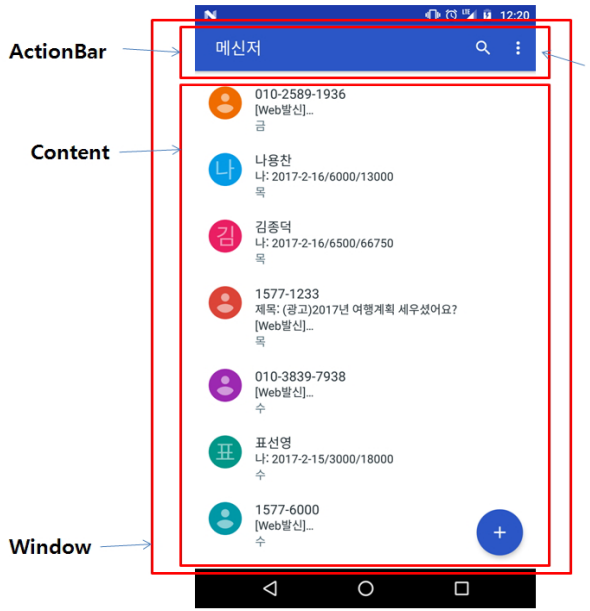
12장. ActionBar와 메뉴

12.1 ActionBar

12.1.1. ActionBar 구성

- ActionBar 구성을 개발자가 특별하게 하지 않으면 기본 타이틀 문자열이 나오지만, App Icon, View Control, Action Button, Overflow Menu 등이 나오도록 구성 가능



12.1 ActionBar

- 테마 지정으로 안 보이게 설정

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="windowNoTitle">true</item>
    <item name="windowActionBar">false</item>
</style>
```

- ActionBar 객체를 프로그램에서 획득

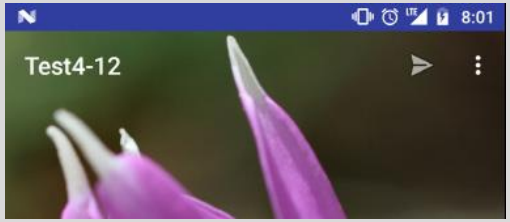
```
actionBar=getSupportActionBar();
```

```
public void onClick(View v) {
    if(v==btn1){
        actionBar.show();
    }else if(v==btn2){
        actionBar.hide();
    }
}
```

12.1 ActionBar

- ActionBar가 Content 영역 위에 떠 있는듯하게 처리

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <item name="colorPrimary">#00000000</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="windowActionBarOverlay">true</item>
</style>
```



12.1.2. 표시 옵션

- ActionBar를 위한 label을 설정

```
<activity android:name=".MainActivity" android:label="상세 보기">
```

- setDisplayHomeAsUpEnabled(boolean showHomeAsUp): 아이콘을 Up 이미지로 표시 설정
- setDisplayShowCustomEnabled(boolean showCustom): CustomView 표시 설정
- setDisplayShowHomeEnabled(boolean showHome): 홈 아이콘 표시 설정
- setDisplayShowTitleEnabled(boolean showTitle): 타이틀 문자열 표시 설정
- setDisplayUseLogoEnabled(boolean useLogo): 로고 표시 설정

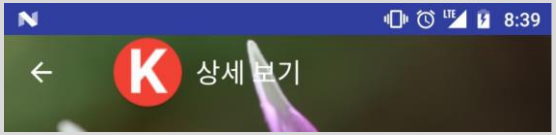
12.1 ActionBar

```
actionBar=getSupportActionBar();
actionBar.setDisplayShowHomeEnabled(true);
actionBar.setDisplayHomeAsUpEnabled(true);
actionBar.setIcon(R.drawable.icon);
```

- setLogo(int resId): 리소스로 로고 지정
- setLogo(Drawable logo): 이미지 객체로 로고 지정
- setTitle(int resId): 리소스로 타이틀 지정
- setTitle(CharSequence title): 문자열로 타이틀 지정
- setSubtitle(int resId): 리소스로 서브 타이틀 지정
- setSubtitle(CharSequence subtitle): 문자열로 서브 타이틀 지정
- setCustomView(int resId): 리소스로 커스텀 뷰 지정
- setCustomView(View view): 뷰로 커스텀 뷰 지정

- HOME_AS_UP

```
public boolean onOptionsItemSelected(MenuItem item) {
    if(item.getItemId()==android.R.id.home){
        Toast t=Toast.makeText(this, "HOME AS UP Click", Toast.LENGTH_SHORT);
        t.show();
        return true;
    }
    return super.onContextItemSelected(item);
}
```

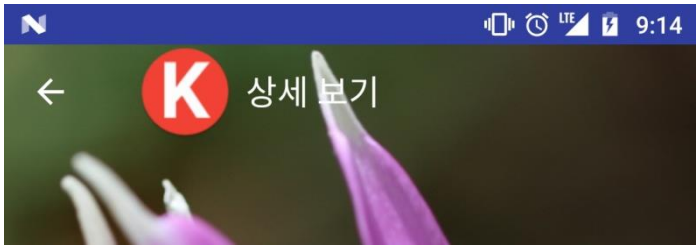


Step by Step 12-1 – ActionBar

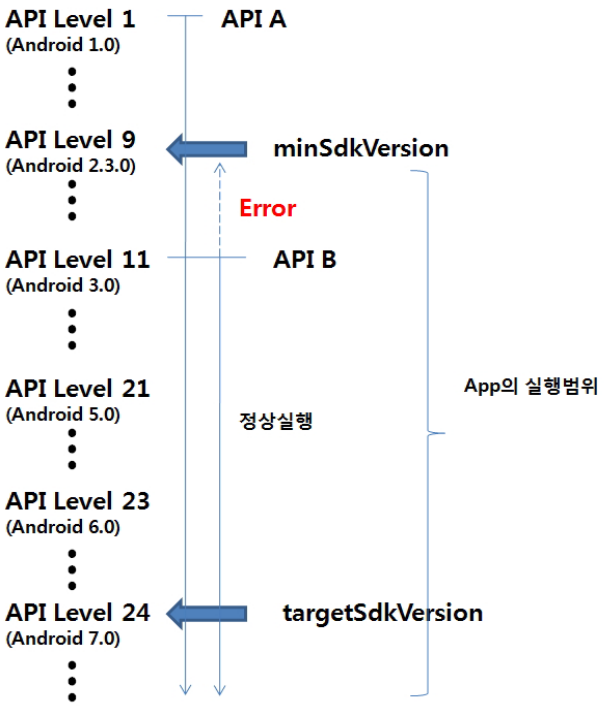
ActionBar을 이용하는 실습

Activity에서 기본으로 ActionBar에 몇몇개의 구성요소를 추가

1. Module 생성
2. 파일 복사
3. styles.xml 추가
4. AndroidManifest.xml 에 label 추가
5. MainActivity 추가
6. 실행



12.2 안드로이드 API Level과 하위 호환성



- 구글의 Support 라이브러리의 도움을 받는다.
- 오픈소스 라이브러리의 도움을 받는다.
- 개발자 코드에서 버전을 직접 식별해서 처리한다.

12.2 안드로이드 API Level과 하위 호환성

Support 라이브러 이용

- Support 라이브러리는 구글의 라이브러리
- 그레이들 파일에 dependencies 연결

```
dependencies {
implementation fileTree(dir: 'libs', include: ['*.jar'])
implementation 'com.android.support:appcompat-v7:26.0.1'
implementation 'com.android.support.constraint:constraint-layout:1.0.2'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'com.android.support.test:runner:1.0.0'
androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.0'
}
```

```
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    //.....
}
```

- 액티비티 를 위한 테마 설정이 appcompat에서 제공하는 테마를 상속받아 정의

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!--중략-->
</style>
```

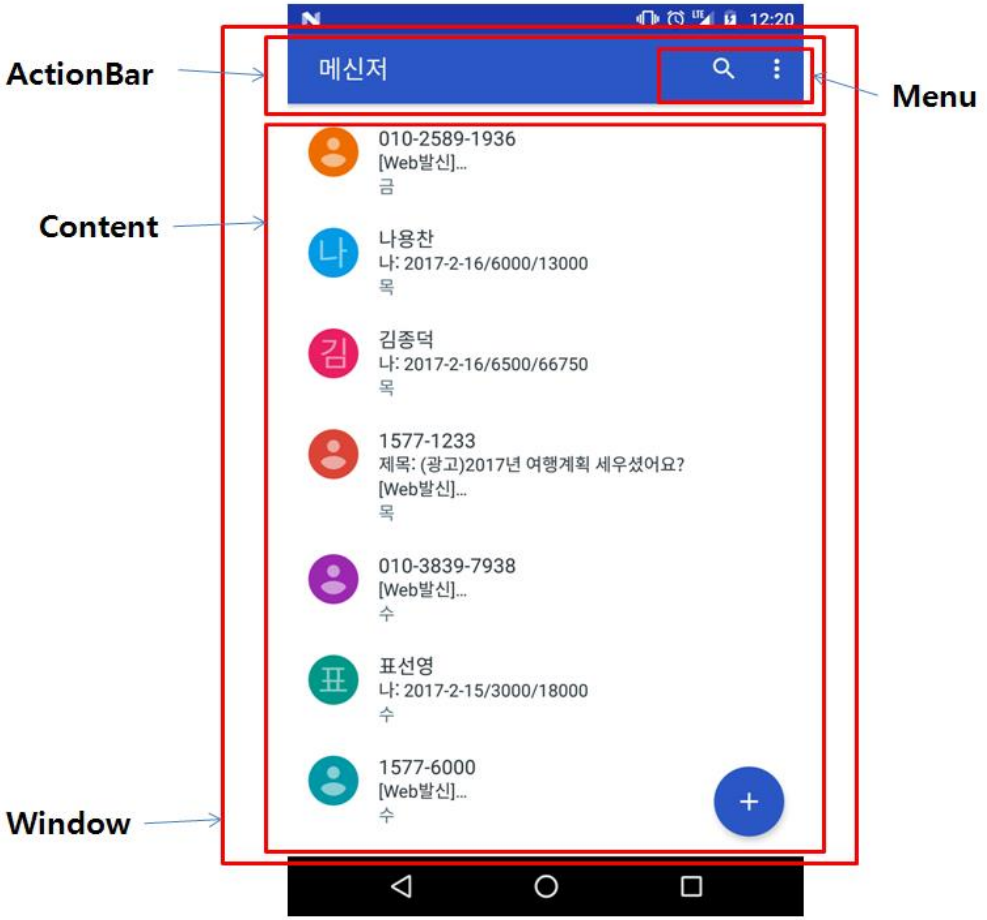
12.2 안드로이드 API Level과 하위 호환성

- 개발자 코드로 버전 확인

```
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP){
    //.....}else {
    //.....}
```

12.3 메뉴

- 메뉴는 액티비티의 구성요소



12.3 메뉴

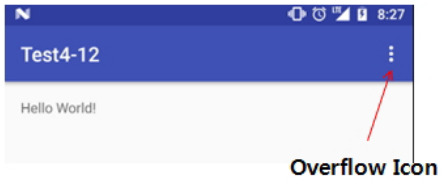
12.3.1. 메뉴 작성 방법

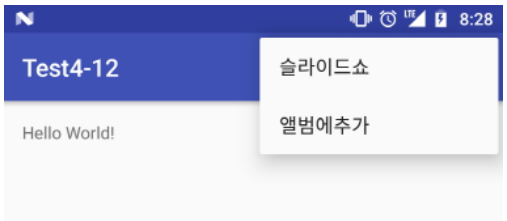
메뉴 구성을 위한 함수

- onCreateOptionsMenu(Menu menu): 메뉴가 만들어질 때 최초 한 번 호출
- onPrepareOptionsMenu(Menu menu): 메뉴가 화면에 보일 때마다 반복 호출

```
public boolean onCreateOptionsMenu(Menu menu) {
    MenuItem item1=menu.add(0,0,0,"슬라이드쇼");
    MenuItem item2=menu.add(0,1,0,"앨범에추가");
    return true;
}
```

- add(CharSequence title)
- add(int groupId, int itemId, int order, int titleRes)
- add(int groupId, int itemId, int order, CharSequence title)





12.3 메뉴

MenuItem의 메뉴 구성을 위한 함수

- setIcon(int iconRes)
- setShortcut(char numericChar, char alphaChar)
- setTitle(CharSequence title)
- setVisible(boolean visible)
- setEnabled(boolean enabled)

```
public boolean onOptionsItemSelected(MenuItem item) {
    if(item.getItemId()==0){
        //.............
    }else if(item.getItemId()==1){
        //.............
    }
    return super.onContextItemSelected(item);
}
```

12.3 메뉴

12.3.2. MenuInflater 활용

- XML을 이용하여 메뉴를 구현
- 메뉴 XML 파일이 저장될 위치는 res 폴더 하위의 menu

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu1"
        android:icon="@drawable/ic_menu_1"
        android:title="선택..." />
</menu>
```

- 액티비티에 메뉴를 적용하려면 자바 코드에서 MenuInflater를 이용

```
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater=getMenuInflater();
    inflater.inflate(R.menu.menu_main, menu);
    return true;
}
```

- 메뉴를 식별할 때R 파일의 int 변수로 식별해서 이벤트를 처리

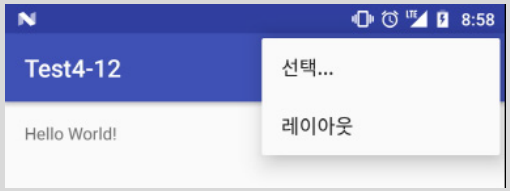
```
public boolean onOptionsItemSelected(MenuItem item) {
    if(item.getItemId()==R.id.menu1){
        //.............
    }
    return super.onContextItemSelected(item);
}
```

12.3 메뉴

12.3.3. 메뉴 다양하게 이용하기

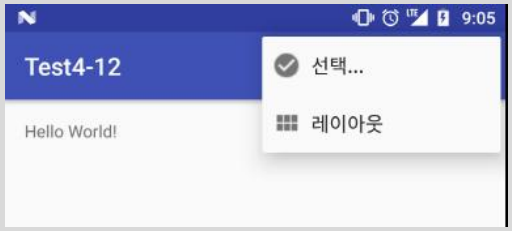
- 아이콘 출력

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu1"
        android:icon="@drawable/ic_menu_1"
        android:title="선택..." />
    <item
        android:id="@+id/menu2"
        android:icon="@drawable/ic_menu_2"
        android:title="레이아웃" />
</menu>
```



- 메뉴는 내부적으로 MenuBuilder에 의해 만들어져 액티비티에 적용
- MenuBuilder 객체를 획득해 아이콘 설정 부분을 조정

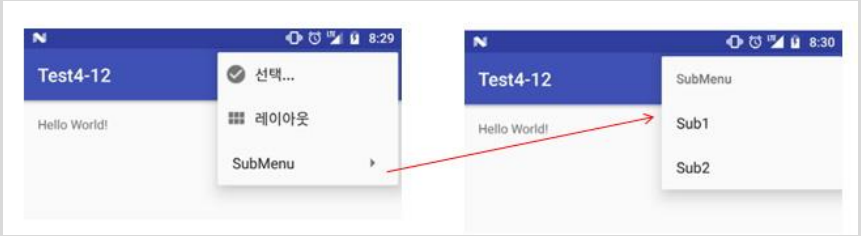
```
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater=getMenuInflater();
    inflater.inflate(R.menu.menu_main, menu);
    if(menu instanceof MenuBuilder){
        MenuBuilder m = (MenuBuilder) menu;
        m.setOptionalIconsVisible(true);
    }
    return true;
}
```



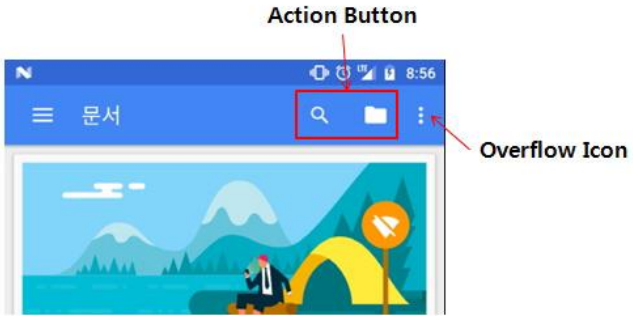
12.3 메뉴

- 서브 메뉴

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
   <item
      android:id="@+id/menu1"
      android:icon="@drawable/ic_menu_1"
      android:title="선택..." />
   <item
      android:id="@+id/menu2"
      android:icon="@drawable/ic_menu_2"
      android:title="레이아웃" />
   <item android:id="@+id/file"
      android:title="SubMenu" >
      <menu>
         <item android:id="@+id/sub1"
            android:title="Sub1" />
         <item android:id="@+id/sub2"
            android:title="Sub2" />
      </menu>
   </item>
</menu>
```



- 액션 버튼



12.3 메뉴

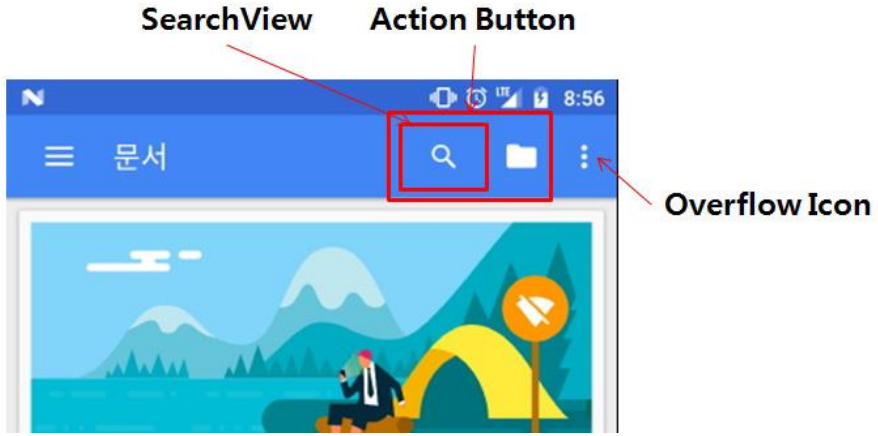
```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto">

   <!--중략-->
   <item
      android:id="@+id/menu4"
      android:icon="@android:drawable/ic_menu_send"
      android:title="Actions"
      app:showAsAction="always"/>
</menu>
```

- never: 기본값이며 showAsAction을 선언하지 않은 경우. 항상 오버플로 메뉴로 구성
- always: 항상 액션 버튼으로 구성
- ifRoom: ActionBar에 아이콘이 올라갈 공간이 있으면 액션 버튼으로 나타내고, 그렇지 않으면 오버플로 메뉴로 나타남. 자 동 판단
- withText: 아이콘과 더불어 문자열이 함께 보이게 하는 설정이지만 문자열이 나타날 공간이 없다면 문자열이 나타나지 않음. 대부분 스마트폰에서는 문자열이 함께 출력되지 않음

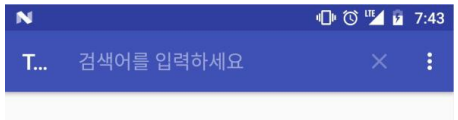
12.3 메뉴

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/menu3"
        app:showAsAction="ifRoom|collapseActionView"/>

</menu>
```

- ActionView
- ActionView는 ActionBar에 제공되는 뷰




```
<item
   android:id="@+id/menu_main_search"
   app:actionViewClass="android.support.v7.widget.SearchView"
   android:icon="@android:drawable/ic_menu_search"
   app:showAsAction="always"
   android:title="Search"/>
```



12.3 메뉴

- SearchView 객체 획득

```
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater=getMenuInflater();
    inflater.inflate(R.menu.menu_lab2, menu);
    MenuItem menuItem=menu.findItem(R.id.menu_main_search);
    searchView=(SearchView) MenuItemCompat.getActionView(menuItem);
    searchView.setQueryHint(getResources().getString(R.string.query_hint));
    searchView.setOnQueryTextListener(queryTextListener);
    return true;
}
```

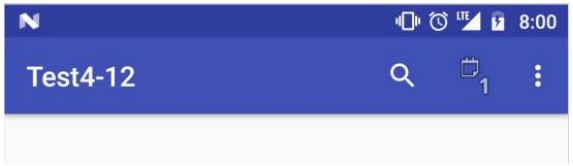
- SearchView 이벤트

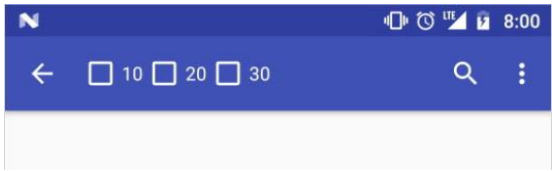
```
private SearchView.OnQueryTextListener queryTextListener = new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String query) {
        //.....
        return false;
    }
    @Override
    public boolean onQueryTextChange(String newText) {
        // TODO Auto-generated method stub
        return false;
    }
};
```

12.3 메뉴

- actionLayout
- 개발자가 임의의 뷰를 ActionView로 설정

```
<item
    android:id="@+id/menu3"
    app:showAsAction="ifRoom|collapseActionView"
    android:title="Check"
    android:icon="@android:drawable/ic_menu_day"
    app:actionLayout="@layout/actionview_check"/>
```





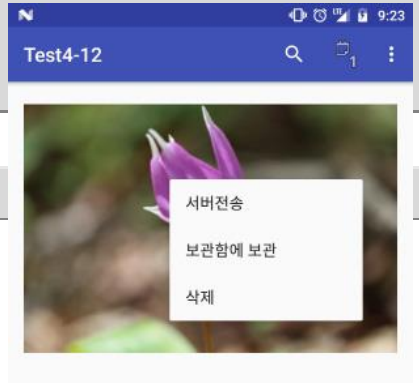
- ContextMenu
- 화면에 출력되는 특정 뷰와 연결되어서 뷰를 오래 누르면 보이는 메뉴

```
public void onCreateContextMenu(ContextMenu menu, View v,
                        ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.add(0,0,0,"서버전송");
    menu.add(0,1,0,"보관함에 보관");
    menu.add(0,2,0,"삭제");
}
```

12.3 메뉴

```
public boolean onContextItemSelected(MenuItem item) {
    //.....................
    return true;
}
```

```
registerForContextMenu(imageView);
```



Step by Step 12-2 - Menu

메뉴들을 테스트

•메뉴 구성을 목적으로 실습이 진행되며 메뉴 이벤트 처리는 단순 Toast로 처리

1. Activity 생성
2. 파일 복사
3. menu 폴더 생성
4. menu 파일 생성
5. menu_lab2.xml 파일 작성
6. Lab12_2Activity 작성
7. Lab12_2Activity.java 실행