

입력 처리

코틀린 안드로이드 개발 기초 강의

강사: 배정만



목차

- ① 입력 컨트롤 개요
- ② 뷰의 초점(focus)
- ③ 자유형(Freeform) 텍스트 및 숫자
- ④ 선택 제공

입력 컨트롤 개요

사용자 입력 수락

- 자유형 텍스트 및 숫자: EditText (키보드 사용)
- 선택 제공: 체크박스, 라디오버튼, 스피너
- 시위치 켜기/끄기(on/off): 토글 스위치
- 범위에서 값 선택: Seekbar

입력 컨트롤 예시

1. EditText

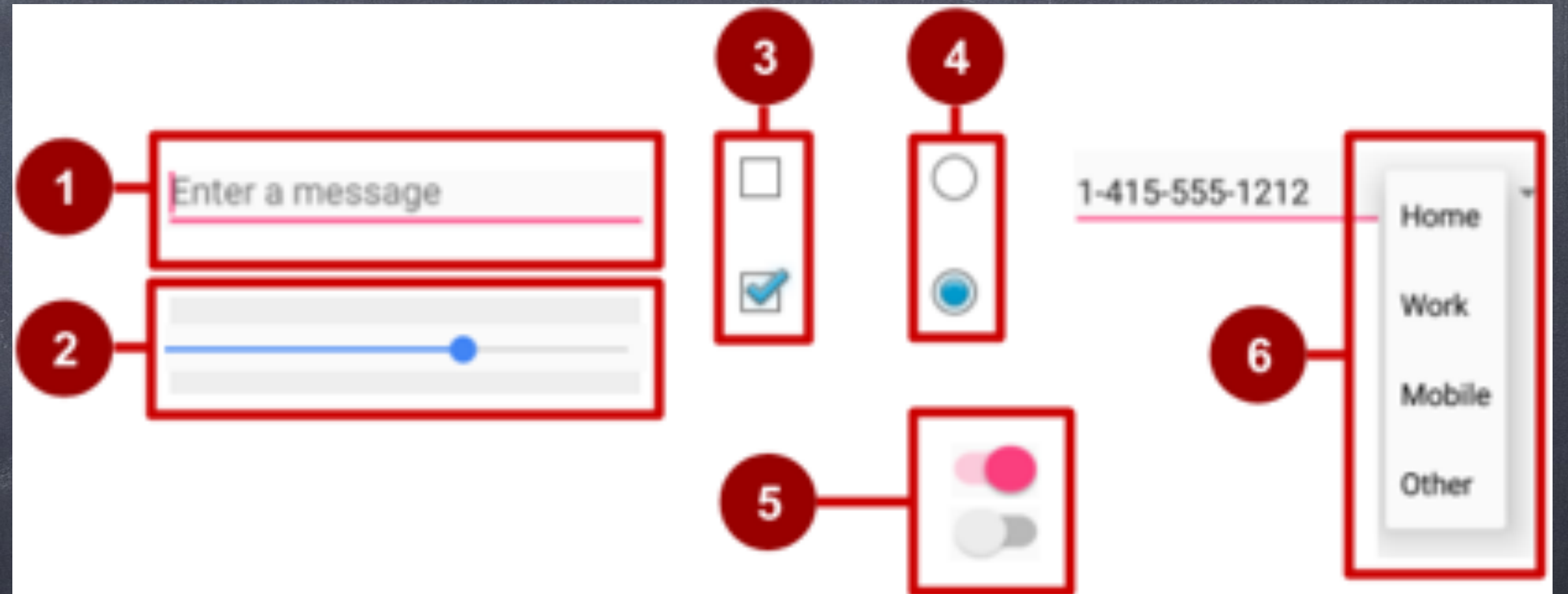
2. SeekBar

3. CheckBox

4. RadioButton

5. Switch

6. Spinner



입력 컨트롤의 동작 방식

1. EditText - 키보드를 이용하여 텍스트 입력
2. SeekBar - 좌/우로 밀어서 값 설정
3. Checkbox - 한개 이상의 요소를 선택
4. RadioButton - RadioGroup 내에서 단 한개의 선택
5. Switch - 두드려서(tab) 켜거나 끄기
6. Spinner - 회전하는 목록 요소에서 하나를 선택

입력 컨트롤의 기본 클래스 - 뷰

- 뷰 클래스는 입력 컨트롤을 포함하여 모든 UI 구성요소에 대한 기본 코드 뭉치
- 뷰 클래스는 상호작용하는 UI 요소의 기본 클래스
- 뷰는 `android:onClick` 을 통해 상호작용을 제공

뷰의 초점
View focus

초점 (Focus)

- ◉ 사용자 입력을 받는 뷰에는 초점(Focus) 기능이 있음
- ◉ 한 번에 한 개의 뷰만 초점을 가질 수 있음
- ◉ 초점은 어떤 뷰가 입력을 받는지 명확하게 만듦
- ◉ 초점은 다음으로 부터 할당됨
 - ◉ 유저의 뷰 두드림(tap)
 - ◉ 리턴, 두드림(tab), 방향키로 다음 입력 컨트롤로 이동
- ◉ `requestFocus()` 함수로 초점 가능한 뷰에 임의로 초점을 호출할 수 있음

클릭 가능 vs 초침 가능

- 클릭 가능(Clickable) - 뷰는 클릭이나 두드림(tapped) 에 응답할 수 있음
- 초침가능(Focusable) - 뷰는 입력을 받기 위해 초점을 얻을 수 있음

키보드 같은 입력 컨트롤은 초점을 가진 뷰에 입력을 전달함

다음 초점을 갖는 부

- 터치 아래의 맨 위 부
- 사용자가 입력을 제출한 후에 초점이 가장 가까운 위치로 이동 - 우선순위는 왼쪽에서 오른쪽, 위에서 아래
- 초점은 사용자의 방향키 입력으로 변경 가능

사용자 안내

Guiding users

- 초점이 있는 뷰를 시작적으로 표시하여 사용자가 입력한 위치를 알 수 있도록 함
- 초점을 가질 수 있는 뷰를 시작적으로 표시하여 사용자가 흐름을 확인할 수 있도록 도움
- 예측 가능하고 논리적 - 당황시키지 않음

초점 안내

- ◉ 레이아웃의 입력 컨트롤을 왼쪽에서 오른쪽으로, 위에서 아래로 정렬하여 초점이 지정되도록 함
- ◉ 레이아웃의 뷰 그룹 내에 입력 컨트롤을 위치
- ◉ 순서를 XML 로 정의 가능

```
android:id="@+id/top"
```

```
android:focusable="true"
```

```
android:nextFocusDown="@+id/bottom"
```

명시적으로 초점 지정

- 뷰의 함수를 이용해 초점 지정
 - `setFocusable()` - 뷰에 초점이 가능한지 설정
 - `requestFocus()` - 특정 뷰에 초점 지정
 - `setOnFocusChangeListener()` - 초점이 지정되거나 잃을 때에 대한 리스너 설정
 - `onFocusChanged()` - 뷰의 초점이 변경될 때 호출

초점이 있는 뷰 찾기

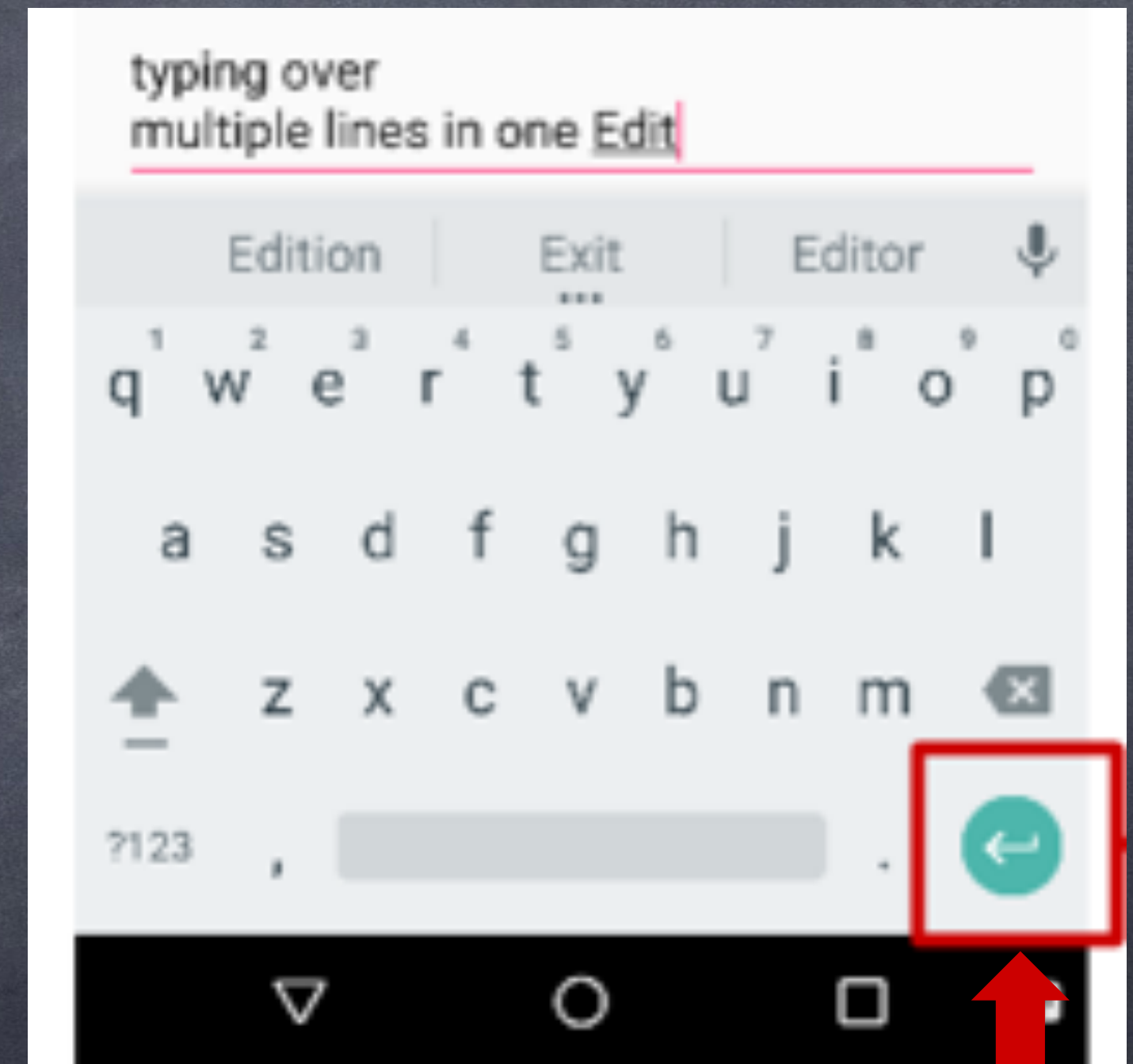
- `Activity.getCurrentFocus()`
- `ViewGroup.getFocusedChild()`

자유형 텍스트 및 숫자

Freeform text and numbers

여러줄 텍스트를 위한 EditText

- EditText 기본 값
- 영숫자 키보드
- 제안(suggestions) 나타남
- Return (Enter) 키를 눌러 새 줄 시작



Return key

사용자 지정 입력 유형

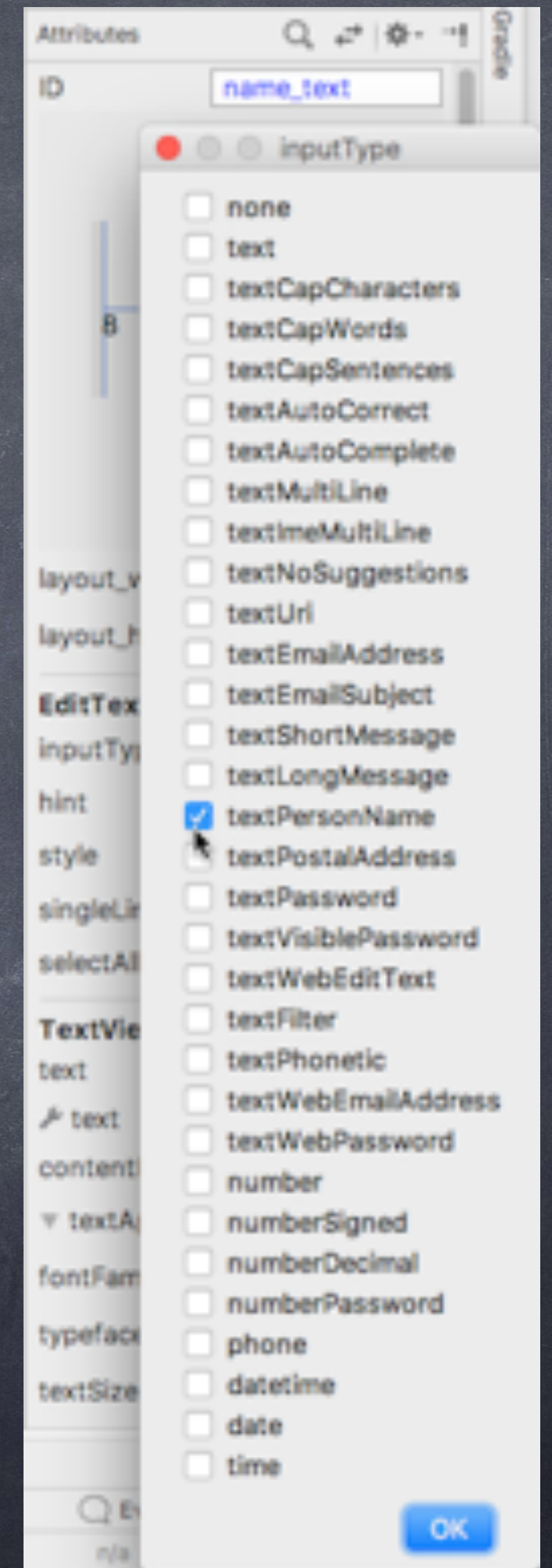
- 레이아웃 편집기의 특성 화면에서 설정
- EditText 용 XML:

<EditText

```
android:id="@+id/name_field"
```

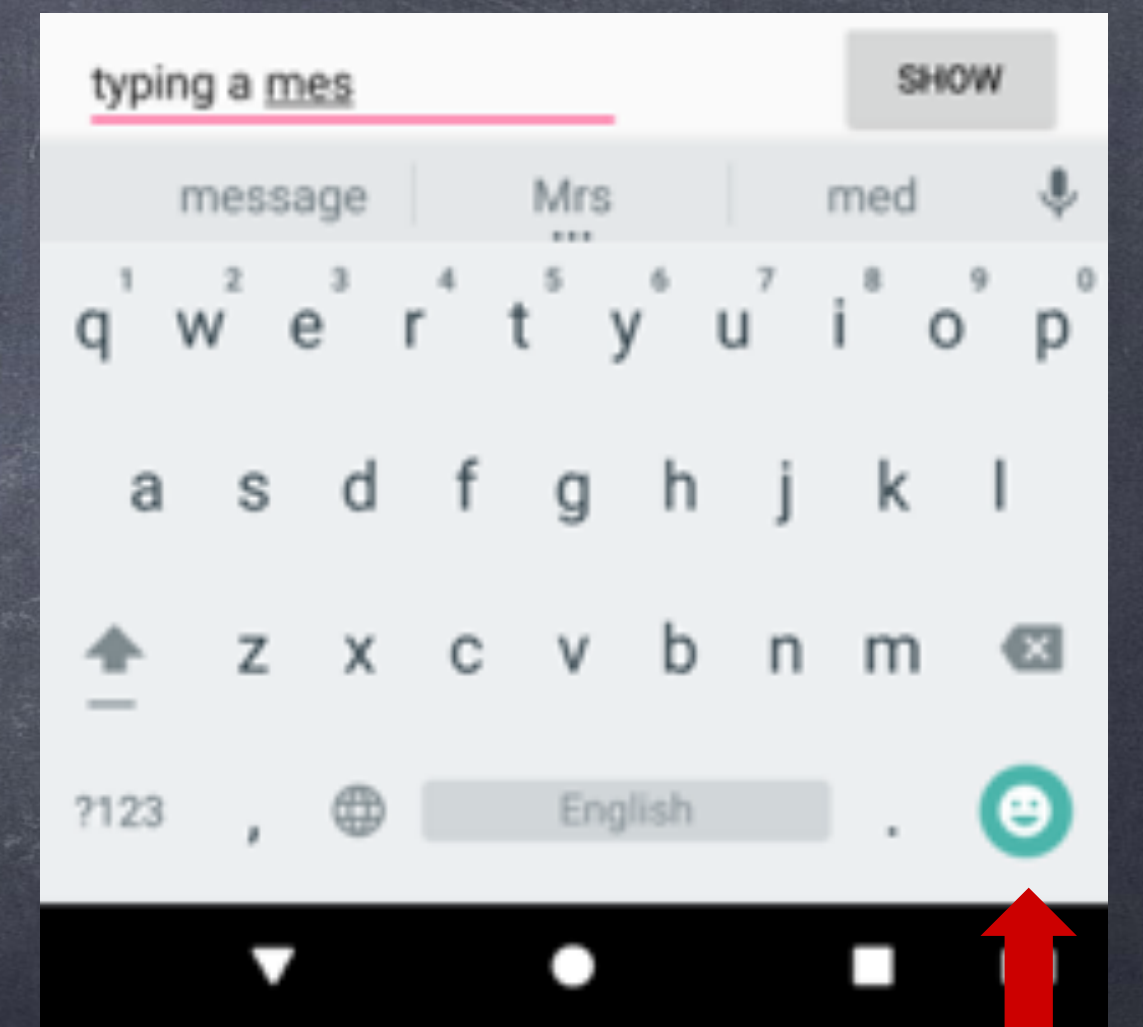
```
android:inputType =  
    "textPersonName"
```

...



메세지

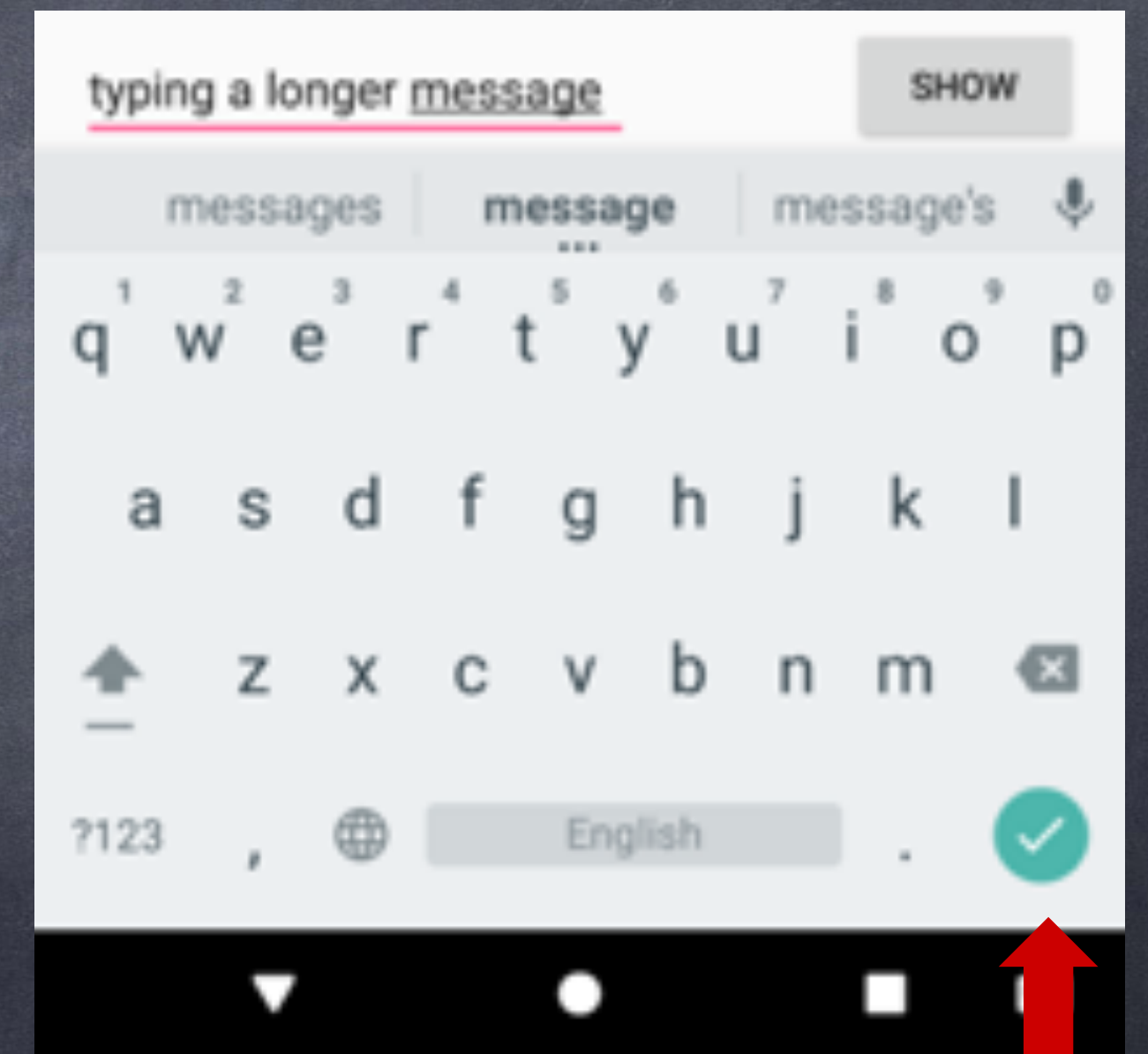
- `android:inputType='textShortMessage'`
- 한 줄 텍스트
- 이모티콘 키를 누르면 이모티콘 입력모드로 바뀜



Emoticons

한 줄 텍스트

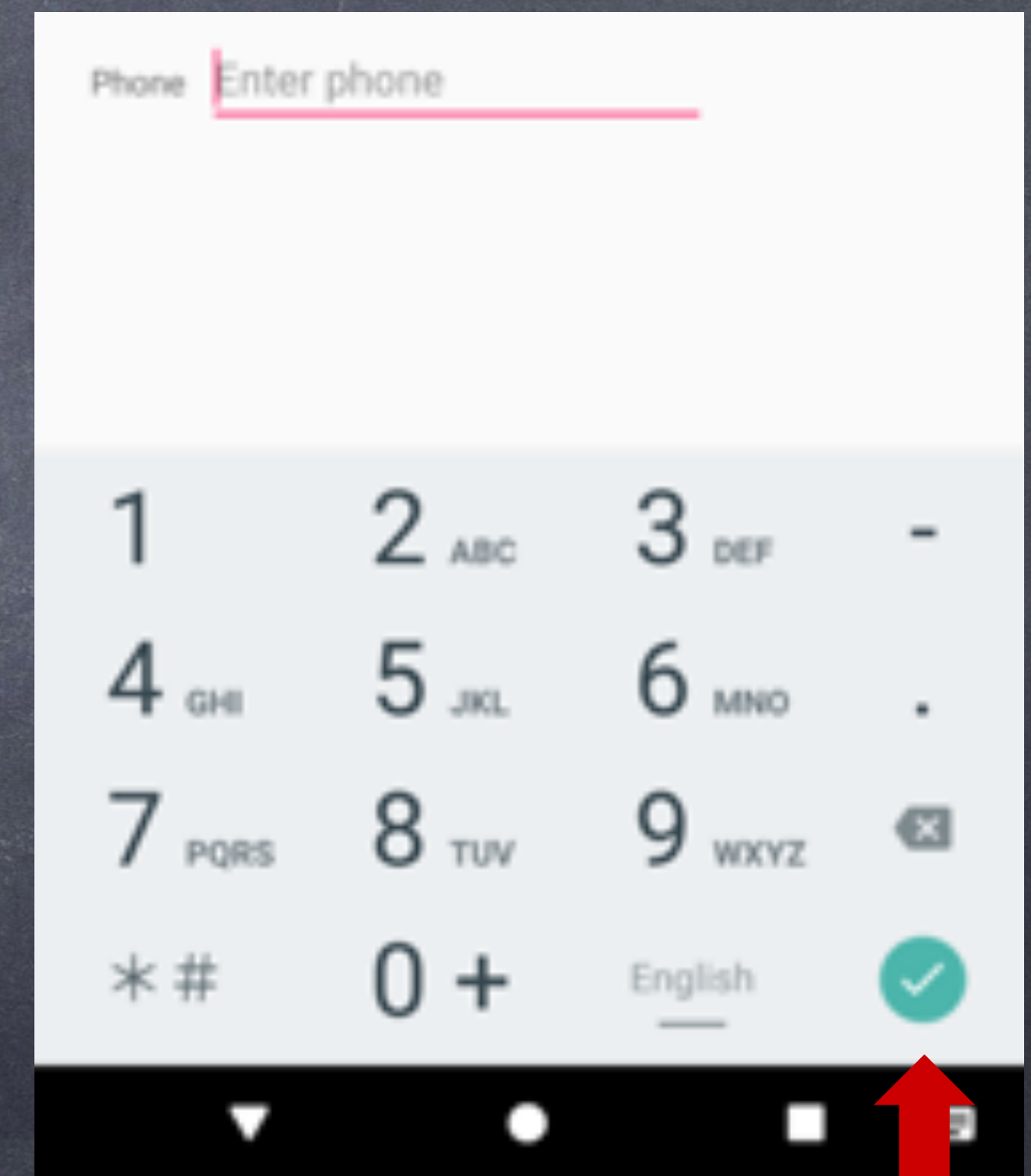
- 'textLongMessage', 'textPersonName'
둘 다 한 줄 텍스트 지원
- Done(완료)를 눌러 초점을 다음으로



Done key

숫자 키패드

- `android:inputType = "phone"`
- 숫자만 출력 됨
- Done(완료)를 눌러 초점을 다음으로



Done key

텍스트 가져오기

- EditText 객체 가져오기

```
val simpleEditText: EditText =  
findViewById(R.id.edit_simple)
```

- CharSequence 를 가져와 문자열(String) 으로 변경

```
val strValue = simpleEditText.text.toString()
```

공통 입력 유형

- ◉ `textCapCharacters` : 대문자로 설정
- ◉ `textCapSentences` : 대문자로 각 문장 시작
- ◉ `textPassword` : 입력 된 암호 숨기기
- ◉ `number` : 텍스트 입력을 숫자로 제한
- ◉ `textEmailAddress` : @와 함께 키보드 표시하기
- ◉ `phone` : 전화 키패드 표시
- ◉ `datetime` : 날짜와 시간을 입력하기위한 슬래시(/)와 콜론(:)이있는 숫자 키패드 표시

선택 제공

선택을 제공하기 위한 UI 요소

☉ 체크박스, 라디오 버튼

Chocolate Syrup
 Sprinkles
 Crushed Nuts

Choose a delivery method:
 Same day messenger service
 Next day ground delivery
 Pick up

☉ 토크 버튼, 스위치

Turn on or off: Turn on or off:

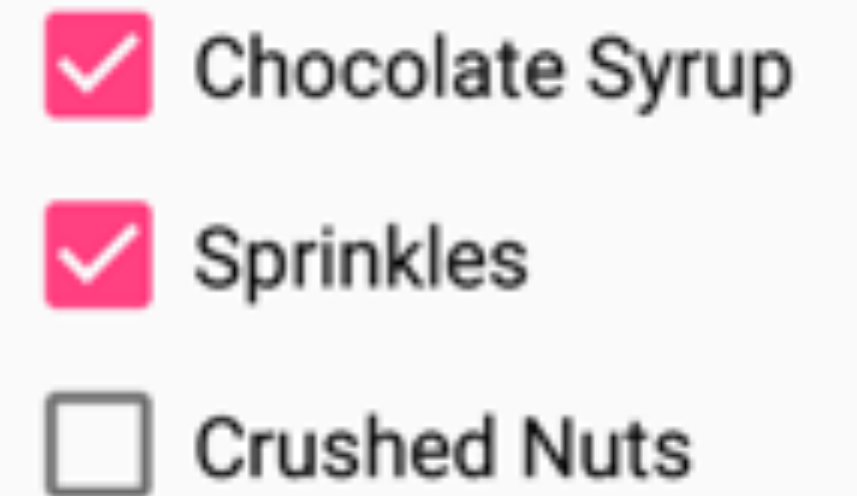
Turn on or off: Turn on or off:

☉ 스피너

1-415-555-1212
Home
Work
Mobile
Other

체크박스(CheckBox)

- 사용자는 원하는 수의 항목을 선택할 수 있음
- 하나의 선택이 다른 요소를 선택해제하지 않음
- 사용자는 세로목록의 체크박스를 기대함
- 일반적으로 제출(submit) 버튼과 함께 사용
- 모든 체크박스는 뷰 이고 onClick 함수를 가지고 있음



라디오버튼

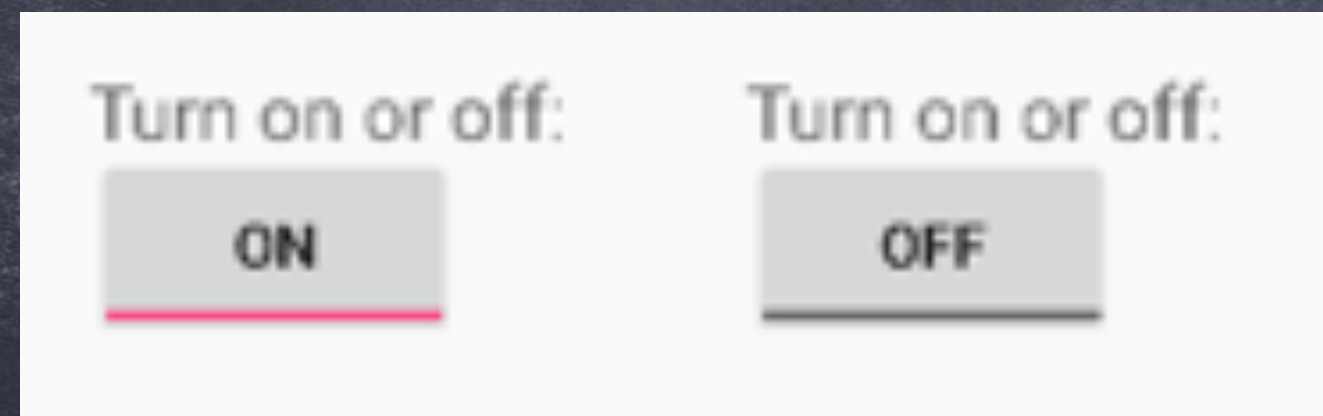
- RadioGroup의 RadioButton 요소를 세로 목록으로 (레이블이 짧은 경우 가로)
- 사용자는 선택 사항 중 하나만 선택할 수 있음
- 하나의 선택이 그룹에있는 다른 모든 선택을 취소
- 각 RadioButton에는 onClick 함수가 있음
- RadioGroup에 대한 제출 버튼과 함께 일반적으로 사용

Choose a delivery method:

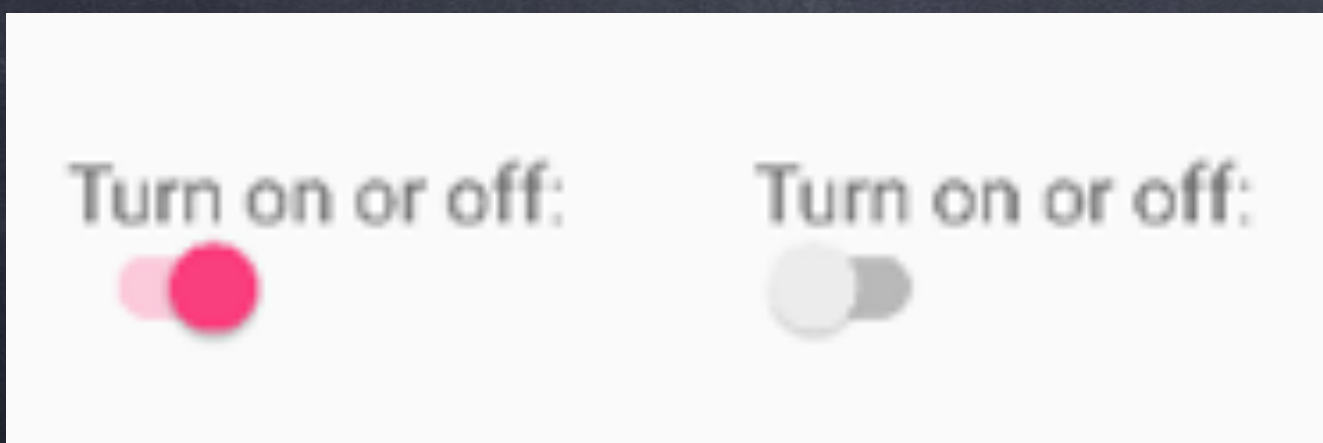
- Same day messenger service
- Next day ground delivery
- Pick up

토글 버튼과 스위치

- 사용자는 켜짐(on)과 꺼짐(off)을 전환할 수 있음
- `android:onClick` 을 사용할 수 있음



토글 버튼



스위치

Learn more

- [Input Controls](#)
- [Radio Buttons](#)
- [Specifying the Input Method Type](#)
- [Handling Keyboard Input](#)
- [Text Fields](#)
- [Spinners](#)