

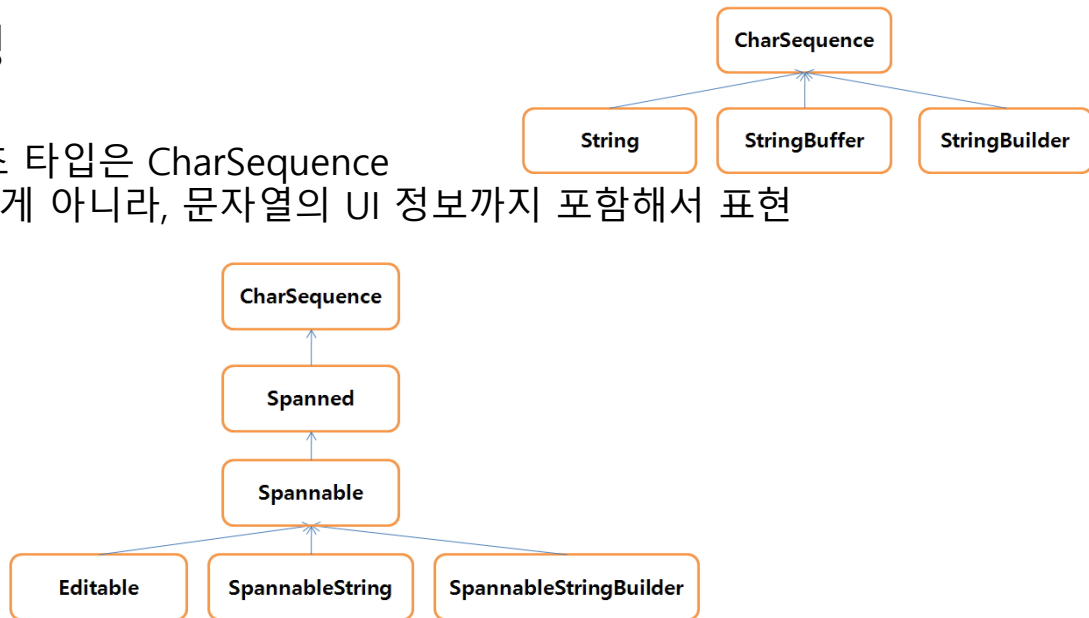


11장. 다양한 뷰 활용

11.1 Spannable

11.1.1. Spannable의 필요성

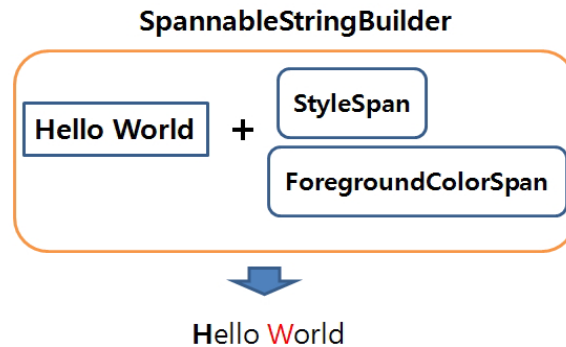
- 안드로이드에서 문자열의 기초 타입은 CharSequence
- 문자열이 데 이터만 표현하는 게 아니라, 문자열의 UI 정보까지 포함해서 표현



Hello World

- 데이터가 어떻게 화면에 나올지의 UI 정보를 데이터소스로 가지고 뷰는 그 정보를 참조해서 화면에 출력

11.1 Spannable



11.1.2. Spannable 적용

- TextView가 Spannable을 참조해서 화면에 출력하려면 bufferType이라는 속성을 지정
- EditText의 bufferType은 기본값이 "editable"이고 editable은 spannable을 내장하는 개념

```
<TextView
    android:id="@+id/spanView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:bufferType="spannable"/>
```

11.1 Spannable

```
//Spannable을 포함하는 문자열
SpannableStringBuilder builder=new SpannableStringBuilder(data);

//img 문자열의 시작위치
int start=data.indexOf("img");
if(start>-1){
    //img 문자열의 끝 위치
    int end=start+"img".length();
    //이미지 획득
    Drawable dr=ResourcesCompat.getDrawable(getResources(),
    //이미지의 화면 출력정보 설정
    dr.setBounds(0, 0, dr.getIntrinsicWidth(), dr.getIntrinsicHeight());
    //ImageSpan 준비
    ImageSpan span=new ImageSpan(dr);
    //SpannableStringBuilder에 ImageSpan 적용
    builder.setSpan(span, start, end, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
}
```

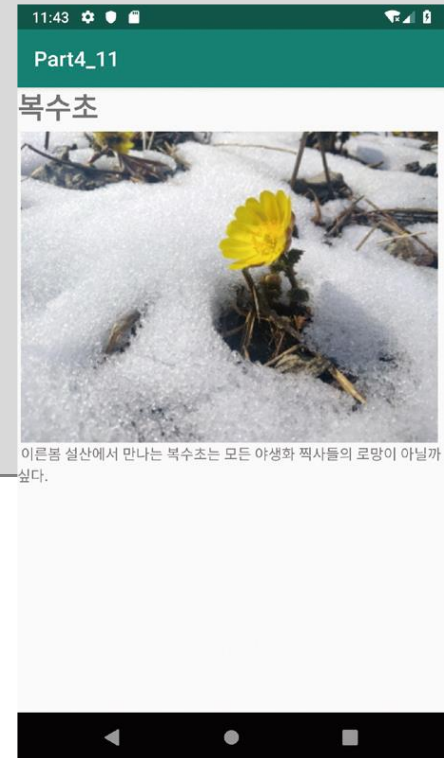
R.drawable.img1, null);

- SPAN_EXCLUSIVE_EXCLUSIVE: 왼쪽 제거, 오른쪽 제거
- SPAN_EXCLUSIVE_INCLUSIVE: 왼쪽 제거, 오른쪽 포함
- SPAN_INCLUSIVE_EXCLUSIVE: 왼쪽 포함, 오른쪽 제거
- SPAN_INCLUSIVE_INCLUSIVE: 왼쪽 포함, 오른쪽 포함

11.1 Spannable

```
//문자열 시작위치 획득
start=data.indexOf("복수초");
if(start > -1){
    //문자열 끝 위치 획득
    int end=start+"복수초".length();
    //BOLD 타입으로 StyleSpan 준비
    StyleSpan styleSpan=new StyleSpan(Typeface.BOLD);
    //기본 크기 보다 2배 크게 표현하는 Span 준비
    RelativeSizeSpan sizeSpan=new RelativeSizeSpan(2.0f);
    //Span 적용
    builder.setSpan(styleSpan, start, end+2, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
    builder.setSpan(sizeSpan, start, end+2, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
}
```

- ForegroundColorSpan: 전경 색상값 적용
- BackgroundColorSpan: 배경 색상값 적용
- UnderlineSpan: 밑줄 적용
- ClickableSpan: 문자열 클릭 이벤트 적용
- AbsoluteSizeSpan: 크기 변경 적용
- ImageSpan: 이미지 데이터 적용
- RelativeSizeSpan: 크기 적용
- StyleSpan: 스타일 적용
- URLSpan: URL 링크 모양과 클릭 이벤트 적용



11.1 Spannable

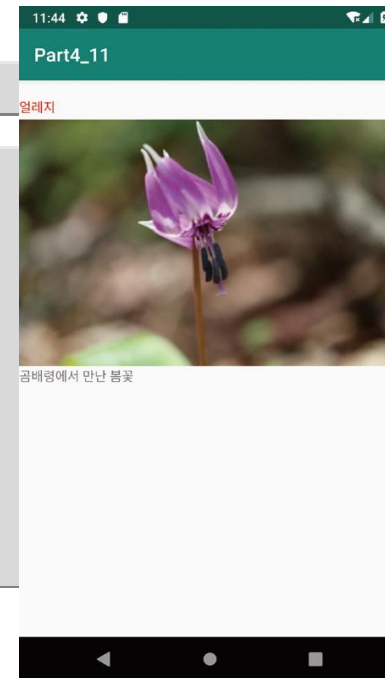
11.1.3. fromHtml() 함수로 적용

HTML 태그로 표현

- fromHtml(String source)
- fromHtml(String source, Html.ImageGetter imageGetter, Html.TagHandler tagHandler)
- fromHtml(String source, int flags, Html.ImageGetter imageGetter, Html.TagHandler tagHandler)

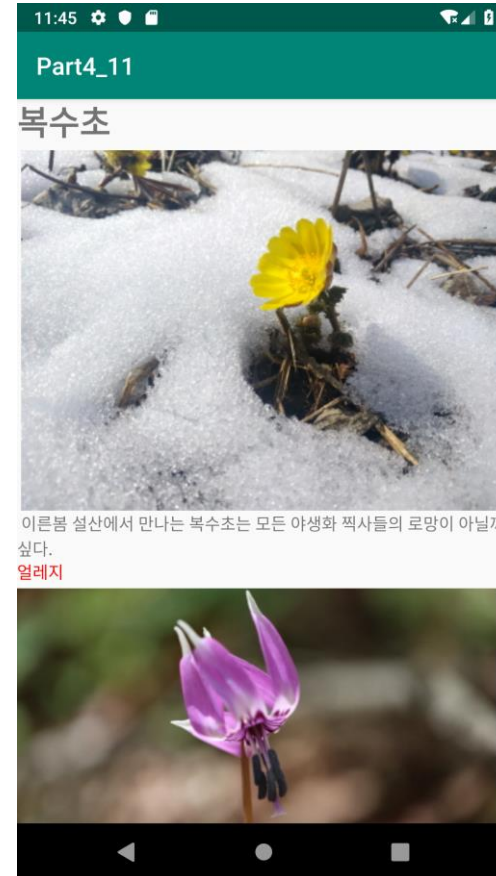
```
htmlView.setText(Html.fromHtml(html, new MyImageGetter(), null));
```

```
class MyImageGetter implements Html.ImageGetter {  
    @Override  
    public Drawable getDrawable(String source) {  
        if(source.equals("img1")){  
            Drawable dr=ResourcesCompat.getDrawable(getResources(), R.drawable.img2, null);  
            dr.setBounds(0, 0, dr.getIntrinsicWidth(), dr.getIntrinsicHeight());  
            return dr;  
        }  
        return null;  
    }  
}
```



Step by Step 11-1 - Spannable

- 모듈 생성
- 파일 복사
- MainActivity 작성
- 실행



11.2 WebView

11.2.1. WebView 활용

- WebView는 안드로이드 앱에서 내장 브라우저 역할을 하는 뷰



11.2 WebView

- 서버 URL로 HTML 파일을 가져와 WebView 설정

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

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

```
WebSettings settings=webView.getSettings();  
settings.setJavaScriptEnabled(true);  
  
webView.loadUrl("http://m.daum.net");
```

- HTML 파일을 앱 내부에 둘려면 assets 폴더를 이용

```
webView.loadUrl("file:///android_asset/test.html");
```

11.2 WebView

11.2.2. 자바스크립트와 자바 연동

- 자바스크립트에서 자바의 함수를 호출, 자바에서 자바스크립트 함수를 호출
- Javascript에서 호출할 함수에 @JavascriptInterface이라는 어노테이션(annotation)을 선언

```
class JavascriptTest {  
    @JavascriptInterface  
    public String getChartData(){  
        StringBuffer buffer=new StringBuffer();  
        //.....  
        return buffer.toString();  
    }  
}
```

- 자바스크립트를 위한 클래스를 공개

```
webView.addJavascriptInterface(new JavascriptTest(), "android");
```

- 자바스크립트에서 함수를 호출

```
var data=window.android.getChartData();
```

- 자바에서 자바스크립트 함수를 호출

```
webView.loadUrl("javascript:lineChart()");
```

11.2 WebView

11.2.3. 이벤트 처리

- HTML에서 발생한 사용자 이벤트를 자바에서 처리

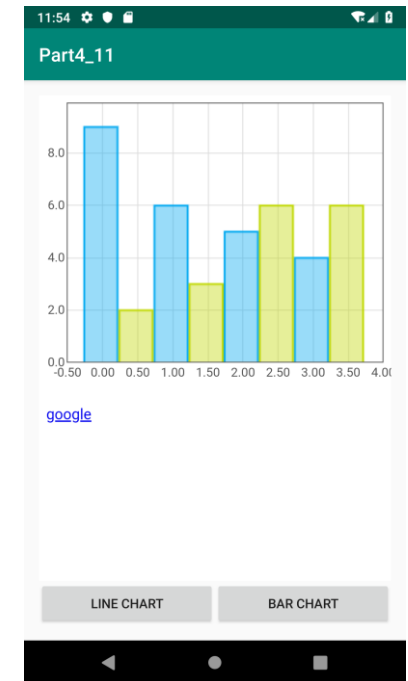
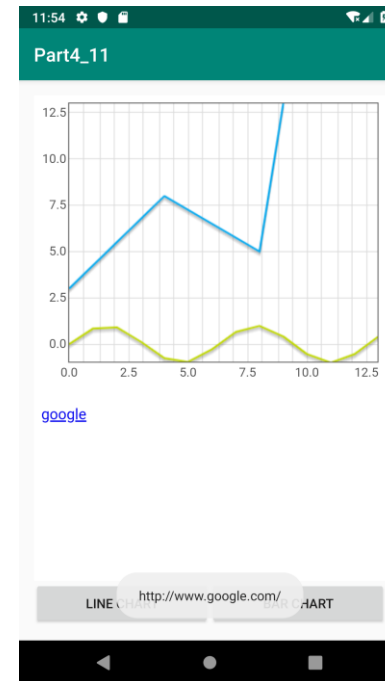
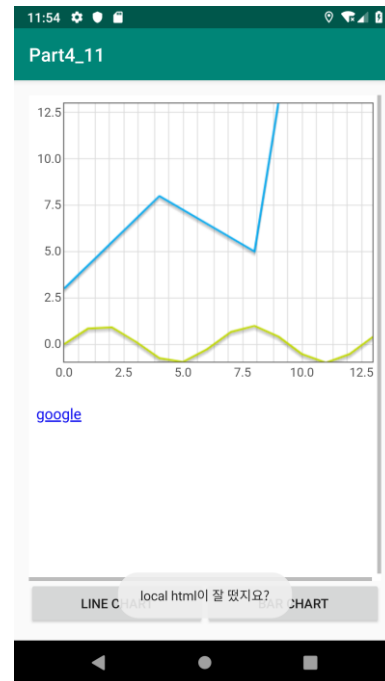
```
class MyWebClient extends WebViewClient{
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        //.....
        return true;
    }
}
```

- 브라우저 자체 이벤트

```
class MyWebChrome extends WebChromeClient{
    @Override
    public boolean onJsAlert(WebView view, String url, String message, JsResult result) {
        //.....
        result.confirm();
        return true;
    }
}
```

Step by Step 11-2 - WebView

- 액티비티 생성
- assets 폴더 생성
- 파일 복사
- Lab11_2Activity 작성
- 실행



11.3 기타 유용한 뷰

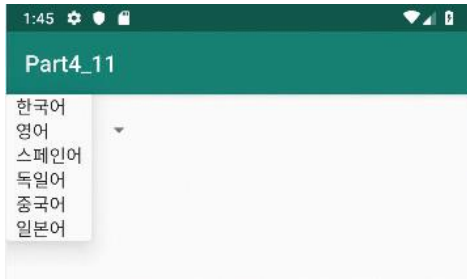
11.3.1. 콤보박스 : Spinner

- AdapterView이며 Spinner 또한 Adapter를 이용하여 구성

```
<Spinner  
    android:id="@+id/spinner"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

```
ArrayAdapter<String> aa=new ArrayAdapter<String>(this, android.R.layout.simple_dropdown_item_1line,  
datas);  
aa.setDropDownViewResource(android.R.layout.simple_spinner_item);  
spinner.setAdapter(aa);
```

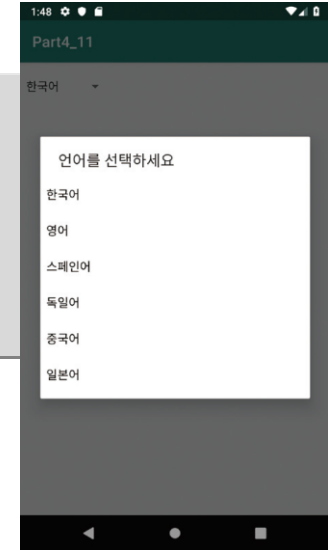
- 라이브러리에서 제공하는 펼침목록 레이아웃은 simple_spinner_item과 simple_spinner_dropdown_item



11.3 기타 유용한 뷰

- 다이얼로그 형태

```
<Spinner
    android:id="@+id/spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:prompt="@string/spinner_prompt"
    android:spinnerMode="dialog"
/>
```

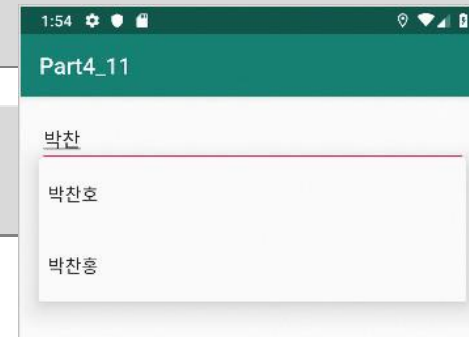


11.3.2. 텍스트 자동완성 : AutoCompleteTextView

- AdapterView의 일종이며 Adapter 클래스를 이용하여 완성

```
<AutoCompleteTextView
    android:id="@+id/auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```
ArrayAdapter<String> autoAdapter=new ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line,autoDatas);
autoCompleteTextView.setAdapter(autoAdapter);
```



11.3 기타 유용한 뷰

- android:completionThreshold="1": 자동완성을 위한 펼침목록이 한 글자 입력되었을 때 보인다. 기본은 두 글자
- android:completionHint="항목을 선택하세요": 펼침목록 아래에 설명 글
- android:dropDownWidth="200dp": 펼침목록의 가로 크기
- android:dropDownHeight="100dp": 펼침목록의 세로 크기
- android:dropDownVerticalOffset="100dp": 펼침목록과 AutoCompleteTextView 세로 방향과 오프셋 크기
- android:dropDownHorizontalOffset="100dp": 펼침목록과 AutoCompleteTextView 가로방향과 오프셋 크기

11.3.3. 프로그레스바 : ProgressBar

- 원 모양: 작업의 시작과 끝을 정확하게 알 수 없을 때
- 막대 모양: 작업의 시작과 끝을 정확하게 알 때

```
<ProgressBar  
    android:id="@+id/progress"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center" />
```



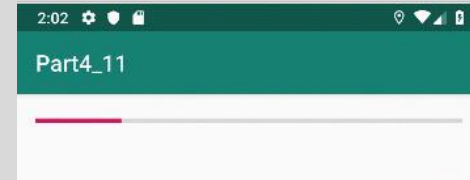
11.3 기타 유용한 뷰

```
<ProgressBar
    android:id="@+id/progress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    style="?android:attr/progressBarStyleHorizontal"
    android:max="100"/>
```

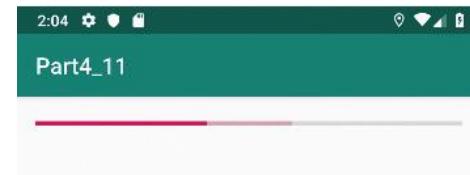
막대 모양은 style 속성값으로 "?android:attr/progressBarStyleHorizontal"을 지정

- setProgress(int progress): 매개변수로 ProgressBar의 값을 지정할 때 사용
- incrementProgressBy(int diff): 매개변수 값을 현재 값에서 더하거나 뺄 때 사용

```
class ProgressThread extends Thread {
    @Override
    public void run() {
        for(int i=0; i<10; i++){
            SystemClock.sleep(1000);
            progressBar.incrementProgressBy(10);
        }
    }
}
```



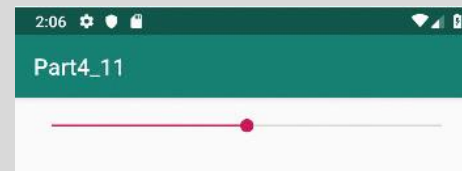
- setSecondaryProgress(int secondaryProgress)
- incrementSecondaryProgressBy(int diff)



11.3 기타 유용한 뷰

11.3.4. 값을 입력받는 프로그레스바 : SeekBar

```
<SeekBar  
    android:id="@+id/seek"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:max="100"  
    android:progress="50"/>
```



```
seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
    @Override  
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
  
    }  
  
    @Override  
    public void onStartTrackingTouch(SeekBar seekBar) {  
  
    }  
  
    @Override  
    public void onStopTrackingTouch(SeekBar seekBar) {  
  
    }  
});
```

11.3 기타 유용한 뷰

- thumb 속성을 이용해 개발자가 원하는 특정 이미지 지정

```
<SeekBar  
    android:id="@+id/seek"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:max="100"  
    android:progress="50"  
    android:thumb="@drawable/ic_seek"/>
```



Step by Step 11-3 – 다양한 View

- 액티비티 생성
- 파일 복사
- activity_lab11_3.xml 파일 작성
- Lab11_3Activity 작성
- 실행

