



레이아웃 익히기

≡ 태그

1. 레이아웃 속성

- orientation
 - vertical, horizontal
- gravity
 - 자신(레이아웃)에게 포함된 자식(위젯)을 어디에 위치시킬지를 결정
 - left, right, center, top, bottom → right | bottom
- layout_gravity
 - 자신(위젯)의 위치를 부모(레이아웃)의 어디에 위치시킬지를 결정
- padding
- layout_weight
 - 레이아웃이 전체 화면에서 차지하는 공간의 가중값을 설정, 중복 레이아웃일때 사용
- baselineAligned (true/false)
 - 크기가 다른 위젯들을 보기 좋게 정렬

2. 레이아웃 종류



- LinearLayout (= 선형 레이아웃)
 - 레이아웃의 왼쪽 위부터 아래쪽 또는 오른쪽으로 차례로 배치
- RelativeLayout (= 상대 레이아웃)
 - 위젯 자신이 속한 레이아웃의 상하좌우 위치를 지정하여 배치하거나 다른 위젯으로부터 상대적인 위치를 지정
- TableLayout
 - 행과 열의 개수를 지정한 테이블 형태로 위젯을 배열
- GridLayout
 - 테이블레이아웃과 비슷하지만 행 또는 열을 확장하여 다양하게 배치할 때 더 편리
- FrameLayout
 - 위젯을 왼쪽 위에 일률적으로 겹쳐서 배치하여 중복되어 보이는 효과를 나타냄
 - 여러 개의 위젯을 배치한 후 상황에 따라서 필요한 위젯을 보이는 방식에 주로 활용

▼ 직접 풀어보기 5-1) gravity, layout_gravity

```
<Button
    android:text="버튼1"
    android:gravity="right|top"
    android:layout_gravity="center|top" />

<Button
    android:text="버튼2"
    android:gravity="left|center"
    android:layout_gravity="left|center" />

<Button
    android:text="버튼3"
    android:gravity="right|bottom"
    android:layout_gravity="right" />
```



▼ 직접 풀어보기 5-2) layout_weight

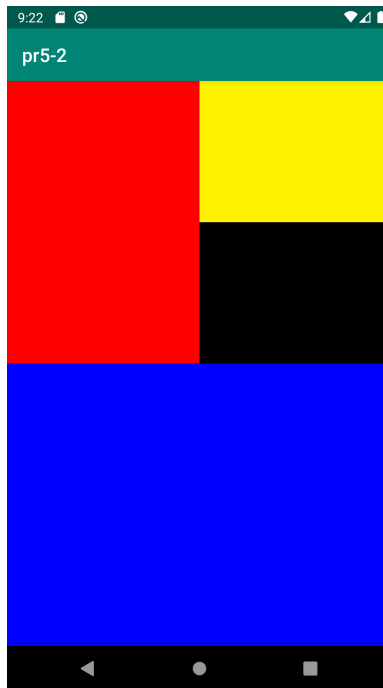
```
<LinearLayout>
  <LinearLayout
    android:orientation="horizontal"
    android:layout_weight="1"
    android:background="#FF0000">

    <LinearLayout
      android:orientation="vertical"
      android:layout_weight="1">
    </LinearLayout>

    <LinearLayout
      android:orientation="vertical"
      android:layout_weight="1">
      <LinearLayout
        android:layout_weight="1"
        android:background="#fff200">
      </LinearLayout>
      <LinearLayout
        android:layout_weight="1"
        android:background="#000000">
      </LinearLayout>
    </LinearLayout>
  </LinearLayout>

  <LinearLayout
    android:background="#0000FF"
    android:layout_weight="1">
```

```
</LinearLayout>
</LinearLayout>
```



▼ 직접 풀어보기 5-3) XML파일 없이 Kotlin 코드만으로 작성

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        //setContentView(R.layout.activity_main)

        val params = LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,
            LinearLayout.LayoutParams.MATCH_PARENT)

        val baseLayout = LinearLayout(this)
        baseLayout.orientation= LinearLayout.VERTICAL
        setContentView(baseLayout, params)

        var edit = EditText(this)
        edit.hint= "입력하세요"
        baseLayout.addView(edit)

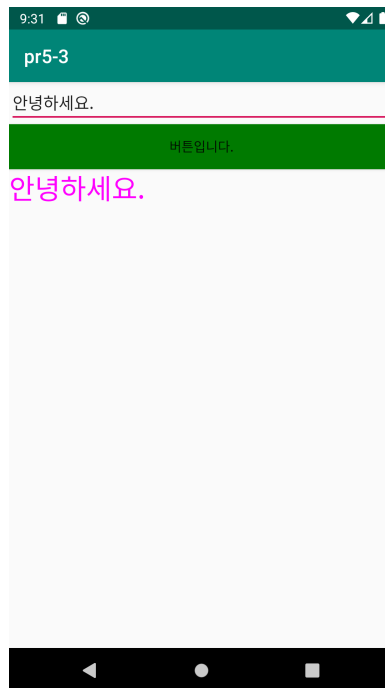
        var btn = Button(this)
        btn.text= "버튼입니다."
        btn.setBackgroundColor(Color.rgb(0, 122, 0))
        baseLayout.addView(btn)
    }
}
```

```

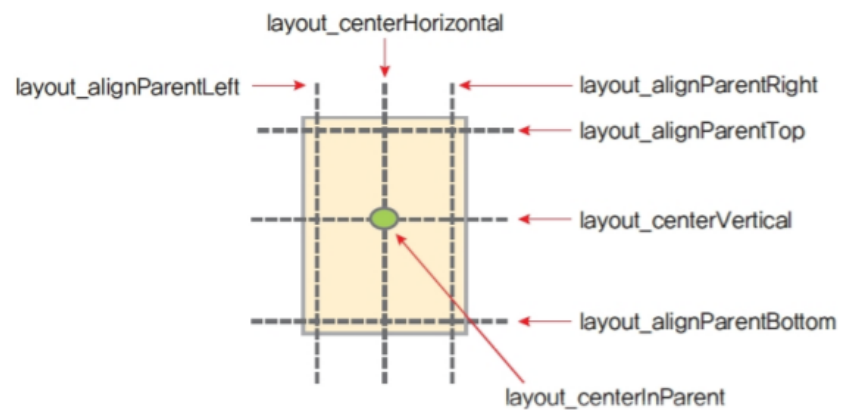
var tes = TextView(this)
tes.textSize= 30F
tes.setTextColor(Color.MAGENTA)
baseLayout.addView(tes)

btn.setOnClickListener{
    var sentence : String = edit.text.toString()
    tes.setText(sentence)
}
}
}

```

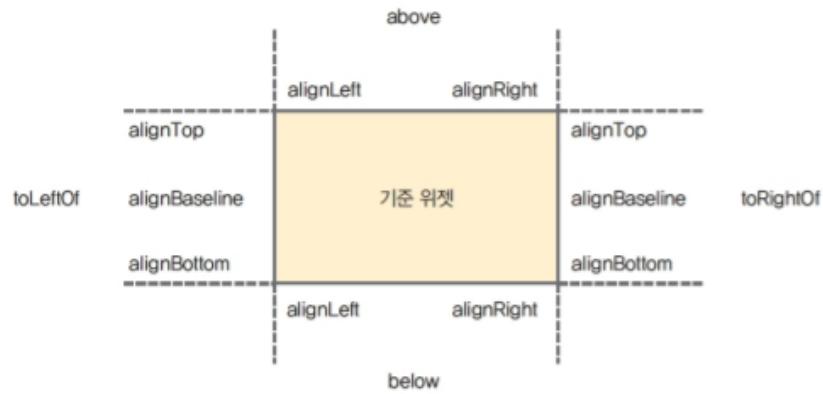


3. RelativeLayout 상화좌우 배치



ex) android:layout_centerInParent="true"

4. 다른 위젯의 상대 위치에 배치



▼ 예제 5-11) RelativeLayout

```
<Button
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:id="@+id/baseBtn1"
    android:text="기준1" />

<Button
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:id="@+id/baseBtn2"
    android:text="기준2" />

<Button
    android:layout_above="@+id/baseBtn2"
    android:layout_toRightOf="@+id/baseBtn1"
    android:text="1번"/>

<Button
    android:layout_alignParentRight="true"
    android:layout_below="@+id/baseBtn1"
    android:text="2번"/>
```



▼ 직접 풀어보기 5-4) 1. 중복 LinearLayout, 2. RelativeLayout

```
<LinearLayout>
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    ...
  </LinearLayout>

  <LinearLayout
    android:orientation="horizontal"
    android:gravity="right">

    <Button
      android:text="입력" />

  </LinearLayout>
</LinearLayout>
```

```
<RelativeLayout>

  <TextView
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:id="@+id/text"
    android:text="전화번호" />

  <EditText
```

```

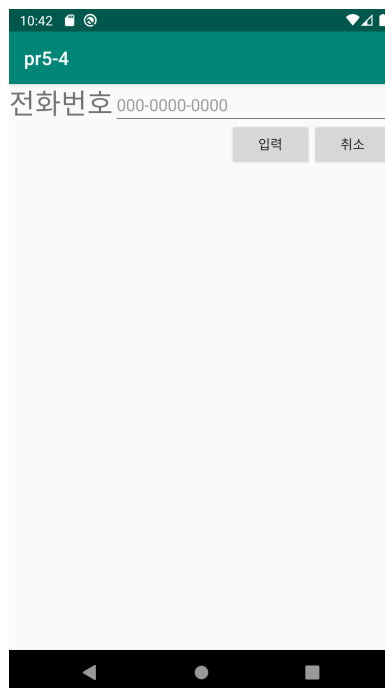
        android:hint="000-0000-0000"
        android:id="@+id/edit"
        android:layout_toRightOf="@id/text" />

<Button
    android:id="@+id/btn1"
    android:layout_toLeftOf="@+id/btn2"
    android:layout_alignBaseline="@+id/btn2"
    android:text="입력"/>

<Button
    android:id="@+id/btn2"
    android:layout_alignParentRight="true"
    android:layout_below="@id/text"
    android:text="취소"/>

</RelativeLayout>

```



5. TableLayout 속성

- layout_span
 - 열을 합쳐서 표시
- layout_column
 - 지정된 열에 현재 위젯을 표시

- stretchColumns
 - 지정된 열의 너비를 늘리기
 - stretchColumns="*" - 각 셀을 모두 같은 크기로 확장하여 전체 화면이 꽉차는 효과

▼ Project5-2) 배열 활용 문제

▼ XML 코드 - TableLayout

```
<TableLayout>
  <TableRow>
    <EditText
      android:id="@+id/Edit1"
      android:hint="숫자1 입력"
      android:layout_span="5"/>
  </TableRow>
  <TableRow>
    <Button
      android:id="@+id/Btn0"
      android:text="0" />
    <Button
      android:id="@+id/Btn1"
      android:text="1" />
    <Button
      android:id="@+id/Btn2"
      android:text="2" />
  </TableRow>

  <TableRow>
    <Button
      android:id="@+id/BtnAdd"
      android:layout_margin="5dp"
      android:layout_span="5"
      android:text="더하기" />
  </TableRow>
  <TableRow>
    <TextView
      android:id="@+id/text1"
      android:layout_margin="5dp"
      android:layout_span="5"
      android:text="계산 결과 : "
      android:textColor="#FF0000"
      android:textSize="20dp"/>
  </TableRow>
</TableLayout>
```

▼ Kotlin 코드

- arrayOf, isFocused, getText

```

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        lateinit var num1 : String
        lateinit var num2 : String
        var result : Int

        var numBtnIDs = arrayOf(Btn0, Btn1, Btn2, Btn3, Btn4, Btn5, Btn6, Btn
7, Btn8, Btn9)

        for (i in 0..numBtnIDs.size-1 step 1){
            numBtnIDs[i].setOnClickListener{
                if (Edit1.isFocused == true) {
                    num1 = Edit1.text.toString() + numBtnIDs[i].getText().toS
tring()

                    Edit1.setText(num1)
                } else if (Edit2.isFocused == true) {
                    num2 = Edit2.text.toString() + numBtnIDs[i].getText().toS
tring()

                    Edit2.setText(num2)
                } else {
                    Toast.makeText(applicationContext, "먼저 에디트텍스트를 선택하세
요", Toast.LENGTH_SHORT).show()
                }
            }
        }

        BtnAdd.setOnClickListener{view, motionEvent->
            num1 = Edit1.text.toString()
            num2 = Edit2.text.toString()
            result = Integer.parseInt(num1) + Integer.parseInt(num2)
            text1.text= "계산 결과 : " + result.toString()
            false
        }
    }
}

```

▼ 결과



(+) `var numButtons = ArrayList<Button>(10)`

6. GridLayout 속성

- rowCount
- columnCount
- orientation
- layout_row
- layout_column
- layout_rowSpan
- layout_columnSpan
- layout_gravity
 - 행 또는 열이 확장 되었을 때 위젯을 꽉 채우는 효과
 - fill, fill_vertical, fill_horizontal

▼ 직접 풀어보기 5-5) GridLayout

▼ XML 코드

```

<GridLayout>
    <EditText
        android:id="@+id/Edit1"
        android:layout_column="0"
        android:layout_row="0"
        android:layout_columnSpan="5"
        android:layout_gravity="fill_horizontal"
        android:hint="숫자1 입력" />

    <Button
        android:id="@+id/Btn0"
        android:layout_column="0"
        android:layout_row="2"
        android:text="0" />

    <Button
        android:id="@+id/Btn1"
        android:layout_column="1"
        android:layout_row="2"
        android:text="1" />

    <Button
        android:id="@+id/Btn2"
        android:layout_column="2"
        android:layout_row="2"
        android:text="2" />

    <Button
        android:id="@+id/BtnAdd"
        android:layout_column="0"
        android:layout_row="4"
        android:layout_columnSpan="5"
        android:layout_gravity="fill_horizontal"
        android:text="더하기" />

    <TextView
        android:id="@+id/Text1"
        android:layout_column="0"
        android:layout_row="8"
        android:layout_columnSpan="5"
        android:layout_gravity="fill_horizontal"
        android:text="계산 결과 : "
        android:textColor="#FF0000"
        android:textSize="20dp"/>

</GridLayout>

```

7. FrameLayout 속성

- foreground

- 프레임레이아웃의 전경 이미지를 지정
- foregroundGravity
 - 전경 이미지의 위치를 지정
 - fill, right, left, top, bottom

ex) android:foreground="@drawable/dog"

android:foregroundGravity="center|fill_horizontal"