CSS keyframes

In this reading, you will become acquainted with the @keyframes rule and you’ll learn how to use it with the animation property in CSS. The rules covered so far are for alignment and styling of web pages using CSS. Keyframes are a type of at-rule which are represented by the '@' suffix. At-rules are statements inside CSS that describe how to behave or perform certain actions. In line with that, keyframes are defined as '@keyframes' in the CSS code. @keyframes are part of the animation sequence and help in defining the steps inside it. Imagine an object on your web page moving from point A to point B. You can use the transition and transform properties to do that, but animation sequences are used to accomplish more complex behaviors in an easier way.

**from{} and to{} keywords and using percentages(%) syntax**

}

}

to {

property-a: value-b;

}

@keyframes animation-name {

from {

property-a: value-a;





The 'from' and 'to' keywords are used inside the @keyframes rule to mark the transition of one or more properties and can be seen as the start and end points of that transition. As can be seen from the syntax, the values of ‘property-a’ changes from ‘value-a’ to ‘value-b’. To expand on the use of ‘from’ and ‘to’, the @keyframes allows you to add more steps to your animation by using a percentage that represents the completion of the animation.

100% {transform: rotate(50deg);}





The different percentages used in the example demonstrate the progression of the animation. Note that it doesn’t have to be the same property that you modify in these steps. That's not possible using the transition property, but you can do it with @keyframes. This flexibility is what makes @keyframes so powerful. Another advantage is how these animations can also loop infinitely, run forwards, reverse and alternate.

@keyframes are tied in with the animation-name to which they are going to be applied. To give an overview of the animation property, it consists of other sub-properties. Of these, animation-name and animation-duration must be defined while others such as timing-function, delay, direction, fill-more, iteration-count and so on can be added.

Animation property shorthand:

The shorthand for the animation property consists of the following properties with their default values:

* animation-name: none
* animation-duration: 0s
* animation-timing-function: ease
* animation-delay: 0s
* animation-iteration-count: 1
* animation-direction: normal
* animation-fill-mode: none
* animation-play-state: running
* animation-timeline: auto

If the values of any of these are not defined, you should assume that they are the default. Of these, the first property of the animation-name is what's used to tie it to the @keyframes rule.

Let’s now examine an example of how you can use @keyframes and the animation property.

**Animation example**

HTML code:

<body>

<div class="box"></div>

</body>

</html>





CSS code:

body{

padding: 30px;}

.box{

background-color: lightcoral;

width: 50px;

height: 50px;

animation: myanimation 3s infinite ease-in;

}

@keyframes myanimation{

from{width: 50px;

}

to{width: 100px;

}





And this is the output:

In the example, the width of the object changes from 50 pixels to 100 pixels over a span of 3 seconds and loops infinitely afterward.

This is a very simple example of how you can use the animation property with the help of @keyframes rule to create your desired animation.

If you modify this code and change the animation rules to percentages, it will have the same output.

@keyframes myanimation{

0%{width: 50px;

}

100%{width: 100px;

}





Now, let’s add an intermediary step at 50%.

@keyframes myanimation{

0%{width: 50px;

}

50%{background-color: aquamarine;

    height: 20px;

}

100%{width: 100px;

}





Output:

The output is drastically different now after adding just two lines of code. Just like this, you can keep adding steps inside your @keyframes rule to make it even more dynamic and add the desired animation effects. Animation property and @keyframes rule can be used in very creative ways to enhance a web page.