

## CSS-4

### Agenda

- Important components of Animation
  - Transform
  - Transition
  - Keyframe

### Coordinate System Convention

- In web development, the screen is considered to be in the x-y plane.
- You, the viewer, are positioned in the z-plane, looking towards the screen.
- \*Yha hm z-axis se cheezo ko dekh rahe hote hai so accordingly x and y ki cheeze



### Transform Property

The `transform` property allows you to modify the coordinate space of the CSS visual formatting model. Here are key transformations:

#### 1. Rotate

- Rotates an element around a specified axis.

```
.one:hover {  
  transform: rotateX(50deg);  
}  
.two:hover {  
  transform: rotateY(50deg);  
}  
.three:hover {
```

```
transform: rotate(50deg);  
}
```

- **rotateX(50deg)**: Rotates the element around the x-axis.
- **rotateY(50deg)**: Rotates the element around the y-axis.
- **rotate(50deg)**: Rotates the element in the 2D plane (z-axis rotation).

## 2. Scale

- Resizes the element in either the x, y, or both directions.

```
.one:hover {  
  transform: scaleX(1.5);  
}  
.two:hover {  
  transform: scaleY(1.5);  
}  
.three:hover {  
  transform: rotate(50deg) scale(0.5);  
}
```

- **scaleX(1.5)**: Scales the width of the element by 1.5 times.
- **scaleY(1.5)**: Scales the height of the element by 1.5 times.
- **rotate(50deg) scale(0.5)**: Rotates the element and then scales it to half its size.

## 3. Translate

- Moves the element from its original position.

```
.one:hover {  
  transform: translateY(50%);  
}  
.two:hover {  
  transform: translateX(50%);  
}  
.three:hover {  
  transform: translate(50%, 50%);  
}
```

- **translateY(50%)**: Moves the element down by 50% of its height.
- **translateX(50%)**: Moves the element right by 50% of its width.
- **translate(50%, 50%)**: Moves the element diagonally down and right by 50%.

## Transition Property

The `transition` property is used to animate changes to CSS properties, allowing for smooth transitions between different states.

```
.box {  
  height: 100px;  
  width: 100px;  
  transition-duration: 1s;  
}  
  
.one {  
  background-color: aqua;  
}  
  
.two {  
  background-color: lightcoral;  
}  
  
.three {  
  background-color: lightgreen;  
}
```

### 1. Transition Duration

- Specifies how long the transition should take.

```
.one {  
  transition-duration: 2s;  
}  
.two {  
  transition-duration: 1s;  
}
```

### 2. Transition Timing Function

- Controls the speed curve of the transition.

```
.one {  
  transition-timing-function: ease;  
}
```

### 3. Transition Delay

- Specifies a delay before the transition starts.

```
.two {  
  transition-delay: 1s;  
}
```

```
}
```

## Centering a Div

### 1. Using Flexbox:

- Add `display: flex;` to the container.
- Use `justify-content: center;` to center items horizontally.
- Use `align-items: center;` to center items vertically.

```
.container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

### 2. Using Position and Translate:

- Set the container to `position: relative;`.
- Set the box to `position: absolute;`.
- Use `top: 50%;` and `left: 50%;` to position the box at the center.
- Use `transform: translate(-50%, -50%);` to adjust the box position.

```
.container {  
  position: relative;  
}  
.box {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```

## Animation Properties

1. **animation-name:** Specifies the keyframes to be used for the animation.

```
.box {  
  animation-name: journey-to-circle;  
}
```

2. **@keyframes:** Defines the stages of the animation with different styles at different percentages.

```
@keyframes journey-to-circle {  
  50% {  
    background-color: lightcoral;  
    border-radius: 50%;  
  }  
  75% {  
    border-radius: 10%;  
  }  
  100% {  
    background-color: honeydew;  
    border-radius: 0%;  
  }  
}
```

3. **animation-duration:** Defines how long the animation takes to complete one cycle.

```
.box {  
  animation-duration: 5s;  
}
```

4. **animation-iteration-count:** Sets the number of times the animation will repeat., It can be a number and can also be infinite.

```
.box {  
  animation-iteration-count: 2;  
}
```