

S H O U T . . O U R . . P A S S I O N . . T O G E T H E R

21<sup>st</sup> UNI SOPT

# 5차 세미나

SOPT'

# Contents

21<sup>st</sup> UNI SOPT

## 5차 세미나

### 1. Today 1

- 싱글トン패턴
- 스플래쉬
- 폰트
- 아이콘
- 이런 것은..?

### 2. Today 2

- 해상도
- px의 이해
- dp의 이해
- sp의 이해

### 3. Today 3

- 디자이너와 대화하기

### 4. 합동~

- 조는 세미나 때 공개^~^

SOPT

## 5 차 세 미 나

01. Today1

# 싱글톤

- Application 클래스를 상속받는 클래스
  - Application클래스는 모든 액티비티가 공동으로 사용
  - 즉 해당 클래스에 있는 데이터를 모든 액티비티가 접근할 수 있음.  
(오늘 할 폰트나 향후 할 네트워크 등록이 여기에 속한답니다)
- Manifest에 해당 클래스 등록해주어야 함

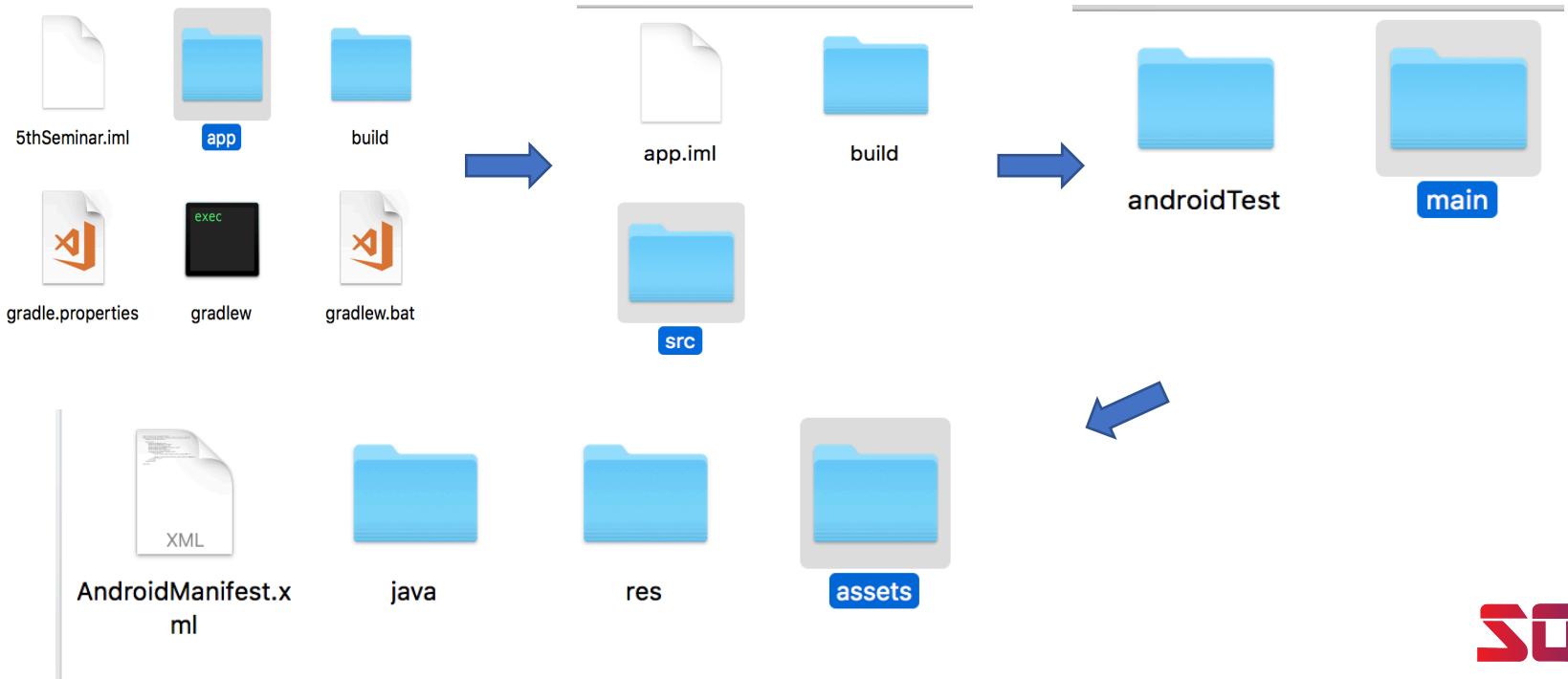
```
<application  
    android:allowBackup="true"  
    android:name=".ApplicationController"  
    android:icon="@mipmap/ic_launcher"  
    android:label="@string/app_name"  
    android:roundIcon="@mipmap/ic_launcher_round"
```



- 백그라운드에 있지 않는 어플리케이션을 실행했을 때  
**맨 처음 나오는 화면**(즉 로그인 전에 나오는 화면)
- 사용자로 하여금 앱이 실행되고 있음을 보여줌
- 로그인 전에 서버와 통신해야 할 부분이나 기타 처리할 데이터가 있으면 이 화면에서 처리함

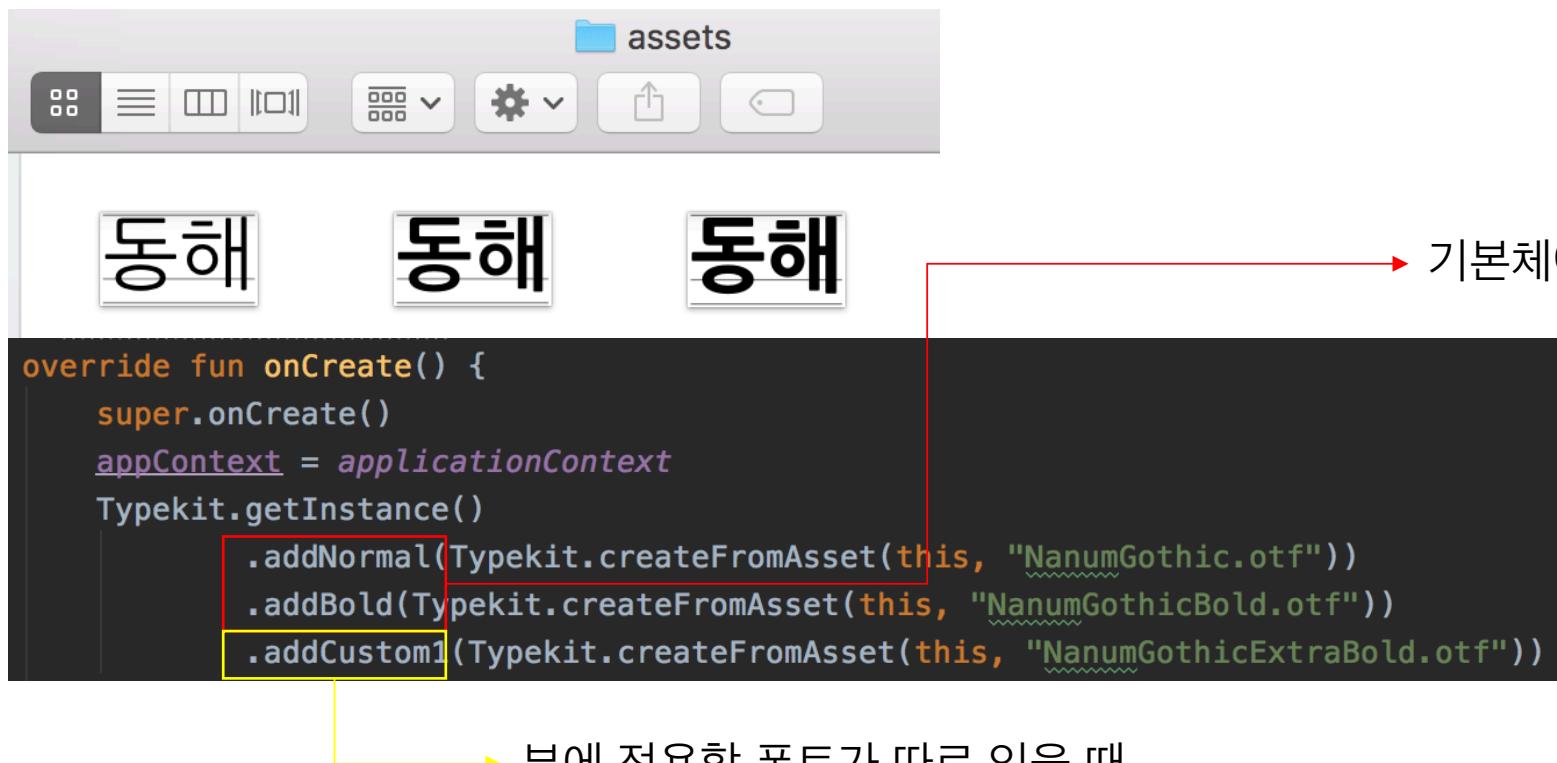
## 폰트

- 그레들에 `compile 'com.tsengvn:Typekit:1.0.0'` 추가
- 프로젝트 폴더->app->src->main에서 assets라는 이름의 폴더 생성



### 폰트

- assets 폴더 안에 앱에 적용할 폰트를 넣는다.
- ApplicationController 안에 onCreate함수에서 해당 폰트를 적용



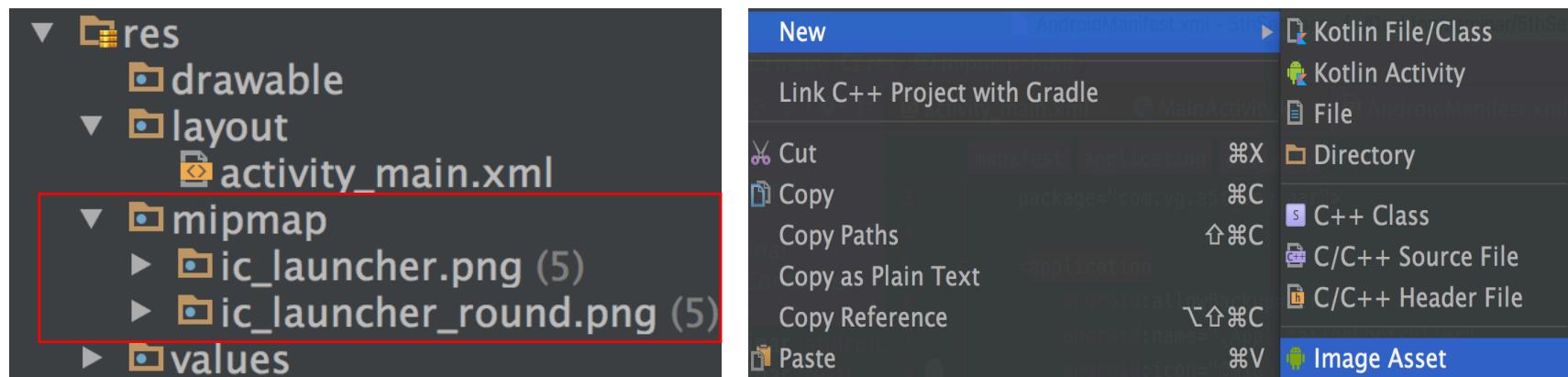
### 폰트

- 폰트를 적용할 Activity에 해당 폰트를 붙여 줌
  - attachBaseContext 함수를 override

```
override fun attachBaseContext(newBase: Context) {  
    super.attachBaseContext(TypekitContextWrapper.wrap(newBase))  
}
```

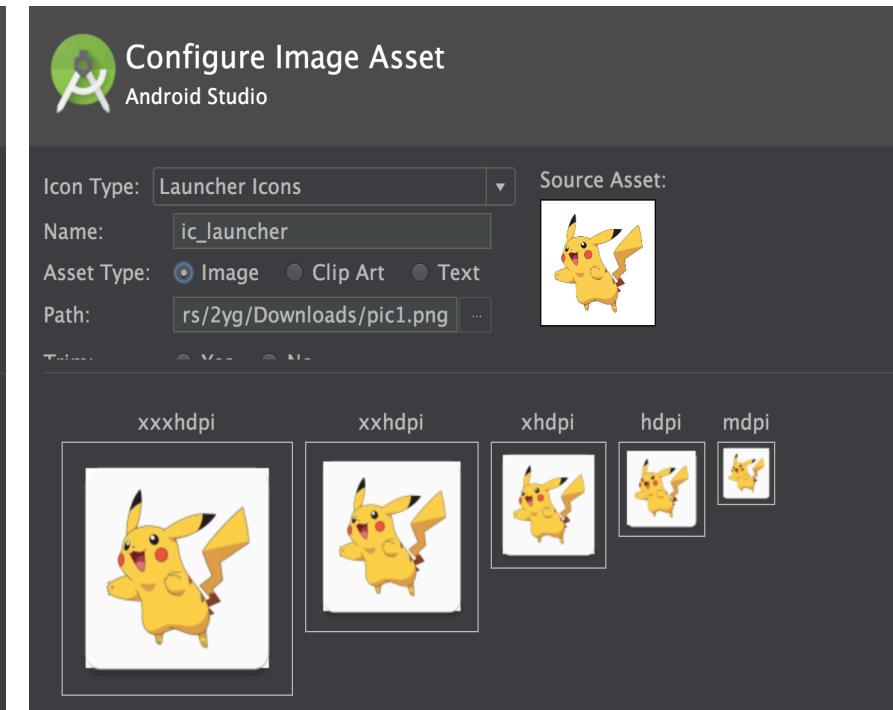
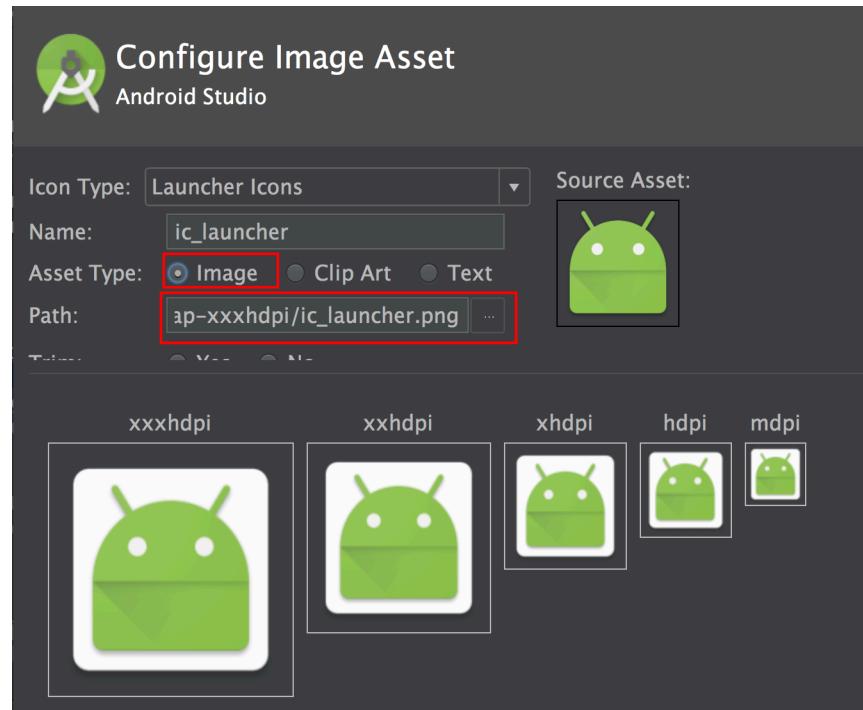
## 아이콘

- 휴대폰 바탕화면에 있는 앱의 대표 이미지
- res 폴더에 아이콘 이미지가 담겨 있는 폴더가 있음(mipmap)
- mipmap폴더에서 우클릭한 후 new->image assets 항목에서 새로운 아이콘 생성 가능



# 아이콘

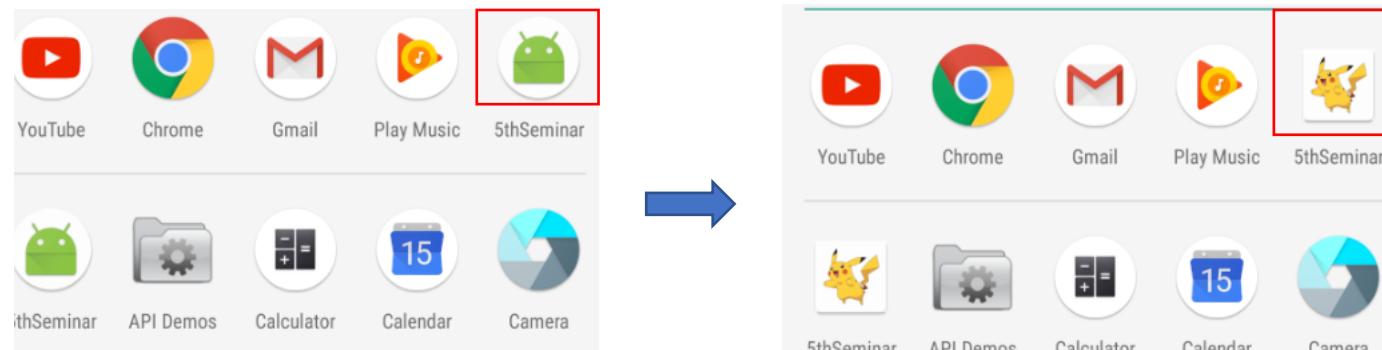
- Image를 클릭하고 아이콘으로 쓸 이미지를 Path를 통해 찾는다.



## 아이콘

- 완료하고 실행을 누르면 아이콘이 바뀌어 있음을 확인할 수 있다.  
( API25 이상은 Manifest에서 roundIcon 지워주세요)

```
<application>
    android:allowBackup="true"
    android:name=".ApplicationController"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
```



이런 것은..?

다음 단계로

이런 것은..?

다음 단계로

- 각이 둥글둥글
- 그라데이션

이런 것은..?

어떻게…?

이런 것은..?

디자이너님 잘 부탁드립니다 ^^7

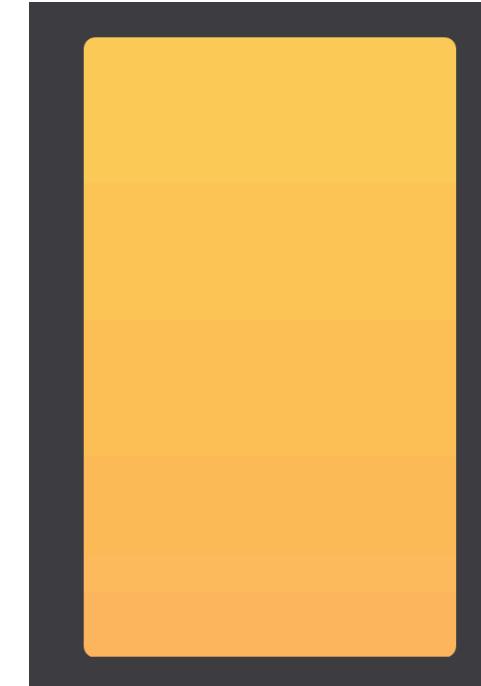
이런 것은..?

직접 만든다.

### 이런 것은..?

- drawable에서 직접 모서리 및 그라데이션을 만들고 해당 view의 background를 만든 drawable layout으로 쓴다.

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- background (위로) -->
    <item>
        <shape>
            <corners android:radius="12dp" />
            <gradient
                android:startColor="#ffcd54"
                android:endColor="#ffb65d"
                android:angle="270"/>
        </shape>
    </item>
</selector>
```



## 5 차 세 미 나

야무지게 달려보아요

쉬 는 시 간

잠시 쉬어업

## 5 차 세 미 나

02. Today2

## 해상도

- 하나의 이미지를 표현하는데 몇 개의 픽셀(pixel) 또는 도트(dot)로 나타내는 지의 정도를 알려주는 지표

## 해상도



은하노트 8호

- 6.3인치
- 2960 x 1440



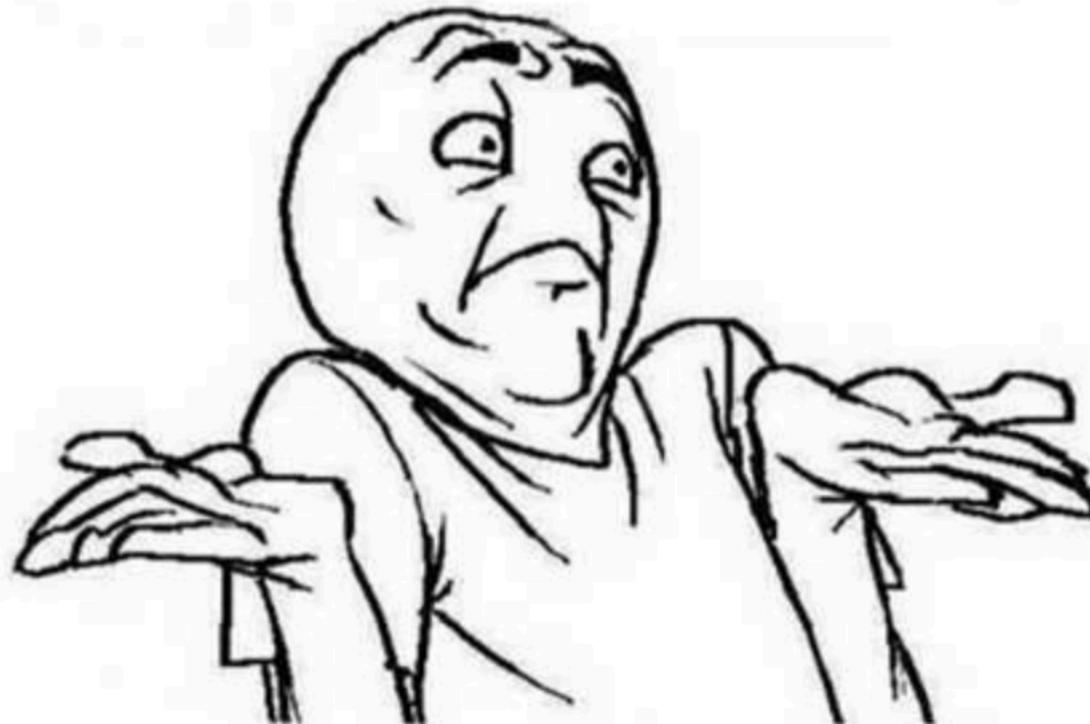
은하성 8호

- 5.8인치
- 2960 x 1440

해상도

그건 그렇고..

해상도



대체 뭐가 문제일까요?

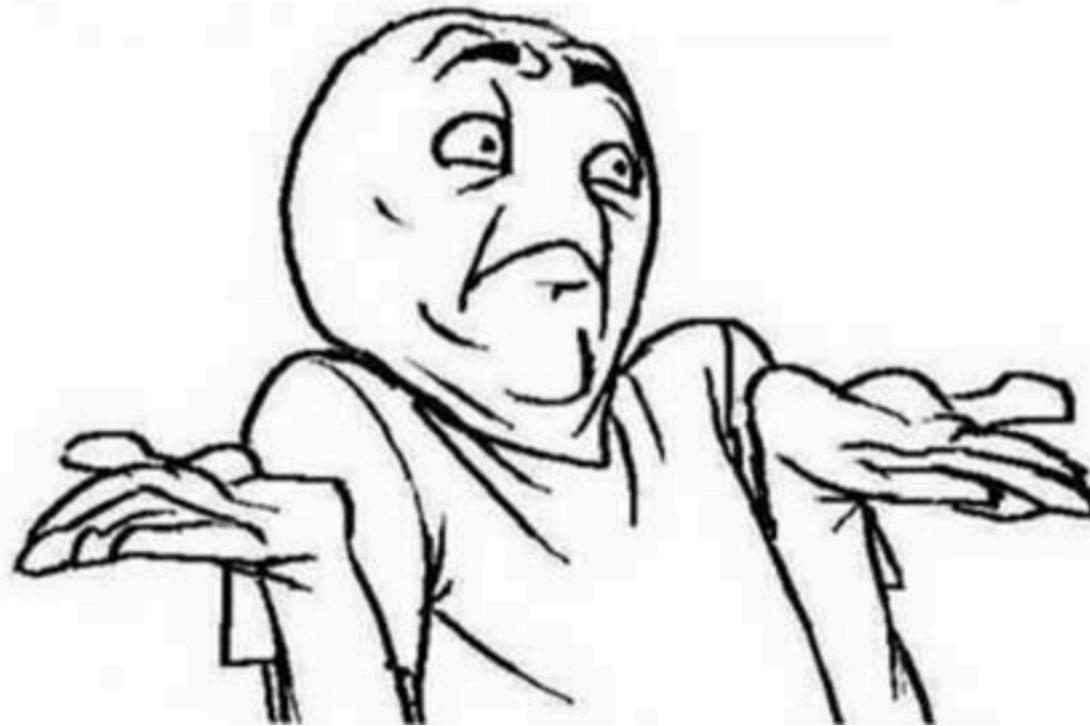
SOPT

### 해상도

| PHONE           | OPERATING SYSTEM | PHY SIZE " | WIDTH PX | HEIGHT PX |
|-----------------|------------------|------------|----------|-----------|
| Sony Xperia Z   | Android          | 5.0        | 1080     | 1920      |
| Samsung Nexus S | Android          | 4.0        | 480      | 800       |
| Nokia Lumia 925 | Windows          | 4.5        | 768      | 1280      |
| Nokia Lumia 920 | Windows          | 4.5        | 768      | 1280      |
| Nokia Lumia 900 | Windows          | 4.3        | 480      | 800       |
| Nokia Lumia 720 | Windows          | 4.3        | 480      | 800       |

기기별로 해상도가 다양

해상도



○ 짜?

SOPT'

해상도

RX

## px의 이해

- 픽셀은 고정값이다
- 그래서 크기가 변한다!

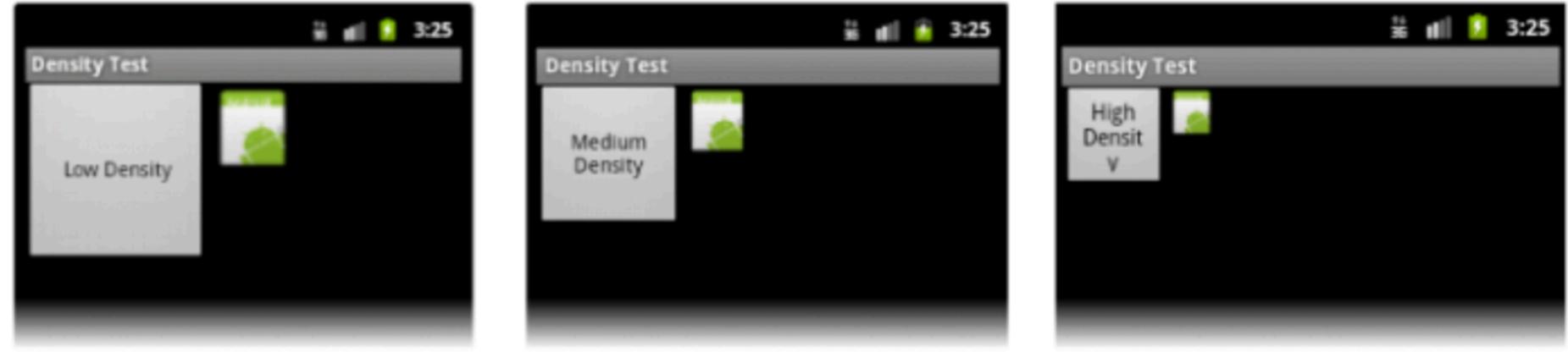
px의 이해

이게 뭔



소리죠?

### px의 이해



- 현재 회색 네모는 셋 다 같은 해상도
- 기기 해상도에 따라 네모 사이즈가 달라 보임.  
(100중 30과 500중 30과 1000중 30은 같은 30이라도 달라 보일 것)
- 그래서 값은 '고정'이어도 크기는 변한다

px의 이해



어떡하죠..?

px의 이해



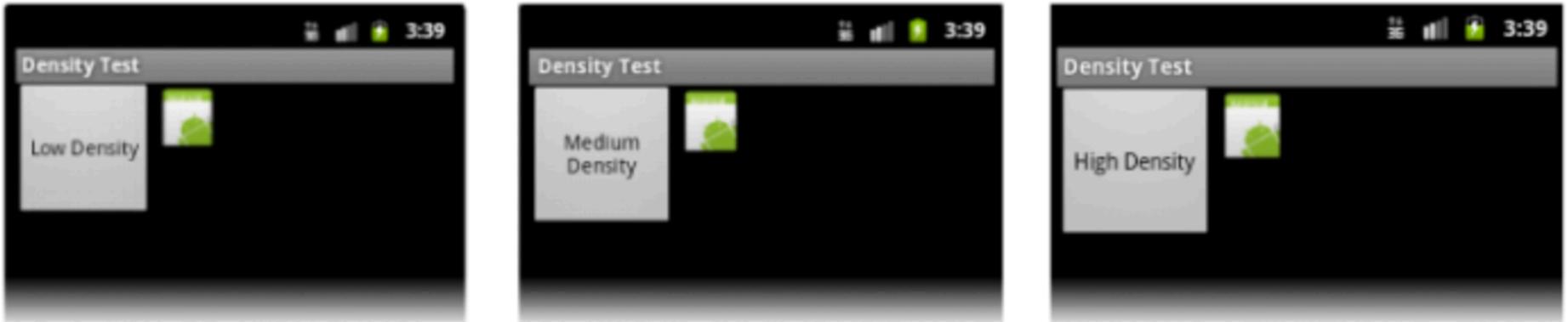
DP

### dp의 이해



- dp는 디바이스에 따라 값이 변한다
- 그래서 크기가 고정된다!

### dp의 이해



- 현재 회색 네모는 셋 다 같은 dp
- 기기 해상도에 따라 네모 사이즈가 같아보임.  
(디바이스 해상도에 따라 크기가 사이즈가 조정됐기 때문)
- 그래서 디바이스는 달라도 크기는 고정된다
- 이 때문에 디자이너가 여러 dp에 대응하게끔 이미지를 줍니다.

## dp의 이해

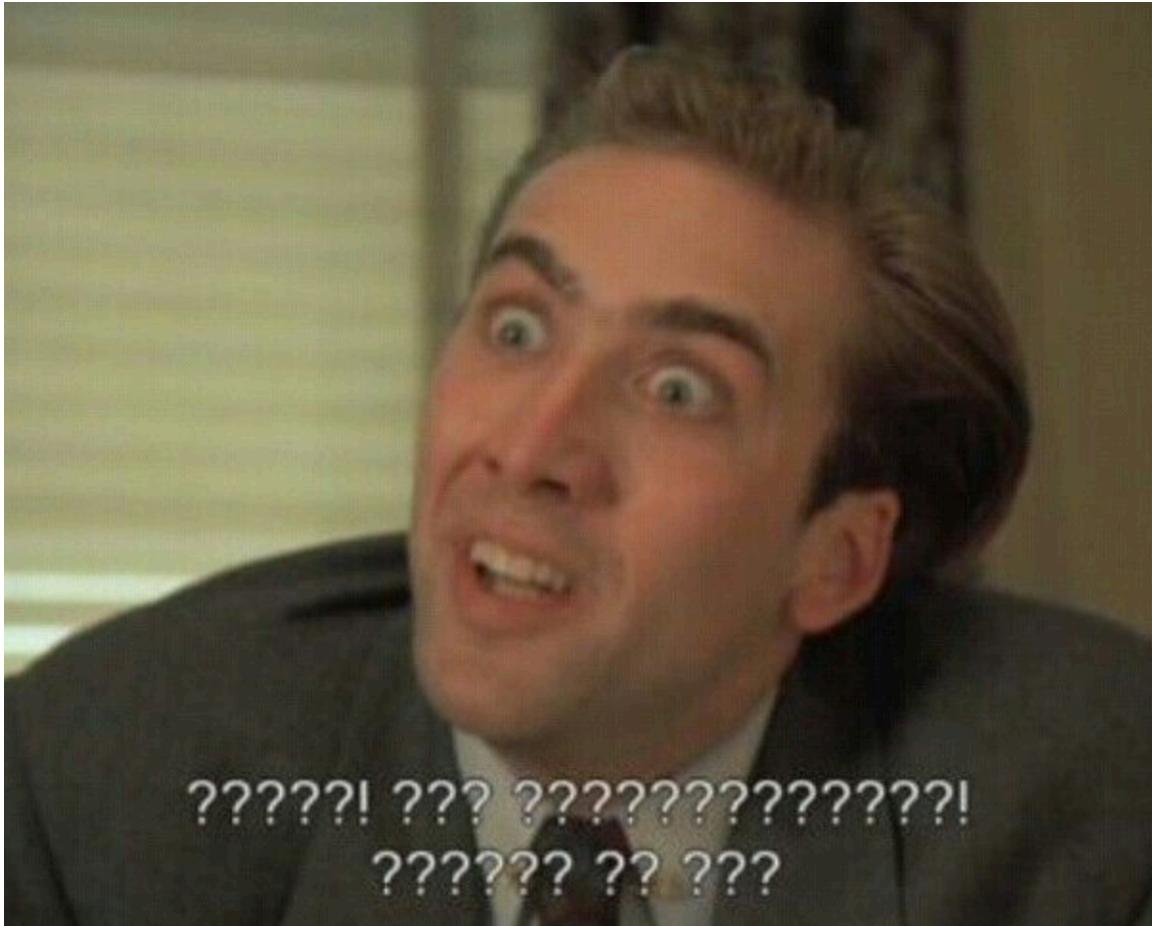


그러면 dp를 쓰면 되겠네요?

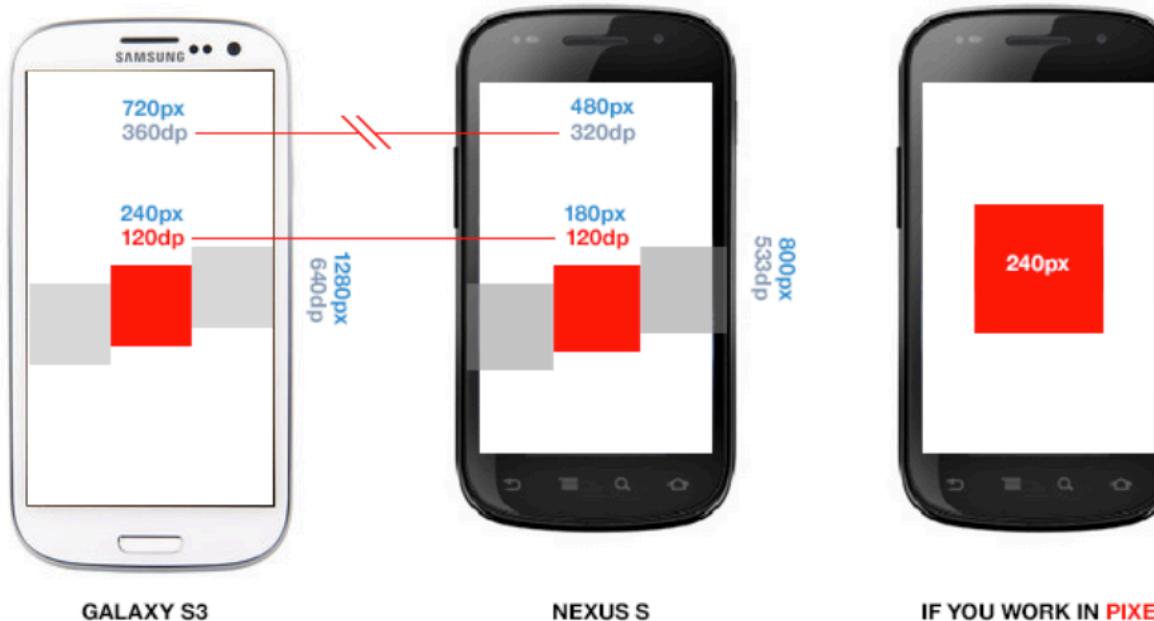
dp의 이해

아니요

## dp의 이해



### dp의 이해



- 같은 dp라 할지라도 화면 너비, 높이에 따라 차지하는 정도는 달라질 수 있음  
즉, **여백 문제**가 발생함
- 크기 중심으로 갈 지 **여백 중심**으로 갈 지 잘 판단해야 함.
- 우선적으로 **대표 기기**를 선정하는 것이 그나마 수치 오차를 줄일 수 있음.

## sp의 이해

- 글자에 많이 쓰이는 사이즈 단위
- 기본 원리는 **dp와 유사**
- 사용자가 설정한 **글꼴에 따라 환호**되어 사이즈 조정

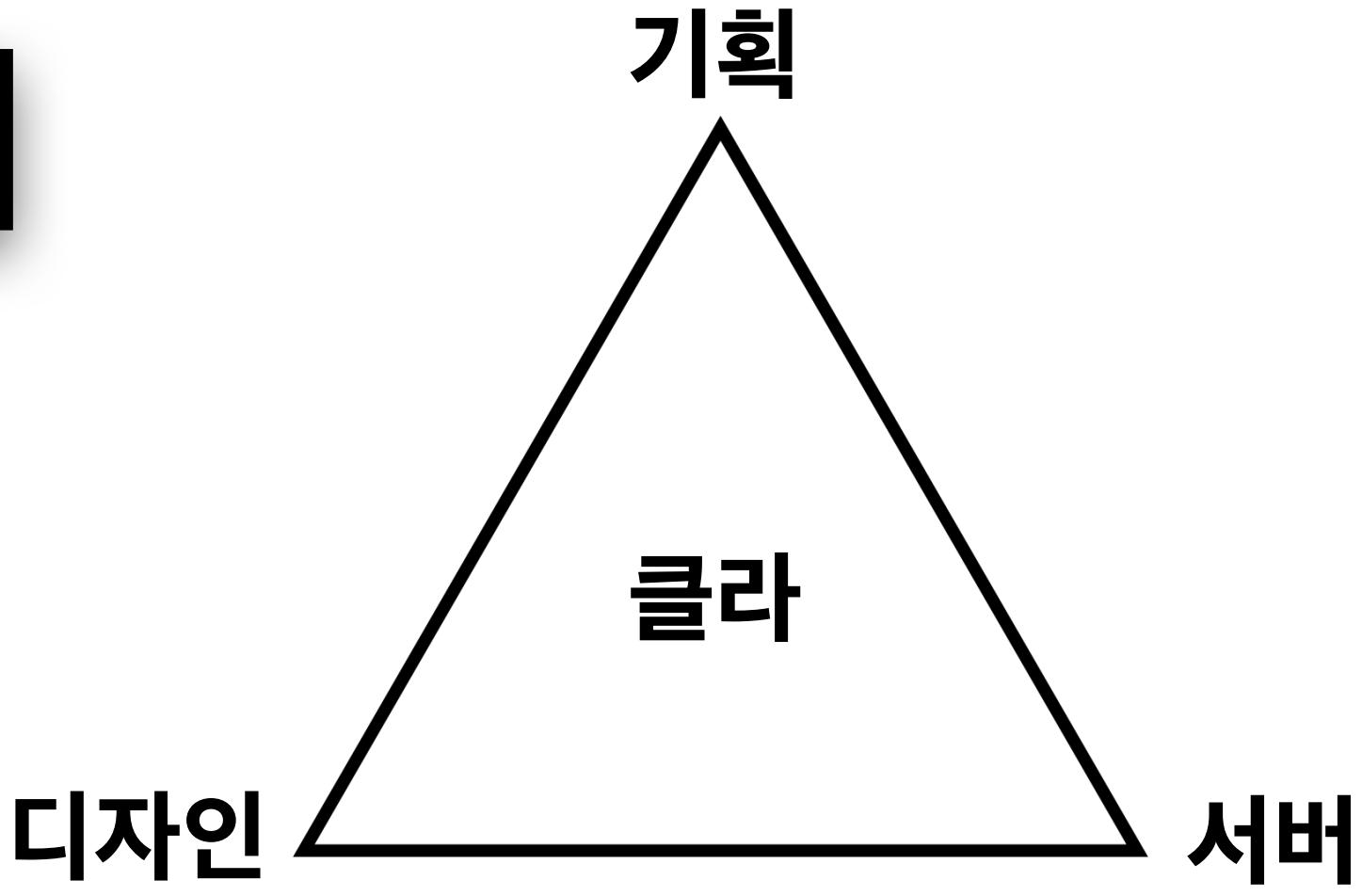
## 5 차 세 미 나

03. Today 3

디자이너와  
대화하기

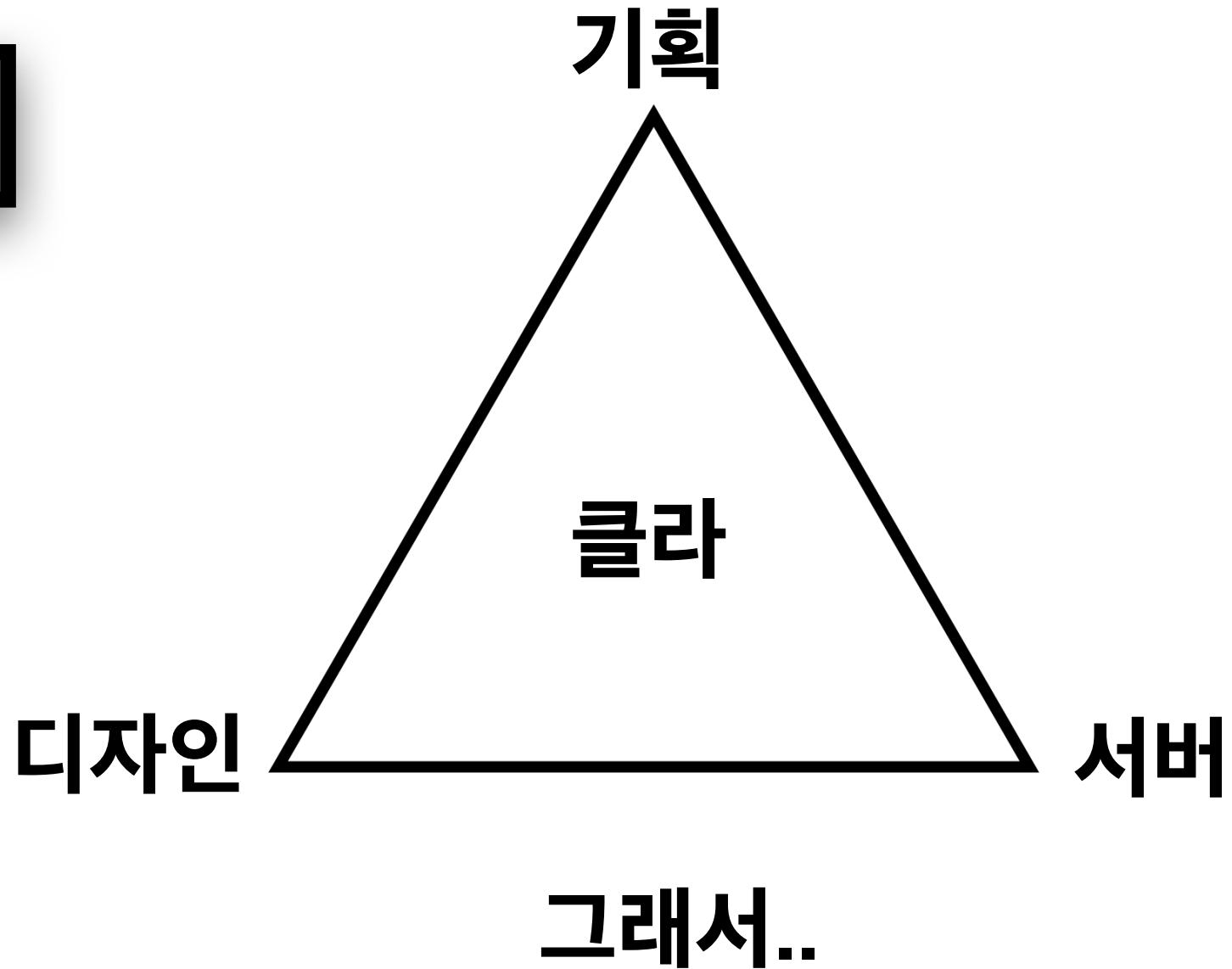
클라이언트는..

디자이너와  
대화하기



모두와 엮여 있습니다.

디자이너와  
대화하기



디자이너와  
대화하기



기획자한테도...

디자이너와  
대화하기



서버 개발자한테도..

디자이너와  
대화하기



디자이너한테도..

디자이너와  
대화하기

**극딜(extreme deal)을 당하게 됩니다.**

디자이너와  
대화하기

농담인 거 아시죠? ^~^

디자이너와  
대화하기

(아닐 수도)

디자이너와  
대화하기

디자이너?

### 디자이너와 대화하기

- ‘비교적’ 쉽게 설득이 가능
- **화면에 보여지는 것을 만드는 목적이 공통적이라 자주 소통하게 됨**
- 프로그램에 옷을 입혀주는 사람들이라 더욱 잘 모셔야 함.

디자이너와  
대화하기

어떻게?

### 디자이너와 대화하기

- 중요도를 묻는다.
  - 뷰 자체의 크기가 중요하면 뷰 위주로 값을 넣을 것.
  - 배치 및 공간이 중요하면 여백 위주로 값을 넣을 것.
- 기기들의 태생적 **한계를** 공유한다.
  - 기기가 너무 다양한 나머지 디자인 적 **차이와 한계가** 존재.
  - 디자이너가 **기준으로** 정한 기기 **이외의 기기들과**는 보이는 모습이 조금 **달라질 수 있음을** 충분히 인지시킬 것.
- 수치를 최대한 자세하게 요구한다.
  - 길이든 색상이든 일단 **최대한 빈틈없이** 알려달라 할 것.
  - 수치를 보면 **무엇이 중요한지**(자체 크기, 공간) 어느정도 판단 가능.
- 이미지 이름은 꼭 ‘**소문자, 밑줄(.) 그리고 숫자 조합**’으로 지어달라 한다.

디자이너와  
대화하기

화이팅~