

npm@, maxlg@
July 2019

Here we present reasoning for the text aggregation approach used in [Element Timing](#).

Selected approach:

- Aggregate a text node to the nearest [containing block](#) ancestor

This approach is beneficial for several reasons:

- Associated with a well-defined CSS concept.
- Intuitively matches what we want: text nodes aggregated in blocks.
- Can be implemented in a performant way by using a stack during paint traversal.

Alternatives considered:

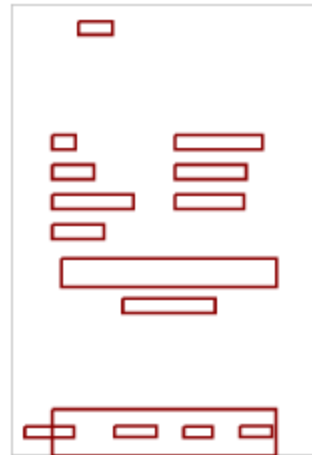
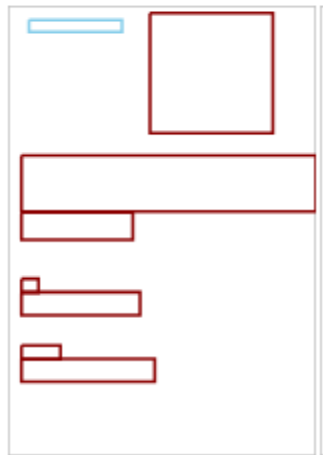
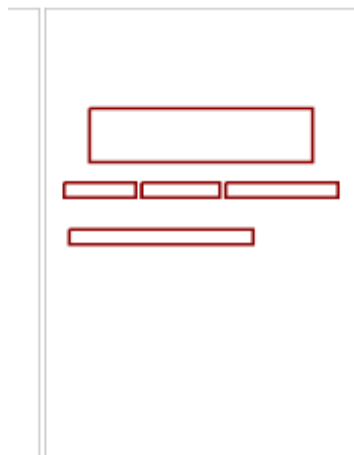
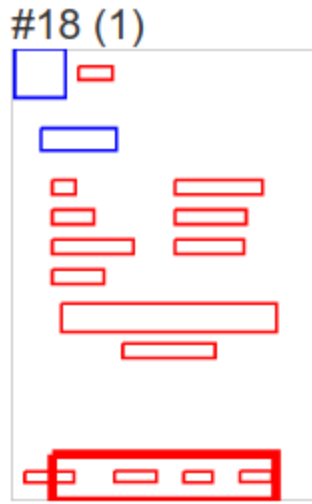
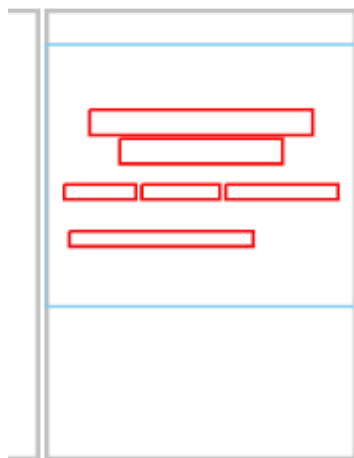
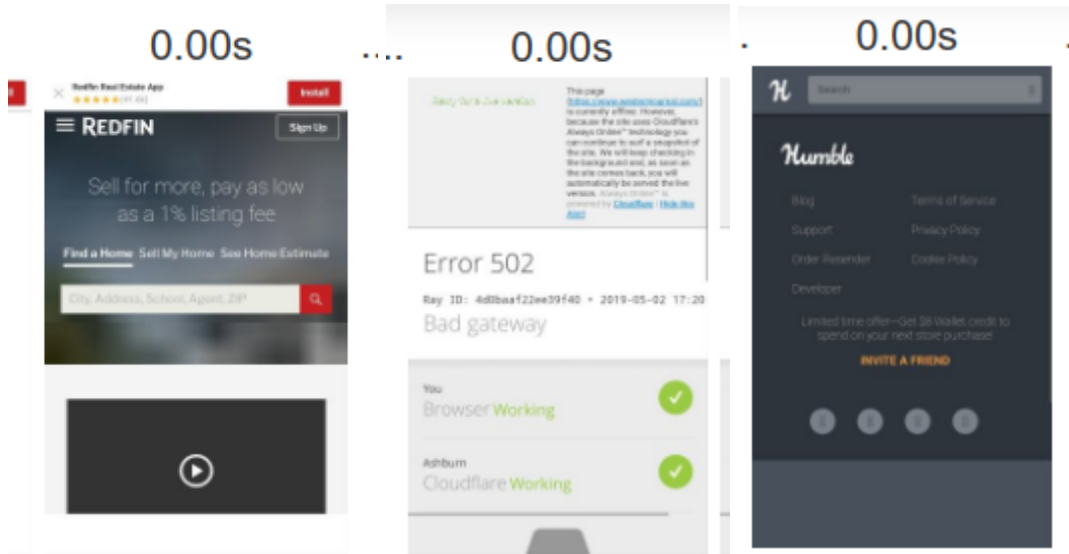
- Using some notion of depth: arbitrary and does not necessarily match user experience. For example, having “a href” should not cause the text to change where it belongs.
- Using some notion of “top-level” elements: text nodes belong to the nearest top-level ancestor. This could potentially work well, but defining a new type of element is not future-proof as it would require constantly updating them.
- Using [phrasing content](#) instead: this could also work well, but harder to implement than the block-level approach and no clear benefit from the added complexity.

Examples (screenshots)

In the following examples:

red=text node rects

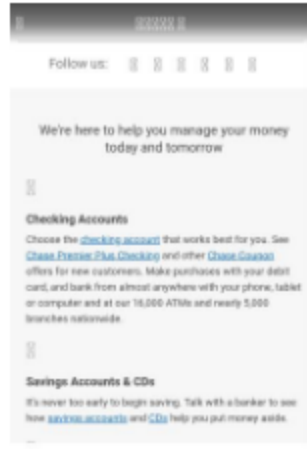
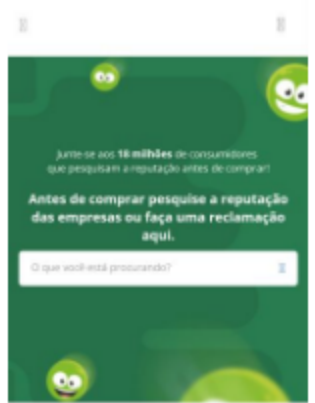
dark red / brown=aggregated text rects



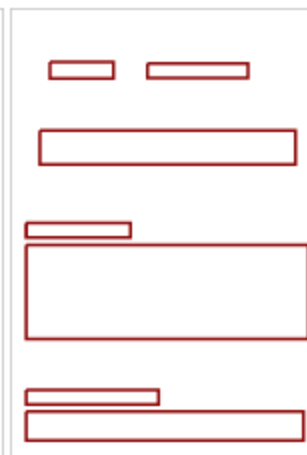
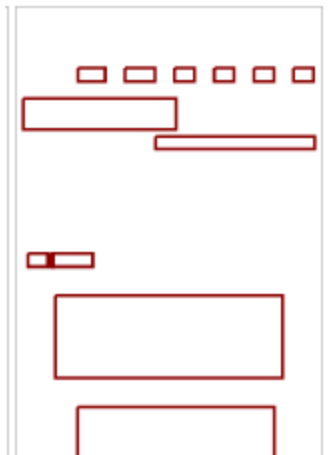
Better title

Better paragraph

Unchanged



#48 (1)



Better paragraph

Aggregates bold text

Paragraph with links