



Animating variable fonts

One of the things I find the most interesting about variable fonts is that you can animate them between different values.

This post will explore how to create animations using a Variable font axes, both default axes like weight and custom axes like casual available in [Recursive](#)

I will be using the following HTML:

```
<h1 class="weight">Hello World</h1>

<h1 class="casual">Hello World</h1>
```

The CSS is broken in multiple blocks for readability and ease of explanations.

The first block performs the following tasks

1. Load the variable font using a modified [@font-face](#) syntax
 1. The format for the font changes to reflect that it's a variable font
 2. We repeat the url to accommodate two values for the format attribute
 3. font-weight takes two values representing the boundary values for the attribute
2. Define the default values for the variable fonts in the :root attribute. We use :root rather than html because :root has higher specificity
3. Add a set of default attributes to the universal selector. This will match all elements on the page so we don't have to add them individually to all elements
4. We add padding to the HTML and enlarge h1 elements

```
@font-face {
  font-family: "RecursiveRecursive VF"->
  src: url("fonts/recursive.woff2") format("woff2 supports variations"),
       "woff2 supports variations" url("fonts/recursive.woff2") format("woff2");
  font-weight: 300 1000;
  font-display: swap;
```

```

}
:root-mono: 0;
  --recursive-casual: 0;
  --recursive-weight: 400;
  --recursive-slant: 0;
  --recursive-italic: 0.5;
}

* {
  font-family: "Recursive VF", Verdana, sans-serif; <!-- 3 -->
  font-weight: var(--recursive-weight);
  font-variation-settings:
    "MONO" var(--recursive-mono),
    "CASL" var(--recursive-casual),
    "slnt" var(--recursive-slant),
    "ital" var(--recursive-italic);
}

body { <!-- 4 -->"Recursive VF"body { <!-- 4 -->
  padding: 2em;
}

h1 { <!-- 4 -->
  font-size: 3em;
}

```

Then for each animation we need to do two things:

1. Define the animation using the [animation](#) using either the shorthand or individual attributes
 1. Both examples use the shorthand syntax
2. Create the keyframes sets using the [@keyframes](#) at-rule
 1. The name must match the animation property defined in the previous step

```

.weight {
  opacity: 0;
  animation: weightAnim linear 2s forwards; /* 1 */
}

```

```

}

@keyframes weightAnim { /* 2 */
  from {
    opacity: 0;
  }
  to {
    font-weight: 1000;
    opacity: 1;
  }
}

```

```

.casual {
  animation: casualAnim linear 4s forwards; /* 1 */
}

@keyframes casualAnim { /* 2 */
  from {
    font-variation-settings: "CASL" 0;
  }
  "CASL"to {
    font-size: 8em;
    font-variation-settings: "CASL" 1;
  }
}

```

Tools like [Splitting.js](#), [Lettering.js](#) or its jQuery-less counterpart (in this [Gist](#)) allow for more complex effects targetting portions of a sentence or even individual characters.

See Mandy Michael's [Interactivity and Animation with Variable Fonts](#) for a more in-depth coverage of how to animate variable fonts and examples of what this looks like.