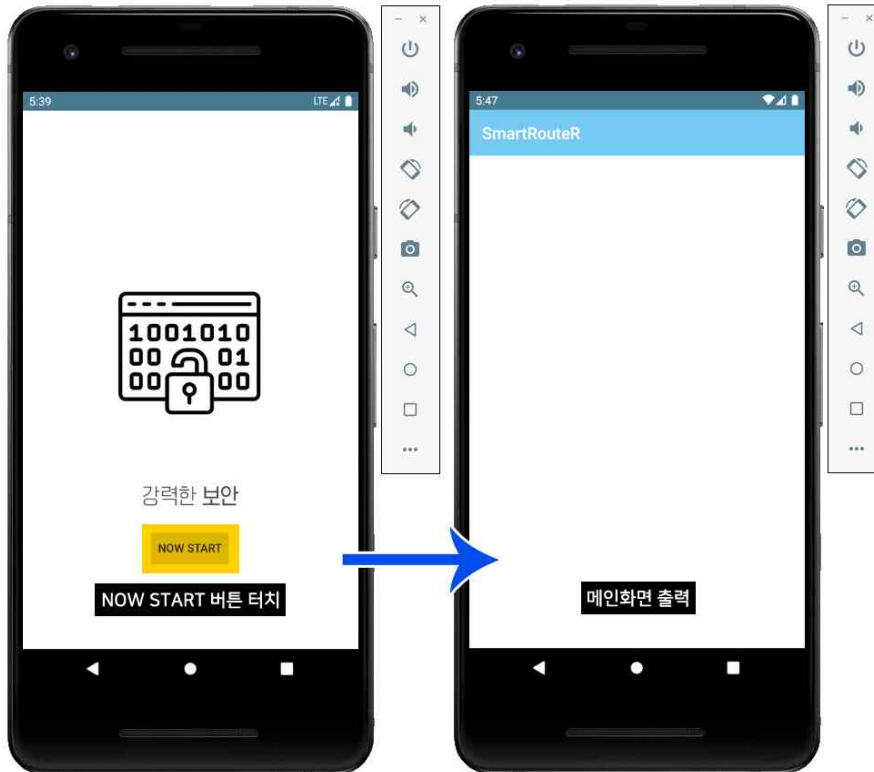


<이번 주 진행된 내용>

1. intent를 사용하여 설명화면(fragment)에서 메인화면(activity)으로 이동하기

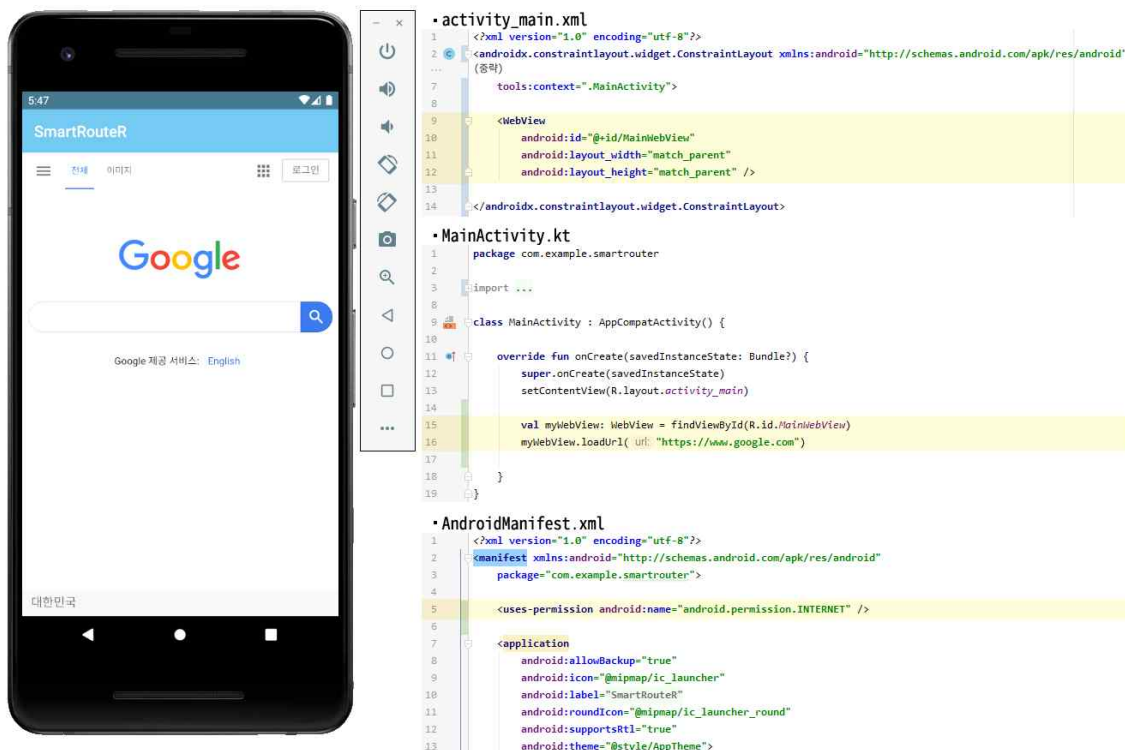


intent를 활용하여 설명화면인 fragment에서 메인화면인 activity로 전환하는 부분을 구현하였다. activity에서 activity로 이동할 때와 fragment에서 activity로 이동할 때 사용하는 코드가 약간 다르기에 아래와 같이 주의할 필요가 있다.

```
1 package com.example.smartrouter
2
3 import ...
4
13
14 class FragmentC : Fragment() {
15
16     (중략)
23 override fun onCreateView(view: View, savedInstanceState: Bundle?) {
24     tv_fragment_name.text = "설명 3"
25     imageView.setImageResource(R.drawable.sub_3)
26
27     button1.setOnClickListener { it: View!
28         val intent = Intent(getActivity(), MainActivity::class.java)
29         getActivity()?.startActivity(intent)
30     }
31
32 }
33
34 }
```

해당 버튼이 들어간 파일의 onCreateView 안에 버튼을 통해 화면이 전환되는 코드를 넣는다. 보통 activity끼리 넘어가는 경우에는 현재 activity를 this로 정의하여 코드를 작성하지만 fragment의 경우에는 this를 인식하지 못하기 때문에 getActivity()를 쓴다. 또한 startActivity(intent) 앞에 getActivity()?.를 넣어주어야 한다.

2. 웹뷰



해당 xml에 웹뷰를 띄워주고 activity에 사이트 url을 입력하여 연결한 뒤 manifest에 인터넷 사용 권한을 주는 과정을 통해 웹페이지를 activity에 연결시킬 수 있다. 가장 중요한 점은 웹페이지 그대로 가져오는 것이 아닌 디바이스별로 페이지 디자인이 대응한다는 점인데 이를 '반응형'이라고 한다. 사용자의 편리성을 위해 반응형 웹페이지를 구현하는 것은 필수적으로 요구된다.

웹뷰가 기본적으로 내장 브라우저 기능을 하지만, 자바스크립트를 이용해 데이터를 주고받음으로써 한 몸처럼 행동할 수가 있다.

WebView에 넣을 html을 이용한 버튼

```
<input type="button" value="test" onclick="showAndroid('Hello')" />
```

그리고 WebView에 넣을 자바스크립트

```
<!--Data from Javascript-->
```

```
<script type="text/javascript">
```

```
function showAndroid(toast) {  
    Android.showToast(toast);  
}
```

```
</script>
```

안드로이드에서도 클래스를 하나 만들어준다.

```
import android.content.Context;
import android.webkit.JavascriptInterface;
import android.widget.Toast;

public class JavaScriptPasser {

    Context mContext;

    public JavaScriptPasser(Context c) {
        mContext = c;
    }

    @JavascriptInterface
    public void showToast(String toast) {
        Toast.makeText(mContext, toast, Toast.LENGTH_SHORT).show();
    }
}
```

어노테이션이 빠지면 동작이 안되는 경우가 많다고 하니 조심한다.(sdk 17이상) 그리고 안드로이드에서 데이터를 넘겨주게끔 통신을 다음과 같이 설정한다.

```
WebView webView = (WebView)findViewById(R.id.webview);
webView.addJavascriptInterface(new JavaScriptPasser(this), "Android");
```

이렇게 되면 웹뷰에서도 버튼을 눌렀을 때, showToast라는 메서드가 작동하게 된다. 다만 클릭을 하지 말고 처음에 스크립트가 실행되었을 때 객체를 넘기고 싶다면 다음과 같은 메서드를 쓸 수 있다.

```
public void passData(final WebView webView, final Data data) {
    webView.setWebViewClient(new WebViewClient() {
        public void onPageFinished(WebView view, String url) {
            Gson gson = new Gson();
            String dataString = gson.toJson(data);
            view.loadUrl("javascript:initData(" + dataString + ")");
        }
    });
}
```

GSON은 JSON을 쉽게 해주는 라이브러리인데 굳이 현재 최신버전은 2.6이며, 신기하게도 기본적으로 라이브러리가 안에 (2.3버전) 내장되어있기 때문에 그냥 쓰면 객체를 JSON객체로 변환해준다. 저렇게 웹뷰로 데이터를 넘길 때 객체로는 넘겨지지 않으므로 string으로 변하게 해서 진행해야 한다.

[출처] Android <-> Webview 사이 데이터 통신하기 (<https://gorakgarak.tistory.com/758>)

3. 반응형 웹디자인

기본적으로 반응형 웹을 구현하기 위해서는 다음을 고려해야한다.

폭 또는 너비가 비율에 따라 크기가 변하는 수치를 사용할 것

고정된 값을 사용하면 브라우저의 크기를 변화시켜도 즉각적인 반응이 나타나지 않으므로 픽셀(pixel)이 아닌 퍼센트등의 비율이 적용되는 단위를 사용해야 함

CSS 미디어 쿼리의 활용

디바이스에 따른 스타일을 조정하기 위해서는 반드시 css media query를 이용. css가 없이 자바스크립트만을 사용하는 방법은 시간도 많이 필요하지만 성능에도 이슈가 따름

변하는 컬럼과 변하지 않는 컬럼을 어떻게 구현할 것인가

화면 변화에 따라 바뀌는 UI를 반응형으로 구현할 경우 자바스크립트 또는 제이쿼리의 사용이 필요할 수도 있음

웹사이트의 화면설계에 따라 단순하게 CSS만을 이용하여 구현하는 것이 매우 어렵다. 이는 웹사이트가 점점 더 복잡해지고 다양한 기능구현이 필요하기 때문이기도 하다.

CSS 미디어 쿼리의 사용 방법

CSS media query를 사용할 수 있어야 반응형웹이 비로서 완성될 수 있다. Media query는 특정 resolution에 맞는 각각의 css를 적용할 수 있는 css3 기법이다.

가변 영역과 아닌 부분이 함께 있는 경우 반응형으로 구현하기

화면에 보여질 컬럼이 2개 이상이고 이중 하나를 고정된 길이인 경우 문제가 된다.

아래의 두 컬럼 모두 변하는 경우 각각 20%, 80%와같이 비율을 정하면 되겠지만 하나는 무조건 200px이라 언급했으므로 반드시 하나의 값에서 200을 제외해야한다. 그래서 떠오르는 부분이 바로 자바스크립트를 사용해 브라우저 크기의 변화 시 이벤트를 감지해 전체 폭에서 정해진 200px을 제외한 값을 계속해서 다른 컬럼에 적용하는 방법이다.

! css를 사용하여 반응형웹에서 가변 영역 구현하는 팁

CSS만을 사용해서 구현할 수 있는 방법이 있다. 아래 두가지 속성을 사용하는 방법이다.

```
padding-left: 200px;
box-sizing: border-box;
```

여기서 사용된 box-sizing 속성은 전체너비에서 padding에 사용된 값을 포함할지 아니면 따로 계산할지의 여부를 결정한다. 좌측의 padding 200px은 당연히 왼쪽에 위치할 컬럼과 레이아웃이 겹치게 된다. 그럴 수 있도록 position 속성을 absolute를 써야한다.

중요한 것은 좌측 공간을 비우기 위해 left나 margin을 사용하지 않는 것이다. 반드시 전체 너비와 함께 계산할 수 있는 padding을 사용한다.

[출처] 반응형 웹페이지를 가장 쉽게 만드는 팁과 노하우

<https://webisfree.com/2016-01-18/%EB%B0%98%EC%9D%91%ED%98%95-%EC%9B%B9%ED%8E%98%EC%9D%B4%EC%A7%80%EB%A5%BC-%EA%B0%80%EC%9E%A5-%EC%89%BD%EA%B2%8C-%EB%A7%8C%EB%93%9C%EB%8A%94-%ED%8C%81%EA%B3%BC-%EB%85%B8%ED%95%98%EC%9A%B0>

<다음 주 진행할 내용>

1. 데이터를 주고받는 방법 중 Json 방식에 관하여 조사
2. 반응형 웹페이지를 어떻게 제작할 것인지 조사