

# 모바일 프로그래밍 입문

## (2019년도 1학기)

---

## 날짜 시간 관련 위젯

## ▶ 아날로그 시계, 디지털 시계

```
java.lang.Object
```

```
    └ android.view.View
```

```
        └ android.widget.AnalogClock
```

```
        └ android.widget.TextView
```

```
            └ android.widget.DigitalClock
```

아날로그시계, 디지털시계 계층도

## ▶ 아날로그 시계

### 시계 관련 XML 코드

```
1 <LinearLayout>
2     <AnalogClock
3         android:layout_width="match_parent"
4         android:layout_height="wrap_content" />
5 </LinearLayout>
```



아날로그 시계



클릭



아날로그 시계



지금 시간은 22 시38분 입니다.

지금 시간은 22 시38분 입니다.

## ▶ 아날로그 시계 (MainActivity.java)

### ▶ AnalogClock ID 선언

▶ `AnalogClock analogTime;`  
선언된 ID

### ▶ 입력된 AnalogClock 찾기

▶ `analogTime = (AnalogClock)findViewById(R.id.analogTime);`  
선언된 ID                      xml에서 AnalogClock ID analogTime 찾기

### ▶ AnalogClock 클릭시 이벤트 발생하기

▶ `analogTime.setOnClickListener(new View.OnClickListener() {  
    // AnalogClock 클릭시 실행행위 감지  
    public void onClick(View v){  
        // 클릭시 실행되는 메소드  
    }  
});`

### ▶ 시간 및 분 가져오기

▶ `final Calendar c = Calendar.getInstance();`  
// 달력 정보 가져오기

`int mHour = c.get((Calendar.HOURLY));`  
// 달력에서 시간 정보 가져오기  
`int mMinute = c.get((Calendar.MINUTE));`  
// 달력에서 분 정보 가져오기

## ▶ 디지털 시계

### 시계 관련 XML 코드

```
1 <LinearLayout>
2     <DigitalClock
3         android:layout_width="match_parent"
4         android:layout_height="wrap_content"
5         android:gravity="center" />
6 </LinearLayout>
```

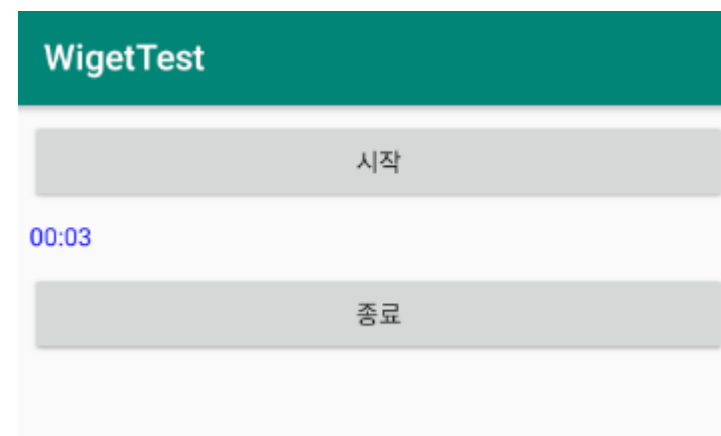
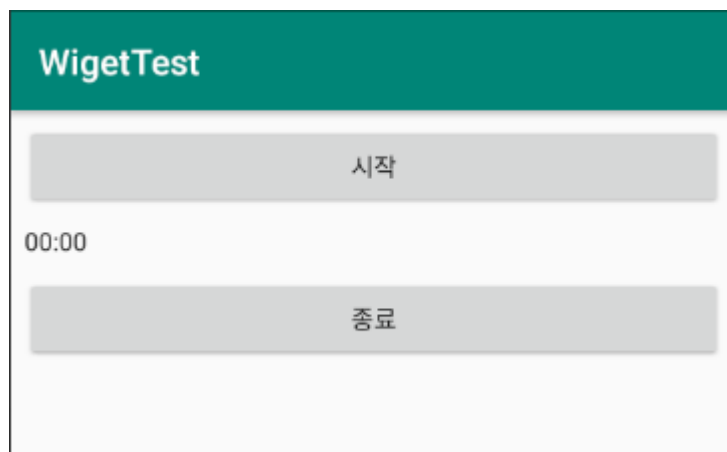
4:54:40 오전

## ▶ 크로노미터

### 크로노미터의 XML 코드

```
1 <LinearLayout>
2     <Chronometer
3         android:id="@+id/chronometer1"
4         android:layout_width="match_parent"
5         android:layout_height="wrap_content"
6         android:format="시간 측정 : %s"
7         android:gravity="center"
8         android:textSize="30dp" />
9 </LinearLayout>
```

시간 측정 : 00:00



## ▶ 크로노미터 시작 종료 (MainActivity.java)

### ▶ Chronometer ID 선언

- ▶ Chronometer `timeResult`;  
선언된 ID

### ▶ 시작 버튼 입력시 Chronometer 실행

- ▶ `timeResult.setBase(SystemClock.elapsedRealtime());`  
`timeResult.start();`  
`timeResult.setTextColor(Color.RED);`

### ▶ 종료 버튼 클릭시 Chronometer 종료

- ▶ `timeResult.stop();`  
`timeResult.setTextColor(Color.BLUE);`

- ▶ 타임피커, 데이트피커, 캘린더 뷰
  - ▶ 타임피커(TimePicker) : 시간을 표시, 조절
  - ▶ 데이트피커(DatePicker) : 날짜를 표시, 조절
  - ▶ 캘린더뷰(CalendarView) : 날짜를 표시, 조절

```
java.lang.Object
```

```
    └ android.view.View
```

```
        └ android.view.ViewGroup
```

```
            └ android.widget.FrameLayout
```

```
                └ android.widget.TimePicker
```

```
                └ android.widget.DatePicker
```

```
                └ android.widget.CalendarView
```

타임피커, 데이트피커, 캘린더뷰 계층도



## ▶ 타임피커 예제

## 타임피커의 XML 코드

```

1 <LinearLayout>
2   <TimePicker
3       android:timePickerMode="spinner"
4       android:layout_width="match_parent"
5       android:layout_height="wrap_content" />
6 </LinearLayout>

```



## Time Picker

12 36

---

1 : 37 AM

---

2 38 PM

시간 분 가져오기

TextView



## Time Picker

12 36 AM

---

1 : 37 PM

---

2 38

시간 분 가져오기

13시37분 선택하셨습니다.

## ▶ 타임피커 시간 분 가져오기 (MainActivity.java)

### ▶ textViw에 값 출력

```
textView.setText(Integer.toString((timePicker.getCurrentHour()))+ "시"  
    + Integer.toString((timePicker.getCurrentMinute())) +"분 선택하셨습니다. ");
```

## ▶ 데이트피커 예제

### 데이트피커의 XML 코드

```
1 <LinearLayout>
2   <DatePicker
3       android:datePickerMode="spinner"
4       android:layout_width="match_parent"
5       android:layout_height="wrap_content" />
6 </LinearLayout>
```

