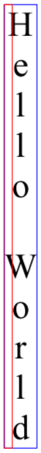
Multi-line Text Layout
Text is split into lines based on the application specified wrapping width or the presence of newline characters ('\n'). Every line will contain at least one character even if it exceeds wrapping width. The splitting into lines takes place before any horizontal text alignment, which works on the already split lines.

Alignment
Text is by default left aligned, but may also be center or right aligned, and also justified. Alignment first has to determine the “alignment width” to use for the node. Typically this is the wrapping width. But if wrappingWidth is not set and the text was split based solely on the presence of new lines, the maximum calculated line advance is used. Additionally as a special case, if the wrappingWidth is set to a very small value which cannot fit even a single glyph then also the width of the widest glyph is used. This is essentially maxAdvance. The alignment algorithm uses the difference between the alignment width of the node, and the advance of each lines to determine the amount to adjust the text position. Alignment will not cause text to be aligned to the left of its origin, even if right aligned and wrapping width is less then a single glyph (which follows the behaviour of OpenOffice for example).

If wrapping width is less than the widest glyph present in the text, then each glyph will align to the advance of the widest glyph. Thus if content is changed the text will be rendered in a different location. This means that while wrapping width exceeds the widest glyph advance, text is centered mid-way between the origin and wrapping width, however when wrapping width is less the max char advance, remaining text will align to center of mass (see illustration red is wrapping width)



layoutBounds
In logical bounds mode this reports height based on the number of lines and width based on the wrapping width if set else the maximum line advance. Note layout bounds is not guaranteed to contain all glyph/pixel data. Having layoutBounds width correspond to wrappingWidth ensures that horizontal layout will not change based on the content of a particular line.

boundsInLocal
In logical bounds mode, this is based on l*ayoutBounds,* but extended to include any glyph pixels which go outside this.

SetBoundsType set to VISUAL
Does not affect alignment or multi-line text layout. It changes *LayoutBounds* and *BoundsInLocal* to be minimal visual bounds.