This is a series of tutorials. The intention is to cast light on the potential of **OpenType font format specification as a compact format for creating all kinds of dynamic graphics** (not only fonts) for web-use. Assume that if you are familiar with

animation and variable fonts

a bit.
Therefore the introductions will be not extensive.
I will put all the links in the description for further discoveries.
I will use Glyphs3 for these series because I found the user interface easy to understand and there are plenty of plug-ins you can download for free and play around.
I will focus on the variable font editor part of the software during the series.
We will need a tiny bit of coding skills to animate our results but I will provide a code template which you can reuse to see your results.
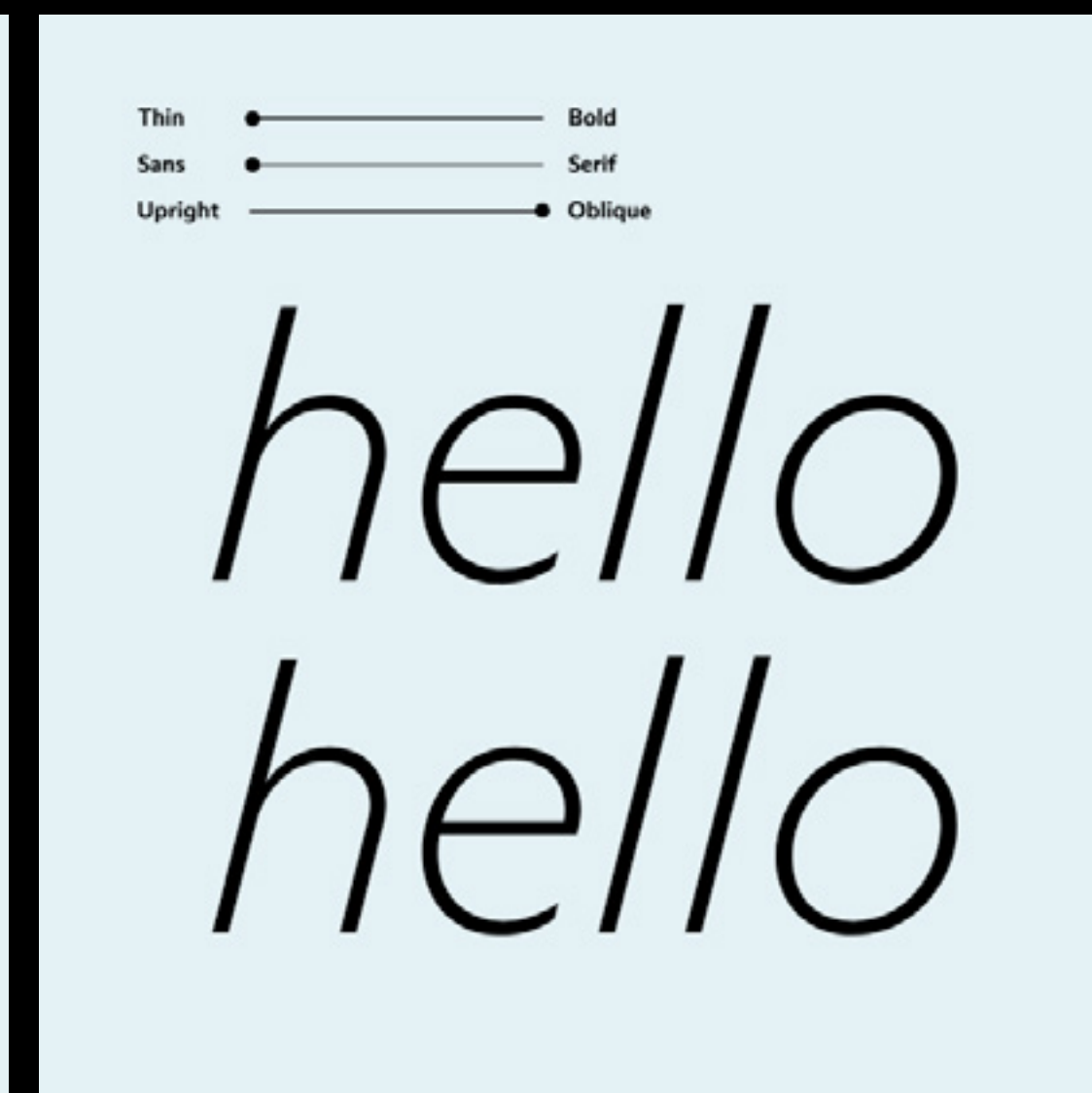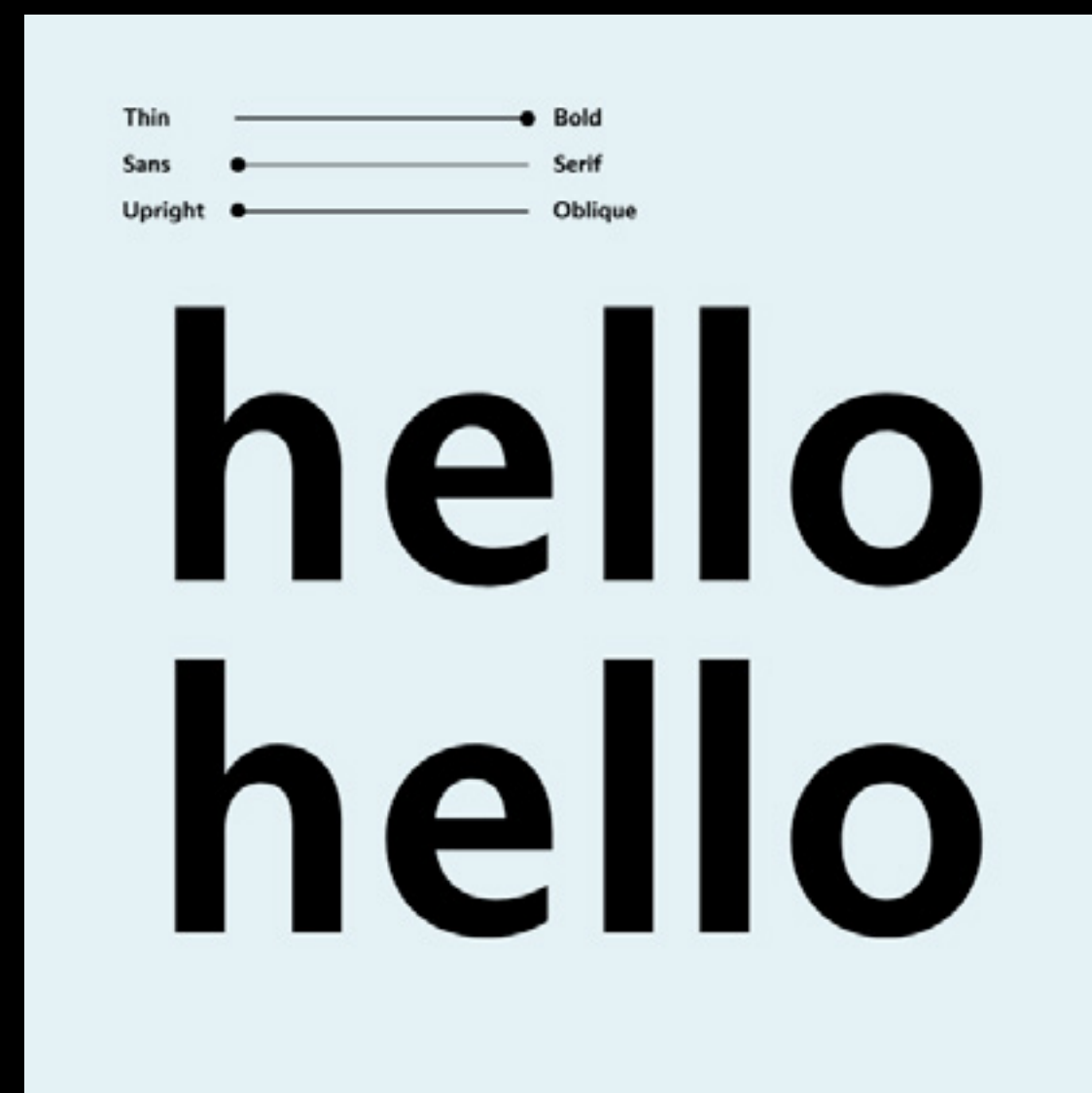
Welcome!

1.

What is a variable font?

A variable font is a single font file that behaves like multiple fonts.

Compact file-size, faster loading than other web fonts, best option to create dynamic design elements.

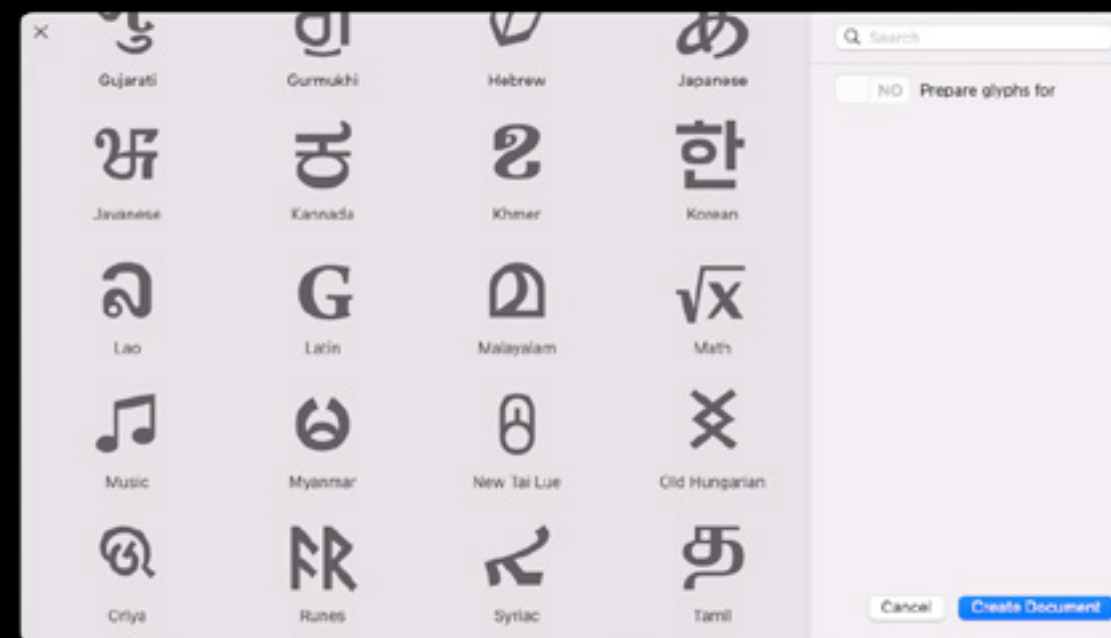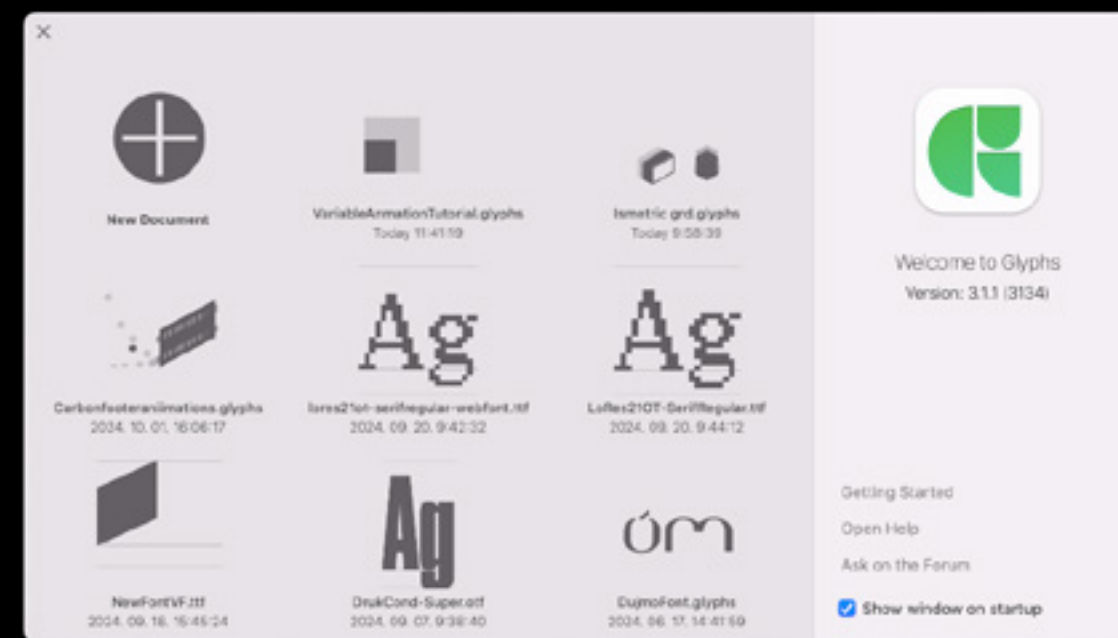You can think about **variable masters as key frames** of an animation.

Read more:

Introduction into Variable Fonts https://fonts.google.com/knowledge/introducing_type/introducing_variable_fonts
Introduction into Glyphs3 Variable Font making https://glyphsapp.com/learn/creating-a-variable-font
Introduction into OpenType https://medium.com/variable-fonts/https-medium-com-tiro-introducing-opentype-variable-fonts-12ba6cd2369

2.

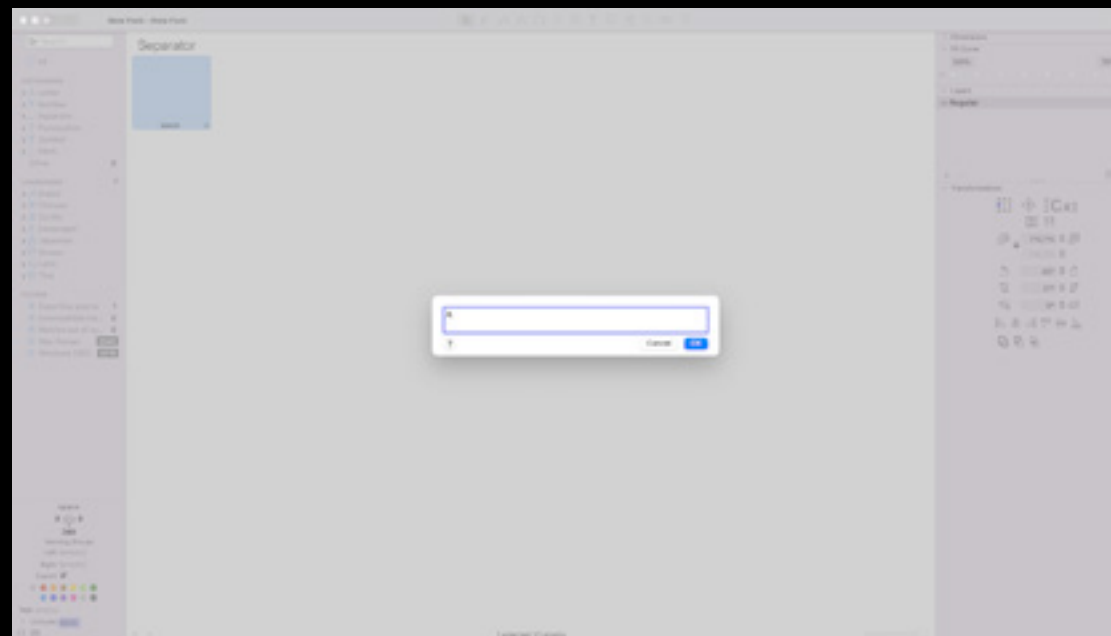## Let's go. Set up a new document in Glyphs3.

1. **Click > New Document**

2. Choose any alphabet. I will use Latin in this tutorial.

3. This is your empty document. You will collect your animations here.

4. You can add a new character by **Click > Glyph > Add Glyphs..** Type 'A' for instance.

3.

Glyphs3 is originally designed to draw letter—shapes and type—setting. Letters are constructed of lines, curves which create closed shapes. You can find these **basic tools** for drawing **in the top bar**.
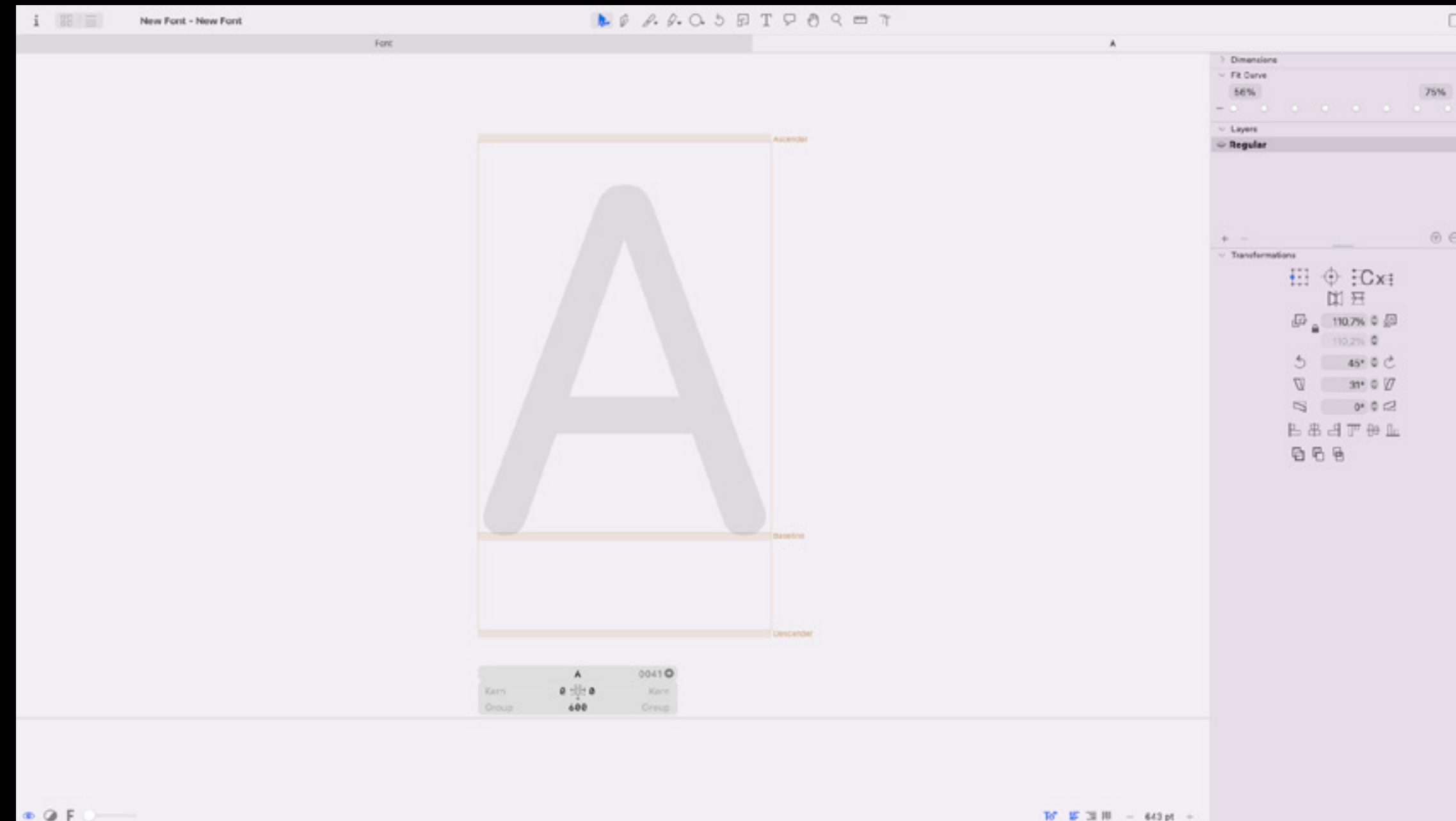

On the right side of the interface you can find the basic **transformation tools** such as Position, Anchor—point, Align, Rotate, Reflect. (These might come handy while animating).

On the bottom you can see a bar with metrics that are used in Type—design (Kerning). In our case we will only use it to **set the width of your canvas**.
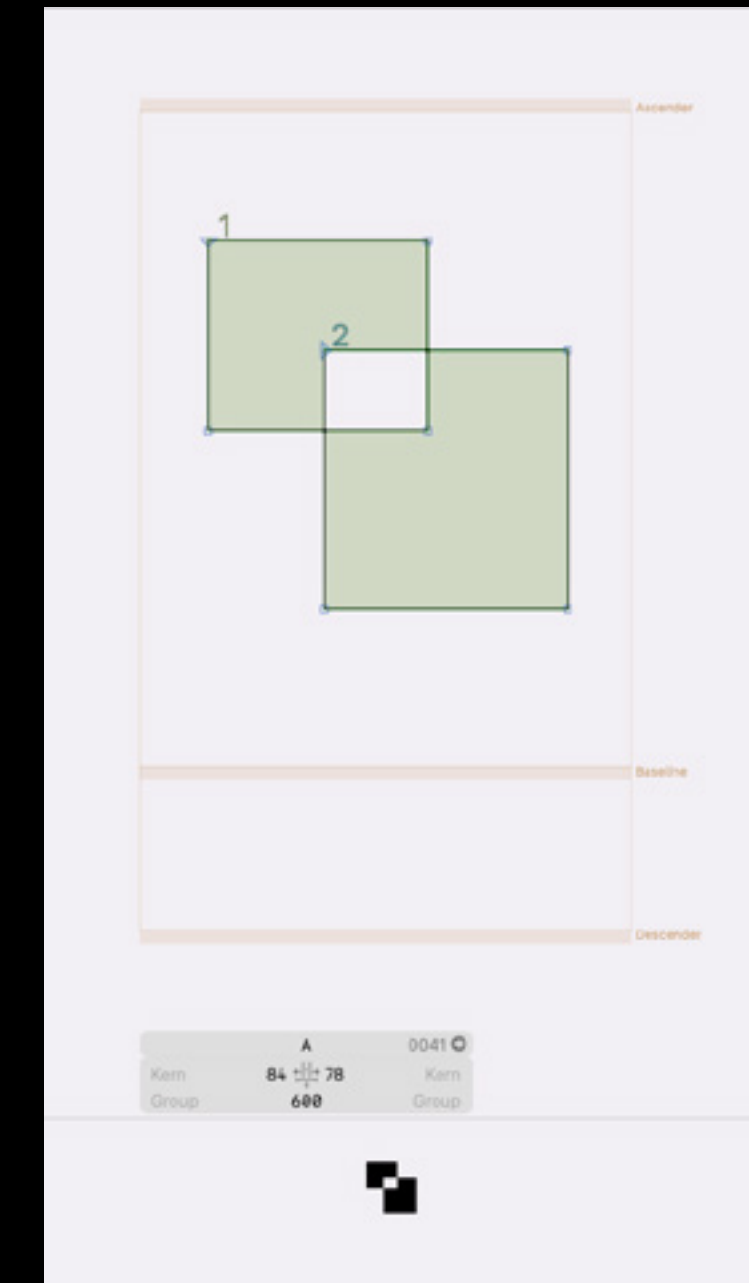
4.

Exercise:
1. Draw some shapes using the basic tools.
2. Try out the transformation tools.
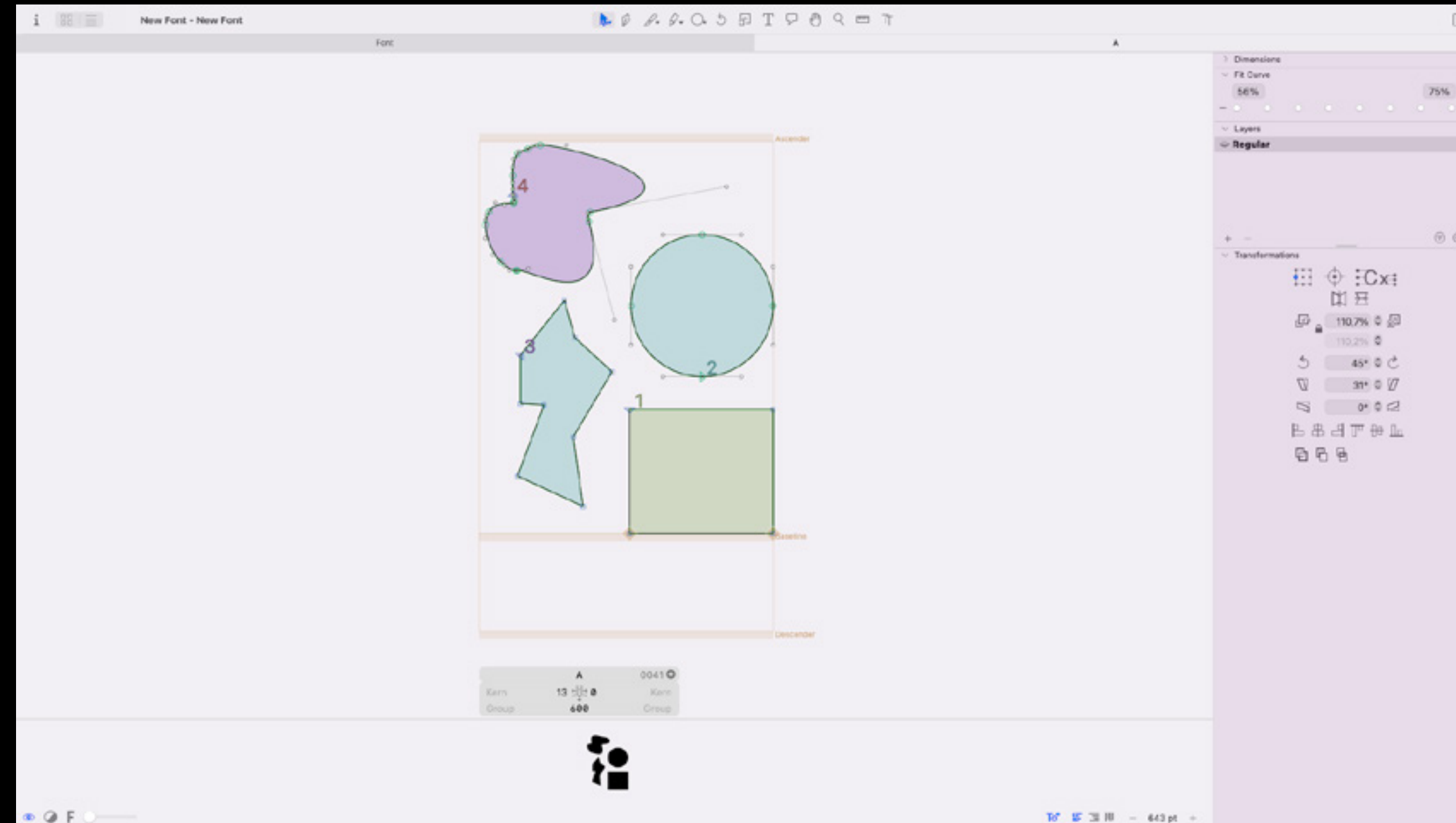
When you draw something on the canvas **there is a small preview on the bottom** of the interface. This will show you how the shape will look when you export it as a font file.

Try to experiment more and see how the preview changes when drawing overlapping shapes and changing the path direction. **Do you see any strange things???**

5. Let's set up two masters (key frames) for the variable axis animation.
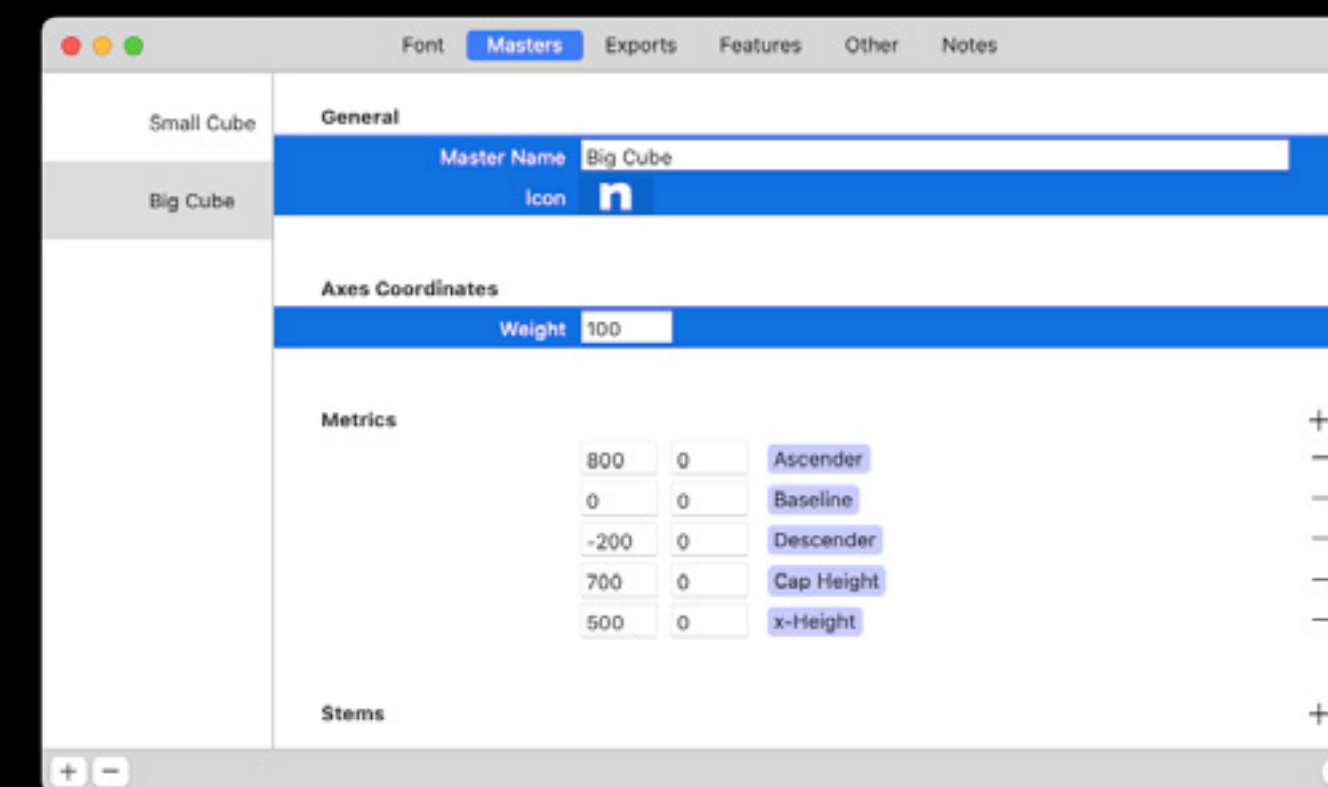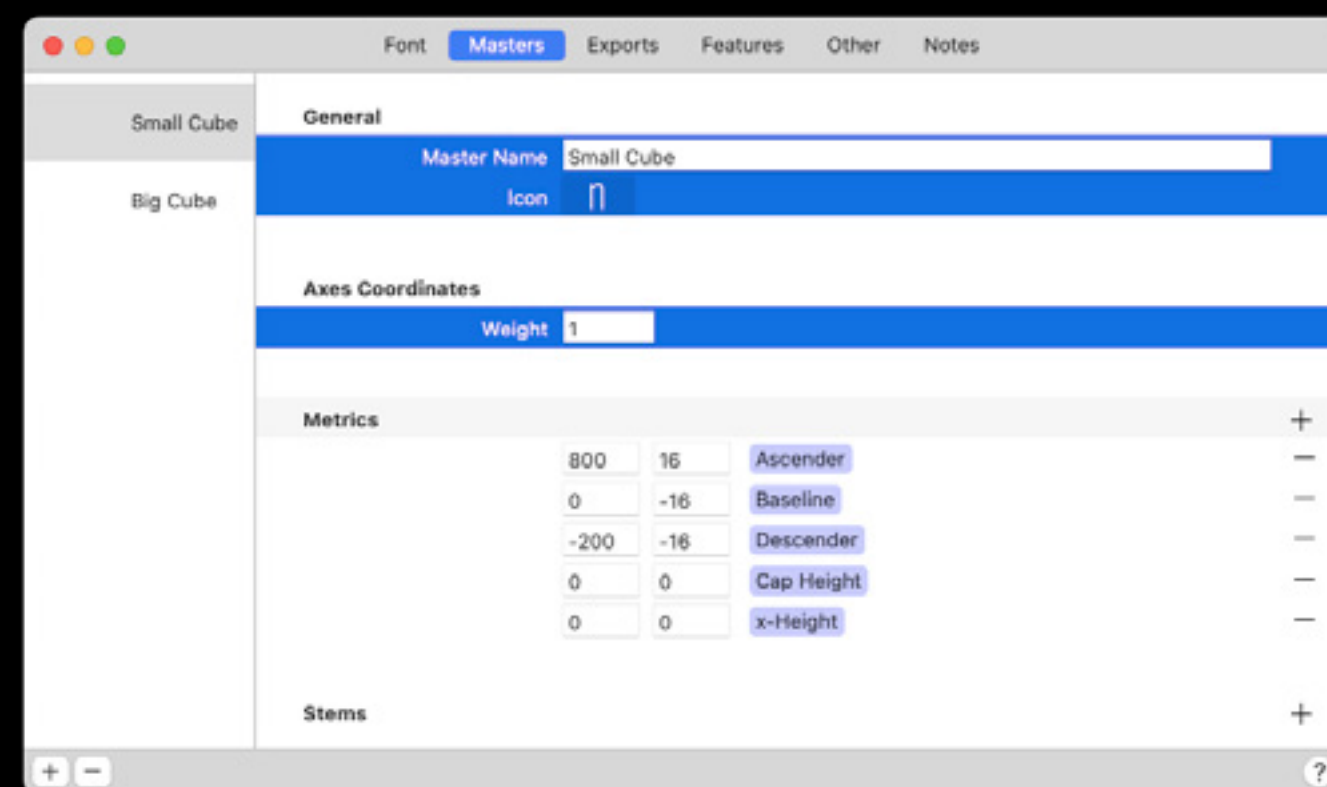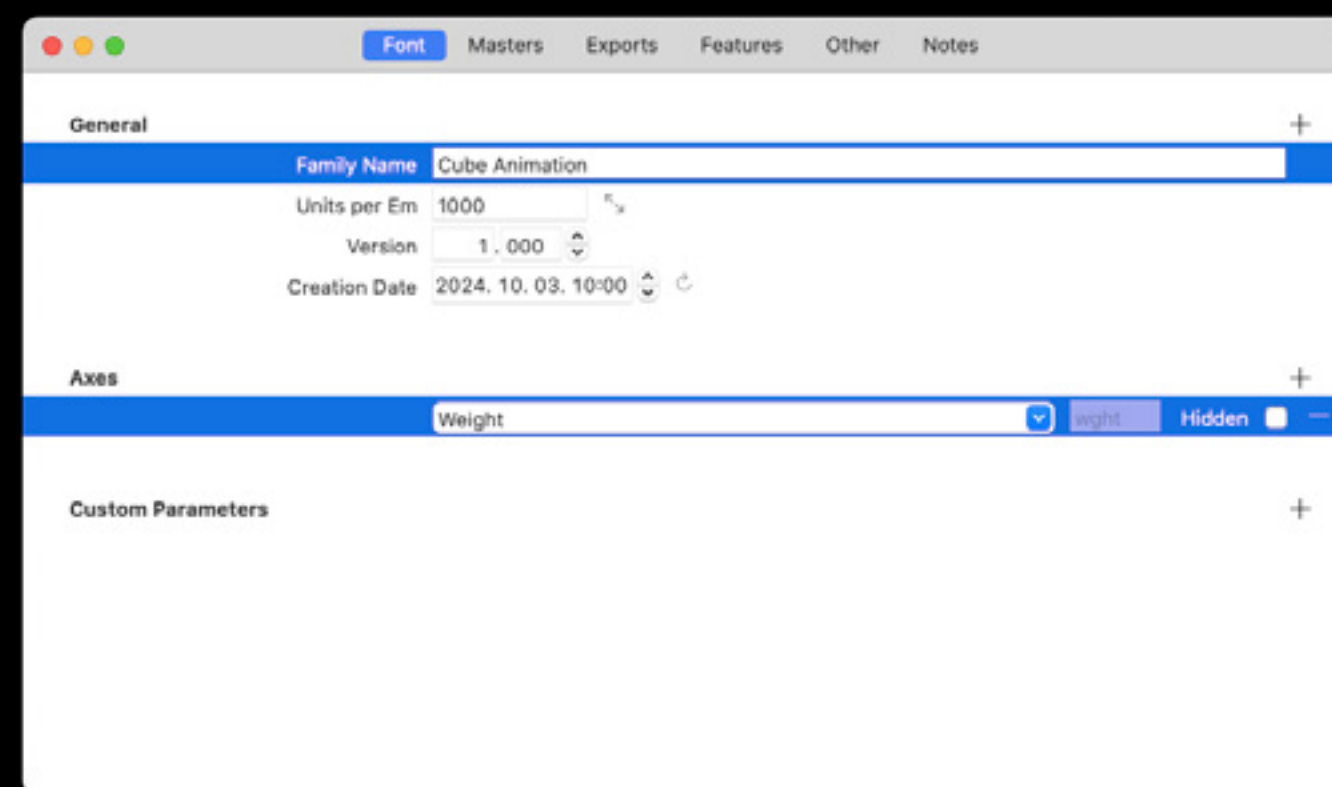
1.

Now you can set up the masters which will serve as the key frames in our variable animation. You can reach the 'Font Info' menu by **Click > i** on the top left corner or using the short cut **CMD + i**.

Read more:
Variable Axis https://fonts.google.com/knowledge/glossary/axis_in_variable_fonts

2.

Here you will find several global settings. Under the 'Font' menu point you can change the name of your file and add more axes and custom parameters.

Click > '+' (Next to Axis)

3.

The 'Masters' menu point is where you can add masters aka keyframes to your sketch.

Click > '+' > Add Master

4.

When you create a New Master always name them so later they make sense for you when making a complex animation. At the **Axes Coordinates** give **different values for each** masters' (key frames) **Weight** (this will allow you later to access to these values as 'font-weight' in CSS and animate them)
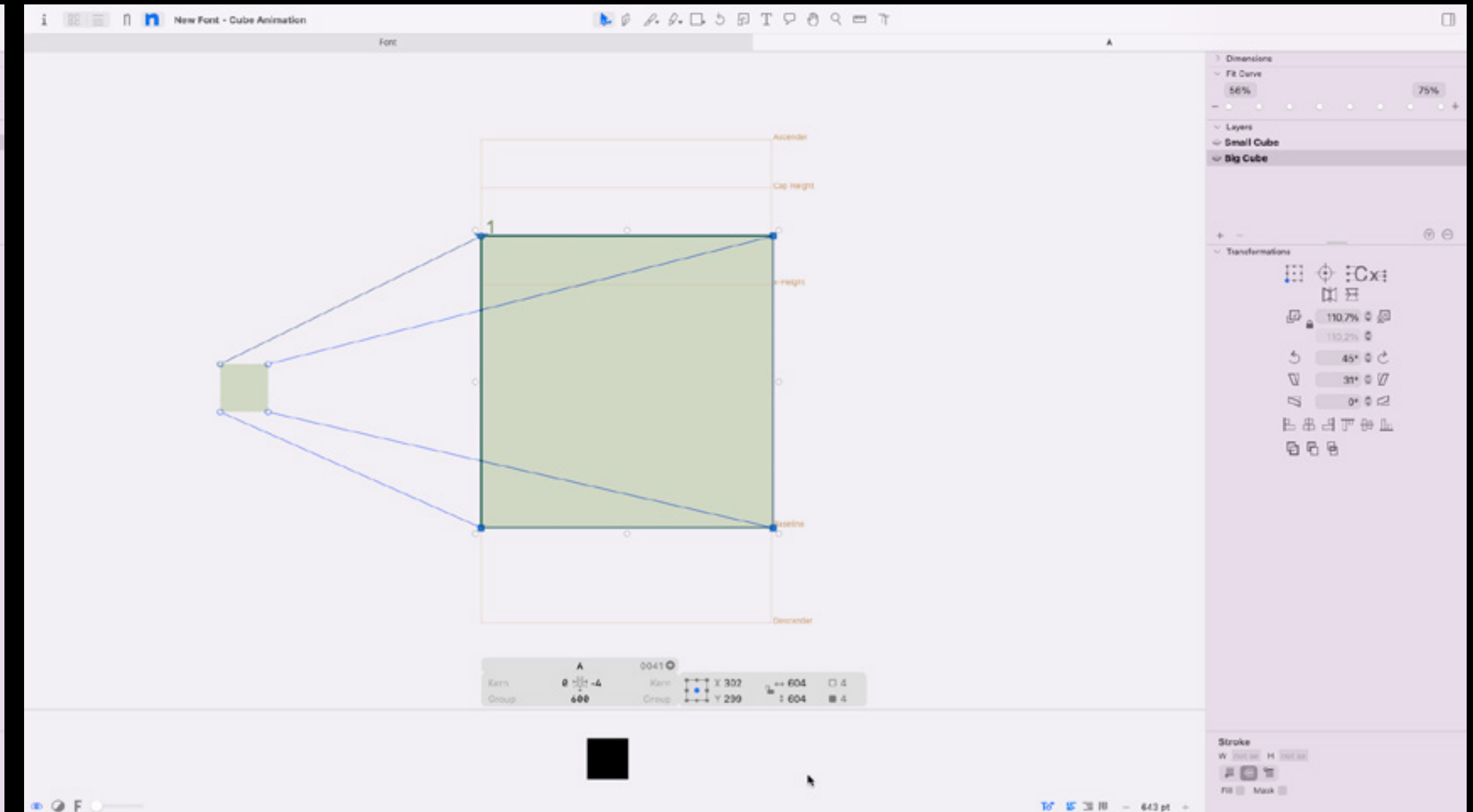
6.

Hurray You made it!!! Now you can get creative and draw your first key frames. :)
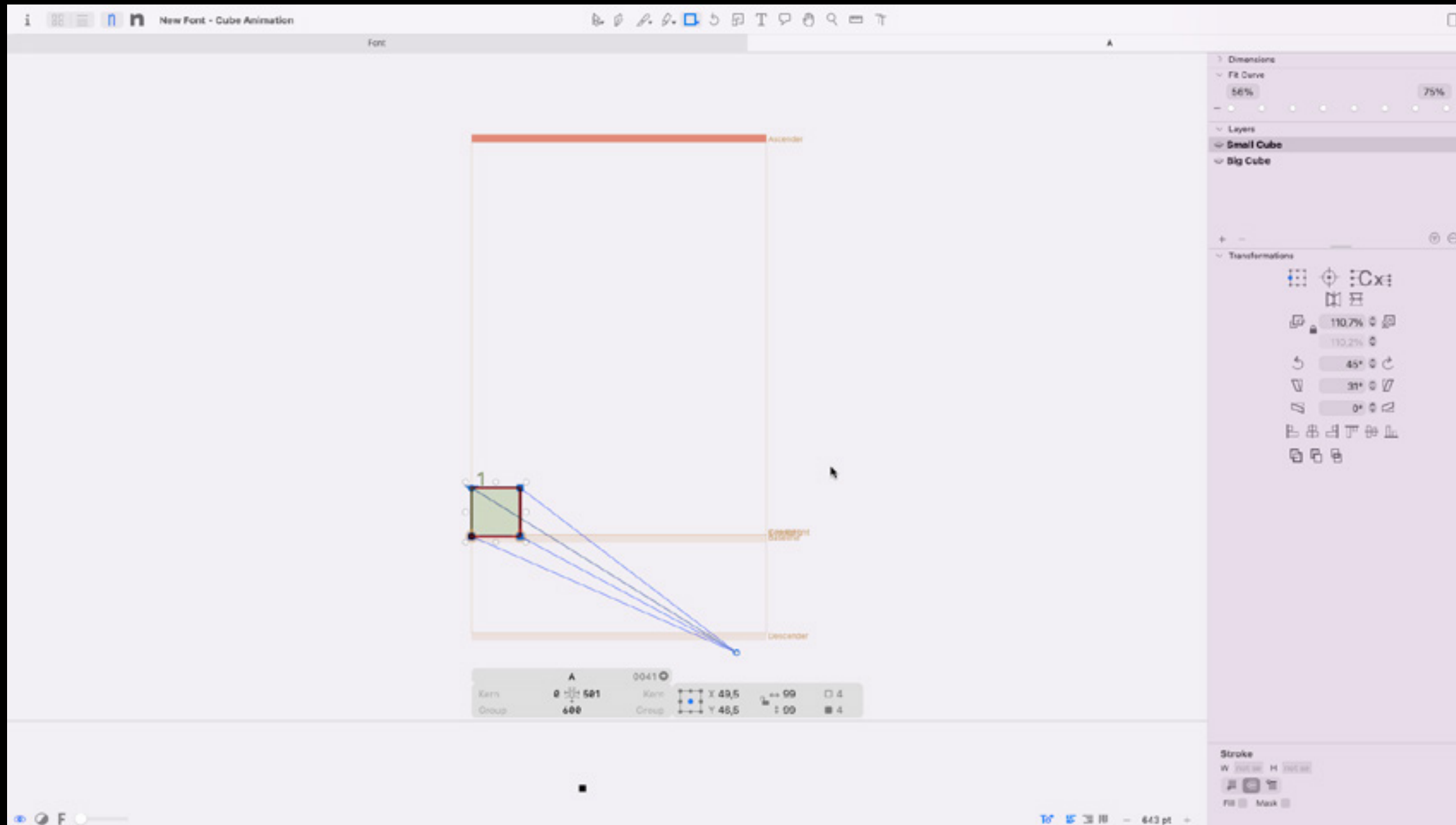
1.

Now you can switch between the two masters (key frames) by **Click > n** icon on the top left corner or hold **CMD + 1 or 2 or 3 …**

Let's draw a small rectangle on the first master and select everything with **CMD + A**, then **CMD + C**.
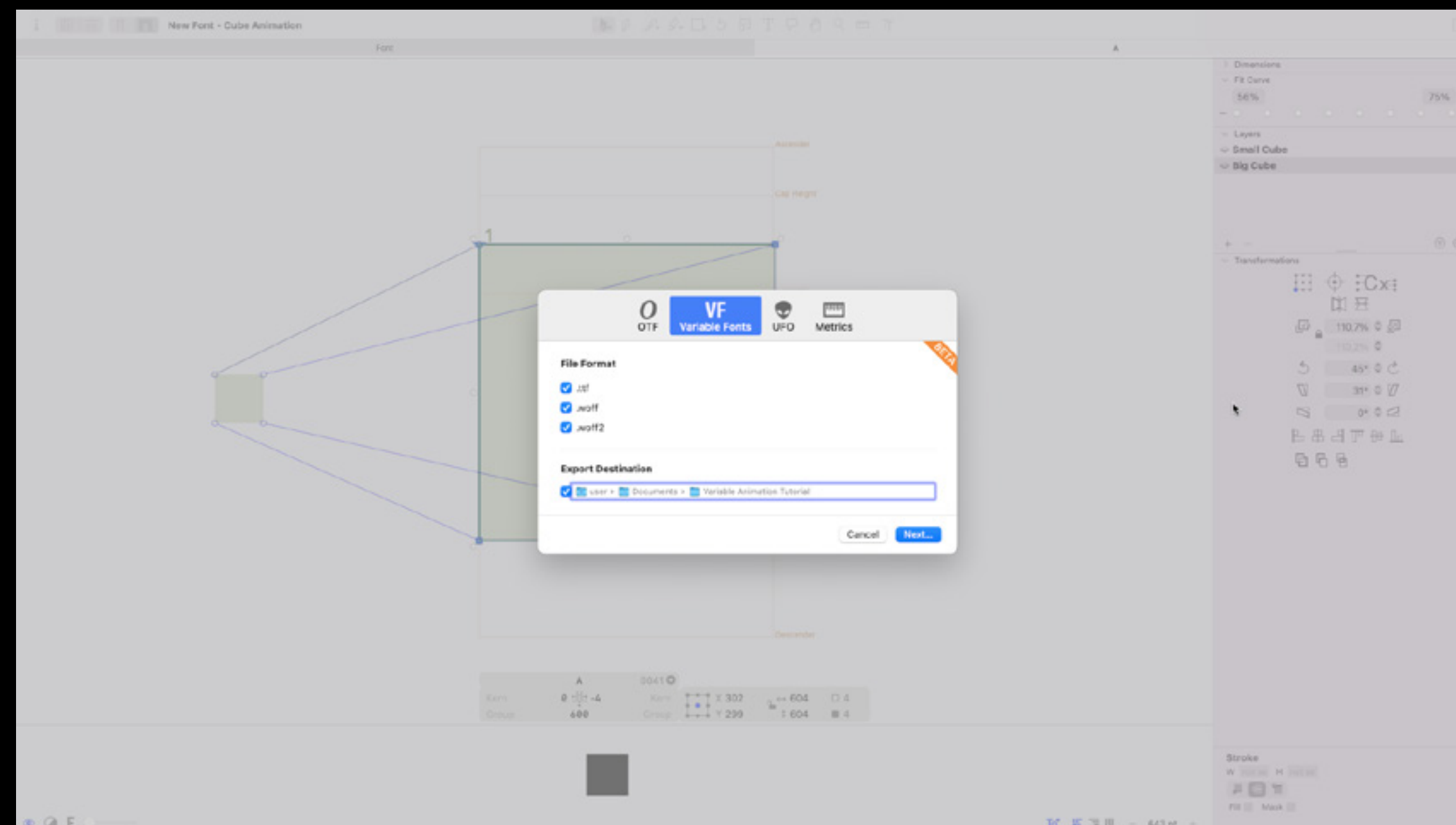
2.

Switch to the second master.

3.

Place the same rectangle **CMD + V** on it, and scale it up by using the **Scale** transformation tool on the right or simply drag the corner of box around the selected shape.

7.

It's time to export.

1.

Click > File > Export

2.

Chose VF Variable Fonts and tick .woff2 (this is the newest font specification which means this has the smallest compression and the fastest loading).

3.

Create **a new folder** (we will use this folder to run a small piece of HTML, CSS in your browser to see the master piece You've created).

8.

Create an .html file in the folder and copy paste the code.

1. **Open the folder in VS Code** by dragging the folder on the interface.
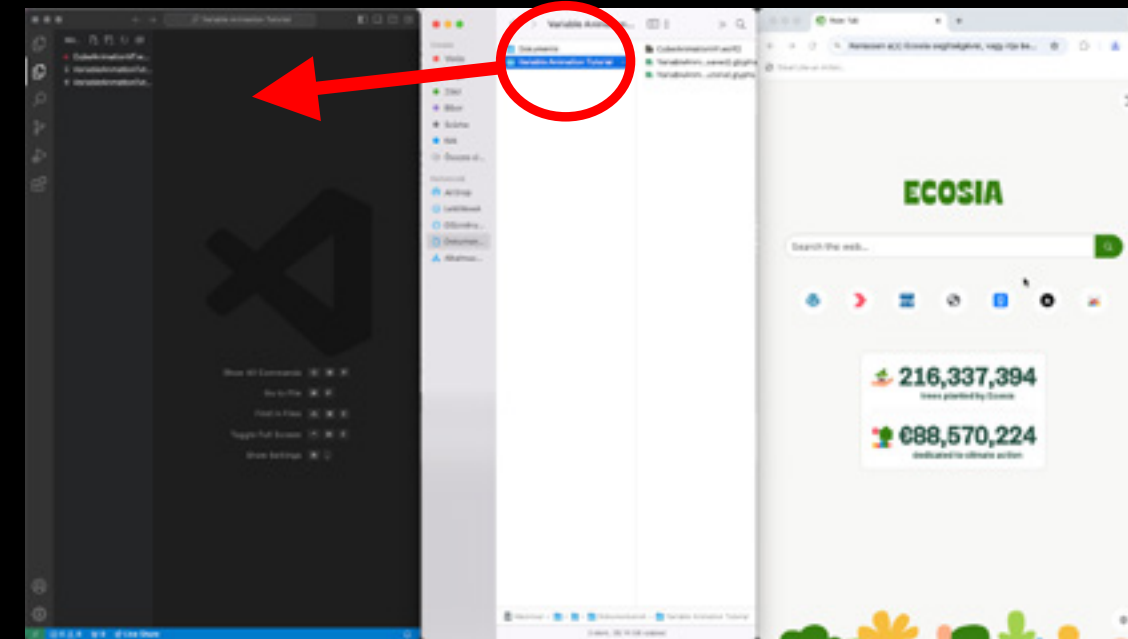
2. Create a New File by **Click > New File** (Paper icon with a + sign)

And **name it** Animation_Test.html
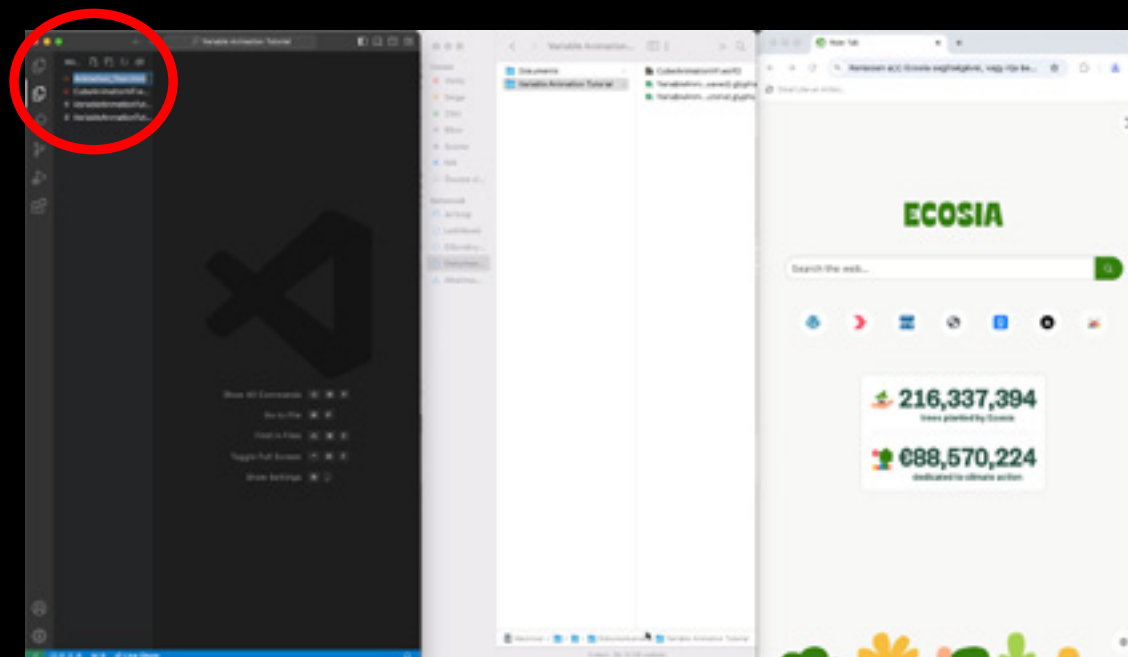
3. And voila... **You have an empty .html file.**
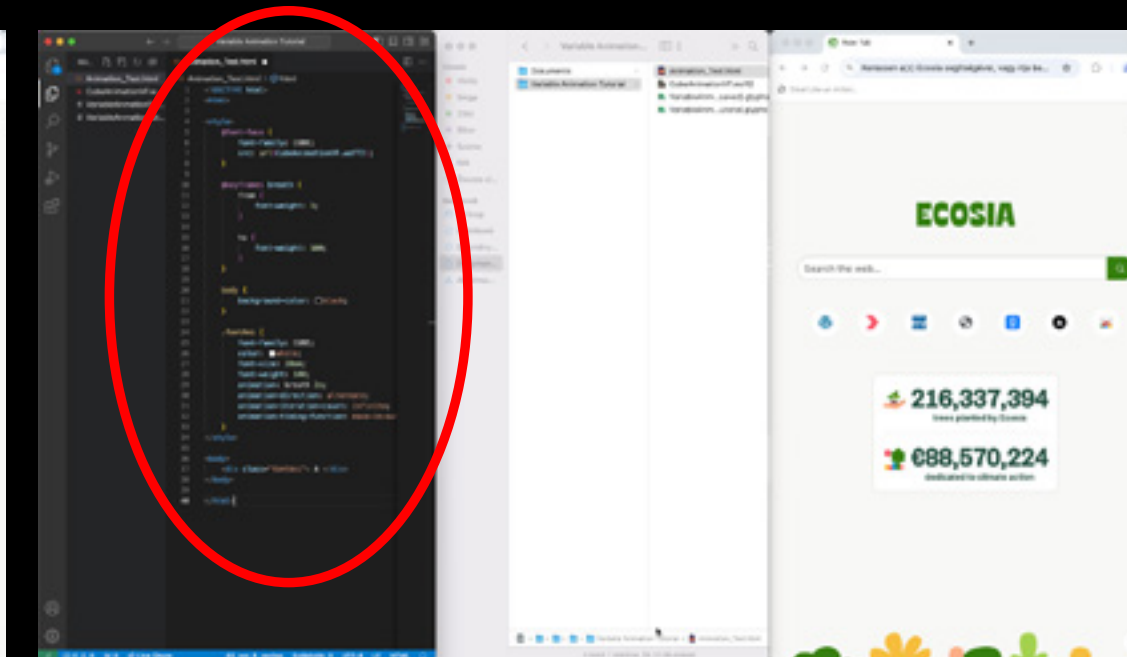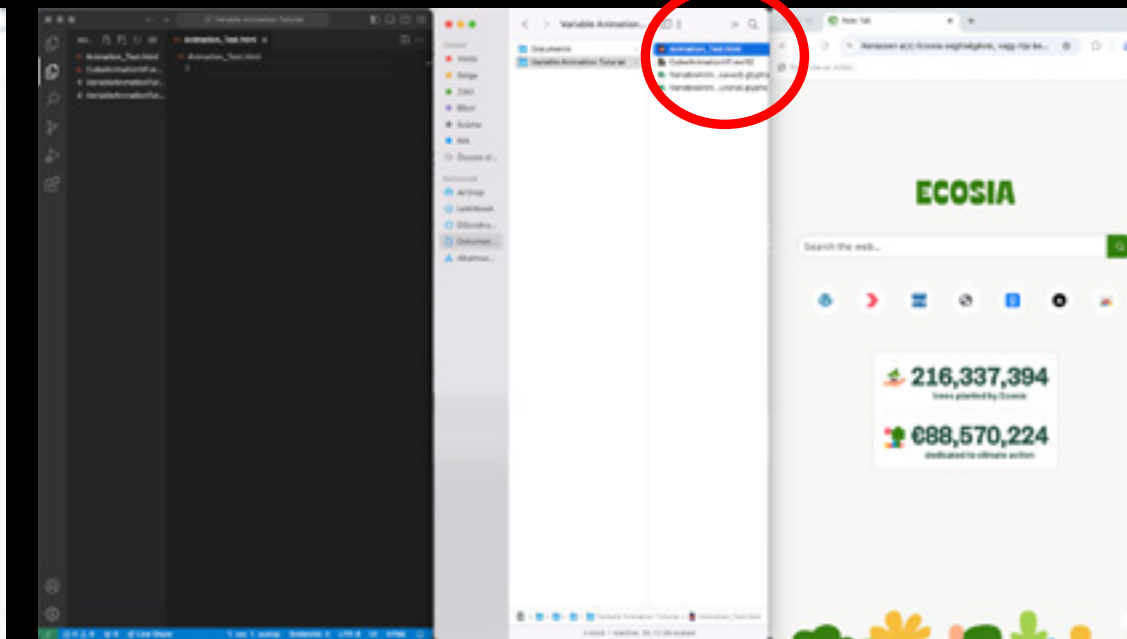
4. **Copy and Paste** the code from this slide into the newly created .html file and hit **CMD + s** to save it.

```
<!DOCTYPE html>
<html>

<style>
    @font-face {
        font-family: CUBE;
        src: url(CubeAnimationVF.woff2);
    }

    @keyframes breath {
        from {
            font-weight: 1;
        }

        to {
            font-weight: 100;
        }
    }

    body {
        background-color: black;
    }

    .fontAni {
        font-family: CUBE;
        color: white;
        font-size: 20em;
        font-weight: 100;
        animation: breath 2s;
        animation-direction: alternate;
        animation-iteration-count: infinite;
        animation-timing-function: ease-in-
out;
    }
</style>

<body>
    <div class="fontAni"> A </div>
</body>

</html>
```

9.

See the result.
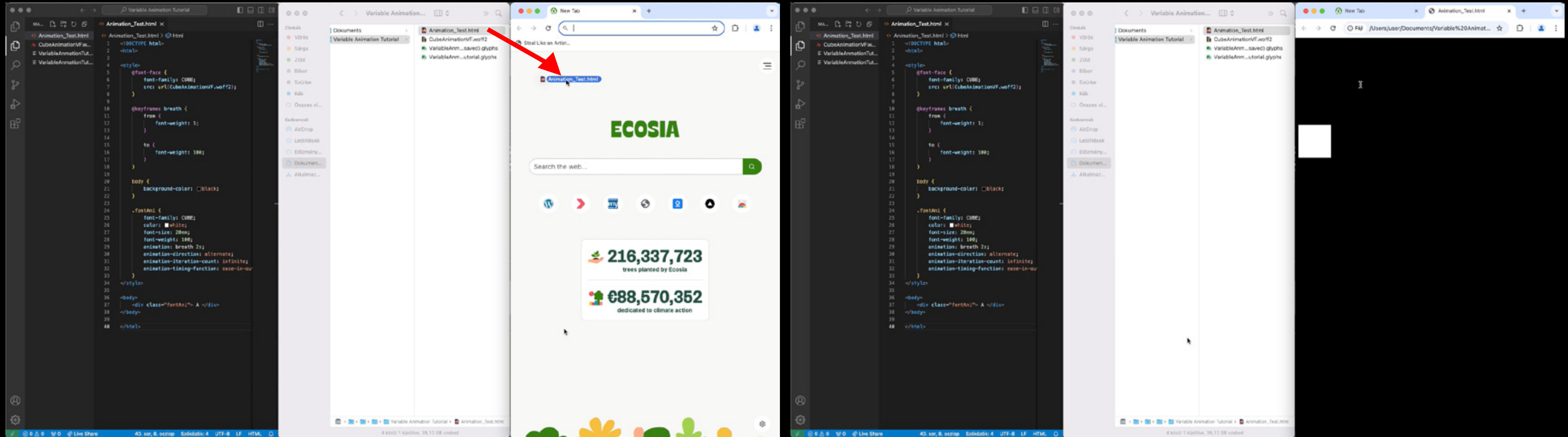
1. **Drag and drop the saved .html into your browser.**

2. It will play the animation automatically. **CONGRATULATIOINS!!**

If something is not working or **you want to know more** about how to control your animation in CSS **go to the next slide.**

10.

Explaining the code.

This is the style sheet aka CSS where you can import the font file create animation and give styles to it.

@font-face {} is where you can import the font file we exported earlier and name it so it is easier to reuse in the code.
IMPORTANT: use exactly the same name of your font file within the:

src: url(**CubeAnimationVF.woff2**);

@keyframes scale {} is where you name your animation and access the font weight properties as key frames we have created before in Glyphs3.

body {} is where you can change the color of the background of instance.

.fontAni {} is where you can change the style of the font or shapes of your animation and have access to manipulate the animation properties as well (timing, easing ect.)

This is the HTML body **where you can type the variable characters** from the font file which represents a loop of animation in this case.. to be continued ...

Read more:

CSS https://www.w3schools.com/css/
@font-face https://www.w3schools.com/cssref/css3_pr_font-face_rule.php
CSS animation https://www.w3schools.com/css/css3_animations.asp

In the next tutorial we will animate a cube, distorting its' shape.
And delve into setting up custom
canvas in Glyphs and learn some of
the basic animation principles.

Have fun creating!