Text is passed to the output. Beginning to implement filled paragraphs.

There are various ways of executing a command to change a section of text. All commands are introduced by the @ character, known internally as the "sigil".

The simple command @i for italics, can use the short form with various sets of delimiters.

@i[italics] produces *italics*

@i(italics) *italics*

@i<italics> *italics*

@i{italics} *italics*

@i 'italics' *italics*

@i :italics; *italics*

These last two require an extra space after the command name since the backquote and colon are not postscript delimiters.

Incidentally, since the command name is scanned with 'token' and executed with 'exec', it can even be a postscript procedure.

@{/Courier 11 selectfont}text in Courier

```
text in Courier
```

Only "short-form" commands take a delimited argument. The @{arbitrary ps code} commands are not "short-form", and do not take an argument but apply directly to the current state. But they do receive the remaining portion of the line as a string, so a custom command may consume data from the string and yield the remainder to be printed (it should leave a string on the stack).

Now with **bold**.

I finally remembered what "deferal" was all about. So let's see if it works. It should allow bracketed commands to span multiple lines. Like so *This sentence should be all italics despite spanning lines, in a line-oriented scanning routine.* And back to normal.

Long-form uses the same command names, but it is now the *argument* to the @Begin{} or @End{} command.

@Begin{i}*Start italics. @End{i}*End italics.

```
typewriter-text oblique
```

Haha! I just read in the scribe paper that @Begin() and @End() sections should always be properly nested. So I just wasted some effort getting this to work:

@Begin{i} italic @Begin{b} bold-italic @End{i} bold @End{b} normal *italic* ***bold-italic*** **bold** normal

But it's probably best to nest things properly anyway. This should be considered "backup" behavior.

An interest has developed in changing the font size. Currently, this can be hacked with explicit postscript.

Big text. back to normal. Double-size text. back to normal. Has the lead actually been reset, or am I fooling myself? I think I may have written a bug where the lead can only increase. This extra text explains the purpose for this extra rambling text. Whew. Fixed.

I just read in the scribe user manual that the same brackets should be able to nest. This I have to fix. I had assumed that the variety of bracket choices () [] {} <> was for the convenience of the implementation, but I was wrong. It is for the convenience of the user, and the implementation has a little more work to do. Currently ibis does not correctly handle nesting of the same delimiters, and you should use different ones when nesting so it doesn't get confused.

`T` **`B`** *`I`* Roman should override all of ***italics,*** **bold,** and `typewriter` flags.