

---

<Semester IV>

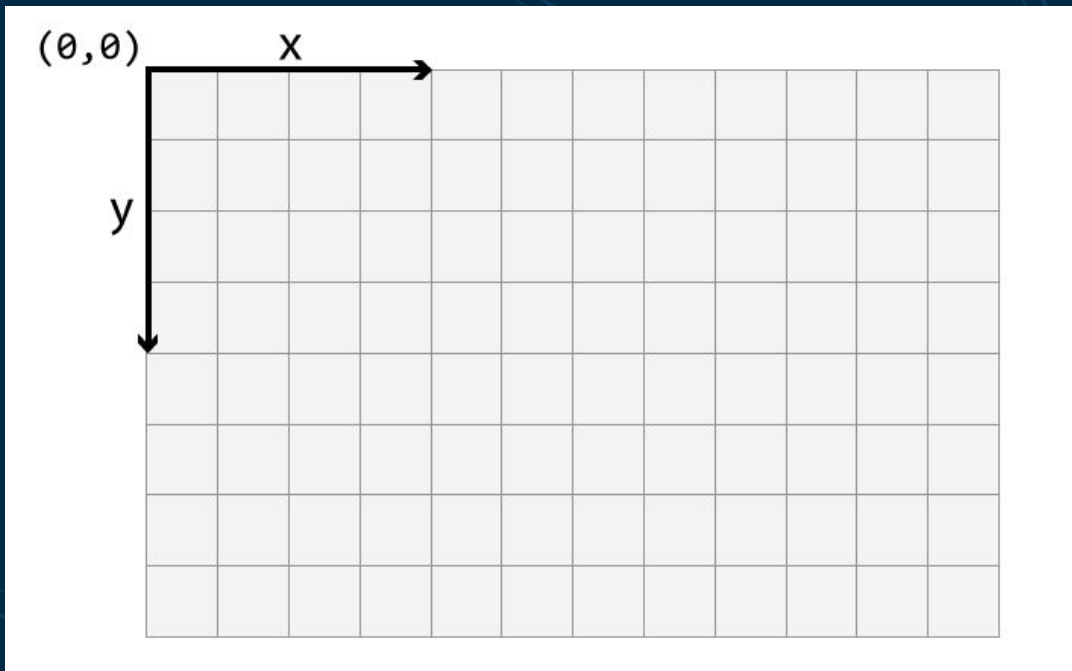
# <Web Development for Designers>

Week 5

---

# CSS Coordinate System

In CSS, the coordinate system is a reference point for the *offset* values of the position, transform, transition and many other properties that relate to spatial positions of an element. The origin of the coordinate system in CSS is located at the top left corner of the element.



# CSS Transform

- ❑ CSS transforms change the shape and position of the affected content without disrupting the normal document flow.
- ❑ The transform property in CSS is used to change the coordinate space of the visual formatting model. This is used to add effects like skew, rotate, translate, etc on elements.
- ❑ These transformations include rotation, skewing, scaling, and translation both in the plane and in the 3D space.
- ❑ Two major properties are used to define CSS transforms: transform (or the individual translate, rotate, and scale properties) and transform-origin.
- ❑ Example in action: [Geeks for Geeks CSS Transform Properties](#)

```
.cta-button:hover {  
  transform: rotate(-10deg);  
}
```

Rotate

```
.cta-button:hover {  
  transform: skewx(-10deg);  
}
```

Skew

```
.cta-button:hover {  
  transform: scale(1.1);  
}
```

Scale

```
.cta-button:hover {  
  transform: translatex(10px);  
}
```

Translate



# TRANSFORM



**IN 9 MINUTES!**

# CSS Transitions

CSS transitions are used to create smooth animations between two states of an element, enhancing interactivity and user experience.

- ❑ Transitions can animate properties like color, size, and position.
- ❑ Use selectors and pseudo-classes (e.g., :hover) to trigger transitions.
- ❑ Key transition properties include transition-property, transition-duration, transition-timing-function, and transition-delay.

To create effective transitions, you should use at least two of the four key properties: **transition-property** and **transition-duration**.

# Transitions Syntax

```
<head>
<style>
  .box {
    width: 100px;
    height: 100px;
    background-color: blue;
    transition-property: width;
    transition-duration: 1s;
    transition-delay: 0.5s;
    transition-timing-function: ease-in-out;
  }
  .box:hover {
    width: 200px;
  }
</style>
</head>
```

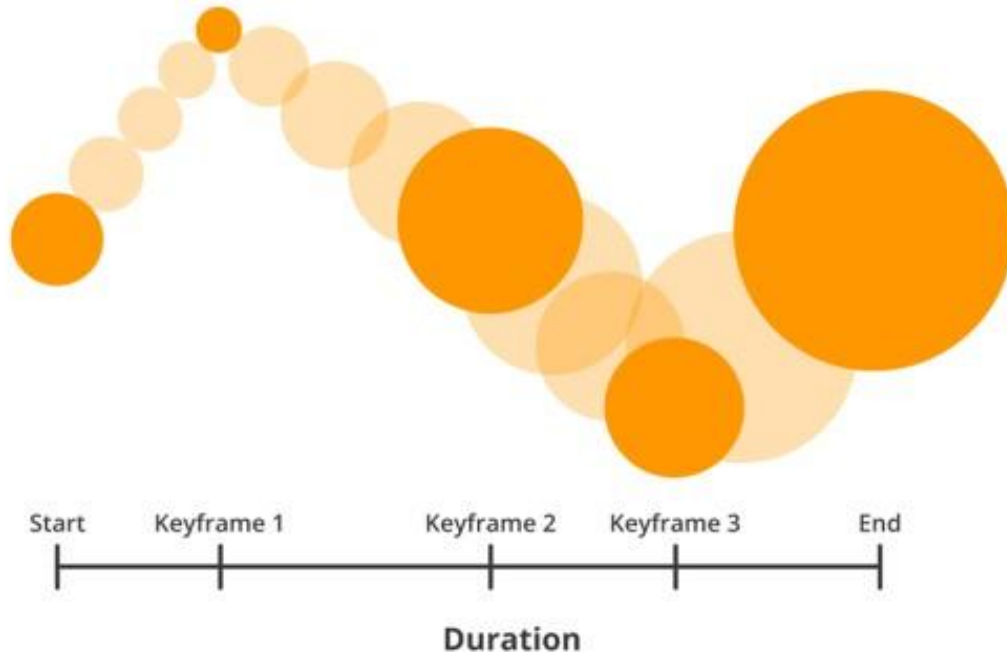
# CSS Transitions





# What are Keyframes?

A keyframe in animation is a specific reference point in an animation where a change or adjustment is made to an object's state or property.



# CSS Animations

CSS animations make it possible to animate transitions from one CSS style configuration to another. Animations consist of two components: a style describing the CSS animation and a set of keyframes that indicate the start and end states of the animation's style, as well as possible intermediate waypoints.

The animation property in CSS can be used to animate many other CSS properties such as color, background-color, height, or width. Each animation needs to be defined with the @keyframes at-rule which is then called with the animation property. The @keyframes rule defines how an element's styles change over time during an animation.

# CSS Keyframes Syntax

```
@keyframes animation-name {  
  from {  
    /* Initial Styles */  
  }  
  to {  
    /* Final Styles */  
  }  
}
```

- ❑ The from defines the starting styles, and to defines the ending styles.
- ❑ You can also use percentage values to specify intermediate steps.

# CSS Animation Properties

PROPERTY	DESCRIPTION
animation-name	Specifies the name of the keyframe you want to bind to the selector
animation-duration	Specifies how many seconds or milliseconds an animation takes to complete
animation-timing-function	Specifies the speed curve of the animation
animation-delay	Specifies a delay before the animation will start
animation-iteration-count	Specifies how many times an animation should be played
animation-direction	Specifies whether or not the animation should play in reverse on alternate cycles
animation-fill-mode	Specifies what values are applied by the animation outside the time it is executing
animation-play-state	Specifies whether the animation is running or paused

# CSS Animation Syntax Example

```
/* The animation code */
@keyframes example {
  0%   {background-color: red;}
  50%  {background-color: violet;}
  100% {background-color: blue;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 3s;
}
```

# CSS Animation

## HOW TO USE



```
animation: 10s spin infinite  
animation-delay  
animation-timing-function  
animation-play-state  
animation-fill-mode
```



**Homework: Try different transforms, transitions, and animations in your code. Try the exercise given in the last video (animation properties video)**

**W3Schools HTML Tutorial**

**HTML: HyperText Markup Language | MDN**

---

# Reference



1. [CSS Transforms - MDN](#)
  2. [Geeks for Geeks CSS Transform Properties](#)
  3. [Geeks for Geeks CSS Transition Properties](#)
  4. [Transition Property - MDN](#)
  5. [Keyframes in Animation - SVGator](#)
  6. [CSS Animations - MDN](#)
  7. [CSS Animations - W3Schools](#)
-



# Fonts & colors used

This presentation has been made using the following fonts:

## **Blinker**

(<https://fonts.google.com/specimen/Blinker>)

## **Inconsolata**

(<https://fonts.google.com/specimen/Inconsolata>)

#ffffff

#022a46

#72ffdd

#72fff

#0d3a58