

1. 기본 naming들

1.1 Java class Naming(Camel case)

자바 클래스는 반드시 대문자로 시작하여야하며, 해당 클래스가 어떤 목적으로 만들어졌는지 유추할 수 있도록 명사를 활용하여 사용합니다.
만약 다양한 단어가 쓰일 경우, UserProfileManager 이런식으로 각 단어의 첫 시작을 대문자로 작성합니다.

클래스 내의 함수의 경우 소문자로 시작하며, 첫 단어 이후로 따라오는 단어의 첫 글자는 대문자로 씁니다.
메소드는 동사로 시작하는 것이 일반적입니다. 예) printResult()

인터페이스의 경우 대문자로 시작하며, 형용사를 주로 사용합니다. 예) ActionListener

카멜케이스

클래스, 인터페이스, 메소드, 변수 등의 이름을 명명할 때 카멜케이스 구문을 따르는데 이는 이름에 두 가지 이상의 단어가 혼합되어있을 경우 두 번째 단어의 첫 글자는 대문자로 표시하는 것을 뜻합니다.

1.2 xml Naming

ANDROID RESOURCE NAMING CHEAT SHEET

by @MOLS.JEROEN

<WHAT>

<WHERE>

<DESCRIPTION>

<SIZE>

fixed set of options
choose the right one below

custom part Android view subclass
"all" if reused in * screens

differentiate multiple elements in one screen

always optional
[xdp] or bucket [small]

LAYOUTS <WHAT>_<WHERE>.XML	<WHAT> is activity, fragment, view, item or layout e.g. activity_main.xml
STRINGS <WHERE>_<DESCRIPTION>	e.g. main_intro all_done
DRAWABLES <WHERE>_<DESCRIPTION>_<SIZE>	e.g. all_infoicon_small main_background
IDS <WHAT>_<WHERE>_<DESCRIPTION>	<WHAT> is name of Android/Custom view class e.g. linearlayout_main_fragmentcontainer
DIMENSIONS <WHAT>_<WHERE>_<DESCRIPTION>_<SIZE>	<WHAT> is width, height, size, margin, padding, elevation, keyline or textsize e.g. keyline_all_text

xml에서의 naming 규칙은 < WHAT>< WHERE>< DESCRIPTION>< SIZE>로 이루어져 있습니다.

< what>은 자원이 실제로 무엇을 나타내는 지 나타냅니다. ex) activity_main.xml

Prefix	Usage
activity	contentview for activity
fragment	view for a fragment
view	inflated by a custom view
item	layout used in list/recycler/gridview
layout	layout reused using the include tag

< where>은 논리적으로 앱이 속해있는 위치를 나타냅니다.

여러 화면에서 사용되는 리소스일 경우 all을 사용합니다.

< description>은 한 화면에서 사용되는 여러 요소들을 구분합니다. ex) title, content

< size> 해당 태그의 상세 정보(24dp)등을 나타냅니다.

레이아웃의 경우 < what>_ < where>.xml

- activity_main : MainActivity의 content view
- fragment_articledetail : ArticleDetailFragment의 뷰
- view_menu : 인플레이드되는 MenuView (커스텀 뷰)
- item_article : ArticleRecyclerView의 list item
- layout_actionbar_backbutton : back버튼이 있는 액션바 레이아웃

String의 경우 < where>_ < description> (어디서 사용되는가)

- articledetail_title : ArticleDetailFragment의 제목
- feedback_explanation : FeedbackFragment의 피드백 설명
- feedback_namehint : FeedbackFragment의 이름 필드 힌트
- all_done : 일반적인 "완료" 문자열

Drawable의 경우 < where>_ < description>_ < size>

- articledetail_placeholder : ArticleDetailFragment의 placeholder
- all_infoicon : 일반적인 info 아이콘
- all_infoicon_large : large 버전의 info 아이콘
- all_infoicon_24dp : 24dp 버전의 info 아이콘

각 위젯의 id의 경우 < what>_ < where>_ < description>

- tv_menu_title
- imageview_menu_profile

dimension의 경우 < what>_ < where>_ < description>_ < size>

Prefix	Usage
width	width in dp
height	height in dp
size	if width == height
margin	margin in dp
padding	padding in dp
elevation	elevation in dp
keyline	absolute keyline measured from view edge in dp
textsize	size of text in sp

- height_toolbar : 모든 툴바의 높이
- textsize_medium : 모든 텍스트의 중간 크기
- size_menu_icon : 메뉴 아이콘 크기

1.3 변수 Naming

변수 또한 메소드와 유사하게 카멜케이스를 적용하여 작성합니다. 예) var numCount

상수의 경우 모든 글자를 대문자로 표현하며, 여러 단어가 있을 경우 언더바_를 활용하여 구분합니다.

예) USER_PROFILE_MANAGER

2. 기본 Widget 들

2.1 Widget 사용 방법 (Widget의 사용방법을 알아야 실습하기 수월해요)

2.2 Widget 실습

- 실습한걸 Github에다가 올려주세요~
- toast 버튼 클릭 시 toast 메시지 띄우기 (ex) number is 15)
- count 버튼 클릭 시 숫자 1씩 증가
- random 버튼 클릭 시 두번째 Activity로 전환하여 (0 ~ count된 값) 중 random number 출력
- (Click event를 처리하는 방법은 여러가지가 있어요 여러방법을 알아보고 실습을 해봅시다~)