

Day 27

"Web Development + Security"

CSS Transforms:

What is transform in CSS?

The transform property lets you visually change the shape, position, and size of an element without affecting surrounding elements.

Example:

```
div {  
    transform: rotate(45deg);  
}
```

Common Transform Functions:

Function	Description	Example
translate(x, y)	Moves (shifts) the element along X & Y axes	transform: translate(50px, 20px);
rotate(angle)	Rotates the element clockwise or counterclockwise	transform: rotate(45deg);
scale(x, y)	Scales (zooms) the element	transform: scale(1.5, 1.5);
skew(x, y)	Skews (tilts) the element	transform: skew(20deg, 10deg);
matrix(a, b, c, d, e, f)	Advanced 2D combination of all transforms	Rarely used manually

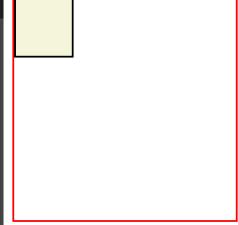
A basic example without any transforms:

The screenshot shows a browser window displaying a simple HTML page. The page contains a single div with the class "box". The CSS for this div is defined in the head section:

```
index.html > html > head > style > .box
1 <html lang="en">
2   <head>
3     <style>
4       .box{
5         height: 50px;
6         width: 50px;
7         background-color: beige;
8         border: 2px solid black;
9       }
10      </style>
11    </head>
12    <body>
13      <div class="container">
14        <div class="box">
15        </div>
16      </div>
17    </body>
18  </html>
```

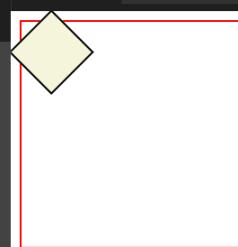
The browser's developer tools are visible on the left, showing the DOM structure. The page itself displays a single, centered beige square with a black border.

One more basic example: without transform



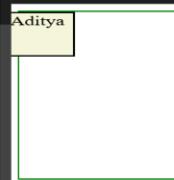
```
index.html
1 <html lang="en">
2   <head>
3     <style>
4       .box{
5         height: 50px;
6         width: 50px;
7         background-color: beige;
8         border: 2px solid black;
9       }
10      .container{
11        border: 2px solid red;
12        height: 200px;
13        width: 200px;
14      }
15    </style>
16  </head>
17  <body>
18    <div class="container">
19      <div class="box">
20      </div>
21    </div>
22  </body>
23 </html>
```

Now, applying the transform:rotate(45deg):



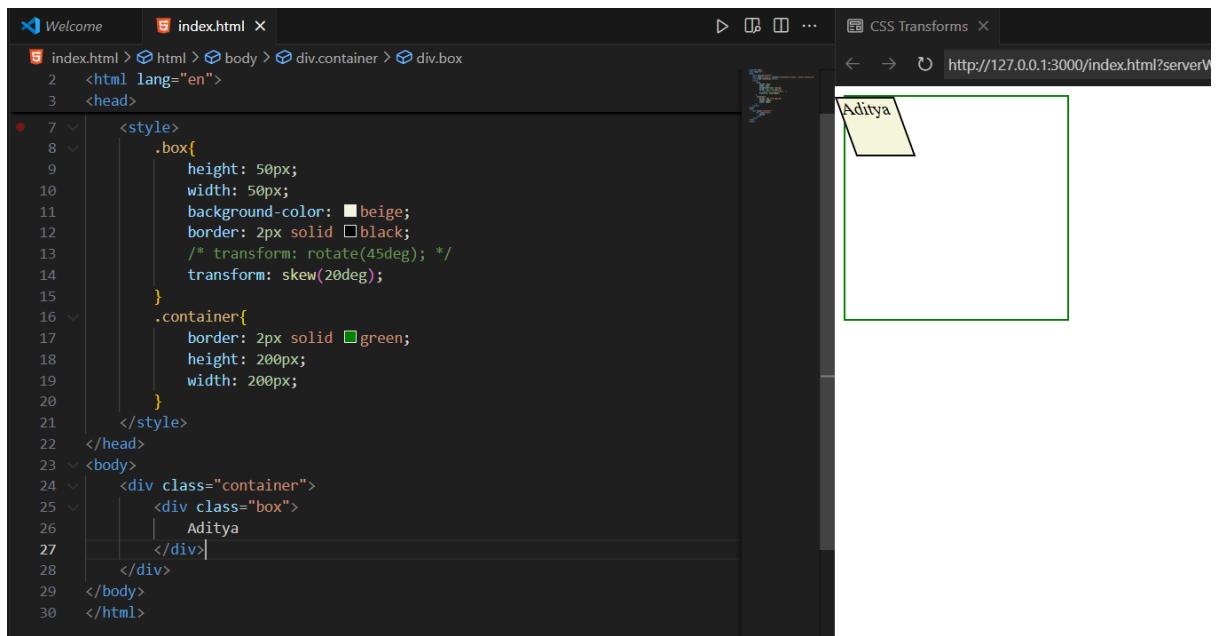
```
index.html
1 <html lang="en">
2   <head>
3     <style>
4       .box{
5         height: 50px;
6         width: 50px;
7         background-color: beige;
8         border: 2px solid black;
9         transform: rotate(45deg);
10        }
11       .container{
12         border: 2px solid red;
13         height: 200px;
14         width: 200px;
15       }
16     </style>
17   </head>
18   <body>
19     <div class="container">
20       <div class="box">
21       </div>
22     </div>
23   </body>
24 </html>
```

Example: we used scale() in order to scale the element



```
index.html
1 <html lang="en">
2   <head>
3     <style>
4       .box{
5         height: 50px;
6         width: 50px;
7         background-color: beige;
8         border: 2px solid black;
9         /* transform: rotate(45deg); */
10        transform: scale(1.5);
11        }
12       .container{
13         border: 2px solid green;
14         height: 200px;
15         width: 200px;
16       }
17     </style>
18   </head>
19   <body>
20     <div class="container">
21       <div class="box">
22         Aditya
23       </div>
24     </div>
25   </body>
26 </html>
```

Example: skew()

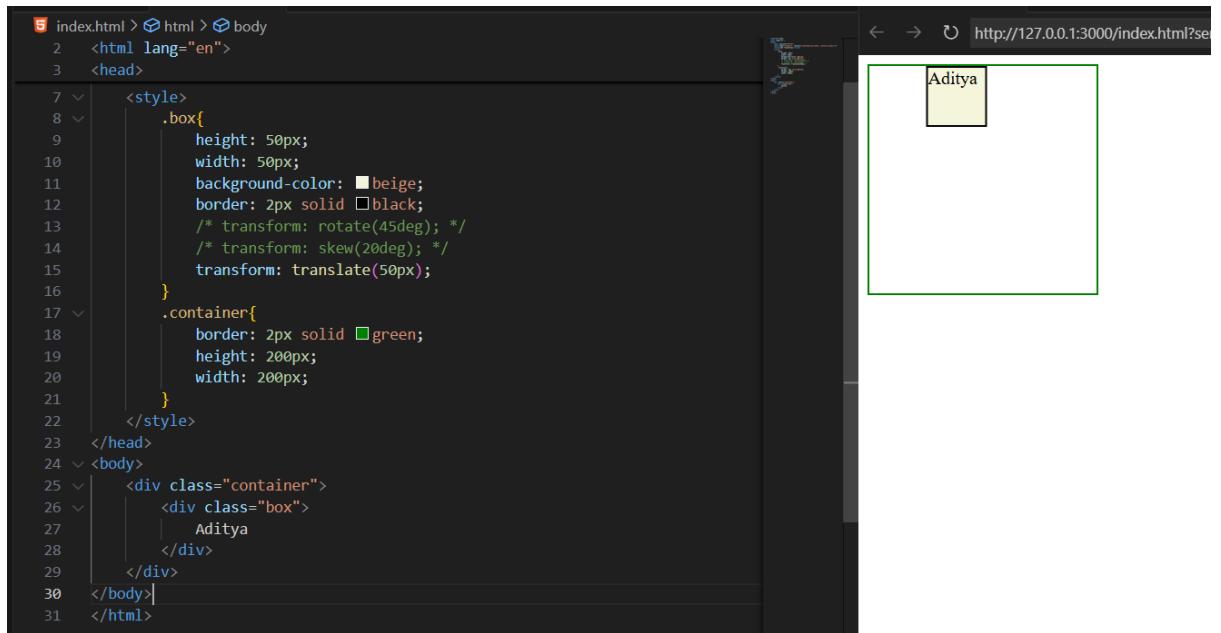


The screenshot shows a browser developer tools window with the CSS Transform panel open. The code editor on the left contains an HTML file named index.html with the following content:

```
<html lang="en">
<head>
<style>
    .box{
        height: 50px;
        width: 50px;
        background-color: beige;
        border: 2px solid black;
        /* transform: rotate(45deg); */
        transform: skew(20deg);
    }
    .container{
        border: 2px solid green;
        height: 200px;
        width: 200px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="box">
            Aditya
        </div>
    </div>
</body>
</html>
```

The rendered page on the right shows a yellow square with the text "Aditya" inside, positioned at the top-left corner of a larger green-bordered container.

Example: translate()



The screenshot shows a browser developer tools window with the CSS Transform panel open. The code editor on the left contains an HTML file named index.html with the following content:

```
<html lang="en">
<head>
<style>
    .box{
        height: 50px;
        width: 50px;
        background-color: beige;
        border: 2px solid black;
        /* transform: rotate(45deg); */
        /* transform: skew(20deg); */
        transform: translate(50px);
    }
    .container{
        border: 2px solid green;
        height: 200px;
        width: 200px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="box">
            Aditya
        </div>
    </div>
</body>
</html>
```

The rendered page on the right shows a yellow square with the text "Aditya" inside, shifted to the right by 50px relative to its original position.

CSS Transitions:

What is a CSS Transition?

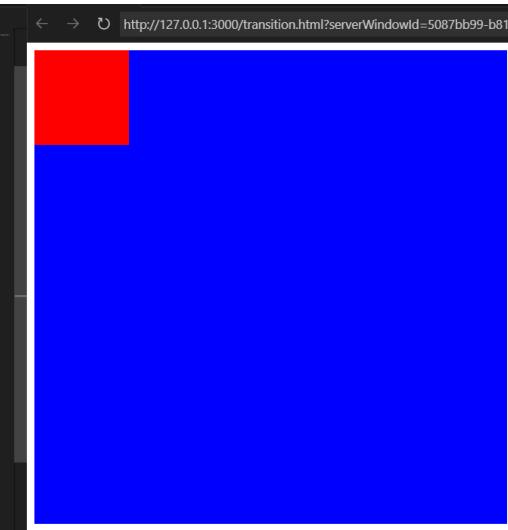
A transition lets you gradually change a CSS property's value over time — instead of changing it instantly.

Basic syntax:

transition: property duration timing-function delay;

Part	Meaning	Example
property	CSS property to animate	width, color, transform, etc.
duration	How long the change takes	1s, 0.5s
timing-function	Speed curve of animation	ease, linear, ease-in, ease-out, ease-in-out
delay	When to start (optional)	0s, 1s

A very basic example: where two <div> without transition are there:

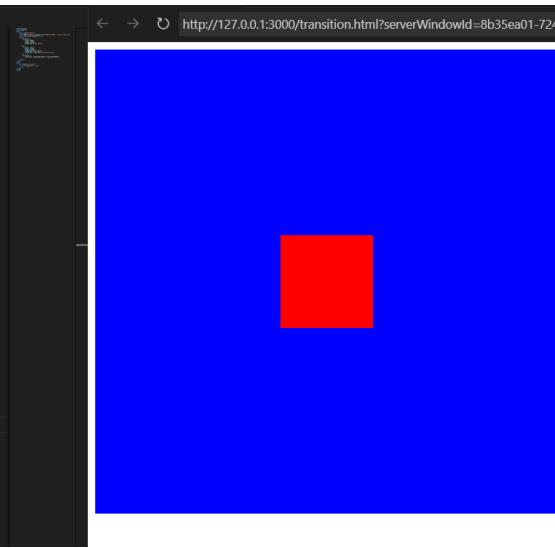


The screenshot shows a browser window displaying a simple HTML page. On the left, the code editor shows the following HTML and CSS:

```
1 transition.html > ⌂ html
2   <html lang="en">
3     <head>
4       <style>
5         .container{
6           width:500px;
7           height:500px;
8           background-color: blue;
9         }
10        .box{
11          width:100px;
12          height:100px;
13          background-color: red;
14        }
15      </style>
16    </head>
17    <body>
18      <div class="container">
19        <div class="box">
20        </div>
21      </div>
22    </body>
23  </html>
```

The browser window on the right shows a blue square container with a red square box inside it. The red box is positioned in the top-left corner of the blue container.

Now, applying the transition property: we will see that red box slowly, moves to 200px in both X and Y axis.



A screenshot of a browser window displaying a simple HTML page. The page contains a single red square box centered on a blue background. The browser's developer tools are open, showing the source code on the left and the rendered page on the right. The source code includes a CSS rule for a ".box" element with a "transition: transform 3s ease-in-out 1s;" declaration, which is responsible for the smooth movement of the box.

```
transition.html > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Transition Example</title>
7      <style>
8          .container {
9              width: 500px;
10             height: 500px;
11             background-color: blue;
12         }
13         .box {
14             width: 100px;
15             height: 100px;
16             background-color: red;
17             transition: transform 3s ease-in-out 1s;
18         }
19         .box:hover {
20             transform: translateX(200px) translateY(200px);
21         }
22     </style>
23 </head>
24 <body>
25     <div class="container">
26         <div class="box"></div>
27     </div>
28 </body>
29 </html>
```

CSS Animations:

What is a CSS Animation?

CSS animations let you move, change, or transform elements automatically over time, without user interaction. Unlike transitions, animations don't need hover or click — they can start on page load, repeat, and have multiple stages.

Basic syntax:

```
selector {
    animation-name: myAnimation;
    animation-duration: 2s;
    animation-timing-function: ease-in-out;
    animation-delay: 0s;
    animation-iteration-count: infinite;
    animation-direction: alternate;
}
```

Then define the animation using @keyframes 

```
@keyframes myAnimation {
    from { background-color: red; }
    to { background-color: blue; }
}
```

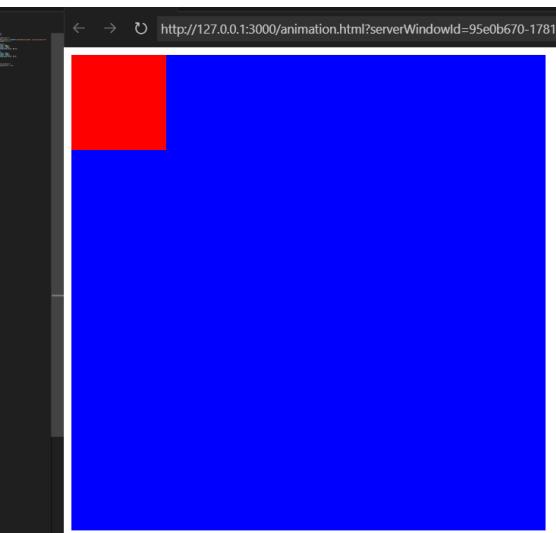
Or with percentage stages:

```
@keyframes myAnimation {  
    0% { transform: translateX(0); }  
    50% { transform: translateX(100px); }  
    100% { transform: translateX(0); }  
}
```

Main animation properties:

Property	Description	Example
animation-name	Name of @keyframes	slide
animation-duration	How long it lasts	2s
animation-timing-function	Speed curve	ease, linear, etc.
animation-delay	Wait before start	1s
animation-iteration-count	How many times	infinite, 3
animation-direction	normal, reverse, alternate	alternate = forward then backward
animation-fill-mode	Keeps end state	forwards
animation-play-state	running or paused	pause/resume control

A basic example without any animation:



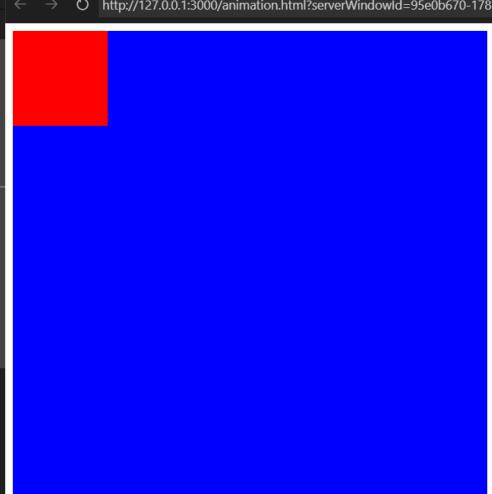
The screenshot shows a browser window displaying a simple HTML page. The page contains a single

element with the class "container". Inside this container is another

element with the class "box". The container has a width and height of 500px and a blue background color. The box has a width and height of 100px and a red background color. The browser interface includes a code editor on the left showing the HTML and CSS code, and a preview window on the right showing the visual result.

```
1  animation.html > ⌂ html  
2  <!DOCTYPE html>  
3  <html lang="en">  
4  <head>  
5      <meta charset="UTF-8">  
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7      <title>Document</title>  
8      <style>  
9          .container {  
10              width: 500px;  
11              height: 500px;  
12              background-color: blue;  
13          }  
14          .box {  
15              width: 100px;  
16              height: 100px;  
17              background-color: red;  
18      }  
19  </style>  
20  </head>  
21  <body>  
22      <div class="container">  
23          <div class="box"></div>  
24      </div>  
25  </body>  
26 </html>
```

Now, adding the animation:



```
1 animation.html > html > head > style > .box
2   html lang="en">
3   head>
4     <style>
5       .container {
6         width: 500px;
7         height: 500px;
8         background-color: blue;
9       }
10      .box {
11        width: 100px;
12        height: 100px;
13        background-color: red;
14        animation-name: Aditya;
15        animation-duration: 3s;
16        animation-timing-function: ease-in-out;
17        animation-delay: 1s;
18        animation-iteration-count: 3;
19      }
20    }
21  }
22  @keyframes Aditya {
23    from {
24      transform: translateX(0) translateY(0);
25    }
26    to {
27      transform: translateX(200px) translateY(200px);
28    }
29  }
30  </style>
31  </head>
32  <body>
33    <div class="container">
34      <div class="box"></div>
35    </div>
36  </body>
37  </html>
```

--The End--