Css Animation Learning : Day-1

Bhai, tu full animation master banega!

Main tereko step-by-step har topic samjhaunga, basics se lekar advance tak, example ke saath. Chal shuru karein!

---

1. `opacity` - Element ka Transparency (Fade Effect)

Basic Explanation

- `opacity` ka use hota hai kisi element ko **transparent** ya **visible** banane ke liye.
- `opacity: 1;` → Fully visible
- `opacity: 0;` → Fully invisible
- `opacity: 0.5;` → 50% transparent

Example 1: Hover Par Fade Effect

```
css
CopyEdit
.fade-box {
    width: 200px;
    height: 200px;
    background-color: red;
    opacity: 1;
    transition: opacity 1s ease-in-out;
}

.fade-box:hover {
    opacity: 0.2; /* Hover karne par element transparent ho jayega */
}
```

Explanation: Jaise hi tu `div` par mouse le jaayega, voh transparent ho jayega **(1s ke andar smoothly)**.

---

2. `transform` - Scale, Rotate, Translate, Skew

(A) `scale()` - Element Ka Size Badhana/Chhota Karna

```
css
CopyEdit
.scale-box {
    width: 100px;
    height: 100px;
    background-color: blue;
    transition: transform 0.5s ease-in-out;
}
```

```
.scale-box:hover {
    transform: scale(1.5); /* 1.5x size bada ho jayega */
}
```

scale() Property – Kitna Number De Sakte Hai?

`transform: scale(X);` ek element ka **size badhata ya ghata** hai.

X ka value:

- `scale(1.0);` → **Original size** (koi change nahi)
- `scale(1.05);` → **5% bada** (Thoda zoom-in effect)
- `scale(1.2);` → **20% bada** (Aur bada ho jayega)
- `scale(2);` → **2x size bada** (Dugna ho jayega)
- `scale(0.8);` → **20% chhota** (Zoom-out hoga)
- `scale(0.5);` → **50% chhota** (Aadha ho jayega)

X aur Y alag bhi de sakte hai:

- `scale(1.2, 1);` → Sirf width **20% bada**, height same.
- `scale(1, 1.5);` → Sirf height **50% bada**, width same.

Example:

```
css
.skills-card-container:hover {
    transform: scale(1.2, 1.2); /* 20% bada */
}
```

(B) `rotate()` - Element Ko Ghumana

```
css
.rotate-box {
    width: 100px;
    height: 100px;
    background-color: green;
    transition: transform 0.5s ease-in-out;
}

.rotate-box:hover {
    transform: rotate(45deg); /* 45-degree ghoomega */
}
```

(C) `translate()` - Element Ko Move Karna

```
css
.move-box {
    width: 100px;
    height: 100px;
    background-color: orange;
    transition: transform 0.5s ease-in-out;
}
```

```
.move-box:hover {
    transform: translateX(50px); /* X-axis par 50px move karega */
}
```

(D) `skew()` - Element Ko Tilt Karna

```
css
.skew-box {
    width: 100px;
    height: 100px;
    background-color: purple;
    transition: transform 0.5s ease-in-out;
}

.skew-box:hover {
    transform: skew(20deg); /* 20-degree tilt hoga */
}
```

---

3. `background-color` - Smooth Color Transition

```
css
.color-box {
    width: 150px;
    height: 150px;
    background-color: red;
    transition: background-color 1s ease-in-out;
}

.color-box:hover {
    background-color: blue; /* Red se Blue smoothly change hoga */
}
```

---

4. `color` - Text Ka Smooth Color Transition

```
css
.text-box {
    font-size: 24px;
    color: black;
    transition: color 1s ease-in-out;
}

.text-box:hover {
    color: red; /* Black se Red smoothly change hoga */
}
```

Next Step: Advanced Animations (Combining Multiple Effects)

Bhai, yeh toh sirf **basic effects** the. Ab **advance level** pe chalenge, jisme multiple effects ek sath apply karenge, jaise: Opacity + Transform

Background-Color + Scale
Translate + Rotate + Skew

Bata, kya tu aur bhi **pro-level CSS animations** seekhna chahta hai?

4o

You said:

ha ye sab Opacity + Transform Background-Color + Scale Translate + Rotate + Skew

ChatGPT said:

Bhai, ab tu **CSS Animation Pro Max** banega!
Chal, ab **advanced animations** ka game shuru karte hain!

---

1. `Opacity + Transform` (Fade + Scale)

Koi element **fade-in aur zoom-in** ho jaye smoothly? Yeh raha code:

```
css
.opacity-scale-box {
    width: 150px;
    height: 150px;
    background-color: royalblue;
    opacity: 0; /* Start me invisible hoga */
    transform: scale(0.5); /* Chhota dikhai dega */
    transition: opacity 1s ease-in-out, transform 1s ease-in-out;
}

.opacity-scale-box:hover {
    opacity: 1; /* Pura dikhai dega */
    transform: scale(1); /* Normal size pe aayega */
}
```

Hover karte hi element smoothly fade hoke bada ho jayega!

---

2. `Background-Color + Scale` (Color Change + Zoom)

```
css
.bg-scale-box {
    width: 150px;
    height: 150px;
    background-color: green;
    transition: background-color 0.8s ease-in-out, transform 0.8s ease-in-out;
}

.bg-scale-box:hover {
    background-color: yellow; /* Color change hoga */
    transform: scale(1.2); /* Element bada ho jayega */
}
```

Hover karne pe color smoothly change hoga aur element bada ho jayega!

---

3. Translate + Rotate + Skew (Move + Rotate + Tilt)

```
css
.combo-box {
    width: 150px;
    height: 150px;
    background-color: crimson;
    transition: transform 1s ease-in-out;
}

.combo-box:hover {
    transform: translateX(50px) rotate(45deg) skew(10deg);
}
```

Hover karne pe element right move karega, rotate hoga aur tilt bhi hoga!

Project Code: What I do run it to see

- Developed Using Bootstrap and Css Animation with above concepts

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous" />
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>
  </head>
  <body>
      <style>
          html{
              font-size: 16px;
          }
          .my-services-bg-container{
              background-color: #52606d;
              min-height: 100vh;
              font-family: "Roboto";
              padding-top: 32px;
```

```
}
.services-title{
    color: #cced00;
    font-size: 1.6rem;
    font-weight: bold;
    transition: 0.2s ease-in;
}
.services-title:hover{
    transform: scale(1.03);
    text-shadow: 0 0 20px white;
}
.main-title{
    color: white;
    font-weight: bolder;
    font-size: 2.4rem;
    transition: transform 0.3s ease-in-out;
}
.main-title:hover{
    transform: scale(0.84);
    text-shadow: 0px 0px 20px white;

}
.project-container{
    padding: 16px;
    background-color: #323f4b;
    border-radius: 12px;
    transition: box-shadow 0.3s ease-in-out, background-color 0.3s ease-in-out, transform 0.3s ease-in-out;
}
.project-container:hover{
    background-color: #52606d;
    box-shadow: 0 0 20px white;
    transform: scale(1.03);
}
.project-logo{
    max-width: 10%;
    transition: transform 0.3s ease-in-out, box-shadow 0.3s ease-in;
}
.project-logo:hover{
    transform: translate(20px);
    box-shadow: 4px 0 20px #9aa5b1;
}
.project-card-title{
    font-size: 2rem;
    color: white;
    font-weight: bolder;
    padding-top: 8px;
    display: inline-block;
```

```
          transition: 0.3s ease-in-out;
        }
        .project-card-title:hover{
          transform: rotate(360deg) scale(0.5);
          /* transform: scale(0.5); */
        }
        .project-card-description{
          font-size: 1.4rem;
          color: #9aa5b1;
          transition: text-shadow 0.2s ease-in-out;
        }
        .project-card-description:hover{
          text-shadow: 0px 0px 80px #cced00;
        }
    </style>
  <div class="my-services-bg-container">
      <div class="container">
          <div class="row">
              <div class="col-12 mb-3">
                  <h3 class="services-title">Services</h3>
                  <h1 class="main-title">What I do</h1>
              </div>
              <div class="col-12 col-md-6 col-lg-4 mb-4">
                  <div class="project-container">
                      <img src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-responsive-website/portfolio-web-design-img.png"
                      class="project-logo" alt="project-img" />
                      <div>
                      <h1 class="project-card-title">Static Websites</h1>
                      </div>
                      <p class="project-card-description">A Static Website container Web pages with fixed content. Each page is
                      developed using HTML and CSS and displays the same information to every visitor.</p>
                  </div>
              </div>
              <div class="col-12 col-md-6 col-lg-4 mb-4">
                  <div class="project-container">
                      <img src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-responsive-website/portfolio-web-design-img.png"
                      class="project-logo" alt="project-img" />
                      <div>
                      <h1 class="project-card-title">Static Websites</h1>
                      </div>
                      <p class="project-card-description">A Static Website container Web pages with fixed content. Each page is
                      developed using HTML and CSS and displays the same information to every visitor.</p>
                  </div>
              </div>
              <div class="col-12 col-md-6 col-lg-4 mb-4">
                  <div class="project-container">
                      <img src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-responsive-website/portfolio-web-design-img.png"
```

```
            class="project-logo" alt="project-img" />
            <div>
            <h1 class="project-card-title">Static Websites</h1>
            </div>
            <p class="project-card-description">A Static Website container Web pages with fixed content. Each page is
            developed using HTML and CSS and displays the same information to every visitor.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
 </body>
</html>
```

Problems Occur During Project

- transform: rotate() kabhi kabhi work nhi karti acche se inline ya block level elements me like p, h1, span, etc then inhe div ke ander rakh ke inke css selector yani transform: rotate() ke uper display: inline-block; ka use kiya jata hai ye kya karta hai ye elements ko block level and inline level done tarah behave karata hai.
- Double transform property use karwane ke liye alag alag nhi likh sakte ye override ho jati hai ek hi me likha jaata hai like ---- transform: rotate(20deg) scale(0.4) skew(10deg).

Css Topic : transition

transition syntax -- **transition: <property> <duration> <timing-function>**

Agar `transition` me koi specific property mention nahi ki gayi, jaise:

```
css
.skills-card-container {
    transition: 0.3s ease-in-out;
}
```

To ye sabhi animatable properties pe apply hoga jo hover ya kisi aur state change me modify ho rahi hain.

Example:
Agar `.skills-card-container:hover` me **`transform`** aur **`box-shadow` dono change ho rahe hain, to transition dono pe lagega**.

Example Code:

```
css
CopyEdit
.skills-card-container {
    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.2);
    transition: 0.3s ease-in-out; /* Property mention nahi hai */
}

.skills-card-container:hover {
    transform: scale(1.1); /* Ye animate hoga */
    box-shadow: 0px 8px 16px rgba(0, 0, 0, 0.3); /* Ye bhi animate hoga */
}
```

Explanation:

- **`transform: scale(1.1)`** aur **`box-shadow`** dono change ho rahe hain.
- Kyunki `transition` me **koi specific property nahi likhi**, to **automatically har animatable property** pe apply ho jayega jo hover pe change ho rahi ho.
- Agar **sirf `transform` pe apply karna hota**, to likhna padta:

  ```
  css
  transition: transform 0.3s ease-in-out;
  ```

Question :
kaise pata kare ki konsi konsi animatable property hai ?

ChatGPT said:

Bhai, **animatable properties** wo hoti hain jo **value change hone par smoothly transition ya animation support karti hain**. Har CSS property **animate nahi hoti**.

List of Animatable Properties

Ye kuch **common animatable properties** hain jo smooth transition support karti hain:

Property	Description
`opacity`	Element ka transparency change hota hai (fade effect).
`transform`	Element ka `scale()`, `rotate()`, `translate()`, `skew()` change hota hai.
`background-color`	Smooth color transition hota hai.
`color`	Text ka color smoothly change hota hai.
`width` / `height`	Element ka size smoothly change hota hai.
`box-shadow`	Shadow ka size, blur, aur color smoothly change hota hai.
`border-radius`	Corners smoothly round ya sharp ho sakte hain.
`top` / `left` / `right` / `bottom`	Position smoothly change hoti hai (but prefer `transform: translate()` for better performance).
`margin` / `padding`	Spacing smooth change hota hai.

Property	Description
`letter-spacing`	Letters ke beech ka gap smoothly change hota hai.
`line-height`	Text ki line ki height smoothly change hoti hai.

These Property are done above in details

`opacity`
`transform`
`background-color`
`color`

Detailed Study about transition css property

`transition` ke Values – Kya Kya Use Kar Sakte Hai?

`transition` property animation ko **smooth** banati hai.
Syntax:

```
css
transition: <property> <duration> <timing-function> <delay>;
```

Breakdown:

- **`property` → Kisme transition lagana hai? (`all`, `transform`, `opacity`, `box-shadow`, etc.)**
- **`duration` → Kitni der tak effect hoga? (`0.3s`, `1s`, `500ms`)**
- **`timing-function` → Animation speed kaise chalega? (`ease`, `linear`, `ease-in-out`, etc.)**
- **`delay` → Effect start hone me kitna time lagega? (`0s`, `0.2s`, etc.)**

Example:

```
css
.skills-card-container {
    transition: transform 0.5s ease-in-out;
}
```

Matlab: transform property ka effect **0.5s** me **smoothly (ease-in-out)** hoga.

`transition` me kya kya values de sakte hai?

1. **`all`** – Sab properties pe transition lagayega.

```
css
transition: all 0.4s ease-in;
```

(Slowly start hoga, phir fast hoga, har property pe lagega)

2. `ease` – Slow start, fast beech me, slow end. (Default)

```
css
transition: all 0.3s ease;
```

(Smooth feel dega, most common hai!)

3. `ease-in` – Start slow, end fast.

```
css
transition: all 0.3s ease-in;
```

(Jhatke se khatam hoga, sudden effect lagega)

4. `ease-out` – Start fast, end slow.

```
css
transition: all 0.3s ease-out;
```

(Pehle fast chalega, phir smoothly rukega, premium feel aayega!)

5. `ease-in-out` – Start aur end slow, beech me fast.

```
css
transition: all 0.5s ease-in-out;
```

(Best animation effect, smooth aur natural lagta hai!)

6. `linear` – Same speed har time.

```
css
transition: all 0.3s linear;
```

(Bilkul straight forward, no slow-fast effect, robotic feel!)

7. `steps(n, start/end)` – Step-by-step effect.

```
css
transition: all 0.5s steps(5);
```

(5 steps me transition hoga, jisme har step pe thoda thoda change hoga)

How to Use more than one Animatable property in with transition

- ✔ **Used Two Animatable Property**
- ✔ **1st: transform: scale();**
- ✔ **2nd: box-shadow: x-axis y-axis blur-radius spread-radius color; (box-shadow takes 6 values in total seperated by space)**

Pro-Level Example

```
css
.skills-card-container {
    transition: transform 0.4s ease-in-out, box-shadow 0.4s ease-in-out;
}

.skills-card-container:hover {
    transform: scale(1.1);
    box-shadow: 0px 8px 16px white;
}
```

Kya Hoga?

Mouse hover pe **10% zoom-in hoga**
Shadow aur dark aur bada hoga (depth effect)
Smooth transition hoga (Dheere dheere effect aayega)

Lets Learn About Box Shadow

Syntax –

box-shadow: x-axis y-axis blur-radius spread-radius color;

- ✔ **box-shadow contains 6 values in total seperated by space.**
- ✔ **all values unit will be in "px" except color.**

Box Shadow ka Concept

```
css
.skills-card-container {
    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.2);
}
```

Iska Breakdown:

- **`box-shadow`** → Yeh ek shadow banata hai kisi element ke neeche, jo depth effect deta hai.
- **`0px`** → X-axis shadow (left-right move), yahan 0px hai toh side me nahi jayega.
- **`4px`** → Y-axis shadow (upar neeche move), yahan **4px neeche** shadow dikhega.
- **`6px`** → Blur radius, jitna zyada hoga, utna smooth shadow dikhega.
- **white** → Colour

Shadow aur mast chahiye toh spread radius bhi use kar sakta hai:

```
css
box-shadow: 0px 4px 10px 3px black;
```

Yahan `10px` **blur radius** aur `3px` **spread radius hai**, jo shadow ko aur phailayega.

Text me Shadow add karna hai to use text-shadow

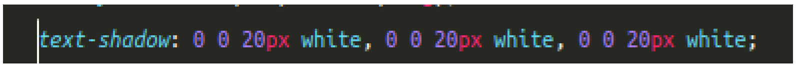
- Ye 4 values leta hai: x-axis y-axis blur-radius and color.
- Saari values space seperated hoti hai
- syntax same as box-shadow bas (isme spread-radius hat jaata hai).

Important baat:

<svg> element ke liye n text-shadow work karti hai or na box-shadow. Ok remember.

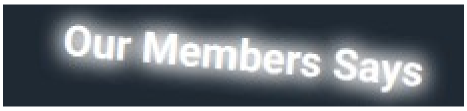
Ham text-shadow and box-shadow me eise bhi values de sakte hai

- **Eise karne se glow effect jyada aata hai same property repeat kar do comma se seperate karke. .**



```
text-shadow: 0 0 20px white, 0 0 20px white, 0 0 20px white;
```

- **Dekho isme effect jyada aata hai like below .**


Our Members Says