Custom Counter styles

Ready-made Counter Styles provide a collection of predefined counter styles you can just drop into a stylesheet.

These pre-defined custom counters are great to work in international documents, since many non-latin languages define their own numbering systems and the characters they use on their systems. They also provide several Arabic and Roman predefined number styles.

The following example creates fifty Arabic numbers with a circle around them (it uses Unicode codepoints to make the numbers).

```
@counter-style circled-decimal {
  system: fixed 0;
  symbols: "\24EA""\2460""\2461""\2462""\2463""\2464""\2465""\2466""\2467
  suffix: " ";
}
```

Once we have the style, we can reference it in our definition for a list item. In this case we use the default numbered list.

```
ol {
  list-style-type: circled-decimal;
}
```

We could also create multiple counter styles and reference them by class or ID.

If we define a `filled-circle-decimal` counter style, we can use it to create a list that uses it.

```
.filled-list {
  list-style-type: filled-circled-decimal;
}
```

One big caveat with custom counter styles is that you must define all the numbers that you intend to use. That’s why the `circled-decimal` style defined fifty numbers. I doubt I would ever use that many but if I ever do, numbers above fifty will not get the circled style.

Same with `filled-circle-decmal`. It will only display the first ten items with the style and everything after that will be displayed with the default style. We can define more numbers in the style declaration to avoid this problem.

Another consideration is Safari and WebKit browsers. They don’t support custom counter styles so we’ll have to work around that using a feature query or perhaps using Unicode directly.

A working example can be found in this codepen