

CONSTRAINT LAYOUT

2020년도 1학기 앱 프로그래밍 3주차 수업



목 차

1. 레이아웃 정의
2. 위젯 크기 설정
3. 위젯 정렬
4. Constraint Toolbar
5. 속성 창
6. 실습 예제
7. 체이닝과 그룹
8. 앱 프로그래밍 학습을 위한 복습

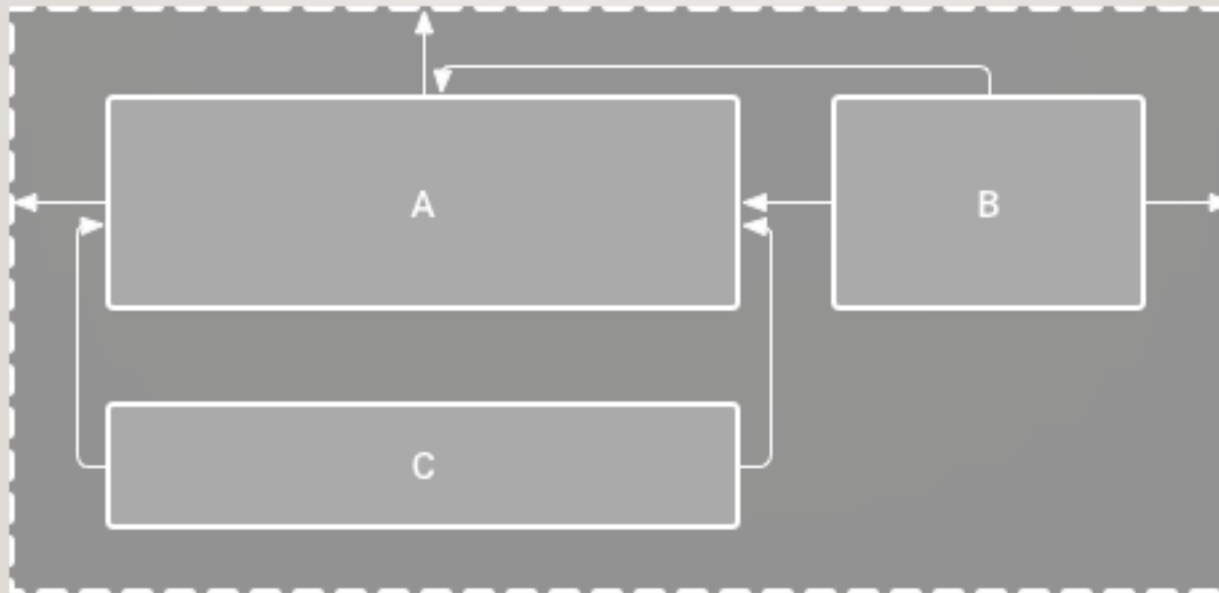
| 교시 |

2020년도 1학기 앱 프로그래밍 - 개요



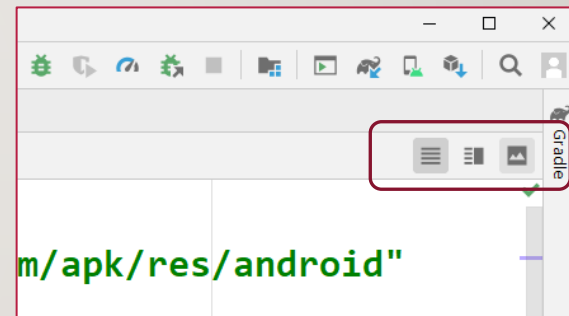
I. CONSTRAINTLAYOUT

- 정의
 - 제약 레이아웃
 - 레이아웃에 배치되는 뷰들에 다양한 제약(constraint)을 적용하여 각 뷰의 위치와 크기 결정



2. 위젯 크기 설정

- 개요
 - 뷰에 배치될 위젯의 크기 설정 2가지 제공
 1. 리사이즈 핸들(resize handle) 사용 방법
 2. 직접 값을 입력하는 방법
 - 주의 : XML 편집보다 Design 편집 사용!!!
- 안드로이드 스튜디오 업데이트
 - Design과 XML code 그리고 Split 버튼 확인

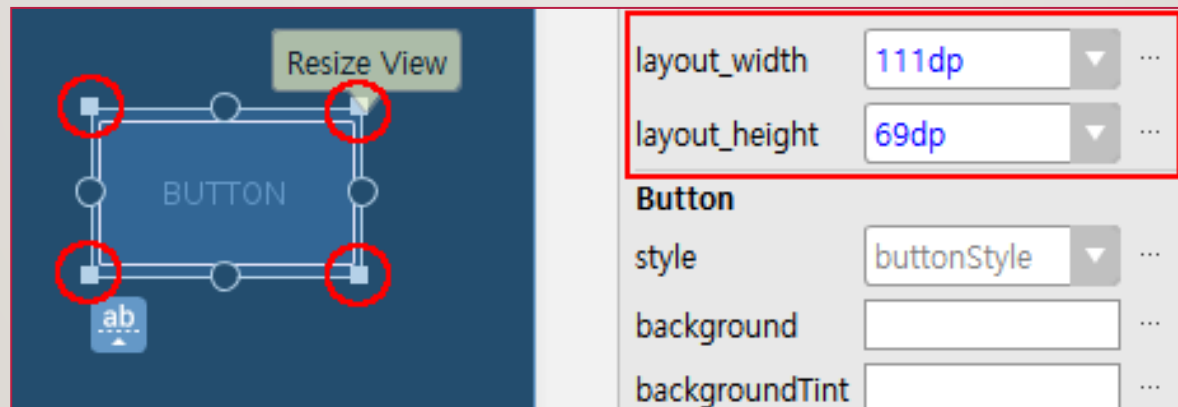


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

[XML Code 보기 선택 상태]

2. 위젯 크기 설정(계속)

- 리사이즈 핸들 사용법
 - 디자인 뷰에서 위젯 선택
 - 위젯 모서리에 나타나는 4부분의 사각형 핸들을 마우스로 조작
 - 드래그 시 우측 속성창의 layout_width, layout_height 값 변경 확인



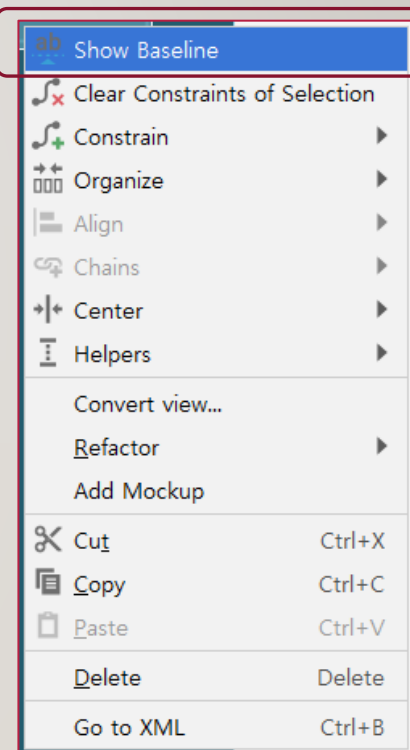
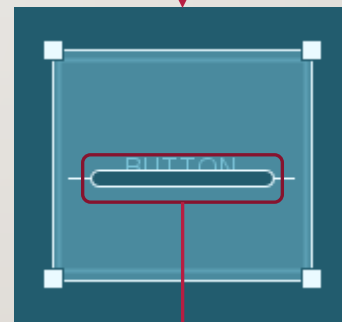
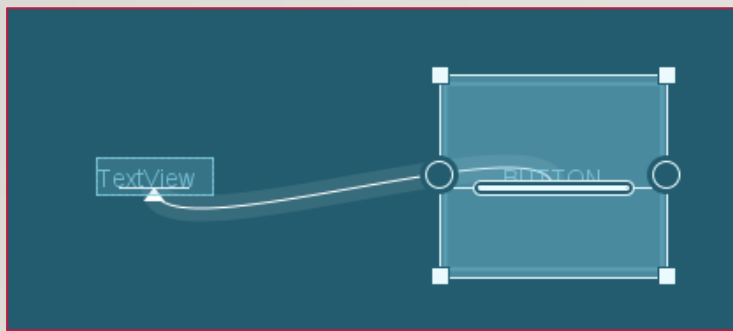
- 사이드 핸들(side handle)이용법

- 사이드 핸들을 이용하여 위젯이 다른 위젯 또는 화면의 경계에 대해서 어떻게 정렬될지 지정
 - 사이드 핸들 : 위젯의 사면에 나타나는 동그란 모양
- 좌우 사이드 핸들 : 가로축에 위치한 위젯들 간의 정렬 설정
- 상하 사이드 핸들 : 세로축에 위치한 위젯들 간의 정렬 설정
 - 주의 : 좌우 핸들을 상하 핸들에 연결할 수 없음!
- 위젯들 뿐만 아니라 화면의 각 경계 또는 임의로 추가된 수직, 수평 가이드 라인과 정렬 설정 가능



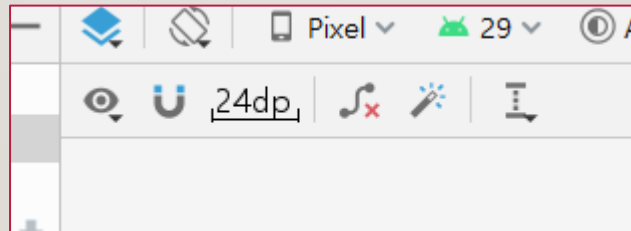
3. 위젯 정렬(계속)

- 위젯 내에 포함된 텍스트 위치로 정렬 기준 설정
 - 베이스라인 핸들 (baseline handle)
- 위젯에 마우스 우측버튼 클릭 “Show Baseline” 선택
 - 위젯 내부에 베이스 라인 핸들 나타남
- 베이스 라인 핸들 선택
- 다른 위젯의 텍스트에 연결
- 연결된 위젯은 연결된 텍스트 하단 정렬

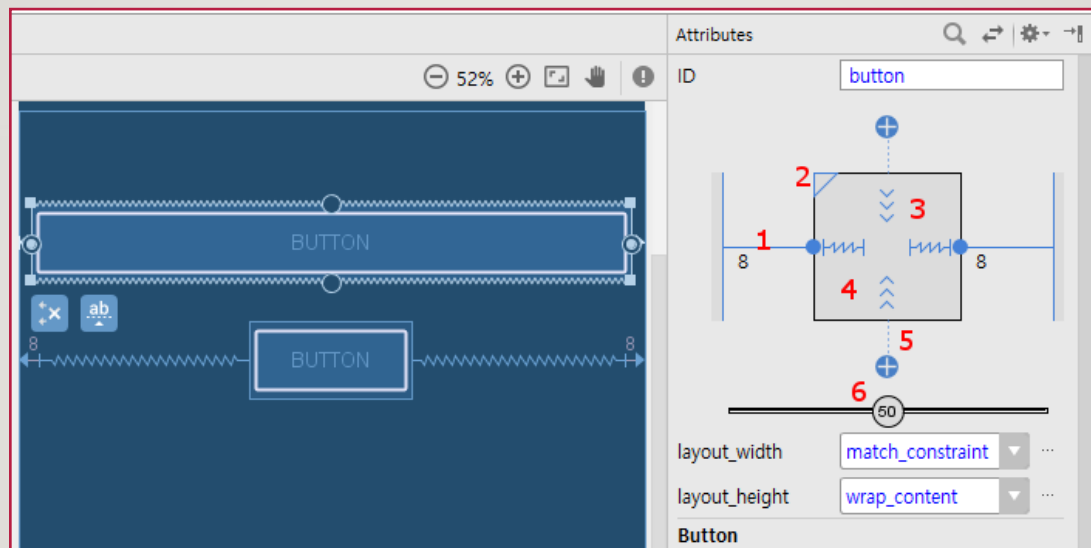


4. CONSTRAINT TOOLBAR

- View Options : 제약 관계 보기 버튼
- Autoconnect : 위젯을 배치할 때 자동으로 이웃한 위젯과 관계를 맺음
- Default Margins : 위젯 간의 기본 마진 값 지정
- Clear All Constraints : 모든 관계를 삭제
- Infer Constraints : 레이아웃내 배치된 모든 위젯을 현재 위치를 기반으로 관계 맺음
- Guidelines : 수직, 수평 가이드 라인 생성(가이드 라인과 위젯 정렬에 사용)

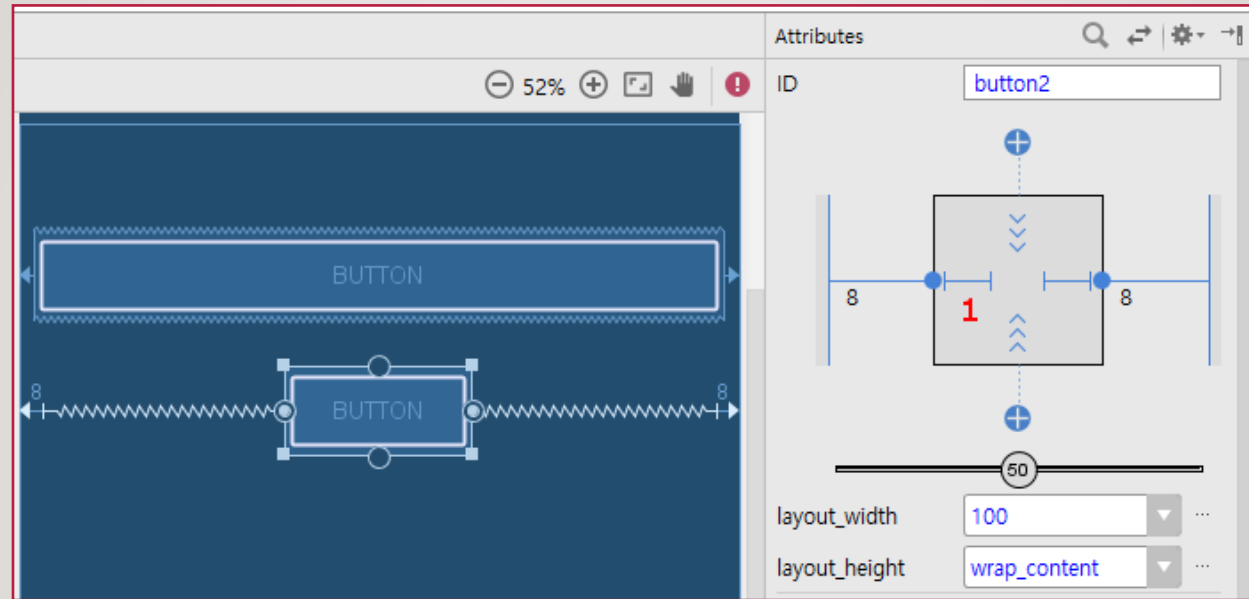


5. 속성 창



1. Margin : layout_width가 match_constraint 이므로 좌우 여백 8만 있고 전체를 위젯이 차지
2. Aspect Ratio : 클릭하면 위젯의 가로 세로 비율을 지정할 수 있음
3. Wrap Content : 쉬프트로 된 모양은 wrap_content 로 지정되어 있음을 나타냄
4. Any Size : 톱니 모양은 최대한의 너비를 차지하게 됩니다. match_constraint로 지정되어 있음
5. 점선 : 관계가 없음
6. Horizontal Bias : 위젯의 좌우측 위치 비율을 지정합니다. 드래그해서 바꿀 수 있음

5. 속성 창(계속)



1. Fixed : 길이가 고정되어 입력된 상태를 나타냄
 - layout_width가 100dp로 된 예를 나타냄

2교시

2020년도 1학기 앱 프로그래밍 - 예제 실습



6. 실습 예제 I

- 상위 요소 포지셔닝

상위 요소 포지셔닝

보기의 측면을 레이아웃의 대응하는 가장자리로 제한합니다.

그림 7에서 보기의 왼쪽은 상위 레이아웃의 왼쪽 가장자리에 연결되어 있습니다. 여백을 사용하여 가장자리로부터의 거리를 정의할 수 있습니다.

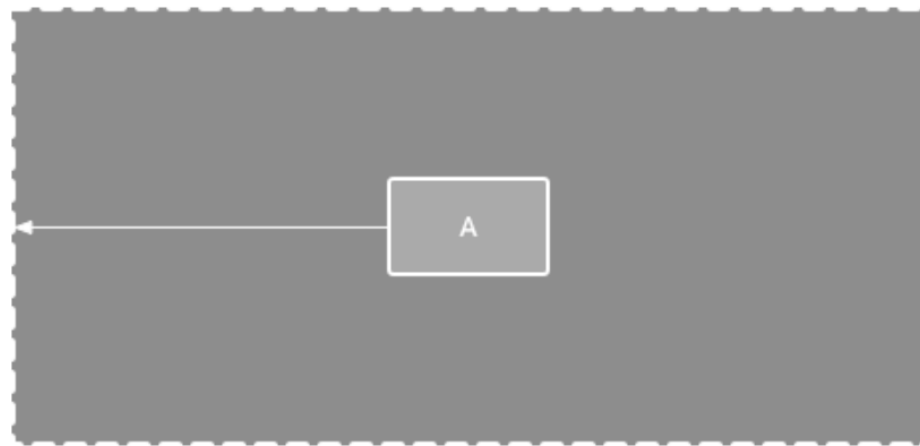


그림 7. 상위 요소의 가로 제약조건

6. 실습 예제 2

- 위치 순서 지정

위치 순서 지정

가로 또는 세로로 두 보기가 표시되는 순서를 정의합니다.

그림 8에서 B는 항상 A의 오른쪽에 있도록 제한되고 C는 A 아래 있도록 제한됩니다. 그러나 이 제약조건은 정렬을 의미하지 않으므로, B는 여전히 위아래로 이동할 수 있습니다.

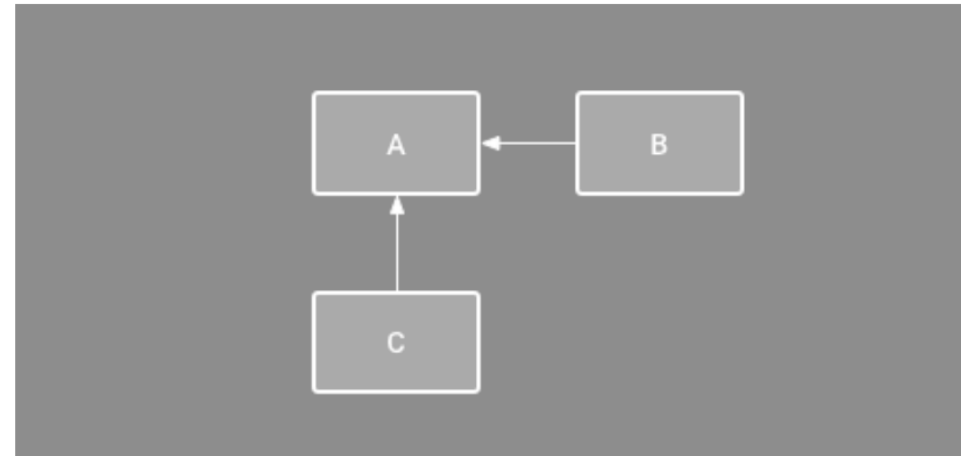


그림 8. 가로 및 세로 제약조건

6. 실습 예제 3


- 정렬

정렬

보기의 가장자리를 다른 보기의 가장자리에 맞게 정렬합니다.

그림 9에서 B의 왼쪽은 A의 왼쪽에 맞게 정렬됩니다. 보기 중심을 정렬하려면 양쪽에 제약조건을 만듭니다.

제약조건에서 안쪽으로 보기를 드래그하여 기준에서 벗어나 정렬할 수 있습니다. 예를 들어 그림 10에서는 24dp 벗어나 정렬된 B를 보여줍니다. 이와 같은 오프셋은 제한된 보기의 여백으로 정의됩니다.

정렬할 보기를 모두 선택한 다음 툴바에서 **Align**  을 클릭하여 정렬 유형을 선택할 수도 있습니다.

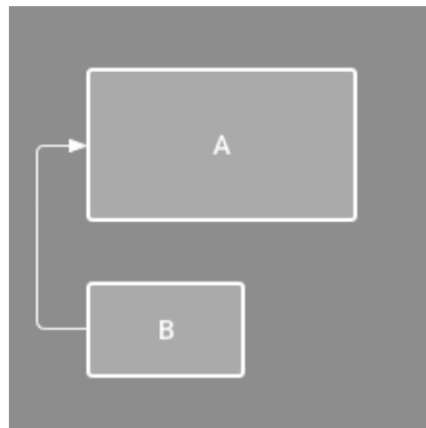


그림 9. 가로 정렬 제약조건

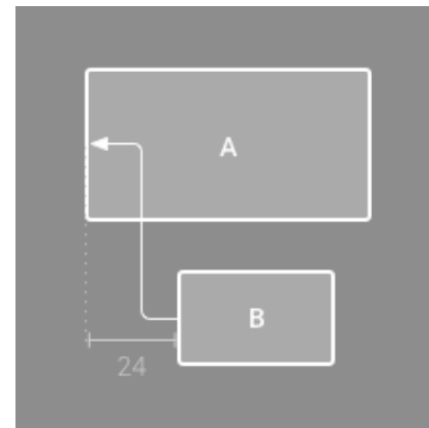


그림 10. 오프셋 가로 정렬 제약조건

6. 실습 예제 4

- 정렬

기준선 정렬

보기의 텍스트 기준선을 다른 보기의 텍스트 기준선에 맞추는
다.

그림 11에서 B의 첫 번째 줄은 A의 텍스트에 맞게 정렬됩니다.

기준선 제약조건을 만들려면 제한할 텍스트 보기를 마우스 오른쪽 버튼으로 클릭한 다음 **Show Baseline**을 클릭합니다. 그런 다음 텍스트 기준선을 클릭하고 선을 다른 기준선으로 드래그합니다.

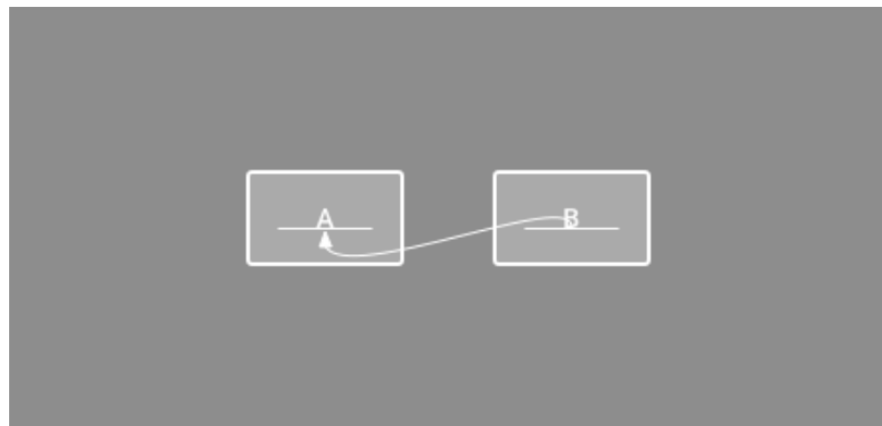


그림 11. 기준선 정렬 제약조건

3교시

2020년도 1학기 앱 프로그래밍 – 체이닝, 실습 및 복습 과제



7. 체이닝

체인은 양방향 위치 제약조건을 통해 서로 연결된 보기 그룹입니다. 체인의 보기는 세로 또는 가로로 분산될 수 있습니다.

체인 스타일은 다음 방법 중 하나로 지정할 수 있습니다.

- 1 **넓히기:** 보기가 균등하게 분산됩니다(여백을 처리한 후). 이는 기본값입니다.
- 2 **내부에서 넓히기:** 첫 번째 보기와 마지막 보기는 체인의 각 끝에 있는 제약조건에 고정되고 나머지 보기는 균등하게 분산됩니다.
- 3 **가중:** 체인이 넓히기 또는 내부에서 넓히기로 설정되면 하나 이상의 보기를 '제약조건과 일치'로 설정하여 나머지 공간을 채울 수 있습니다(0dp). 기본적으로 공간은 '제약조건과 일치'로 설정된 각 보기 사이에 균등하게 분배되지만, `layout_constraintHorizontal_weight` 및 `layout_constraintVertical_weight` 속성을 사용하여 각 보기에 중요도 가중치를 할당할 수 있습니다. **선형 레이아웃**의 `layout_weight`와 작동 방식이 동일합니다. 따라서 가중치가 가장 높은 보기에 가장 많은 공간이 지정되고, 가중치가 동일한 보기에는 동일한 크기의 공간이 지정됩니다.
- 4 **채우기:** 보기가 여백을 제외한 간격 없이 배치됩니다. 체인의 헤드 보기 편향을 변경하여 전체 체인의 편향을 좌우 또는 상하로 조정할 수 있습니다.

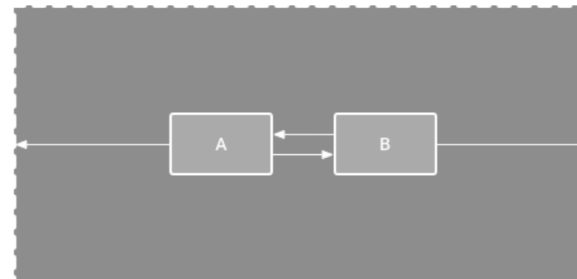


그림 17. 두 개의 보기가 있는 가로 체인

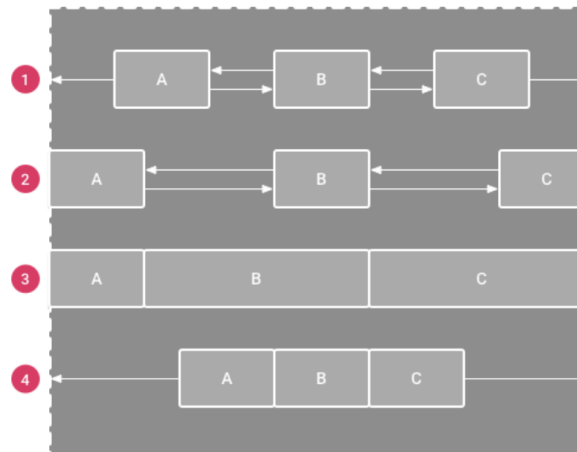


그림 18. 각 체인 스타일의 예

8. 앱 프로그래밍 학습을 위한 복습 (과제)

- 복습 문제
 - 2교시와 3교시에 설명한 예제를 모두 실습해본다.
 - 각 실습 결과가 저장된 xml 파일들을 제출한다.
- 제출 방법
 - <http://ctl.gtec.ac.kr> 의 과제 제출란에 제출
 - 5개의 실습을 수행하고 각 xml 파일 5개를 압축하여 제출한다.
 - 압축파일은 자신의 “반_학번_이름.zip” 로 한다.
 - 제출 일자와 시간을 엄수하여 제출하세요!