* 안드로이드 개발에 사용되는 언어

Java, 코틀린

둘 다 같이 사용가능

* 프로젝트 시작 시 입력하는 것들 중 bundle ID (package name)의 구성 방법

Organization name(회사) + application name(앱)

* 프로젝트 시작 시 생성되는 파일들에 대하여

Java, xml, icon png, icon xml, build script, manifests

* Android Studio & Gradle build system
* Android API Level, Version Number, Code name 등

1.5 ~ 숫자에 따라 디저트 이름을 붙여 사용

Android 1.5(cupcake) : API Level1 계속 올라감

Build Level : 설정한 버전으로 build 하겠다는 것

API Level : 개발자들이 사용하는 버전. 정수로 증가

Version Number : 사람들이 아는 버전

Code name : 디저트 이름으로 사용

* 실기기를 개발용으로 전환하는 방법

기기를 개발자모드로 설정 후 컴퓨터와 연결하여 사용

* Bitmap Resource 사용법 및 다양한 해상도에 대해 대비하기

해상도 별, 언어 별, 기종 별로 이미지를 준비

* String Resource 사용법 및 다양한 언어에 대해 대비하기

String.xml

I18n(internationalization) : 리소스를 빼는 작업

L10n(localization) : 영어로 되어 있는 프로그램을 한국어로 지원하게 하는 것

* Resource Filename 사용규칙

Filename이 java identifier 규칙

숫자 사용 금지, 소문자만 사용 가능 등

* Layout Managers, Units (단위),
* View ID 및 Runtime에 이 값들을 활용하기

소스 코드 상에서 해당 컨트롤을 접근하기 위해 사용

* findViewById의 비용

생성 시점에서 사용하는 것이 좋음

* XML의 기본 구조, 구성 요소. Elements/Tags, Attributes, DTD, Schema

< : Elements/Tags

Android::textSize 등 : Attributes

DTD : xml문서를 정의하는 문서

Schema : DTD를 다시 xml로 정의한 거

* JSON의 기본 구조, 구성 요소. Key/Value. Boolean/Number/String/Array/Object

Object 혹은 Array로 정의

Object : key(문자열 사용; value

Array :

* View/Component/Widget properties: TextView, Button, ImageView, EditText, CheckBox, Slider, ...
* Log Levels
* View.tag

getTag, setTag

* Padding vs Margin
* Activity Restart on Configuration Change

Configuration Change : 언어 변경, 가로세로 변경 등

Screen orientation : 가로모드 혹은 세로모드로만 사용되도록

* Garbage Collection in Java

할당된 메모리 알아서 처리

* Canvas save()/restore(), Transform matrix

Transform matrix 3x3 행렬을 2차원으로 변환

모든 뷰가 행렬을 하나 가지고 있음

canvas에서 이동, 회전 등 가능

스택 구조로 save, restore

* Event-driven programming

프로그램 실행 주도권을 코드에서 가지는 것

Event-driven : 프로그램 실행 주도권을 시스템이 가짐

* Android Studio Tool 사용법
  + Alt+Enter: Import, Error correction, ...
  + Refactoring: Extract/Introduce, ...

Pull up : 하위 클래스의 함수를 상위 클래스로 올리는 것 반대) pull down

* + Alt+1, Shift+ESC, Shift+Ctrl+Backspace, ...
* Thread Safety with Android

Thread Safety는 꼭 필요한 곳에서만 사용

Android는 Thread Safety x

* Design Patterns: Builder, Singleton, Factory, ...

Builder

Singleton : 한 개의 객체만 존재하도록

Factory :

* Game Loop
* 수업 중 진행했던 내용들에 대한 Review
  + ImageSwitcher: Button, LinearLayout, interface as parameter, SharedPreferences, ...
  + Cards: Image Resource, Toast, AlertDialog, Multi-language,  ...
  + MoreControls: Text Watcher, Custom View, Paint, ...
  + SampleGame: Choreographer/VSync, GameObject, Touch event, FPS, Rotation, ...
  + DragonFlight: BitmapPool, package refactor, dimens.xml, Activity Life-cycle, Collision, ...