`alignment.` An *alignment* is a construct for aligning material, such as a table, in columns or rows. To form an alignment you need to (a) describe the layout of the columns or rows and (b) tell TEX what material goes into the columns or rows. A tabbing alignment or a horizontal alignment is organized as a sequence of rows; a vertical alignment is organized as a sequence of columns. We first describe tabbing and horizontal alignments and then more briefly describe vertical alignments.

Tabbing alignments are defined by plain TEX. They are simpler but less flexible than horizontal alignments. Tabbing and horizontal alignments differ principally in how you describe their layouts.

To construct a tabbing alignment you first issue a `\settabs` command (p. '`\settabs`') that specifies how TEX should divide the available horizontal space into columns. Then you provide a sequence of rows for the table. Each row consists of a `\+` control sequence (p. '`\@plus`') followed by a list of "entries", i.e., row/column intersections. Adjacent entries in a row are separated by an ampersand (`&`). The end of a row is indicated by `\cr` after its last entry. If a row has fewer entries than there are columns in the alignment, TEX effectively fills out the row with blank entries.

As long as it's preceded by a `\settabs` command, you can put a row of a tabbing alignment anywhere in your document. In particular, you can put other things between the rows of a tabbing alignment or describe several tabbing alignments with a single `\settabs`. Here's an example of a tabbing alignment:

```
{\hsize = 1.7 in \settabs 2 \columns
\+cattle&herd\cr
\+fish&school\cr
\+lions&pride\cr}
```

The `\settabs 2 \columns` command in this example (p. '`\settabs`') tells TEX to produce two equally wide columns. The line length is 1.7 inches. The typeset alignment looks like this:

cattle	herd
fish	school
lions	pride

There's another form of tabbing alignment in which you specify the column widths with a template. The column widths in the template determine the column widths in the rest of the alignment:

```
{\settabs\+cattle\quad&school\cr
\+cattle&herd\cr
\+fish&school\cr
\+lions&pride\cr}
```

Here's the result:

cattle	herd
fish	school
lions	pride

Horizontal alignments are constructed with `\halign` (p. '`\halign`'). TEX adjusts the column widths of a horizontal alignment according to what is in the columns. When TEX encounters the `\halign` command that begins a horizontal alignment, it first examines all the rows of the alignment to see how wide the entries are. It then sets each column width to accommodate the widest entry in that column.

A horizontal alignment governed by `\halign` consists of a "preamble" that indicates the row layout followed by the rows themselves.

- The preamble consists of a sequence of templates, one for each column. The template for a column specifies how the text for that column should be typeset. Each template must include a single `#` character to indicate where TEX should substitute the text of an entry into the template. The templates are separated by ampersands (`&`), and the end of the preamble is indicated by `\cr`. By providing an appropriate template you can obtain effects such as centering a column, left or right justifying a column, or setting a column in a particular font.
- The rows have the same form as in a tabbing alignment, except that you omit the `\+` at the beginning of each row. As before, entries are separated by `&` and the end of the row is indicated by `\cr`. TEX treats each entry as a group, so any font-setting command or other assignment in a column template is in effect only for the entries in that column.

The preamble and the rows must all be enclosed in the braces that follow `\halign`. Each `\halign` alignment must include its own preamble.

For example, the horizontal alignment:

```
\tabskip=2pc
\halign{\hfil#\hfil &\hfil#\hfil &\hfil#\hfil \cr
  &&\it Table\cr
\noalign{\kern -2pt}
   \it Creature&\it Victual&\it Position\cr
\noalign{\kern 2pt}
  Alice&crumpet&left\cr
  Dormouse&muffin&middle\cr
  Hatter&tea&right\cr}
```

produces the result:

Creature	*Victual*	*Table Position*
Alice	crumpet	left
Dormouse	muffin	middle
Hatter	tea	right

The `\tabskip` (p. '`\tabskip`') in this example tells TEX to insert `2pc` of glue between the columns. The `\noalign` (p. '`\noalign`') commands tell TEX to insert vertical mode material between two rows. In this example we've used `\noalign` to produce some extra space between the title rows and the data rows, and also to bring "Table" and "Position" closer together. (You can also use `\noalign` before the first row or after the last row.)

You can construct a vertical alignment with the `\valign` command (p. '`\valign`'). A vertical alignment is organized as a series of columns rather than as a series of rows. A vertical alignment follows the same rules as a horizontal alignment except that the roles of rows and columns are interchanged. For example, the vertical alignment:

```
{\hsize=0.6in \parindent=0pt
\valign{#\strut&#\strut&#\strut\cr
  one&two&three\cr
  four&five&six\cr
  seven&eight&nine\cr
  ten&eleven\cr}}
```

yields:

```
one     four    seven   ten
two     five    eight   eleven
three   six     nine
```

The `\strut` commands (p. '`\strut`') in the template are necessary to get the entries in each row to line up properly, i.e., to have a common baseline, and to keep the distance between baselines uniform.