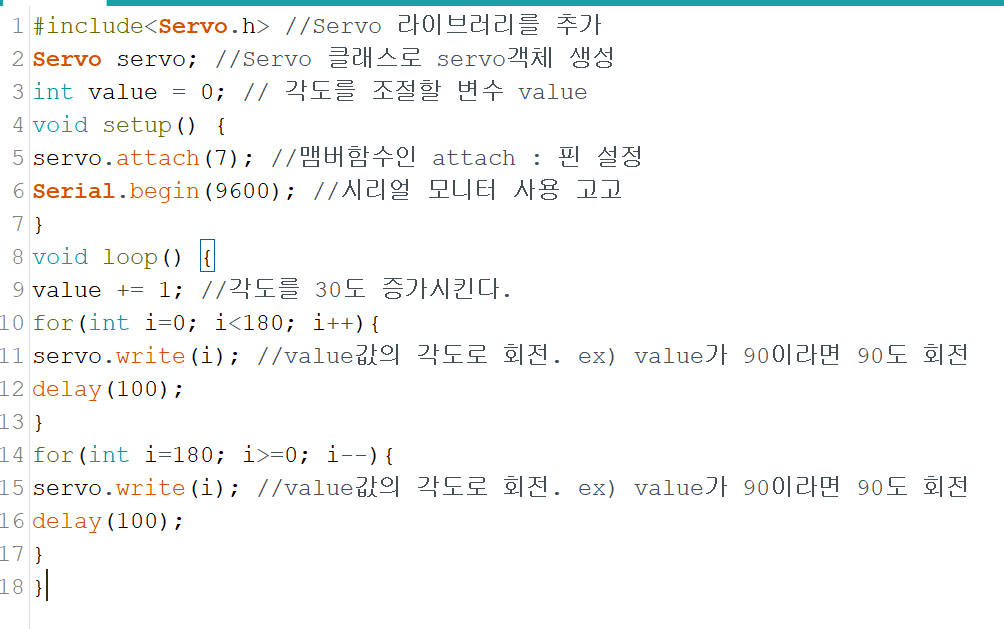
**서보 모터의 개요 (PPT2)**

* 일반 모터와 다르게 회전이 정해져 있다.
* 때문에 정확한 움직임이 필요한 경우에 사용하는 모터
* 사용자가 원하는 각도, 속도로 움직임이 가능
* 로봇, 장난감, 드론 CCTV 등 광범위하게 사용
* SG90 서보 모터 : 저렴하지만 0~180도까지만 회전이 가능하다는 단점

**서보 모터의 사용 방법 (PPT3)**

* 서보 모터의 펄스폭 1ms~2ms의 범위
* 1ms 일 때 0도, 2ms 일 때 180도(1.25ms - 45도, 1.5ms - 90도)

**서보 모터 동작 실습 (PPT6)**

****

**릴레이의 개요 및 특징**

**릴레이(Relay) 개요 (PPT7~8)**

* 도선에 전류가 흐르면 주변에 자기장이 형성
* 코일을 원통에 감게 되면 자기장이 중첩
* 코일의 중앙에 한 방향으로 작용하는 자기장이 형성
* 코일안에 철심을 넣으면 자기장이 강해지면서 자석과 같은 힘이 발생
* 이와 같은 원리를 이용하여 만든 스위치를 릴레이라고 함
* 전류가 흐르면 자력이 발생하여 스위치가 닫힌 상태
* 전류가 흐르지 않을 때에는 자력이 없어져 스위치가 열린 상태

**릴레이(Relay) 특징 (PPT9)**

* 전기적으로 독립된 회로를 연동시켜 제어할 수 있다는 장점
* 적은 전력으로 큰 전력을 제어
* 하나의 신호로 다수의 전원을 제어하는 등의 목적으로 사용
* 릴레이는 기계적으로 접점을 닫거나 열기 때문에 고속동작이 어려우며 소음도 발
* 생시키는 단점
* 최근에는 반도체를 활용한 릴레이가 많이 사용되고 있습니다

**PIR센서의 동작 원리 및 구조 (PPT 14)**

**PIR 센서의 동작 원리**

* 인체에서는 약 9 ㎛ ~ 11㎛ 정도의 적외선이 방출
* 방출된 적외선이 집광렌즈인 Fresnel Lens를 통과
* 센서 표면부에 위치한 Window에 집광
* 이 적외선 신호가 전압으로 출력되어 센서에 내장된 증폭기로 전달
* 조명 On/Off 제어 응용

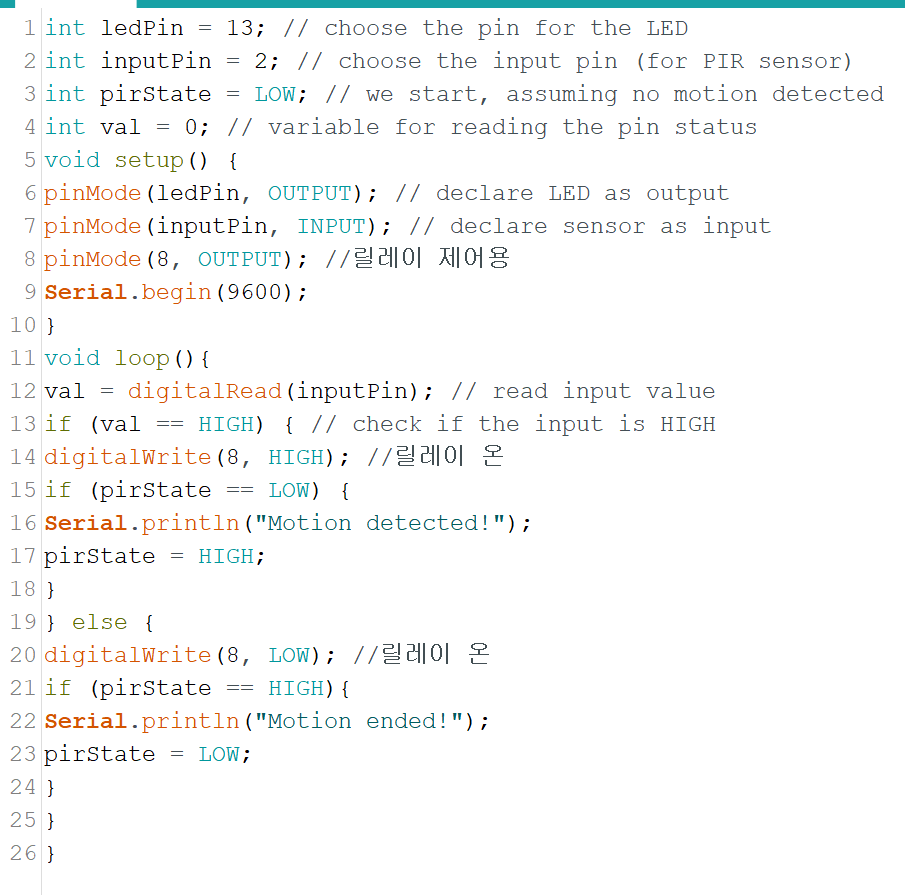
**F-Lens(Fresnel Lens)의 역할**

* 집광렌즈로 인체에서 발생된 적외선을 통과시키고
* 센서 표면부 Window에 적외선을 모아주는 역할
* 감지거리 확대와 감도 극대화
* 외부 환경 요소로부터 센서 보호 역할

**Window의 역할**

* 편광 필터의 역할 수행
* 일정한 주파수 대역의 적외선만 통과 시키는 역할

**릴레이와 모션 센서 활용 실습(PPT 19)**

****

**적외선 리모컨 활용 함수(PPT 6)**

|  |  |
| --- | --- |
| **Serial.begin(전송속도);**  시리얼 통신 포트를 컴퓨터와 연결  일반적으로 9600으로 설정한다. 19200, 57600, 115200 등의 값을 설정할 수 있다 | **실습결과.decode\_type;**  수신된 리모컨 코드의 타입이 저장된다(SONY, NEC 등) |
| **Serial.print(전송내용);**  괄호 안의 내용을 시리얼 통신으로 전송 (눈으로 본다) | **실습결과.value;**  수신된 리모컨의 코드가 저장된다 |
| **Serial.println(전송내용);**  Serial.print에 줄 바꿈 | **실습결과.bits;**  수신된 리모컨 코드의 비트수가 저장된다 |
| **IRrecv 리모컨이름(리모컨 수신 핀번호);**  리모컨 이름으로 리모컨 수신핀을 설정 | **실습결과.rawbuf;**  수신된 적외선 펄스의 시간이 배열 형태로 저장된다 |
| **리모컨이름.enableIRln();**  리모컨 수신시 타이머 인터럽트를 활성화 | **실습결과.rawlen;**  수신된 적외선 펄스의 시간이 배열의 개수가 저장된다. |
| **decode\_result 실습결과;**  입력된 신호의 결과를 ‘실습결과’ 변수에 입력 | **리모컨이름.resume();**  수신 후에 다른 신호를 수신하기 위하여 준비한다. |
| **리모컨이름.decode(&실습결과);**  리모컨 수신이 이루어졌을 때 그 값을 회신 | **리모컨이름.blink13(true);**  Arduino에 내장된 LED(13번핀)을 리모컨 수신 시 점등 |

**블루투스 API 기능(PPT 2)**

* 블루투스 설정
* 블루투스 장치 스캔
* 블루투스 장치와 페어링
* 블루투스 장치간 데이터 전송 및 수신

**애플리케이션 제작 과정(PPT 3~4)**

1. **프로젝트 생성**
2. **블루투스 권한설정**

****

1. **activity\_main.xml 코드 작성**
2. **MainActivity.java 코드 작성**

**AT 명령어(PPT 7~11)**

|  |  |  |
| --- | --- | --- |
| AT | 🡺 | 테스트 “OK” |
| AT+UART=9600,0,0 | 🡺 | 통신속도 및 stop bit, parity bit 설정 |
| AT+NAME? | 🡺 | 블루투스 이름 확인 |
| AT+NAME=LWB01 | 🡺 | 블루투스 이름 설정(변경) |
| AT+PSWD(PIN)? | 🡺 | 블루투스 패스워드 확인 |
| AT+PSWD(PIN)=”6877” | 🡺 | 블루투스 패스워드 설정(변경) |
| AT+ROLE=0 | 🡺 | Slave( 0 ), Master( 1 ) |

**안드로이드의 특징(PPT 10)안드로이드의 특징**

* 안드로이드의 핵심 커널(Kernel)은 리눅스(Linux)로 구성
* 안드로이드 애플리케이션 개발 언어는 Java를 사용
* 안드로이드 SDK에서 많은 라이브러리를 포함하고 있어 개발이 용이
* 오픈 소스를 지향하기 때문에 운영체제부터 관련 문서, 개발 도구 등 거의 모든 것을 무료로 사용 가능
* 지속적인 업그레이드를 제공

**안드로이의 구조(5항목, PPT 12~13)**

**응용 프로그램(Applications)**

* 안드로이드 스마트폰에서 사용할 수 있는 일반적인 응용 프로그램
* 웹 브라우저, 달력, 구글맵, 연락처, 게임 등 사용자 입장에서 가장 많이 사용
* Java로 제작됨

**응용 프로그램 프레임워크(Application Framework)**

* 안드로이드 API가 존재하는 곳
* 안드로이드폰 하드웨어에 접근할 때는 Java에 직접 접근하는 것이 아니라 API를 통해서 가능

**안드로이드 런타임(Android Runtime)**

* Java 코어 라이브러리와 달빅 가상 머신(Dalvik Virtual Machine) 또는 아트 런타임(ART Runtime)으로 구성

**라이브러리(Libraries)**

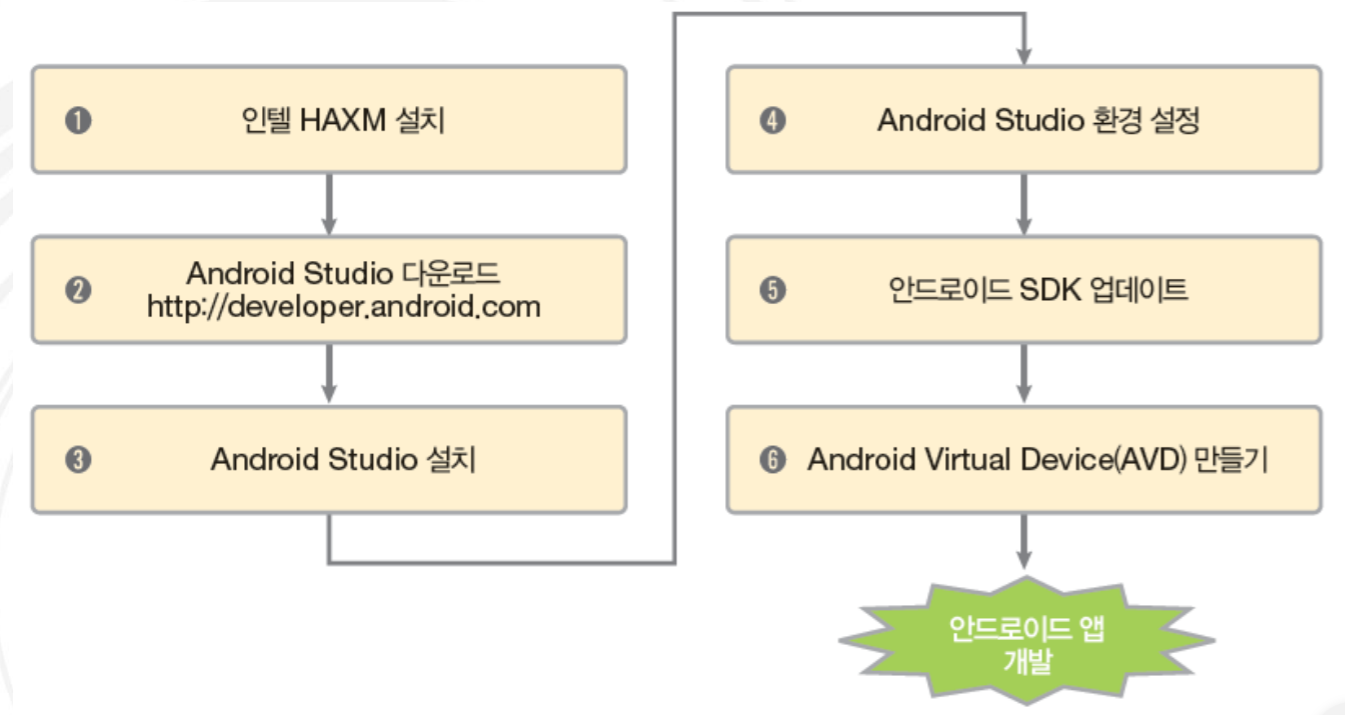
* 안드로이드에서 사용되는 여러 시스템 라이브러리는 시스템 접근 때문에 Java가 아닌 C로 작성

**리눅스 커널(Linux Kernel)**

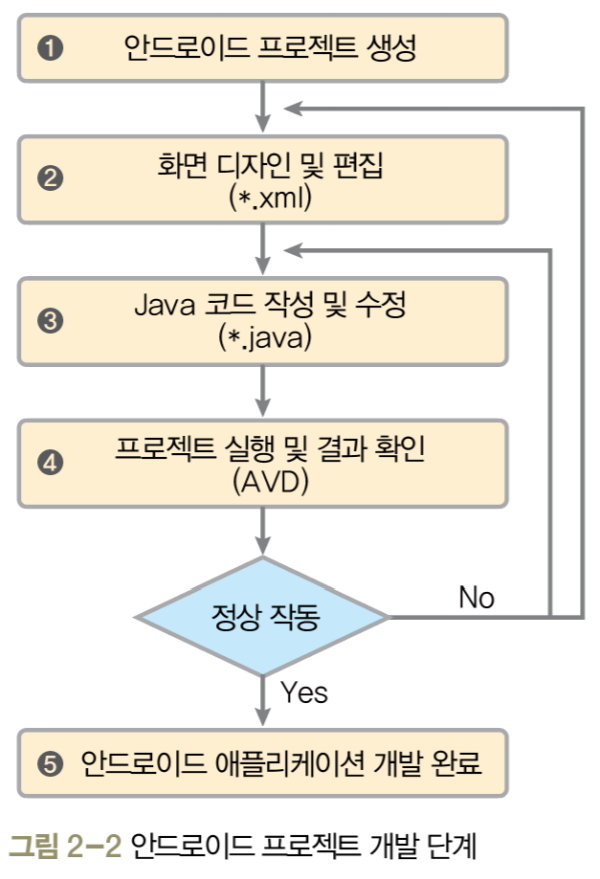
* 하드웨어의 운영과 관련된 저수준의 관리 기능이 들어 있음
* 주로 메모리 관리, 디바이스 드라이버, 보안 등

**안드로이드 개발 환경 종류 및 설치 순서(PPT 16~17)**

****

****

**안드로이드 프로젝트 개발 단계(PPT 4)**

****

**단축키 [Ctrl]+[Alt]+[O](PPT 55)의 동작 원리**

[Ctrl]+[Alt]+[O]를 누르면 불필요하게 임포트된 문장 제거

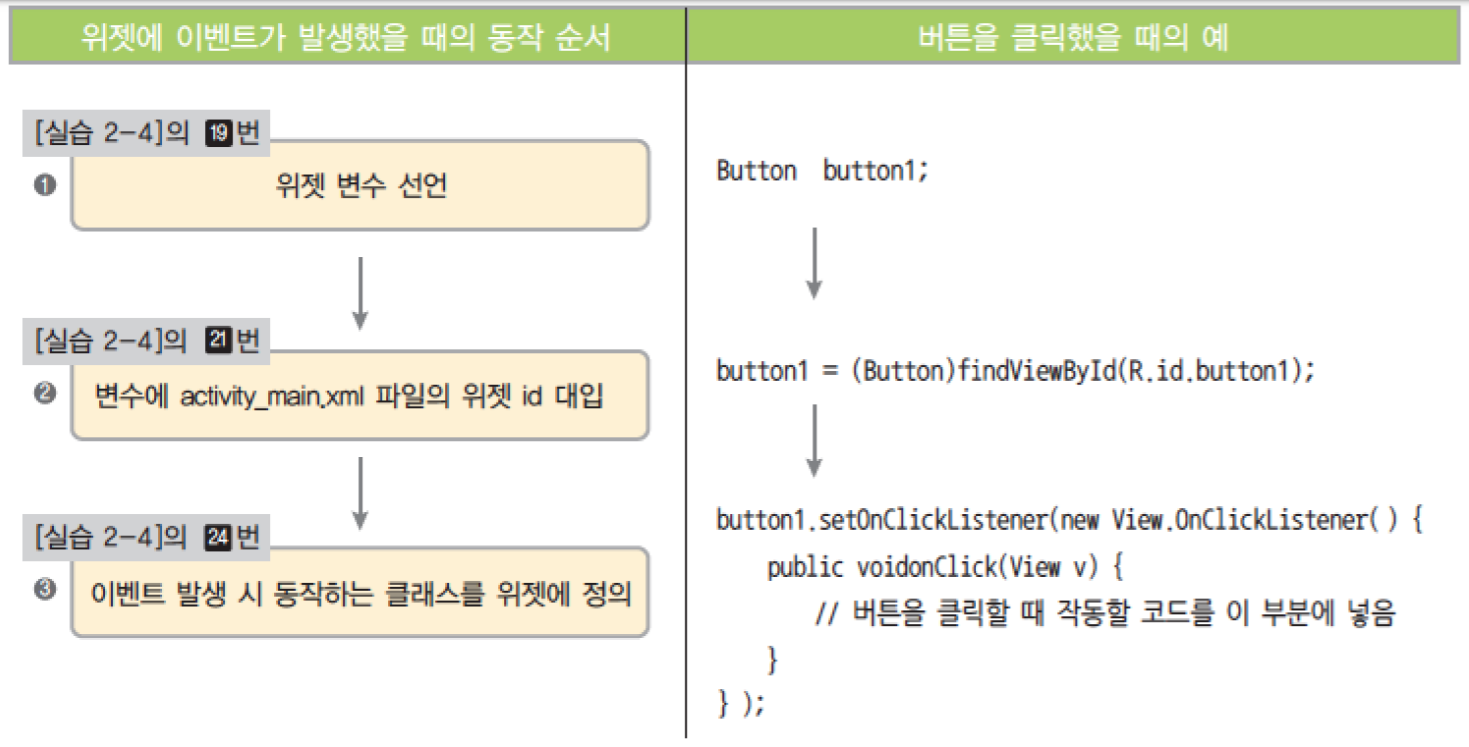
**단축키 [Alt]+[Enter](PPT 57)의 동작 원리**

[Alt]+[Enter] 누르면 Button과 관련된 클래스가 자동으로 import문에 추가됨

**setContentView(), findViewByid() 메소드 활용법**

**setContentView** : XML녀석을 띄우기????

**findViewByid :** 아이디 값으로 content 찾기

**위젯에 이벤트가 발생했을 때 동작 순서(PPT 65)**

**Layout 계층도 및 레이아웃에 자주 사용되는 속성(PPT 4~5)**

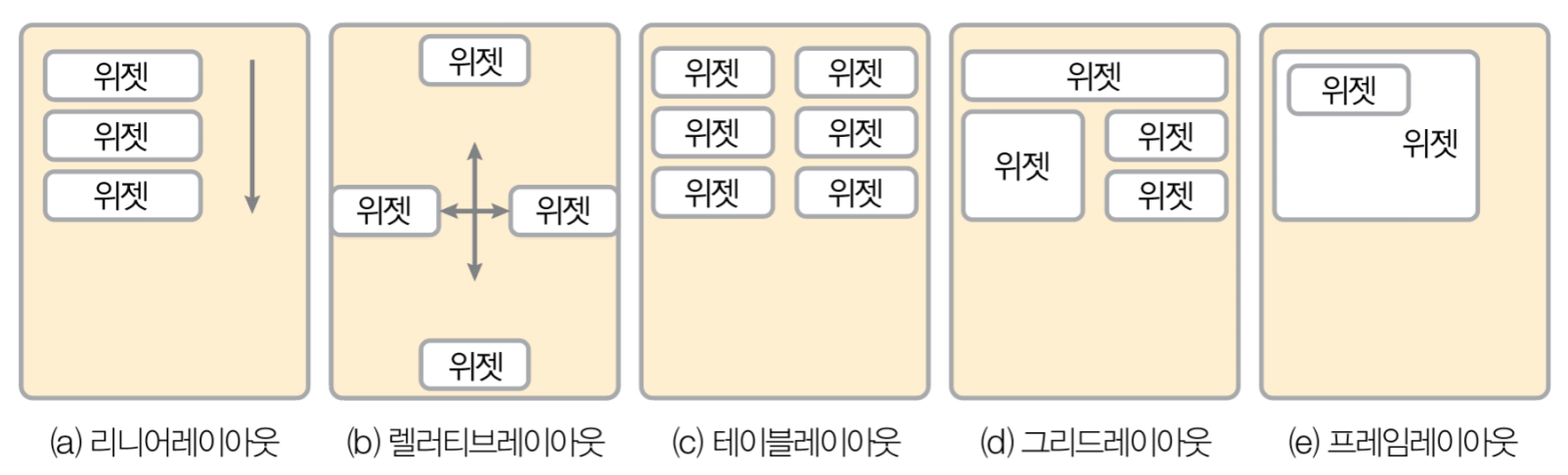
**Layout 계층도(PPT4)**

****

**자주 사용되는 속성(PPT5)**

* **orientation :** 레이아웃 안에 배치할 위젯의 수직 또는 수평 방향을 설정
* **gravity :** 레이아웃 안에 배치할 위젯의 정렬 방향을 좌측, 우측, 중앙으로 설정
* **padding :** 레이아웃 안에 배치할 위젯의 여백을 설정
* **layout\_weight :** 레이아웃이 전체 화면에서 차지하는 공간의 가중값을 설정, 여러 개의 레이아웃이 중복될 때 주로 사용
* **baselineAligned :** 레이아웃 안에 배치할 위젯을 보기 좋게 정렬

**레이아웃의 종류(PPT 6)**

****

**리니어레이아웃 :** 왼쪽 위부터 아래쪽 또는 오른쪽으로 차례로 배치

**렐러티브레이아웃 :** 위젯 자신이 속한 레이아웃의 상하좌우의 위치를 지정하여 배치

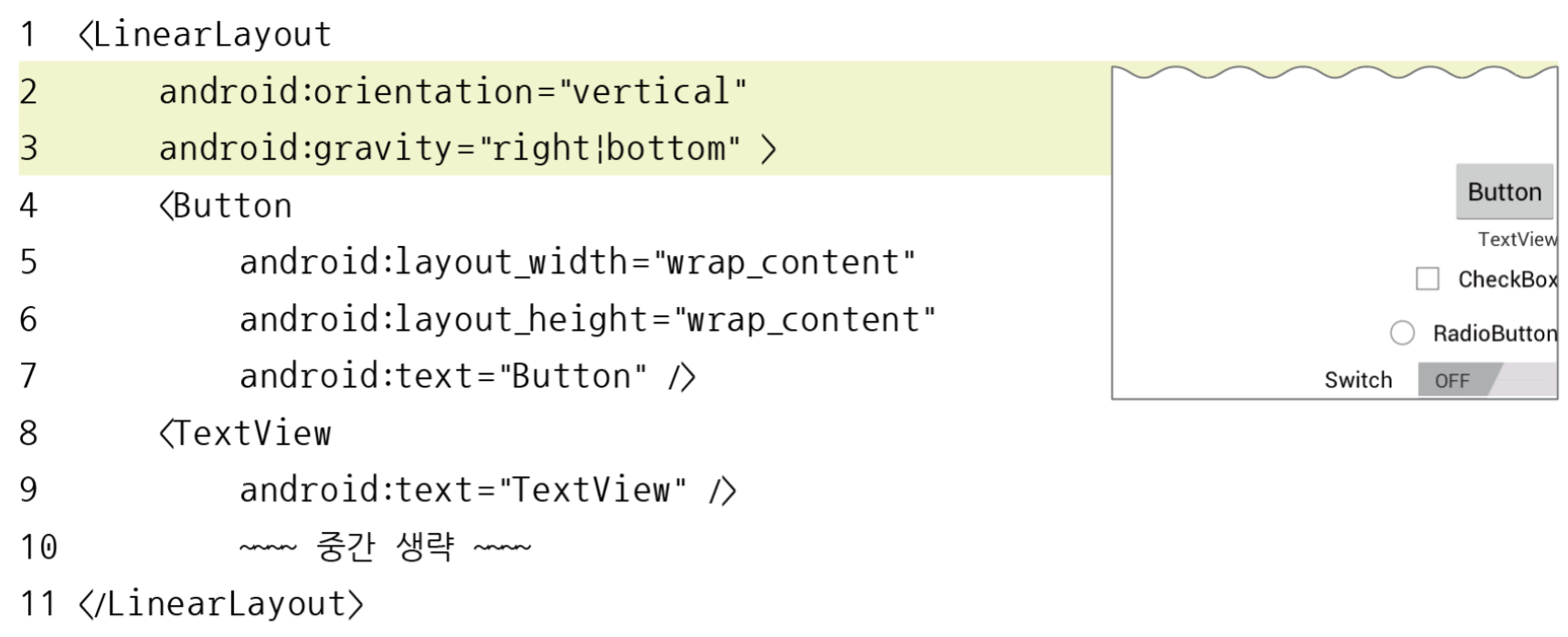
**테이블레이아웃 :** 위젯을 행과 열의 개수를 지정한 테이블 형태로 배열

**그리드레이아웃 :** 테이블레이아웃과 비슷하지만, 행 또는 열을 확장하여 다양하게 배치할 때 더 편리

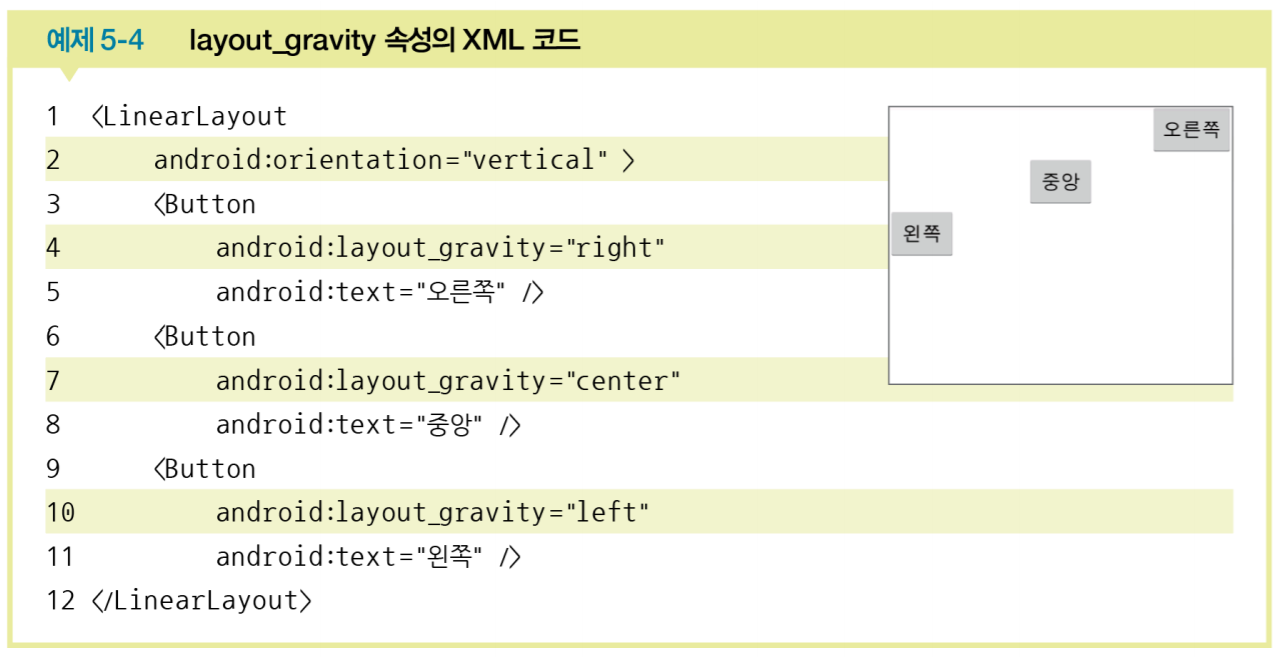
**프레임레이아웃 :** 위젯들을 왼쪽 위에 일률적으로 겹쳐서 배치하여 중복해서 보이는 효과를 냄

**gravity 속성 설정(PPT 10~11) 완성**

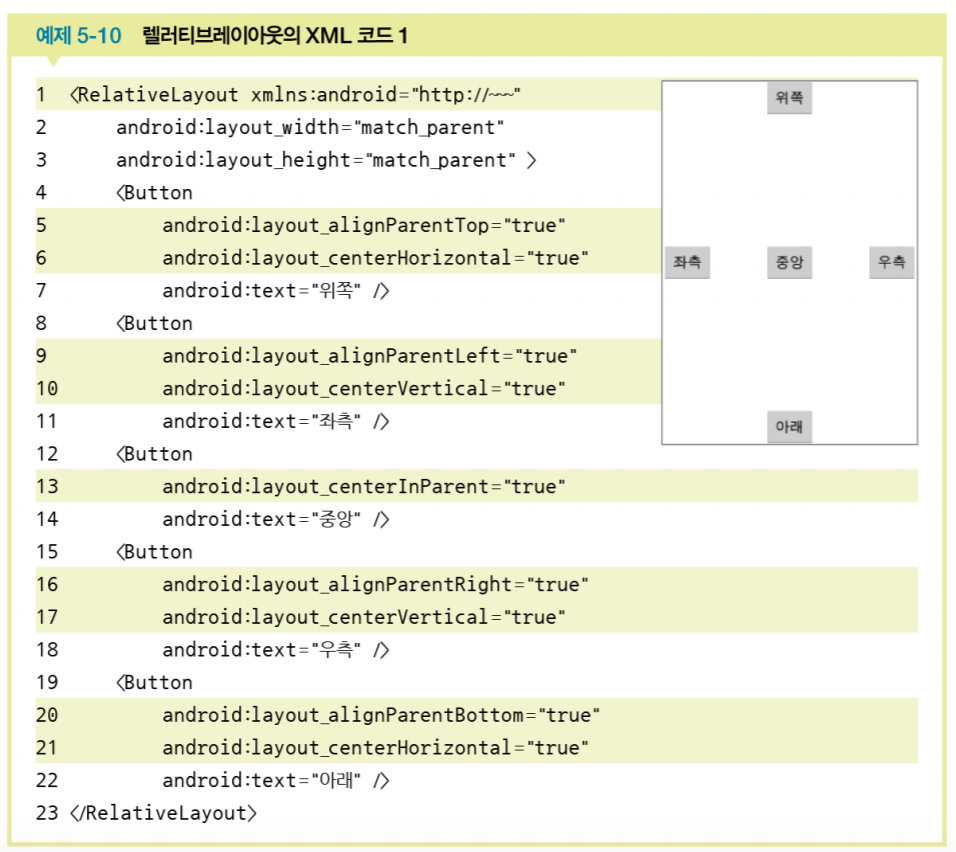
**gravity 속성은 레이아웃 안의 위젯을 어디에 배치할 것인지를 결정 (PPT 10)**

****

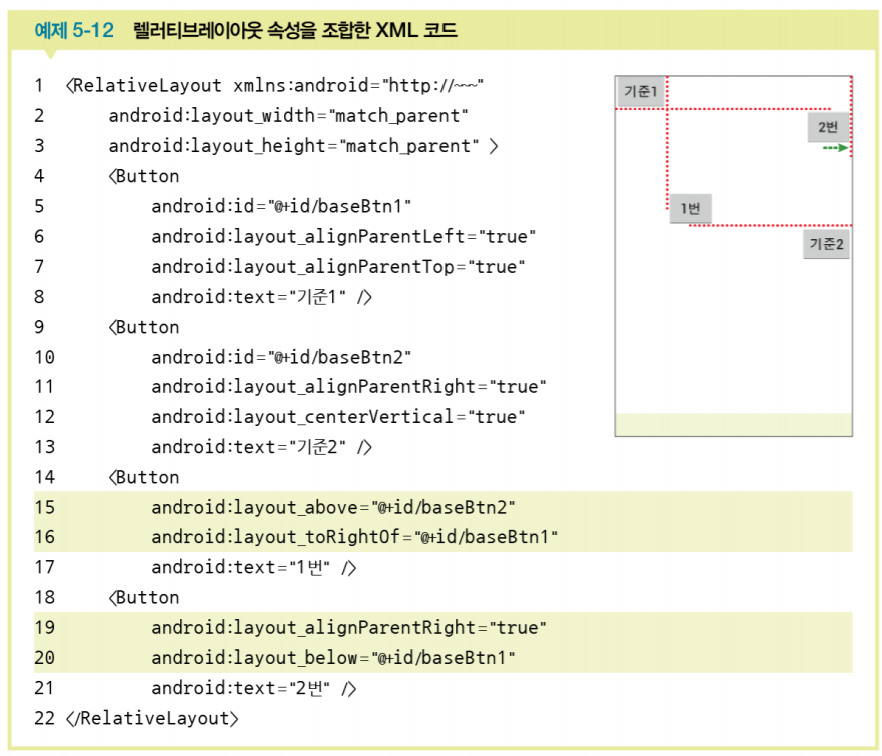
**layout\_gravity는 자신의 위치를 부모의 어디쯤에 위치시킬지를 결정 (PPT 11)**

****

**렐러티브레이아웃 예제(PPT 25, 28)**

****

**[ppt25]**

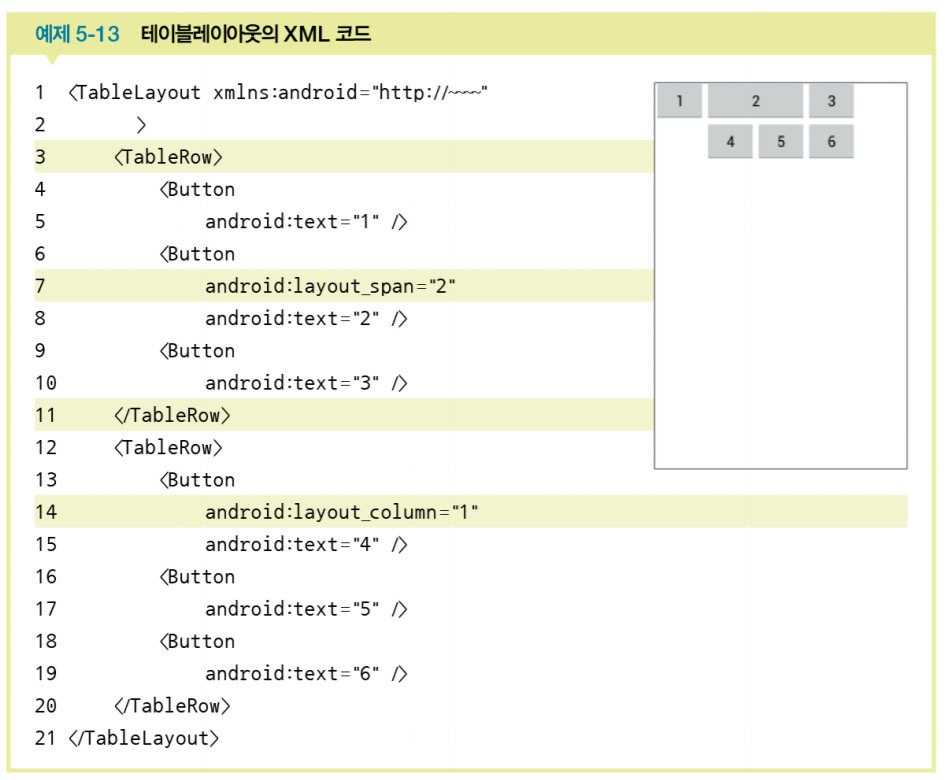
****

**[ppt28]**

**테이블레이아웃의 속성 및 예제(PPT 31~32)**

**테이블레이아웃의 속성(PPT31)**

* layout\_column : 지정된 열에 현재 위젯을 표시하라는 의미
* stretchColumns : 지정된 열의 폭을 늘리라는 의미
* stretchColumns =“\*” : 각 셀을 같은 크기로 확장, 전체 화면이 꽉 차는 효과



**[ppt32]**

**그리드레이아웃의 속성 및 예제(PPT 39~40)**

**그리드레이아웃 속성(PPT39)**

**① <GridLayout> 자체에 자주 사용되는 속성**

rowCount : 행 개수

columnCount : 열의 수

Orientation : 그리드를 수평 방향을 우선할지, 수직 방향을 우선할지를 결정

**② 그리드레이아웃 안에 포함될 "위젯"에서 자주 사용되는 속성**

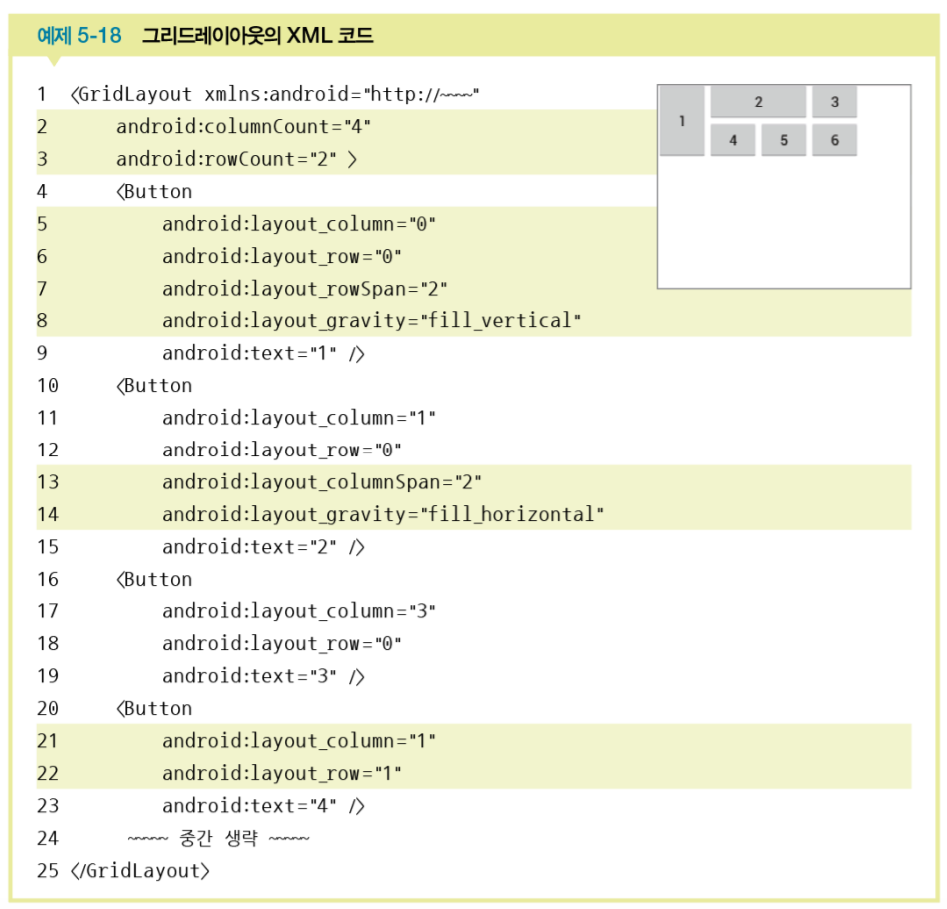
Layout\_row : 자신이 위치할 행 번호(0번부터 시작)

Layout\_column : 자신이 위치할 열 번호(0번부터 시작)

Layout\_rowSpan : 행을 지정된 수만큼 확장

Layout\_columnSpan : 열을 지정된 수만큼 확장

Layout\_gravity : 주로 fill, fill\_vertical, fill\_horizontal 등으로 지정행 또는 열 확장시, 위젯을 확장된 셀에 꽉 채우는 효과를 냄

****

**[ppt40]**

**프레임레이아웃의 속성(PPT 43)**

**foreground :** 프레임레이아웃의 전경 이미지를 지정

**foregroundGravity :** 전경 이미지의 위치를 지정