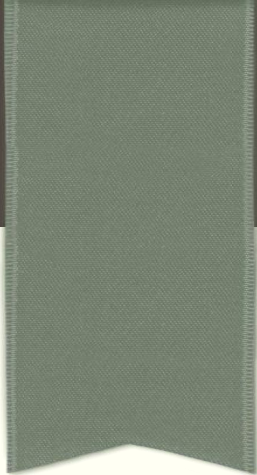


23강. 고급 컨트롤



목차

- 프로그래스 바
- 날짜와 시간 컨트롤
- 이미지 리스트
- 탭 컨트롤
- 리스트 뷰



프로그래스 바

Progress Bar

프로그래스 바 progress bar

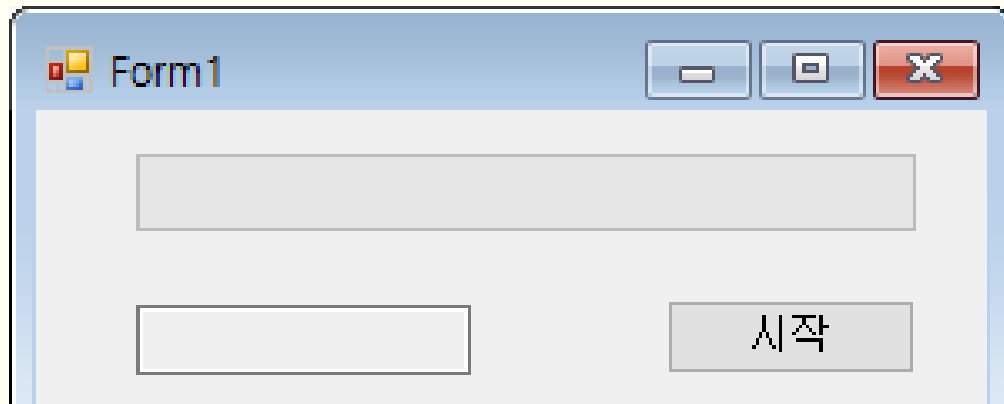
- 역할
진행되는 정도를 막대로 표시
- 기본적인 범위
0 ~ 100 을 상대적인 값으로 막대 표시
- Style 속성을 통해 바 모양 설정

Style	Marquee
TabIndex	Blocks
Value	Continuous
Visible	Marquee

프로그래스 바 progress bar

- ProgressBar.Value 속성
막대바의 위치 지정
- 타이머 + 버튼 + 프로그래스 바
100 밀리세컨드마다 1씩 바의
위치를 진행하며 값을 출력

소스 보기



프로그래스 바 progress bar

- Step 속성에 따라 바의 값을 증가
`ProgressBar.PerformStep()`

[소스 보기](#)

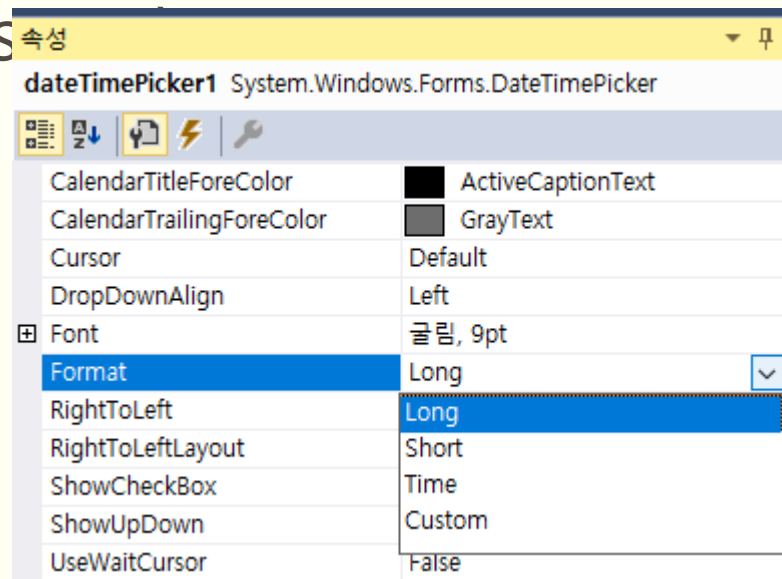
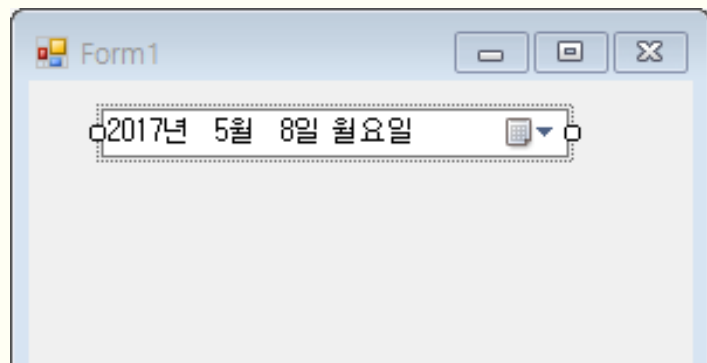


날짜와 시간 컨트롤

DateTimePicker

DateTimePicker

- 날짜와 시간을 선택하고 출력
DateTimePicker
- 형식 설정
DateTimePicker.Format 에
DateTimePickerFormat 열거형 설정
(Long, Short, Time, Custom)



DateTimePicker

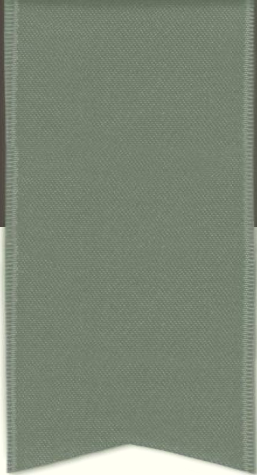
■ 형식 설정

format	설명
Long	년/월/일 형식, 2015년 12월 16일 수요일
Short	년/월/일의 간단한 형식, 2015-12-16
Time	시간 형식, 오후 4:11:30
Custom	사용자가 임의로 지정한 형식

더 다양한 형식은 MSDN을 찾아 볼 것

■ 날짜를 textBox에 출력하는 소그

소스 보기

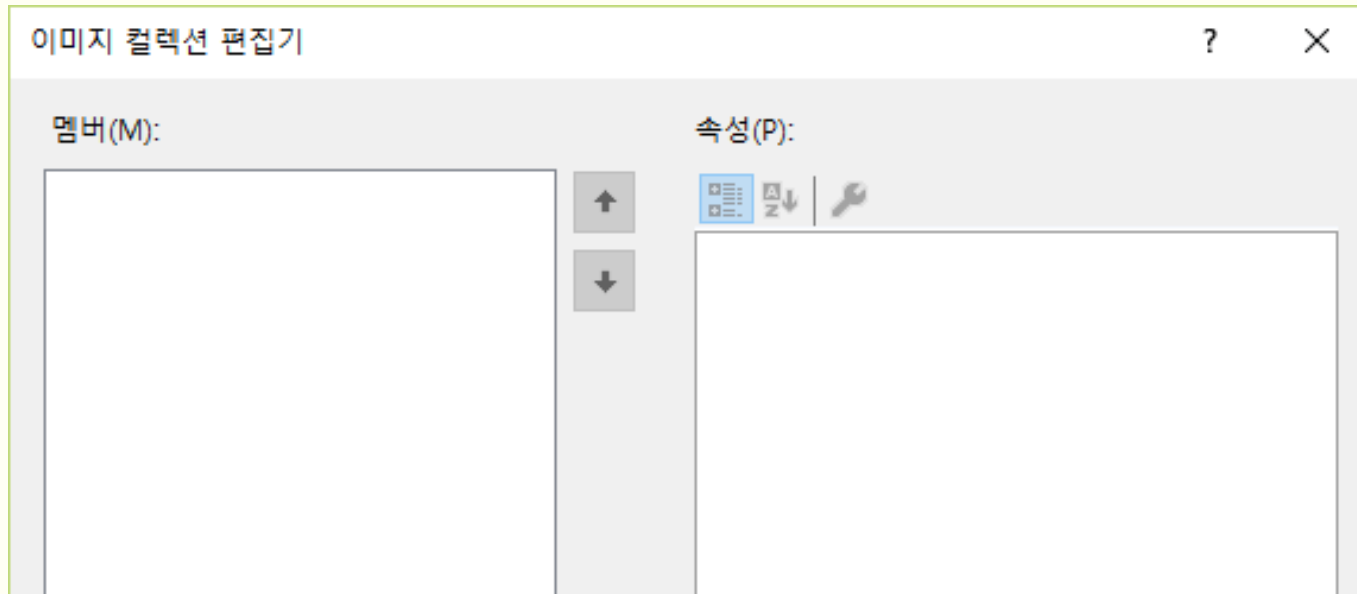


이미지 리스트

ImageList

ImageList

- 용도
여러개의 이미지를 배열과 같이 사용
- 이미지 등록
 - Images 속성에 컬렉션을 통해 등록
 - 이미지 컬렉션 편집기 사용



ImageList

- 이미지 등록
 - Add(), Clear(), Remove(), RemoveAt() 사용
- 이미지 크기 설정
 - ImageSize 속성에서 크기 설정
- 이미지 출력 메서드
 - Draw(Graphics g, int x, int y, int index)

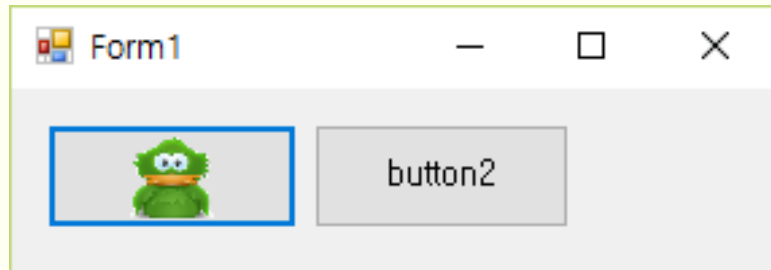
소스 보기

- 버튼에 이미지 출력

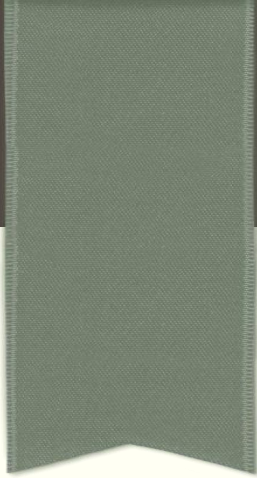
소스 보기

ImageList

- 버튼에 ImageList 연결하기



버튼 속성 → ImageList 설정
→ Images 속성(컬렉션)에서 이미지 선택
→ ImageIndex에서 출력할 ImageList의
인덱스 선택



탭 컨트롤

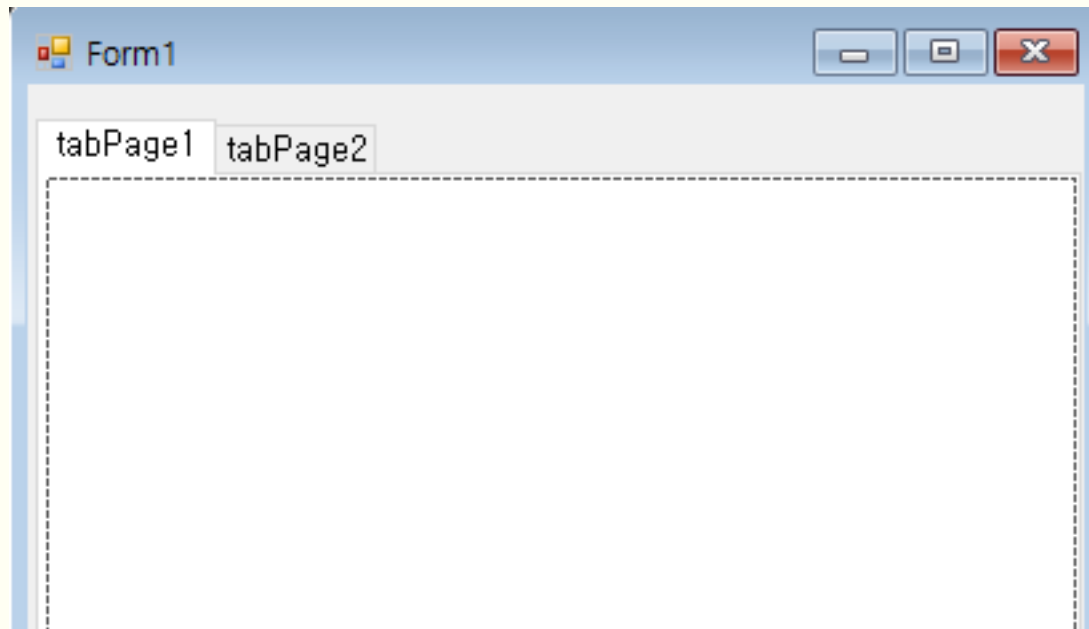
Tab Control

탭 컨트롤

- 용도
한 개의 폼 안에서 여러 개의 입력 형식 사용

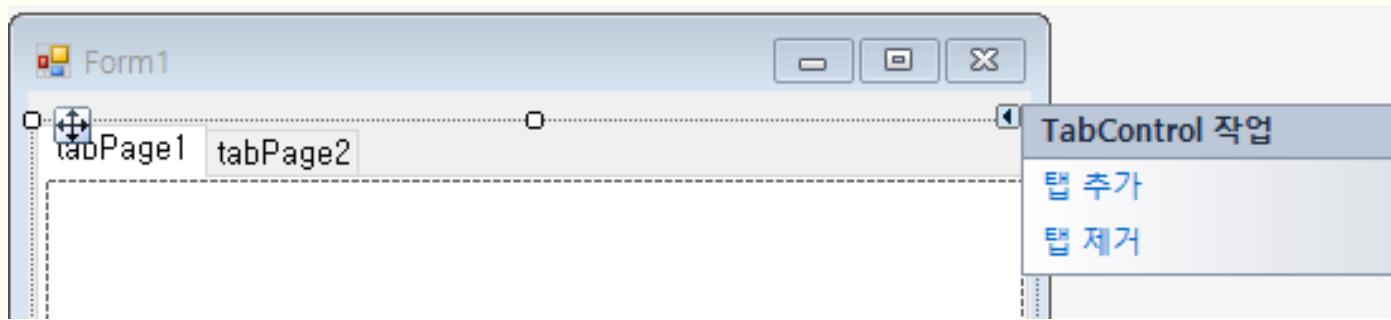


- 탭 페이지(tab page) 개념
한 폼에 포함된 구별된 형식



탭 컨트롤

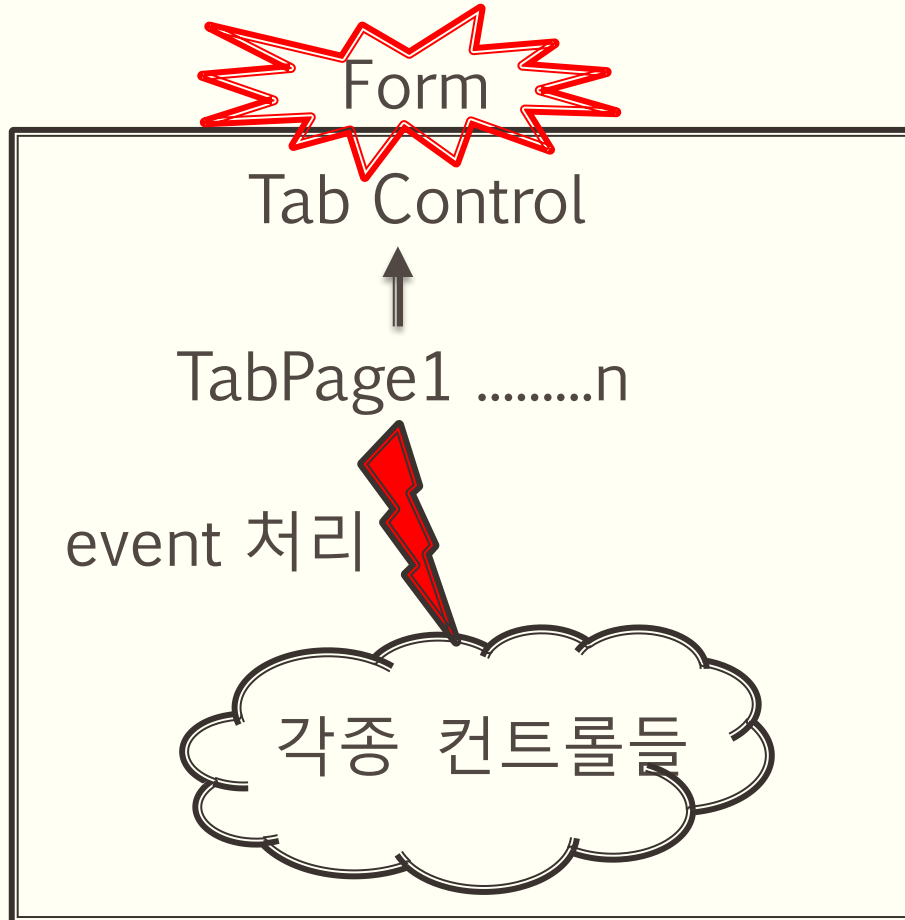
- 페이지 추가, 삭제
TabPages 속성 → 컬렉션 편집기
→ 페이지 추가



TabPages	(컬렉션)	...
TabStop	True	

탭 컨트롤

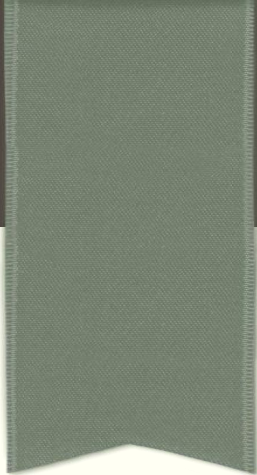
- 탭 컨트롤과 페이지 구조



탭 컨트롤

- 탭마다 메시지 박스 출력
탭과 페이지, 컨트롤과의 구조 확인

소스 보기

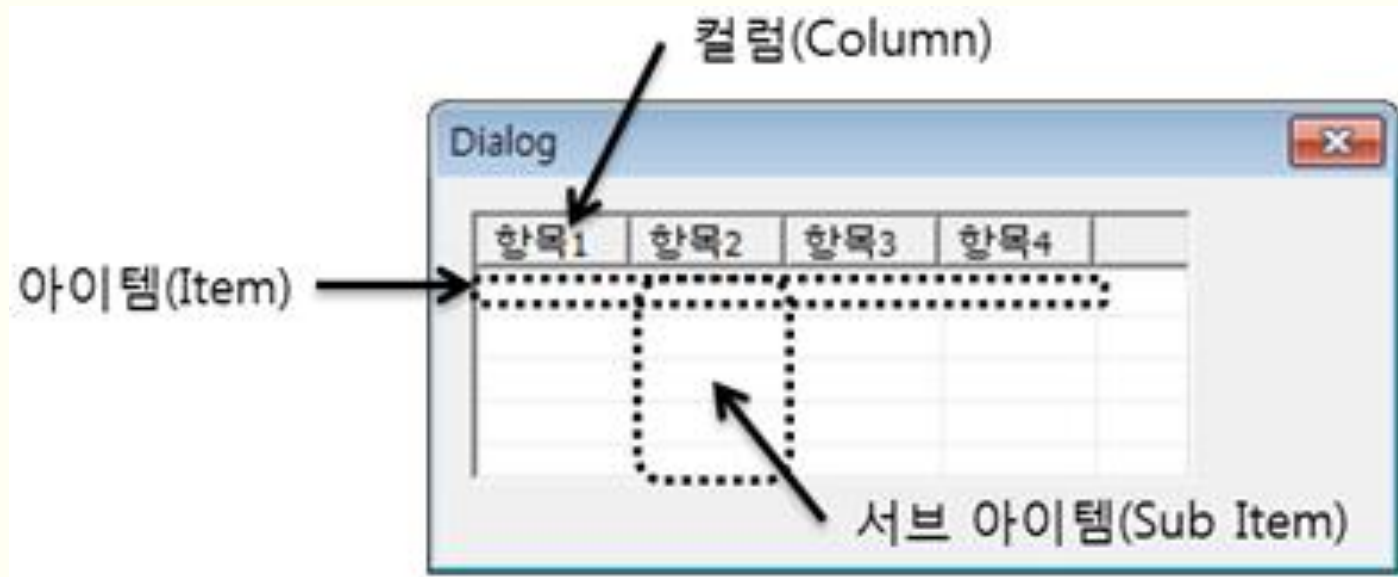


리스트 뷰

List View

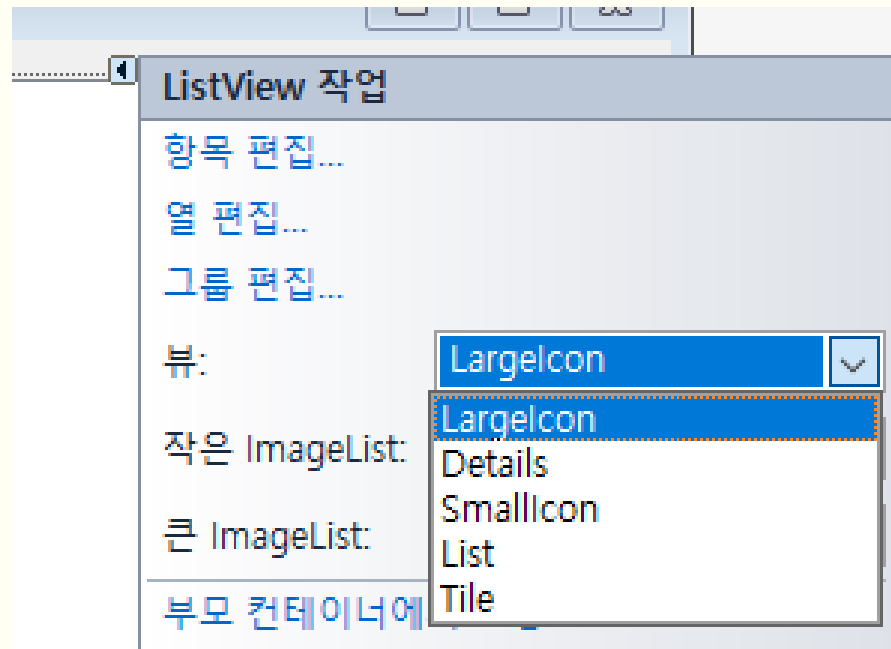
리스트 뷰

- 용도
데이터를 항목별로 구별하여 관리
- 기본 개념



리스트 뷰

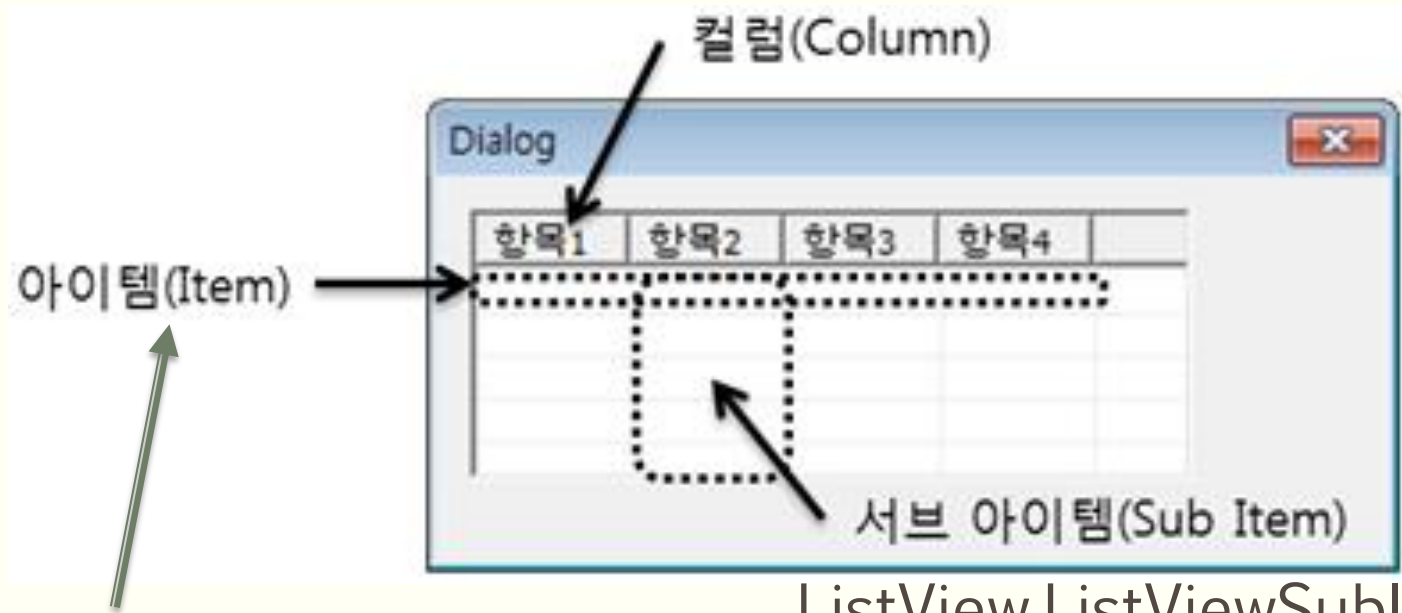
- 기능
 - 행별로 추가, 삭제, 삽입, 수정이 가능
 - 링크드 리스트 역할
- 리스트 형태
Icon, List, Tile 형태 등



리스트 뷰

- 리스트 뷰 관리 객체

ListView.ColumnHeaderCollection



ListView.ListViewSubItem

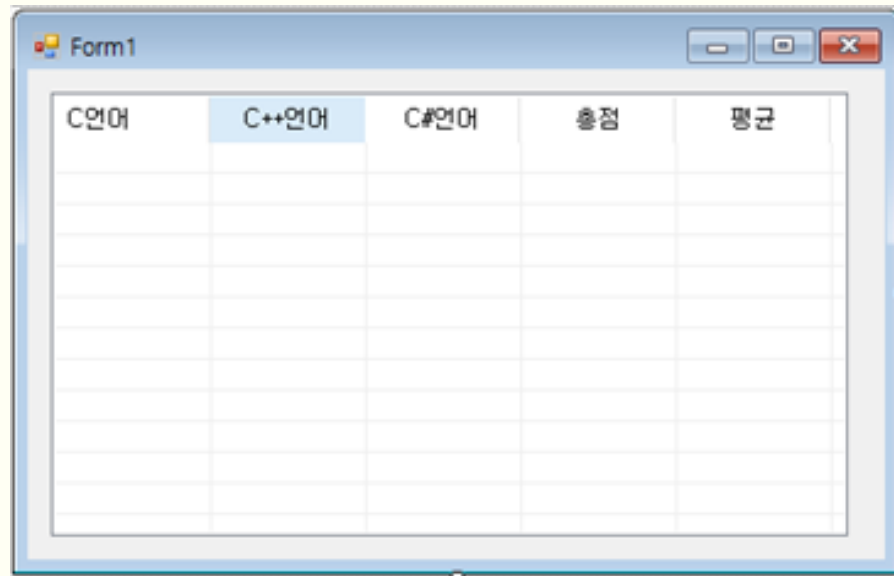
ListView.ListViewItemCollection

ListView.ColumnHeaderCollection → 고정적
그 외에는 유동적이어서 코드로 처리

리스트 뷰

■ 리스트 뷰 형태

ListView.FullRowSelect, ListView.GridLines





리스트 뷰 단계별로 다루기

1 단계 : 리스트 뷰 설정

- 리스트 뷰 스타일
Detail , FullRowSelect, GridLine

2 단계: 컬럼 정하기

- ColumnHeader 컬렉션 편집기로 컬럼 헤더 설정
C언어 , C++언어 , C# 언어, 총점, 평균

3 단계: Item 생성과 SubItem 입력

- Item 생성 되어야만 SubItem 입력 가능
 - 편집기로 입력하는 경우 ← 컬럼 헤더 입력
 - 코드로 추가/삭제/삽입/수정

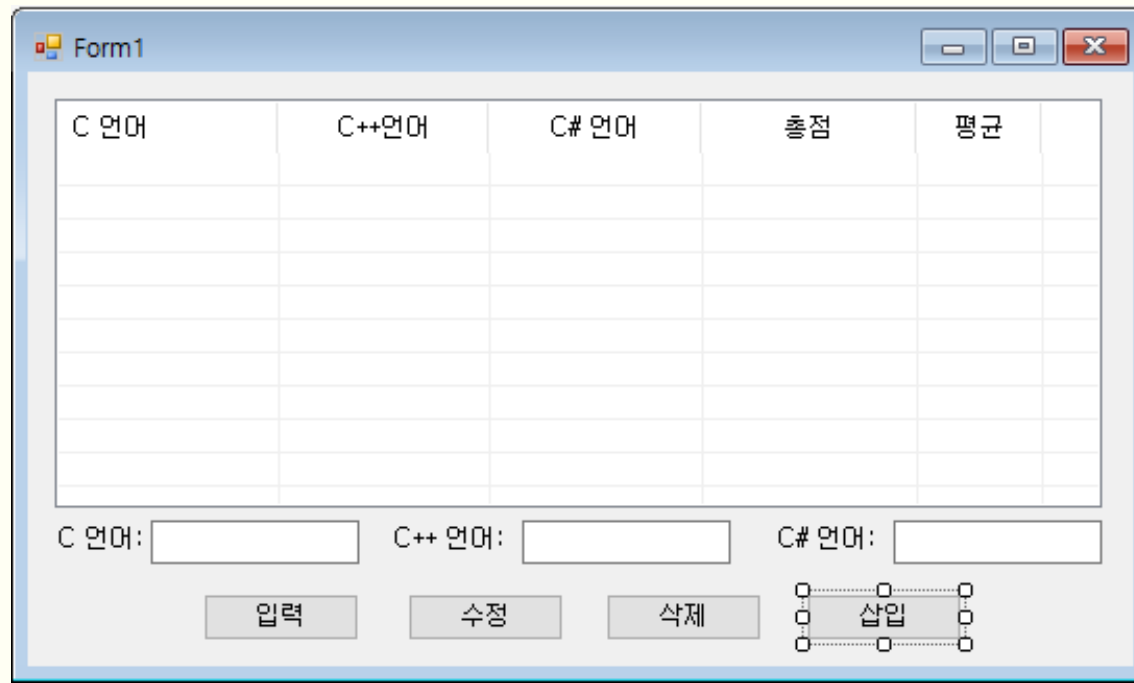
3 단계: Item 생성과 SubItem 입력 (코드)

- Item 생성 및 추가

`ListView.ListViewItemCollection Items { get; }`

`Items.Add()` → 오버로드 된 메서드
 `ListViewItem` 생성

→ `Add(ListViewItem)`



The screenshot shows a Windows application window titled "Form1". Inside the window is a table with 5 columns: "C 언어", "C++언어", "C# 언어", "총점", and "평균". The table has 10 rows, with the first row containing headers and the remaining 9 rows being empty. Below the table, there are three input fields labeled "C 언어:", "C++ 언어:", and "C# 언어:". To the right of these input fields are four buttons: "입력" (Input), "수정" (Modify), "삭제" (Delete), and "삽입" (Insert). The "삽입" button is highlighted with a dashed border.

3 단계: Item 생성과 SubItem 입력 (코드)

- 수정
선택된 아이템은 `ListView.SelectItems[0]`이며
`SubItem[0... n]` 을 부분적으로 선택
- 선택한 값을 수정하는 코드

소스 보기

3 단계: Item 생성과 SubItem 입력 (코드)

- 삭제
`ListView.Items.RemoveAt(0)` or
`ListView.SelectedItems[0].Remove()`
- 전체 삭제
`ListView.Items.Clear()`
- 선택한 항목을 삭제하는 코드

소스 보기

3 단계: Item 생성과 SubItem 입력 (코드)

- 삽입
`ListView.Items.Insert()`
- 삽입하고자 하는 인덱스 추출
`ListView.SelectedIndex[0]`
- 원하는 행에 아이템 삽입하는 코드

[소스 보기](#)