0.1 Sample Pier file

0.1.1 Chapter & Sections

A line starting with `!` becomes a chapter heading. Use multiple `!` to create sections and subsections

To refer to a section or chapter, put an anchor (equivalent to \label{chapterAndSections} in Latex) using the `@chapterAndSections` syntax on a *separate line*. Then, when you want to link to it (equivalent to \ref{chapterAndSections} in Latex), use the `*chapterAndSections*` syntax. Anchors are invisible and links will be rendered as: .

0.1.2 Paragraphs and framed paragraphs

An empty line starts a new paragraph.

An annotated paragraph starts a line with `@@` followed by either `todo` or `note`. For example,

```
@@note this is a note annotation.
```

generates

this is a note annotation.

And,

```
@@todo this is a todo annotation
```

generates a todo annotation that is not visible in the output.

this is a todo annotation

0.1.3 Lists

Unordered lists

```
-A block of lines,
-where each line starts with ==-==
-is transformed to a bulleted list, where each line is an entry.
```

generates

- A block of lines,

- where each line starts with −
- is transformed to a bulleted list, where each line is an entry.

Ordered lists

```
#A block of lines,
#where each line starts with ==#==
#is transformed to an ordered list, where each line is an entry.
```

generates

1. A block of lines,
2. where each line starts with #
3. is transformed to an ordered list, where each line is an entry.

Description lists

Description lists are lists with labels:

```
;blue
:color of the sky
;red
:color of the fire
```

generates

blue color of the sky

red color of the fire

List nesting

```
– Lists can also be nested.
−#Thus, a line starting with ==−#==
−#is an element of a bulleted list that is part of an ordered list.
```

generates

- Lists can also be nested.
    1. Thus, a line starting with −#
    2. is an element of a bulleted list that is part of an ordered list.

0.1.4 Formatting

There is some sugar for font formatting:

- To make something **bold**, write `""bold""`
- To make something *italic*, write ”`italic`”
- To make something `monospaced`, write ==`monospaced`==
- To make something ~~strikethrough~~, write `-strikethrough-`
- To make something subscript, write `@@subscript@@`
- To make something superscript, write ↑↑`superscript`↑↑
- To make something underlined, write __`underlined`__

0.1.5 Tables

To create a table, start off the lines with | and separate the elements with |s. Each new line represents a new row of the table. Add a single `!` to let the cell become a table heading.

```
|!Language |!Coolness
|Smalltalk | Hypra cool
|Java | baaad
```

Language	**Coolness**
Smalltalk	Hypra cool
Java	baaad

The contents of cells can be aligned left, centered or aligned right by using |`{`, || or |`}` respectively.

```
||centered||centered||centered
|{ left |} right || center
```

generates

centered	centered	centered
left	right	center

0.1.6 Links

Internal Links and Anchors

To put an anchor (equivalent to \label in Latex), use the `@anchorName` syntax on a *separate line*. Then, when you want to link to it (equivalent to \ref in Latex), use the `*anchorName*` syntax. Anchors are invisible and links will be rendered as: .

External Links

To create links to externals resources, use the `*Pharo>http://pharo-project.org/*` syntax which is rendered as Pharo.

0.1.7 Pictures

To include a picture, use the syntax `+caption>file://filename|parameters+`:

```
+Label of the picture>file://pier-logo.png|width=50|label=pierLogo+
```

generates Figure (this reference has been generated using `*pierLogo*`).

This is the label of the picture

0.1.8 Scripts

```
[[[
foo bar
]]]
```

generates

```
foo bar
```

If you want either a label (to reference the script later) or a caption (to give a nice title to the script), write the following:

```
[[[label=script1|caption=My script that works|language=Smalltalk
self foo bar
]]]
```

which produces

Script 1: *My script that works*

```
self foo bar
```

This script can then be referenced with *script1* (produces).

This is another script (referenced as), with no caption this time:

Script 2:

```
foo bar
```

0.1.9 Verbatim

If you want to include verbatim text into a page you must enclose it in {{{ and }}}, otherwise Pier ensures that all text appears as you type it.

A good practice is to always specify for which kind of export the verbatim text must be outputted by starting the block with {{{latex: or {{{html: (for now only LaTeX and HTML are supported). For example, the following shows a formula, either using LaTeX or an image depending on the kind of export.

```
{{{latex:
\begin{equation}
  \label{eq:1}
  \frac{1+\sqrt{2}}{2}
\end{equation}
}}}
{{{html:
<img src="equation1.png" title="Equation 1" />
}}}
```

This results in

$$\frac{1+\sqrt{2}}{2} \tag{1}$$

Take care: avoid terminating the verbatim text with a } asthis will confuse the parser. So, don't write ~~{{{\begin{scriptsize}}}}~~ but {{{\begin{scriptsize}}}} instead.

0.1.10 Preformatted (less used)

To create a preformatted block, begin each line with =. A preformatted block uses equally spaced text so that spacing is preserved.

```
= this is preformatted text
= this line as well
```

0.1.11 Commented lines

Lines that start with a % are considered comments and will be rendered as such in the output document (e.g., in HTML, such a line would be surrounded by <!– and –>).