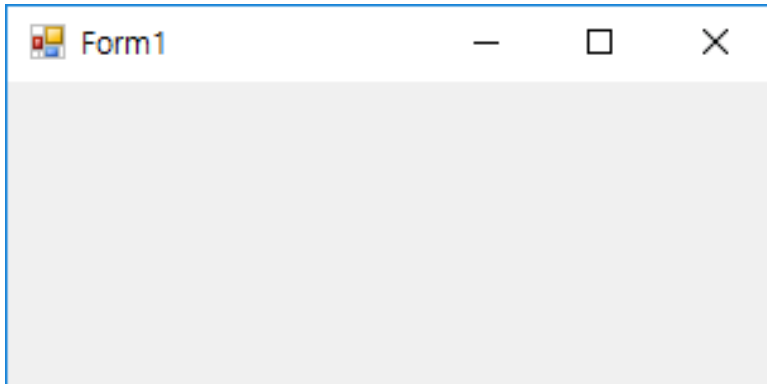


## ■ 원폼 애플리케이션

- 컨트롤을 사용하여 프로그래머가 원하는 화면을 구성하고 이벤트가 발생했을 때 처리하고자 하는 작업을 이벤트 처리기에 기술하는 방식으로 프로그래밍된 프로그램.
- 윈도우 폼에 컨트롤 또는 컴포넌트를 배치하고 이벤트 처리기를 등록하여 사용자의 다양한 요구를 입력으로 받아 처리한 후 실행 결과를 응답해 주는 방식으로 작동.

## ■ 윈도우 폼

- 단순히 폼이라고 부름.
- 운영체제에서 제공하는 기본적인 화면 단위인 창을 말하는 개념.
- 사각형 모양의 작은 화면 영역을 의미
- 사용자에게 정보를 제공하고 사용자가 입력하는 자료를 받음.



# 원폼 애플리케이션의 핵심 클래스

## ■ 폼 클래스

- 윈도우 폼을 나타내는 클래스.
- System.Windows.Forms 네임스페이스 속함.

## ■ 컴포넌트 클래스

- 컨트롤 클래스의 베이스 클래스이며 화면에 직접적으로 나타나지 않으나 개념적인 부분을 나타내는 클래스.

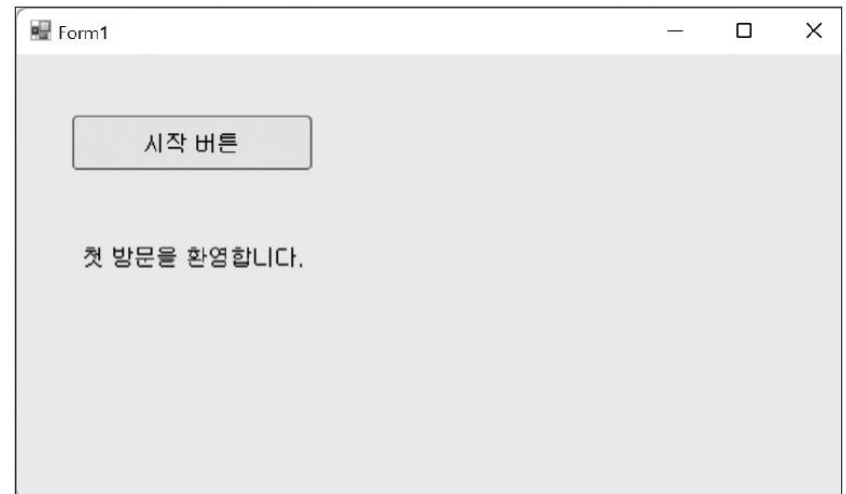
## ■ 컨트롤 클래스

- 폼에 직접 표시되는 컨트롤을 위한 클래스.

# 1. 프로젝트 생성

## ■ 원폼 화면

- 콘솔 창이 아닌 윈도우 폼 화면에서 C# 프로그램 실행 결과를 확인
- 새 프로젝트는 C#에서 'Windows Forms 앱(.NET Framework)'을 선택하여 생성



- **Step 01** | 프로젝트 생성 : 화면에서 [새 프로젝트 만들기] 항목을 선택합니다.

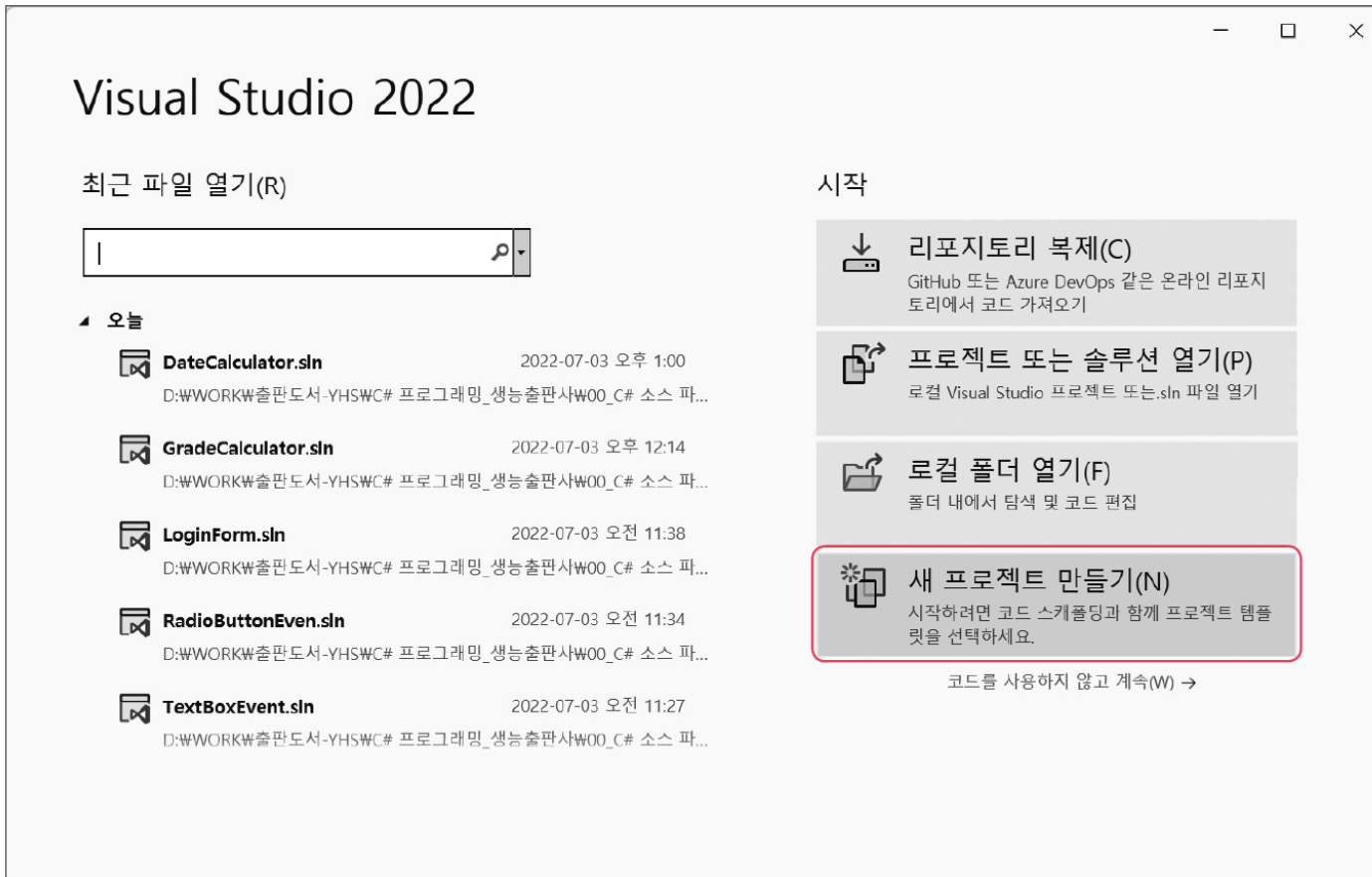


그림 9-1 새 프로젝트 만들기

- **Step 02** | 프로젝트 종류 선택 : ‘Windows Forms 앱(.NET Framework)’을 선택 후 <다음>을 누릅니다.

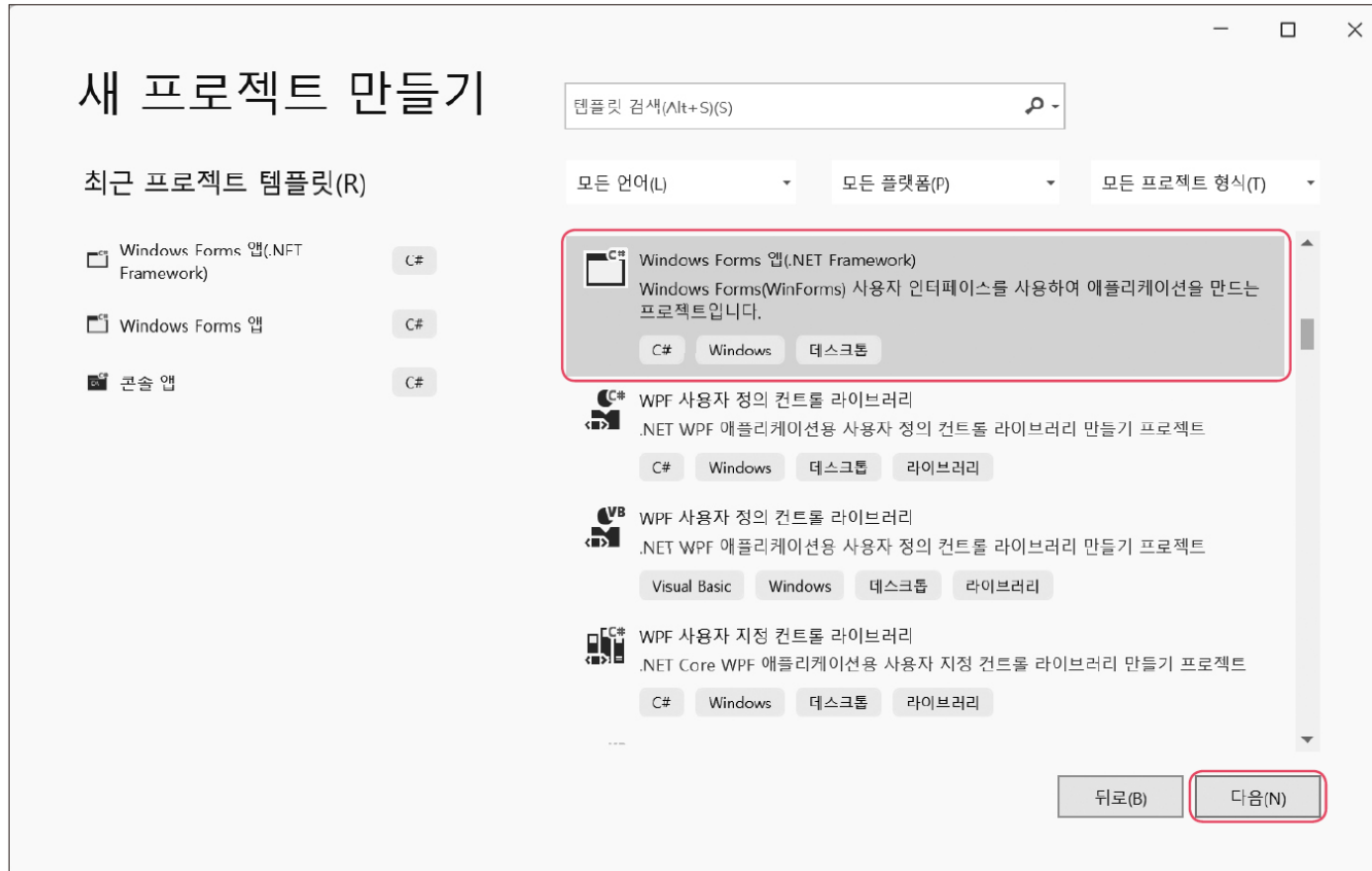


그림 9-2 원폼 앱 프로젝트 선택

- **Step 03** | 프로젝트명 입력과 경로 지정 : 프로젝트명은 'FirstFormApp'으로 입력하고 소스 파일명은 그대로 둡니다. 경로명은 각자 알아서 지정합니다.

새 프로젝트 구성

Windows Forms 앱(.NET Framework) C# Windows 데스크톱

프로젝트 이름(I)

FirstFormApp

위치(L)

D:\WORK\출판도서-YHS\WC# 프로그래밍\_생능출판사\W00\_C# 소스 파일\ch09\W

솔루션 이름(M) ⓘ

FirstFormApp

☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

프레임워크(F)

.NET Framework 4.7.2

뒤로(B) 만들기(C)

그림 9-3 원폼 앱 프로젝트 선택

• **Step 04** | 원폼 편집 화면 : 새 프로젝트가 생성되면 원폼 편집 화면을 보여줍니다.

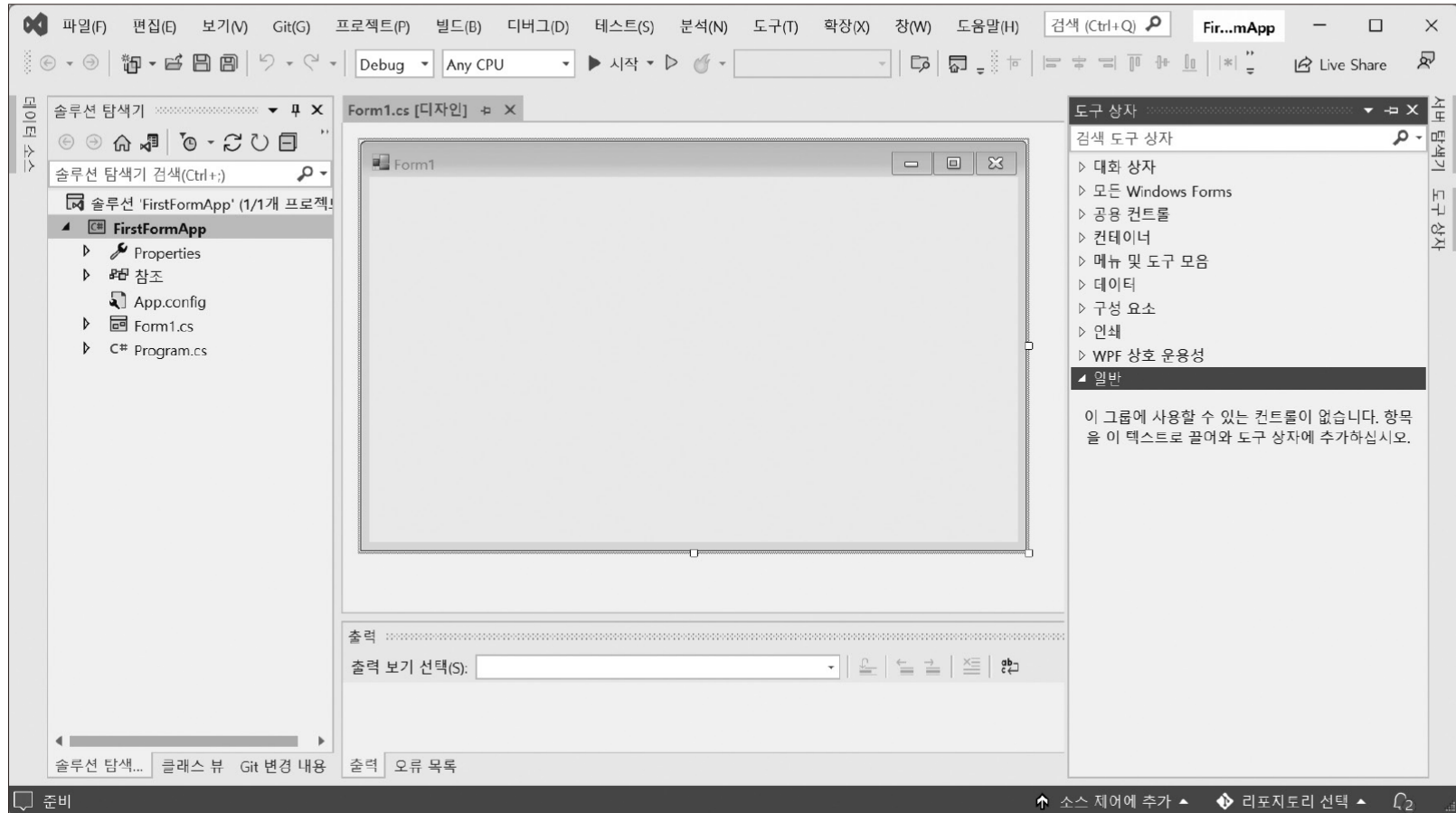


그림 9-4 원폼 편집 화면



## 2. 원폼 기본 구조

### ■ 프로젝트의 기본 구조

- [솔루션 탐색기] 창에서 각 항목 옆에 있는 삼각형(▷)을 눌러보면 원폼 디자인 응용 프로그램에 대한 기본 구조를 확인할 수 있음
- 이 중에서 3개(Program.cs / Form1.Designer.cs / Form1.cs) 파일에 대한 소스 코드와 역할에 대해 간략하게 살펴보기로 함



그림 9-5 프로젝트의 기본 구조

## 2. 원품 기본 구조

### ■ Program.cs 파일

- [솔루션 탐색기] 창에서 Program.cs 파일명을 더블 클릭하게 되면 오른쪽 화면에서 소스 코드를 확인할 수 있음
- 이 소스 코드는 비주얼 스튜디오에서 자동으로 생성되는 파일
- 별도의 작업은 필요 없음

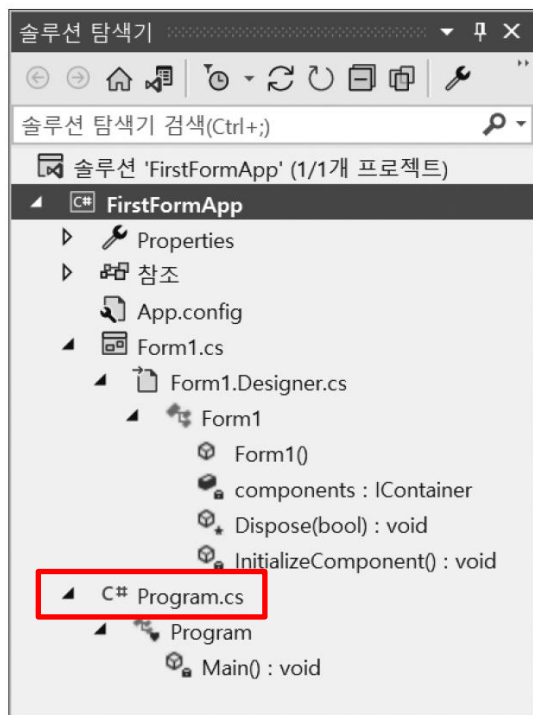


그림 9-5 프로젝트의 기본 구조

```
01 using System;
02 using System.Collections.Generic;
03 using System.Linq;
04 using System.Threading.Tasks;
05 using System.Windows.Forms;
06
07 namespace FirstFormApp
08 {
09     internal static class Program
10     {
11         /// <summary>
12         /// 해당 애플리케이션의 주 진입점입니다.
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new Form1());
20         }
21     }
22 }
```

## 2. 원폼 기본 구조

### ■ Form1.Designer.cs 파일

- 원폼 화면인 Form1 클래스의 디자인을 지정
- 이 파일 또한 프로젝트를 생성하게 되면 자동으로 제공
- 소스 파일은 건드리지 않고 그냥 구성된 내용만 살펴봄

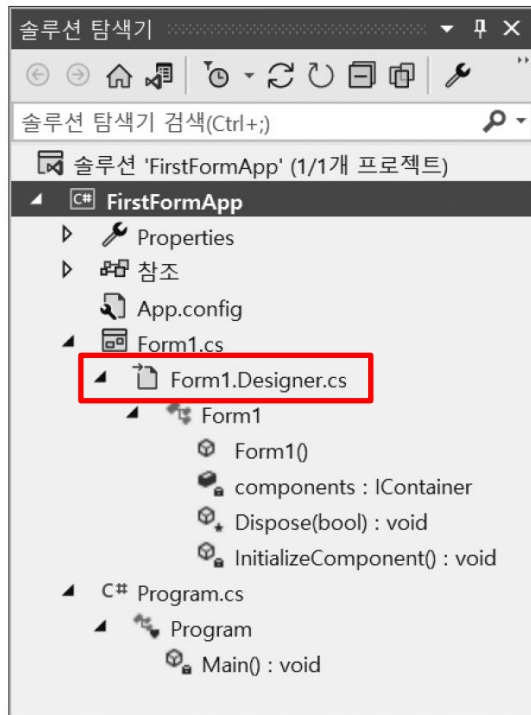


그림 9-5 프로젝트의 기본 구조

```
01 namespace FirstFormApp
02 {
03     partial class Form1
04     {
05         /// <summary>
06         /// 필수 디자이너 변수입니다.
07         /// </summary>
08         private System.ComponentModel.IContainer components = null;
09
10         /// <summary>
11         /// 사용 중인 모든 리소스를 정리합니다.
12         /// </summary>
13         /// <param name="disposing">관리되는 리소스를 삭제해야 하면 true이고,
14         /// 그렇지 않으면 false입니다.</param>
15         protected override void Dispose(bool disposing)
16         {
17             if (disposing && (components != null))
18             {
19                 components.Dispose();
20             }
21             base.Dispose(disposing);
22     }
```

```
23
24     #region Windows Form 디자이너에서 생성한 코드
25
26     /// <summary>
27     /// 디자이너 지원에 필요한 메서드입니다.
28     /// 이 메서드의 내용을 코드 편집기로 수정하지 마세요.
29     /// </summary>
30     private void InitializeComponent()
31     {
32         this.components = new System.
33             ComponentModel.Container();
34         this.AutoScaleMode = System.Windows.
35             Forms.AutoScaleMode.Font;
36         this.ClientSize = new System.Drawing.Size(800, 450);
37         this.Text = "Form1";
38     }
39
40     #endregion
41 }
42 }
```

# 소스코드 보기 [1/4]

## ■ 생성된 프로젝트의 파일

- Program.cs: 원폼 응용 프로그램의 시작점을 포함하는 C# 소스파일
- Form1.cs: 원폼 응용 프로그램의 C# 소스파일
- Form1.Designer.cs: 원폼 응용 프로그램 폼 디자인 정의를 포함하는 C# 소스파일
- WindowsApplication1.csproj: 원폼 응용 프로그램 프로젝트 파일
- Properties/AssemblyInfo.cs: 프로젝트가 생성하는 어셈블리를 설명하고 버전 관리 정보를 지정하는 데 사용하며, 애트리뷰트 정의를 포함하는 C# 소스파일
- Properties/Resources.Designer.cs: 원폼 응용 프로그램의 자원에 대한 C# 정의를 포함하는 C# 소스파일
- Properties/Resources.resx: 원폼 응용 프로그램의 자원 파일
- Properties/Settings.Designer.cs: 프로젝트 설정에 대한 C# 정의를 포함하는 C# 소스파일
- Properties/Settings.settings : 프로젝트에 대한 설정 파일

## ■ Program.cs

```
static class Program
{
    /// <summary>
    /// 해당 응용 프로그램의 주 진입점입니다.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());        // --- ①
    }
}
```



## ■ Form1.cs

```
public partial class Form1 : Form
{
    public Form1()                // --- ②
    {
        InitializeComponent();
    }
}
```

# 소스코드 보기 [4/4]

## ■ ① Main() 메소드

- 응용 프로그램의 시작점
- Application 클래스의 Run() 메소드를 호출하여 응용 프로그램을 실행.

## ■ ② 생성자

- 폼에서 사용하는 각종 컴포넌트와 클래스의 멤버 초기화
- 폼에 있는 각종 컴포넌트들 초기화하는 InitializeComponent() 메소드 호출

# 원폼 애플리케이션 작성하기 [1/3]

## ■ 디자인

- 컨트롤
- 컴포넌트
- 프로퍼티
- 이벤트

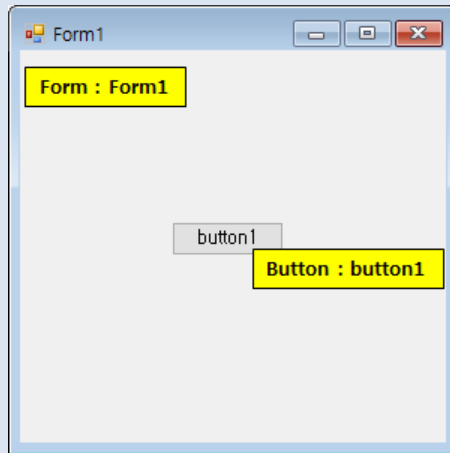
## ■ 코드

- 멤버
- 이벤트처리기

## ■ 애플리케이션 실행

[예제 7.1 – DisplayStartDateApp.cs]

1) 디자인



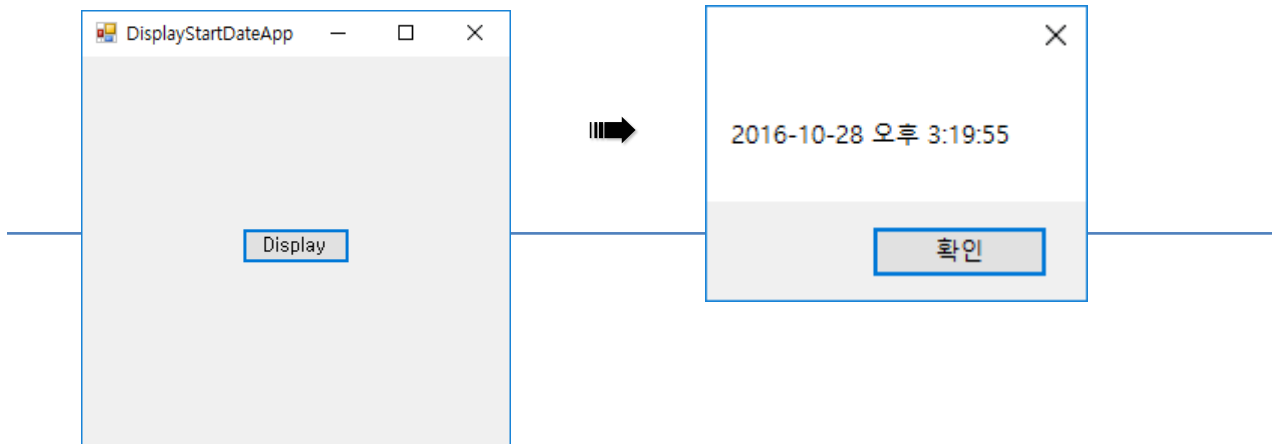
컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	DisplayStartDateApp
Button : button1	Text	Display
컨트롤 : (Name)	이벤트	메소드명
Button : button1	Click	button1_Click()

2) 코드

```
public Form1() {  
    //...  
    startDateTime = DateTime.Now;  
}  
DateTime startDateTime;  
public DateTime GetStartDateTime() {  
    return startDateTime;  
}  
private void button1_Click(object sender, EventArgs e) {  
    MessageBox.Show(GetStartDateTime().ToString());  
}
```

실행 방법 : 애플리케이션을 실행한 후, 폼에 있는 "Display" 버튼을 클릭한다.

실행 결과 :



# (1) 디자인

## ■ 컨트롤 배치

- 폼에 컨트롤을 배치한 모습 표시.
- 폼에 배치되는 컨트롤의 종류와 배치된 컨트롤의 이름과 위치를 그림 형태로 표시.
- 노란색 텍스트 상자는 "ClassType : Name" 형식으로 표시
  - 컨트롤의 종류(클래스 이름)와 이름(객체 이름)을 나타냄.

## ■ 컴포넌트 목록

컨트롤 : (Name)	프로퍼티	값
Timer : timer1	Images	FLGSKOR.ICO

## ■ 프로퍼티 목록

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	DisplayStartDateApp
Button : button1	Text	Display

## ■ 이벤트 처리기

- 통합 환경에서 생성하는 처리기의 이름
  - <이벤트 처리기를 등록하려는 객체의 이름>\_<이벤트 이름>

컨트롤 : (Name)	이벤트	메소드명
Button : button1	Click	button1_Click()

## (2) 코드

### ■ 멤버

- 컴포넌트와 컨트롤을 제외한 클래스 멤버에 대한 선언과 초기화 등을 소스코드로 작성하는 곳.

### ■ 멤버 코드 추가

- 생성자에 초기화 부분 추가

```
public Form1(){  
    InitializeComponent();  
  
    //  
    startDateTime = DateTime.Now;  
}
```

- 멤버 선언이나 메소드 등의 소스 코드 추가
  - 폼 클래스내의 적당한 곳에 추가.



## (5) 이벤트 처리기 [1/2]

### ■ 이벤트 처리기

#### ■ 매개변수

```
private void button1_Click(object sender, EventArgs e) {
    MessageBox.Show(GetStartDateTime().ToString());
}
```

- 이벤트를 발생시킨 객체
- 이벤트에 관련된 정보를 가진 객체
  - EventArgs 클래스형이나 이의 파생 클래스형

### 3. 디자인 화면 요소

#### ■ 도구상자 활성화

- 메뉴에서 [보기]-[도구상자]를 선택하여 도구상자를 활성화

예제 09-02

원폼 화면에 버튼과 레이블 추가하기

- **Step 01** | 기존 프로젝트 사용 : [예제 09-01]에서 생성한 프로젝트의 원폼 디자인 화면에서 작업을 이어서 수행합니다.
- **Step 02** | 도구상자 활성화 : 원폼 디자인 화면에 디자인 요소를 선택하기 위해 메뉴에서 [보기]-[도구상자]를 선택하여 도구상자를 열어줍니다.

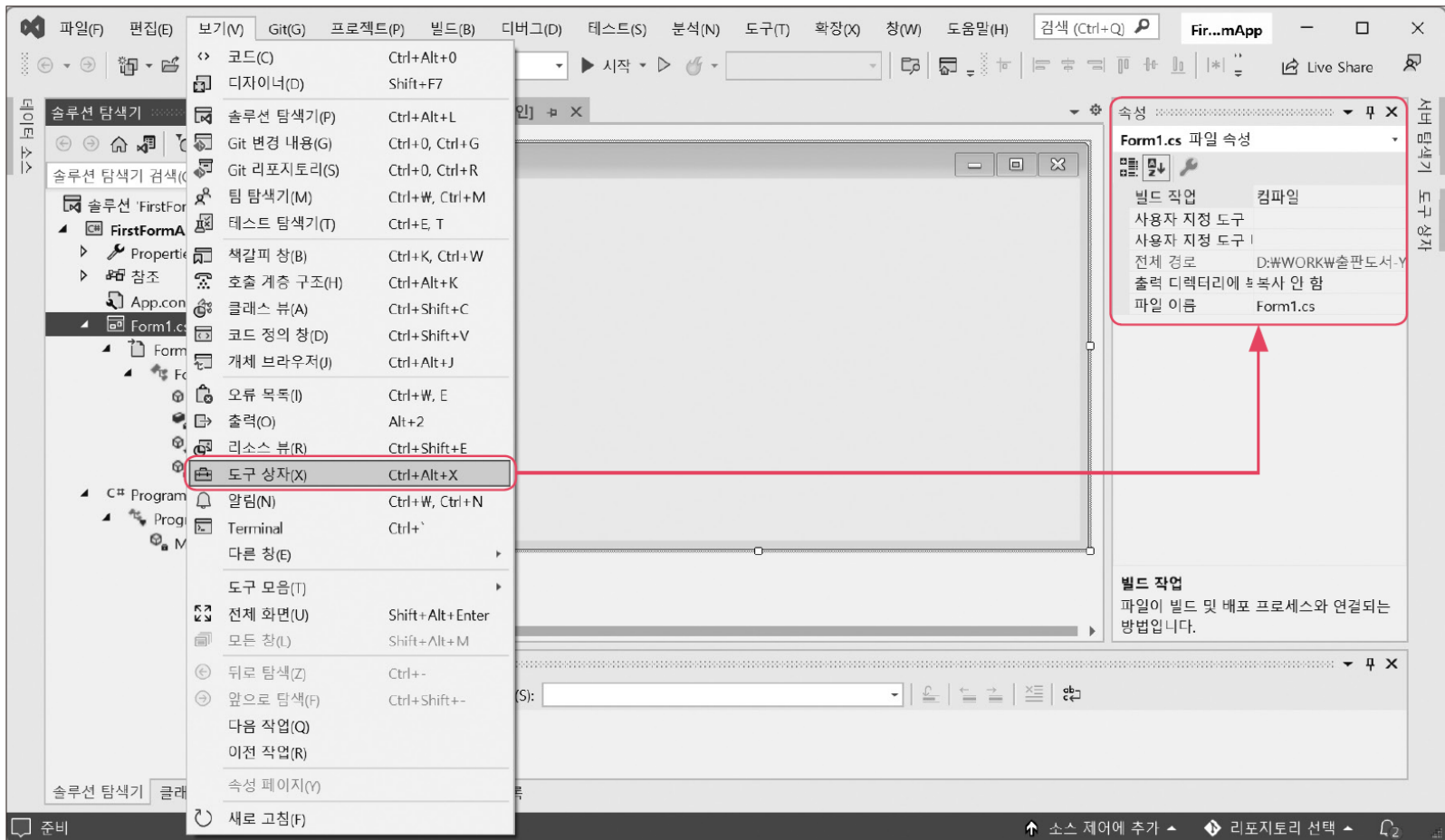


그림 9-7 도구상자 창 활성화

- **Step 03** | 버튼 추가 : 원폼 디자인 화면에 버튼을 추가합니다. 원폼 디자인 화면에 추가된 버튼의 이름은 button1과 같이 자동으로 부여됩니다.

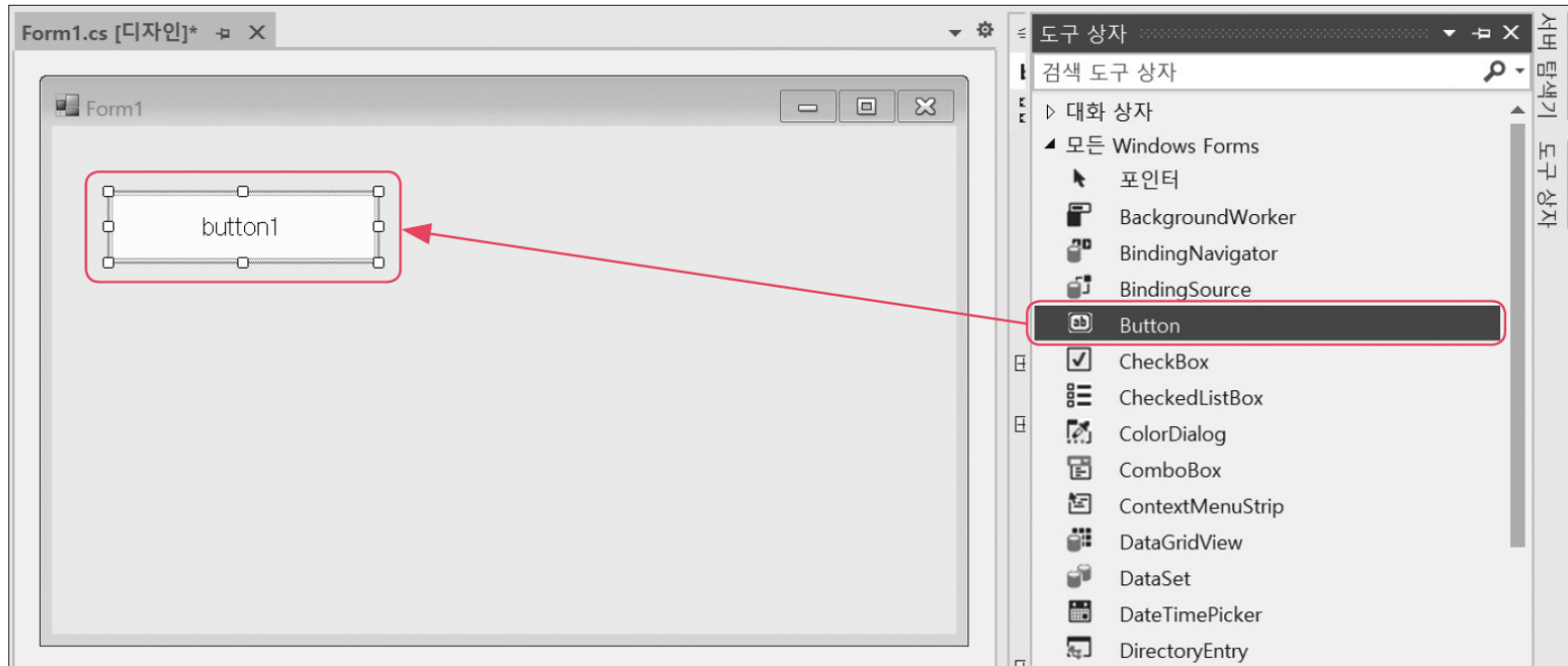


그림 9-8 원폼 디자인 화면에 버튼 추가

- **Step 04** | 레이블 추가 : 원폼 디자인 화면에 레이블을 추가합니다.

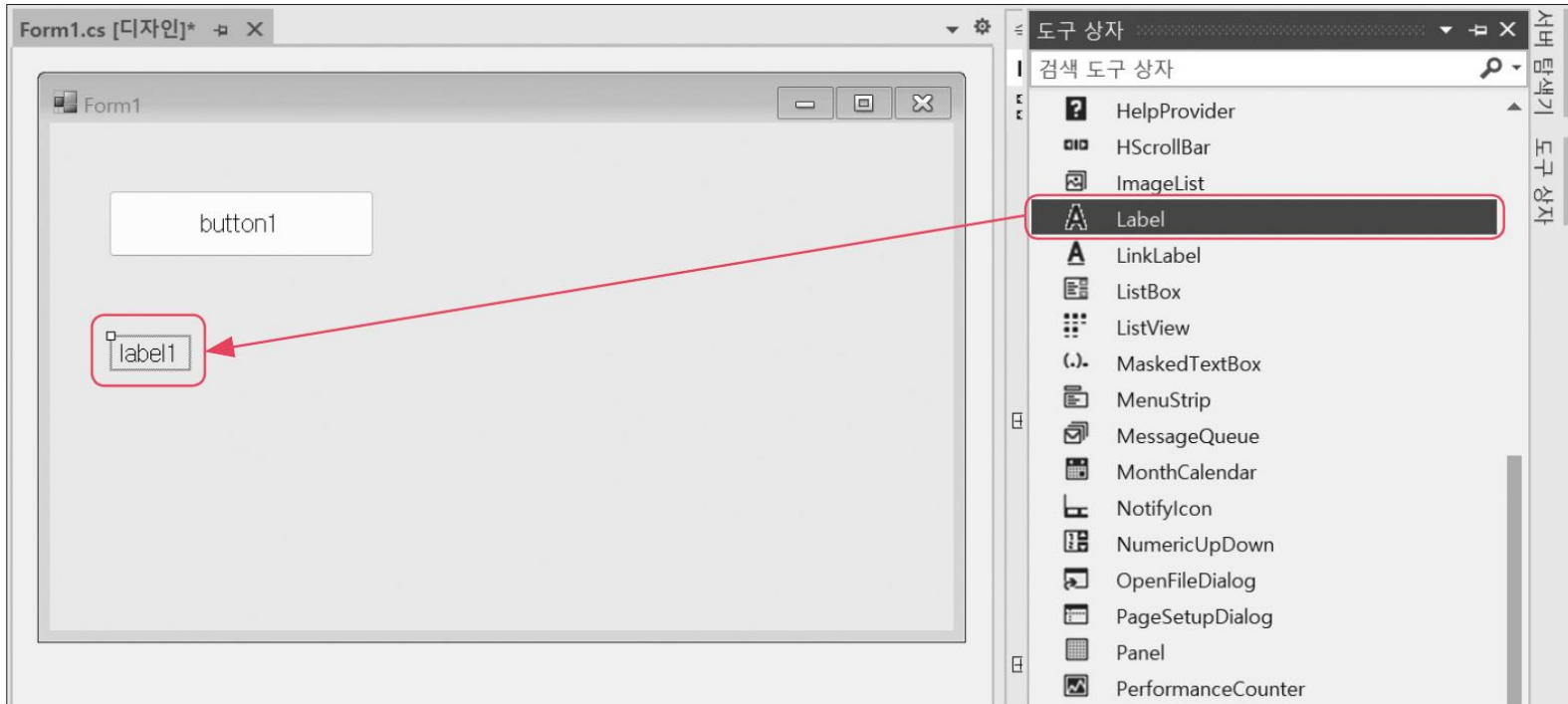


그림 9-9 원폼 디자인 화면에 레이블 추가

- **Step 01** | 버튼의 좌표 위치 설정 : 속성 창에서 아이콘으로 표기된 [속성]을 누릅니다. 그런 다음 Location 항목에서 원폼 화면에 배치된 버튼의 좌표값 x, y의 속성을 설정합니다.

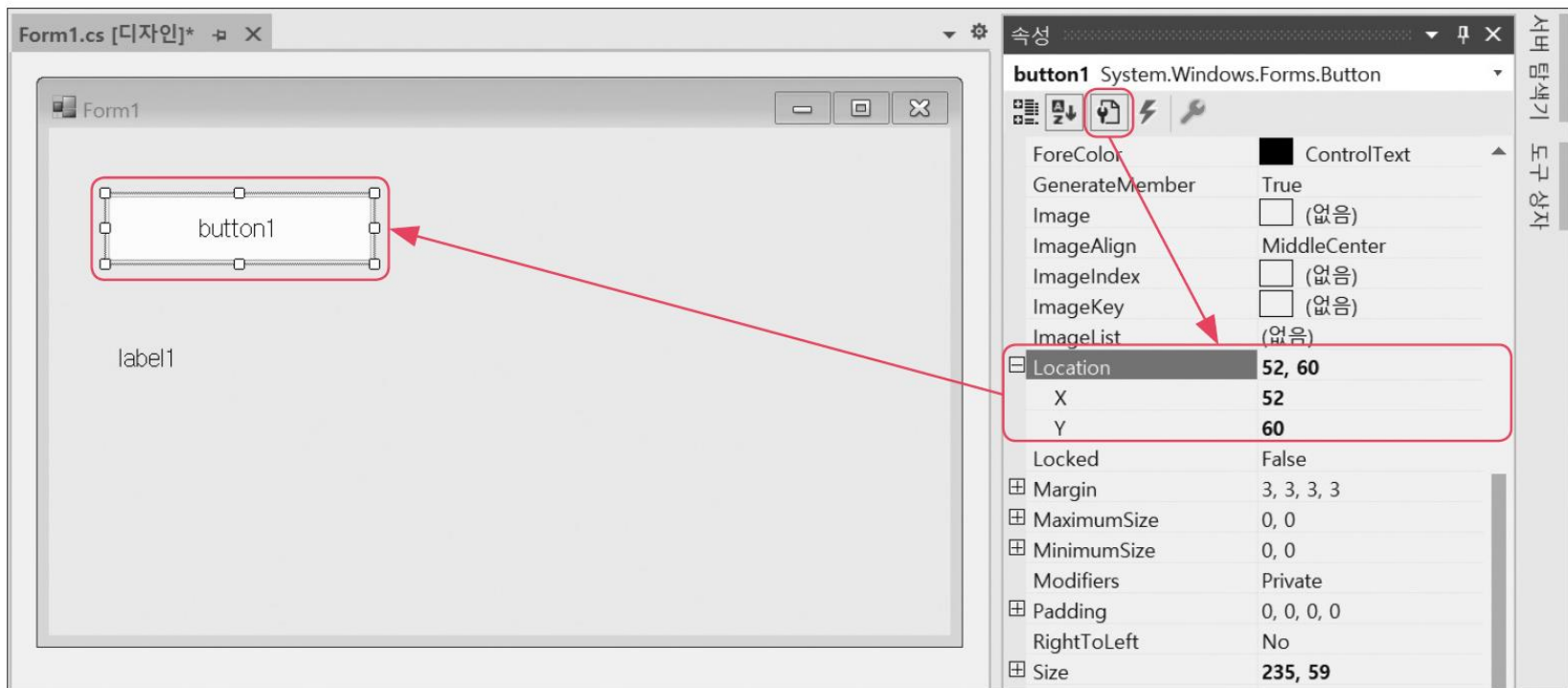


그림 9-10 버튼의 좌표값 속성 설정

- **Step 02** | 버튼의 텍스트 속성 변경 : 버튼의 Text 속성을 '시작 버튼'으로 변경합니다.

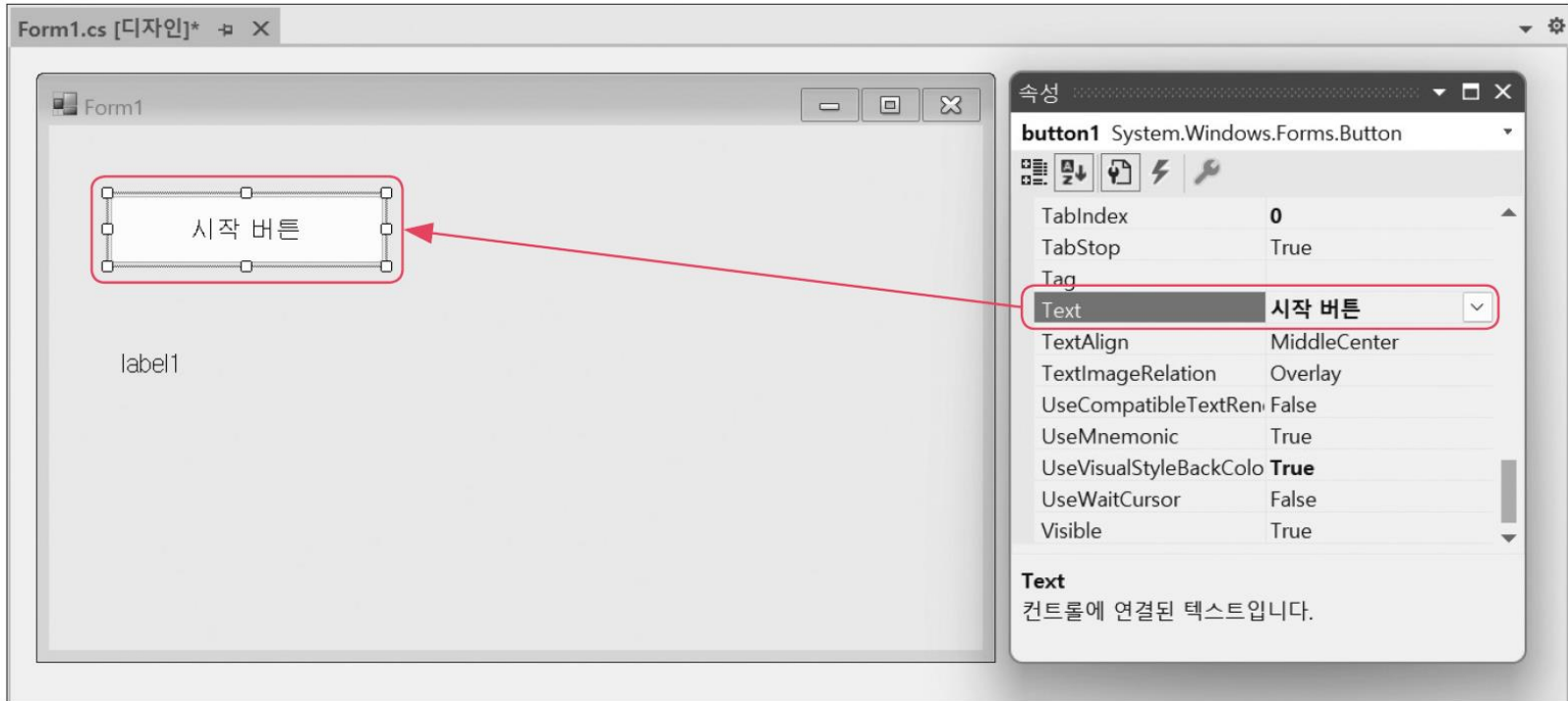


그림 9-11 버튼에 표기된 텍스트 속성 변경

## 4. 디자인 코드

### ■ Program.cs 파일 확인

- 버튼을 추가하기 전 소스 코드와 특별하게 달라진 부분은 없음

ch09/FirstFormApp/FirstFormApp/Program.cs

```
01 using System;
02 using System.Collections.Generic;
03 using System.Linq;
04 using System.Threading.Tasks;
05 using System.Windows.Forms;
06
07 namespace FirstFormApp
08 {
09     internal static class Program
10     {
11         /// <summary>
12         /// 해당 애플리케이션의 주 진입점입니다.
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new Form1());
20         }
21     }
22 }
```



## 4. 디자인 코드

### ■ Form1.Designer.cs 파일 확인

- 소스 코드는 기존 42행에서 77행으로 추가된 것을 확인할 수 있음
- 소스 코드 라인은 컴퓨터 환경에 따라 다르게 나타날 수 있음

ch09/FirstFormApp/FirstFormApp/Form1.Designer.cs

```
01 namespace FirstFormApp
02 {
03     partial class Form1
04     {
05         /// <summary>
06         /// 필수 디자이너 변수입니다.
07         /// </summary>
08         private System.ComponentModel.IContainer components = null;
09
10         /// <summary>
11         /// 사용 중인 모든 리소스를 정리합니다.
12         /// </summary>
13         /// <param name="disposing">관리되는 리소스를 삭제해야 하면 true이고,
14         /// 그렇지 않으면 false입니다.</param>
```

```
15         protected override void Dispose(bool disposing)
16     {
17         if (disposing && (components != null))
18         {
19             components.Dispose();
20         }
21         base.Dispose(disposing);
22     }
23
24     #region Windows Form 디자이너에서 생성한 코드
25
26     /// <summary>
27     /// 디자이너 지원에 필요한 메서드입니다.
28     /// 이 메서드의 내용을 코드 편집기로 수정하지 마세요.
29     /// </summary>
30     private void InitializeComponent()
31     {
32         this.button1 = new System.Windows.Forms.Button();
33         this.label1 = new System.Windows.Forms.Label();
34         this.SuspendLayout();
```

```
35         //
36         // button1
37         //
38         this.button1.Font = new System.Drawing.Font("굴림", 11F);
39         this.button1.Location = new System.Drawing.Point(52, 60);
40         this.button1.Name = "button1";
41         this.button1.Size = new System.Drawing.Size(235, 59);
42         this.button1.TabIndex = 0;
43         this.button1.Text = "시작 버튼";
44         this.button1.UseVisualStyleBackColor = true;
45         //
46         // label1
47         //
48         this.label1.AutoSize = true;
49         this.label1.Font = new System.Drawing.Font("굴림", 11F);
50         this.label1.Location = new System.Drawing.Point(58, 193);
51         this.label1.Name = "label1";
52         this.label1.Size = new System.Drawing.Size(62, 22);
53         this.label1.TabIndex = 1;
54         this.label1.Text = "label1";
```

```
55         //
56         // Form1
57         //
58         this.AutoScaleMode = new System.
59             Drawing.SizeF(10F, 18F);
60         this.AutoScaleMode = System.
61             Windows.Forms.AutoScaleMode.Font;
62         this.ClientSize = new System.Drawing.Size(800, 450);
63         this.Controls.Add(this.label1);
64         this.Controls.Add(this.button1);
65         this.Name = "Form1";
66         this.Text = "Form1";
67         this.ResumeLayout(false);
68         this.PerformLayout();
69
70     }
71
72     #endregion
73
```

```
74         private System.Windows.Forms.Button button1;  
75         private System.Windows.Forms.Label label1;  
76     }  
77 }
```

---

- **Form1.Designer.cs 파일**
  - 이 파일은 사용자가 임의로 수정해선 안 됨

## 4. 디자인 코드

### ■ Form1.cs 파일 확인

- 이 소스 코드는 버튼을 추가하기 전과 특별하게 달라진 부분은 없음

ch09/FirstFormApp/FirstFormApp/Form1.cs

```
01 using System;
02 using System.Collections.Generic;
03 using System.ComponentModel;
04 using System.Data;
05 using System.Drawing;
06 using System.Linq;
07 using System.Text;
08 using System.Threading.Tasks;
09 using System.Windows.Forms;
10
11 namespace FirstFormApp
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19     }
20 }
```

## 5. 코드 요소의 속성

### ■ 각 요소별 고유 이름 부여

- 버튼 등 원폼 디자인 화면에 추가되는 요소에는 각각 고유의 이름이 자동으로 부여됨

예제 09-04

원폼 화면에 배치된 버튼 이벤트 설정하기

- **Step 01** | 버튼 이벤트 설정 : 원폼 화면에 배치된 버튼을 마우스로 더블클릭합니다. 그러면 Form1.cs 파일이 열리게 됩니다. 소스 코드 22행부터 다음과 같이 소스 코드를 추가합니다.

```
// 소스 코드 추가  
label1.Text = "첫 방문을 환영합니다.";
```

- **Step 02** | 객체 리스트 : 소스 코드를 추가하다 보면 알파벳 첫 글자만 입력해도 관련 객체 리스트를 보여줍니다. 리스트에서 객체를 선택해도 되고 키보드로 입력해도 됩니다.



그림 9-12 객체 리스트



- **Step 03** | 객체의 속성 리스트 : 객체명을 입력 후 마침표(.)를 입력하게 되면 해당 객체에 대한 속성 리스트를 보여줍니다. 속성 또한 리스트에서 선택해도 되고 키보드로 입력해도 됩니다.

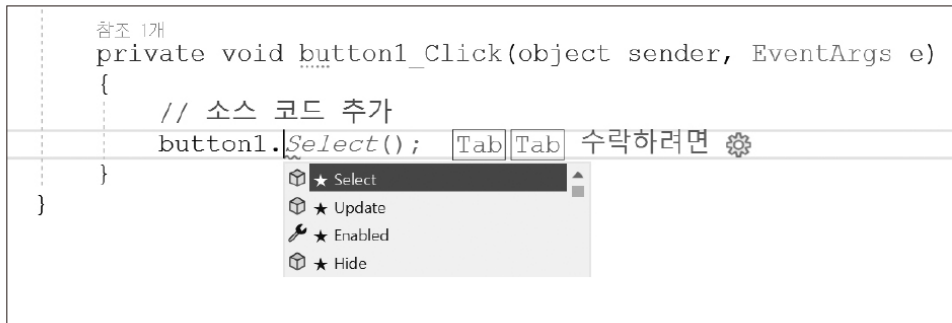


그림 9-13 객체의 속성 리스트

- **Step 04** | 추가된 소스 코드 확인 : 버튼 이벤트를 수행하기 위해 추가된 소스 코드를 확인합니다.

ch09/FirstFormApp/FirstFormApp/Form1.cs

---

```
01 using System;
02 using System.Collections.Generic;
03 using System.ComponentModel;
04 using System.Data;
05 using System.Drawing;
06 using System.Linq;
07 using System.Text;
08 using System.Threading.Tasks;
09 using System.Windows.Forms;
10
11 namespace FirstFormApp
12 {
13     public partial class Form1 : Form
```

```
14      {
15          public Form1()
16          {
17              InitializeComponent();
18          }
19
20          private void button1_Click(object sender, EventArgs e)
21          {
22              // 소스 코드 추가
23              label1.Text = "첫 방문을 환영합니다.";
24          }
25      }
26 }
```

---

- **Step 05** | 프로젝트 실행 : 디자인 화면에서 단축키 **Ctrl+F5** 를 눌러 디버깅하지 않고 프로젝트 실행합니다. <시작 버튼>을 누르게 되면 레이블에 문자열이 출력됩니다.

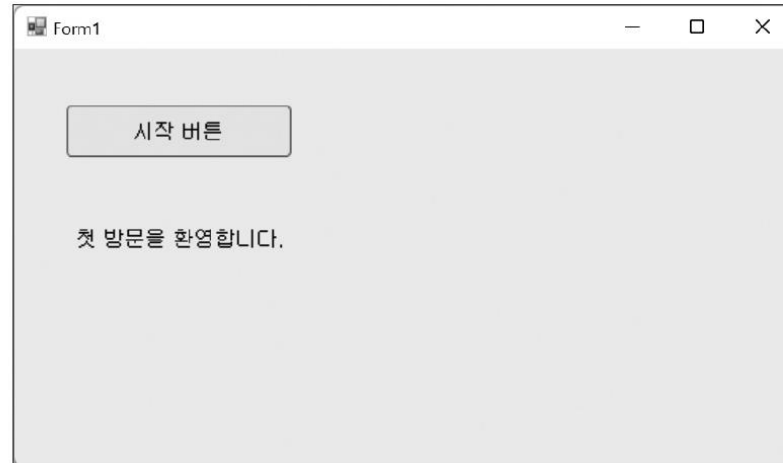


그림 9-14 실행 화면

자신감 뽐뽐!

**C#** 프로그래밍 Hard Carry



*Thank You*