

TMemo^{메모} 컨트롤

여러 줄을 입력하는 텍스트박스를 이용하여 문장을 다룰 수 있게 됩니다.

TMemo^[티메모]는 일종의 텍스트박스인데 여러 줄의 글을 넣고 편집할 수 있다(한글로는 ‘메모’라고 적겠다).

프로그램이 실행되는 중에 *TMemo* 안에 내용을 만들어 넣으려면, *TMemo*의 프로퍼티와 메소드를 사용한다.

*TMemo*의 *Font*와 *Readonly* 프로퍼티는 *TEdit*에 있는 프로퍼티와 같은 것들이다. *TMemo*의 *Text* 프로퍼티에는 *TMemo*의 모든 글이 들어간다. 하지만 이 *Text* 프로퍼티는 애플리케이션이 실행되는 중에만 사용할 수 있고 개발 중에는 사용하지 못한다.

Lines^줄 프로퍼티

TMemo 컨트롤에 글을 입력하면 *Lines* 프로퍼티 즉, 인덱스번호가 매겨진 줄들의 집합에 저장된다. 첫 번째 줄의 번호는 0이고, 두 번째 줄의 번호는 1이다.

프로그램에서 TMemo의 Lines를 사용하는 예문

```
var
  s: String;
begin
  s:= memEx.Lines[2]; //3번째 줄의 값을 변수 s에 할당
  memEx.Lines[3]:='안녕'; //문자열 상수 '안녕'을 4번째 줄에 할당
  ...
end;
```

개발 중에는 오브젝트 인스펙터를 사용하여 TMemo에 Lines를 만들어 넣을 수 있다. 프로그램 실행 중에는 키보드로 글을 입력하고 엔터 키를 누를 때 마다 줄이 추가된다. 코드에 서는 `memEx.Lines.Add('입력 내용');`을 사용한다.

TMemo에 내용을 넣을 때에도 Lines 프로퍼티가 사용된다. 오브젝트 인스펙터에서 Lines 프로퍼티 안에 있는 작은 버튼을 누르면 Lines 편집기 창이 열린다. 이 창에 입력할 줄들을 적어 넣으면 된다. 줄을 바꾸려면 <엔터> 키를 누른다.

다 적어 넣고 나면 <OK> 버튼을 누른다.

Count^{개수} 프로퍼티

Count 프로퍼티는 전체 줄 수를 반환한다.

```
k:=memEx.Lines.Count;
```

Count는 읽기 전용이다. 즉 Count를 변경/편집 할 수는 없다.

WordWrap^{자동 줄 바꿈} 프로퍼티

WordWrap이 True이면, TMemo 컨트롤 안에 있는 글이 오른쪽 끝에 도달하면 자동으로 다음 줄로 넘어간다. 편집 공간의 크기에 맞추어 입력된다.

MaxLength^{최대 길이} 프로퍼티

MaxLength는 이 컨트롤에 입력될 수 있는 최대 문자 수를 지정한다. 만약 MaxLength가 0이면, 그 TMemο는 입력되는 문자 수 제한이 없다.

ScrollBars^{스크롤바} 프로퍼티

ScrollBars 프로퍼티는 TMemο 컨트롤에 스크롤바를 넣을 것인가를 지정한다.

Alignment^{정렬} 프로퍼티

Alignment^[알라인먼트] 정렬 프로퍼티는 입력 화면 안에 들어 있는 글의 줄맞춤 형식을 지정한다.



Tip | TMemο에는 Alignment와 이름이 비슷한 Align이라는 프로퍼티도 있으니 주의하세요. Align은 컨트롤의 위치를 정렬하는 프로퍼티이므로 콘트를 대부분이 자신의 위치를 지정하기 위해 가지고 있습니다. 하지만, Alignment는 글을 표시하는 콘트에만 있습니다.

메소드^{Method}들

메소드 몇 개를 살펴보자.

Delete^{삭제} 메소드

명시된 번호의 줄을 삭제한다. 나머지 줄들은 자동으로 옮겨지고 번호는 다시 매겨진다.

```
memEx.Lines.Delete(0); //첫 번째 줄 (번호가 0인 줄) 삭제
```

줄수 줄만 남겨두는 예문

```
k:= memEx.Lines.Count;
for i:=k-1 Downto 1 Do
  if (i mod 2) <>0 then
    memEx.Lines.Delete(i);
```

Exchange^{교환} 메소드

Exchange^{교환}는 명시된 번호의 줄 2개를 맞바꾼다.

```
memEx.Lines.Exchange(0,1); //0번과 1번 줄을 맞바꾼다.
```

Move^{이동} 메소드

Move^{이동}는 줄을 다른 줄로 이동한다.

예문

```
memEx.Lines.Move(1,5);
```

이 메소드는 텍스트 중 1번 줄(즉, 두 번째 줄. 번호가 0부터 시작하기 때문)이 제거되고, 뒤에 있던 줄들이 한 줄씩 위로 이동한다. 그리고 나서 제거되었던 1번 줄은 5번 줄로 들어간다.

실습

Exercise 1.

TMemo를 채운다. 버튼을 클릭하면 TMemo에 아래 내용만 남도록 만들어 보자.

- 네 줄이 넘는다면, 앞에 네 줄만 남긴다.
- 네 줄이 되지 않는다면 첫 번째 줄만 남긴다.

Exercise 2.

TMemo를 채운다. 그리고 나서 다시 텍스트를 편집해본다.

버튼을 클릭하면 레이블에 결과가 표시되는 프로그램을 만들어보자.

- a) TMemo에서 현재 편집 중인 (즉, 커서가 위치한) 줄의 번호를 표시한다. 만약 편집 중인 줄이 없으면, '현재 편집 중인 줄이 없습니다' 라는 메시지를 표시한다.
- b) 편집되는 과정에서 입력된 내용이 있는 줄들의 번호들을 모두 표시한다. 해당되는 줄이 없으면 없다는 메시지를 표시한다.

Exercise 3.

TMemo에 몇 줄만 채운다.

그 중 한 줄에는 '강아지'라는 단어를 넣고, 다른 한 줄에는 '고양이'라는 단어를 꼭 넣는다. 이 두 단어를 맞바꾸는 프로그램을 만들어 보자.

Exercise 4.

TMemo를 채운다.

가장 내용이 긴 줄을 찾아서 가장 위로 옮기는 프로그램을 만들어보자.

Exercise 5.

TMemo에 한 줄 당 숫자 하나씩을 넣는다.

입력된 숫자들 중에서 짝수들은 모두 두 배로 만들어서 두 번째 TMemo 컨트롤로 옮기고, 홀수들은 모두 2로 나누어서 세 번째 TMemo 컨트롤로 옮기도록 프로그램을 작성한다.