



고급 위젯 다루기

☰ 태그

(+) FrameLayout → TimePicker, DatePicker, CalendarView

1. AnalogClock, DigitalClock

- 화면에 시간을 표시하는 위젯으로 시계를 표현하는 용도

2. Chronometer

- 타이머 형식의 위젯, 시간 측정
- start(), stop(), reset() 메소드
- android:format="시간 측정 : %s"

3. TimePicker

- 시간 표시와 조절

4. DatePicker, CalendarView

- 날짜 표시와 조절
- 캘린더뷰의 XML 속성
 - showWeekNumber : 현재 몇 주 차인지를 각 주의 맨 앞에 출력

▼ 실습 6-1) 날짜/시간 예약 앱

▼ XML 코드

- Chronometer, height="0dip", FrameLayout

```
<LinearLayout>
  <LinearLayout>
    <Chronometer
      android:id="@+id/chronometer1"
      android:format="예약에 걸린 시간 %s"
      android:gravity="center"/>

    <Button
      android:id="@+id/btnStart"
      android:text="예약 시작" />
  </LinearLayout>

  <RadioGroup>

    <RadioButton
      android:id="@+id/rdoCal"
      android:text="날짜 설정 (캘린더뷰)" />

    <RadioButton
      android:id="@+id/rdoTime"
      android:text="시간 설정" />
  </RadioGroup>

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dip"
    android:layout_weight="1" >

    <FrameLayout
```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center" >

        <CalendarView
            android:id="@+id/calendarView"
            android:showWeekNumber="false" />

        <TimePicker
            android:timePickerMode="spinner"
            android:id="@+id/timePicker" />
    </FrameLayout>
</LinearLayout>

<LinearLayout>

    <Button
        android:id="@+id/btnEnd"
        android:text="예약완료" />

    <TextView
        android:id="@+id/tvYear"
        android:text="0000년 0월 0일 0시 00분" />
</LinearLayout>
</LinearLayout>

```

▼ Kotlin 코드

- visibility = View.INVISIBLE, start(), stop(), setTextColor

```

selectYear, selectMonth, selectdayOfMonth,
currentHour, currentMinute,
setOnDateChangeListener, SystemClock.elapsedRealtime()

```

```

class MainActivity : AppCompatActivity() {
    var selectYear : Int = 0
    var selectMonth : Int = 0
    var selectdayOfMonth : Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        timePicker.visibility = View.INVISIBLE
        calendarView.visibility = View.INVISIBLE

        rdoCal.setOnClickListener {
            timePicker.visibility = View.INVISIBLE
            calendarView.visibility = View.VISIBLE
        }

        rdoTime.setOnClickListener {
            timePicker.visibility = View.VISIBLE
            calendarView.visibility = View.INVISIBLE
        }

        btnStart.setOnClickListener {
            chronometer1.base = SystemClock.elapsedRealtime()
            chronometer1.start()
            chronometer1.setTextColor(Color.RED)
        }

        btnEnd.setOnClickListener {
            chronometer1.stop()
            chronometer1.setTextColor(Color.BLUE)

            tvYear.text = "" + selectYear + "년" + selectMonth + "월" + selectdayOfMonth + "일" + timePicker.currentHour + "시"
        }

        calendarView.setOnDateChangeListener { calendarView, i, i2, i3 ->
            selectYear = i
            selectMonth = i2 + 1
            selectdayOfMonth = i3
        }
    }
}

```

▼ 결과



▼ 직접 풀어보기 6-1)

- ▼ 캘린더뷰 대신에 **데이트피커**를 사용하여 날짜를 설정

```
<DatePicker
    android:id="@+id/datePicker"
    android:datePickerMode="spinner"
    android:showWeekNumber="false"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

- ▼ <예약 시작> 기능은 크로노미터를 클릭하면 동작하게 하고, <예약완료> 기능은 화면 하단의 연도(0000년)를 롱클릭하면 동작하게 한다.

- **init**

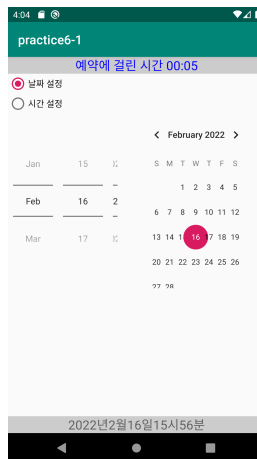
```
datePicker.init(datePicker.year, datePicker.month,datePicker.dayOfMonth) { datePicker, i: Int, i1: Int, i2: Int ->
    selectYear = i
    selectMonth = i1 + 1
    selectdayOfMonth = i2
}
```

- **init, onChanged**

```
override fun onChanged(p0: DatePicker?, p1: Int, p2: Int, p3: Int) {
    selectYear = p1
    selectMonth = p2 + 1
    selectdayOfMonth = p3
}

datePicker.init(datePicker.year, datePicker.month,datePicker.dayOfMonth, this)
```

- ▼ 결과



5. ProgressBar

- 작업의 진행 상태를 바(bar) 또는 원 형태로 제공
- 바 형태
 - 어느 정도 진행되었는지를 확인 가능
- 원 형태
 - 현재 진행 중이라는 상태만 확인 가능
- 속성
 - max : 범위 지정
 - progress : 시작 지점을 지정 (= 현재 지정값)
 - secondaryProgress : 두 번째 프로그레바 지정

6. SeekBar

- 사용자가 터치로 임의 조절이 가능

7. RatingBar

- 진행 상태를 별 모양으로 표시
- 속성
 - numStars : 별의 개수
 - rating : 초깃값 지정
 - stepSize : 한 번에 채워지는 개수를 정함

▼ 기타 위젯 활용

▼ XML 코드 - SeekBar, View

```
<SeekBar
    android:id="@+id/redBar"
    android:max="255"
    android:progress="20" />

<TextView
    android:id="@+id/redText"
```

```

        android:text="red 값 : " />

<Button
    android:id="@+id/btnOk"
    android:text="확인" />

<View
    android:id="@+id/dpView"
    android:background="#000000"
    android:layout_margin="20dp"
    android:layout_width="match_parent"
    android:layout_height="200dp" />

</LinearLayout>

```

▼ Kotlin 코드 - `setOnSeekBarChangeListener` `setOnClickListener`

```

class MainActivity : AppCompatActivity() {
    var red = 0
    var green = 0
    var blue = 0
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        red = redBar.progress
        green = greenBar.progress
        blue = blueBar.progress
        redText.text= "red 값 : " + red
        greenText.text= "green 값 : " + green
        blueText.text= "blue 값 : " + blue

        var seekBarInstance = NoNameSeekBarListener()
        redBar.setOnSeekBarChangeListener(seekBarInstance)

        greenBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
            override fun onProgressChanged(p0: SeekBar?, p1: Int, p2: Boolean) {
                green = p1
                greenText.text= "그린값 : " + green
            }
            override fun onStartTrackingTouch(p0: SeekBar?) {
            }
            override fun onStopTrackingTouch(p0: SeekBar?) {
            }
        })

        blueBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
            override fun onProgressChanged(p0: SeekBar?, p1: Int, p2: Boolean) {
                blue = p1
                blueText.text= "블루값 : " + blue
            }

            override fun onStartTrackingTouch(p0: SeekBar?) {
            }

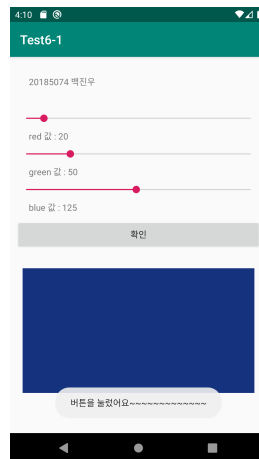
            override fun onStopTrackingTouch(p0: SeekBar?) {
            }
        })

        btnOk.setOnClickListener(object : View.OnClickListener {
            override fun onClick(p0: View?) {
                dpView.setBackgroundColor(Color.rgb(red, green, blue))
                Toast.makeText(applicationContext, "버튼을 눌렀어요", Toast.LENGTH_LONG).show()
            }
        })
    }

    inner class NoNameSeekBarListener : SeekBar.OnSeekBarChangeListener {
        override fun onProgressChanged(p0: SeekBar?, p1: Int, p2: Boolean) {
            red = p1
            redText.text= "red 값 : " + red
        }
        override fun onStartTrackingTouch(p0: SeekBar?) {
        }
        override fun onStopTrackingTouch(p0: SeekBar?) {
        }
    }
}

```

▼ 결과



8. 람다식 표현

- OnClickListener - 추상 메소드 onClick

```
btnOk.setOnClickListener (object : View.OnClickListener {
    override fun onClick(p0: View?) {
        dpView.setBackgroundColor(Color.rgb(red, green, blue))
        Toast.makeText(applicationContext, "버튼을 눌렀어요", Toast.LENGTH_LONG).show()
    }
})
```

- OnSeekBarChangeListener

```
var seekBarInstance = NoNameSeekBarListener()
redBar.setOnSeekBarChangeListener(seekBarInstance)

inner class NoNameSeekBarListener : SeekBar.OnSeekBarChangeListener {
    override fun onProgressChanged(p0: SeekBar?, p1: Int, p2: Boolean) {
        red = p1
        redText.text= "red 값 : " + red
    }
    override fun onStartTrackingTouch(p0: SeekBar?) {
    }
    override fun onStopTrackingTouch(p0: SeekBar?) {
    }
}
```

- OnSeekBarChangeListener - 추상 메소드 3개 (람다식 사용 X)

```
greenBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
    override fun onProgressChanged(p0: SeekBar?, p1: Int, p2: Boolean) {
        green = p1
        greenText.text = "그린값 : " + green
    }
    override fun onStartTrackingTouch(p0: SeekBar?) {
    }
    override fun onStopTrackingTouch(p0: SeekBar?) {
    }
})
```

9. FrameLayout → ScrollView

- 위젯이나 레이아웃이 화면에 넘칠 때 스크롤 효과
- 수직(위아래)으로 스크롤하는 기능
 - 수평(좌우)으로 스크롤하는 수평 스크롤 뷰(HorizontalScrollView)

- 단 하나의 위젯만 넣을 수 있음

```
<ScrollView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >

        <Button
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:text="버튼 1" />
        <!-- 생략 (버튼 7개) -->
    </LinearLayout>
</ScrollView>
```

10. ViewGroup → SlidingDrawer

- 위젯을 서랍처럼 열어서 보여주거나 닫아서 감춤

```
<LinearLayout>
    <TextView
        android:text="여기는 서랍 밖입니다." />
    <SlidingDrawer
        android:content="@+id/content"
        android:handle="@+id/handle" >

        <Button
            android:id="@+id/handle"
            android:text="서랍 손잡이" />

        <LinearLayout
            android:id="@+id/content"
            android:background="#00FF00"
            android:gravity="center" >

            <TextView
                android:text="여기는 서랍 안입니다." />
        </LinearLayout>
    </SlidingDrawer>
</LinearLayout>
```

11. FrameLayout → ViewAnimator → ViewFlipper

- 안에 여러 개의 위젯을 배치하고 필요에 따라 화면을 왼쪽 또는 오른쪽으로 밀어서 위젯을 하나씩 보여주는 방식의 뷰 컨테이너

```
<LinearLayout
    <LinearLayout
        android:orientation="horizontal" >
        <Button
            android:id="@+id/btnPrev"
            android:text="이전화면" />
        <Button
            android:id="@+id/btnNext"
            android:text="다음화면" />
    </LinearLayout>

    <ViewFlipper
        android:id="@+id/viewFlipper1">
        <LinearLayout
            android:background="#FF0000">
        </LinearLayout>
        <LinearLayout
            android:background="#00FF00">
        </LinearLayout>
    </ViewFlipper>
</LinearLayout>
```

```
btnPrev.setOnClickListener {
    vFlipper.showPrevious()
}
```

▼ 직접 풀어보기 6-2)

▼ XML 코드

```
<LinearLayout>
    <Button
        android:id="@+id/btnPrev"
        android:layout_weight="1"
        android:text="사진보기 시작"/>
    <Button
        android:id="@+id/btnNext"
        android:layout_weight="1"
        android:text="사진보기 정지"/>
</LinearLayout>

<ViewFlipper
    android:id="@+id/viewFlipper1"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/pic1"
        android:src="@drawable/pic1" />

    <ImageView
        android:id="@+id/pic2"
        android:src="@drawable/pic2" />
</ViewFlipper>
</LinearLayout>
```

▼ Kotlin 코드

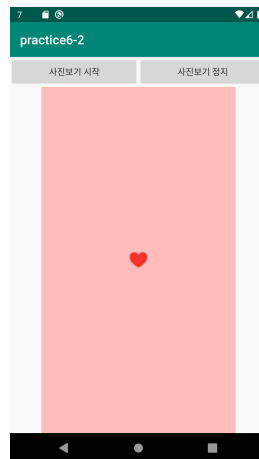
- btnPrev.setOnClickListener(object : View.OnClickListener {
- setFlipInterval, startFlipping

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btnPrev.setOnClickListener(object : View.OnClickListener {
            override fun onClick(p0: View?) {
                viewFlipper1.setFlipInterval(1000)
                viewFlipper1.startFlipping()
            }
        })
        btnNext.setOnClickListener(object : View.OnClickListener {
            override fun onClick(p0: View?) {
                viewFlipper1.stopFlipping()
            }
        })
    }
}
```

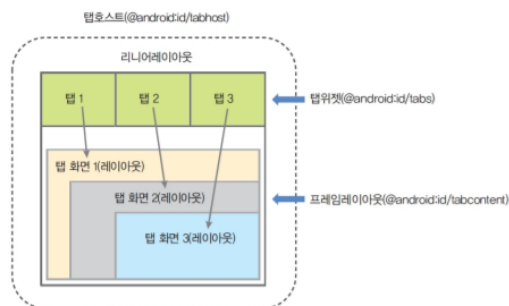
▼ 결과



12. TabHost

- 여러 탭을 두고 각 탭을 클릭할 때마다 해당 화면이 나오도록 설정하는 뷰 컨테이너

▼ 탭호스트의 구성 방식



▼ 예제 6-17) TabHost

▼ XML 코드 - TabWidget, FrameLayout,

```
<TabHost>
  <LinearLayout
    android:id="@+id/linearLayout1">

    <TabWidget
      android:id="@android:id/tabs">
    </TabWidget>

    <FrameLayout
      android:id="@android:id/tabcontent"
      android:layout_width="match_parent"
      android:layout_height="match_parent" >

      <LinearLayout
        android:id="@+id/tabSong"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#f00000"
        android:orientation="vertical">
      </LinearLayout>

      <LinearLayout
        android:id="@+id/tabArtist"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#f0f000"
        android:orientation="vertical">
      </LinearLayout>
    </FrameLayout>
  </LinearLayout>
</TabHost>
```

```

        <LinearLayout
            android:id="@+id/tabAlbum"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="#f000ff"
            android:orientation="vertical">
        </LinearLayout>
    </FrameLayout>
</LinearLayout>
</TabHost>

```

▼ Kotlin 코드 - tabHost

- tabHost.newTabSpec("SONG").setIndicator("음악별")
- setContent, addTab

```

class MainActivity : TabActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        var tabHost = this.tabHost

        var tabSpecSong = tabHost.newTabSpec("SONG").setIndicator("음악별")
        tabSpecSong.setContent(R.id.tabSong)
        tabHost.addTab(tabSpecSong)

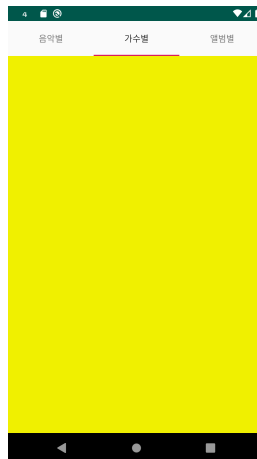
        var tabSpecArtist = tabHost.newTabSpec("ARTIST").setIndicator("가수별")
        tabSpecArtist.setContent(R.id.tabArtist)
        tabHost.addTab(tabSpecArtist)

        var tabSpecAlbum = tabHost.newTabSpec("ALBUM").setIndicator("앨범별")
        tabSpecAlbum.setContent(R.id.tabAlbum)
        tabHost.addTab(tabSpecAlbum)

        tabHost.currentTab = 0
    }
}

```

▼ 결과



```

var tabHost = this.tabHost //탭호스트 변수 생성
// 탭스펙(탭을 구성하는 요소들의 집합) 생성
var tabSpecSong = tabHost.newTabSpec("TAB1").setIndicator("탭에 출력될 글자")
tabSpecSong.setContent(R.id.tabSong) //탭스펙을 탭과 연결
tabHost.addTab(tabSpecSong) //탭을 탭호스트에 부착

```

13-1. actionBar

- 태블릿과 같은 대형 화면에서 여러 화면을 사용하기 위해 고안됨

13-2. fragment

- 대형 태블릿의 화면을 효율적으로 사용하기 위함
- 액티비티보다 작은 단위의 화면
- 대형 화면에서 액티비티 화면을 분할하여 표현
- 소형 화면에서는 화면의 분할 보다는 실행 중에 화면을 동적으로 추가하거나 제거하는데 더 많이 활용

14. ViewGroup → AbsoluteLayout → WebView

- 사용자가 웹브라우저 기능을 앱 안에 직접 포함할 수 있는 위젯

▼ 실습 6-2)

▼ XML 코드

```
<LinearLayout>
    <EditText
        android:id="@+id/edtUrl"
        android:layout_weight="1"
        android:hint="URL을 입력하세요."
        android:singleLine="true" >
    </EditText>

    <Button
        android:id="@+id/btnGo"
        android:text="이동" />

    <Button
        android:id="@+id/btnBack"
        android:text="이전" />
</LinearLayout>
<WebView
    android:id="@+id/webView"
    android:clickable="true" />

</LinearLayout>
```

▼ Kotlin 코드 - WebViewClient, builtInZoomControls, loadUrl

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        webView.webViewClient= CookWebViewClient()

        var webSet : WebSettings = webView.settings
        webSet.builtInZoomControls= true

        btnGo.setOnClickListener{
            webView.loadUrl(edtUrl.text.toString())
        }

        btnBack.setOnClickListener{
            webView.goBack()
        }
    }

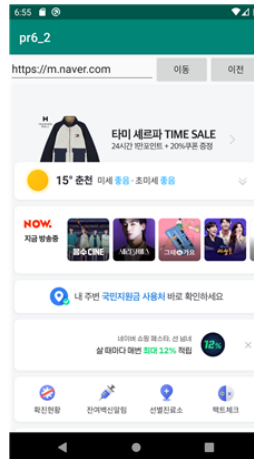
    class CookWebViewClient : WebViewClient() {
        override fun shouldOverrideUrlLoading(view: WebView?, url: String?): Boolean {
            return super.shouldOverrideUrlLoading(view, url)
        }
    }
}
```

▼ AndroidManifest.xml (manifests)

- 4행 : 프로젝트의 인터넷 사용을 허가한다는 의미

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

▼ 결과



15. 자동완성텍스트뷰와 멀티자동완성텍스트뷰

- EditText → EditText.AutoCompleteTextView → EditText.MultiAutoCompleteTextView
- AutoCompleteTextView
 - 단어 1개가 자동 완성
- MultiAutoCompleteTextView
 - 쉼표(,)로 구분하여 여러 개의 단어가 자동 완성

▼ 예제 6-10)

▼ XML 코드 - AutoCompleteTextView, MultiAutoCompleteTextView

```
<AutoCompleteTextView
    android:id="@+id/autoCompleteTextView1"
    android:completionHint="선택하세요"
    android:completionThreshold="2"
    android:hint="자동완성텍스트뷰">
</AutoCompleteTextView>

<MultiAutoCompleteTextView
    android:id="@+id/multiAutoCompleteTextView1"
    android:completionHint="선택하세요"
    android:completionThreshold="2"
    android:hint="멀티자동완성텍스트뷰"/>
```

▼ Kotlin 코드

- ArrayAdapter, CommaTokenizer, setTokenizer, setAdapter

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        var items = arrayOf("CSI-뉴욕", "CSI-라스베가스", "CSI-마이애미", "Friends", "Fringe", "Lost")

        var adapter = ArrayAdapter(this, android.R.layout.simple_dropdown_item_1line, items)
        autoCompleteTextView1.setAdapter(adapter)
    }
}
```

```

    var token = MultiAutoCompleteTextView.CommaTokenizer()
    multiAutoCompleteTextView1.setTokenizer(token)
    multiAutoCompleteTextView1.setAdapter(adapter)
}

```

▼ 결과

