

# Training Day 17 Report

16 July 2025

## CSS Transitions, Animations & Transforms

### 1. CSS Transitions

#### Definition:

A CSS transition allows you to smoothly change property values over time, rather than instantly. It's commonly used to animate changes such as color, size, position, or opacity when user interaction happens (like hover, focus, etc.).

#### Shorthand Syntax:

transition: property duration timing-function delay;

#### Example:

```
.button {  
  transition: background-color 0.3s ease-in-out 0s;  
}
```

### 2. CSS Keyframe Animations

#### Definition:

Keyframes are used to create custom animations by defining intermediate steps (or frames) between the start and end of an animation. CSS keyframes tell the browser what styles to apply at certain times during the animation sequence.

#### -Syntax:

```
@keyframes animation-name {  
  0% { /* starting state */ }  
  50% { /* middle state */ }  
  100% { /* ending state */ }  
}
```

-Applying Animation to an Element:

```
.element {  
  animation-name: animation_name;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
}
```

Property	Definition
-----	-----
`animation-name`	Name of the `@keyframes` animation
`animation-duration`	Time for one cycle of animation
`animation-iteration-count`	How many times it repeats (`1`, `infinite`, `3`, etc.)

-Example: Bouncing Ball

```
@keyframes bounce {  
  0% { transform: translateY(0); }  
  50% { transform: translateY(-100px); }  
  100% { transform: translateY(0); }  
}
```

```
.ball {  
  animation-name: bounce;  
  animation-duration: 0.5s;  
  animation-iteration-count: infinite;  
}
```

### 3. CSS Transform

#### -Definition:

The transform property in CSS is used to visually manipulate elements by applying rotation, scaling, translation, or skewing — all without affecting the element's actual layout in the page.

#### -Types of Transforms:

##### ->Rotate

Rotates the element clockwise or anticlockwise.

```
transform: rotate(45deg); /* Clockwise */
```

```
transform: rotate(-30deg); /* Anticlockwise */
```

##### -Scale

Scales the element's size (width and height).

```
transform: scale(1.2); /* 120% of original size */
```

```
transform: scale(0.5, 1.5); /* Width 50%, Height 150% */
```

##### -Translate

Moves the element along the X and/or Y axis from its original position.

```
transform: translate(50px, 0); /* Move right */
```

```
transform: translate(0, -30px); /* Move up */
```

#### -Combine Transforms:

```
transform: rotate(45deg) scale(1.2) translate(20px, 10px);
```

## 4. Spinning Loader Animation (Example)

-Example Code:

HTML

```
<div class="loader"></div>
```

CSS

```
.loader {  
  width: 40px;  
  height: 40px;  
  border: 4px solid #ccc;  
  border-top: 4px solid #3498db;  
  border-radius: 50%;  
  animation: spin 1s linear infinite;  
}
```

```
@keyframes spin {  
  0% { transform: rotate(0deg); }  
  100% { transform: rotate(360deg); }  
}
```

-Explanation:

->Circular border mimics a spinner.

->Only the top border is colored, so it creates a visual rotation.

->The spin keyframe rotates it infinitely for a smooth loader.