Got it ✅ — you want an **Angular + GSAP** animation where an element changes from its original color to another color, then smoothly returns to the original.

Here's a **complete runnable Angular example** using **GSAP**'s `to()` and `yoyo` features.

1️⃣ Install GSAP

```
npm install gsap
```

2️⃣ Angular Component Example

```
// color-animate.component.ts
import { Component, ElementRef, ViewChild, AfterViewInit } from
'@angular/core';
import { gsap } from 'gsap';

@Component({
  selector: 'app-color-animate',
  template: `
    <div #box class="color-box">
      Animate Me
    </div>
    <button (click)="animateColor()">Start Animation</button>
  `,
  styles: [`
    .color-box {
      width: 200px;
      height: 100px;
      line-height: 100px;
      text-align: center;
      background-color: #3498db; /* Original color */
      color: white;
      font-weight: bold;
      border-radius: 8px;
      margin: 20px auto;
      transition: background-color 0.3s ease;
    }
```

```
  `]
})
export class ColorAnimateComponent implements AfterViewInit {
  @ViewChild('box') box!: ElementRef<HTMLDivElement>;
  private originalColor = '#3498db'; // Store original color

  ngAfterViewInit() {
    // Ensure GSAP knows the starting color
    gsap.set(this.box.nativeElement, { backgroundColor:
this.originalColor });
  }

  animateColor() {
    gsap.to(this.box.nativeElement, {
      backgroundColor: '#e74c3c', // Target color
      duration: 1,
      yoyo: true, // Go back to original
      repeat: 1,  // Repeat once (forward + backward)
      ease: 'power1.inOut'
    });
  }
}
```

3️⃣ How It Works

1. `yoyo: true` makes GSAP reverse the animation after completion.
2. `repeat: 1` means it will play forward once, then backward once.
3. `backgroundColor` is tweened smoothly (GSAP handles color interpolation).
4. `ease` controls the smoothness of the transition.

4️⃣ Example Behavior

1. Starts at **blue** (#3498db).
2. Animates to **red** (#e74c3c) in 1 second.
3. Returns to **blue** in another 1 second.

💡 **Tip:** If you want this to happen automatically on load, call `this.animateColor()` inside `ngAfterViewInit()` instead of on button click.

If you want, I can also give you a **version with infinite looping color changes** using GSAP's `repeat: -1` and `yoyo: true`.
Do you want me to prepare that?