Fixing Line Layout and Vertical Rhythm

Elika J. Etemad aka fantasai
W3C CSS Working Group

CSS Line Box Model of Paragraphing

Constraint: We're not sure what the content is.

Goal: Don't overlap content.

Foundation: Lots of boxes?

1. Put a bunch of stuff on a line
2. Draw a box around it
3. Stack up all the boxes
4. You're done!

Typographers Line Layout Model

- Use consistent baseline-to-baseline spacing to create vertical rhythm.
- Carefully tune the top line to visually align with other things.
- Pick appropriate line spacing, for pleasant reading and to accommodate ascenders/descenders/superscripts/subscripts/ruby.
- If your stuff overlaps it's your fault. So review and tweak your things before you print them!

Fixing the Line Layout Model

Goal: Make typographers happy. Also don't overlap the content.

Current State of Line Layout

- Year 2000: Eric Meyer
- Year 2000: https://dbaron.org/css/2000/01/dibm
- Year 2014: https://christopheraue.net/design/vertical-align
- Year 2017: https://iamvdo.me/en/blog/css-font-metrics-line-height-and-vertical-align

Font Metrics We Care About Now

- Font ascent + descent
- Other ascent + descent
- Line gap (sometimes)?
- Em box top / bottom
- Baseline

- Font Height
- Em Height

Half-leading

- Applied to get line-height box
- Two ways to think about it: ascent + descent + leading = line-height
- Font-size + leading = line-height
- Browser does 1st one because we need to know where the baseline is.

Why half-leading instead of line gaps?

- Paragraph-to-paragraph spacing stays consistent. [diagram]

CSS Line Layout: the Simplest Case

- Root Inline Box – wraps all the content of the paragraph [diagram]
- Line-height says how tall it is [diagram]
- Half-leading says how it's positioned within its line-height [diagram]
- Uses metrics of the first available font [define]
- Stack them all up -> Typographer is happy!

1st Complication: Inline Elements

- Same font / size -> maintain rhythm! [diagram]
- Larger font size -> line height increases [diagram]
- Different font -> line height increases [diagram]
- Smaller font size -> line height increases! [diagram]
- Basically any change in font causes a line height change. ☹

3rd Complication: Vertical Alignment

- Like baseline alignment, vertical alignment causes line boxes to shift wrt each other.

- Can poke out of root line box -> increases line box sizes -> breaks rhythm.

2ⁿᵈ Complication: Font Fallback

- 'normal' says "Use the font metrics" -> uses ascent + descent
- Some browsers add the line-gap metric, too
- Font fallback: substituting a different font for the glyph than the rest of the text
- Apply half-leading to all glyphs, then align them, then draw the line box around them [diagram]

Small Wins

- We draw the line box only around the first available font's metrics for non-normal values, so fallback doesn't affect those.

Happiness is Rare

- Venn diagram: What the typographer wants, what CSS does, typographer happy = no changes in font family, size, or vertical alignment

How do we fix this?

- Well, we have boxes. Let's start with the root inline box…
- If it stacks nicely, and we ignore everything else, we have the typographer's model.
- But we fail to prevent overlap when things get really big. ☹
- Only increase height if things don't fit within the outer edges?

Defining "fit within a line box".

- Safe model: fit within half-leading. Sparse model: fit within leading. [diagram]

- Including leading is clearly the wrong answer, so let's get rid of the leading.

- But remember two concepts of leading? Which box to use: font height, em height, something else? [diagram]

- But wait, what if leading is negative? Line-height: 0.8 vs inline without leading…

Ok, so what about visible boxes?

- Margins, borders, padding completely ignored
- Using leading box now:
- Future: use margin box? (Everyone understands what it is: you can see it!)
- But… that's the font box, not the em box.

Syntax

- Line-sizing: legacy | normal
- Legacy = what CSS2.1 says
- Normal = TBD!

- Rhythm within a paragraph: done;
- What about rhythm across paragraphs?

Spacing as a multiple of line-height

- Lh and rlh units
- Margin: 1em 0 [diagram] -> margin: 1lh 0 [diagram]

Interrupting the Rhythm

- [diagram heading interruption]
- [diagram line-height: 4lh]
- [diagram wrapped]
- [diagram wrapped ideal]
- [diagram images, tables]

Fun Question: What happens at a Page Break?

- [diagram]

Trimming Text

- [diagram image + text with gap]
- [diagram spacing above /below the letter T]
- Leading-trim: …
- What is it useful for?
- Spacing [diagram] centering [diagram] inlines? [diagram]
- Before/after fragmentation breaks? New property → leading-break???
- Should metric inherit separately from whether to trim? (i18n)
- Other thoughts/ use cases? Tell me.

TrimmingExtra Space: Top/bottom of page margins

- [diagram with space] [diagram without space]
- Margin-break property

Vertical Alignment

- Base case: everything aligns on the baseline.
- Vertical alignment = baseline match + shift
- But which baseline? [diagram baselines]
- Choosing an alignment baseline:
  - Dominant-baseline [alphabetic | mathematical | central | ideographic]
  - Alignment-baseline [alignment-baseline: middle]
  - Vertical-align shorthand [vertical-align: central]
- Shift amount (default = 0 = match baselines) baseline-shift
- Vertical-align: … [show longhand expansions for some common values]
- Special values: top [diagram], bottom [diagram], middle cmp central cmp center [diagram]?

Drop caps

- Re-using same metrics for drop caps and vertical alignment
- [drop cap diagram]

Metrics: chicken or egg?

- Font metrics generally suck.
- OpenType only has metrics for Latin and CJK. What about Hebrew? Devanagari? Thai? Arabic???? [diagram each letter]

How to help

- Think about / comment on issues.
- Bug browser vendors to implement once specced.
- Bug font vendors to provide correct metrics.
- Bug OpenType to define missing metrics.