



CSS Tutorial: Creating Animations & Keyframes | Web Development Tutorials #36



Show Course Contents (+)

[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

CSS Tutorial: Creating Animations & Keyframes | Web Development Tutorials #36

In this tutorial, we are going to learn about how to create different animations with the help of CSS. Creating animations with CSS is the most exciting part of the whole web development. We will make a new file as *tut36.html* and add a boilerplate for the basic HTML template. Then give the title as **Keyframes and Animations** in the `<title>` tag.

You will definitely appreciate the techniques shared in this tutorial about animations. Let us now write the HTML code for our website as follows-

```
<body>
  <div class="container">
    <div class="box">
      This is a box
    </div>
  </div>
</body>
```

Let us now style our *container* and *box* by adding some of the CSS as follows-

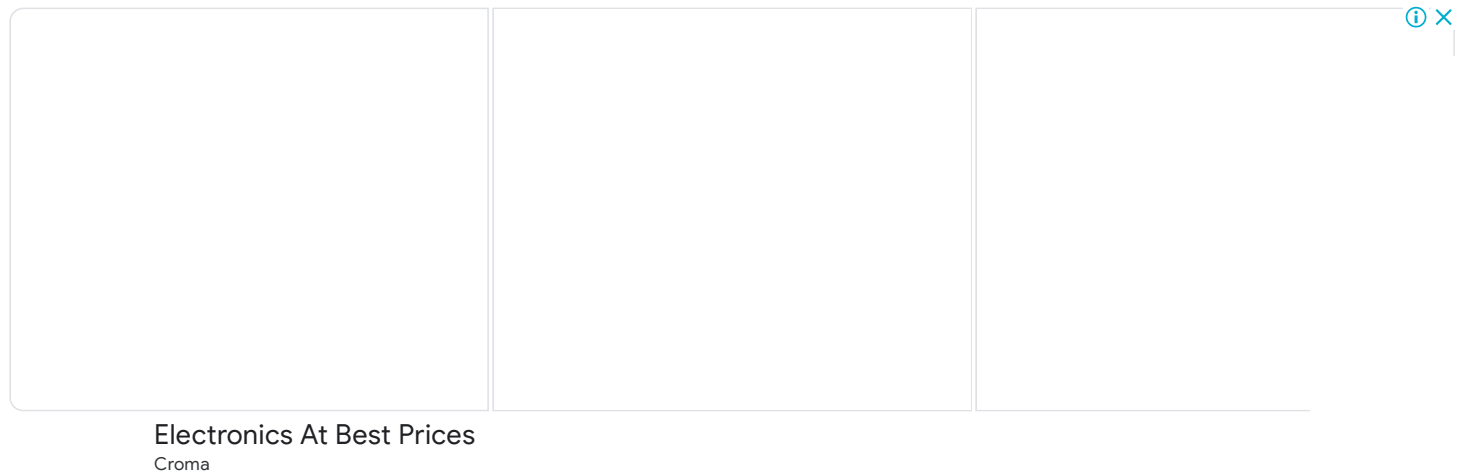
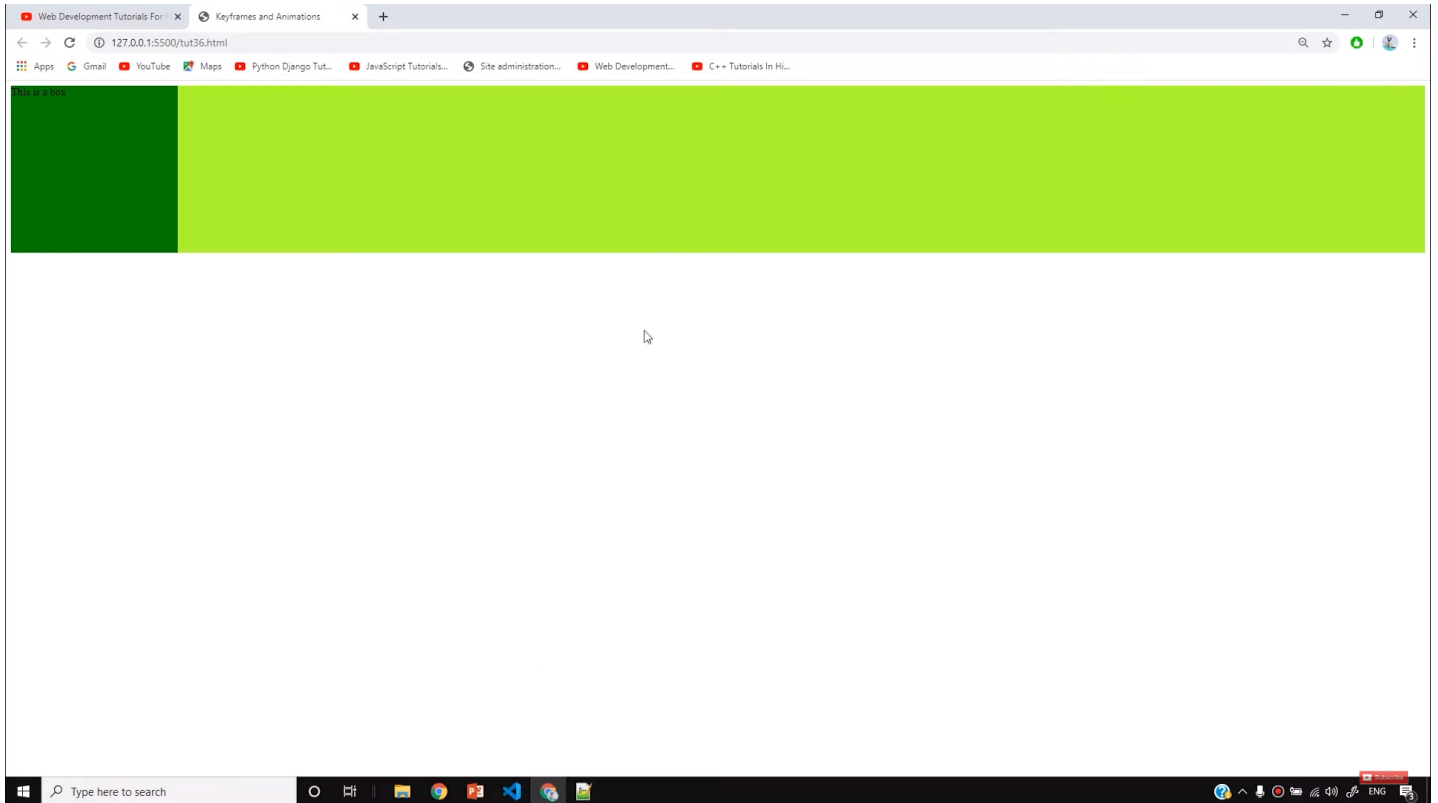
```
.container {
  background-color: greenyellow;
```

```
}
```

```
.box {  
    background-color: green;  
    width: 250px;  
    height: 250px;  
    position: relative;  
}
```

```
}
```

The output will be as follows-



The position of the box is set to be *relative* so that we can move it within our webpage.

For making our animation, we need to start by giving the **animation-name**. We can give any name here. It is just used to define our animation. The code for designing the animation is as follows-

```
.box {
  background-color: green;
  width: 250px;
  height: 250px;
  position: relative;
  /* animation-name: harry1; */
  animation-name: harry2;
  animation-duration: 8s;
  animation-iteration-count: 1;
}
```

In the above example, we are using the animation-name as **harry1**. The next property used is **animation-duration**. It is used to decide how long our animation will run. The last property is **animation-iteration-count**. It is used to decide the number of times the animation will run.

Now we will define the animation we made, i.e. *harry1* as follows-

```
@keyframes harry1 {
  from {
    width: 200px;
  }

  to {
    width: 1400px;
  }
}
```

The **keyframes** are used to make the animation. **From** and **to** are used to decide how the animation will move in the webpage. In the above example, we are moving the animation harry 1 from 200 px to 1400px. These types of animations are used to design scroll bars or progress bars on the webpage.

There are some other properties also to customize the animations like-

animation-fill-mode:

If we want to keep the last property applied to the animation then we can set the animation-fill-property as *forward* as follows-

```
animation-fill-mode: forward;
```

animation-timing-function:

We can define this property with three different values-

ease-in

After applying this, the animation will start slowly and becomes fast towards the end.

ease-out

After applying this, the animation will begin fastly and become slow towards the end.

ease-in-out

After applying this, the animation will start slowly, then become fast in the midway, and ends slowly.

animation-delay

It is used to define the time after which the animation will start.

```
animation-delay: 3s;
```

animation-direction:

This property is used to define the direction of the animation. For example, if we select it as *reverse*, it will move the animation in reverse direction.

```
animation-direction: reverse;
```

There is another method of creating animation apart from keyframes. For this, we will give the name as *harry2*.

```
@keyframes harry2 {  
  0%{  
    top:0px;  
    left:0px;  
  }  
  25%{  
    top: 250px;  
    left: 0px;  
  }  
  75%{
```

```
        top: 250px;
        left: 250px;
    }
    100%{
        top: 0px;
        left: 250px;
    }
}
```

In this way, we can create the animations in terms of different percentages. We can assign different values here that will occur when that percentage of animation will be completed. Keep practicing these animations and in the upcoming tutorials, you will learn about practical implementations of these animations.

Code as described/written in the video

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Keyframes and Animations</title>
</head>
<style>
    .container {
        background-color: greenyellow;
    }
}
```

```
.box {  
    background-color: green;  
    width: 250px;  
    height: 250px;  
    position: relative;  
    /* animation-name: harry1; */  
    animation-name: harry2;  
    animation-duration: 8s;  
    animation-iteration-count: 1;  
    /* animation-fill-mode: alternate; */  
    /* animation-timing-function: ease-in-out; */  
    /* animation-delay: 3s; */  
    /* animation-direction: reverse; */  
  
    /* These properties can be set using this shorthand */  
    /* animation: animation-name animation-duration animation-timing-function  
    /* animation: harry 5s ease-in 1s 12 backwards; */  
  
}  
  
@keyframes harry2 {  
    0%{  
        top:0px;  
        left:0px;  
    }  
    25%{  
        top: 250px;  
        left: 0px;  
    }  
    75%{  
        top: 250px;  
        left: 250px;  
    }  
    100%{  
        top: 0px;  
        left: 250px;  
    }  
}  
  
@keyframes harry1 {  
    from {
```

```
        width: 200px;
    }

    to {
        width: 1400px;
    }
}
</style>

<body>
    <div class="container">
        <div class="box">
            This is a box
        </div>
    </div>
</body>

</html>
```

[Previous](#)[Next](#)**CodeWithHarry**

Copyright © 2022 CodeWithHarry.com

