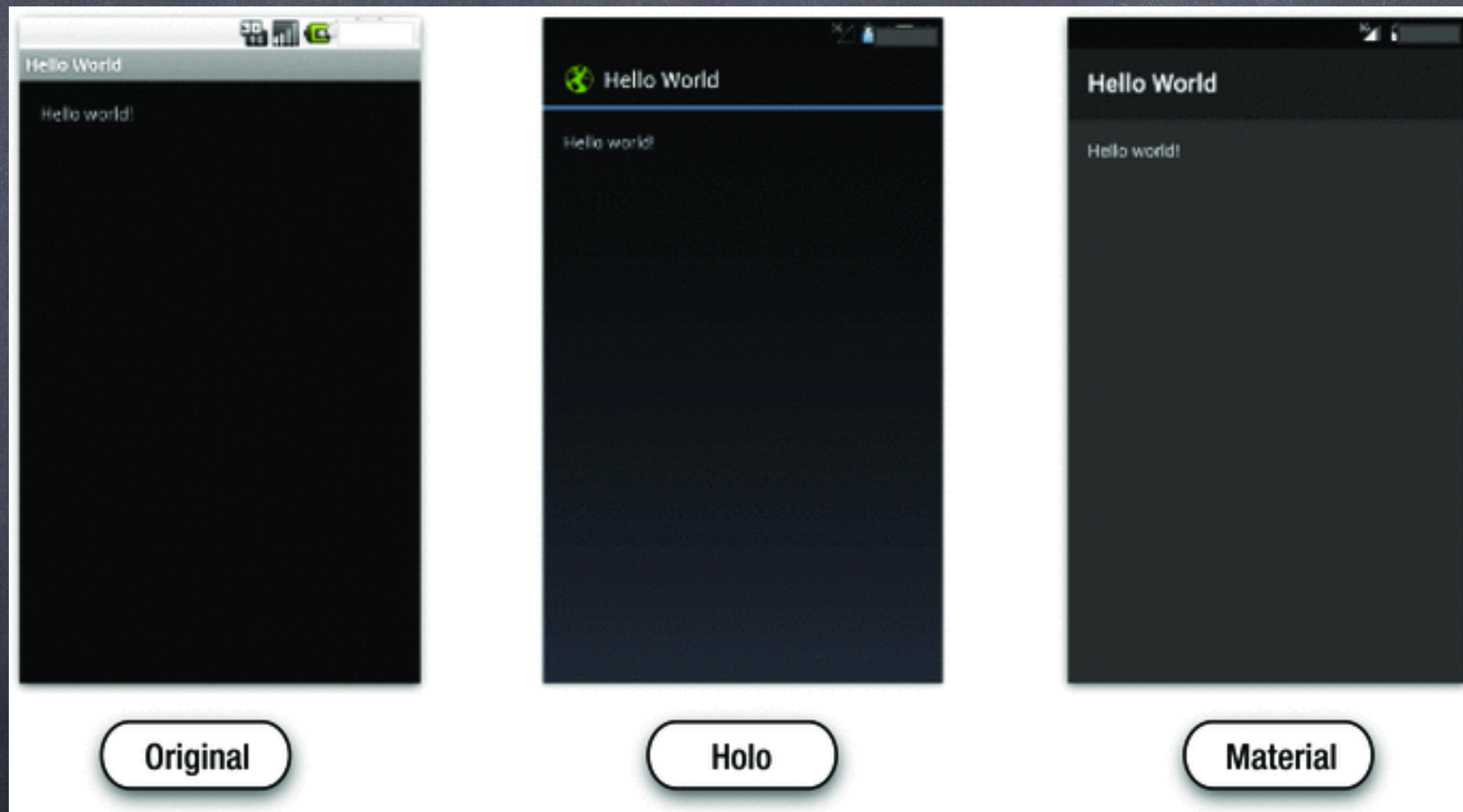


UI 요소

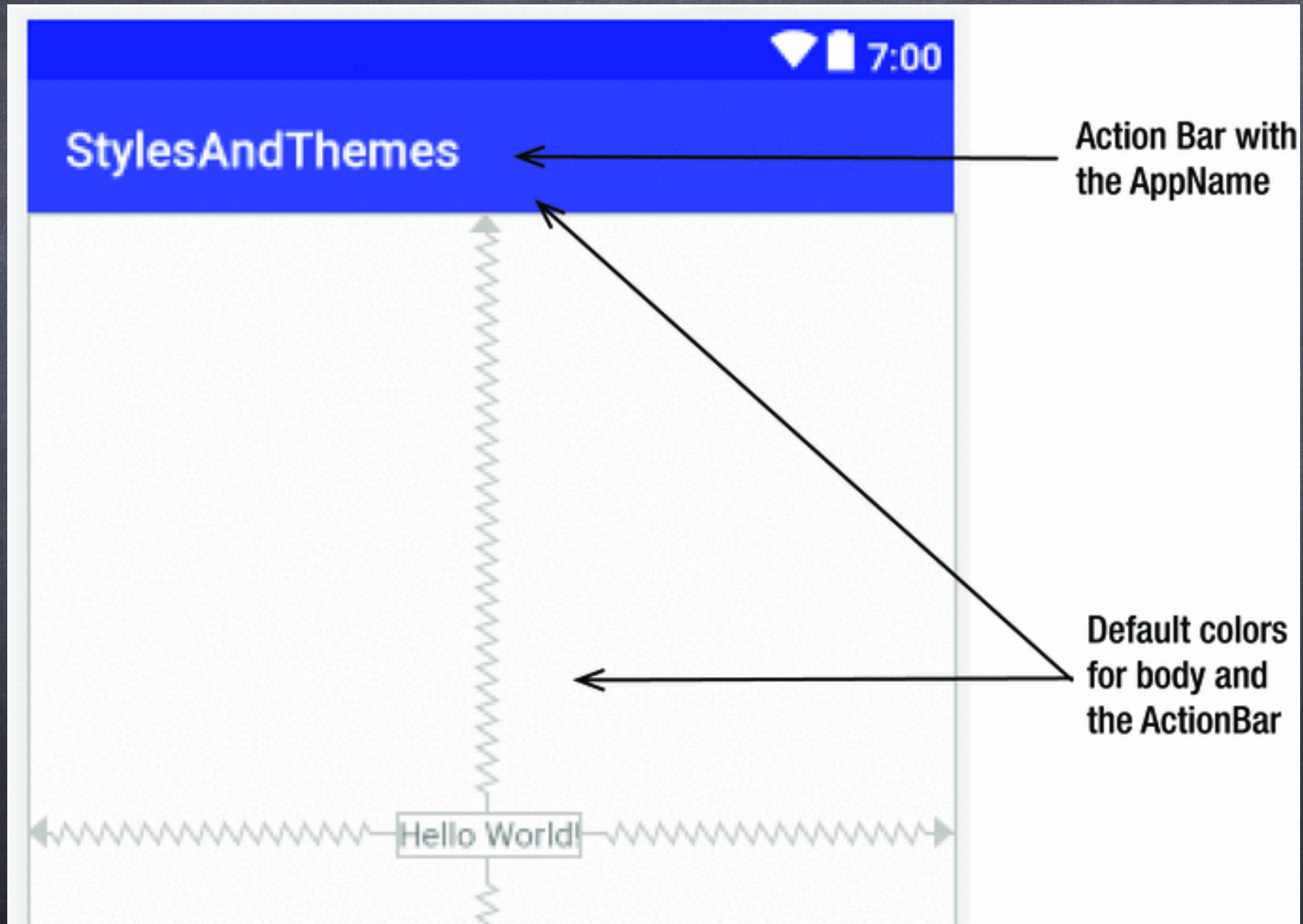
코틀린 안드로이드 앱 개발

강사: 배정만

안드로이드 디자인 변화



테마 및 색상



매니페스트 파일

```
<?xml version="1.0" encoding="utf-8"?>
<manifest package="com.ted.stylesandthemes"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application></manifest>
```

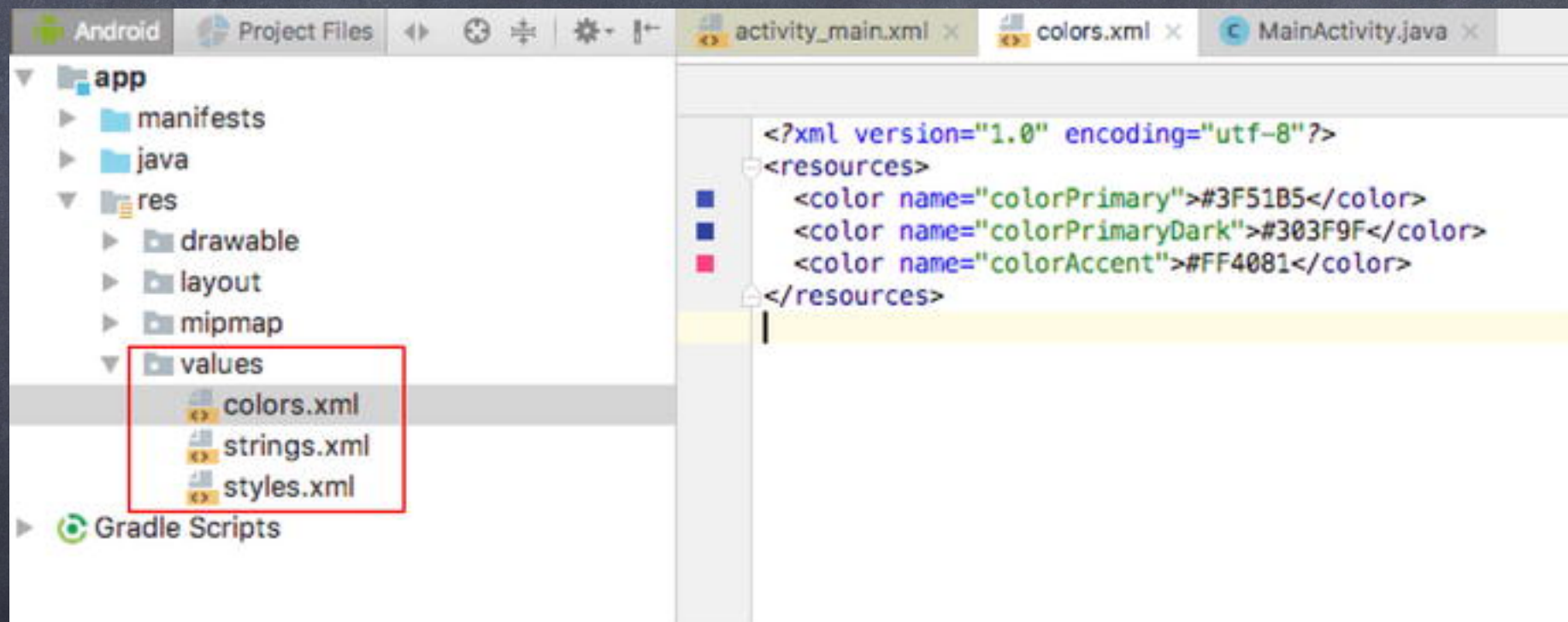
- app > manifests > AndroidManifest
- @string: app/res/values/strings.xml
- @style: app/res/values/styles.xml

app/res/values/styles.xml

```
<resources>
  <!-- Base application theme. -->
  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
  </style>
</resources>
```

- 기본 응용 프로그램 테마
- 기본 테마의 색상 선택
- 앱 바, 버튼의 대표 색상을 지정할 수 있음

app/res/values/colors.xml



app/res/values/colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

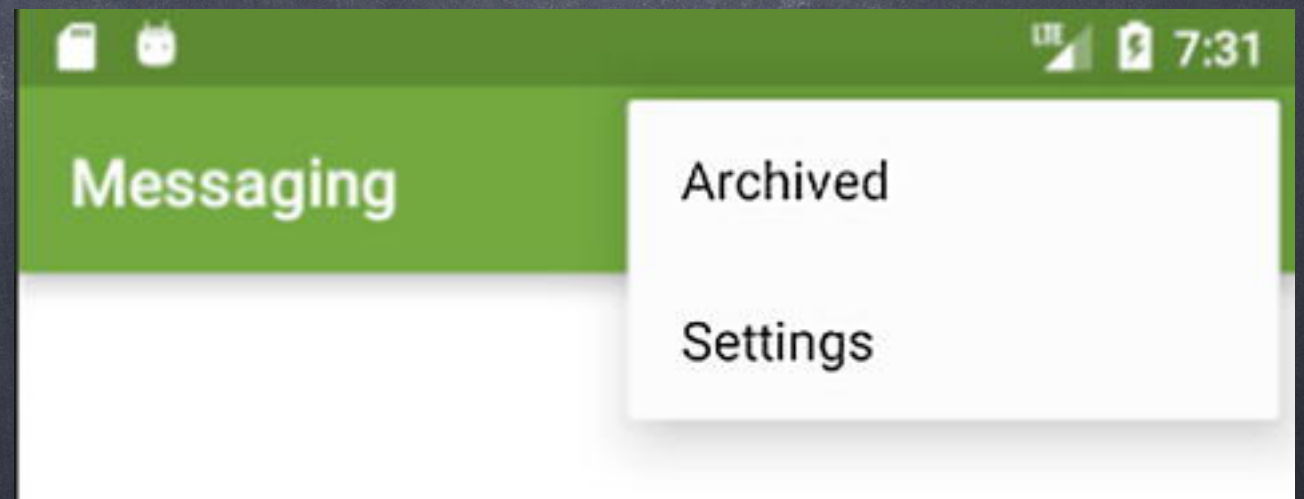
- 색상의 16 진수 값
- 편집기 좌측에 실제 구현되는 색상 표시됨
- 기본, 어두운 기본 색상, 강조 색상을 기본으로 정의

Materialpalette

<http://www.materialpalette.com/>

AppBar

- 메뉴는 UI 디자인에서 매우 중요
- 메뉴 시스템을 통해 사용자 기능을 사용 할 수 있음
- API 11(3.0) 이전에는 하드웨어 버튼 사용
- 현재는 화면에 액션버튼을 구성하는 형태로 변경됨



/app/res/values/menu/ main_menu.xml

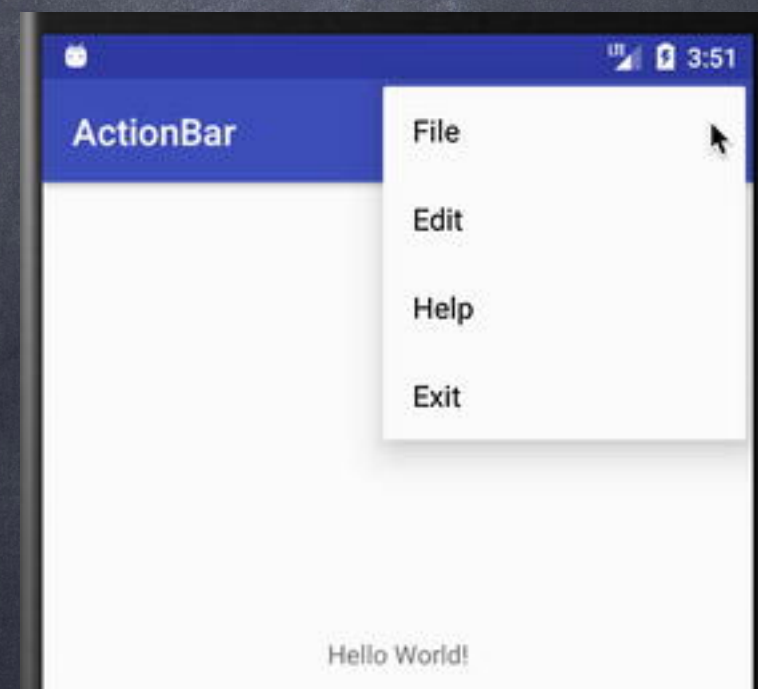
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menuFile"
          android:title="@string/menuFile"/>
    <item android:id="@+id/menuEdit"
          android:title="@string/menuEdit"/>
    <item android:id="@+id/menuHelp"
          android:title="@string/menuHelp"/>
    <item android:id="@+id/menuExit"
          android:title="@string/menuExit"/>
</menu>
```

- @+id 는 R 클래스에서 접근할 수 있는 아이디
- title도 R 클래스에서 접근 가능
- title 에 들어가는 @string 은 string.xml에 정의 됨

메뉴 아이템 추가하기

```
override fun onCreateOptionsMenu(menu: Menu): Boolean {  
    val inflater = menuInflater  
    inflater.inflate(R.menu.main_menu, menu)  
    return true  
}
```

- menuInflater: XML 파일을 Menu 객체로 인스턴스화하는 데 사용
- 성능상의 이슈로 컴파일된 R 클래스 파일에 의존



코드로 메뉴 추가하기

```
override fun onCreateOptionsMenu(menu: Menu?): Boolean {  
    menu?.add("Web")  
    menu?.add("Map")  
    menu?.add("Phone number")  
    menu?.add("Go to ClickEventActivity")  
    return super.onCreateOptionsMenu(menu)  
}
```

- xml을 정의하고 로드하는 것과 동일

메뉴 선택 이벤트 오버로딩

```
override fun onOptionsItemSelected(item: MenuItem?): Boolean {  
    when (item?.toString()) {  
        "Web" -> {  
        }  
        "Map" -> {  
        }  
        "Phone number" -> {  
        }  
        "Go to ClickEventActivity" -> {  
            val intent = Intent(this, Activity::class.java)  
            startActivity(intent)  
        }  
    }  
}
```


실습 - 메뉴 아이템

- /app/res/values/menu/
main_menu.xml 에 메뉴 아이템을 생성
 - 웹, 지도, 전화 메뉴 아이템 생성
- 메뉴를 액티비티에 로드하고, 이벤트 처리
 - 웹, GPS, tel: 을 사용 (지난시간 참고)

부분 UI (Fragments)

- 안드로이드 초기에는 한 액티비티가 한 UI를 담당
- 태블릿과 고해상도 스크린이 많아지면서 부분 UI(Fragment)가 추가됨(2011 년 API 11)
- 작은 액티비티
- 이벤트 처리를 통해 부분 UI 를 숨기거나 표시할 수 있다 (Intent 통하지 않음)
- 각 부분UI(Fragment) 마다 레이아웃 XML을 가짐

실습 - Fragment

1. 메인 액티비티를 만듬 (빈 액티비티)
2. Fragment 클래스를 만들고 임의의 레이아웃 XML 을 만듬 (Book, BookDescription)
3. 레이아웃 디자인 모드를 이용해 부분UI를 디자인 함
4. Fragment 클래스에서 onCreateView 를 만들고 XML 파일을 inflate 함
5. 메인 액티비티 레이아웃에 부분UI를 추가함

실습 심화 - Fragment

1. 메인액티비티 가로 레이아웃 파일 생성
2. 부분UI 를 가로에 맞춰 배치
3. 각 부분UI 에 디자인 요소 적용
 - Book: 제목 라디오 버튼
 - BookDescription: 책 줄거리
4. 부분UI 이벤트 처리를 통해 액티비티에 전달하기
 - Coordinator 인터페이스 생성하고 onBookChanged 함수 구현하기