

둘째 마당, 안드로이드 완벽 가이드

01. 안드로이드 스튜디오와 기본 레이아웃 익히기

01-2. 뷰와 뷰의 크기 속성 이해하기

- 뷰 : 사용자의 눈에 보이는 화면의 구성 요소
- 뷰 그룹 : 뷰를 여러개 포함하고 있는 것
 - 뷰 그룹 안에 뷰의 위치를 지정할 수 있다.
 - 여러 개의 뷰를 담고 있는 뷰 그룹도 또 다른 하나의 뷰로 본다.
- 버튼 : 텍스트뷰를 상속하여 만들어진 것
 - 텍스트뷰의 속성을 그대로 가지고 있다.
- 위젯 : 뷰 중에서 일반적인 컨트롤의 역할을 하는 것
- 레이아웃 : 뷰그룹 중에서 내부에 뷰를 포함하면서 그것들을 배치하는 역할을 하는 것

뷰의 크기 속성

- 뷰는 위젯과 레이아웃으로 구분한다.
- 뷰의 가로 크기와 세로 크기가 없으면 안드로이드는 XML 레이아웃이 잘못되었다고 알려준다.
- XML 레이아웃
 - 디자이너 도구에서 [Text] 탭을 눌러보면 태그와 속성으로 구분되어 있다.
 - 태그 : 꺾쇠 모양(<)의 기호 안에 글자가 들어간 형태이며, 시작 태그와 끝 태그로 구성되어 있다.
 - 끝 태그는 꺾쇠 안에 / 기호가 추가로 붙어 있습니다.
 - 태그
 - <TextView : 시작 태그
 - 여러개의 속성
 - /> : 끝 태그
 - 바깥쪽의 태그
 - 다른 뷰를 담고 있는 레이아웃
 - 태그 안에 들어간 속성들은 모두 공백이나 줄바꿈으로 구별됩니다.
 - 대부분 android: 가 앞에 붙는다.
 - android: 는 안드로이드의 기본 API에서 정의한 속성이라는 의미
 - 직접 정의했거나 외부 라이브러리를 사용했을 때 그 안에 정의된 속성이라면 android가 아닌 다른 단어가 속성 앞에 붙는다.
 - 태그 안에 추가하는 속성의 형태

속성 Prefix: 속성명="속성값1"

```
android:layout_width="wrap_content"
app:layout_constraintLeft_toLeftOf="parent"
```

◦ 속성 중 공통 속성

- layout_width : 뷰의 폭을 설정
- layout_height : 뷰의 높이를 설정
- 뷰가 화면 안에 들어가려면 뷰의 크기와 위치가 결정되어야 하기 때문에 이 두 속성이 필수
- 없으면 오류.
- 가로 크기와 세로 크기의 값으로는 세 가지 값 중의 하나가 들어감


wrap_content : 뷰에 들어 있는 내용물의 크기에 자동으로 맞춥니다.
match_parnet : 뷰를 담고 있는 뷰그룹의 여유 공간을 꽉 채웁니다.
숫자로 크기 지정 : 숫자를 사용해 크기를 지정한다. (크기는 단위가 있어야 한다.)

01-3. 제약 레이아웃 익히기

안드로이드 스튜디오에서는 디폴트 레이아웃을 **제약 레이아웃**이라는 것으로 자동 설정한다.

제약 조건 이해하기

- 제약 레이아웃의 가장 큰 특징
 - 뷰의 크기와 위치를 결정할 때 제약 조건을 사용한다.
 - 제약조건 : 뷰가 레이아웃 안의 다른 요소와 어떻게 연결되는지 알려주는 것
 - 뷰의 연결점과 대상을 연결합니다.
 - 이때, 버튼의 입장에서 자신을 감싸고 있는 레이아웃을 **부모 레이아웃**이라 한다.
 - 이 연결점은 **핸들**이라고 한다.
 - 즉, 잡아서 조절할 수 있는 것
- ex) 버튼의 왼쪽을 부모 레이아웃과 연결해 주세요.
 - 버튼의 왼쪽 연결점을 부모 레이아웃의 왼쪽 벽면과 연결할 수 있다.
 - 그러면 부모 레이아웃이 타깃이 되어 연결점과 타깃이 연결됩니다.
- 꼭 필요한 제약 조건
 - 타깃 (연결선을 만들 때, 뷰의 연결점과 타깃이 필요함.)
 1. 같은 레이아웃 안에 들어있는 다른 뷰의 연결점
 2. 부모 레이아웃의 연결점
 3. 가이드라인
 - 대상 뷰와 타깃의 연결점
 1. 위, 아래, 오른쪽, 왼쪽
 2. 가로축의 가운데, 세로축의 가운데

- 3. 베이스라인 -> 텍스트를 보여주는 뷰인 경우에만 적용됨.
- 마진 : 연결점과 타겟과의 거리를 나타낸다.
-  1546934258458

화면 가운데에 뷰 배치하기

- 뷰는 가로축과 세로축을 기준으로 배치된다.
- 가로축에 해당하는 왼쪽과 오른쪽에 모두 연결선이 만들어지는 경우에는 양쪽의 타겟 중간에 위치하게 된다.
 - 마진 값 고려 x
- 만약, 정확히 한 가운데가 아니라, 약간 위쪽으로 치우치거나 왼쪽으로 치우치게 만들고 싶다면
 - Attributes --> 레이아웃 속성 -> 가로/세로 방향 바이어스
 - 바이어스는 한 쪽으로 얼마나 치우쳐져 있는지 나타냄
 - 기본값 50
- 뷰의 크기를 더 크게 하고 싶다면, 버튼 선택 후 각 모서리에 표시된 작은 점을 끌어당기면 됨.

1. 부모 여유 공간 채우기

- 사각형 안쪽의 선이 구불구불한 선으로 표시됨
- layout_width의 값은 **match_constraint**

2. 뷰의 내용물 채우기

- 사각형 안쪽의 선이 중앙을 향하는 화살표로 표시됨
- layout_width의 값은 **wrap_content**

3. 고정 크기

- 사각형 안쪽의 선이 직선으로 표시됨
- layout_width 와 layout_height의 값은 **지정한 값**

- 리니어 레이아웃의 경우, 하나의 버튼을 추가한 후 다른 버튼을 또 추가하면 첫 번째 버튼이 가질 수 있는 여유 공간이 없거나 줄어들 수 있다.
- 제약 레이아웃의 경우, 뷰를 담고 있는 부모 레이아웃 안에서 크기나 위치를 마음대로 조절할 수 있다.

가이드라인 사용하기

- 가이드라인 : 여러 개의 뷰를 일정한 기준 선에 정렬할 때 사용

XML 원본에 추가된 속성 확인하기

- 최상위 레이아웃 태그

```
<?xml version="1.0" encoding="utf-8"?>
```

- 위젯이나 레이아웃이 안드로이드 기본 API에 포함되어 있다면 그 위젯이나 레이아웃의 이름만 입력하면 됨.
- 만약 외부 라이브러리에서 불러온 것이라면 패키지 이름도 같이 입력해야 한다.
- xmlns: 태그
 - xmlns 뒤에 있는 접두어의 의미
 1. xmlns:android
안드로이드 기본 SDK에 포함되어 있는 속성을 사용합니다.
 2. xmlns:app
프로젝트에서 사용하는 외부 라이브러리에 포함되어 있는 속성을 사용합니다.
 3. xmlns:tools
안드로이드 스튜디오의 디자이너 도구 등에서 화면에 보여줄 때 사용
이 속성은 앱이 실행될 때는 적용되지 않고, 안드로이드 스튜디오에서만 적용됩니다.
 - 즉, android:layout_width와 android:layout_height 속성이 실제로는 안드로이드 기본 SDK에 들어있는 layout_width와 layout_height 속성을 사용한다는 의미다.
- id 속성은 뷰를 구분하는 구분자 역할을 한다.
 - id 속성이 사용되는 용도
 1. XML 레이아웃 파일 안에서 뷰를 구분할 때.
 2. XML 레이아웃 파일에서 정의한 뷰를 자바 소스 파일에서 찾을 때
- 제약 레이아웃에서 하나의 뷰를 다른 뷰와 연결할 때 사용하는 XML 속성의 이름 규칙

layout_constraint[소스 뷰의 연결점]_[타겟 뷰의 연결점]="[타겟 뷰의 id]"

- 여기서 id 속성 값은

@+id/아이디 값

- 제약 레이아웃에서 사용할 수 있는 속성

```
layout_constraintTop_toTopOf
layout_constraintTop_toBottomOf
layout_constraintBottom_toTopOf
layout_constraintBottom_toBottomOf
layout_constraintLeft_toTopOf
layout_constraintLeft_toBottomOf
layout_constraintLeft_toLeftOf
layout_constraintLeft_toRightOf
layout_constraintRight_toTopOf
layout_constraintRight_toBottomOf
layout_constraintRight_toLeftOf
layout_constraintRight_toRightOf
```

- layout_constraintGuide_begin 속성

1. `layout_constraintGuide_begin`
세로 방향인 경우 왼쪽부터, 가로방향인 경우 위쪽부터의 거리 지정
2. `layout_constraintGuide_end`
세로 방향인 경우 오른쪽부터, 가로방향인 경우 아래쪽부터의 거리 지정
3. `layout_constraintGuide_percent`
`layout_constraintGuide_begin` 속성 대신 지정하되 % 단위로 거리 지정

- `match_parent`, `wrap_content`를 사용하면 여러 단말의 해상도를 지원할 수 있다.
- 뷰의 폭과 높이 사용 가능 단위와 그 의미

단위	단위표현	설명
px	픽셀	화면 픽셀의 수
dp 또는 dip	밀도 독립적 픽셀	160dpi 화면을 기준으로 한 픽셀 ex) 1인치 당 160개의 점이 있는 디스플레이 화면에서 1dp는 1px와 같다. 1인치 당 320개의 점이 있는 디스플레이 화면에서는 1dp는 2px와 같다.
sp 또는 sip	축척 독립적 픽셀	텍스트 크기를 지정할 때 사용하는 단위 가변 글꼴을 기준으로 한 픽셀로 dp와 유사하나 글꼴의 설정에 따라 1sp당 픽셀수가 달라짐.
in	인치	2인치로 된 물리적 길이
mm	밀리미터	1밀리미터로 된 물리적 길이
em	텍스트 크기	글꼴과 상관없이 동일한 텍스트 크기 표시

`app:layout_constraintLeft_toLeftOf` 속성

--> 이 버튼의 왼쪽의 연결점과 타겟 뷰의 왼쪽 연결점을 연결하여 제약 조건을 만듭니다.