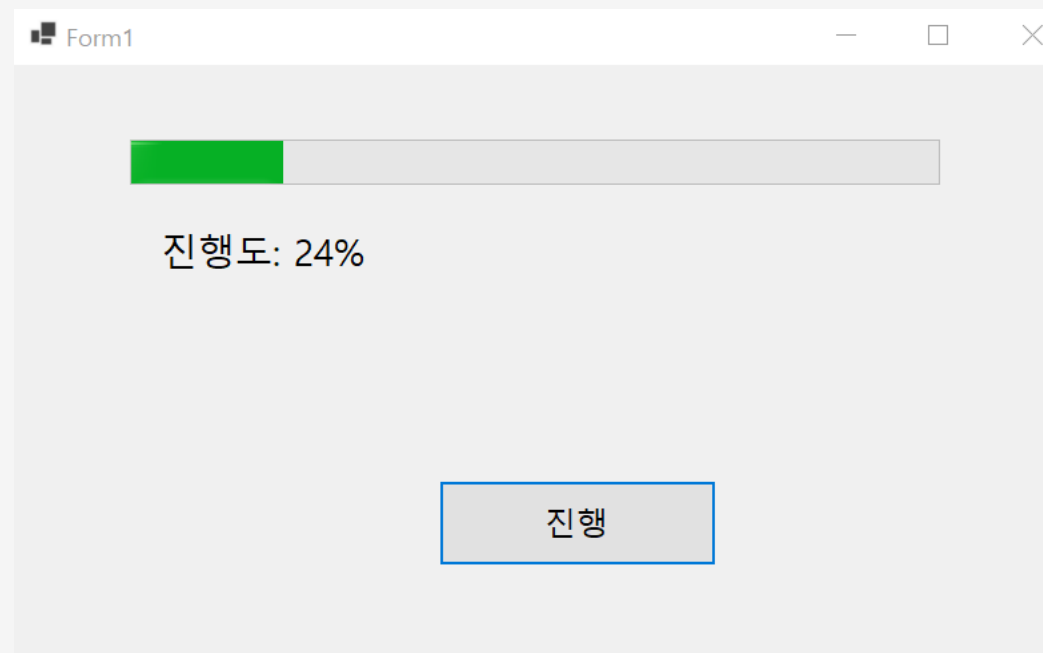


# ProgressBar

# 실습 ProgressBar를 진행도 표현

---

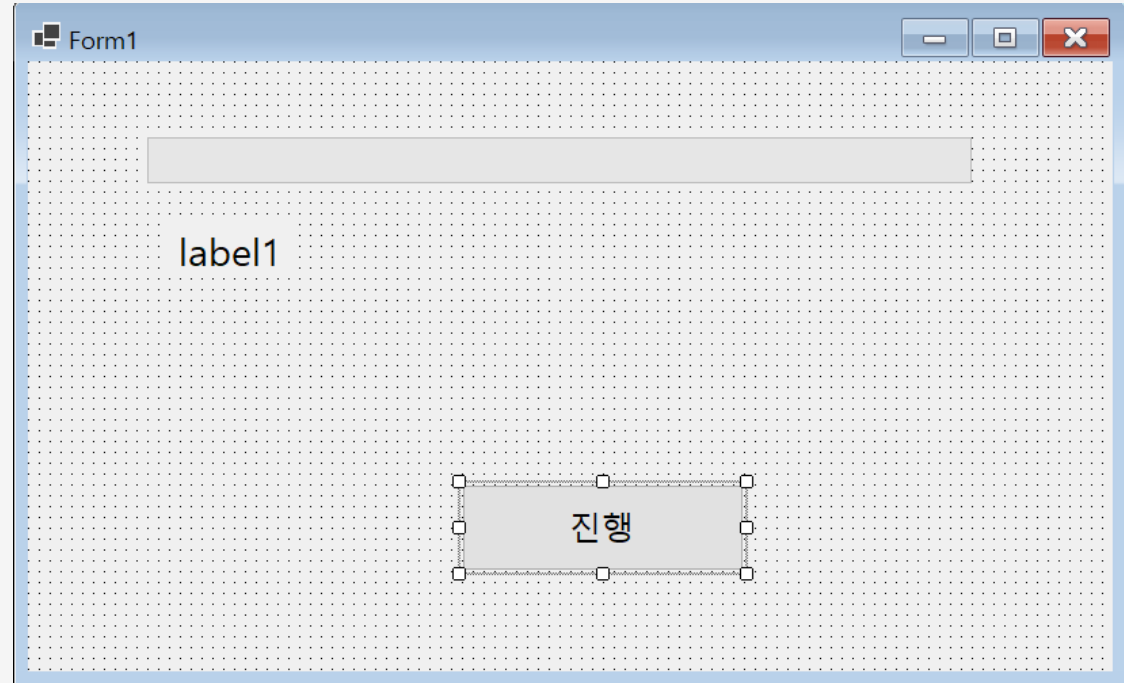
- 프로그레스바를 이용한 진행도 표현
- 프로그램의 진행상황을 그래프로 표현해줌
- 라벨에 진행값을 적용하여 숫자로 보여줄 수 있음



# 실습 ProgressBar를 진행도 표현 (계속)

---

- UI 구성
  - progressBar1
  - progressLabel <- label1
  - button1
  - timer1



# 실습 ProgressBar를 진행도 표현 (계속)

- 컨트롤 초기화

- progressBar1 Maximum 값 100
- Minimum 값 0

| 동작                    |        |
|-----------------------|--------|
| ContextMenuStrip      | (none) |
| Enabled               | True   |
| MarqueeAnimationSpeed | 100    |
| Maximum               | 100    |
| Minimum               | 0      |
| ...                   | ...    |

```
private void Form1_Load(object sender, EventArgs e)
{
    progressBar1.Minimum = 0;
    progressBar1.Maximum = 100;
    progressBar1.Value = 0;

    progressLabel.Text = "진행도: 0%";

    timer1.Interval = 100; // 100 milliseconds
}
```

## 실습 ProgressBar를 진행도 표현 (계속)

- 버튼 클릭 시 타이머가 진행
  - 타이머 시작 전 컨트롤 초기화
    - progressBar.Value 0로 설정
    - progressBar1.Value 0으로 설정
    - progressBar.Text 진행도: 0%로 설정

```
private void button1_Click(object sender, EventArgs e)
{
    progressValue = 0;
    progressBar1.Value = 0;
    progressLabel.Text = "진행도: 0%";
    timer1.Start();
}
```

# 실습 ProgressBar를 진행도 표현 (계속)

- 핵심코드

- 타이머 틱에 따른 프로그레스바 변화

- 0.1초 마다 progressValue 1 증가

- 100% 진행까지 라벨의 텍스트 변화 후

- 100% 도달하면 타이머 정지

```
private int progressValue; //멤버변수
```

```
private void timer1_Tick(object sender, EventArgs e)
{
    progressValue += 1;
    if (progressValue <= 100)
    {
        progressBar1.Value = progressValue;
        progressLabel.Text = $"진행도: {progressValue}%";
    }
    else
    {
        timer1.Stop();
        MessageBox.Show("진행 완료!");
    }
}
```

## 실습 ProgressBar를 진행도 표현 (완성)

---

