

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

inno SOPT

안드로이드파트 1st

SHOUT OUR PASSION TOGETHER

SOPT

01

Hello Android

가볍게 보는 특징

최근 이슈

코틀린 '정말'
가.볍.게 살펴보기

02

View/Layout

View ?

View 기본 속성

Layout?

LinearLayout

RelativeLayout

03

실습

오늘 할 일

진짜 오늘 할 것

04

과제^{^~^}

복습

응용

심화

01

Hello Android



Android

Google에서 만든 모바일 운영체제

가장 많이 사용하는 OS

오픈소스

다양한 라이브러리 지원

다양한 플랫폼

개발 접근이 용이

빈번한 SDK 업그레이드

- 그만큼 이름도 다양(킷캣, 마쉬멜로우, 오레오 등등)

로봇이 귀엽다

최근 재미난 이슈가 발생

01 Hello Android

02
최근 이슈

SHOUT OUR PASSION TOGETHER
SOPT





Kotlin



- 자동 변환 기능을 허용하지 않는다(타입이 다르면 꼭 형변환을 해야한다)
- **NPE에 대응**(NPE = Null Pointer Exception)
- 뛰어난 **자바와의 호환성**
- 표현이 자바에 비해 훨씬 **간결**
- 문장 끝에 **세미콜론**을 찍지 **않는다**.

```
class Android{  
    int count;  
    String leader;  
    boolean check;  
    Android(int count, String leader, boolean check){  
        this.count = count;  
        this.leader = leader;  
        this.check = check;  
    }  
    public void setCount(int count){  
        this.count = count;  
    }  
    public void setLeader(String leader){  
        this.leader = leader;  
    }  
    public void setCheck(boolean check){  
        this.check = check;  
    }  
    public int getCount(){  
        return this.count;  
    }  
    public String getLeader(){  
        return this.leader;  
    }  
    public boolean getCheck(){  
        return this.check;  
    }  
}
```

```
data class Android(  
    var count : Int,  
    var leader : String,  
    var check : Boolean  
)
```

- 이렇게나 간결하다.
- 코틀린에 관한 내용은 세미나 하면서 알아가는 것으로..
(시간이 없어서요..8ㄱ8)

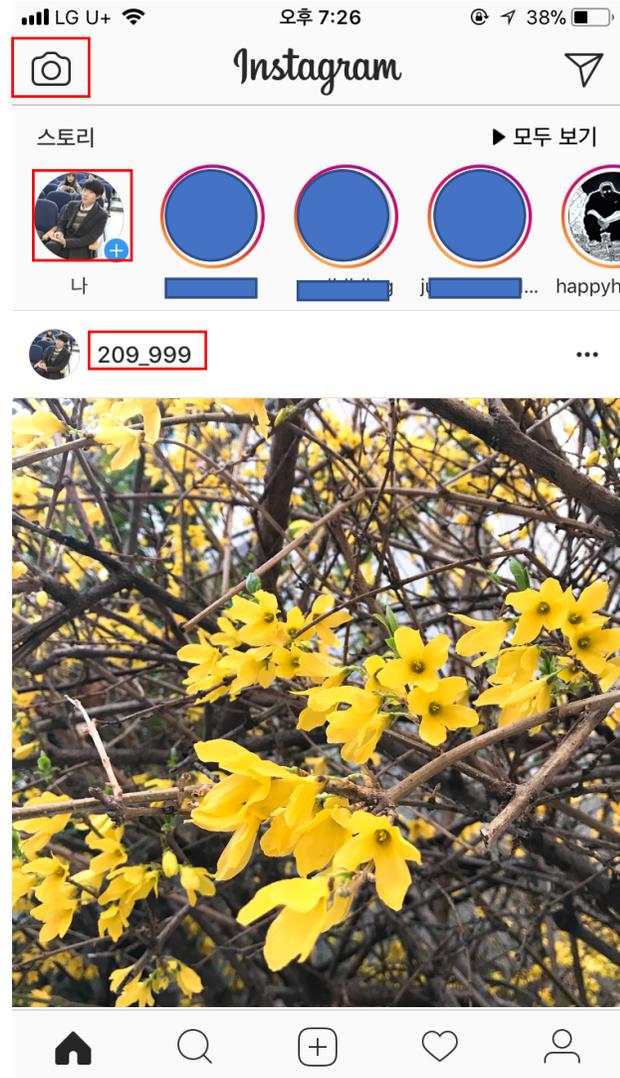
02

View/Layout

- 버튼, 텍스트, 이미지 등 모든 유저 인터페이스(UI)의 통칭
- 사용자 눈에 보이는 화면 구성 요소
- Activity 상에서 직접 구현할 수 있지만 더 쉽고 정교하게 하기 위해 XML을 사용

02 View/Layout

01
View?



```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/inno_sopt"
    android:background="#000000"
    android:layout_margin="10dp"
    android:padding="10dp"
    android:visibility="visible"
/>
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/inno_sopt"  
    android:background="#000000"  
    android:layout_margin="10dp"  
    android:padding="10dp"  
    android:visibility="visible"  
>
```

- **layout_width** -> 뷰의 너비(가로 길이)
layout_height -> 뷰의 높이(세로 길이)
- **wrap_content** -> 내용에 길이(혹은 크기)에 맞게 조절
match_parent -> 부모 뷰에 꽉 채워진 크기(여백 제외)
위 두 가지가 아닌 경우에는 직접 설정

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/innosopt"
    android:background="#000000"
    android:layout_margin="10dp"
    android:padding="10dp"
    android:visibility="visible"
/>
```

- kotlin에서 참조하기 위한 **View의 이름**

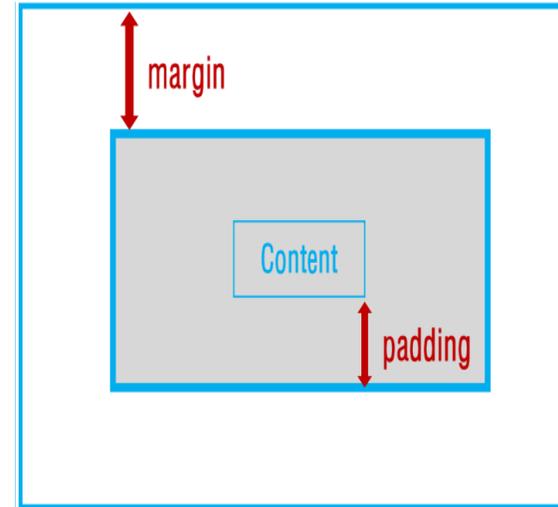
```
textView = findViewById(R.id.uni_sopt) as TextView
```

- **+id@/아이디명**

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/innosopt"
    android:background="#000000"
    android:layout_margin="10dp"
    android:padding="10dp"
    android:visibility="visible"
/>
```

- 배경 지정
 - 색 뿐만 아니라 **이미지**도 지정 가능

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/innosopt"  
    android:background="#000000"  
    android:layout_margin="10dp"  
    android:padding="10dp"  
    android:visibility="visible"  
>
```



- View의 여백 설정
- margin -> 부모 혹은 외부 View와의 여백
- padding -> View 외곽과 내용 사이의 여백

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/inno_sopt"
    android:background="#000000"
    android:layout_margin="10dp"
    android:padding="10dp"
    android:visibility="visible"
/>
```

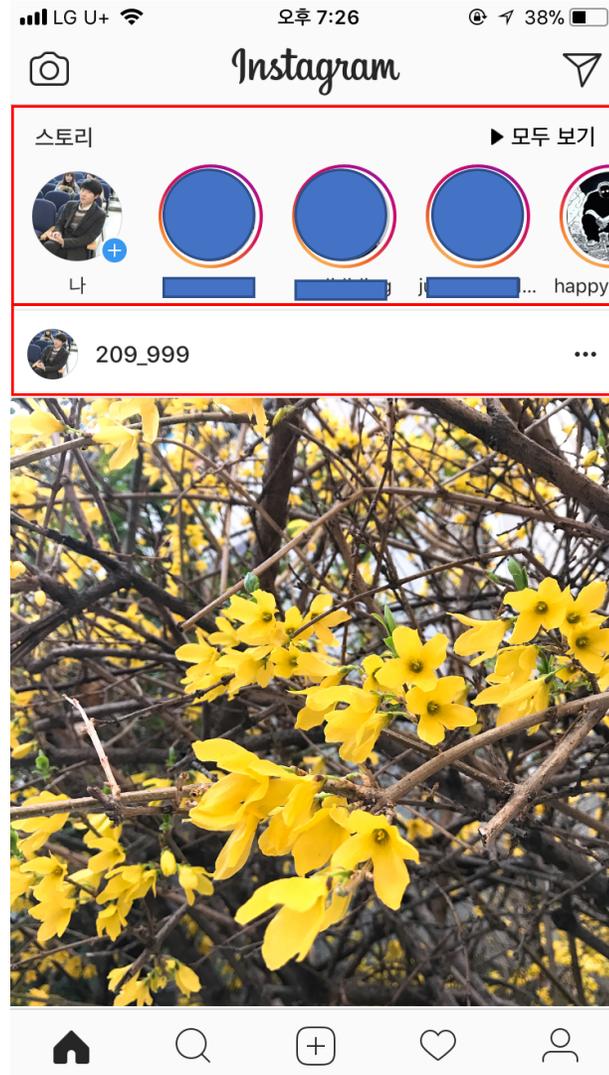
- 화면에 **보여지는 속성**
 - **visible** : 공간도 차지하고 보임
 - **invisible** : 공간은 차지하는데 보이지는 않음
 - **gone** : 공간도 차지하지 않고 보이지도 않음

- **View**들이 배치 될 그룹
- **Layout**에 따라 **배치**되는 **방식**이 달라짐
- **LinearLayout, RelativeLayout,**
FrameLayout, ConstraintLayout 등이 존재
- XML을 통해 작성

02 View/Layout

03
Layout?

SHOUT OUR PASSION TOGETHER
SOPT

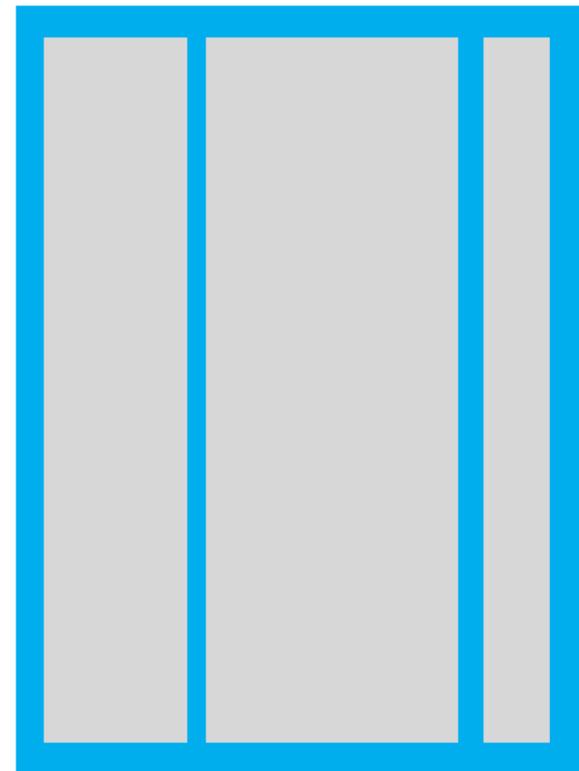
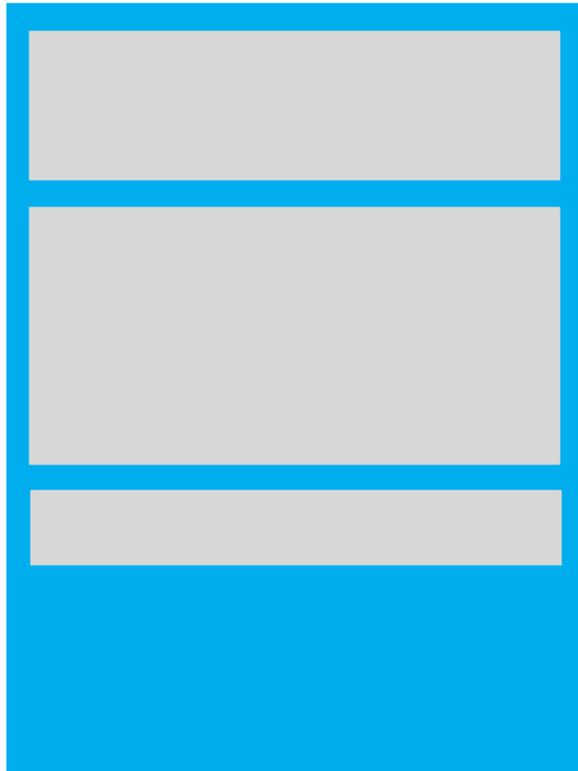


- **View**들의 **선형 배치**를 제공하는 Layout
- **수직**방향, **수평**방향으로 배치 가능
- 수직 배치시 다음 뷰는 **가장 아래 뷰를 기준으로 밑에**
수평 배치시 다음 뷰는 **가장 오른쪽 뷰를 기준으로 오른쪽에** 배치
- 배치 기준이 정해져 있어서 직접 정해주지 않아도 됨
- 특정 뷰 기준으로 배치하는 것은 불가
- 뷰위에 뷰를 쌓는 것 역시 불가
- 계단식 배치에 용이

02 View/Layout

04
LinearLayout

SHOUT OUR PASSION TOGETHER
SOPT

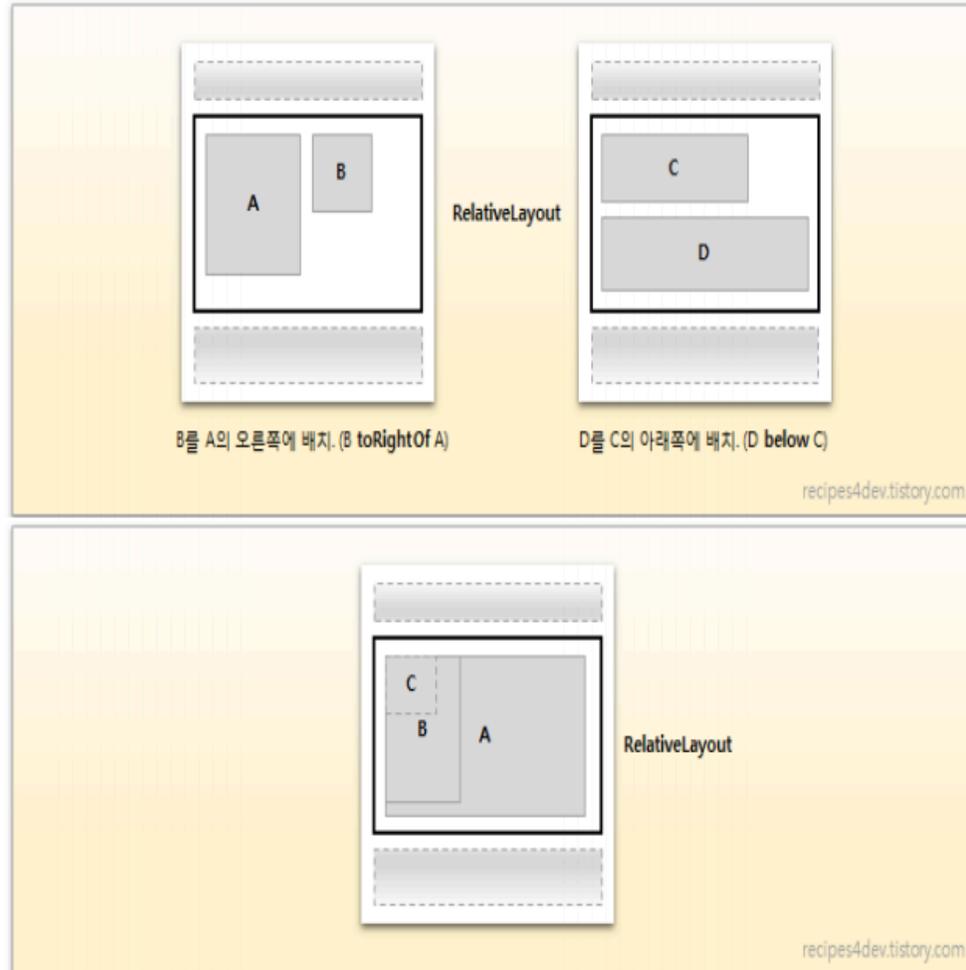


- **orientation**
 - **뷰 배치에 방향성 부여(horizontal, vertical)**
- layout_gravity
 - 부모 컨테이너 안에서 뷰의 위치를 정함
- gravity
 - 뷰 안에서 내용물의 위치를 정함
- layout_weight
 - 뷰의 가중치로 비율을 통해 너비 혹은 높이 길이를 정함

- **View**들의 **상대적 배치**를 제공하는 Layout
- **특정 뷰를 기준으로** 다른 뷰 배치
- 기준을 정해주지 않으면 왼쪽 상단 위에 쌓인다.
- 뷰를 겹칠 수 있다.
- 원하는 배치를 하기 위해선 기준을 직접 정해줘야 함
- 복잡한 뷰에 용이

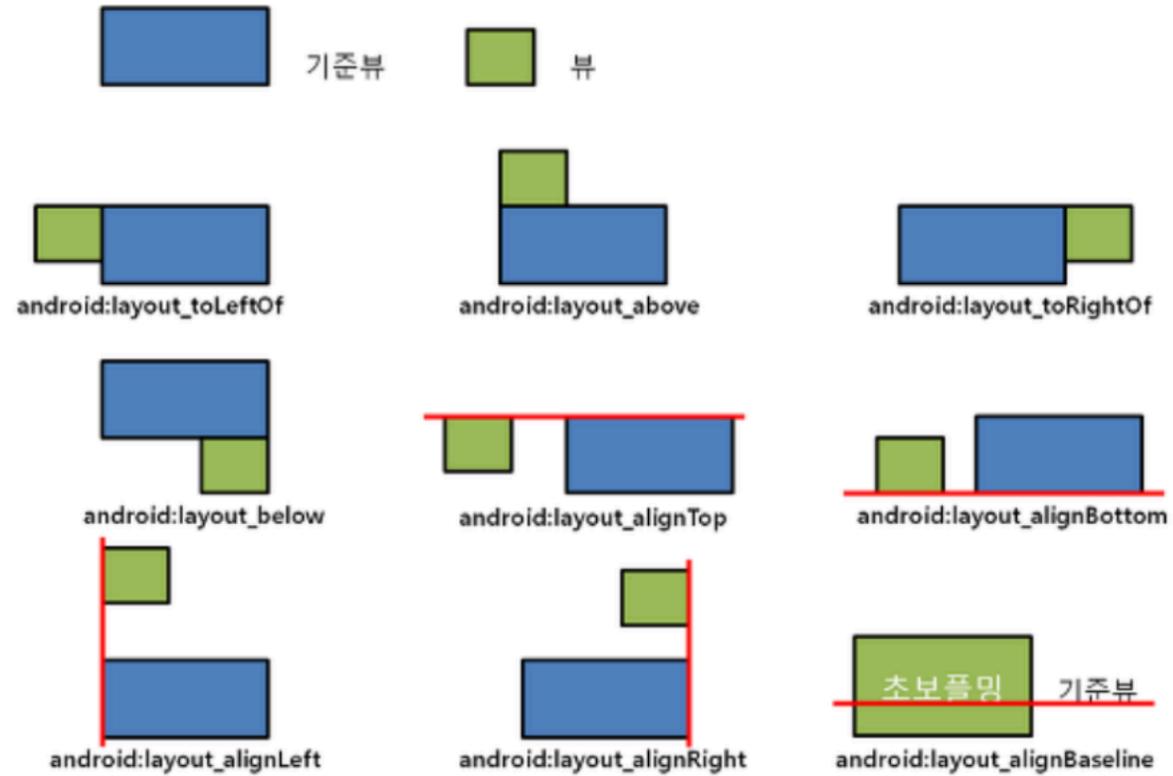
02 View/Layout

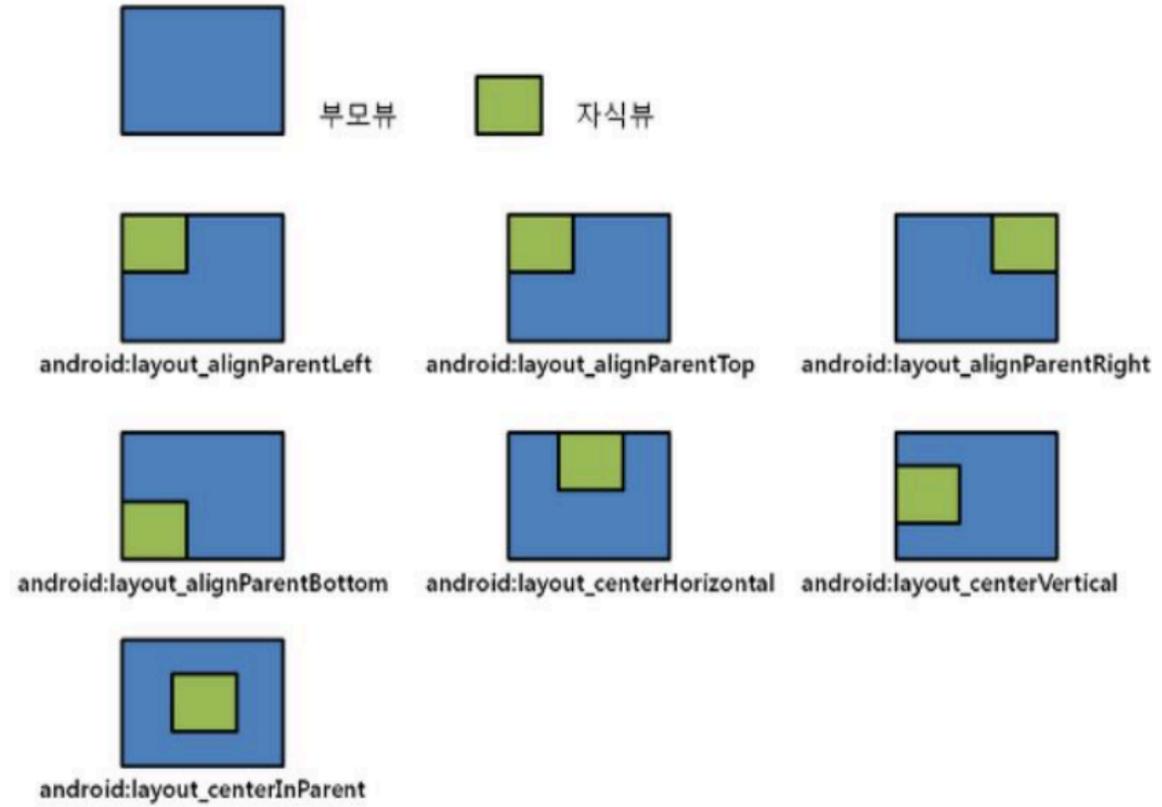
05 RelativeLayout



02 View/Layout

05 RelativeLayout





03

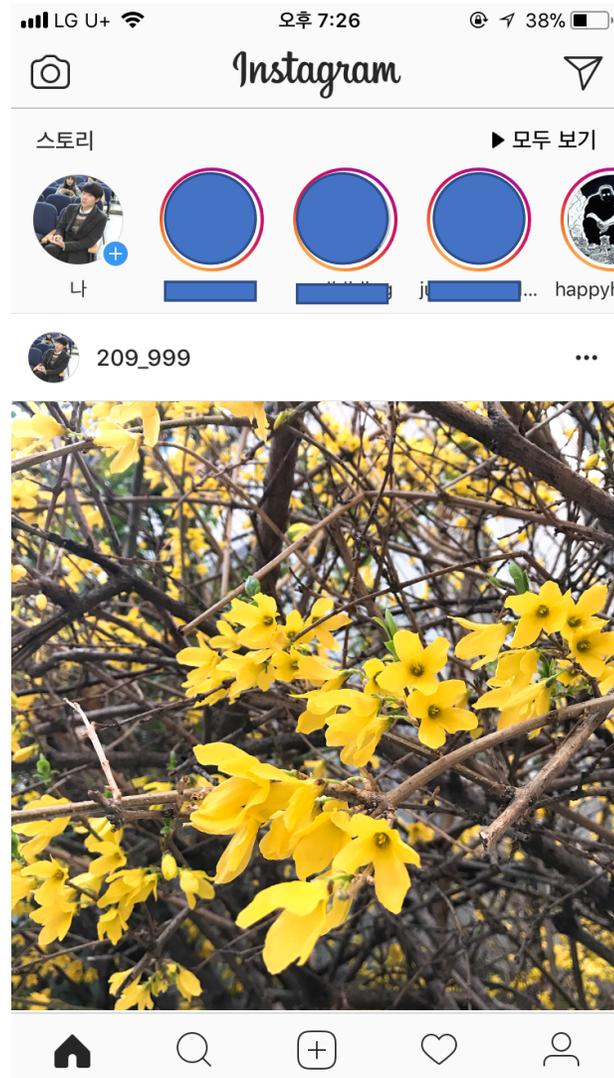
실습

대략 눈치 채셨겠지만..

03 실습

01
오늘 할 일(세상..)

SHOUT OUR PASSION TOGETHER
SOPT



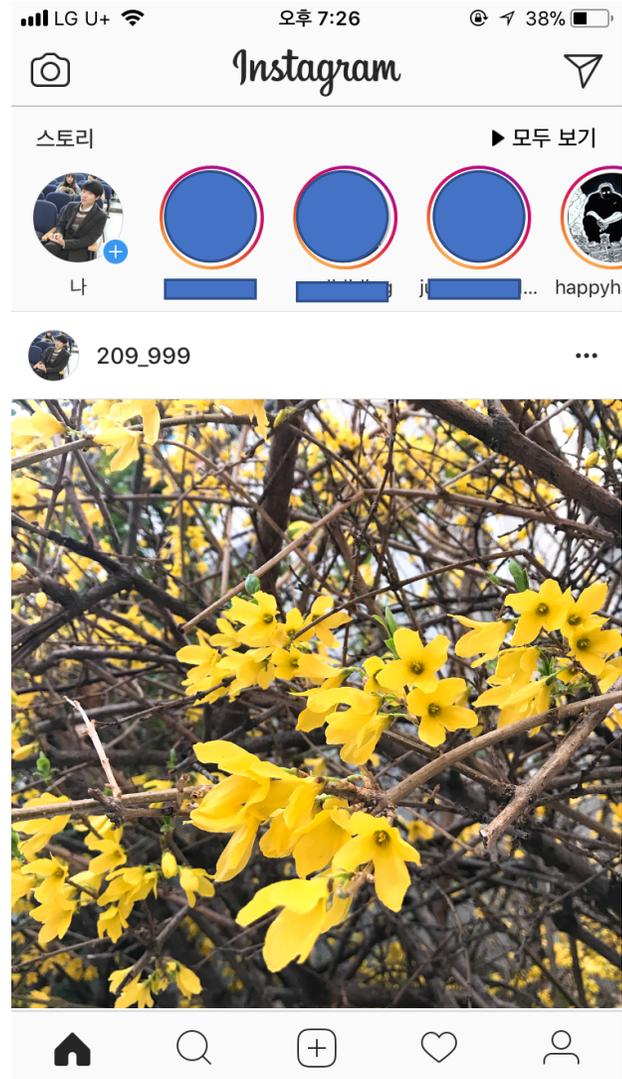
인스타그램을 만들 것입니다.

03 실습

01

오늘 할 일(놀래라..)

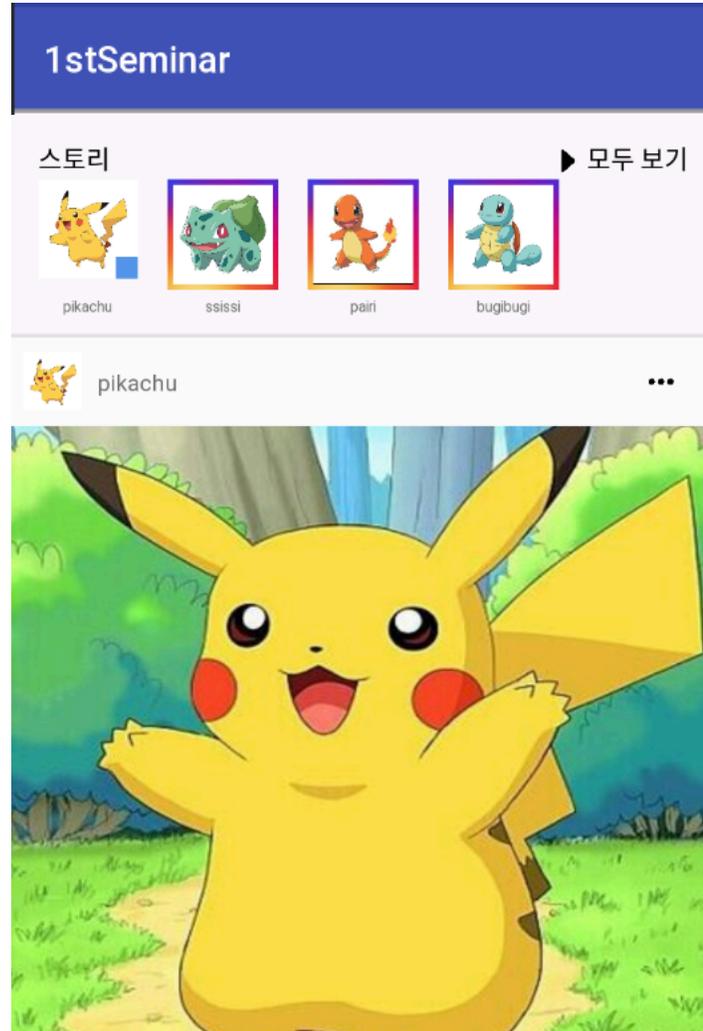
SHOUT OUR PASSION TOGETHER
SOPT



물론 완전 카피는 아니지만..

03 실습

02
진짜 오늘 할 것(두둥..!)



딱 요정도..!(간단하쥬?^~^)

달려봅시다..!

03
실습

02
진짜 오늘 할 것(두둥..!)

SHOUT OUR PASSION TOGETHER
SOPT

그전에..!

잠시 쉬어여 😊

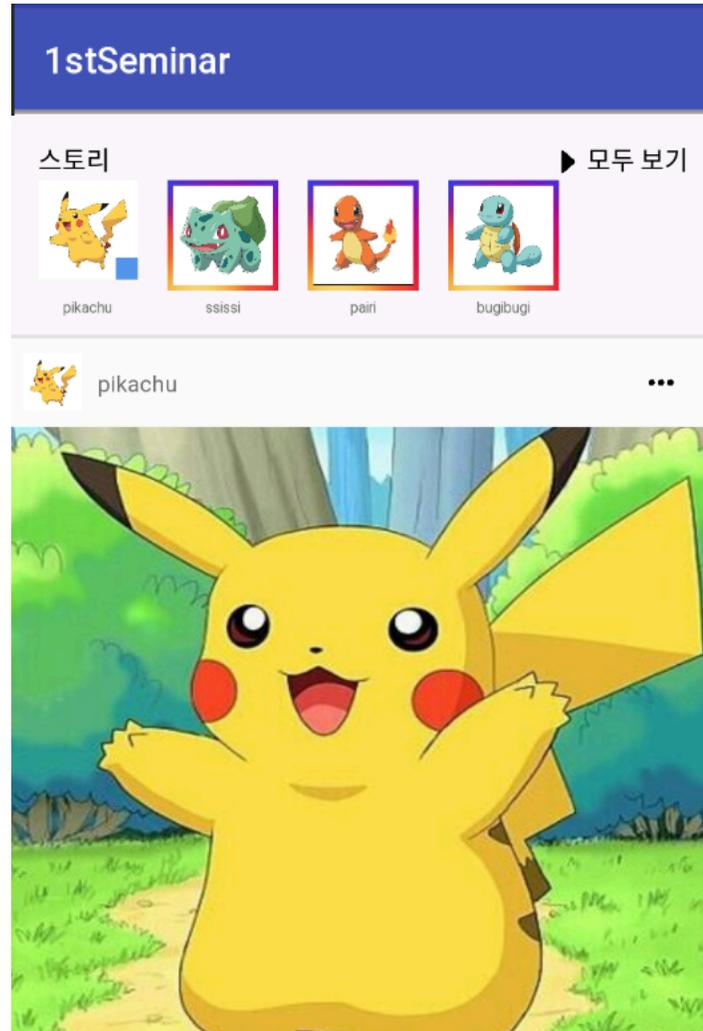
04

과제 ^~^

04 과제

01
복습

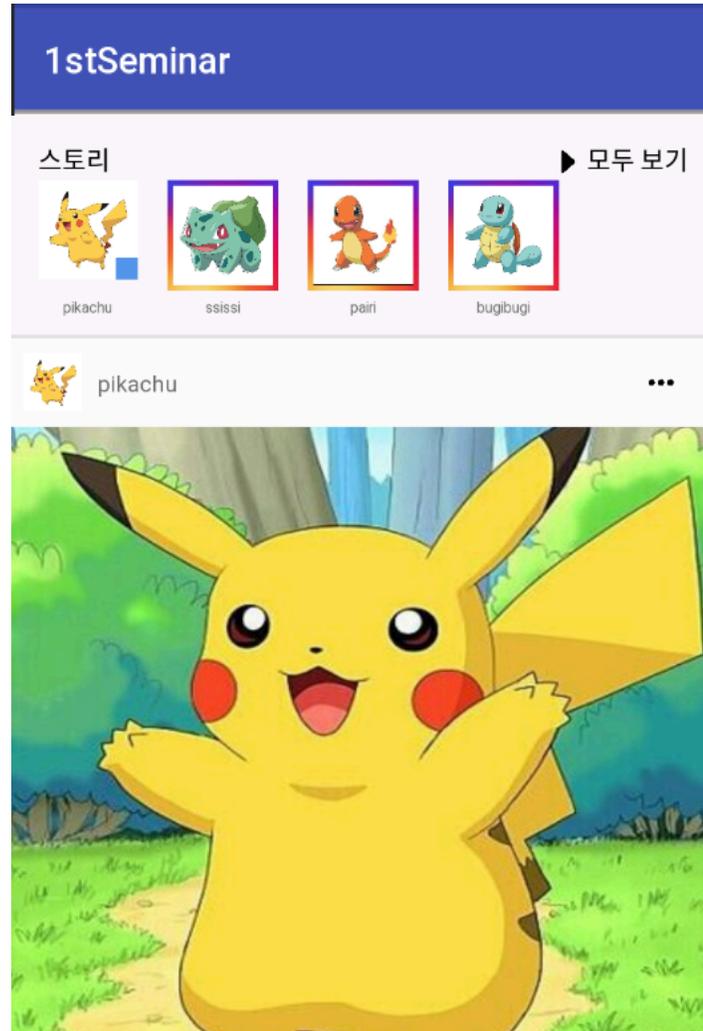
SHOUT OUR PASSION TOGETHER
SOPT



혹시 다 못만드셨다면 만들어주세요!
만약 다 만드셨다면 일단 과제 끝..!

04 과제

02
응용

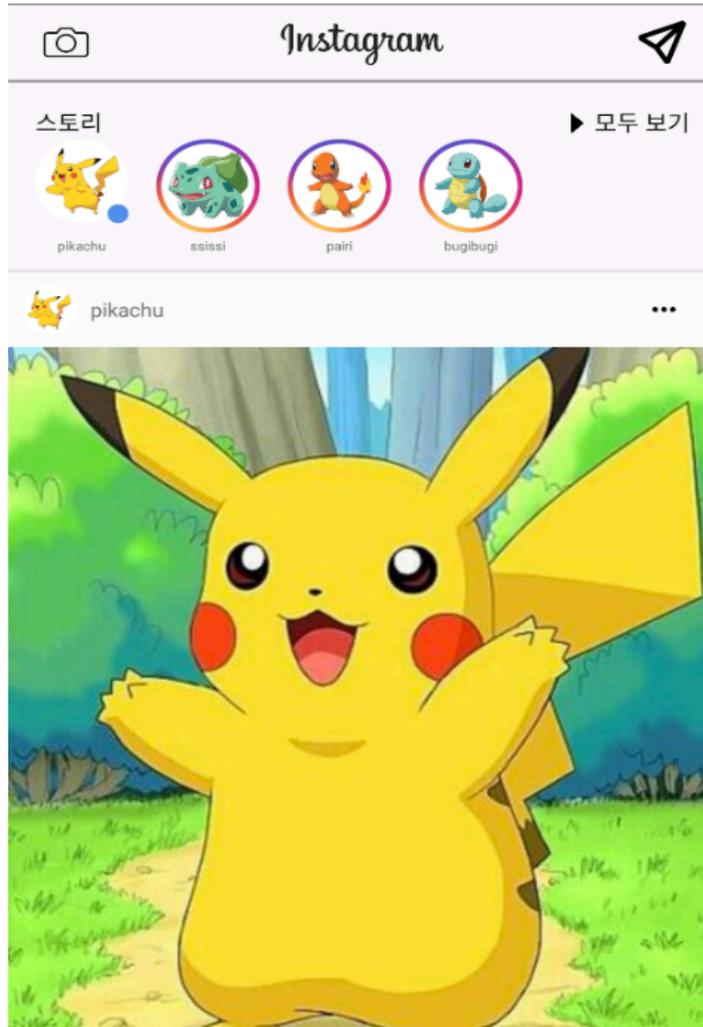


피드를 여러개 만들고 스크롤도
되게 해봅시다!(scrollview이용)

04 과제

03
심화

SHOUT OUR PASSION TOGETHER
SOPT



상단바도 실제 인스타그램처럼!
그리고 프로필 이미지는 동그략게!

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

inno SOPT

끼움~~

SHOUT OUR PASSION TOGETHER
SOPT