

◆ 레이아웃 클래스: 화면에 출력되는 대상이지만, 자체 UI는 가지지 않는다. 뷰들을 자신에게 포함(Add)하고 적절하게 배치하여 한꺼번에 출력한다.

● LinearLayout: 자신에 포함된 뷰들을 순서대로 가로나 세로 방향으로 나열한다. 때문에 필수 속성으로 orientation이 있다 (vertical 가로/horizontal 세로). LinearLayout을 중첩하여 레이아웃 클래스 하위에 다른 레이아웃 클래스를 배치하면 복잡한 화면도 구현할 수 있다.

LinearLayout 속성

▶ gravity: 뷰의 내용(content)을 뷰 영역 내에서 어디에 나타낼지 설정하는 속성. 한 가지 이상 방향 설정 시 | 사용.

▶ layout_gravity: 뷰를 레이아웃 영역 내에서 어디에 나타낼지 설정하는 속성. 한 가지 이상 방향 설정 시 | 사용. LinearLayout은 방향에 따라 포함된 뷰를 차례대로 나열하므로 orientation 이 vertical이면 세로 방향으로는 뷰가 포함된 순서대로 나열되어 세로 방향의 layout_gravity가 적용되지 않고, 반대로 orientation 속성이 horizontal이면 가로 방향의 layout_gravity가 적용되지 않는다.

▶ weight: 화면의 여백을 배치된 뷰들이 얼마만큼 확장해서 차지할지 설정하는 속성. 만약 LinearLayout의 방향이 vertical 세로 방향이면, layout_height를 모두 0dp로 설정한 다음에, weight을 설정하여 여백 확장 기법을 적용하고, LinearLayout의 방향이 horizontal 가로 방향이면, layout_width를 모두 0dp로 설정한 다음에, weight을 설정해야 한다. layout_weight의 값은 절대적 수치가 아닌 상대적으로 계산되는 값으로, 모든 weight 값을 더해서 등분으로 계산된다. 예를 들면 1, 2, 1로 지정하였으므로 전체를 더하면 4가 되고, 첫번째 TextView가 1/4, 두번째 TextView가 2/4, 세번째 TextView가 1/4의 여백을 차지하게 된다.

Resources: 강샘의 안드로이드 프로그래밍