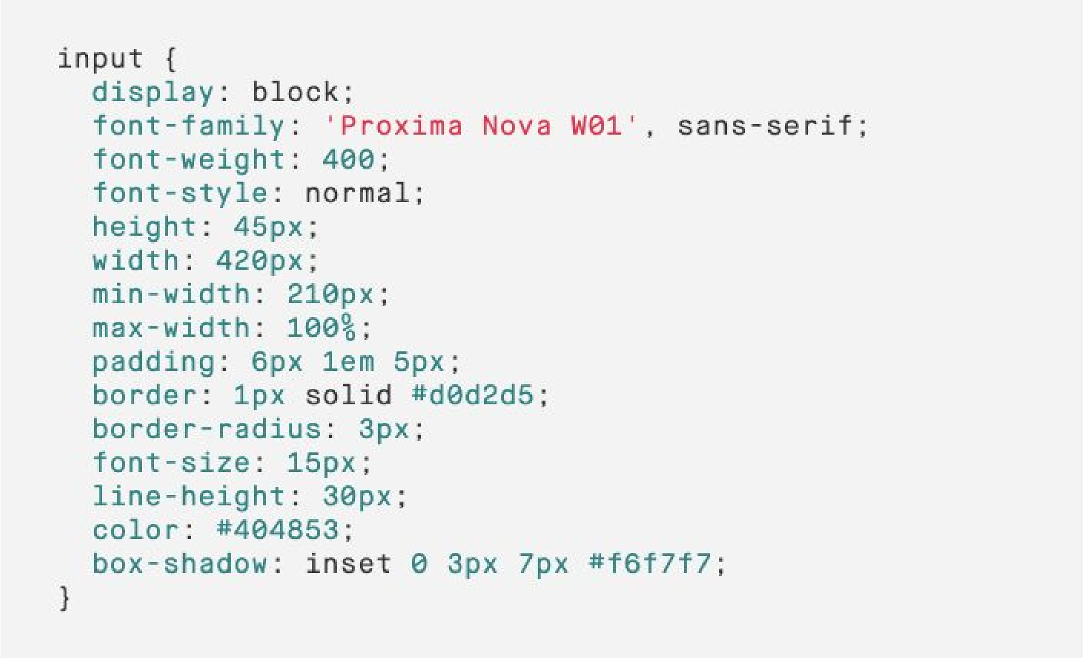
An Easy Guide to Understanding CSS Animatable Properties

Have you ever clicked through a website and wondered how the developers were able to use different fonts, colors or even placements of images? That’s CSS (Cascading Style Sheets). While HTML is all about the structure and content of a web page (aka, the ‘content layer’), CSS is about style and appearance. It’s used to control the visual aspects of the content on a page - from fonts and colors, to size, animations and more.

Now let’s take it up a notch. Have you ever scrolled through a web page and wondered how the developers were able to make the background color *change*, the font size get *bigger* or *smaller*, or *filter* an image from its original colors to black and white? That’s by using CSS Animatable Properties. In this post, we’ll explain the basics of adding animation to your elements and provide some examples as well.

What are CSS Animatable Properties?

Properties are used to specify how you want your elements on a web page to look to a user. These are specifications like, font-family, height, font-size, margins and more.



```
input {
  display: block;
  font-family: 'Proxima Nova W01', sans-serif;
  font-weight: 400;
  font-style: normal;
  height: 45px;
  width: 420px;
  min-width: 210px;
  max-width: 100%;
  padding: 6px 1em 5px;
  border: 1px solid #d0d2d5;
  border-radius: 3px;
  font-size: 15px;
  line-height: 30px;
  color: #404853;
  box-shadow: inset 0 3px 7px #f6f7f7;
}
```

Some CSS properties, however, are *animatable*, meaning that their values can be made to change gradually over a given amount of time.

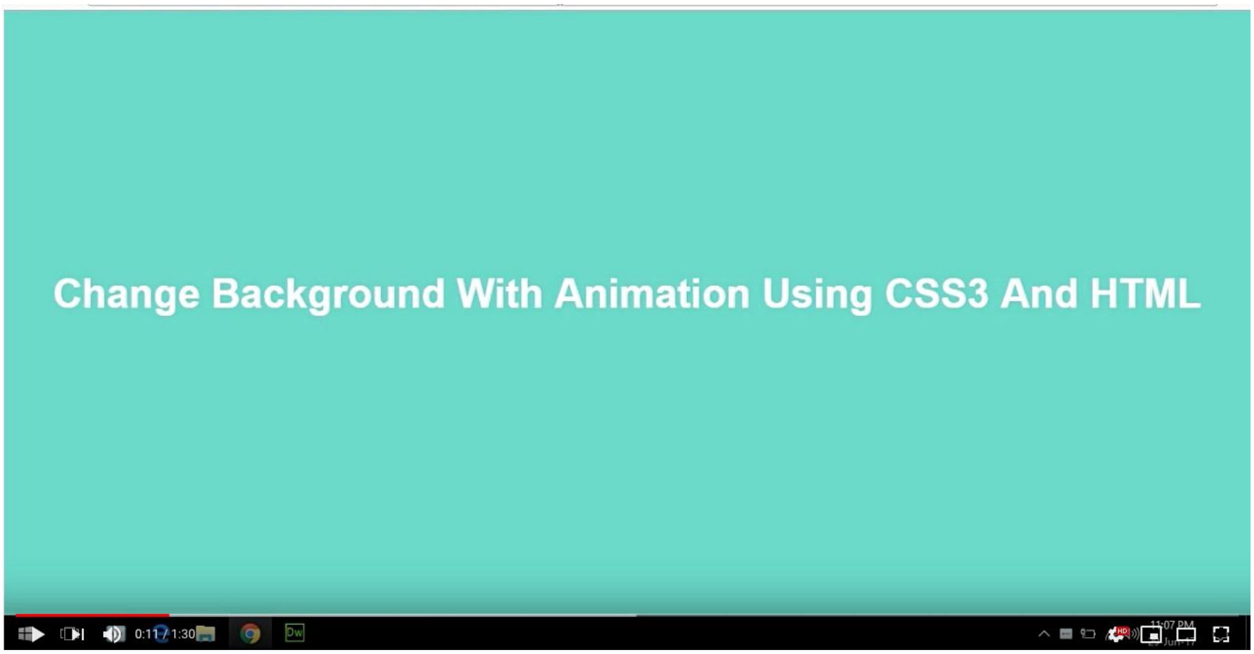
Let’s look at some examples for reference.

Examples of CSS Animatable Properties

While CSS supports nearly 200 properties, only a large handful can be animated. Here are three of the most common:

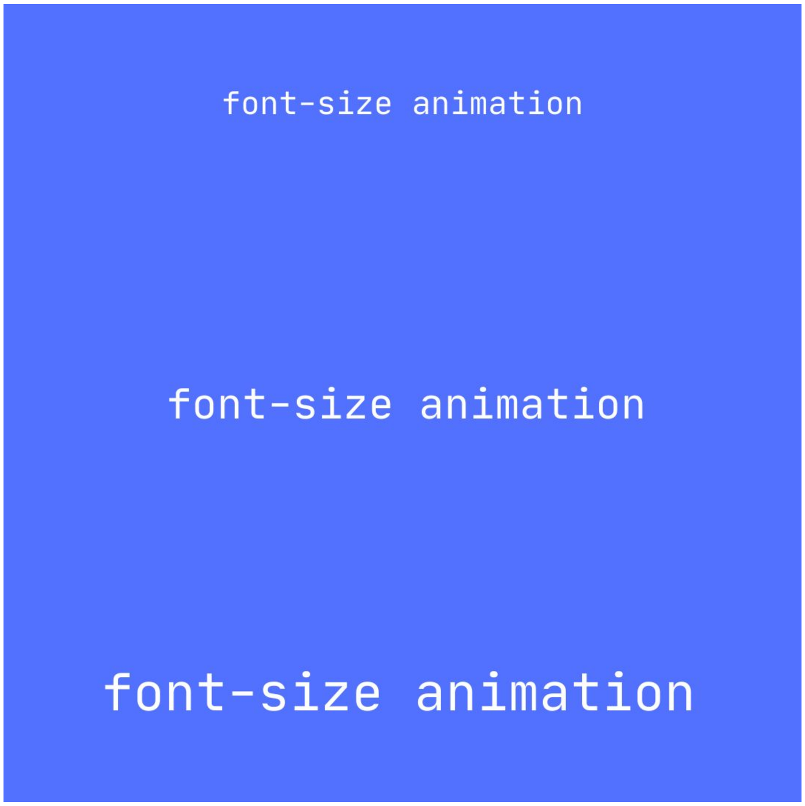
Animation of the background-color property

The background-color property sets the background color of an HTML element. Whether it’s the full body of a web page, or just a small section, this property is used to set the initial value that the user will see. By using animation, you are able to gradually change the color of the element you determine over a set duration period. You can try it out for yourself here.



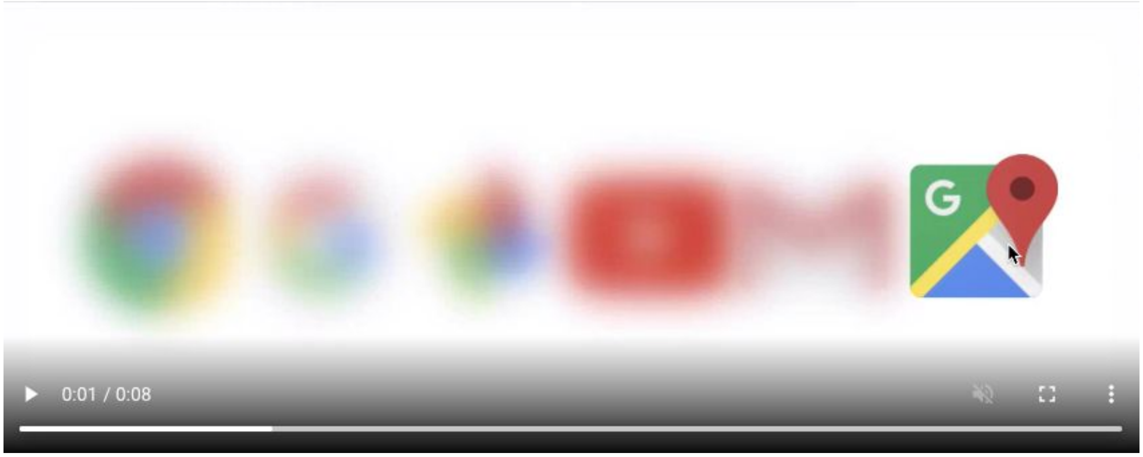
Animation of the font-size property

The font-size property is used to set the size of a header, paragraph, list item or other HTML element. CSS Animation allows you to gradually make the font-size bigger or smaller than its initial set value. Give it a try in this example.



Animation of the filter property

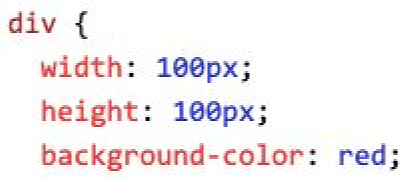
The filter property defines the visual effects used on an element, like an image. The most common filter options are ‘blur,’ ‘brightness,’ ‘contrast,’ ‘grayscale,’ and ‘opacity.’ These can be used with animation to change the initial color of the image you are using on your web page. See what it looks like to filter a full-color image of a pineapple to grayscale here.



Now that we've seen some examples, let's learn how you can utilize them in your next web page.

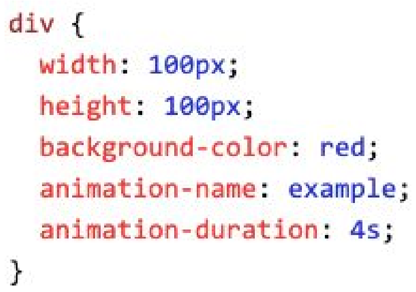
How to use CSS Animatable Properties in Your Own Code

Before you try to implement CSS Animatable Properties, you have to give the property of the element you want to animate an initial value. This value will attribute to how the element looks when the web page first loads. In this example, we will use CSS to animate the background-color property. But before doing so, we need to set its initial value to 'red.'



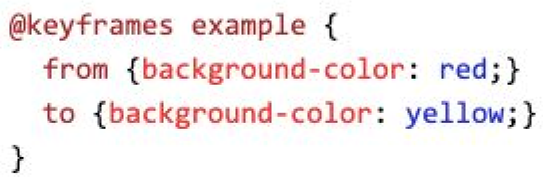
```
div {
  width: 100px;
  height: 100px;
  background-color: red;
```

Next, create an animation-name and animation-duration which will specify the name of the animation and time that it should take the animation to complete one cycle.



```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

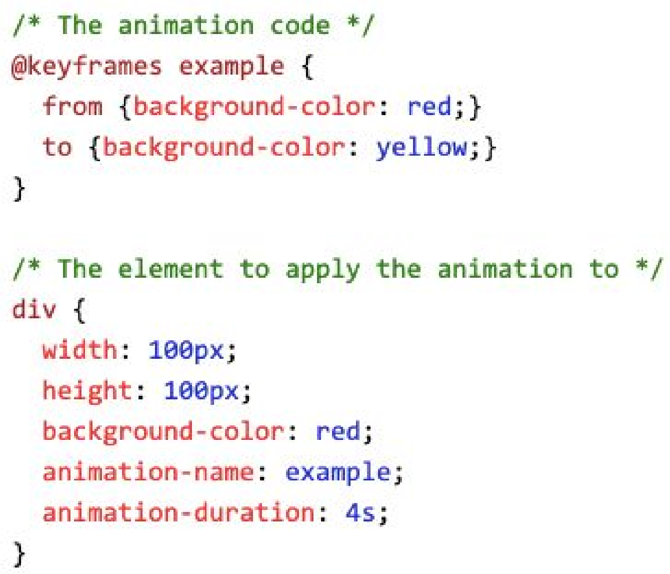
Now it's time to set the *@keyframes*. These rules determine what styles the element you want to animate will have at certain times. Meaning, these will hold the new values you set for the property you want to change.



```
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
```

Here, you will see that we are using animation to change the background-color from its initial value of 'red' to 'yellow.' It's important to note that you need to use the same name for the keyframe that you did for the animation-name. This is what tells the keyframe what element and animation to look for and manipulate.

Here is an example of it all together:



```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

The name and duration of your animation are not the only property values you can set. To see the full list of values with detailed explanations, check out w3schools. You can also learn more about how to optimize @*keyframes* to their full potential here.

Best Practices to Remember

While having animation on your web page can be a fun trick to use, it’s important to keep in mind the user experience when choosing what to implement. As you set styles, there are two things you'll need to keep in mind - creativity and usability. On the one hand, this is an opportunity to be creative by choosing colors, fonts, images and other presentation elements for your web page. On the other hand, the page should be easy to use - you don't want styling to get in the way of usability.

The decisions you make about styling your web page should be driven first and foremost by what you know about your user and the experience you're trying to give them. Remember, your web page is a representation of your brand, product and/or persona.

Apart from thinking intentionally about animation, here are a few other styling areas to consider as well:

Color palette

Choose your color palette wisely. While you may decide that a white background with black text is exactly what you want, chances are that you can deliver a better user experience by using colors throughout your page. There's a ton of resources on color theory out there, but if you're looking for a quick lesson, this HubSpot guide is an excellent resource to guide you through choosing a color palette that fits nicely with the goal of your site.

Font families

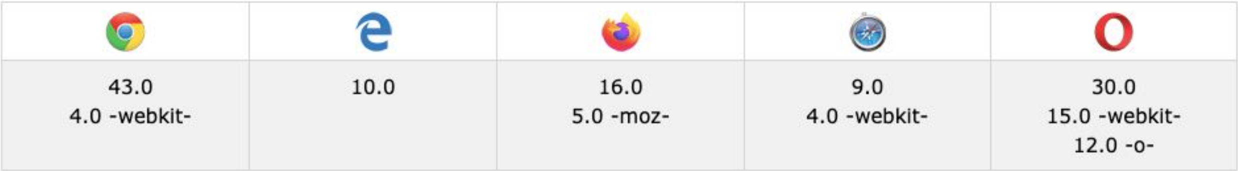
It’s also important to choose your fonts wisley. Your choice of font family can go a long way to establishing the feeling of your site. Some fonts scream "Fun!" or "Cool!", while others are more neutral. Your choice of font should be driven by two questions: first, which fonts will contribute to the overall feeling you want this site to have? Second, which fonts will be readable for your users? As with colors, you'll need to balance these two needs. If you'd like to read more about choosing fonts for your site, this blog post is an excellent starting point.

Responsiveness

Don't forget to make your web page responsive. It should look great and be fully functional on mobile, tablets, and desktop. More likely than not, that means using media queries and a responsive grid system. The last thing you want is for a user to click to your page and discover that it’s unreadable or unusable.

Ready to give CSS Animatable Properties a chance? Give them a go and try them out! See what works for you and makes the most sense for your web page.

Important to note: The browsers and versions shown below are the earliest versions that fully support CSS Animations.



43.0 4.0 -webkit-	10.0	16.0 5.0 -moz-	9.0 4.0 -webkit-	30.0 15.0 -webkit- 12.0 -o-