CSS math functions min(), max() and clamp()

CSS provides a set of functions for creating numerical expressions to use without having to resort to complex calculations using `calc()` and variables.

Clamp

My favorite use for these functions is to control font sizing in a dynamic environment. Until I discovered `clamp()` I couldn't think of any way to not let the font get smaller or larger than preset values.

The following CSS rule will restrict the size of our paragraph text:

```
p {
  font-size: clamp(16px, 1.5vw, 24px);
}
```

The rule means that: If 1.5vw is smaller than 16px then keep it at 16px. Likewise, if 1.5vw is larger than 24px then keep it at 24px and if it's between 16 and 24px then we keep the 1.5vw value.

Min and Max

We also have `min()` and `max()` functions that work by taking one of two values; which value it takes depends on the function we use.

In the first example, we create a container that is 40em or 400px, whichever is smaller at the current screen size.

```
.container {
  border: 2px solid red;
  margin: 0 auto;
  padding: 0 2em;
  width: min(400px, 40em);
```

```
}
```

This example provides the opposite example. The width will be 400px or 40 em, whatever is **larger** at the current screen size.

```
.container2 {
  border: 2px solid red;
  margin: 0 auto;
  padding: 0 2em;
  width: max(400px, 40em);
}
```

We can also chain multiple `min()` and `max()` functions to create more complex values.

This example will calculate the `min()` value between 5em and 64px and then use that as the value of the outer `max()` function.

```
.container {
  width: max(10px, min(64px, 5em))
```

You can also have more than two values for either `min()` or `max()` but it becomes harder to reasson throug the rule so I'd recommend always keeping it down to two final values.

Links and Resources

- Max
- min
- clamp