

# TEXNICAL PROGRAMMING

Slurp  
January, 2023

- T<sub>E</sub>X formats and engines: T<sub>E</sub>X vs L<sup>A</sup>T<sub>E</sub>X
- Typesetting in plain T<sub>E</sub>X
- Understanding macros
- Repeated macros
- Lists

T<sub>E</sub>X creates its pages by creating a vertical list and filling it with boxes, glue, leaders, penalties, kerns, etc. There are two types of boxes, horizontal boxes and vertical boxes.

Boxes have width, height, and depth. The height of a box is visually its positive vertical displacement relative to the baseline, and its depth is its negative vertical displacement relative to the baseline. For example:

$$\begin{array}{r}
 113.3337\text{pt} \\
 + \quad 27.77779\text{pt} \\
 \quad 7.77779\text{pt} \\
 \hline
 \text{yelling}
 \end{array}$$

**\hbox**

`\hbox to/spread <dimension>{<horizontal material>}`: Creates a horizontal box, if *to dimension* is specified then the width of the box is *dimension*.

If *spread dimension* is specified then the width of the box is *dimension* more than its natural dimension.

The height and depth of the box is equal to the height and depth of its contents.

**\vbox**

`\vbox to/spread <dimension>{<horizontal material>}`: Creates a vertical box, if *to dimension* is specified then the height of the box is *dimension*.

If *spread dimension* is specified then the height of the box is *dimension* more than its natural dimension.

The width of the box is the width of its contents, and the depth of the box is the depth of the final box in it.

Between two boxes there is something called *glue*, which connects the boxes. Glue is one type of blank space (the other being a kern).

Glue has three attributes: its natural length, its maximum stretchiness, and its maximum shrinkage. Glue stretches and shrinks only when it needs to, and T<sub>E</sub>X uses these attributes in order to fit material into widths the material couldn't properly fit into in its natural width.

For example:

```
1 \hbox to 5cm{hello\hskip 3cm plus 2cm}there
```

Creates

hello there

Without the `plus2cm` we'd get the same output but with an overfull hbox warning.

The amount of stretchiness and shrinkage can be infinite.

All the dimensions may be negative as well.

**\hskip**

**\hskip** *<natural length>* **plus** *<stretch>* **minus** *<shrink>*: Adds horizontal glue with the specified natural length, maximal stretch and shrinkage.

The stretch and shrink are optional.

**\vskip**

**\vskip** *<natural length>* **plus** *<stretch>* **minus** *<shrink>*: Adds vertical glue with the specified natural length, maximal stretch and shrinkage.

The stretch and shrink are optional.

**\kern**

**\kern** *<dimension>*: Adds a kern whose dimension is *dimension*. Kerns, unlike glue are nonbreaking, nonstretching, and nonshrinking. The orientation of the kern (horizontal or vertical) is inferred by the context.

T<sub>E</sub>X has 3 orders of infinities for glue stretching:

**fil** First order **fil**: `\hskip 0pt plus 1fil\relax` creates glue which has no natural length but has infinite stretchiness. A primitive version, `\hfil`, exists as well in place of the code above.

**fill** Second order **fill**: `\hskip 0pt plus 1fill\relax` creates glue which also has no natural length and infinite stretchiness. It takes precedent over first order infinities. A primitive version, `\hfill`, exists as well.

**filll** Third order **filll**: Same as the other two, but takes precedent over both of them. No primitive version exists.

Vertical versions of `\hfil` and `\hfill` exist, `\vfil` and `\vfill`.

Another important primitive is `\hss` which can both shrink and stretch infinitely. It is analogous to `\hskip 0pt plus 1fil minus 1fil`.

It too has a vertical version `\vss`.

```

1  \def\line{\hbox to \hsize}
2  \def\centerline#1{\line{\hfil#1\hfil}}
3  \def\rightline #1{\line{\hfil#1}}
4  \def\leftline  #1{\line{#1\hfil}}
5  \def\rlap#1{\hbox to 0pt{#1\hss}}
6  \def\llap#1{\hbox to 0pt{\hss#1}}

```

`\line` creates a box which spans the entire line.

`\centerline` centers input relative to the line.

`\rightline` and `\leftline` right and left-justify input respectively.

`\rlap` typesets input and then seems to move back as if it hadn't been typeset.

`\llap` moves back the width of its material and then typesets it.



```

1  \centerline{Centered Text}
2  \rightline{Right-Justified}
3  \leftline{Left-Justified}
4
5  \quitvmode\llap{outside}\hfill1\llap{0} and \rlap{1}0
6      \hfill\rlap{outside}

```

		Centered Text		
	Left-Justified		Right-Justified	
outside		0 and 0		outside