Custom View와 Toast

- http://developer.android.com

Custom View

- 기존 뷰를 상속 받아 개발자가 새로운 뷰를 구현

- View 클래스를 상속받아 필요한 메소드 재정의
  - XML에서 직접 작성한 View를 사용하기 위해서는 View의 생성자 모두 재정의

```
class MyCustomView : View {
    // 필수 생성자
    constructor(context: Context?) : super(context)

    constructor(context: Context?, attrSet: AttributeSet?): super(context, attrSet)

    constructor(context: Context?, attrSet: AttributeSet?, defStyleAttrSet: Int)
            : super(context, attrSet, defStyleAttrSet)

    // 뷰 모양 그리기 메소드
    override fun onDraw(canvas: Canvas?) {
        // 그리기 코드
    }
}
```

• 기본 생성자 (필수)

레이아웃 XML에서 태그형식으로 추가하고자 할 때는 모든 생성자를 반드시 재정의

뷰가 화면에 그려져야 할 때 호출

그리기 예제

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
//        setContentView(R.layout.activity_main)
        val myView = MyView(this)
        setContentView(myView)
    }

    // nested class 형태로 선언할 경우
    class MyView(context: Context?) : View(context) {
        override fun onDraw(canvas: Canvas?) {
            super.onDraw(canvas)
            canvas?.drawColor(Color.LTGRAY)
            val paint = Paint()
            paint.setColor(Color.BLUE)
            canvas?.drawCircle(200.toFloat(), 200.toFloat(), 100.toFloat(), paint)
        }
    }
}
```

- 내부(아래)에 선언한 MyView 객체 생성 후 setContentView에 설정
- 내부 클래스로 선언 하여 사용할 경우
- 그림이 그려지는 영역
- 그림을 그리는 도구 생성 및 속성 지정
- 캔버스에 지정한 위치와 도구로 그림을 그림
- 매개변수 타입이 float 여야 하므로 타입변환

Custom View를 XML Layout에서 사용하기

Custom View Class 작성 (MyOuterView.kt 추가)

```
class MyOuterView : View {
    constructor(context: Context) : super(context)

    constructor(context: Context, attrs: AttributeSet) : super(context, attrs)

    constructor(context: Context, attrs:AttributeSet, defStyleAttr:Int)
            : super(context, attrs, defStyleAttr)

    override fun onDraw(canvas: Canvas?) {
        super.onDraw(canvas)
        canvas?.drawColor(Color.LTGRAY)
    }
}
```

- View 상속
- 기본 생성자를 모두 재정의

activity_main.xml (주 레이아웃)에 추가

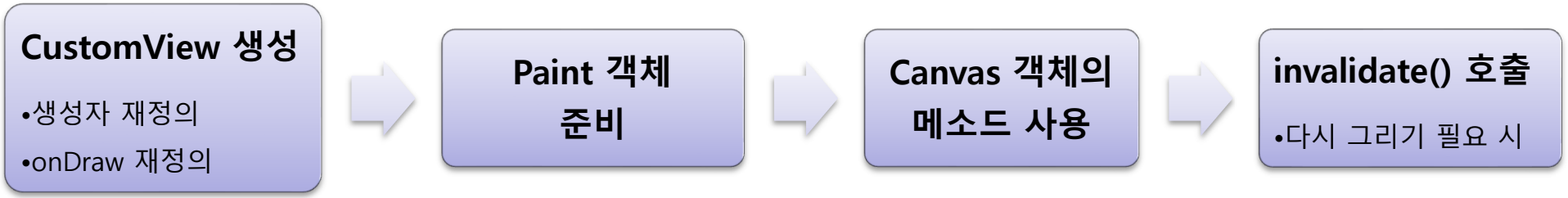
```
<ddwu.com.mobile.customviewtest.MyOuterView
    android:id="@+id/myOuterView"
```

- 반드시 Full Package를 포함한 클래스명으로 기록
- 최신 버전 안드로이드 스튜디오의 경우 Palette 에서 선택 가능

Custom View 그리기 1

Custom View 사용 시의 그리기 절차

- View 클래스 상속 후 생성자, onDraw() 재정의
- onDraw()에 그릴 내용 지정
  - canvas 에 어떠한 형식으로 그릴지 Paint 객체로 지정
  - 매개변수로 전달 받은 canvas의 메소드를 사용하여 그리기 수행
- CustomView 객체를 생성
- 화면을 다시 그릴 필요가 있을 때
  → 커스텀뷰객체.invalidate() 호출
  - 화면을 무효화 한 후 onDraw()가 호출되어 화면이 다시 그려짐



Custom View 그리기 2

Canvas 클래스

- 그림을 그리는 영역을 담당하는 클래스
- Canvas의 메소드를 사용하여 화면에 그리기 지정
- View 클래스의 onDraw() 메소드의 매개변수로 전달
  - 전달받은 Canvas 객체를 수정하여 그려지는 모양 변경
- 기본 도형 출력
  - Canvas의 멤버 메소드 호출
  - 오버로딩 되어 있으므로 매개변수 확인 필요

```
// 점 그리기
Canvas객체.drawPoint (x: Float, y: Float, paint: Paint)
// 선 그리기
Canvas객체.drawLine (startX: Float, startY: Float, stopX: Float, stopY: Float, paint: Paint)
// 원 그리기
Canvas객체.drawCircle (x: Float, y: Float, radius: Float, paint: Float)
// 사각형 그리기
Canvas객체.drawRect (left: Float, top: Float, right: Float, bottom: Float, paint: Float)
// 텍스트 그리기
Canvas객체.drawText (text: String, x: Float, y: Float, paint: Float)
```

Custom View 그리기 2

Paint

- Canvas에 그림을 그리기 위해 사용하는 펜 역할
- 그리기에 대한 속성 정보를 지정
- 선 경계 부드럽게 출력
  - *paint객체*.setAntiAlias(true)
  - new Paint(Paint.ANTI_ALIAS_FLAG)
- 색상 변경
  - 투명도 지정 RGB: *paint객체*.setARGB (a: Int, r: Int, g: Int, b: Int)
  - 색상만 지정: *paint객체*.setColor (int color) // Color.RED 등

토스트(Toast)

- 화면 하단에 일정 시간 동안 나타나는 안내용 출력창
  - 간단한 상태 정보 표시
  - 디버깅 시 값 확인용으로도 사용할 수 있음

- Toast 사용 예

• Context 정보 (Activity가 상속하므로 this 사용 가능)

• 출력 메시지

• Toast 출력 시간

```
// Toast 생성
val myToast = Toast.makeText(this, "Toast Message", Toast.LENGTH_SHORT)
// myToast.setGravity(Gravity.CENTER_HORIZONTAL, 0, 0)
// myToast.setMargin(0f, 0f)
myToast.show()  // Toast 출력

// 생성 및 출력
Toast.makeText(this, "Toast Message", Toast.LENGTH_SHORT).show()

// this 객체가 MainActivity (Context) 가 아닐 경우
Toast.makeText(this@MainActivity, "Toast Message", Toast.LENGTH_SHORT).show()
```

• show() 를 실행시켜야 화면에 표시

실습

다음과 같은 레이아웃을 갖는 앱 생성

- 버튼을 누르면 원이 임의의 위치에 새로 그려짐
- 원의 좌표를 Toast 로 출력

- HINT
  - MyView를 XML 레이아웃에 배치하여야 하므로 View의 모든 생성자 재정의

- 최초 실행 시 x좌표 100, y좌표 100 에 cyan 색상의 원 표시

- 노란색 부분은 직접 만든 CustomView (MyCustomView)

- HINT
  - 버튼을 누르면 MyView 다시 그리기 호출

Random 클래스 사용

- x, y, r 의 값을 random 지정

```
// Random 객체 생성 (import 필요)
val random = Random()
// 0 ~ 500 생성
val x = random.nextInt(500)
val y = random.nextInt(500)
// 100~200 생성
val r = (random.nextInt(2) + 1) * 100
```