

사용자 인터페이스(I)



시작하면서



- ◎ 목차

- 위젯
- 레이아웃

위젯



◎ 뷰(View)

- 사용자에게 정보를 표시하거나 사용자로부터 입력을 받는 기본 클래스
- 종류: 텍스트 뷰, 편집텍스트, 이미지 뷰, 누름 버튼, 체크박스, 날짜 선택, 진행 바, 줌 슬라이더 등
- <표 5-1, 대표적인 뷰 종류>

◎ 액티비티

- 사용자와 상호작용하는 주체
- 휴대 단말의 화면 관리
- 하위 컴포넌트들인 뷰, 뷰 그룹, 대화창으로 구성

◎ 위젯

- 뷰가 화면에 나타난 이미지 형태

◎ 뷰그룹(ViewGroup)

- 화면에 다양한 뷰들을 배치하는 클래스
- Layout 클래스가 대표적



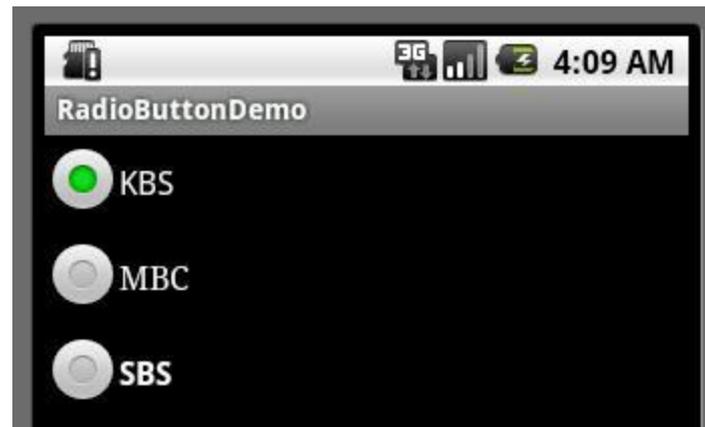
- 한 목록에서 여러 개 선택시 사용
 - 체크 버튼과 텍스트로 구성
 - 설정/해제에 따른 상태 변화는 운영체제 처리
 - 개발자는 체크박스의 상태 변경에 따른 응용프로그램의 반응만 처리
- ◎ **<실습 5-1>**
- CheckBoxDemo 프로젝트 생성
 - <코드 5-1> main.xml
 - <코드 5-2> checkBoxDemo.java
 - [실행결과]



라디오 버튼



- 여러 항목 중에서 하나만 선택할 때 유용
 - 라디오 버튼과 텍스트로 구성
 - 라디오 버튼들을 라디오 그룹으로 묶어 사용
 - 하나의 라디오 버튼을 선택만 선택
- ◎ <실습 5-2>
- RadioButtonDemo 프로젝트 생성
 - <코드 5-3> main.xml
 - [실행결과]

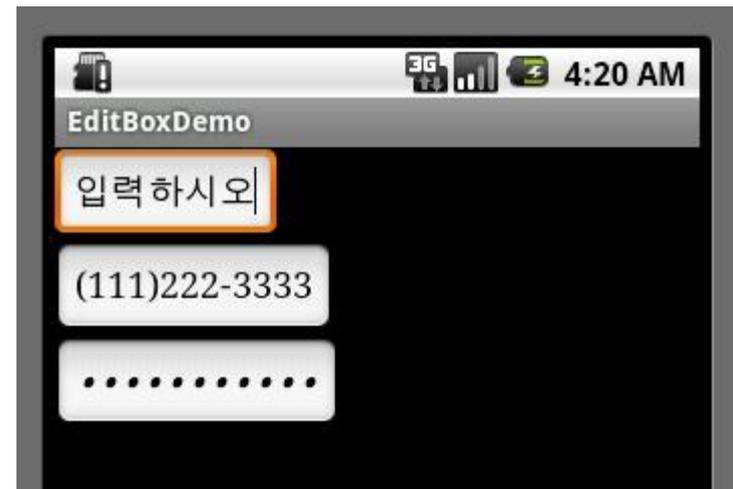




- 사용자로부터 텍스트 입력시 사용
- 입력과 관련된 여러 기능 활성화
 - hint 특성: 미리 설정한 텍스트를 제시하고 사용자가 입력 시작시 내용이 사라지는 기능
 - phoneNumber 특성: 사용자의 입력 범위를 숫자로 제한
 - password 특성: 패스워드 입력 시 입력된 문자를 감춤

◎ <실습 5-3>

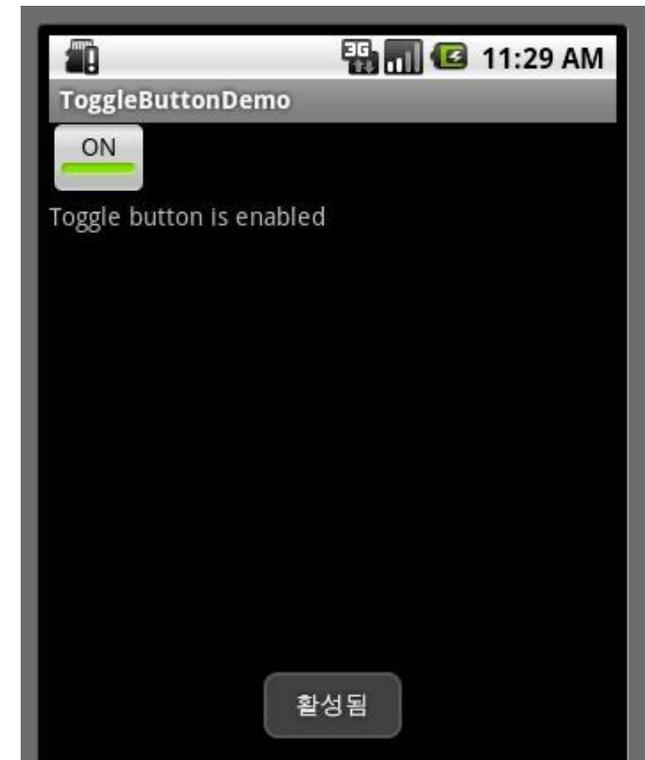
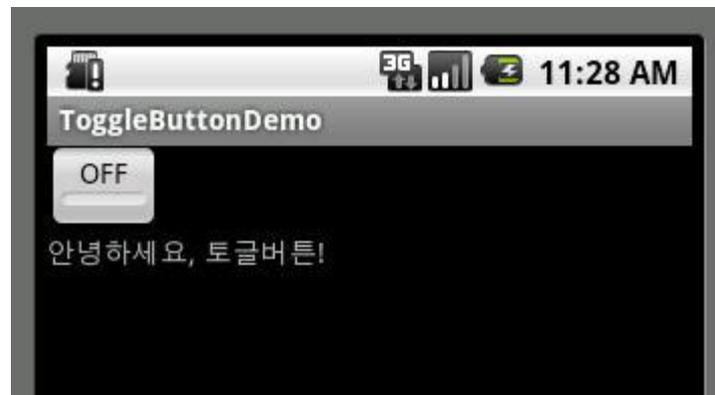
- EditTextDemo 프로젝트 생성
- <코드 5-4> main.xml
- [실행결과]



토글 버튼



- 어떤 대상의 켜짐 또는 꺼짐 상태를 표시
- 텍스트가 버튼 안에 표시
- 리소스 파일에서 켜진 상태와 꺼진 상태를 설정
 - 텍스트 설정 속성: textOn, textOff
 - phoneNumber 특성: 사용자의 입력 범위를 숫자로 제한
 - password 특성: 패스워드 입력 시 입력된 문자를 감춤
- ◎ <실습 5-4>
 - ToggleButtonDemo 프로젝트 생성
 - <코드 5-5> main.xml
 - <코드 5-6> ToggleButtonDemo.java
 - [실행결과]



레이아웃

레이아웃



◎ 뷰 그룹

- 다른 위젯들을 담는 컨테이너
 - 부모 뷰: 위젯들을 담는 그릇 역할을 하는 ViewGroup 객체
 - 자식 뷰: ViewGroup 안에 포함된 위젯들
 - 최상단 꼭대기에 존재
- 뷰의 계층 구조: [그림 5-6]

◎ 화면 레이아웃

- ViewGroup과 LayoutParams 객체들로 구성
- 모든 뷰는 배경색, 색깔 등 속성 명시
- LayoutParams은 영역 정보, 상대 위치, 절대 위치, 여백 속성 등 정의

예) <LinearLayout> 요소를 먼저 정의한 후, 그 안에 <TextView> 요소들을 포함

레이아웃



◎ 레이아웃 개념

- View의 일종으로서 자식들의 화면 배치를 처리
- 화면에 무엇인가를 나타내는 것은 아님
 - 레이아웃 객체들은 사용자와 직접 상호작용하지 않음
- 선형 레이아웃, 상대 레이아웃, 절대 레이아웃, 프레임 레이아웃, 표 레이아웃 등

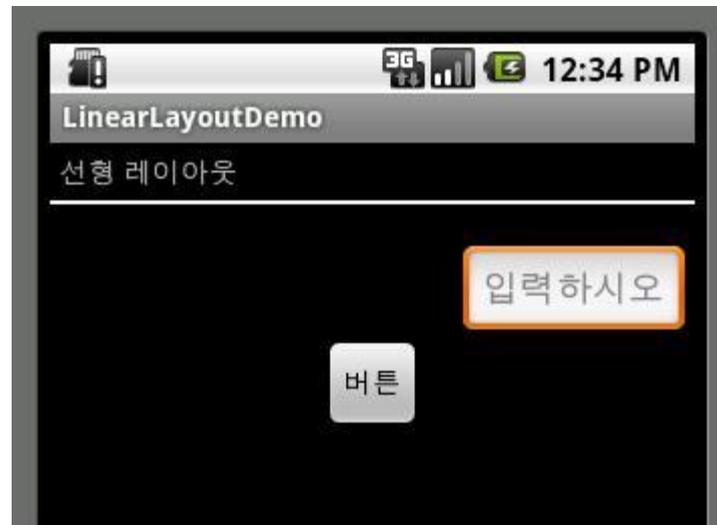
◎ 뷰그룹의 공통 특성

- layout_width, layout_height 속성
- layout_margin 속성: 상하좌우 방향의 여백 크기 지정
- layout_marginTop 속성: 위젯 위쪽의 여백 크기 설정
- 뷰 그룹의 공통특성: [표 5-3]
- 레이아웃 그래비티 속성: [표 5-4]

선형 레이아웃



- 위젯, 하위 컨테이너들을 가로나 세로 방향으로 나열
 - 방향, 채우기, 가중치, 그라비티, 패딩 속성 등 활용
- ◎ <실습 5-5>
- LinearLayoutDemo 프로젝트 생성
 - <코드 5-7> main.xml 복사
 - [실행결과]



절대 레이아웃



- 자식 뷰들의 위치를 x, y 좌표로 정확하게 표기
 - 위젯들을 픽셀 수준에서 정밀하게 조정 가능
 - 화면 크기가 다른 기기들간에 레이아웃이 깨질 수 있음
- ◎ <실습 5-6>
- AbsoluteLayout 프로젝트 생성
 - <코드 5-8> main.xml 복사
 - [실행결과]





표 레이아웃

◎ 실습 5-6 표 레이아웃(p.178)

- 자식 뷰들을 표 형태로 배치
- 행이나 열의 개수를 지정, 특정 열의 폭을 조정
- TableRow 클래스: 개별 자식 뷰들을 가로로 배치
 - 수평 방향의 선형 레이아웃에 해당
 - 표 레이아웃: TableRow들의 수직 방향 나열
- 표 레이아웃 속성: [표 5-5]

◎ <실습 5-7>

- TableLayout 프로젝트 생성
 - 실습할 표 레이아웃: [표 5-6]
 - <코드 5-9> main.xml 복사
- [실행결과]



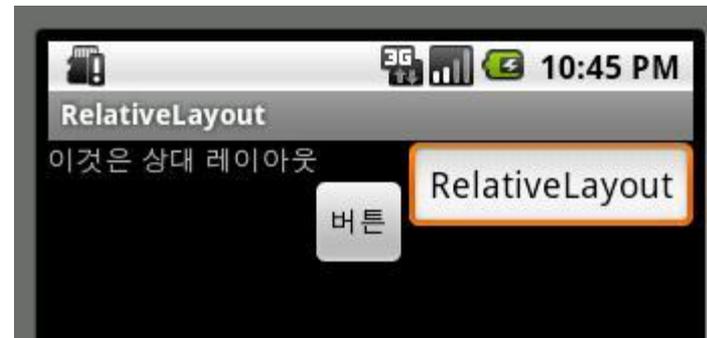
상대 레이아웃



- 자식 뷰들을 상대적인 관계에 따라 배치
예) 자식 A는 자식 B의 옆에, 자식 C는 자식 B의 아래에 두라
- 특성: `android.widget.RelativeLayout.LayoutParams`에서 정의
- 부모 뷰 기준으로 상대 배치: [표 5-7]
- 특정 뷰 기준으로 상대 배치: [표 5-8]
- 특정 뷰 기준으로 정렬 속성: [표 5-9]

◎ <실습 5-8>

- RelativeLayout 프로젝트 생성
 - <코드 5-10> main.xml 복사
 - [실행결과]





프레임 레이아웃

- 자식 뷰들을 화면의 왼쪽 상단을 기준으로 배치
- 새로운 자식 뷰들은 이전의 자식 뷰 위에 중첩
- 마지막으로 정의된 자식 뷰가 맨 위에 놓여 화면에 표시
- 프레임 레이아웃 속성: [표 5-10]

◎ <실습 5-9>

- FrameLayout 프로젝트 생성
 - <코드 5-11> main.xml 복사
 - [실행결과]

