrites most other classes and tries to make the box span multiple pages if necessary. Currently only works for tables and code blocks.

**Table 3.3:** Box classes

**Table 3.4:** Box attributes

Attribute	Option	Description
wrap		determines the alignment of a wrapping box (requires .wrap)
	left	floats the box to the left of the text
	right	floats the box to the right of the text
	outer	(default) floats the box towards the outer margins of the page
width	inner	floats the box towards the inner margins of the page
		Determines the width of the box in relation to the page. In a multicolumn document the percentage is in relation to the column unless .span is specified then it is in relation to the page. May be specified as a percentage e.g.: "50%" or as a ratio e.g.: "0.5".
style		Set the style of the box
	plain	(Default) Renders the box without any borders and with captions placed at the bottom.
	plaintop	Renders the box without any borders and with captions placed at the top.
	boxed	Renders the box with border and captions placed on the bottom, underneath the border.

<sup>1</sup><http://johnmacfarlane.net/pandoc/README.html#header-identifiers-in-html-latex-and-context>

Attribute	Option	Description
tablestyle	ruled	Renders the box with borders on the top and bottom with the caption placed above with an additional border above the caption as well.
		Sets the style of a table, only available for tables not using .long.
align	plain	Renders the table without any rules or borders
	ruled	(Default) Renders the table with a top, middle and bottom rule.
		Determines the alignment of the inner content of the box. This attribute takes two arguments separated by a space character. First argument to specify the vertical alignment and the second to specify the horizontal alignment.
	top	Vertical
	centre / center	Vertical
	bottom	Vertical
	left	Horizontal
	centre / center	Horizontal
	justified	Horizontal
	right	Horizontal

## 3.2 Numbered References

This extension extends the syntax for in-document links and allows auto generated numbers to be included in the link itself. The extension will replace the character # with the generated number for the referenced section heading, figure, table, etc..

For example to reference this section you may write:

See [Section # Numbered References](#numbered-references).

and ppp will generate:

See Section 3.2 Numbered References.