Using Dialogues

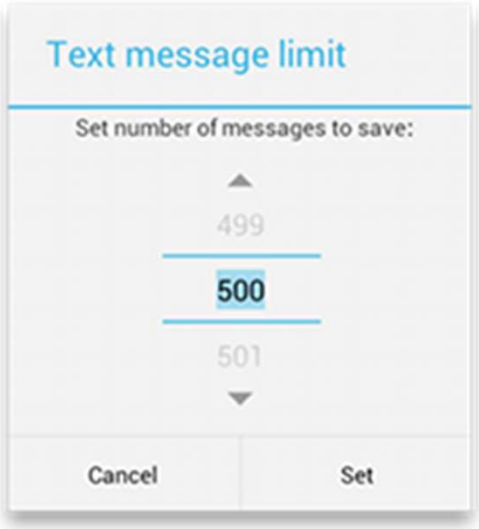
Mobile Software
2019 Fall

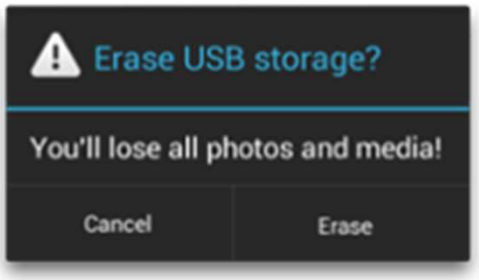
What to do next?

- **Dialog와 AlertDialog**
- Toast와 Snackbar
- DatePicker와 TimePicker
- DatePickerDialog

Dialog (대화 상자)

- 사용자에게 메시지를 전달하거나, 사용자로부터 선택적 입력을 받기 위한 user interface





- 종류
  - **AlertDialog**
  - **ProgressDialog**
  - **DatePickerDialog**
  - **TimePickerDialog**

AlertDialog

- 대화 상자의 기본 클래스는 Dialog 이지만,
  - 사용하기 쉽게 wrapping된 **AlertDialog** 클래스를 주로 사용

- AlertDialog 객체 생성: 내부 클래스인 **Builder** 사용

  val dialog = AlertDialog.Builder (this@MainActivity)
    - **this** 는 대화 상자를 생성하는 부모 activity를 가리킴

  - **AlertDialog.Builder**에서 제공하는 메소드
    - **setMessage, setTitle, setIcon**
    - **create**
    - **show**

실습 준비

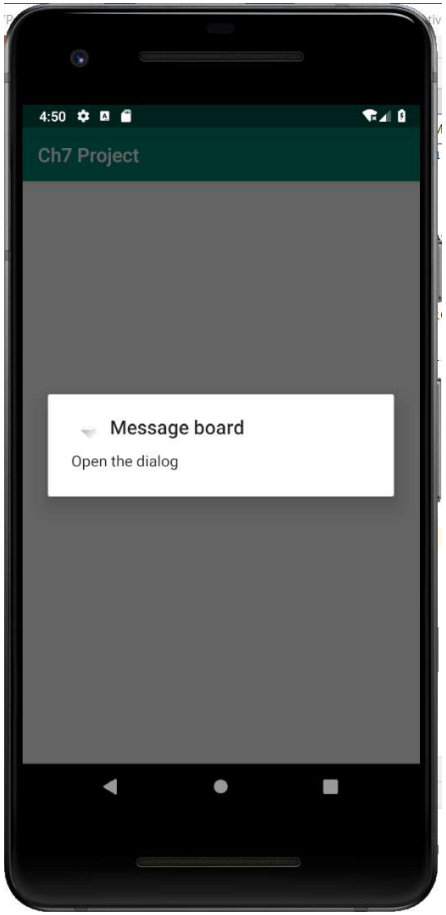
- 새 프로젝트 생성
  - **Application name**
    - **Ch7 project**
  - **Target Android Devices**
    - **Phone and Tablet**
      - minimum SDK **– API 26** 이상
  - **Activity**
    - **Empty Activity**

- 자동 생성된 layout은 **ConstraintLayout**
  - TextView 삭제

실습 1: A Basic AlertDialog

```
class MainActivity : AppCompatActivity() {

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

        callBtn.setOnClickListener {
            val builder = AlertDialog.Builder(this@MainActivity)
            builder.setTitle("Message board")
            builder.setMessage("Open the dialog")
            builder.setIcon(R.drawable.ic_launcher_foreground)
            builder.show()
        }
    }
}
```

applicationContext 를 사용하면
runtime error 발생

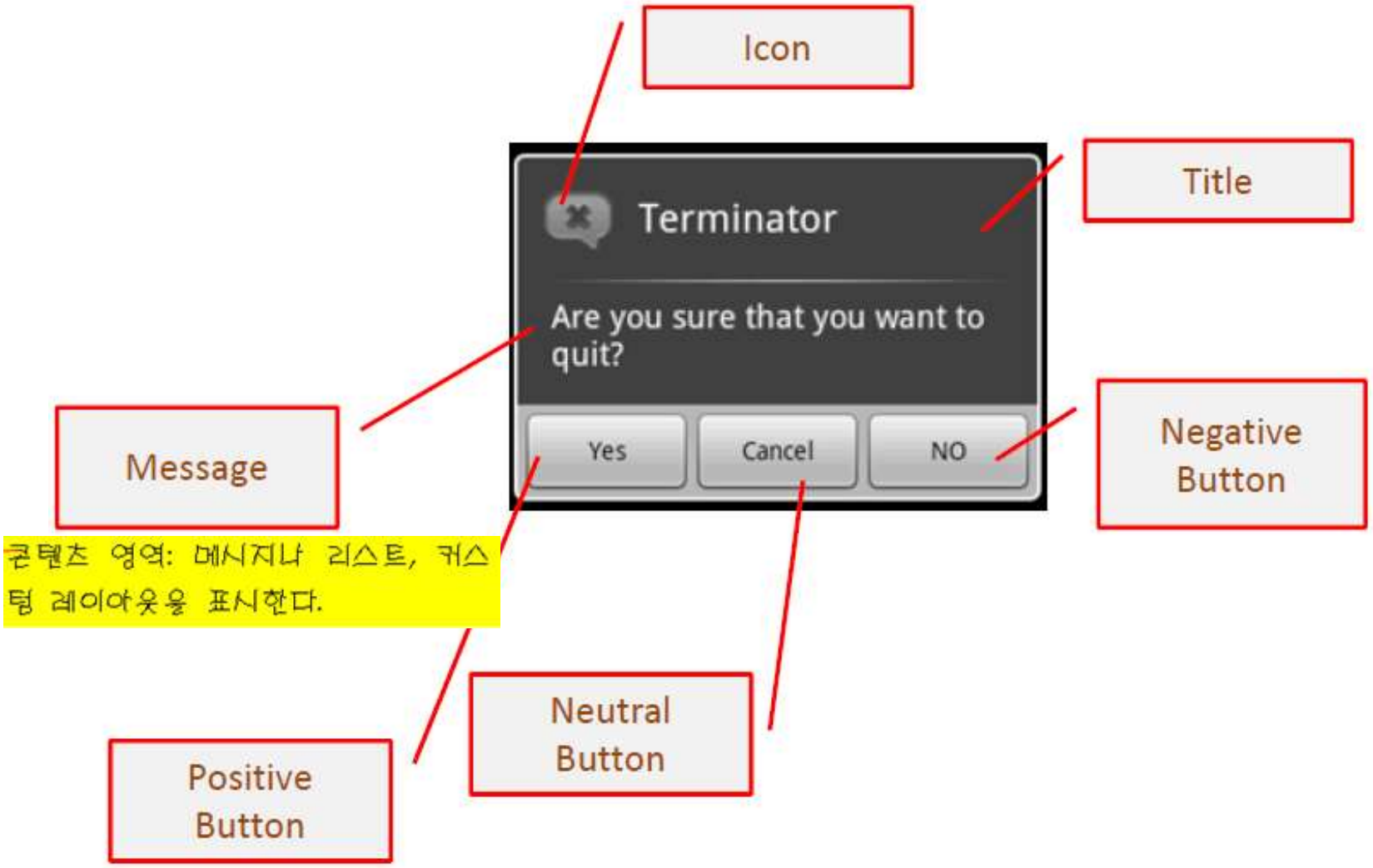


잠깐! A Different Coding Style

```
callBtn.setOnClickListener {
    val builder = AlertDialog.Builder(this@MainActivity)
    builder.setTitle("Message board")
    builder.setMessage("Open the dialog")
    builder.setIcon(R.drawable.ic_launcher_foreground)
    builder.show()
}
```

```
callBtn.setOnClickListener {
    AlertDialog.Builder(this@MainActivity)
        .setTitle("Message board")
        .setMessage("Open the dialog")
        .setIcon(R.drawable.ic_launcher_foreground)
        .show()
}
```

A Complete AlertDialog



실습 1-2: 응답 버튼 추가

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

    callBtn.setOnClickListener {
        AlertDialog.Builder(this@MainActivity)
            .setTitle("Final question")
            .setMessage("Are you sure want to quit?")
            .setIcon(R.drawable.ic_launcher_foreground)
            .setPositiveButton("Yes", null)
            .setNegativeButton("No", null)
            .setNeutralButton("Cancel", null)
            .show()
    }
}
```

실습 1-3: 이벤트 핸들러 추가

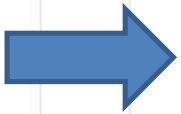
```
callBtn.setOnClickListener {
    AlertDialog.Builder(this@MainActivity)
        .setTitle("Final question")
        .setMessage("Are you sure want to quit?")
        .setIcon(R.drawable.ic_launcher_foreground)
        .setPositiveButton("Yes") { dialog, which ->
            finish()
        }
        .setNegativeButton("No") { _, which ->
            Toast.makeText(applicationContext, "I do not wanto to quit.",
                Toast.LENGTH_SHORT).show()
        }
        .setNeutralButton("Cancel", object:DialogInterface.OnClickListener{
            override fun onClick(dialog: DialogInterface?, which: Int) {
                Toast.makeText(applicationContext,
                    "The command is cancelled.",
                    Toast.LENGTH_SHORT).show()
            }
        })
        .show()
}
```

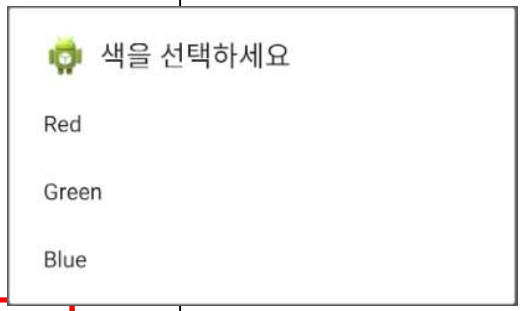
Lambda로 변환하지 않은 코드

무명 클래스로 구현할 때 괄호 없이 중괄호로 시작할 것

실습 1-4: 목록 선택

```
class MainActivity : AppCompatActivity() {

    private val items = arrayOf<String>("Red", "Green", "Blue")

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

        callBtn.setOnClickListener {
            AlertDialog.Builder(this@MainActivity)
                .setTitle("색을 선택하세요")
                .setIcon(R.drawable.ic_launcher)
                .setItems(items) { _, which ->
                    Toast.makeText(applicationContext,
                        "${items[which]} selected",
                        Toast.LENGTH_SHORT).show()
                }
                .show()
        }
    }
}
```





실습 1-5: string-array 리소스 참조

```
class MainActivity : AppCompatActivity() {

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

        val items = resources.getStringArray(R.array.colors)
        callBtn.setOnClickListener {
            AlertDialog.Builder(this@MainActivity)
                .setTitle("색을 선택하세요")
                .setIcon(R.drawable.ic_launcher)
                .setItems(items) { _, which ->
                    Toast.makeText(applicationContext,
                        "${items[which]} selected",
                        Toast.LENGTH_SHORT).show()
                }
                .show()
        }
    }
}
```

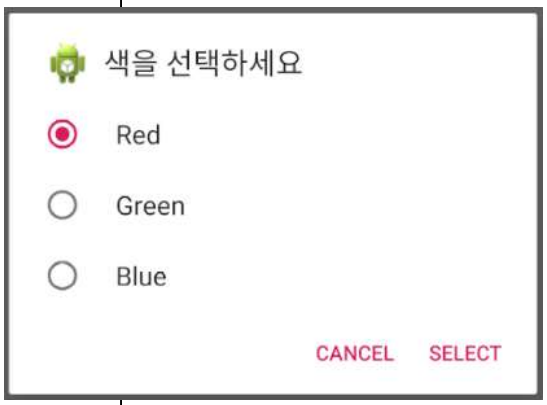
리소스 생성이 끝난 뒤에 리소스 참조를 할 수 있음.

res > **values** > right click
New > Values resource File
"**color_array**" > OK

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="colors">
        <item>Red</item>
        <item>Green</item>
        <item>Blue</item>
    </string-array>
</resources>
```

실습 1-6: SingleChoiceItems

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

    val items = resources.getStringArray(R.array.colors)
    var mSelect = 0

    callBtn.setOnClickListener {
        AlertDialog.Builder(this@MainActivity)
            .setTitle("색을 선택하세요")
            .setIcon(R.drawable.ic_launcher)
            .setSingleChoiceItems(items, mSelect) { _, which ->
                mSelect = which
            }
            .setPositiveButton("Select"){ _, which ->
                Toast.makeText(applicationContext,
                    "${items[mSelect]} selected",
                    Toast.LENGTH_SHORT).show()

            }
            .setNegativeButton("Cancel", null)
            .show()
    }
}
```

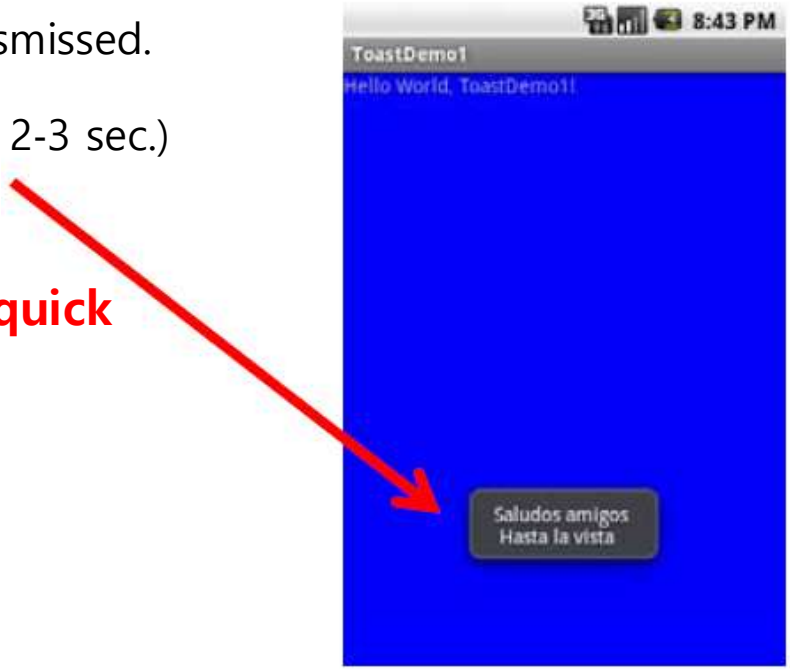


What to do next?

- Dialog와 AlertDialog
- **Toast와 Snackbar**
- DatePicker와 TimePicker
- DatePickerDialog

Toast View(1/2)

- Android provides *two* primitive forms of dialog boxes:
  - **AlertDialog :** shows a floating screen and waits for the user to click on a button to be dismissed.
  - **Toast :** briefly displays a message (about 2-3 sec.) and quietly disappears.
- A **Toast** is a transient view containing a **quick little message** for the user.
- They appear as a **floating view** over the application.
- They *never receive* **focus**.



Toast View(2/2)

```
Toast.makeText ( context,  message,  duration ).show();
```

Context : A reference to the view's environment (what is around me…)

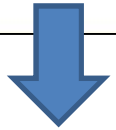
Message : The thing you want to say

Duration : SHORT or LONG exposure

Toast. LENGTH_SHORT
Toast. LENGTH_LONG

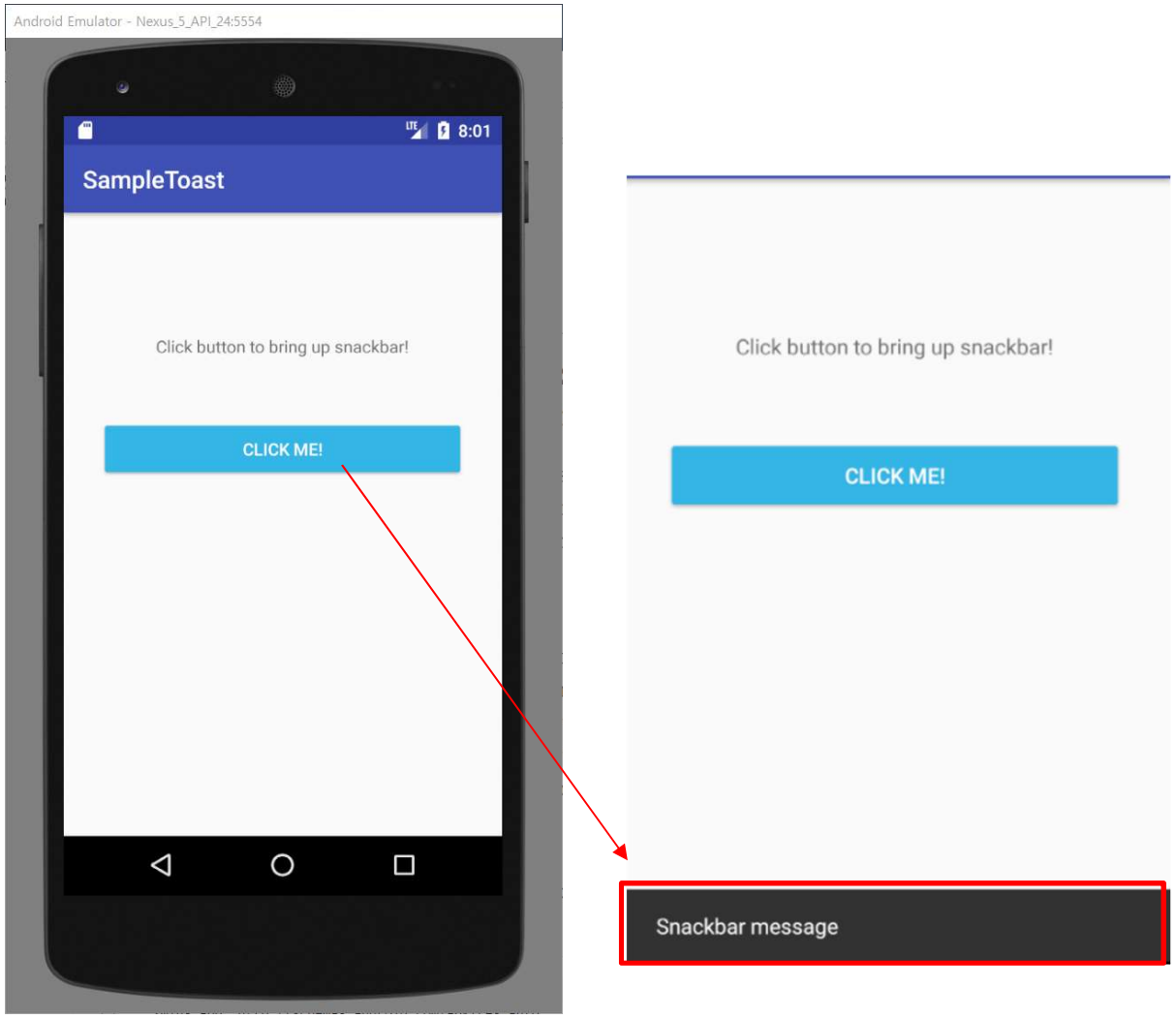
Project Structure 확인하고 가실께요

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation"org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.core:core-ktx:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.android.material:material:1.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
}
```



Sync Now
클릭

Snackbar



- Toast와 기능이나 구현 방식은 비슷함
  - 화면 밑에 banner 형태로 표시됨
  - 간단한 에러 메시지, 경고 메시지, 확인 메시지 등을 보여줄 때 효과적

코드 참조: https://spin.atomicobject.com/2017/07/10/android-snackbar-tutorial/

실습 2: Snackbar

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

        callbackButton.setOnClickListener { v ->
            val view = findViewById<View>(R.id.main_layout_id)
            var message = "Snackbar message"
            var duration = Snackbar.LENGTH_SHORT

            showSnackbar(view, message, duration)
        }
    }

    fun showSnackbar(view: View, message: String, duration: Int) {
        Snackbar.make(view, message, duration).show()
    }
}
```

```
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main_layout_id"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

실습 2-2: Snackbar – action 추가

```
fun showSnackbar(view: View, message: String, duration: Int) {
    val snackbar = Snackbar.make(view, message, duration)

    snackbar.setAction("DISMISS"){ _ ->
        snackbar.dismiss()
    }
    snackbar.show()

}
```

```
snackbar.setAction("DISMISS", object: View.OnClickListener{
    override fun onClick(v: View?) {
        snackbar.dismiss()
    }
})
```

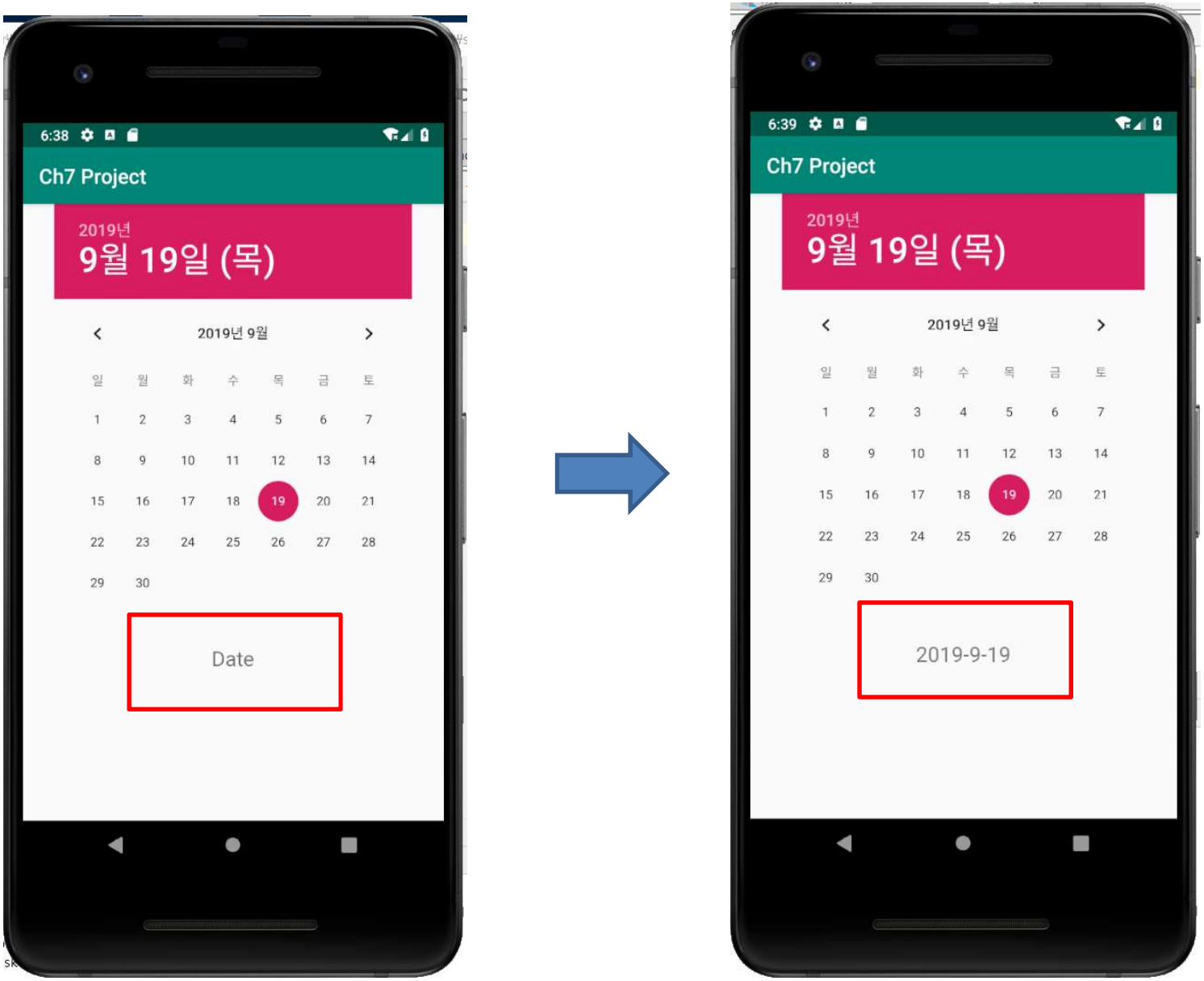
What to do next?

- Dialog와 AlertDialog
- Toast와 Snackbar
- **DatePicker와 TimePicker**
- DatePickerDialog

Date Selection Widgets

- Date Selection
  - Android supports **widgets** (**DatePicker, TimePicker**) and **dialogs** (**DatePickerDialog, TimePickerDialog**)
  - The **DatePicker** and **DatePickerDialog** allow you to set the starting date for the selection, in the form of a year, month, and day.
    - Value of month runs from 0 for January through 11 for December.
  - Each widget provides a *callback* object
    - **DatePicker : OnDateChangedListener**
    - **DatePickerDialog : OnDateSetListener**
    - 날짜 선택 후 대화 창을 닫으면 날짜 정보가 저장되지 않으므로, 선택한 날짜 정보를 저장(**set**)해야 함.

실습 3: DatePicker



실습 3: DatePicker - 레이아웃

```
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context=".MainActivity">

    <DatePicker
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:calendarViewShown="false"
            android:layout_gravity="center"
            android:id="@+id/datePicker"/>
    <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/txtDate"
            android:text="Date"
            android:textSize="20sp"
            android:layout_gravity="center"/>

</LinearLayout>
```

실습 3: DatePicker - 코드

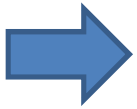
```
class MainActivity : AppCompatActivity() {

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

        val dateChangeListener = DatePicker.OnDateChangedListener{
                _, year, monthOfYear, dayOfMonth ->
            txtDate.text = String.format(
                Locale.KOREA, "%d-%d-%d",
                year, monthOfYear + 1, dayOfMonth
            )
        }
        datePicker.init(
            datePicker.year, datePicker.month,
            datePicker.dayOfMonth, dateChangeListener)
    }
}
```

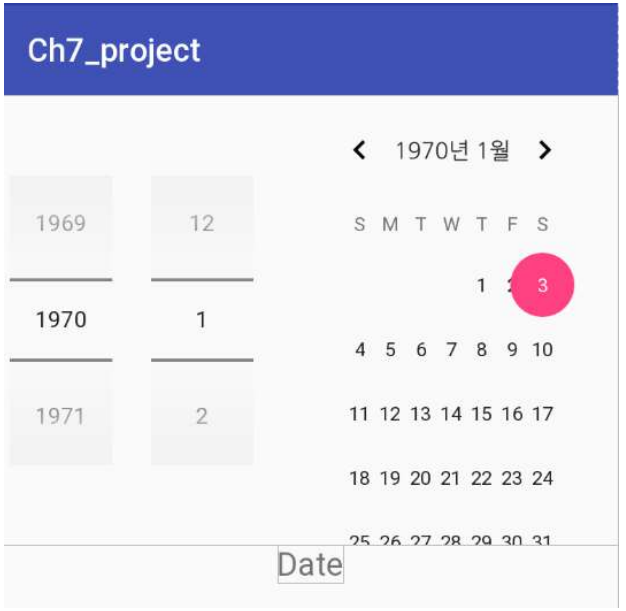
실습 3: DatePicker - 수정

```
class MainActivity : AppCompatActivity() {

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

        datePicker.init(
            datePicker.year,
            datePicker.month,
            datePicker.dayOfMonth,
            DatePickerListener() )
    }

    inner class DatePickerListener : DatePicker.OnDateChangedListener{
        override fun onDateChanged(view: DatePicker?,
                                   year: Int, monthOfYear: Int, dayOfMonth: Int) {
            txtDate.text = String.format(
                Locale.KOREA, "%d-%d-%d",
                year, monthOfYear + 1, dayOfMonth
            )
        }
    }
}
```

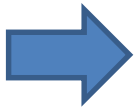
잠깐! 날짜 선택 모양 바꾸기

```
<DatePicker
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:calendarViewShown="true"
    android:datePickerMode="spinner"
    android:id="@+id/datePicker"/>
```






```
<DatePicker
    android:id="@+id/datePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:calendarViewShown="false"
    android:datePickerMode="spinner" />
```



Time Selection Widgets

- Time Selection
  - The widgets **TimePicker** and **TimePickerDialog** let you:

1. set the initial time the user can adjust,

   in the form of an hour (0 through 23) and

   a minute (0 through 59)

2. indicate if the selection should be

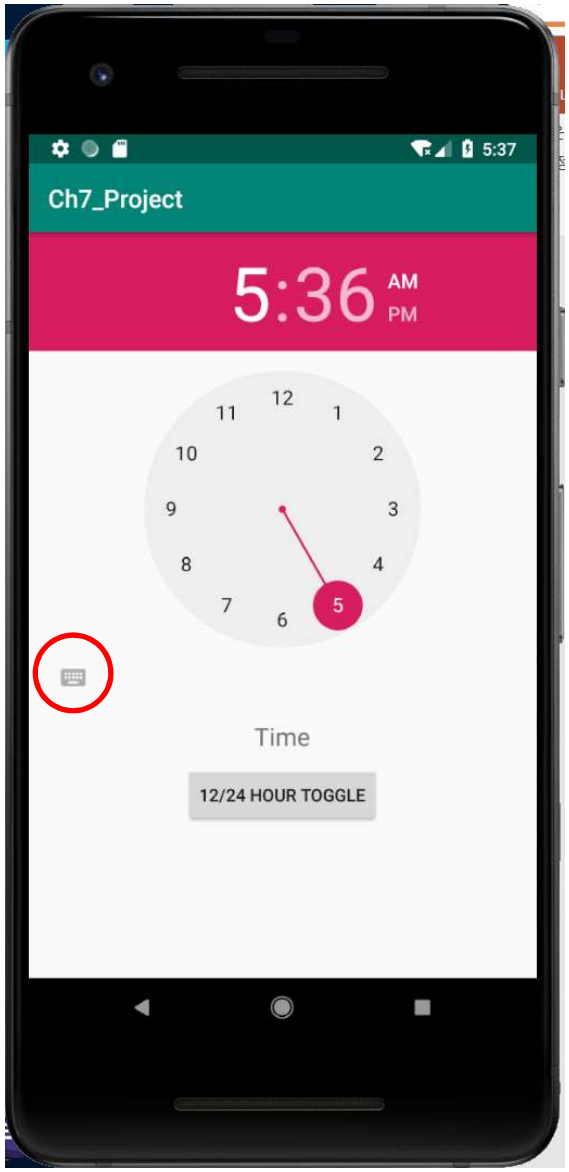
   in 12-hour mode (with an AM/PM toggle), or in 24-hour mode.

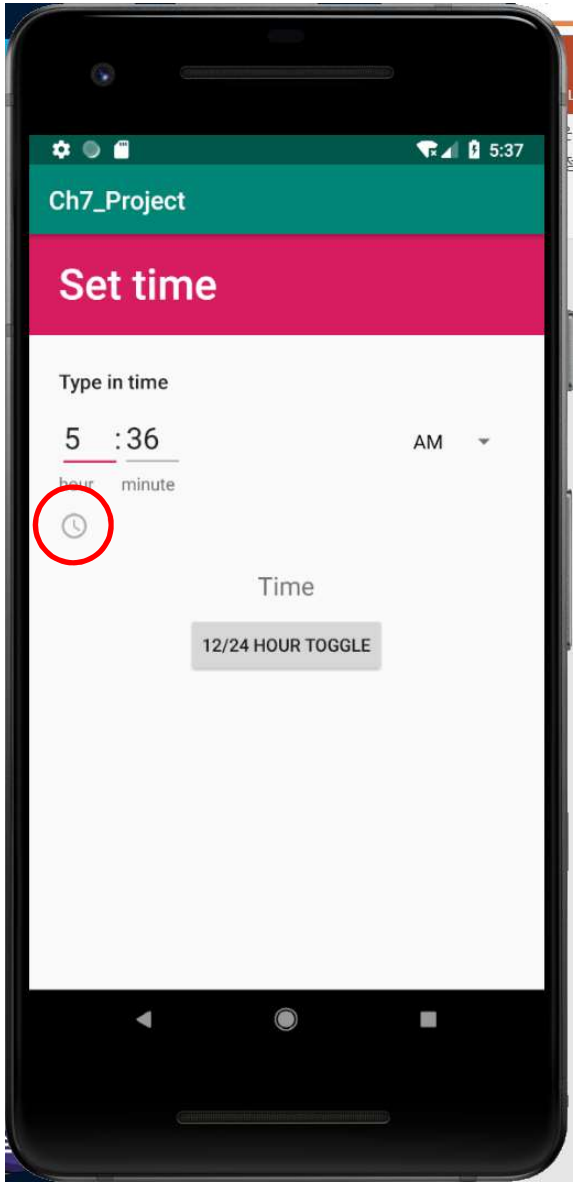
3. provide a callback object :

   TimePicker : OnTimeChangedListener

   TimePickerDialog : OnTimeSetListener

실습 4: TimePicker



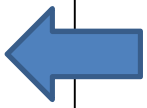


실습 4: TimePicker - 레이아웃

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TimePicker
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/timePicker"/>
    <TextView
        android:id="@+id/txtTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="10dp"
        android:text="Time"
        android:textSize="20sp"/>
    <Button
        android:id="@+id/btnToggle24"
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="12/24 hour toggle"/>

</LinearLayout>
```

실습 4: TimePicker - 코드

```
class MainActivity : AppCompatActivity() {

    var status:Boolean = false

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

        status = timePicker.is24HourView

        timePicker.setOnTimeChangedListener {   , hourOfDay, minute ->
            if (status)
                txtTime.text = "$hourOfDay : $minute"
            else {
                if (hourOfDay > 12)
                    txtTime.text = "${hourOfDay-12}:$minute PM"
                else
                    txtTime.text = "$hourOfDay:$minute AM"
            }
        }

        btnToggle24.setOnClickListener {
            status = !status
        }

    }
}
```



What to do next?

- Dialog와 AlertDialog
- Toast와 Snackbar
- DatePicker와 TimePicker
- **DatePickerDialog**

Date/Time Selection Widgets

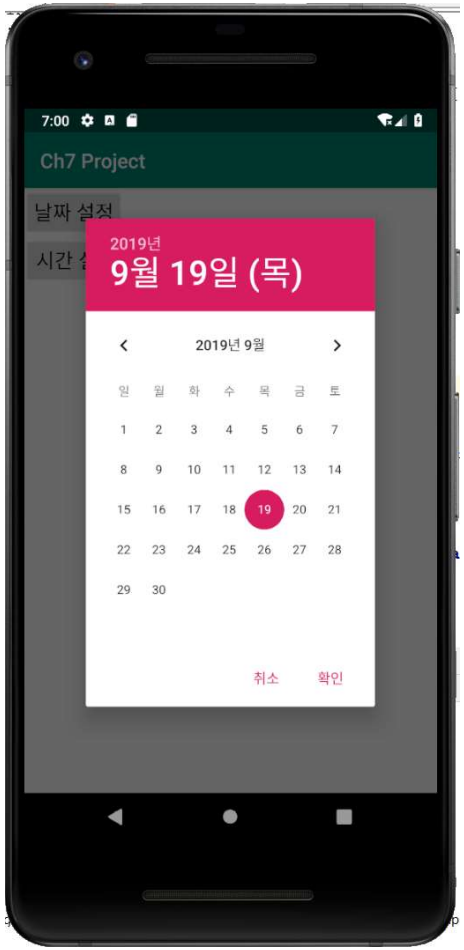
- Widget (**DatePicker, TimePicker**)의 경우
  - 크기가 고정 - 사용하기 편리하지만 공간을 많이 차지.
- Dialog (**DatePickerDialog, TimePickerDialog**) 의 경우
  - widget과 같은 기능이지만, 필요할 때만 dialog 창이 나타남

- 날짜/시간 클래스
  - **Date** 클래스 : 날짜 및 시간도 함께 표시(Time 클래스는 없음)
  - **Calendar** 및 **GregorianCalendar** 클래스
    - 생성자는 외부에서 호출할 수 없음
    - getInstance 메소드를 사용하여 객체를 생성

```
Calendar myCal = Calendar. getInstance( );
Calendar cal = new GregorianCalendar( );
```

DatePickerDialog 와 TimePickerDialog (1/2)

- 날짜와 시간을 입력 받는 대화 상자



DatePickerDialog 와 TimePickerDialog (2/2)

DatePickerDialog (*context :* Context,
listener: **DatePickerDialog.onDateSetListner**,
year: Int,
month: Int,
dayOfMonth: Int)

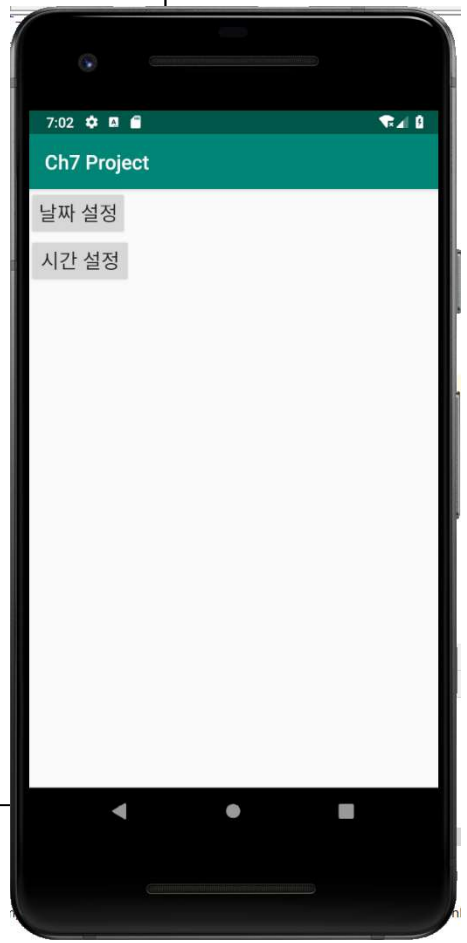
context : the parent context
listener : the listener to call when the user sets the date

TimePickerDialog (*context:* Context,
listener: **TimePickerDialog.onTimeSetListner**,
hourOfDay: Int,
minute: Int,
is24HourView: Boolean)

context : the parent context
listener : the listener to call when the time is set
is24HourView : whether this is a 24 hour view or AM/PM

실습 5: DatePickerDialog - 레이아웃

```
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context=".MainActivity">

    <Button
            android:text="날짜 설정"
            android:textSize="20sp"
            android:padding="10dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/button1"/>

    <Button
            android:text="시간 설정"
            android:textSize="20sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/button2"/>

</LinearLayout>
```



실습 5: DatePickerDialog (1/2) - 코드

```
class MainActivity : AppCompatActivity() {

    val DATE_DIALOG_ID = 0
    val TIME_DIALOG_ID = 1

    val c = GregorianCalendar()
    var mYear: Int = c.get(Calendar.YEAR)
    var mMonth:Int = c.get(Calendar.MONTH)
    var mDay:Int = c.get(Calendar.DAY_OF_MONTH)
    var mHour:Int = c.get(Calendar.HOUR_OF_DAY)
    var mMinute:Int = c.get(Calendar.MINUTE)

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

        button1.setOnClickListener{
            createDialog(DATE_DIALOG_ID).show()}
        button2.setOnClickListener{
            createDialog(TIME_DIALOG_ID).show()}
    }
```

현재 날짜와 시각으로 초기화

날짜 다이얼로그
또는 시간 다이얼로그 생성

실습 5: DatePickerDialog (2/2) - 코드

```
private fun createDialog(id: Int): Dialog {
    if (id == DATE_DIALOG_ID)
        return DatePickerDialog(this,
            mDateSetListener, mYear, mMonth, mDay)
    return TimePickerDialog (this,
            mTimeSetListener, mHour, mMinute, false)
}

private var mDateSetListener =
    DatePickerDialog.OnDateSetListener {
            _, year, monthOfYear, dayOfMonth ->
        showToast("날짜: $year-$monthOfYear-$dayOfMonth")
    }

private var mTimeSetListener =
    TimePickerDialog.OnTimeSetListener {
            _, hourOfDay, minute ->
        showToast("시간: $hourOfDay:$minute")
    }

private fun showToast(str: String) {
    Toast.makeText(
        applicationContext,
        str, Toast.LENGTH_SHORT).show()
}
```

날짜 변경 이벤트 처리
onDateSet

시간 변경 이벤트 처리
onTimeSet

실습 5: DatePickerDialog - 수정

```
private fun createDialog(id: Int): Dialog {
    if (id == DATE_DIALOG_ID)
        return DatePickerDialog(this,
            MyDateSet(), mYear, mMonth, mDay)
    return TimePickerDialog (this,
        MyTimeSet(), mHour, mMinute, false)
}

inner class MyDateSet : DatePickerDialog.OnDateSetListener {
    override fun onDateSet(view: DatePicker?,
                           year: Int, month: Int, dayOfMonth: Int) {
        var m = month + 1
        showToast("날짜: $year-$m-$dayOfMonth")
    }
}

inner class MyTimeSet : TimePickerDialog.OnTimeSetListener {
    override fun onTimeSet(view: TimePicker?,
                           hourOfDay: Int, minute: Int) {
        showToast("시간: $hourOfDay:$minute")
    }
}
```