

모바일 프로그래밍 입문

(2019년도 1학기)

메뉴 구현하기

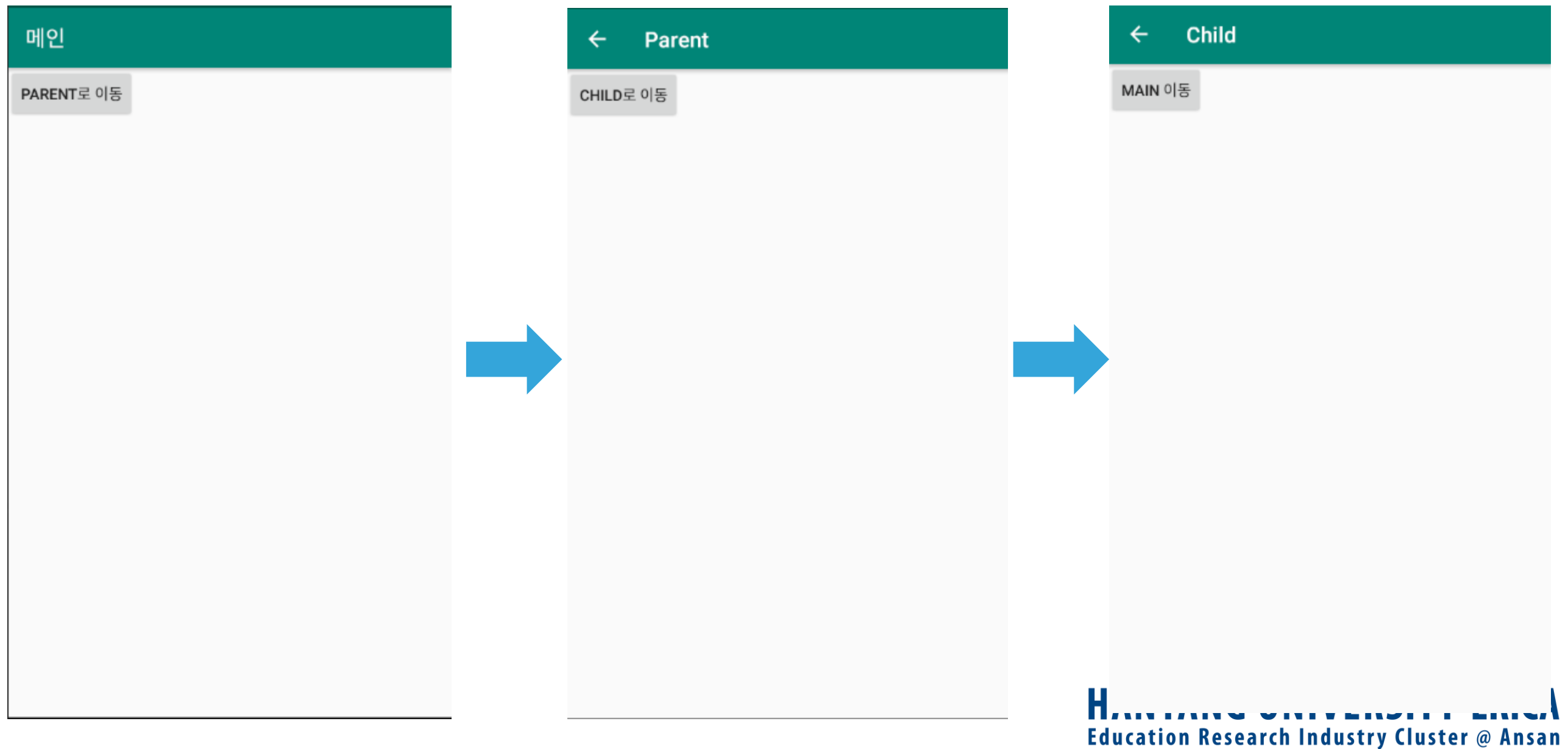
액션바(업 네비게이션), 메뉴

업네비게이션

- ▶ 액티비티가 여러 개 있을 때 액티비티간 계층을 설정하고 업네비게이션을 통해 계층간 뒤로가기 만들기
- ▶ 실습 파일(java 파일과 관련 xml 파일)
 - ▶ MainActivity
 - ▶ ParentActivity
 - ▶ ChildActivity

업네비게이션

▶ 실행화면



업네비게이션

- ▶ main.xml, parent.xml, child.xml 에서 이동 Button 생성
 - ▶ main.xml Button 생성 => 클릭이벤트 moveParent 입력 후 메소드 생성

```
<Button
    android:text="parent로 이동"
    android:onClick="moveParent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

- ▶ parent.xml Button 생성 => 클릭이벤트 moveChild 입력 후 메소드 생성

```
<Button
    android:text="Child로 이동"
    android:onClick="moveChild"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

- ▶ child.xml Button 생성 => 클릭이벤트 moveMain 입력 후 메소드 생성

```
<Button
    android:text="Main 이동"
    android:onClick="moveMain"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

업네비게이션

- ▶ MainActivity, ParentActivity, ChildActivity.java 에서 메소드 코드 작성

- ▶ main.xml 에 의해 생성된 MainActivity.java에 moveParent 메소드 코드 작성

```
public void moveParent(View view) {
    Intent intent = new Intent( packageContext: this, ParentActivity.class);
    startActivity(intent);
}
```

- ▶ parent.xml 에 의해 생성된 ParentActivity.java에 moveChild 메소드 코드 작성

```
public void moveChild(View view) {
    Intent intent = new Intent( packageContext: ParentActivity.this, ChildActivity.class);
    startActivity(intent);
}
```

- ▶ child.xml 에 의해 생성된 ChildActivity.java에 moveMain 메소드 코드 작성

```
public void moveMain(View view) {
    Intent intent = new Intent( packageContext: ChildActivity.this, MainActivity.class);
    startActivity(intent);
}
```

업네비게이션

- ▶ 설정 파일 AndroidManifest.xml에 계층 구조 설정
 - ▶ AndroidManifest.xml 에 ParentActivity의 부모 Activity로 MainActivity 설정

```
<activity android:name=".ParentActivity"
    android:parentActivityName=".MainActivity"/>
```

- ▶ AndroidManifest.xml 에 ChildActivity의 부모 Activity로 ParentActivity 설정

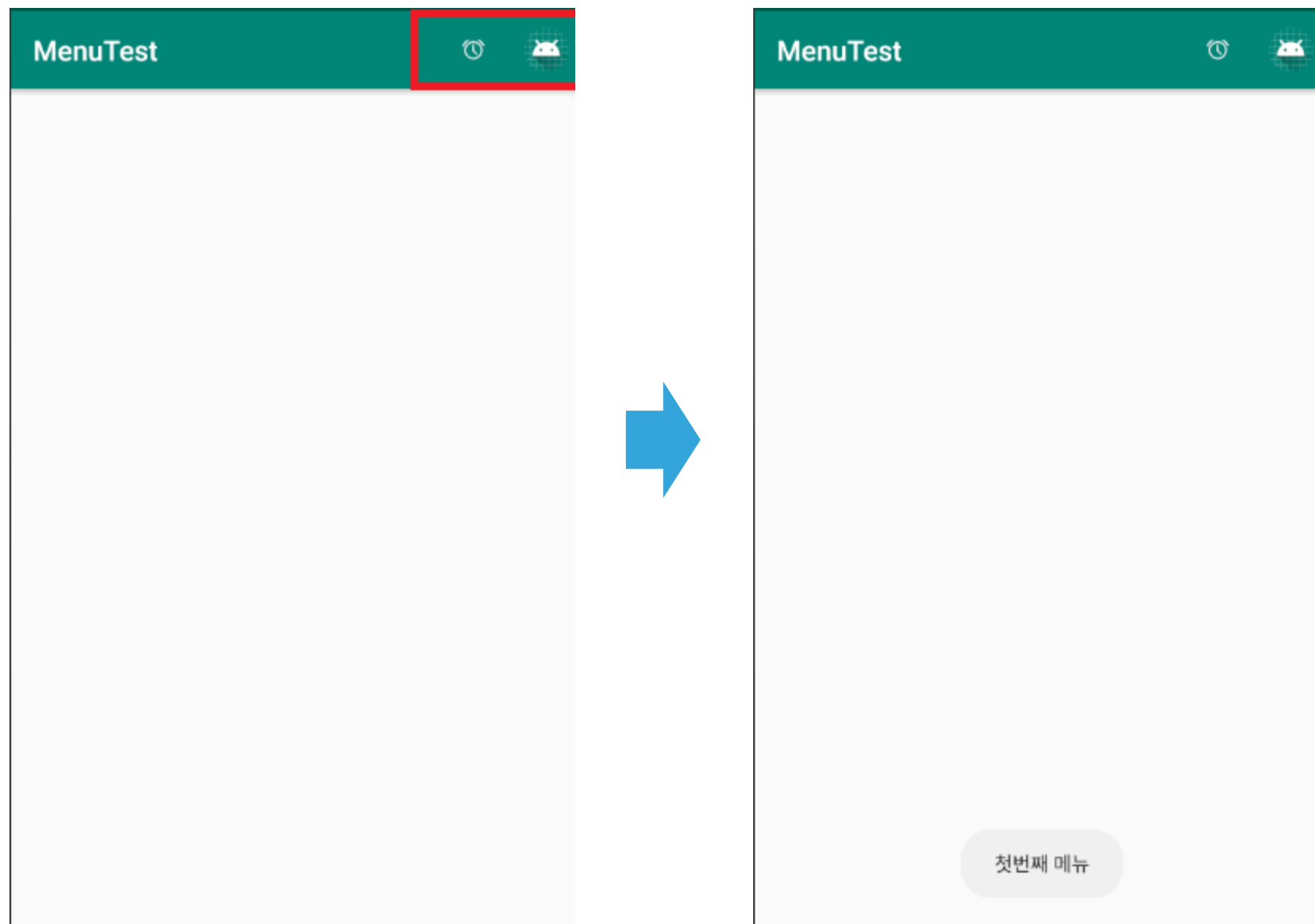
```
<activity android:name=".ChildActivity"
    android:parentActivityName=".ParentActivity"/>
```

메뉴

- ▶ 메뉴를 만들고 메뉴를 클릭했을 때 실행되는 코드 만들기
- ▶ 실습 파일(java 파일과 관련 xml 파일)
 - ▶ MainActivity.java
 - ▶ menu ui 생성
 - ▶ menu 폴더 생성
 - ▶ menu_main.xml 생성

▶ 화면 구성

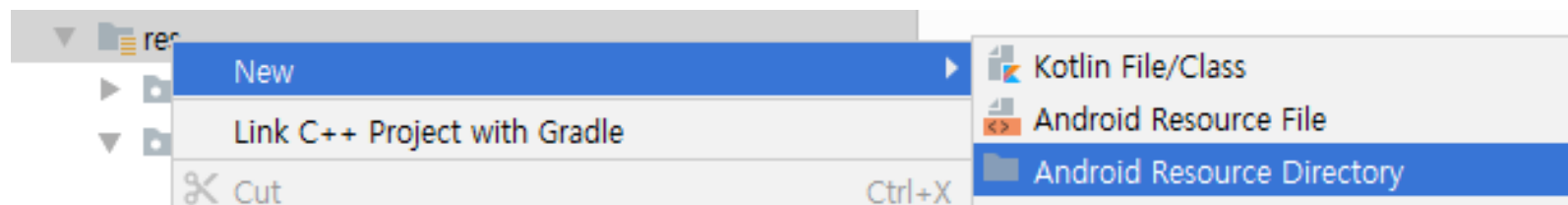
▶ 실행 화면



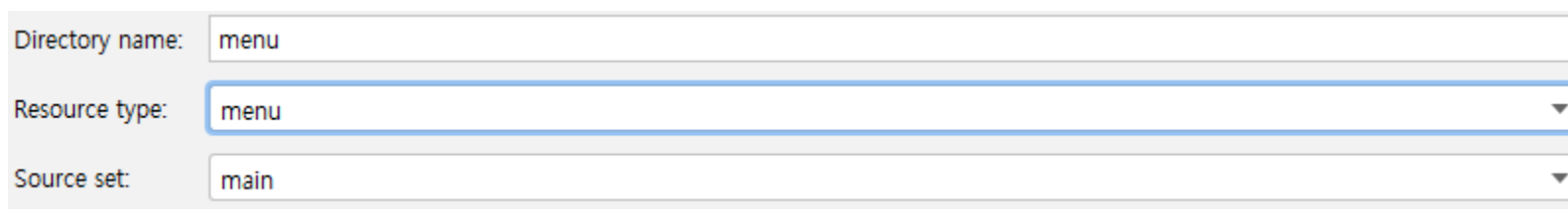
메뉴

- ▶ 메뉴 폴더 및 menu UI 생성

- ▶ res 폴더에서 마우스 우클릭 하여 directory 생성 클릭



- ▶ directory type을 menu 선택 후 생성



- ▶ 생성된 디렉토리에서 마우스 우클릭 하여 menu 파일 생성



- ▶ menu 이름 지정

메뉴

- ▶ menu_main.xml에 메뉴 생성 코드
- ▶ showAsAction의 주요 기능

```
<item
    android:id="@+id/action_menu1"
    android:icon="@android:drawable/ic_lock_idle_alarm"
    android:title="Item1"
    app:showAsAction="ifRoom" />
<item
    android:id="@+id/action_menu2"
    android:icon="@mipmap/ic_launcher_round"
    android:title="Item2"
    app:showAsAction="ifRoom" />
```

- ▶ always : 메뉴 표시
- ▶ naver : 메뉴 숨김
- ▶ Ifroom : 상황에 따라 메뉴 표시 혹은 메뉴 숨김

메뉴

- ▶ menu_main.xml과 MainActivity.java 에 연결

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

- ▶ menu 선택 시 메시지 띄우기

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch(item.getItemId()){
        case R.id.action_menu1:
            Toast.makeText(context: this, text: "첫번째 메뉴", Toast.LENGTH_LONG).show();
            return true;
        case R.id.action_menu2:
            Toast.makeText(context: this, text: "두번째 메뉴", Toast.LENGTH_LONG).show();
            return true;
    }
    return super.onOptionsItemSelected(item);
}
```

모바일 프로그래밍 입문

(2019년도 1학기)

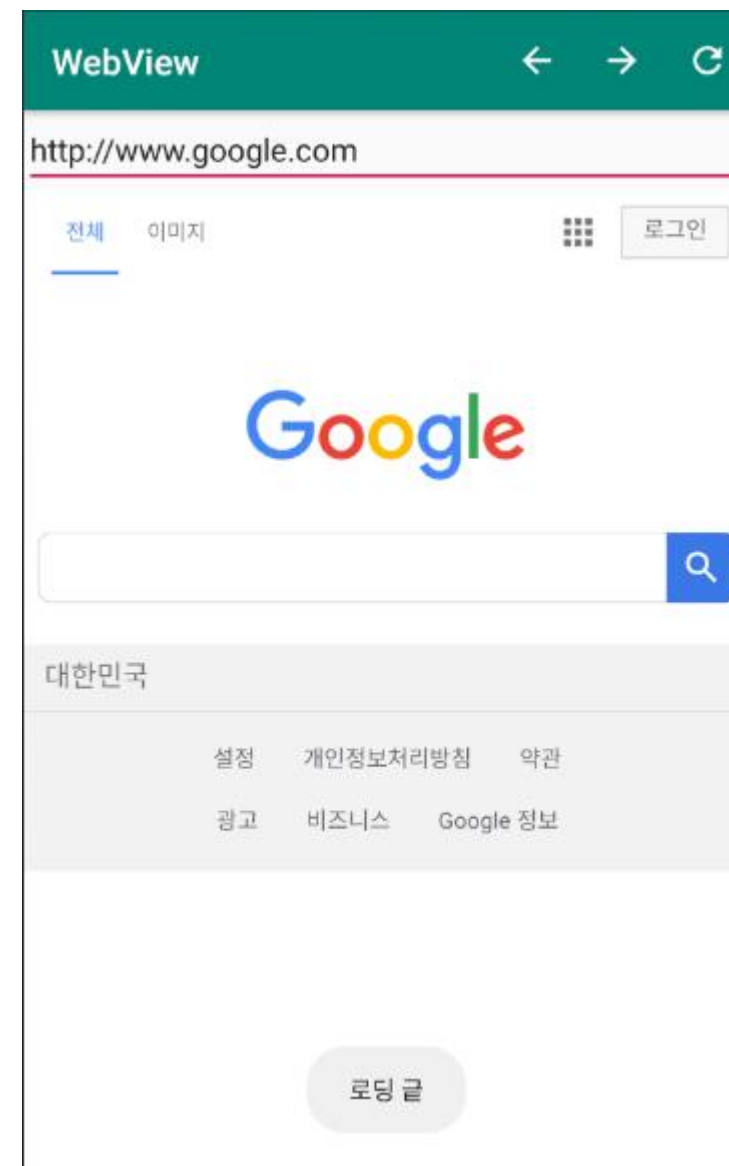
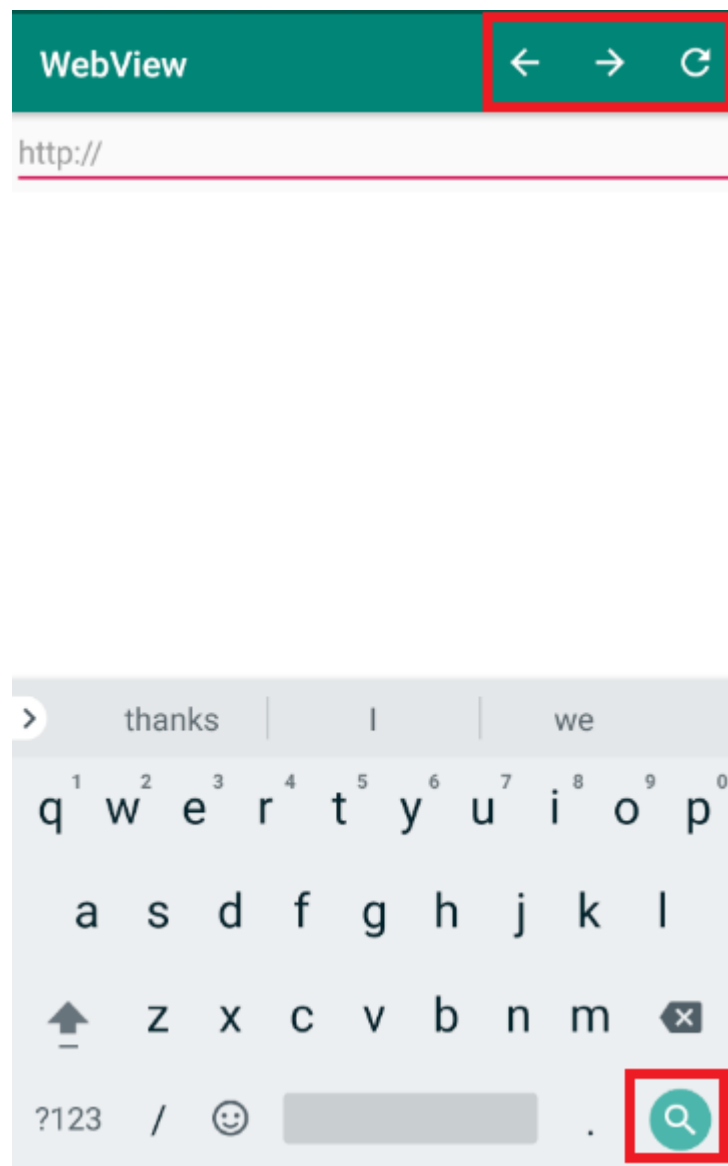
웹뷰 : 웹 브라우저 만들기

웹뷰

- ▶ 웹 브라우저 만들기
- ▶ 실습 파일(java 파일과 관련 xml 파일)
 - ▶ MainActivity.java
 - ▶ menu ui 생성
 - ▶ menu 폴더 생성
 - ▶ menu_main.xml 생성

▶ 화면 구성

▶ 실행 화면



▶ 웹 브라우저 UI 생성

▶ main.xml 코드 작성

- ▶ LinearLayout으로 변경
- ▶ EditText 와 Button 생성
- ▶ WebView 생성

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

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/address_edit"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:hint="http://"
        android:imeOptions="actionSearch"
        android:inputType="textUri"
        android:maxLines="1"
    />
```

```
<Button
    android:id="@+id/move_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onClick"
    android:text="이동"
    android:visibility="gone"
/>

</LinearLayout>
```

```
<WebView
    android:id="@+id/web_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
</WebView>
```

▶ MainActivity.java 에서 UI에 만든 것들 동작하게 코딩

▶ UI에 웹 브라우저 기능이 동작하도록 코딩

▶ UI 생성 객체 선언

```
private EditText address_edit;  
private WebView web_view;  
private Button move_button;
```

▶ 브라우저 기능 셋팅

```
address_edit = findViewById(R.id.address_edit);  
web_view = findViewById(R.id.web_view);  
move_button = findViewById(R.id.move_button);
```

```
WebSettings webSettings = web_view.getSettings();  
webSettings.setJavaScriptEnabled(true);
```

// 선언해 주어야 제대로 동작

```
web_view.setWebViewClient(new WebViewClient(){  
    @Override  
    public void onPageFinished(WebView view, String url) {  
        Toast.makeText(context: MainActivity.this, text: "로딩 끝", Toast.LENGTH_LONG).show();  
    }  
  
    @Override  
    public void onPageStarted(WebView view, String url, Bitmap favicon) {  
    }  
});
```


- ▶ MainActivity.java 에서 UI에 만든 것들 동작하게 코딩

- ▶ 이동 버튼 클릭시 웹 브라우저 보이게 코딩

```
public void onClick(View view) {  
    String address = address_edit.getText().toString();  
    web_view.loadUrl(address);  
}
```

- ▶ 인터넷 사용하기 위한 권한 설정 필요

- ▶ AndroidManifest.xml 파일에 권한 설정 추가

```
<uses-permission android:name="android.permission.INTERNET" />
```

- ▶ MainActivity.java 에서 UI에 만든 것들 동작하게 코딩

- ▶ http:// 입력하지 않아도 사이트 이동 가능하게 코딩

- ▶ 기존 onClick 메소드에 추가

```
public void onClick(View view) {  
    String address = address_edit.getText().toString();  
  
    if(address.startsWith("http://") == false) {  
        address = "http://" + address;  
    }  
  
    web_view.loadUrl(address);  
}
```

- ▶ 뒤로 가기 기능 재정의(기존 뒤로 가기는 Activity의 이동)

```
@Override  
public void onBackPressed() {  
    if(web_view.canGoBack()){  
        web_view.goBack();  
    } else {  
        super.onBackPressed();  
    }  
}
```

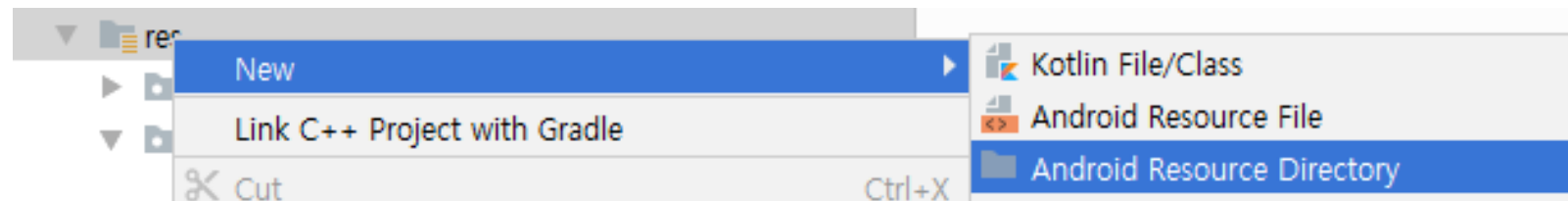
- ▶ MainActivity.java 에서 UI에 만든 것들 동작하게 코딩
 - ▶ imeOption의 기능으로 사이트 이동 가능하게 코딩
 - ▶ 기존 onCreate 메소드에 추가 및 입력 키보드 내리기

```
address_edit.setOnEditorActionListener(new OnEditorActionListener() {  
    @Override  
    public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {  
        if(actionId== EditorInfo.IME_ACTION_SEARCH){  
            move_button.callOnClick();  
  
            InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);  
            imm.hideSoftInputFromWindow(v.getWindowToken(), flags: 0);  
  
            return true;  
        }  
        return false;  
    }  
});
```

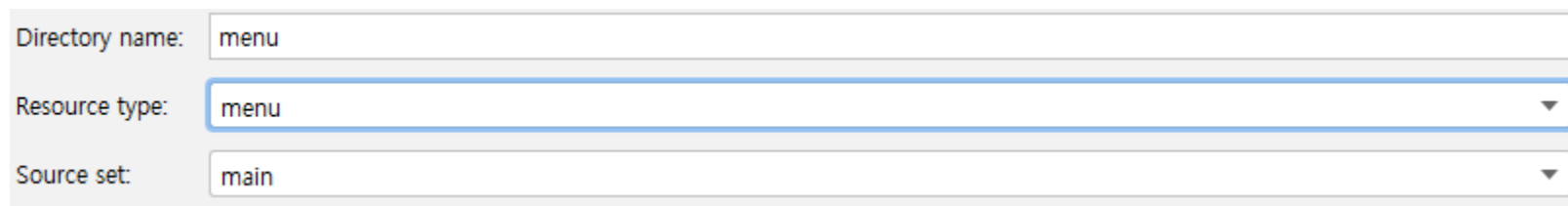
- ▶ 뒤로 가기, 앞으로 가기, 새로 고침 메뉴 추가하기

- ▶ 메뉴 폴더 및 menu UI 생성

- ▶ res 폴더에서 마우스 우클릭 하여 directory 생성 클릭



- ▶ directory type을 menu 선택 후 생성



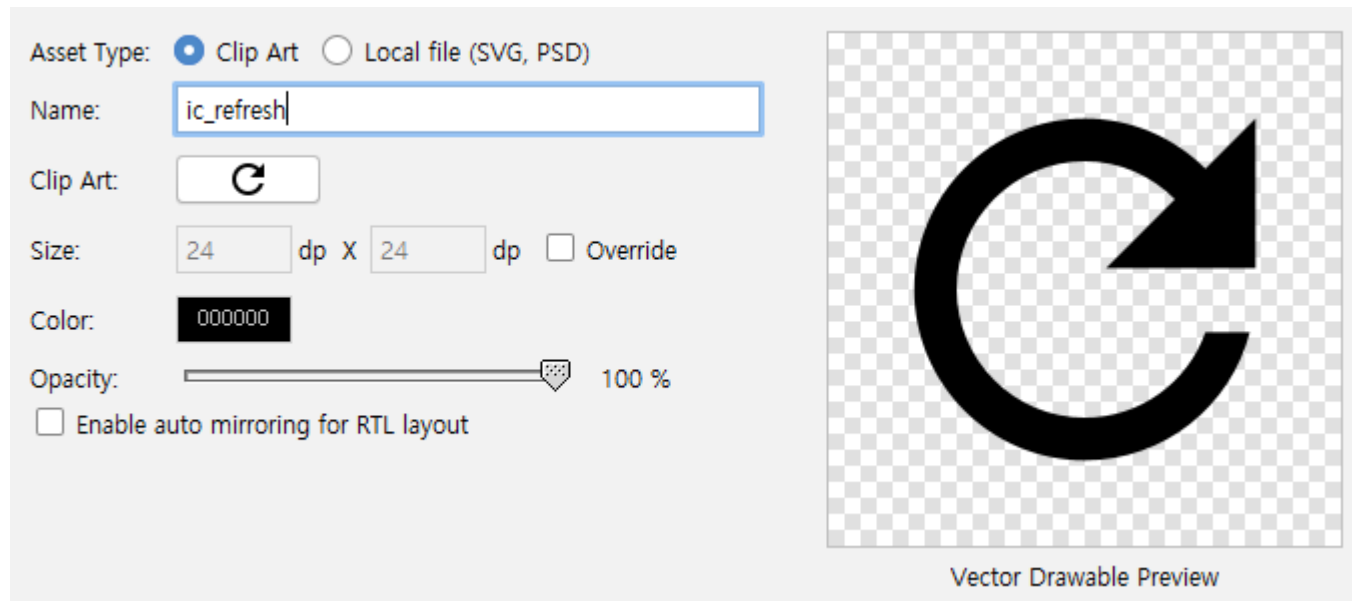
- ▶ 생성된 디렉토리에서 마우스 우클릭 하여 menu 파일 생성



- ▶ menu 이름 지정

▶ 아이콘 제작

- ▶ drawable 폴더에서 우클릭 vector asset 클릭
- ▶ Clip Art 선택하여 이미지 선택 후 Name 지정



- ▶ 뒤로가기, 앞으로가기, 새로고침 3개 만듬

- ▶ 뒤로 가기, 앞으로 가기, 새로 고침 메뉴 추가 및 ICON 대체
- ▶ menu_main.xml에 메뉴 생성 코드

```
<item
    android:id="@+id/action_back"
    android:icon="@drawable/ic_arrow_back"
    android:title="뒤로가기"
    app:showAsAction="ifRoom" />
<item
    android:id="@+id/action_forward"
    android:icon="@drawable/ic_arrow_forward"
    android:title="앞으로 가기"
    app:showAsAction="ifRoom" />
<item
    android:id="@+id/action_refresh"
    android:icon="@drawable/ic_refresh"
    android:title="새로고침"
    app:showAsAction="ifRoom" />
```

- ▶ menu_main.xml과 MainActivity.java 에 연결

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
```

- ▶ menu 선택 시 각각에 맞는 뒤로가기, 앞으로가기, 새로고침 실행

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()){
        case R.id.action_back:
            web_view.goBack();
            return true;
        case R.id.action_forward:
            web_view.goForward();
            return true;
        case R.id.action_refresh:
            web_view.reload();
            return true;
    }

    return super.onOptionsItemSelected(item);
}
```