



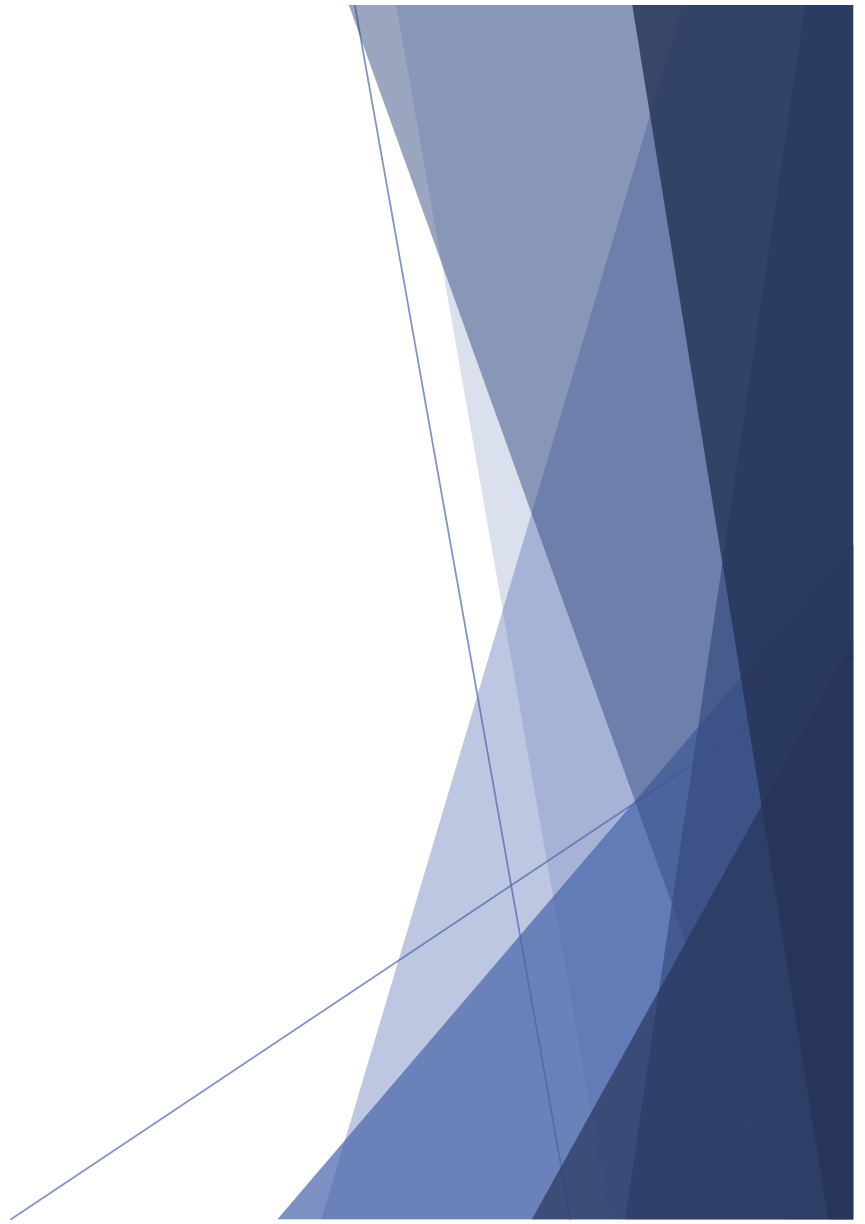
Android Programming

Extra: WebView로 Web Browser 만들기



Android WebView

WebView로 웹 브라우저 만들기



Android WebView

- ▶ **WebView** : 안드로이드 앱에서 웹 페이지를 보여주는 내장 브라우저의 역할을 하는 위젯
 - ▶ 브라우저 기능을 갖고 있지만, **URL** 입력바, 뒤로가기, 다음 페이지 가기 등 기능 버튼은 제공하지 않음
 - ▶ 오픈소스 WebKit 기반의 HTML 렌더링 엔진
 - ▶ 주로 앱 주요 기능과 관련 없는 콘텐츠 페이지 등을 보여줄 때 WebView로 구현
 - ▶ 최근 Hybrid App 기술과 관련, 핵심 기술이 되기도 함

Android WebView

: WebView 사용하기

- ▶ WebView를 사용하려면 AndroidManifest.xml에서 INTERNET 권한을 획득해야

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

- ▶ 레이아웃에서 사용시에는 <WebView> 태그로 선언함

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

Android WebView

: WebView 설정과 URL 로드

- ▶ WebView를 설정하기 위해서는 WebView로부터 WebSettings 객체를 얻어와 설정함

```
WebSettings settings = webView.getSettings();  
settings.setJavaScriptEnabled(true);
```

- ▶ WebView에서 페이지를 불러오기 위해서는 loadUrl 메서드를 이용

```
webView.loadUrl("http://naver.com");
```

Android WebView

: 페이지를 컨트롤하기 위한 WebView의 메서드들

- ▶ Page 로드, 이전 페이지, 다음 페이지 등의 페이지 컨트롤 기능을 사용하기 위해서는 다음과 같은 메서드를 이용

메서드	설명
<code>loadUrl</code>	페이지 URL을 로드함
<code>canGoBack</code>	웹뷰가 이전 페이지로 갈 수 있는지 체크
<code>goBack</code>	웹뷰의 이전 페이지로 이동
<code>canGoForward</code>	웹뷰가 다음 페이지로 갈 수 있는지 체크
<code>goForward</code>	웹뷰의 다음 페이지로 이동

Android WebView

: WebView의 정밀한 컨트롤

- ▶ Page 로드 시작, Page 로드 종료 등, WebView의 정밀한 컨트롤을 위해서는 WebViewClient를 상속한 커스텀 웹 뷰 클라이언트를 새로 정의하여 각종 메서드들을 오버라이드 해야 함

```
class MyWebViewClient extends WebViewClient {  
  
    @Override  
    public void onPageStarted(WebView view, String url, Bitmap favicon) {  
        super.onPageStarted(view, url, favicon);  
    }  
  
    @Override  
    public void onPageFinished(WebView view, String url) {  
        super.onPageFinished(view, url);  
    }  
}
```

Android WebView

: WebView의 정밀한 컨트롤

- ▶ WebViewClient를 등록한 커스텀 웹뷰 클라이언트를 WebView 내에서 사용하기 위해서는 setWebViewClient 메서드를 이용하여 등록

```
webView.setWebViewClient(new MyWebViewClient());
```

[실습] WebViewExample

WebView를 이용하여 간단한 웹 브라우저를 구현하는 예제입니다.

1) 프로젝트를 생성하고, activity_main.xml은 파일을 다운받아 교체한 후 확인합니다. 구성요소는 다음과 같습니다.

- WebView (webView) : 로드된 웹 페이지를 렌더링할 웹뷰
- EditText (editUrl) : URL을 입력받는 EditText
- ProgressBar (progress) : 페이지가 로드되는 동안 진행 상태를 표시
- Button (btnGo) : 입력한 URL 페이지를 WebView에서 로드
- Button (btnPrev) : 브라우저 히스토리에 이전 페이지가 있으면 이전 페이지로 이동
- Button (btnForward) : 브라우저 히스토리에 다음 페이지가 있으면 다음 페이지로 이동



Android WebView

[실습] WebViewExample

WebView를 이용하여 간단한 웹 브라우저를 구현하는 예제입니다.

2) WebSettings를 이용, JavaScript를 사용 가능하게 합니다.

3) WebViewClient를 상속받은 커스텀 WebViewClient를 구현하고, 다음의 기능을 추가합니다.

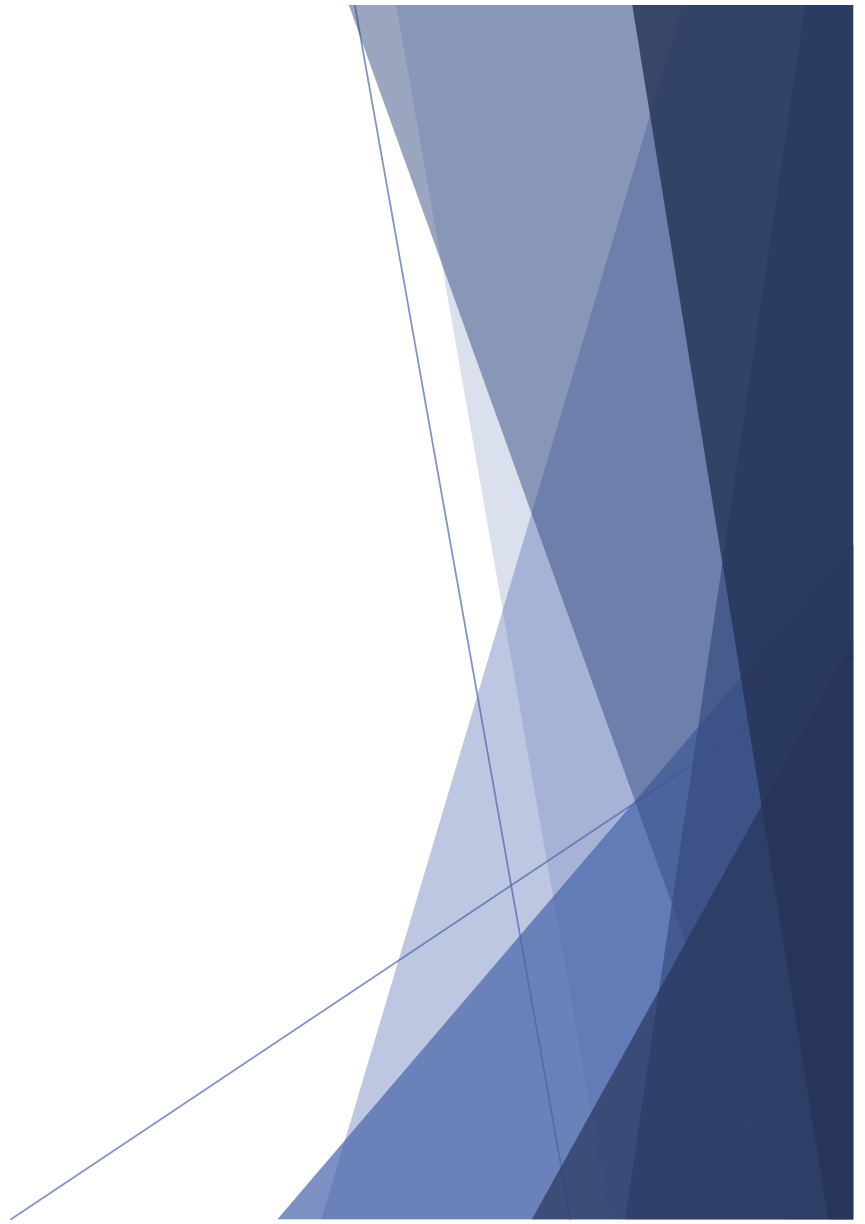
- 페이지 로드 시작시 : ProgressBar를 보이게 한다
- 페이지 로드 완료시 : ProgressBar를 숨긴다.





Android WebView

Hybrid App

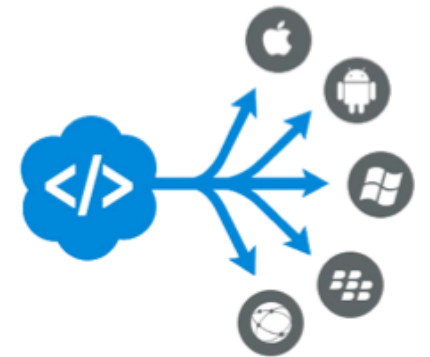


Hybrid App

- ▶ Hybrid App : 웹 서비스를 앱에서 이용 가능하게 하는 기술
 - ▶ 콘텐츠를 여러 모바일 플랫폼에서 쓸 수 있는 웹 기반 구현방식과 모바일 기기의 고유한 특성을 극대화할 수 있는 네이티브 앱 개발 방식을 절충
 - ▶ 앱 내부에 자바스크립트, HTML, 스타일시트 등 웹 기술을 적용해 정보를 다루는 개발 방식을 도입한 것
- ▶ 하이브리드 방식의 개발
 - ▶ 웹 기술로 내부 구조와 인터페이스 개발
 - ▶ 카메라, 단말기, 센서 등의 기능은 단말기 전용 API를 호출하여 사용하는 방식



Hybrid
App



Hybrid App

: 하이브리드 앱을 구현하는 기술

- ▶ 터치스크린 기기용 모바일 웹 개발 프레임워크
 - ▶ Sencha Touch
 - ▶ JQTouch 등
- ▶ 웹 앱을 패키징하기 위한 툴
 - ▶ PhoneGap
 - ▶ ionic
 - ▶ Appspresso 등
 - ▶ 안드로이드 웹 뷰를 커스터마이징 하여 작성

Hybrid App

: WebView와 Activity 통신을 위한 JavaScriptInterface

- ▶ Activity와 WebView 내의 HTML이 서로 데이터를 주고받기 위해서는 JavaScriptInterface를 구현하여 웹 뷰에 추가해 주어야 함
 - ▶ AndroidBridge라는 JavaScriptInterface 클래스를 HybridBridge라는 이름으로 웹뷰에 등록

```
webView.addJavascriptInterface(new AndroidBridge(this), "HybridBridge");
```

- ▶ 이렇게 등록된 인터페이스는 JavaScript에서 window.HybridBridge 로 접근할 수 있음

```
window.HybridBridge.messageFromHtml(message);  
// Activity의 HybridBridge에 구현된 messageFromHtml 메서드를 호출
```

- ▶ Activity에서 WebView 내의 JavaScript를 호출하려면 loadUrl 메서드에 JavaScript 호출 코드를 전송

```
webView.loadUrl("javascript:messageFromActivity('Hello, Hybrid!')");
```

Hybrid App

[실습] HybridExample

WebView를 이용하여 간단한 웹 Hybrid App 기능을 구현합니다.

1) 프로젝트를 만들고 activity_main.xml을 다운받은 파일로 교체합니다.

2) new > Folder > Assets Folder를 생성한 후, assets.zip 파일을 다운받아 압축을 풀고 assets 폴더에 복사합니다. 내용을 간략히 살펴봅니다.

- index.html : 웹뷰에 표시될 html 파일입니다.
- css/skeleton.css : html의 디자인 렌더링을 도와줄 css 파일입니다.
- javascript/main.js : html에서 인터렉션 기능을 담당할 자바스크립트 파일입니다.

3) index.html 파일을 webView에서 로드하고 다음의 작업을 수행합니다.

- btnSendMessage 클릭시 : webView 내 자바스크립트의 sendMessageFromActivity 함수를 호출합니다.
- html 내 버튼을 누르면 호출되는 messageFromHtml 수신 메서드를 JavaScriptInterface 로 구현하여 웹뷰에 등록합니다.

