Spine, Doc Reform
- SiSU Markup

Ralph Amissah

2008-05-22
Copyright © 2007 Ralph Amissah

GPL 3 (part of SiSU documentation)
Spine, Doc Reform - SiSU Markup
Ralph Amissah
## SiSU Markup

### 1. Introduction to SiSU Markup

1.1. Summary

1.2. Markup Rules, document structure and metadata requirements

1.3. Markup Examples

#### 1.3.1. Online

### 2. Markup of Headers

2.1. Sample Header

2.2. Available Headers

### 3. Markup of Substantive Text

3.1. Heading Levels

3.2. Font Attributes

3.3. Indentation and bullets

3.4. Hanging Indents

3.5. Footnotes / Endnotes

3.6. Links

#### 3.6.1. Naked URLs within text, dealing with urls

#### 3.6.2. Linking Text

#### 3.6.3. Linking Images

#### 3.6.4. Link shortcut for multiple versions of a sisu document in the same directory tree

3.7. Grouped Text / blocked text

#### 3.7.1. blocked text curly brace syntax
SiSU Markup
1. Introduction to SiSU Markup

1.1. Summary

This is the D version of the program sisu on which the markup it uses is based. SiSU source documents are plaintext (UTF-8) files. All paragraphs are separated by an empty line.

Markup is comprised of: markup-summary

1. at the top of a document, the document header made up of semantic meta-data about the document and if desired additional processing instructions (such an instruction to automatically number headings from a particular level down)

2. followed by the prepared substantive text of which the most important single characteristic is the markup of different heading levels, which define the primary outline of the document structure. Markup of substantive text includes:

- heading levels defines document structure
- text basic attributes, italics, bold etc.
- grouped text (objects), which are to be treated differently, such as code blocks or poems.
- footnotes/endnotes
- linked text and images
- paragraph actions, such as indent, bulleted, numbered-lists, etc.

1.2. Markup Rules, document structure and metadata requirements

minimal content/structure requirement, minimum being:

metadata

title: "SiSU Spine"
subtitle: "Markup"
creator: 
author: "Amissah, Ralph"

levels

A~ (level A [title])
L~ (at least one level 1 [segment/(chapter)])

structure rules (document heirarchy, heading levels):

there are two sets of heading levels ABCD (title & parts if any) and 123 (segment &

---

1From sometime after SiSU 0.58 it should be possible to describe SiSU markup using SiSU, which
sisu has the following levels (that may be described as document parts, headings and subheadings):

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A~</td>
<td>[title (&amp; author)]</td>
</tr>
<tr>
<td></td>
<td>- document root, required once (== 1)</td>
</tr>
<tr>
<td></td>
<td>- followed by part B~ or level 1~</td>
</tr>
<tr>
<td></td>
<td>- often written in the form:</td>
</tr>
<tr>
<td></td>
<td>A~ @title @creator</td>
</tr>
<tr>
<td></td>
<td>where title and creator are taken from the document header</td>
</tr>
<tr>
<td>B~</td>
<td>[part]</td>
</tr>
<tr>
<td></td>
<td>- part B is followed by a part C~ if there is one or level 1~</td>
</tr>
<tr>
<td>C~</td>
<td>[subpart]</td>
</tr>
<tr>
<td></td>
<td>- part C is followed by a part D~ if there is one or level 1~</td>
</tr>
<tr>
<td>D~</td>
<td>[subsubpart]</td>
</tr>
<tr>
<td></td>
<td>- part D is followed by level 1~</td>
</tr>
<tr>
<td>1~</td>
<td>[heading, segment (chapter)]</td>
</tr>
<tr>
<td></td>
<td>- level 1 required at least once (&gt;= 1)</td>
</tr>
<tr>
<td></td>
<td>- is followed by level 2~ or</td>
</tr>
<tr>
<td></td>
<td>by text which can then be followed</td>
</tr>
<tr>
<td></td>
<td>- by more text or by levels 1~ or 2~ (or relevant part)</td>
</tr>
<tr>
<td></td>
<td>- level 1 in html (and epub) is the basis of a document segment and in a book</td>
</tr>
<tr>
<td></td>
<td>would correspond to a chapter</td>
</tr>
<tr>
<td>2~</td>
<td>[sub-heading]</td>
</tr>
<tr>
<td></td>
<td>- followed by level 3~ or</td>
</tr>
<tr>
<td></td>
<td>by text which can then be followed</td>
</tr>
<tr>
<td></td>
<td>by more text or by levels 1~, 2~ or 3~ (or relevant part)</td>
</tr>
<tr>
<td>3~</td>
<td>[sub-sub-heading]</td>
</tr>
<tr>
<td></td>
<td>- followed by text which can be followed</td>
</tr>
<tr>
<td></td>
<td>by more text or by levels 1~, 2~ or 3~ (or relevant part)</td>
</tr>
</tbody>
</table>

Rules:

- Level A~ is mandatory, it is the (document root and) title
- There can only be one document root == level/part A~
- Heading levels B,C,D, are optional and there may be several of each |
  (where all three are used corresponding to e.g. Book, Part, Section) |
  - sublevels that are used must follow each other sequentially |
  (alphabetically),
- Heading levels A~ B~ C~ D~ are followed by other heading levels rather |
  than substantive text |
  - which may be the subsequent sequential (alphabetic) heading part level |
  - or a heading (segment) level 1~
- There must be at least one heading (segment) level 1~ |
  (the level on which the text is segmented, in a book would correspond |
  to the Chapter level)
- Additional heading levels 1~ 2~ 3~ are optional and there may be several |
  of each
- Heading levels 1~ 2~ 3~ are followed by text (which may be followed by |

though not an original design goal is useful. |

4 Files should be prepared using UTF-8 character encoding.
1.3. Markup Examples

1.3.1. Online

Markup examples are available in the form of prepared texts that were written under creative commons license that permit re-publication.

There is of course this document, which is provided with the program and provides a cursory overview of sisu markup. Running sisu spine against it gives an overview of the output produced by the program.
2. Markup of Headers

The document header is based on yaml, and is the part of the document preceeding the document root marked by "A [Document title & author]"

The document header contains either: semantic meta-data about the document, or processing instructions.

Note: the first line of a document may include information on the markup version used in the form of a comment. Comments within the header section are the hash symbol at the start of a line (and as the first character in a line of text) followed by a space and the comment:

```
# in the header section of a document, this would be a comment
```

### 2.1. Sample Header

This current document is loaded by a master document that has a header similar to this one:

```yaml
# SiSU 8.0

title:
  main: "SiSU"
  subtitle: "Markup"

creator:
  author: "Amissah, Ralph"

date:
  created: "2002-08-28"
  issued: "2002-08-28"
  available: "2002-08-28"
  published: "2008-05-22"
  modified: "2020-04-11"

rights:
  copyright: "Copyright (C) Ralph Amissah 2007, 2020"
  license: "AGPL 3 (part of SiSU Spine documentation)"

classify:
  subject = "ebook, epublishing, electronic book, electronic publishing, electronic document, electronic citation, data structure, citation systems, search"
```

Looking back a bit:

```yaml
# SiSU master 8.0

title:
  main: "SiSU"
  subtitle: "Markup"

creator:
  author: "Amissah, Ralph"
```
2.2. Available Headers

Header tags appear at the beginning of a document and provide meta information on the document (such as the Dublin Core), or information as to how the document as a whole is to be processed. All header instructions take the form headername: or on the next line and indented by two spaces subheadername: All Dublin Core meta tags are available

@identifier: information or instructions

where the “identifier” is a tag recognised by the program, and the “information” or “instructions” belong to the tag/identifier specified

Note: a header where used should only be used once; all headers apart from [title] are optional; the [structure] header is used to describe document structure, and can be useful to know.

This is a sample header

# SiSU 8.0

title:
  main: "SiSU"
  subtitle: "Markup"
  language: "English"
| creator: | [Lastname, First names] |
| author: | [Lastname, First names] |
| illustrator: | [Lastname, First names] |
| translator: | [Lastname, First names] |
| prepared_by: | [Lastname, First names] |

| date: | created: [year or yyyy-mm-dd] |
| issued: [year or yyyy-mm-dd] |
| available: [year or yyyy-mm-dd] |
| published: [year or yyyy-mm-dd] |
| modified: [year or yyyy-mm-dd] |
| valid: [year or yyyy-mm-dd] |
| added_to_site: [year or yyyy-mm-dd] |
| translated: [year or yyyy-mm-dd] |

| rights: | copyright: "Copyright (C) [Year and Holder]" |
| license: "[Use License granted]" |
| text: "[Name, Year]" |
| translation: "[Name, Year]" |
| illustrations: "[Name, Year]" |

# check rest

| subject: "ebook, epublishing, electronic book, electronic publishing, electronic document, electronic citation, data structure, citation systems, search" |
| keywords: "list" |
| loc: "[Library of Congress classification]" |
| dewey: "[Dewey classification]" |

| identifier: | isbn: "[ISBN]" |
| oclc: "" |

| links: | "{SISU }https://www.sisudoc.org", |
| "{FSF }https://www.fsf.org", |

| make: | auto_num_top_at_level: "1" |
| substitute: | [
| [ "[#12](\sisudoc\")", "www.sisudoc.org" ] |
| ] |
| bold: "Debian|SiSU" # [regular expression of words/phrases to be made bold] |
| italics: "Linux|GPL|LaTeX|SQL" # [regular expression of words/phrases to italicise] |
| breaks: "new=8; break=1" |
| home_button_text: "(SISU)https://sisudoc.org; (sources / git)https://git.sisudoc.org/\gitweb/" |
| footer: "(SISU)https://sisudoc.org; (git)https://git.sisudoc.org" |
headings: text to match for each level
(e.g. PART; Chapter; Section; Article; or another: none; BOOK|FIRST|SECOND; none; ←
CHAPTER;)

3. Markup of Substantive Text

3.1. Heading Levels

Heading levels are :A ,:B ,:C ,1 ,2 ,3 ... :A - :C being part / section headings, followed by other heading levels, and 1 -6 being headings followed by substantive text or sub-headings. :A usually the title :A ? conditional level 1 heading (used where a stand-alone document may be imported into another)

:A [heading text] Top level heading [this usually has similar content to the title [title] ] NOTE: the heading levels described here are in 0.38 notation, see heading

:B [heading text] Second level heading [this is a heading level divider]

:C [heading text] Third level heading [this is a heading level divider]

1 [heading text] Top level heading preceding substantive text of document or sub-heading 2, the heading level that would normally be marked 1. or 2. or 3. etc. in a document, and the level on which sisu by default would break html output into named segments, names are provided automatically if none are given (a number), otherwise takes the form 1 my_filename_for_this_segment

2 [heading text] Second level heading preceding substantive text of document or sub-heading 3, the heading level that would normally be marked 1.1 or 1.2 or 1.3 or 2.1 etc. in a document.

3 [heading text] Third level heading preceding substantive text of document, that would normally be marked 1.1.1 or 1.1.2 or 1.2.1 or 2.1.1 etc. in a document

filename level 1 heading,
% the primary division such as Chapter that is followed by substantive text, and may be ← further subdivided (this is the level on which by default html segments are made)

3.2. Font Attributes

markup example:

normal text, *(emphasis)*, ![bold text]!, /(italics)/, -(underscore), "(citation)“, ^(superscript)“, +(subscript)“, +(inserted text), -(strikethrough)“, #{monospace}#

normal text

*(emphasis)* [note: can be configured to be represented by bold, italics or underscore]

![bold text]!

/(italics)/

-(underscore)_

"(citation)"

*{(superscript)}*
resulting output:

normal text, **emphasis**, *bold text*, _italics_, _underscore_, "{citation}", `superscript`, `subscript`, `inserted text`, `strikethrough`, `monospace`

**emphasis** [note: can be configured to be represented by bold, italics or underscore]

*italics*

_underscore_

"{citation}"

`superscript`

`subscript`

`inserted text`

`strikethrough`

`monospace`

### 3.3. Indentation and bullets

markup example:

```
ordinary paragraph
_1 indent paragraph one step
_2 indent paragraph two steps
_9 indent paragraph nine steps
```

resulting output:

ordinary paragraph

indent paragraph one step

indent paragraph two steps

indent paragraph nine steps

markup example:
resulting output:

bullet text
  bullet text, first indent
    bullet text, two step indent

Numbered List (not to be confused with headings/titles, (document structure))

markup example:

1 numbered list numbered list 1., 2., 3, etc.
_1# numbered list numbered list indented a., b., c., d., etc.

3.4. Hanging Indents

markup example:

_0_1 first line no indent (no hang),
rest of paragraph indented one step
_1_0 first line indented,
rest of paragraph no indent
in each case level may be 0-9

resulting output:

first line no indent, rest of paragraph indented one step; first line no indent, rest
of paragraph indented one step; first line no indent, rest of paragraph indented
one step; first line no indent, rest of paragraph indented one step; first line no inden-
t, rest of paragraph indented one step; first line no indent, rest of paragraph indented
one step; first line no indent, rest of paragraph indented one step; first line no indent,
rest of paragraph indented one step; first line no indent, rest of paragraph indented
one step; first line no indent, rest of paragraph indented one step; first line no indent,
rest of paragraph indented one step; first line no indent, rest of paragraph indented
one step; first line no indent, rest of paragraph indented one step;

A regular paragraph.

first line indented, rest of paragraph no indent first line indented, rest of para-
graph no indent first line indented, rest of paragraph no indent first line indented,
rest of paragraph no indent first line indented, rest of paragraph no indent first line
indented, rest of paragraph no indent first line indented, rest of paragraph no indent
first line indented, rest of paragraph no indent first line indented, rest of paragraph
no indent first line indented, rest of paragraph no indent first line indented, rest of
paragraph no indent
live-build  A collection of scripts used to build customized Debian Livesystems.  
    live-build was formerly known as live-helper, and even earlier known as live-package.

live-build  
    A collection of scripts used to build customized Debian Livesystems. live-build 
    was formerly known as live-helper, and even earlier known as live-package.

3.5. Footnotes / Endnotes

Footnotes and endnotes are marked up at the location where they would be indicated within a text. They are automatically numbered. The output type determines whether footnotes or endnotes will be produced.

**markup example:**

~{ a footnote or endnote }~

**resulting output:**

3

**markup example:**

normal text~{ self contained endnote marker & endnote in one }~ continues

**resulting output:**

normal text continues

**markup example:**

normal text ~(+ unnumbered asterisk footnote/endpoint, insert multiple asterisks if required ←~
    continues

normal text ~(** another unnumbered asterisk footnote/endpoint )~ continues

**resulting output:**

normal text * unnumbered asterisk footnote/endpoint, insert multiple asterisks if required continues

normal text ** another unnumbered asterisk footnote/endpoint continues

**markup example:**

\[3\] a footnote or endnote

\[4\] self contained endnote marker & endnote in one
standard (inline) and pair (binary) notation could not be mixed in the same document.

The reason binary notation was provided as an option was for the conversion of documents to sisu markup. Many documents were prepared in such a way that endnotes had been previously marked up in a binary fashion, and this provided a convenient and faster way to make the document conversion, just reflect those markup practices. The reason it has been dropped is it adds a slowing step to something that needs to be done at most once and it prove to be flakey, unnecessarily so even when kept under version control. It is preferable to do a two step conversion of the previously marked up document to sisu: first to the binary/paired footnote markup, then; convert it to the proper form of inline endnote markup with a dedicated helper conversion program, keeping the resulting properly marked up text.

### 3.6. Links

#### 3.6.1. Naked URLs within text, dealing with urls

urls found within text are marked up automatically. A url within text is automatically hyperlinked to itself and by default decorated with angled braces, unless they are contained within a code block (in which case they are passed as normal text), or escaped by a preceding underscore (in which case the decoration is omitted).

**markup example:**

normal text https://www.sisudoc.org/ continues

**resulting output:**

normal text 「https://www.sisudoc.org/」 continues
An escaped url without decoration

**markup example:**

| normal text _https://www.sisudoc.org/ continues |
| deb _https://www.jus.uio.no/sisu/archive unstable main non-free |

**resulting output:**

normal text _https://www.sisudoc.org/ continues

deb _https://www.jus.uio.no/sisu/archive unstable main non-free

where a code block is used there is neither decoration nor hyperlinking, code blocks are discussed later in this document

**resulting output:**

| deb https://www.jus.uio.no/sisu/archive unstable main non-free |
| deb-src https://www.jus.uio.no/sisu/archive unstable main non-free |

### 3.6.2. Linking Text

To link text or an image to a url the markup is as follows

**markup example:**

| about { SiSU }https://url.org markup |

**resulting output:**

about

```
   「 SiSU 」 markup
```

a couple of test urls

```
   「https://example.com/Alice&Bob」
```

```
   「 programs I use」
```

A shortcut notation is available so the url link may also be provided automatically as a footnote

**markup example:**

| about {~^ SiSU }https://url.org markup |

**resulting output:**

about
SiSU\textsuperscript{5} markup

Internal document links to a named (anchor) tagged location, including named headings named inline anchor tags an-inline-anchor-tag or an ocn the heading:

```
\texttt{markup} \texttt{Markup}
```

can be linked to as follows:

```
to find out more see \{ Markup \}\#markup
```

to find out more see Markup

an inline anchor tag is made with the following markup internal-links

```
\texttt{named inline anchor tags \texttt{\^\texttt{-}}an-inline-anchor-tag}
```

and linked to the same way

```
the link \{ an inline anchor tag \}\#an-inline-anchor-tag
```

the link an inline anchor tag or to another part of the document: markup summary

**markup example:**

```
about \{ text links \}\#link_text
```

**resulting output:**

about text links

Shared document collection link

**markup example:**

```
about \{ SiSU book markup examples \}::Sisu/examples.html
```

**resulting output:**

about \{ SiSU book markup examples \}::Susu/examples.html

### 3.6.3. Linking Images

**markup example:**

\textsuperscript{5} \textsuperscript{5} https://www.sisudoc.org/
"Gnu/Linux - a better way"

"Way Better - with Gnu/Linux, Debian and Ruby"
"Ruby"\textsuperscript{6}

"D for me"

"D, hey no fair"\textsuperscript{7}

**linked url footnote shortcut**

\{'\textsuperscript{\texttt{\textasciitilde\textsuperscript{\texttt{}}}} [text to link] } https://url.org

% maps to: { [text to link] }https://url.org ~( https://url.org )~

% which produces hyper-linked text within a document/paragraph, with an endnote providing \textsuperscript{←}
the url for the text location used in the hyperlink

**text marker **~*name*

note at a heading level the same is automatically achieved by providing names to headings 1, 2 and 3 i.e. 2 [name] and 3 [name] or in the case of auto-heading numbering, without further intervention.

\textsuperscript{6} \url{https://www.ruby-lang.org/en/}
\textsuperscript{7} \url{https://github.com/dlang-community/d-mans}
3.6.4. Link shortcut for multiple versions of a sisu document in the same directory tree

markup example:

\[
!_ , ("Viral Spiral")/ , David Bollier
{ "Viral Spiral", David Bollier [3sS]}\text{viral_spiral.david_bollier.sst}
\]

"Viral Spiral", David Bollier

\{ "Viral Spiral", David Bollier [3sS]}\text{viral_spiral.david_bollier.sst}

3.7. Grouped Text / blocked text

There are two markup syntaxes for blocked text, using curly braces or using tics

3.7.1. blocked text curly brace syntax

at the start of a line on its own use name of block type with an opening curly brace, follow with the content of the block, and close with a closing curly brace and the name of the block type, e.g.

\[
\text{code}\{
\text{this is a code block}
\}\text{code}
\]

\[
\text{poem}\{
\text{this here is a poem}
\}\text{poem}
\]

3.7.2. blocked text tic syntax

start a line with three backtics, a space followed by the name of the name of block type, follow with the content of the block, and close with three back ticks on a line of their own, e.g.

\[
```
\text{code}
\text{this is a code block}
```
```
\text{poem}
\text{this here is a poem}
```
\]
3.7.3. Group

The “group” is different from the “block” mark in that “group” does not preserve whitespace, the “block” mark does. The text falling within the block is a single object.

**basic markup:**

```markdown
group{
   Your grouped text here
}group
```

A group is treated as an object and given a single object number.

**resulting group text output:**

“Fury said to a mouse, That he met in the house, “Let us both go to law: I will prosecute YOU. —Come, I’ll take no denial; We must have a trial: For really this morning I’ve nothing to do.” Said the mouse to the cur, “Such a trial, dear Sir, With no jury or judge, would be wasting our breath.” “I’ll be judge, I’ll be jury,” Said cunning old Fury: “I’ll try the whole cause, and condemn you to death.”

**resulting group text output:**

The Road Not Taken Related Poem Content Details BY ROBERT FROST Two roads diverged in a yellow wood, And sorry I could not travel both And be one traveler, long I stood And looked down one as far as I could To where it bent in the undergrowth; Then took the other, as just as fair, And having perhaps the better claim, Because it was grassy and wanted wear; Though as for that the passing there Had worn them really about the same, And both that morning equally lay In leaves no step had trodden black. Oh, I kept the first for another day! Yet knowing how way leads on to way, I doubted if I should ever come back. I shall be telling this with a sigh Somewhere ages and ages hence: Two roads diverged in a wood, and I I took the one less traveled by, And that has made all the difference.

3.7.4. Block

The “block” is different from the “group” mark in that the “block” mark (like the “poem” mark) preserves whitespace, the “group” mark does not. The text falling within the “block” is a single object, which is different from the “poem” mark where each identified verse is an object.

**basic markup:**

```markdown
block{
   Your block text here
}block
```

A block is treated as an object and given a single object number.

**resulting block text output:**

*endnote test
stress test
'Fury said to a
mouse, That he
met in the
house,
"Let us
both go to
law: I will
prosecute
YOU. –Come,
I’ll take no
denial; We
must have a
trial: For
really this
morning I’ve
nothing
to do."
Said the
mouse to the
cur, "Such
a trial,
dear Sir,
With
no jury
or judge,
would be
wasting
our
breath."\(^{10}\)
"I’ll be
judge, I’ll
be jury,"
Said
cunning
old Fury:
"I’ll
try the
whole\(^{11}\)
cause,
and
condemn
you
to
death."

**curly brace delimiter, resulting block text output:**

The Road Not Taken Related Poem Content Details

\(^{10}\)endnote test

\(^{11}\)stress test
BY ROBERT FROST

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I
I took the one less traveled by,
And that has made all the difference.

3.7.5. Poem

The “poem” mark like the “block” preserves whitespace. Text followed by two newlines are identified as verse and each verse is an object i.e. a poem may consist of multiple verse each of which is identified as an object, unlike a text “block” which is identified as a single object.

**basic markup:**

```plaintext
poem{
    Your poem here
}
```

Each verse in a poem is given an object number.

curly brace delimiter, resulting poem text output (broken into verse):

‘Fury said to a mouse, That he met in the
"Let us both go to law: I will prosecute YOU. –Come, I'll take no denial; We must have a trial: For really this morning I've nothing to do."

Said the mouse to the cur, "Such a trial, dear Sir, With no jury or judge, would be wasting our breath."

"I'll be judge, I'll be jury," Said cunning old Fury: "I'll try the whole cause, and condemn you to death."

---

curly brace delimiter, resulting poem text output (broken into verse):

**The Road Not Taken**

by Robert Frost

Two roads diverged in a yellow wood, And sorry I could not travel both And be one traveler, long I stood And looked down one as far as I could To where it bent in the undergrowth;

---

\[sup\]12\[sup\] published in 1916 as the first poem in the collection Mountain Interval.
Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I
I took the one less traveled by,
And that has made all the difference.

The Road Not Taken

by Robert Frost

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I
I took the one less traveled by,
And that has made all the difference.

3.7.6. Code

“Code” blocks are a single text object, in which the original text is preserved. Code tags code{ ... }code (used as with other group tags described above) are

\(^{13}\)published in 1916 as the first poem in the collection Mountain Interval.
used to escape regular sisu markup, and have been used extensively within this
document to provide examples of SiSU markup. You cannot however use code tags
to escape code tags. They are however used in the same way as group or poem
tags.

A code-block is treated as an object and given a single object number. [an option
to number each line of code may be considered at some later time]

**use of code tags instead of poem compared, resulting output:**

```
'Fury said to a
mouse, That he
met in the
house,
"Let us
both go to
law: I will
prosecute
YOU. "--Come,
I'll take no
denial; We
must have a
trial: For
really this
morning I've
nothing
to do."
Said the
mouse to the
cur, "Such
a trial,
dear Sir,
With
no jury
or judge,
would be
wasting
our
breath."
"I'll be
judge, I'll
be jury,"
Said
cunning
old Fury:
"I'll
try the
whole
cause,
and
condemn
you
to
death."
```

From SiSU 2.7.7 on you can number codeblocks by placing a hash after the
opening code tag `code{#` as demonstrated here:

```
'Fury said to a
mouse, That he
met in the
house,
"Let us
both go to
```
law: I will prosecute
YOU. "Come,
I'll take no
denial; We
must have a
trial: For
really this
morning I've
nothing
to do."
Said the
mouse to the
cur, "Such
a trial,
dear Sir,
With
no jury
or judge,
would be
wasting
our
breath."
"I'll be
judge, I'll
be jury,"
Said
cunning
old Fury:
"I'll
try the
whole
cause,
and
condemn
you
to
death."

3.7.7. Tables

Tables may be prepared in two either of two forms

markup example:

```latex
\begin{table}[c]
\begin{tabular}{ccc}
  40 & 30 & 30 \\
\end{tabular}
\end{table}
```

resulting output:

```
| 40 | 30 | 30 |
```
Same as a tic table

This is a table
And here begins another row

Without instruction

This is a table
And here begins another row

A second form may be easier to work with in cases where there is not much information in each column

**markup example:**

![Table 3.1: Contributors to Wikipedia, January 2001 - June 2005](https://www.jus.uio.no/sisu/the_wealth_of_networks.yochai_benkler)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributors*</td>
<td>10</td>
<td>472</td>
<td>2,188</td>
<td>9,653</td>
<td>25,011</td>
<td>48,721</td>
</tr>
<tr>
<td>Active contributors**</td>
<td>9</td>
<td>212</td>
<td>846</td>
<td>3,228</td>
<td>8,442</td>
<td>16,945</td>
</tr>
<tr>
<td>Very active contributors***</td>
<td>0</td>
<td>31</td>
<td>190</td>
<td>692</td>
<td>1,639</td>
<td>3,016</td>
</tr>
<tr>
<td>No. of English language articles</td>
<td>25</td>
<td>16,000</td>
<td>101,000</td>
<td>190,000</td>
<td>320,000</td>
<td>630,000</td>
</tr>
<tr>
<td>No. of articles, all languages</td>
<td>25</td>
<td>19,000</td>
<td>138,000</td>
<td>490,000</td>
<td>862,000</td>
<td>1,600,000</td>
</tr>
</tbody>
</table>

* Contributed at least ten times; ** at least 5 times in last month; *** more than 100 times in last month.

---

### 3.8. Additional breaks - line-breaks within objects, column and page-breaks

#### 3.8.1. line-breaks

To break a line within a "paragraph object", two backslashes with a space before and a space or newline after them may be used.

---

14Table from the Wealth of Networks by Yochai Benkler

https://www.jus.uio.no/sisu/the_wealth_of_networks.yochai_benkler
To break a line within a “paragraph object”, two backslashes `\` with a space before and a space or newline after them `\` may be used.

The html break `br` enclosed in angle brackets (though undocumented) is available in versions prior to 3.0.13 and 2.9.7 (it remains available for the time being, but is depreciated).

To draw a dividing line dividing paragraphs, see the section on page breaks.

### 3.8.2. page breaks

Page breaks are only relevant and honored in some output formats. A page break or a new page may be inserted manually using the following markup on a line on its own:

- `page new ==` breaks the page, starts a new page.
- `page break --` breaks a column, starts a new column, if using columns, else breaks the page, starts a new page.
- `page break line across page -..-` draws a dividing line, dividing paragraphs
- `page break:

```
-\-\--
```

- `page (break) new:

```
=\=\=
```

- `page (break) line across page (dividing paragraphs):

```
-..-
```

### 3.9. Excluding Object Numbers

Object numbers can be switched off by adding a `#` to the end of a text object.

Sometimes it is wished to switch off object numbers for a larger group of text. In this case it is possible before the group, body of text to be without object numbers on a new line with nothing else on it to open the un-numbered object block with `- #` and to close the un-numbered block, and restart object numbering with on a similarly otherwise empty new-line with `-+#`
3.10. Bibliography / References

There are three ways to prepare a bibliography using sisu (which are mutually exclusive): (i) manually preparing and marking up as regular text in sisu a list of references, this is treated as a regular document segment (and placed before endnotes if any); (ii) preparing a bibliography, marking a heading level 1 !biblio (note the exclamation mark) and preparing a bibliography using various metadata tags including for author: title: year: a list of which is provided below, or; (iii) as an assistance in preparing a bibliography, marking a heading level 1 !biblio and tagging citations within footnotes for inclusion, identifying citations and having a parser attempt to extract them and build a bibliography of the citations provided.

For the heading/section sequence: endnotes, bibliography then book index to occur, the name biblio or bibliography must be given to the bibliography section, like so:

1~!biblio

3.10.1. a markup tagged metadata bibliography section

Here instead of writing your full citations directly in footnotes, each time you have new material to cite, you add it to your bibliography section (if it has not been added yet) providing the information you need against an available list of tags (provided below).

The required tags are au: ti: and year: an short quick example might be as follows:

1~!biblio
au: von Hippel, E.
ti: Perspective: User Toolkits for Innovation
lng: (language)
jo: Journal of Product Innovation Management
vo: 18
ed: (editor)
yr: 2001
note:

for which you may alternatively use the full form author: title: and year:
Note that the section name !biblio (or !bibliography) is required for the bibliography to be treated specially as such, and placed after the auto-generated endnote section.

Using this method, work goes into preparing the bibliography, the tags author or editor, year and title are required and will be used to sort the bibliography that is placed under the Bibliography section.

The metadata tags may include shortname (sn:) and id, if provided, which are used for substitution within text. Every time the given id is found within the text it will be replaced by the given short title of the work (it is for this reason the short title has sisu markup to italicize the title), it should work with any page numbers to be added, the short title should be one that can easily be used to look up the full description in the bibliography.

The following footnote—{ quixote1605, pp 1000 - 1001, also Benkler2006 p 1. }—

would be presented as:


---

3.10.2. Tagging citations for inclusion in the Bibliography

Here whenever you make a citation that you wish be included in the bibliography, you tag the citation as such using special delimiters (which are subsequently removed from the final text produced by sisu).

Here you would write something like the following, either in regular text or a footnote:

```
See : Quixote, Don; Panza, Sancho /{ Taming Windmills, Keeping True }/ (1605) .
```

SiSU will parse for a number of patterns within the delimiters to try make out the authors, title, date etc. and from that create a Bibliography. This is more limited than the previously described method of preparing a tagged bibliography, and using an id within text to identify the work, which also lends itself to greater consistency.

3.11. Glossary

Using the section name `!glossary` results in the Glossary being treated specially as such, and placed after the auto-generated endnote section (before the bibliography/list of references if there is one).

The Glossary is ordinary text marked up in a manner deemed suitable for that purpose. e.g. with the term in bold, possibly with a hanging indent.

```
* _0_1 *(GPL)* An abbreviation that stands for "General Purpose License." ... _0_1 [provide your list of terms and definitions]
```

In the given example the first line is not indented subsequent lines are by one level, and the term to be defined is in bold text.
3.12. Book index

To make an index append to paragraph the book index term relates to it, using an equal sign and curly braces.

Currently two levels are provided, a main term and if needed a sub-term. Sub-terms are separated from the main term by a colon.

```
Paragraph containing main term and sub-term.
= {Main term : sub-term}
```

The index syntax starts on a new line, but there should not be an empty line between paragraph and index markup.

The structure of the resulting index would be:

```
Main term, 1
 sub-term, 1
```

Several terms may relate to a paragraph, they are separated by a semicolon. If the term refers to more than one paragraph, indicate the number of paragraphs.

```
Paragraph containing main term, second term and sub-term.
= {first term ; second term : sub-term}
```

The structure of the resulting index would be:

```
First term, 1,
 Second term, 1,
 sub-term, 1
```

If multiple sub-terms appear under one paragraph, they are separated under the main term heading from each other by a pipe symbol.

```
Paragraph containing main term, second term and sub-term.
= {Main term:
     sub-term=2|second sub-term;
     Another term
   }

A paragraph that continues discussion of the first sub-term
```

The plus one in the example provided indicates the first sub-term spans one additional paragraph. The logical structure of the resulting index would be:

```
Main term, 1,
 sub-term, 1-3,
 second sub-term, 1,
 Another term, 1
```
4. Composite documents markup

It is possible to build a document by creating a master document that requires other documents. The documents required may be complete documents that could be generated independently, or they could be markup snippets, prepared so as to be easily available to be placed within another text. If the calling document is a master document (built from other documents), it should be named with the suffix `.ssm`. Within this document you would provide information on the other documents that should be included within the text. These may be other documents that would be processed in a regular way, or markup bits prepared only for inclusion within a master document `.sst` regular markup file, or `.ssi` (insert/information) A secondary file of the composite document is built prior to processing with the same prefix and the suffix `.sst` basic markup for importing a document into a master document

```
<< filename1.sst
<< filename2.ssi
```

The form described above should be relied on. Within the Vim editor it results in the text thus linked becoming hyperlinked to the document it is calling in which is convenient for editing.
5. Substitutions

markup example:

```
The current Debian is ${debian_stable} the next debian will be ${debian_testing}
Configure substitution in _sisu/sisu_document_make
@make:
  -substitute: /${debian_stable}/,'*{Wheezy}*'/${debian_testing}/,'*{Jessie}*'
```

resulting output:

The current Debian is ${debian_stable} the next debian will be
${debian_testing}
Another test ${sisudoc} ok?
Configure substitution in _sisu/sisu_document_make
6. Footnote, endnote stress test

Globalisation is to be observed as a trend intrinsic to the world economy. Rudimentary economics explains this runaway process, as being driven by competition within the business community to achieve efficient production, and to reach and extend available markets. Technological advancement particularly in transport and communications has historically played a fundamental role in the furthe...
competing schools of law and economics, with different perspectives, levels of abstraction, and analytical consequences of and for the world that they model.\footnote{For a good introduction see Nicholas Mercuro and Steven G. Medema, \textit{Economics and the Law: from Posner to Post-Modernism} (Princeton, 1997). These include: Chicago law and economics (New law and economics); New Haven School of law and economics; Public Choice Theory; Institutional law and economics; Neoinstitutional law and economics; Critical Legal Studies.}


\begin{itemize}
\item Difference?\footnote{puzzle away}
\end{itemize}

\texttt{Difference?}

* !glossary

head

\textbf{header} document header, containing document specific (i) metadata information or (ii) make instructions

\textbf{(document) structure} relationship between headings and sub-headings, and the objects they contain. Document structure is extracted from heading levels, which are either: explicitly marked up, or; determined from a make regex provided in the document header. Use of document structure allow for the meaningful representation of documents in alternative ways and the use of ocn permits easy reference across different output formats.

\textbf{heading} document heading, each heading is marked indicating its level (in relation to other headings), and this is used as basis for determininge document structure. There are 8 levels, which are can be distinguesed as being one of three types: (i) 1 title level (marked up A or numeric 0); (ii) 3 optional document division levels, above text separating headings (marked up B - D, or numeric 1 to 3); (iii) 4 text headings (marked up 1 - 4, or numeric 4 to 7)

\texttt{levels == heading levels} document heading level, see heading and structure

marked up headings / mark up level

collapsed headings / collapsed levels

numeric levels

\textbf{dummy heading} a markup level 1 / dummy level 4 that does not exist in the original text that is manually inserted to maintain the documents structure rule that text follows a heading of markup level 1 (rather than A to D) (numeric level 4 rather than 0 to 3)

relatives? see ancestors and descendants

document ...

\textbf{ancestors} heading levels above the current heading level which it logically falls
under and to which it belongs (headings preceding current level under which it
occurs)

**decendants** decendant headings are sub-headings beneath the current heading
level, heading levels below the current heading level which are derived from it and
belong to it (sub-headings contained beneath current level); decendant objects are
the range of objects contained by a heading (ocn ranges for each heading in
document body)

**(document) sections** a document can be divided into 3 parts: front; body and;
back. Front matter includes the table of contents (which is generated from
headings) and any parts of the document that are presented before the document
body (this might include a copyright notice for example). The document body, the
substantive part of the document, all its substantive objects, including: headings,
paragraphs, tables, verse etc. This is followed by optional backmatter: endnotes,
generated from inline markup; glossary, from section using a subset of regular
markup, with an indication that section is to be treated as glossary. Note two things
glossary might do that it does not, there is: no automatic (sorting) alphabetisation
of listing; no creation of term anchor tags (perhaps it should); bibliography,
created from a specially marked up section, with indication that section is to be
treated as bibliography; bookindex generated from dedicated markup appended to
objects providing index terms and the relevant range; blurb made up of ordinary
markup, with indication that section is to be treated as blurb

**segment, segmented text** certain forms of output are conveniently segmented,
e.g. epub and segmented html. The document is broken into chunks indicated by
markup level 1 heading (numeric level 4 headings) as the significant level at which
the document should be segmented, and including all decendant objects of that
level. For a longer text/book this will usually the chapter level. (this is significant in
e.g. for epub and segmented html, which are broken by segment, usually chosen
to be chapter)

**scroll** the document as a “scroll”, e.g. as a single text file, or continuous html
document

**object** a unit of text. Objects include: headings; paragraphs; code blocks; grouped
text; verse of poems; tables. Each substantive object is given an object number,
that should make it citable.

**ocn (object citation number / citation number)** numbers assigned
sequentially to each substantive object of a document. An ocn has the
characteristic of remaining identical across output formats. Translations should be
prepared so number remains identical across objects in different languages

unnumbered paragraph (place marker at end of paragraph)

unnumbered paragraph, delete when not required (place marker at end of
paragraph) [used in dummy headings, eg. sometimes used for segmented html,
e.g. to mark a prologue that is not otherwise identified as such as belonging to its
own segment, segment will be created as such an placed in toc, but will not be
found in scroll versions of the document]

citation number (see ocn / object citation number)

**heading auto-numbering** set in header, switched off in markup level 1 with an
appended minus 1 - or 1 given segment-

document abstraction (== internal representation) intermediate step, preprocessing of document, into abstraction / representation that is used by all downstream processing, i.e. for all output formats. This allows normalisation, reducing alternative markup options to common representations, e.g. code blocks (open and close), tables, ways of instructing that text be bold, shortcut way of providing and endnote reference to a link

(document) internal representation (== document abstraction) see document abstraction node representation

attribute (object attributes) when the document is abstracted attributes associated with an object, for example for a: paragraph, indent (hang ... check & add), bulleted, for a: code block, the language syntax, whether the block is numbered

inline markup when the document is abstracted, markup that remains embedded in the text, such as its font face (bold, italic, emphasis, underscore, strike, superscript, subscript), links, endnotes

sequential all objects backkeeping number?
7. Sample Commands

7.1. general

sdp/bin/sdp-ldc -v -epub -html -sqlite-update -output-dir=tmp/program-output data/sisupod/sisu-manual
time ( sdp/bin/sdp-ldc -v -epub -html -sqlite-update -output-dir=tmp/program-output data/sisupod/* )

7.2. source & sisupod

sdp/bin/sdp-ldc -v -source -sisupod -output-dir=tmp/program-output data/sisudir/media/text/sisu-manual.sst
sdp/bin/sdp-ldc -v -source -sisupod -output-dir=tmp/program-output data/sisupod/sisu-manual
sdp/bin/sdp-ldc -v -source -sisupod -output-dir=tmp/program-output data/sisupod/*

7.3. sqlite

sdp/bin/sdp-ldc -v -sqlite-db-drop -output-dir=tmp/program-output
sdp/bin/sdp-ldc -v -sqlite-db-create -output-dir=tmp/program-output
sdp/bin/sdp-ldc -v -sqlite-db-recreate -output-dir=tmp/program-output
sdp/bin/sdp-ldc -v -sqlite-db-recreate -sqlite-insert -output-dir=tmp/program-output data/sisupod/*
sdp/bin/sdp-ldc -v -sqlite-db-recreate -sqlite-update -output-dir=tmp/program-output data/sisupod/*
sdp/bin/sdp-ldc -v -sqlite-db-drop -sqlite-db-create -sqlite-update -epub -html -output-dir=tmp/program-output data/sisupod/*
sdp/bin/sdp-ldc -v -sqlite-db-drop -sqlite-db-create -sqlite-update -epub -html -output-dir=tmp/program-output data/sisupod/sisu-manual
sdp/bin/sdp-ldc -v -sqlite-db-drop -sqlite-db-create -sqlite-update -epub -html -output-dir=tmp/program-output data/sisupod/sisu-manual
sdp/bin/sdp-dmd -v -epub -html -output-dir=tmp/program-output data/sisudir/media/text/sisu_markup.sst

Endnotes
Index

**block text** 224
**blocked text** 207, 209, 213, curly brace syntax, 209, tic syntax, 213
**book index** 323
**breaks** 277
**code block** 22, 37, 252, 259, curly brace syntax, 37, 259, tic syntax, 22
**composite documents** 338
**endnotes** 111
**footnotes** 111
**group text** 216
**grouped text** 207, 209, 213, curly brace syntax, 209, tic syntax, 22
**indented text** 101, hanging, 101
**line break** 278
**links** 135, 151, 184, images, 135, 184, text, 135, 151
**output** 22, 37, 221, 223, 229, 231, 244, 251, 259, block block markup example, 229, code markup example, 22, 37, 259, group block markup example, 221, 223, 231, 244, 251
**page break** 283
**poems** 232
**references** 298
**SiSU markup** 2, 6, 17, 24, 29, 30, 33, 38, 43, 58, 59, 68, 84, 85, 92, 101, 111, 135, 151, 184, 207, 209, 213, 216, 224, 232, 252, 260, 277, 278, 283, 298, 318, 323, 338, 343, bibliography, 298, block text, 224, book index, 323, breaks (page and line), 277, bullets, 92, citations, 298, code block, 252, composite documents, 338, description, 6, examples online, 30, font attributes, 68, Glossary, 318, glossary, 318, group text, 216, grouped text, 207, 209, 213, hanging indents, 101, 111, headers, 33, headers available, 43, heading levels, 24, 59, indentation, 85, indentation and bullets, 84, line break, 278, links (images), 184, links (text), 151, links (text, images), 135, locating examples, 29, page break, 283, poem, 232, references, 298, rules and requirements, 17, sample header, 38, substantive text, 58, substitutions, 343, tables, 260, test, 2
**SiSU markup output** 22, 37, 221, 223, 229, 231, 244, 251, 259, block block (curly brace syntax), 229, code block (curly brace syntax), 37, 259, code block (tic syntax), 22, group block (curly brace syntax), 221, 223, 231, 244, 251
**substitutions** 343
**tables** 260
**text blocks** 207, 209, 213, curly brace syntax, 209, tic syntax, 213
SiSU was started in 1997, open-sourced in 2005.