REQUIREMENTS

1. A html document can have paragraphs of texts. Each paragraph can contain multiple consecutive **sentences**.
2. On the Left or Right hand side (configurable) of the paragraph, there will be a “gutter” which contains multiple vertical **strips**.
3. Each sentence *can indicate* to the reader that that particular sentence is eg logically complicated to read.
    1. It does this by displaying a grayscale gradient inside each strip. Each strip can contain multiple gradients.
    2. The gradient is defined by the normal distribution curve’s pdf function.

An Example

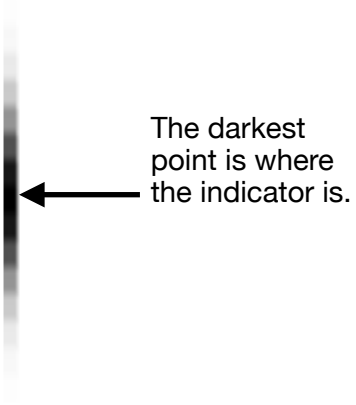
For the example below, the * represents the indicator to the reader that this sentence is logically complicated. The * will not be rendered on screen. It is just for example purposes.

Example 1 (1 paragraph)

Here is a sample sentence.
Here is the second sentence in the paragraph.
Here is the third sentence. Here is the 4th sentence.
Here is the 5th sentence. Here is the 6th sentence.

Here is the 7th * sentence.

Here is the 8th sentence. Here is the 9th sentence.
Here is the 10th sentence. Here is the 11th sentence.
Here is the 12th sentence. Here is the 13th sentence.
Here is the 14th sentence.



Example 2 (1 paragraph)

Here is a sample sentence.
Here is the second sentence in the paragraph.

Here is the * third sentence. Here is the 4th sentence.

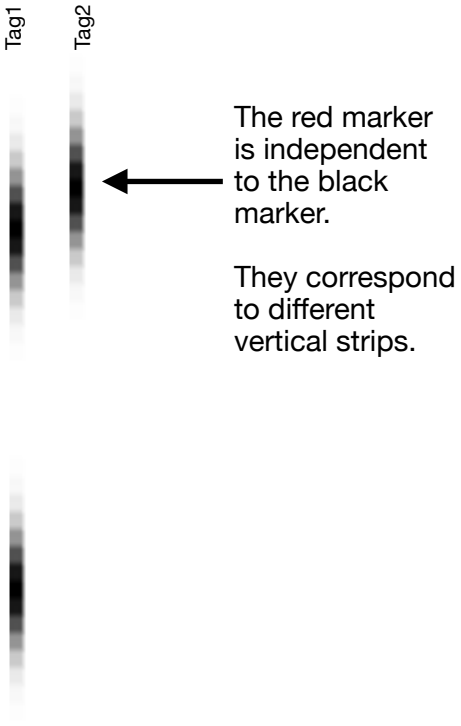
Here is the 5th sentence. Here is the 6th sentence.

Here is the 7th * sentence.

Here is the 8th sentence. Here is the 9th sentence.
Here is the 10th sentence. Here is the 11th sentence.
Here is the 12th sentence. Here is the 13th sentence.
Here is the 14th sentence. Here is the 15th sentence.
Here is the 16th sentence in the paragraph.
Here is the 17th sentence. Here is the 18th sentence.
Here is the 19th sentence. Here is the 20th sentence.

Here is the 21st * sentence.

Here is the 22nd sentence. Here is the 23rd sentence.
Here is the 24th sentence. Here is the 25th sentence.
Here is the 26th sentence. Here is the 27th sentence.
Here is the 28th sentence.



Configuration

Paragraph:

* Left or Right: If the gutter should be left or right
* Gap: distance (in px) of gap between strips
* Width: width of strip
* Tag(s): Some optional text that appears at the very top of the vertical strip. See Example 2.
* Order: Order of each strip from left to right.

Indicator

* Limit = For how many sentences the gradient should be used. It is a hard-limit, after which gradient is totally white.
* Spread (α) = How spread out the normal distribution's pdf is.
* Scale = This is a scaling factor applied to the normal distribution's pdf. It determines the maximum darkness. i.e. 0.6 = 60%

Normal Distribution Curve (Use to color the grayscale gradient)

$f(s) = \exp(-s^2/\alpha)$

* f(s) is the darkness. f(s) is between 0 (white) and 1 (black)
* s is the number of sentences away from the indicator. s=0 is location of middle of sentence containing indicator.
* α is the spread. A smaller α corresponds to a more concentrated gradient.

The gradient is symmetrical about s=0.

s is always from the middle of the sentence containing the indicator.

The function above is always between 0 and 1. However, when it is time to render to the screen, then the scale factor is further applied. If scale is 0.6, then the function's results are scaled from 0 to 0.6. That means it never reaches black and is maximum grey.

Additional Notes

The function above is infinitely long. It must be cutoff so it does not exceed the boundary of the paragraphs. It must also not exceed the Limit (see "Indicator" section above) either. eg. If limit is 2, then the gradient should not exceed ± 2 sentences.

If 2 (or more) indicators (from the same strip) intersect, then the value used to render is the maximum of each. They are not combined/added or averaged out.

The above equation is in terms of sentences *away* from indicator. Obviously when it comes to final rendering, the position of the sentences must map to the y-axis on screen (pixels). This mapping will change when the window size changes or the smartphone is rotated. It must adapt to changes. This includes when multiple sentences may end up on the same line.

Extra requirements

Only pure Javascript and CSS (No jQuery)

Resources

https://stackoverflow.com/questions/5143534/how-to-get-the-position-of-text-within-an-element

https://stackoverflow.com/questions/442404/retrieve-the-position-x-y-of-an-html-element/442474#442474

https://jsfiddle.net/abrady0/ggr5mu7o/

http://jsfiddle.net/BinaryMoon/zy8hY/

https://stackoverflow.com/questions/1589721/how-can-i-position-an-element-next-to-user-text-selection/1589912#1589912

https://stackoverflow.com/questions/16337327/how-to-get-x-y-coordinates-of-text-in-paragraph-like-quora-does