Revisiting multicolumn text

This set of features related to multicolumn text is near and dear to my heart. I first researched this feature when I wrote about it for the Web Platform Documentation Project (since retired).

The idea is to let the browser take care of creating the columns and the details about the columns, the gap and other elements. We can also play with headers and how they span accross the columns.

Browser support, as always, is the stumbling block for using columns accross browsers but it appears to be better than when I first tried it.

I searched caniuse.com for `column-` knowing that all properties we'll discuss in this post begin with that string. Check this link for feature browser support and the CSS Multi-column Layout Module Level 1 specification working draft for more technical information.

The text we'll be working with

I create the Codepen template below as a starting point for all the examples that follow. If you go to Codepen you'll be able to use the template for your own work.

Definining columns

We have two ways to define columns with CSS. We can set a specific number of columns with a given width using `column-count`. This will creaate the specified number of columns regardless of the screen size.

```css
.mc-column {
  column-count: 2
}
```

The other way to specify columns is to set the width of the column and let the browser fit it as many columns of the given width as it can.

```css
.mc-container {
  column-width: 15rem;
}
```

If both `column-count` and `column-width` are present, `column-count` takes precedence. If you use code like this:

```
.mc-container {
  column-count: 2;
  column-width: 10rem;
}
```

There will be two columns of equal width. The browser will ignore the value of `column-width`.

Column spans

There are times when we may want content to span across multiple columns. We can do this using the `column-span` property. This is useful for long titles or images.

```
h2 {
  column-span: all;
}

figure {
  column-span: all;
}
```

Right now, we can only span across all columns or not at all. Using values other than `all` will result in the spaning content overlaping the columns.

Column fill

There are times when we need to balance the column content so the final column doesn’t appear empty. We can use the `column-fill` property to do it.

> The idea is that this property will fill the columns as equally as possible.

```css
.mc-container {
  column-fill: auto;
}
```

The value `balance-all` is currently not supported on any browser.

> The default value for `column-fill` is balance.

Column gap

Like Grid and Flexbox, we can control the separation between columns using the `column-gap` property.

```
.mc-container {
  column-count: 2;
  column-gap: 7em;
}
```

Column gap works particularly well with column rules, described next.

Column rules

Sometimes we have to make the separaration between columns explicit and clearer for the user to see. We can do it using the `column-rule` property. The way I use it, `column-rule` takes three values:

- A color
- A style for the rule
- A length value for the width

This example combines the `column-gap` from the previous example with the `column-rule` property.

```
.mc-container {
  column-count: 2;
  column-gap: 7em;
  column-rule: rebeccapurple dotted 10px;
}
```

The items that make the `column-rule` property can also be expressed individually:

- Color uses `column-rule-color`
- Style uses `column-rule-style`
- Width uses `column-rule-width`

There may be time when the individual properties work better but I find that using the `column-rule` shorthand works better and prevents user errors (like I forgot to add the width :))

Column breaks

There is an additional set of properties that can be used to control how and if the text will break inside a column.

I chose not to cover them because support is uneven and, rather than figure out how to code workarounds, I decided to continue waiting until there is more uniform browser support.

If you’re interested in learning more about these properties, check the column breaks section of the CSS Multicolumn spec and the associated definitions for break-before, break-after, and break-inside in the CSS Fragmentation Module Level 3