

모바일 프로그래밍 실습 보고서

학번	201411203	이름	박소영
순서	4	날짜	2018.03.22 목요일
강의자료	02 - 레이아웃		
주제	여러가지 레이아웃들 ...		

1. GridLayout : 행과 열이 존재함

- 각 칸도 view처럼 볼 수 있다. layout_gravity를 지정해 내부 정렬도 가능하다.
- orientation : 행부터 채울지 열부터 채울지 정한다.
- default : 행과 열이 같은 크기로 만들어짐.
- 특정 view의 위치를 row, column을 지정해서 원하는 칸에 추가할 수 있다.
- 합치기 : columnSpan, rowSpan
- 값을 가지고 오는 것이 주 기능인데, getText()으로 가져온 후 toString()으로 변환해줘야 함.

2. RelativeLayout : 뷰의 위치를 상대적으로 지정함

- 다른 뷰를 기준으로, 상대적으로 위치를 지정한다.

3. FrameLayout : 자식 뷰를 배치하는 규칙이 없다.

- 모든 자식 뷰에 해당하는 것이 좌상단에 카드처럼, 똑같은 위치에 쌓인다.
- 결과적으로 마지막에 있는 뷰가 보이게 된다.
- 자식 뷰의 visibility 속성을 조정해서 보였다 감췄다 하면 된다.
- foreground 속성 : 자식 뷰의 위쪽에 얹어주는 이미지 지정. 반투명하게 할 수 있다.

4. AbsoluteLayout : 절대적인 좌표값 지정. 근데 쓰지 말라고 함.

5. ConstraintLayout : 자식 뷰를 배치하는 규칙이 없다.

- relative + linear 의 장점을 결합한다.
- GUI 상태에서 layout 배치를 하자! XML를 편집하지 않고, 화면에서....
- 기존 linear에서는 중첩되다보니까 hierachy가 너무 깊어지고, 성능상 좋지 않다.
- 뷰를 가능하면 중첩되지 않게, 복잡한 레이아웃도 단순하게 만들어보자는 취지.
- handle : 위치(Side)와 크기(Resize), 글자의 위치 기준 (Baseline)를 바꿔줄 수 있다.
- 자식 표시 (AutoConnected) : 인접한 친구를 연결시켜주겠다
- infer constraints : 배치해 놓은 것을 기준으로 constraints를 자동으로 맺어주겠다.
- **뷰에 하나 이상의 수평 및 수직 제약 조건을 부여해주지 않으면 (0, 0)으로 가버림!**

※ layout_weight 지정할 때는 width나 height 중 선택해서 0dp로 설정해야 한다.

Layout : activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    ...
    android:orientation="vertical">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1.25">

        <ImageView
            android:id="@+id/body"
            android:src="@drawable/body"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:visibility="visible"/>
        ...

    </FrameLayout>

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="0.75"
        android:rowCount="5"
        android:columnCount="2"
        android:layout_gravity="center">

        <CheckBox
            android:id="@+id/hatCheckBox"
            android:text="hat" />
        ...

    </GridLayout>

</LinearLayout>
```

java 소스코드 : MainActivity.java

```
import ...

public class MainActivity extends AppCompatActivity {
    static final int CHECKBOX_COUNT = 10;
    CheckBox[] checkBoxes;
    ImageView imageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        init();
    }

    public void init() {
        checkBoxes = new CheckBox[CHECKBOX_COUNT];
        checkBoxes[0] = (CheckBox)findViewById(R.id.hatCheckBox);
        checkBoxes[1] = (CheckBox)findViewById(R.id.armsCheckBox);
        checkBoxes[2] = (CheckBox)findViewById(R.id.earsCheckBox);
        checkBoxes[3] = (CheckBox)findViewById(R.id.eyebrowsCheckBox);
        checkBoxes[4] = (CheckBox)findViewById(R.id.eyesCheckBox);
        checkBoxes[5] = (CheckBox)findViewById(R.id.noseCheckBox);
        checkBoxes[6] = (CheckBox)findViewById(R.id.glassesCheckBox);
        checkBoxes[7] = (CheckBox)findViewById(R.id.mustacheCheckBox);
        checkBoxes[8] = (CheckBox)findViewById(R.id.mouthCheckBox);
        checkBoxes[9] = (CheckBox)findViewById(R.id.shoesCheckBox);

        for (int i=0; i<CHECKBOX_COUNT; i++) {
            // 변화가 감지될때마다 이 리스너가 호출된다.
            // 체크 상태가 변했을 때 호출됨!
            // 어떤 체크박스가 체크되었는지 해제되었는지 알 수 있다.

            checkBoxes[i].setOnCheckedChangeListener(
                new CompoundButton.OnCheckedChangeListener() {
```

```

@Override
public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
    int imgID = getResources().getIdentifier(
        buttonView.getText().toString(), "id", getPackageName());
    // getResource() : 리소스들을 가져온다.
    // getIdentifier(찾을 아이디, 뭘 찾을 건지 (id), 패키지 이름)
    // 체크박스 text에 아이디가 저장되어 있으므로 그걸 가지고 온다.
    imageView = (ImageView) findViewById(imgID);

    if (isChecked) {
        imageView.setVisibility(View.VISIBLE);
    } else {
        imageView.setVisibility(View.INVISIBLE);
    }
}
});
}
}
}
}

```

실행 결과

