CHAPTER 6: TEXT ENVIRONMENTS

NOAH C. RUDERMAN

ABSTRACT. We will review

- General rules for displayed text environments
- List environments
- Style and size environments
- Proclamations (theorem-like structures)
- Proof environments
- Tabular environments
- Tabbing environments (to be added later)
- Miscellaneous displayed text environments

This is a numbered list made with the `enumerate` environment:

(1) The first item
(2) The second item
The third item

Notice that there are no newline commands between the items. They are by default on separate lines. We see that (1) includes the text "The first item" and that we can use `\eqref` for references that are not equations. We also see that the command `\item[]` gives us an unnumbered item.

We can get a bulleted list with the `itemize` environment:

- The first item
- The second item

  The third item

We see similar commands when we compare the `enumerate` and `itemize` environments. This includes the command `\item[]` to get an unbulleted item.

We can get a captioned list with the `description` environment:

1: This is our first item
Two: This is our second item
She sells sea shells: by the sea shore
: This is our fourth item

We see that the optional argument for the command `\item` is the caption of each item and we cannot omit it and expect favorable results.

Let's try nesting list environments. The limit is four.

(1) This is first item of level 1
(2) This is second item of level 1
    (a) This is the first item of level 2
        (i) This is the first item of level 3
        (ii) This is the second item of level 3
            (A) This is the first item of level 4
    (b) This is the second item of level 2
    (c) This is the third item of level 2

We see that we can reference items such as the first item of level 4 which is item 2(a)iiA.

Let's try the same for bulleted lists:

- This is first item of level 1
- This is second item of level 1
  - This is the first item of level 2
    - This is the first item of level 3
    - This is the second item of level 3
      - This is the first item of level 4
  - This is the second item of level 2
  - This is the third item of level 2

What about descriptions:

first: This is first item of level 1
second: This is second item of level 1
  third: This is the first item of level 2
    fourth: This is the first item of level 3
    fifth: This is the second item of level 3
      sixth: This is the first item of level 4
  seventh: This is the second item of level 2
  eighth: This is the third item of level 2

We can see that nested `description` environments aren't too helpful.

There are several environments for aligning text.

We can use the `flushright` environment to right align text.

We can use the `flushleft` environment to left align text.

Or we can use the `center` environment to center the text.

Take note of the vertical spacing between the
environments that does not exist otherwise.

If you don't want that vertical spacing between lines, you can use the command declarations:
use the `\raggedright` command to left-align.
use the `\raggedleft` command to right-align.
finally use the `\centering` command to center-align.

Typeset Proclamation 1. *This is the theorem-like structure I created.*

If the theorem-like environment starts with a list, you are supposed to include `\hfill` before we declare the list. We can see why below. The first environment doesn't use the command `\hfill` while the second one does.

Typeset Proclamation 2. (1) *this is the first item*

Typeset Proclamation 3.

(1) *this is the first item*

Notice that this text is centered because we never changed things back to how they were orginally.

Now things should be better.
To type two sets of proclamations (theorem-like environments) with consecutive numbering, we must make the optional argument of the second declared theorem like structure the environment name of the first.

Something 4 (this is an optional argument)**.** *See how this is numbered consecutively with the proclamation* ***Typeset Proclamation****?*

Proclamation 0.1. *Here we see that we can number theorem-like structures just like a section. We will see that we can number a theorem-like structure like any section command provided by the document class which includes section, subsection, chapter.*

The general form of a `newtheorem` environment is \newtheorem{*envrname*}[*procCounter*]{*Name*}[*secCounter*] in which the optional arguments are mutually exclusive, meaning that they cannot both exist at the same time.

Haha 0.1. *haha!*

Bruhaha 0.2. *bruhaha!*

I seem to be unable to format the proclamations so that I can get numbering like 1.1, 1.2, 1.3, 2.1, 2.2, etc.

We see that we can get unnumbered theorems:

No Numbering. *This theorem is unnumbered.*

We are also able to have different styles for our theorems:

Plain 1. *This is the formatting for the* plain *theorem-style. This is also the default if not specified.*

Definition 1. Another option is the *definition* style. This is less emphatic than the *plain* style.

Remark 1. Our final option is the *remark* style. Notice that it is the least emphatic.

Proclamation 0.2.

Proof. This a proof, and there is a q.e.d. symbol at the end. Notice how no theorem-like structure was declared for the proof environment. It is probably built in. Also note that this is an environment within an environment. □

Also, for a proof environment, we can substitute the word *proof* for something else by adding an optional argument. Also, in the event that normal typesetting does not place the q.e.d. symbol correctly, we can force the printing of the q.e.d. symbol with the command `\qedhere` command.

PROOF. Here is the equation of a circle:

$$x^2 + y^2 = r^2 \qquad \square$$

Something cool to know is that tables cannot be broken across pages like they can in microsoft word. Also, tables are treated as a single large symbol. We can also create a `table` environment in which the table is set off from surrounding text with vertical space and we can specify where we want our table to appear using float controls.

	Jonathan	Kerry	Bob
Age	19	26	12
Eye color	brown	hazel	blue

TABLE 1. This is our caption

We used the optional argument [`h`] to create the table where the environment was declared instead of at the top or bottom of the page. Note that we format the alignment of the text within columns.
We have the option of specifying the width of individual columns if we like with the `p`{*width*} specifier:

	Jonathan	Kerry	Bob
Age	19	26	12
Eye color	brown	hazel	blue

Notice how we must align the text by using commands on the individual elements of each column rather than when we begin the tabular environment.
We see that the midline characters denote that there will be a horizontal line at that point in the table. Omitting them omits the horizontal line like such: Also, that \hline command creates a horizontal line that we see in the table. Omiiting them removes the corresponding horizontal lines:

	Jonathan	Kerry	Bob
Age	19	26	12
Eye color	brown	hazel	blue

	Jonathan	Kerry	Bob
Age	19	26	12
Eye color	brown	hazel	blue

We can mix and match accordingly:

	Jonathan	Kerry	Bob
Age	19	26	12
Eye color	brown	hazel	blue

Two more features are the command `\cline{a-b}` in which a horizontal line is drawn from column `a` to column `b` and the `\multicolumn`{*number cols*}{*alignment*}{*text*} command which merges row. Here's an example:

Jonathan		Kerry	Bob
Age	19	26	12
Eye color	brown	hazel	blue

Notice how the vertical lines are omitted on either side unless specified in the alignment argument of the `\multicolumn` command. Below I am going to recreate the table from page 137 of Gratzer:

Name	Month	Week	Amount
Peter	Jan.	1	1.00
		2	12.78
		3	0.71
		4	15.00
	Total		29.49
John	Jan.	1	12.01
		2	3.10
		3	10.10
		4	0.00
	Total		25.21
Grand Total of the order			54.70

Table 2. Table 6.3: Table with \multicolumn and \cline.

Yes, you can use the \parbox command to insert a single multiline entry. Also note the use of the \strut command so that the text isn't squished.

The vertical spacing of the table can be adjusted:

	Jonathan	Kerry	Bob
Age	19	26	12
Eye color	brown	hazel	blue

Here is a simple quote:

> It's not that I'm afraid to die. I just don't want to be there when it happens.
>
> *Woody Allen*

Finally we arrive at the `verbatim` environment!. Here's what we can type:

```
The general form of a newtheorem environment is
\newtheorem{envrname}[procCounter]{Name}[secCounter]
in which the optional arguments are mutually exclusive,
meaning that they cannot both exist at the same time.
```

Alternatively, we could use the `\verb|xxx|` command. The midline characters are arbitrary delimiters and the text `xxx` is arbitrary text with is printed verbatim. Try this:

The general form of a newtheorem environment is `\newtheorem{envrname}[procCounter]{Name}[secCounter]` in which the optional arguments are mutually exclusive, meaning that they cannot both exist at the same time.

In the event that the delimiter is used in the text, just use a different delimiter. Anything will do, except an asterisk (*). Also, there is a special version of the `verbatim` environment and `\verb` command. Add an asterisk to show the spaces like such: `today␣is␣Friday`