

Hello
Android!



How to?

UI (User Interface) 개발은 XML
동작이나 기능 개발은 Java or Kotlin



Java



Kotlin

Java



대표적인 객체지향 언어 C언어의 문법과 대체로 유사

- ▶ 캡슐화
- ▶ 상속
- ▶ 추상화
- ▶ 다형성

```
1 #include <stdio.h>
2
3 int main() {
4     char hi[100];
5
6     scanf("%s", hi);
7
8     printf("%s", hi);
9 }
```

c

```
public class hello {
    public static void main(String[] args) {
        String hi;

        Scanner scanner = new Scanner(System.in);

        hi = scanner.next();

        System.out.println(hi);
    }
}
```

Java



▶ 변수선언

C언어 char 배열 = String
배열을 선언하는 방법은 두 가지 방법

```
String A="AB*CD*EF";
```

```
int[] i_array;  
int i_array[];
```

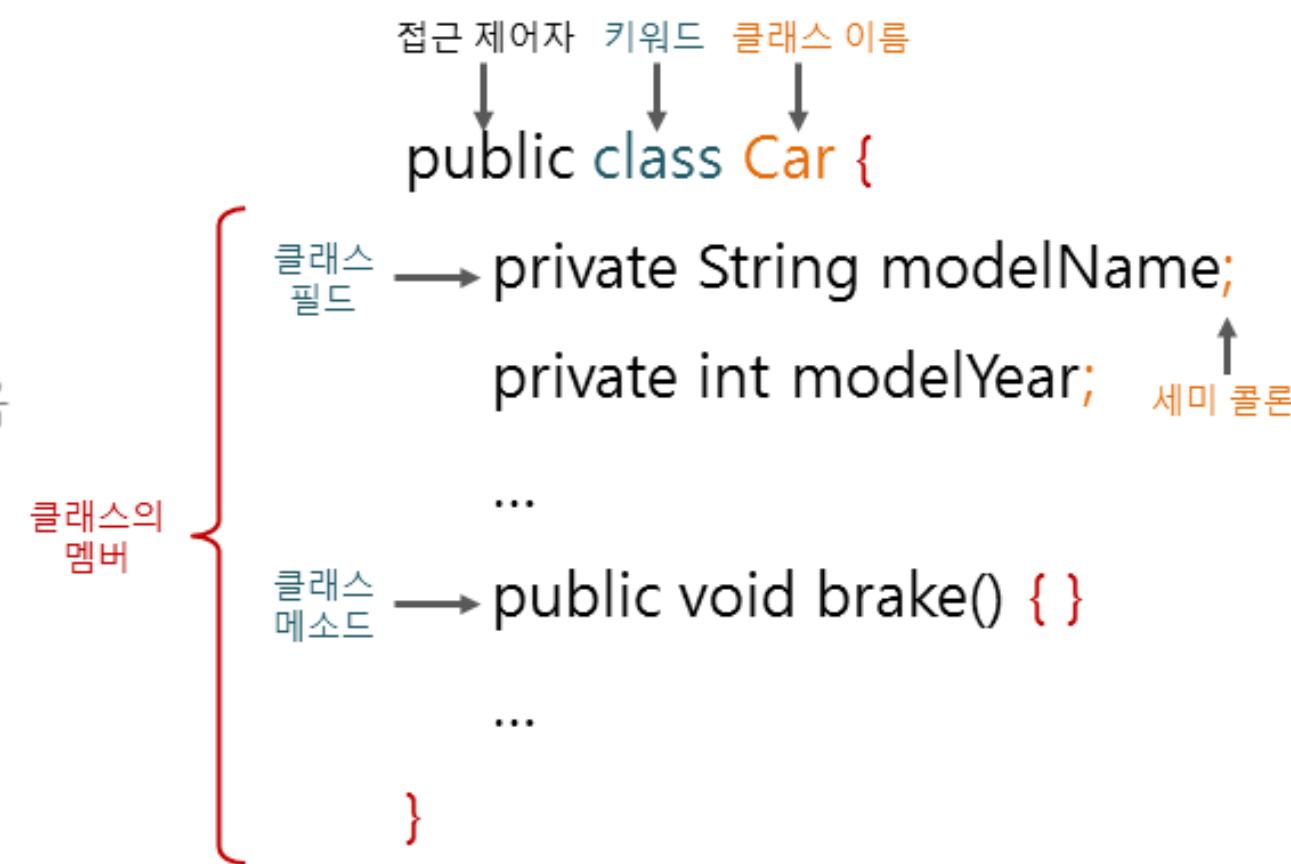
▶ 메소드

소스 코드로 묶어놓고 필요(호출)에
따라 동작하는 기능 함수라고 생각

```
public static 리턴 타입(자료형 또는 void) 메소드명() {  
    실행할 코드 작성:  
    return; // 리턴 타입이 void일 경우 return 없음  
}
```

▶ 클래스

어떠한 객체의 변수, 메소드의 집합
동일한 속성과 행위를 수행하는 객체의 집합



Kotlin

[구글] 안드로이드 공식 언어 채택
통합 개발 환경

- ▶ 간결성 ▶ 안정성
- ▶ 도구친화성 ▶ 상호운용가능성

```
public class hello {  
    public static void main(String[] args) {  
        String hi;  
  
        Scanner scanner = new Scanner(System.in);  
  
        hi = scanner.next();  
  
        System.out.println(hi);  
    }  
}
```

Java

```
fun main(args:Array<String>) {  
    val hi = readLine()  
    print("$hi")  
}
```

Kotlin

▶ 클래스

코틀린에서 클래스는 class 키워드를 사용해서 정의합니다.

```
class Parent(a : Int) {  
}  
  
class Example(age : Int) : Parent(age){  
    constructor(name: String, age : Int) : this(age)  
        print("$name $age")  
    }  
    init {  
        print(20)
```

▶ 메소드

기본적으로 메소드를 선언할때 fun이라는 키워드를 사용한다.

```
fun main(args:Array<String>){  
    println("Hello, World!")  
}
```

▶ var&val

var = 읽기/쓰기가 가능한 일반 변수
val = 읽기만 가능한 final 변수

```
fun main() {  
    val number = 10  
    var language = "Korean"  
    val secondNumber :Int=20  
    ...
```



▶ 타입 추론

`var(혹은 val)` 변수이름 : 변수타입 = 값
변수타입은 생략할 수 있다.

```
val name:String = "daisy"  
val age = 10 // 변수 타입은 생략할 수 있다.
```

▶ null

?를 붙이면 null을 할당할 수 있는 프로퍼티이고.
?가 붙지 않으면 null이 허용되지 않는 프로퍼티를 의미

```
var nullable: String? = "nullable"  
var nonNullable: String = "non-Nullable"
```

What is Android?

developed by Android Studio

Android는 리눅스 기반

주로 Android Studio를
이용하여 개발

MainActivity 와
onCreate()

XML과 UI



▶ Intent

안드로이드간 데이터를 전달하기 위한 객체

```
public void mOnClick(View v) {  
    Intent intent = new Intent(this, SubActivity.class);  
    startActivity(intent);  
}
```

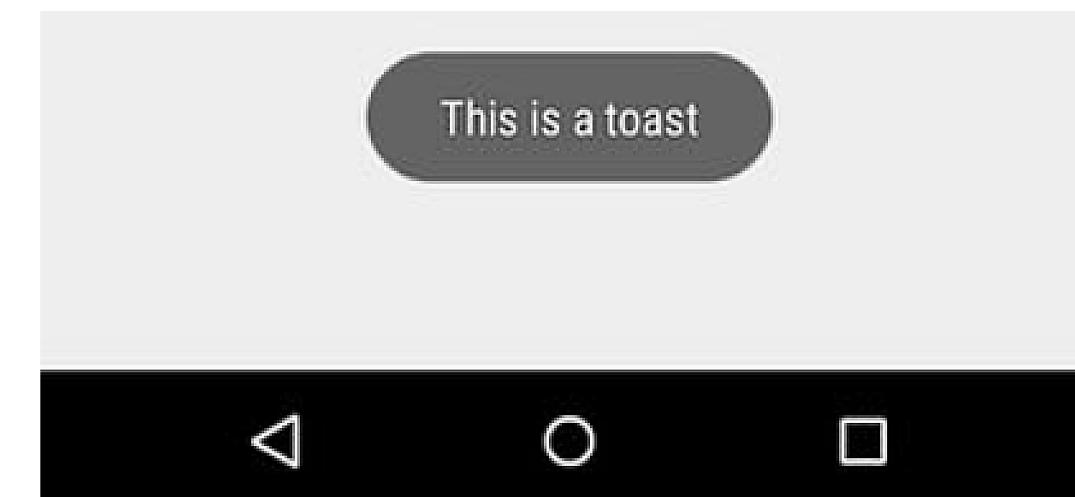
startActivity 메서드를 사용하여

인수를 호출할 대상을 지정하는 Intent 객체를 전달

▶ Toast

간단한 메시지를 짧은 시간에 팝업 형태로 띠운다.

```
public void ClickHandler(View view)  
{  
    Toast myToast = Toast.makeText(this.getApplicationContext(),"", Toast.LENGTH_SHORT);  
    myToast.show();  
}
```



▶ textView

화면에 텍스트를 표시하는 기능을 담당
텍스트 출력 기능을 가진 기본 위젯



```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



▶ **textView**

화면에 텍스트를 표시하는 기능을 담당
텍스트 출력 기능을 가진 기본 위젯

1 **textColor**

색상과 투명도 설정 가능

2 **textSize**

Text의 크기 설정 가능

3 **textStyle**

색상과 투명도 설정 가능

▶ EditText

텍스트 입력을 받는 뷰
힌트를 사용할 수 있다.



```
<EditText  
    android:id="@+id/editTextTextPersonName"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="내용을 입력해주세요"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```



▶ EditText

텍스트 입력을 받는 뷰
힌트를 사용할 수 있다.

1 사용자가 텍스트를 입력할 수 있는
기능을 제공하는 뷰

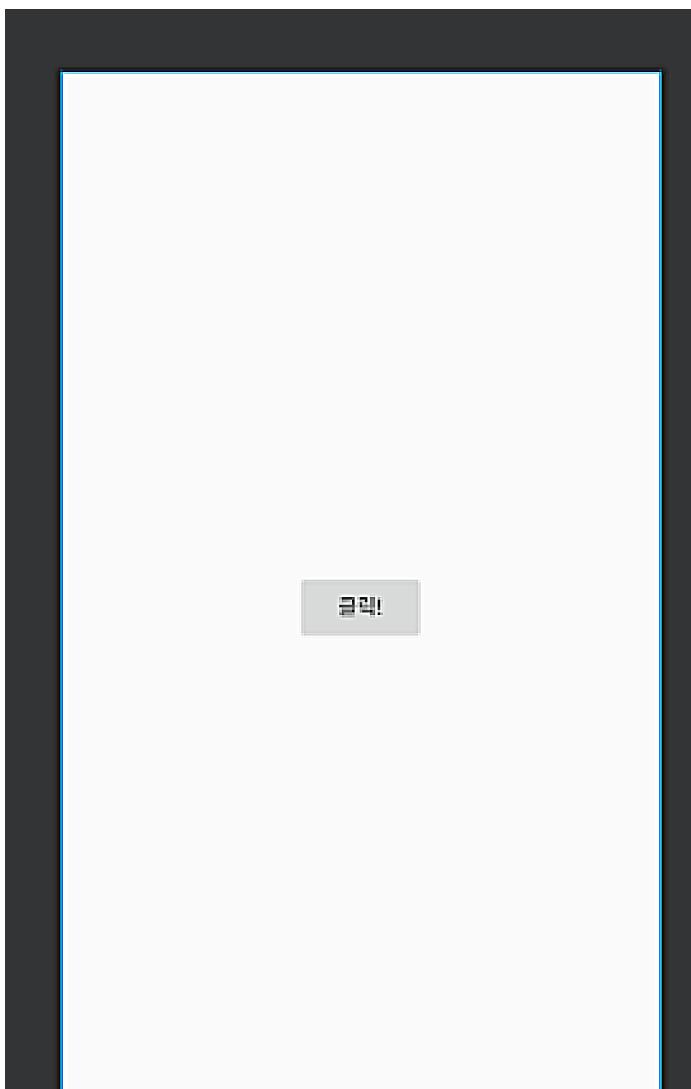
2 텍스트를 받아오거나 출력하고자
하는 텍스트를 출력

3 **hint**

텍스트가 비어있을 상태시 화면에
출력될 텍스트

▶ Button

사용자의 터치 이벤트를 처리
TextView 정의와 속성 사용 가능



```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="클릭!"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```



▶ Button

사용자의 터치 이벤트를 처리

TextView 정의와 속성 사용 가능

1 화면을 클릭했을 때 발생하는
클릭이벤트를 처리하는 뷰

2 UI 구성시 가장 많이 사용하는 뷰