

```
.title the rest of the line is the title
or if there was no rest of the line, then this whole line is the title
.synonyms | .synonym
ignore words
until the next
dot command,
then execute it.
.description | .definition
if doing contents, then keep on ignoring, until next dot command
if doing definition, then stop ignoring, and process until .contents
or end of file. If .contents hit, then set found-contents?, and pop the
interpreter.
.content | .contents
If doing definition, we will have popped the interpreter before getting here.
If doing contents, then stop ignoring, and process until end of file.
EOF
After end of file, or popping interpreter, the interpreter execute
the .end command before it exits.
```

Modes:

```
.do-definition
  save font
  set definition font
  set do-definition mode
  clear contents-exist
  interpret storyboard
  restore font

do-definition mode
  .title
    read line, put line

  .synonyms
    start ignoring words, till .definition | .description

  .definition
    newline
    start formatting words normally

  .contents
    set contents-exist
    put "full article" button
    pop interpreter

  .end
    pop interpreter

.do-contents
  if contents-exist
    save font
    set contents font
    set do-contents mode
    interpret storyboard
    restore font
```

```

.title
    read line, set window title

.synonyms
    start ignoring words, till .contents | .content

.definition
    start ignoring words

.contents
    start formatting

.end
    pop interpreter

```

```

* Canonicalize (lowercaseify) .commands.

```

```

* .para

```

```

* .picture this is the name of a picture

```

```

* .target this is the name of the target
  this is the reference to go to

```

```

* .~ this is a button~

```

```

* .rem this is a remark

```

```

* .quote .anything

```

```

.table
    This is not urgent.
    ...But it will get done.
.endtable

```

```

* .para-indent <# of spaces>

```

```

.window x y w h

```

Need to handle articles with no .definition

Handle fields in any order

implement RETURN to previous article (with the name of the article) button

```

* make it stop choking on bad references

```

```

* handle button font baselines correctly

```

Box around definition window when there is a definition displayed

```

* Put the title in double quotes.

```

```

* make the FULL ENTRY button look like the other options

```

Drop shadow around the windows

Limit the number of different fonts

.title-font

Make picture annotations sensitive

Show some feedback when the user is pressing a button

make it setfont only when necessary

The index manager does not seem to ignore trailing spaces when comparing strings!

make ignore-until exec word in pattern vocab, and have an emergency escape word in all pattern dicts...